



Département d'électronique

Laboratoire de **D**ispositifs de **C**ommunication et de **C**onversion **P**hotovoltaïque

Mémoire de Magister en Electronique

Option : Electricité solaire

Présenté par :

BENDIB Douadi

Ingénieur d'état en électronique, Université de M'SILA

THEME

Etude et réalisation d'une commande MLI on-line sur circuit FPGA

Soutenue publiquement devant le Jury composé de :

M. M.HADDADI	Professeur	ENSP	Président
M. C.LARBES	Maitre de conférences	ENSP	Rapporteur
M. H.BOUSBIA-SALAH	Maitre de conférences	ENSP	Examineur
M. M.S AÏT CHEIKH	Maitre de conférences	ENSP	Examineur
M. A.MALEK	Maitre de recherches	CDER	Examineur

ملخص:

الهدف من هذا العمل هو دراسة و إنجاز مغير سرعة لمحرك لا تزامني ثلاثي الأطوار ، طريقة التحكم المستعملة هي تقنية تعديل عرض النبضة (MLI) مع حذف توافقيات الإشارة والتحكم في المركبة الأساسية. لذلك قمنا بمحاكاة التقنية المستعملة باستعمال البرنامجين MATLAB و PSIM من أجل إثبات نجاعتها في حساب زوايا التبديل بطريقة تضمن التحكم في سرعة المحرك . بعد ذلك قمنا ببرمجة هذه التقنية على دارة مبرمجة من نوع FPGA XILINX باستعمال لغة VHDL و بطاقة التطوير Memec Design . في الأخير تم إنجاز الدارة الكاملة للمحول مستمر- متناوب، ثم تجريبها و التعليق على النتائج المحصل عليها.

كلمات مفتاحية: تغيير السرعة، محرك لا تزامني، تعديل عرض النبضة، FPGA ، VHDL ، دارة مبرمجة، المحول مستمر - متناوب.

Résumé

L'objectif de ce mémoire est l'étude et la réalisation d'un variateur de vitesse pour moteur asynchrone triphasé. La commande utilisée est une MLI calculée On-Line (en temps réel) à élimination d'harmoniques et asservissement du fondamental. C'est une commande scalaire avec V/f constant qui permet d'obtenir un couple maximal et constant, et de donner un bon spectre à la sortie de l'onduleur en éliminant les harmoniques indésirables. Dans ce mémoire nous avons commencé par simuler le système à réaliser dans l'environnement MATLAB et PSIM, puis nous avons implémenté l'algorithme MLI on-line sur un circuit FPGA XC2V1000 de XILINX, en utilisant le langage VHDL et la carte de développement Memec Design V2MB1000. Ensuite nous avons réalisé l'étage de puissance (l'onduleur triphasé), et les étages d'isolation et d'adaptation, Enfin nous avons présenté des tests sur ce variateur et des discussions des résultats obtenus.

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

Abstract

The aim of this work is to design and implement an induction motor based variable speed drive. The technique used is an On-Line harmonic elimination PWM (HEPWM), with fundamental control. It is a scalar technique which allows reducing torque pulsations and generating high-quality output spectra.

We have used a co-simulation based on MATLAB and PSIM to improve the performances of the HEPWM technique used. Then the V2MB1000 Memec Design development card was used to implement the On-Line algorithm in a XILINX FPGA (XC2V1000), using the VHDL language.

Finally, we have built a three phase voltage source inverter based on MOSFET transistors, with two stages one for optical isolation and the other for the bootstrap technique. The test of global system and the discussion of the results obtained are given.

Key words: real time, speed drive, induction motor, HEPWM, harmonics, FPGA, VHDL, Voltage source inverter.

Remerciements

Je tiens à remercier dieu de m'avoir donné la patience de terminer ce travail.

*Mes remerciements les plus particuliers vont à mon promoteur monsieur **LARBES Cherif** qu'il trouve ici l'expression de ma vive gratitude pour ses précieux conseils, son suivi, sa disponibilité et son aide.*

Monsieur Mourad HADDADI professeur à l'ENSP me fait l'honneur de présider le jury de ma soutenance. Qu'il en soit vivement remercié.

Je tiens à remercier les membres de jury, M. H BOUSBIA, M. M.S AÏT CHEIKH et M. A MALEK d'avoir accepté de juger ce travail.

Je réserve une pensée toute particulière à mes chers parents qui m'ont toujours soutenu et guidé tout au long de mes études. Qu'ils trouvent, en ce mémoire, le modeste témoignage de ma profonde reconnaissance.

Enfin, j'aimerais remercier ma famille, ainsi que mes amis qui m'ont aidé et encouragé tout au long de mon travail.

Douadi

Liste des figures

Fig. I.1.	Schéma simplifié d'un moteur à cage d'écureuil.	04
Fig. I.2.	Couple moteur T_m en fonction de la fréquence	05
Fig. I.3.	Schéma de principe d'un onduleur de tension en demi-pont	07
Fig. I.4.	Différentes formes des signaux	08
Fig. I.5.	Schéma de principe d'un onduleur monophasé en pont	08
Fig. I.6.	Tensions et leurs spectres	09
Fig. I.7.	Schéma de principe d'un onduleur triphasé de tension	09
Fig. I.8.	Signaux de sortie	10
Fig. I.9.	Schéma de principe d'un onduleur pour alimentation de secours	10
Fig. I.10.	Schéma de principe d'un onduleur pour la conduite d'une MAS	11
Fig. I.11.	Principe de la MLI Sinus- Triangle pour onduleur monophasé	13
Fig. I.12.	Forme des signaux de commande, la tension de sortie et son spectre PWM à deux niveaux pour un onduleur monophasé en pont.	13
Fig. I.13.	Principe de la MLI Sinus- Triangle pour onduleur triphasé	14
Fig. I.14.	Forme des signaux de commande, la tension de sortie et son spectre PWM pour onduleur triphasé	14
Fig. II.1.	Onduleur monophasé en demi-pont	17
Fig. II.2.	Forme d'onde MLI pour un onduleur monophasé en demi-pont	17
Fig. II.3.	Différentes techniques MLI calculée à élimination d'harmoniques	19
Fig. II.4.	MLI calculée pour onduleur monophasé	20
Fig. II.5.	MLI calculée pour onduleur triphasé	20
Fig. II.6.	Valeurs exactes des angles de commutation en fonction de l'indice de modulation m	22
Fig. II.7.	Courbes représentant $d(m)$ en fonction de m , avec m égal à 9 et k variant de 1 à m	25
Fig. II.8.	Courbes représentant la fonction $d(x)$ pour: $a_2=1$ $p=4$ $a_1=0.5, 1, 1.5$	25
Fig. II.9.	Partie puissance, l'onduleur triphasé et la charge	28
Fig. II.10.	Partie de la commande MLI	28
Fig. II.11.	Les 3 signaux de commande G_1, G_2, G_3	29
Fig. II.12.	Forme du signal de commande G_1 sur une période	29
Fig. II.13.	Tension simple V_{an} pour $m=1$ et $m=5$	30
Fig. II.14.	Spectre de la tension simple pour $m=1$ et $m=5$	30
Fig. II.15.	Tension composée de l'onduleur triphasé pour $m=1$ et $m=5$	31
Fig. II.16.	Spectre fréquentiel de la tension composée pour $m=1$ et $m=5$	31
Fig. II.17.	(a) : Forme des courants, (b) : Spectre de la tension composée	32
Fig. II.18.	Tension simple V_{an} et son spectre fréquentiel pour $m=0.6$ et $m=7$	32
Fig. II.19.	Spectre fréquentiel de la tension composée (a), et du courant (b), pour $m=0.6$ et $m=7$	33
Fig. II.20.	Tension simple V_{an} (a) et son spectre (b) pour $m=0.1$ et $m=23$.	33
Fig. II.21.	Spectre de fréquentiel pour $m=0.1$ et $m=23$. (a) : Tension composée V_{ab} . (b) : Courant I_1	34
Fig. III.1.	Domaines d'utilisation des DSPs et des FPGAs	35
Fig. III.2.	Architecture des circuits FPGAs	36
Fig. III.3.	Structure d'une cellule logique élémentaire	37
Fig. III.4.	Architecture d'un CLB des circuits FPGA de la famille Virtex	39
Fig. III.5.	Cellule logique (CLB)	43
Fig. III.6.	Input Output Block (IOB)	44
Fig. III.7.	Etape de réalisation d'un projet sur circuit FPGA	48
Fig. III.8.	Synoptique de l'application	51
Fig. III.9.	Différents programmes	52
Fig. III.10.	Résultats de simulation pour différentes valeurs de m et m	53
Fig. III.11.	Résultats de simulation fonctionnelle pour $m=1$ et $m=5$	53
Fig. III.12.	Résultats de simulation fonctionnelle pour $m=0.4$ et $m=15$	53

Fig. III.13.	Résultat de l'outil view RTL schematic	54
Fig. III.14.	Résultat de l'outil view technology schematic	54
Fig. III.15.	Diviseur de fréquence	55
Fig. II.16.	Multiplexeur	55
Fig.III.17.	Affichage	55
Fig III.18.	Choix des contraintes	56
Fig III.19.	Ressources utilisées	56
Fig.III.20.	La carte de développement utilisée (V2MB1000)	57
Fig.III.21.	Les trois signaux MLI pour $im=1$ et $m=5$	57
Fig.III.22.	Signaux MLI pour $im=0.8$ et $m=7$	58
Fig.III.23.	Signal généré pour : (a) $m=23$ et $im=0.1$; (b) : $m=19$ et $im=0.2$	58
Fig. IV.1.	Schéma synoptique du variateur de vitesse réalisé	60
Fig. IV.2.	Domaines de fonctionnement du MOSFET	61
Fig. IV.3.	Méthodes de limitation des pics de tension	62
Fig. IV.4.	Commande de MOSFET par portes logiques	62
Fig. IV.5.	Principe de commande d'un bras de pont	63
Fig. IV.6.	Utilisation du transformateur d'impulsion pour la commande des MOSFET	64
Fig. IV.7.	Utilisation d'un optocoupleur pour la commande d'un transistor IGBT	65
Fig. IV.8.	Commande d'un MOFET par alimentation isolée	65
Fig. IV.9.	Principe de la technique de bootstrap	66
Fig. IV.10.	Signaux de commande délivrés par le circuit FPGA	67
Fig. IV.11.	L'optocoupleur HCPL2200 et son schéma interne	68
Fig. IV.12.	Circuit d'isolation	68
Fig. IV.13.	Sortie du FPGA (CH2) et la sortie de l'optocoupleur (CH1)	69
Fig. IV.14.	Connexion typique du circuit IR2111	69
Fig. IV.15.	Schéma interne du driver IR2111	70
Fig. IV.16.	Composants de la technique de bootstrap	70
Fig. IV.17.	Circuit des drivers IR2111	72
Fig. IV.18.	CH1: Signal FPGA; CH2: Sortie low du driver IR2111	72
Fig. IV.19.	CH1: Signal FPGA, CH2: Sortie Lo du driver IR2111	73
Fig. IV.20.	Temps de commutation en fonction de R_G	74
Fig. IV.21.	Schéma de l'étage de puissance	74
Fig. IV.22.	Tension de sortie de l'onduleur	75
Fig. IV.23.	Forme de la tension simple V_{an} pour $im=1$, $m=5$	76
Fig. IV.24.	Spectre de la tension simple pour $im=1$, $m=5$	76
Fig. IV.25.	Spectre de la tension composée pour $im=1$, $m=5$	76
Fig. IV.26.	Forme de la tension V_{an} pour $im=0.9$ et $m=5$	77
Fig. IV.27.	Spectre de la tension simple V_{an} pour $im=0.9$ et $m=5$	77
Fig. IV.28.	Spectre de la tension composée pour $im=0.9$ et $m=5$	77
Fig. IV.29.	Spectre de la tension simple V_{an} pour $im=0.8$ $m=7$	78
Fig. IV.30.	Spectre de la tension composée V_{ab} pour $im=0.8$ $m=7$	78
Fig. IV.31.	Forme de la tension simple V_{an} pour $im=0.2$ $m=19$	79
Fig. IV.32.	Spectre de la tension simple V_{an} pour $im=0.2$ $m=19$	79
Fig. IV.33.	Spectre de la tension composée V_{ab} pour $im=0.2$ $m=19$	79
Fig. IV.34.	Spectre de la tension simple V_{an} pour $im=0.1$ $m=23$	80
Fig. IV.35.	Spectre de la tension composée V_{ab} pour $im=0.1$ $m=23$	80

SOMMAIRE

Introduction générale	01
CHAPITRE I: MOTEUR ASYNCHRONE ET ONDULEUR DE TENSION : ETAT DE L'ART	
Introduction	03
I.1 Constitution du moteur asynchrone	03
I.2 Commande du moteur asynchrone	04
I.2.1 Commande scalaire	04
I.2.2 Commande vectorielle	06
I.3 Inversion du sens de rotation	07
I.4 Onduleur de tension	07
I.4.1 Définition et principe	07
I.4.2 Les Onduleurs monophasés de tension	07
I.4.3 Les onduleurs triphasés	09
I.4.4 Domaines d'application	10
I.4.5 La commande des onduleurs	11
Conclusion	15
CHAPITRE II: ALGORITHME MLI CALCULEE ON-LINE	
Introduction	16
II.1 Avantages de la MLI calculée.	16
II.2 Principe de la MLI Calculée de PATEL et HOFT	16
II.3 Différentes structures de la MLI calculée	19
II.4 Les algorithmes MLI on-line	21
II.5 Algorithme MLI on-Line proposé	22
II.5.1 Approximation des angles exacts	22
II.5.2 Cas k impair	23
II.5.3 Cas k pair	26
II.6 Simulation du nouvel algorithme MLI on-Line	27
II.6.1 Schéma de simulation sous SIMULINK et PSIM	28
II.6.2 Résultats de simulation	28
Conclusion	34
CHAPITRE III: IMPLEMENTATION DE L'ALGORITHME MLI ON-LINE SUR FPGA	
Introduction	35
III.1 Les circuits FPGA	36
III.1.1 Structure générale des circuits FPGAs	36
III.1.2 Méthodes de programmation	37
III.1.3 Ressources typiques des FPGA	38
III.1.4 Applications des FPGAs	38
III.1.5 Architecture des circuits FPGA Virtex II de Xilinx	39
III.1.6 Performance des FPGAs Virtex	44
III.2 Le VHDL	45
III.2.1 Historique	45
III.2.2 Utilité du VHDL	45
III.3 Etapes de réalisation d'un projet sur un circuit FPGA	47
III.3.1 Description	48

III.3.2 Vérification des erreurs	48
III.3.3 Synthèse	49
III.3.4 Implémentation	49
III.3.5 Configuration	51
III.4 Implémentation de l'algorithme MLI on-line proposé	51
III.4.1 Schéma block	51
III.4.2 Développement du code VHDL	52
III.4.3 Résultats de simulation	52
III.4.4 La synthèse	53
III.4.5 Placement et Routage	56
III.4.6 Génération du fichier .bit chargement et vérification	56
III.4.7 Test des signaux de sortie	57
Conclusion	58

CHAPITRE VI: REALISATION D'UN VARIATEUR DE VITESSE POUR MOTEUR ASYNCHRONE TRIPHASE

Introduction	59
IV.1 Description générale de l'application	56
IV.2 Commande des transistors MOSFET	60
IV.2.1 Généralités sur les transistors MOSFET	60
IV.2.2 Commande des MOSFET low side	61
IV.2.3 Transistor high side	62
IV.3 Conception et réalisation de la carte de commande	67
IV.3.1 Etage d'isolation	67
IV.3.2 Le driver IR2111	69
IV.3.3 Sélection des composants bootstrap	70
IV.4 Conception et réalisation de l'étage de puissance	73
IV.4.1 Le MOSFET IRFP450	73
IV.4.2 Choix de la résistance Rg	74
IV.5 Résultats expérimentaux	75
IV.6 Interprétation et discussion des résultats	81
Conclusion	85
Conclusion générale	86

Bibliographie

Annexe : Schémas et circuits imprimés de l'onduleur réalisé

INTRODUCTION GENERALE

Le moteur asynchrone est robuste, de construction facile et peut être utilisé dans des atmosphères difficiles. Cependant, la variation de sa vitesse nécessite dans le cas de la commande statorique, une source d'alimentation alternative variable en amplitude et en fréquence.

L'un des principaux problèmes liés aux convertisseurs statiques, et qui dépend de la stratégie de commande utilisée, est celui de la présence inévitable des harmoniques dans l'onde de sortie. Les harmoniques ont des effets néfastes sur le moteur, ils provoquent l'échauffement excessif, les pulsations du couple et la saturation du circuit magnétique [5].

La commande MLI (Modulation de Largeur d'Impulsion) à élimination d'harmoniques permet d'éliminer les harmoniques indésirables et de donner un bon spectre à la sortie, et par conséquent contribuer au bon fonctionnement de l'association convertisseur-machine [9].

Le problème de cette stratégie est qu'on doit résoudre des systèmes d'équations non linéaires par des méthodes numériques qui demandent un temps important de calcul. Par conséquence, ces commandes ne peuvent être utilisées en temps réel [16].

Afin de résoudre ce problème, on propose dans ce mémoire d'élaborer des commandes MLI on-line basées sur des algorithmes approximatifs, rapides et avec moins de calcul pour pouvoir les implémenter sur des architectures hardwares, des circuits FPGA dans notre cas.

L'autre objectif de ce mémoire est de réaliser un variateur de vitesse utilisant l'algorithme MLI on-line proposé pour la commande en temps réel d'un moteur asynchrone triphasé. Pour atteindre ces objectifs nous avons suivi les étapes suivantes:

En premier lieu, nous avons donné quelques principes sur le moteur asynchrone, son principe de fonctionnement et les commandes utilisées. Les onduleurs de tension à leur tour sont étudiés, en donnant leur principe, et en citant quelques stratégies de commandes utilisées.

La deuxième partie a été consacrée à l'étude et l'élaboration d'un algorithme MLI calculé On-Line, à élimination d'harmoniques et asservissement du fondamental, avec un rapport V/f (tension/fréquence) constant, afin de garder un couple électromagnétique maximale et constant.

Pour vérifier l'algorithme une simulation a été faite dans l'environnement MATLAB, SIMULINK et en utilisant le logiciel PSIM.

La troisième partie, concerne l'implémentation de l'algorithme élaboré sur une architecture reconfigurable de type FPGA. Nous avons implémenté cet algorithme sur un circuit Virtex II de XILINX, en utilisant le langage VHDL, et la carte de développement Memec Design V2MB1000.

La dernière étape est la réalisation de l'application globale, variateur de vitesse, pour cela nous avons réalisé l'étage de puissance, un circuit d'isolation, un circuit de conditionnement et d'adaptation des signaux de commande. Après la réalisation nous avons consacré une partie aux essais, aux relevés des résultats et à leurs interprétations.

CHAPITRE I

MOTEURS ASYNCHRONES ET ONDULEURS DE TENSION: ETAT DE L'ART

Les machines électriques jouent un rôle fondamental dans les différentes branches de l'industrie. Elles sont utilisées dans les entraînements réglés et principalement comme actionneur dans la grande majorité des servomécanismes. Il est connu, pour ce qui est de la partie technologique, que la productivité et la qualité du produit fini sont en partie dépendantes de la conduite des machines électriques intervenants dans le processus.

Le moteur asynchrone présente de nombreux avantages par rapport au moteur à courant continu. L'absence du système balais collecteur permet d'accroître sa fiabilité, et de limiter son coût de fabrication et d'entretien. Par ailleurs, les progrès réalisés en matière de commande et les développements technologiques, tant dans le domaine de l'électronique de puissance que celui de la micro électronique, ont rendu possible l'usage de commandes performantes faisant du moteur asynchrone un concurrent potentiel dans les domaines de la vitesse variable [1].

I.1 Constitution du moteur asynchrone

Le moteur asynchrone, appelé aussi moteur à induction, était conçu en 1888 par le yougoslave Tesla et l'italien Ferraris.

Il se présente sous la forme d'un carter entourant le circuit magnétique qui accueille dans des encoches l'enroulement statorique généralement triphasé bobiné en fil de cuivre isolé, comme il est présenté dans la figure I.1. Les parties principales du moteur asynchrone sont:

- Le stator est un bobinage triphasé, qui engendre un champ magnétique tournant. Pour les petites puissances (usuellement < 10 kW), le rotor est constitué de barres (cuivre ou aluminium) formant un tambour appelé "cage d'écureuil".
- Le rotor, non connecté, est en court-circuit : ce moteur est dépourvu de collecteur et de balais. Pour des puissances plus importantes, le rotor est bobiné (triphase, Y), relié à l'extérieur via un collecteur simplifié à trois bagues, et court-circuité en fonctionnement normal. Mais on peut aussi modifier les propriétés électromécaniques du moteur en agissant sur le rotor par ces connexions.
- Un entrefer entre stator et rotor, de l'ordre de 0.4 mm pour des puissances < 10 kW.

Le bobinage statorique peut se décomposer en deux parties : les conducteurs d'encoches et les têtes de bobines. Les conducteurs d'encoches permettent de créer dans l'entrefer le champ magnétique à l'origine de la conversion électromagnétique. Les têtes de bobines permettent, quant à elles, la fermeture des courants en organisant la circulation judicieuse des courants d'un conducteur d'encoche à l'autre. L'objectif est d'obtenir à la surface de l'entrefer une distribution de courant la plus sinusoïdale possible, afin de limiter les ondulations du couple électromagnétique.

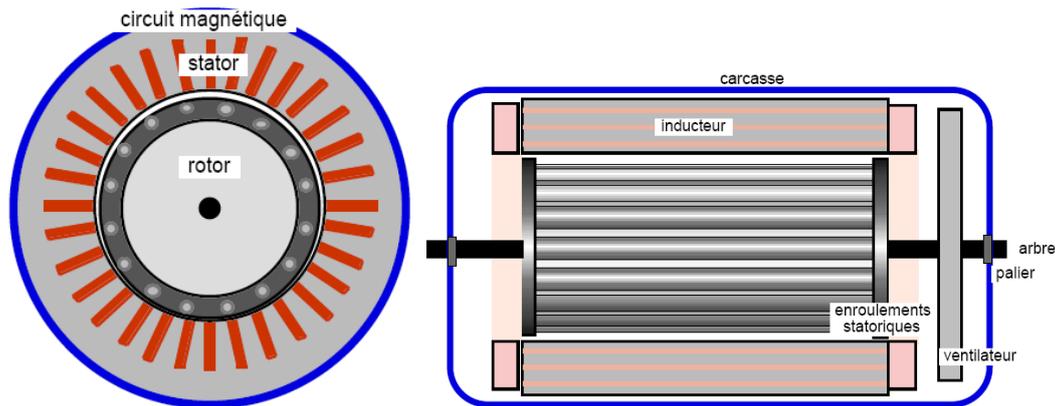


Fig. I.1. Schéma simplifié d'un moteur à cage d'écureuil

I.2 Commande de la machine asynchrone

On distingue deux types de commandes : les commandes scalaires et les commandes vectorielles.

- 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 [2].

I.2.1 Commande scalaire

Plusieurs commandes scalaires existent selon que l'on agit sur le courant ou sur la tension. Elles dépendent surtout de la topologie de l'actionneur utilisé (onduleur de tension ou de courant). L'onduleur de tension étant maintenant le plus utilisé en petite et moyenne puissance, c'est la commande en V/f qui est la plus utilisée.

I.2.1.1 Contrôle en V/f de la machine asynchrone

Son principe est de maintenir $V/f = \text{Constant}$ ce qui signifie garder le flux constant. Le contrôle du couple se fait par l'action sur le glissement.

En effet, d'après le modèle établi en régime permanent, le couple maximum s'écrit :

$$T_{\max} = \frac{3pV^2}{4\pi f (R_s + \sqrt{R_s^2 + [2\pi f(L_s + L_{re})]^2})^2} \quad (\text{I-1})$$

Avec R_s et L_s la résistance et l'inductance du stator, L_{re} l'inductance du rotor ramenée au stator, p le nombre de paires de pôles, V la tension efficace d'entrée du moteur (d'une phase), f la fréquence de la tension d'alimentation.

On voit bien que le couple est directement proportionnel au carré du rapport de la tension sur la fréquence statorique.

En maintenant ce rapport constant et en jouant sur la fréquence statorique, on déplace la courbe du couple électromagnétique (en régime quasi-statique) de la machine asynchrone.

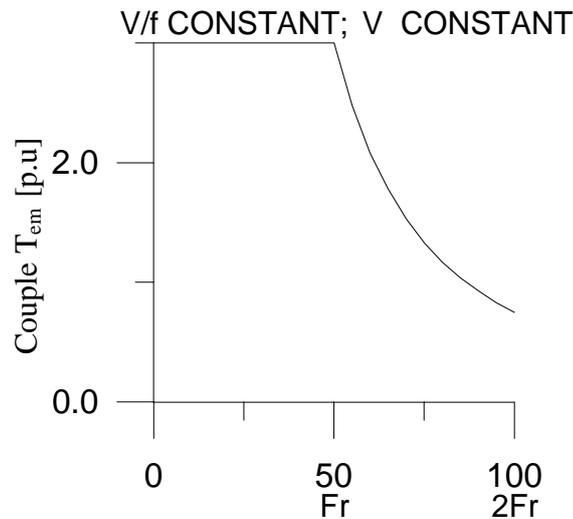


Fig. I.2. Couple moteur T_m en fonction de la fréquence

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 figure I.2. 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. A basse vitesse, la chute de tension ohmique ne peut pas être négligée. On compense alors en ajoutant un terme de tension V_0 .

I.2.1.2 Contrôle scalaire du courant

La différence avec la commande précédente, c'est que c'est un onduleur (commutateur) de courant qui est utilisé. On impose directement des courants dans les phases de la machine. La fréquence du fondamental est calculée de la même manière. La valeur du courant de plateau I_d (courant continu) est égale à une constante près de la valeur efficace du courant imposé I_s . Elle est imposée par régulation à l'aide d'un pont redresseur contrôlé.

I.2.2 Commande vectorielle

La commande vectorielle a été introduite dans les années 70 [44]. Cependant, elle n'a pu être implantée et utilisée réellement qu'avec les avancés en micro-électronique. En effet, elle nécessite des calculs de transformé de Park, évaluation de fonctions trigonométriques, des intégrations, des régulations... ce qui ne pouvait pas se faire en pure analogique.

Le contrôle de la machine asynchrone requiert le contrôle du couple, de la vitesse ou même de la position. Le contrôle le plus primaire est celui des courants et donc du couple, puisque le couple s'écrit directement en fonction des courants :

$$C_e = pM(i_{qs}i_{dr} - i_{ds}i_{qr}) \quad (I.2)$$

Une fois que l'on maîtrise la régulation du couple, on peut ajouter une boucle de régulation externe pour contrôler la vitesse. On parle alors de régulation en cascade ; les boucles sont imbriquées l'une dans l'autre. Il est évident que pour augmenter la vitesse, il faut imposer un couple positif, pour la diminuer il faut un couple négatif. Il apparaît alors clairement que la sortie du régulateur de vitesse doit être la consigne de couple. Ce couple de référence doit à son tour être imposé par l'application des courants ; c'est le rôle des régulateurs de courants.

Cependant, la formule du couple électromagnétique est complexe, elle ne ressemble pas à celle d'une machine à courant continu où le découplage naturelle entre le réglage du flux et celui du couple rend sa commande aisée. On se retrouve confronté à une difficulté supplémentaire pour contrôler ce couple.

La commande vectorielle vient régler ce problème de découplage des réglages du flux à l'intérieur de la machine de celle du couple.

Le couple en régime transitoire (quelconque) s'exprime dans le repère dq comme un produit croisé de courants ou de flux. Si nous reprenons l'écriture :

$$C_e = p \frac{M}{L_r} (\varphi_{dr}i_{qs} - \varphi_{qr}i_{ds}) \quad (I.3)$$

On s'aperçoit que si l'on élimine le deuxième produit $\varphi_{qr}i_{ds}$, alors le couple ressemblerait fort à celui d'une MCC. Il suffit, pour ce faire, d'orienter le repère dq de manière à annuler la composante de flux en quadrature. C'est-à-dire, de choisir convenablement l'angle de rotation de Park de sorte que le flux rotorique soit entièrement porté sur l'axe direct (d) et donc d'avoir $\varphi_{qr} = 0$

I.3 Inversion du sens de rotation

Certains moteurs doivent fonctionner dans les deux sens comme, par exemple, le moteur d'ascenseur. On peut inverser le sens de rotation en changeant la séquence des phases des trois lignes alimentant le moteur [3].

I.4 Onduleur de tension

I.4.1 Définition et principe

Les onduleurs de tension sont des convertisseurs statiques qui servent principalement à alimenter, à fréquence fixe ou variable, des charges alternatives. Le but recherché est l'obtention pour chaque tension de sortie d'une forme d'onde approximant au mieux la sinusoïde.

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

On distingue deux types d'onduleurs :

Les onduleurs autonomes de tension : sont alimentés par une source de tension continue, d'une impédance interne négligeable et de tension constante peu affectée par les variations du courant qui la traverse. Les onduleurs autonomes de courant sont alimentés par une source de courant [5].

I.4.2 Les Onduleurs monophasés de tension

Ce type d'onduleurs est destiné à alimenter des charges alternatives monophasées, on distingue deux configurations de base : [5] [4]

I.4.2.1 Onduleur monophasé en demi-pont

Les figures (I.3) (I.4) montrent respectivement la structure d'un onduleur monophasé en demi-pont et les formes des différentes grandeurs.

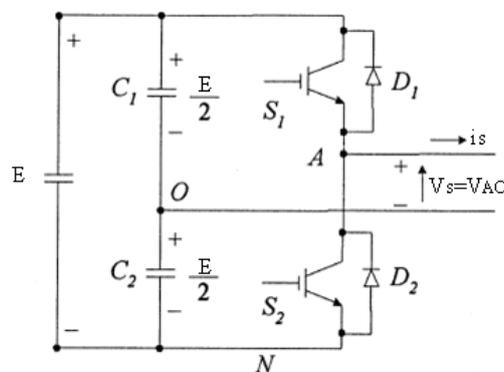


Fig. I.3. Schéma de principe d'un onduleur de tension en demi-pont

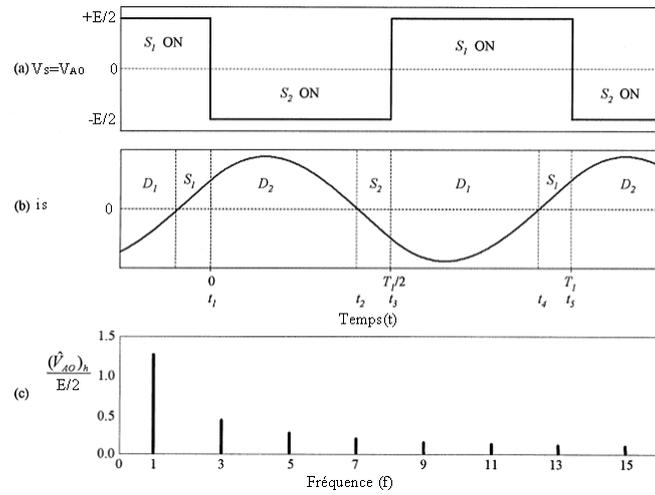


Fig. I.4. Différentes formes des signaux

- (a) Tension de sortie.
- (b) Courant dans la charge.
- (c) Spectre normalisé de la tension de sortie.

I.4.2.2 Onduleur monophasé en pont

Les figures (I.3) (I.4) montrent respectivement la structure d'un onduleur monophasé en pont et les formes des différentes grandeurs.

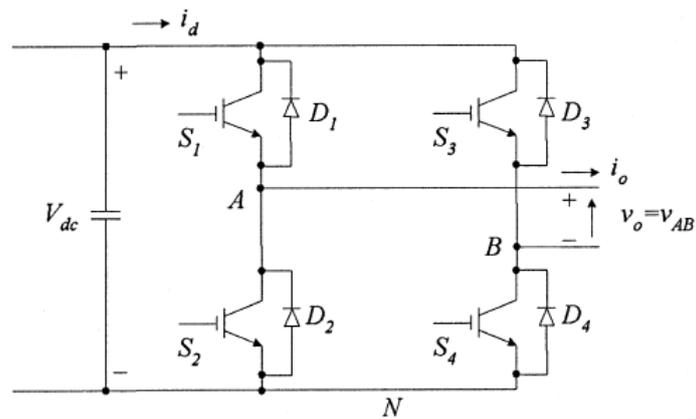


Fig. I.5. Schéma de principe d'un onduleur monophasé en pont

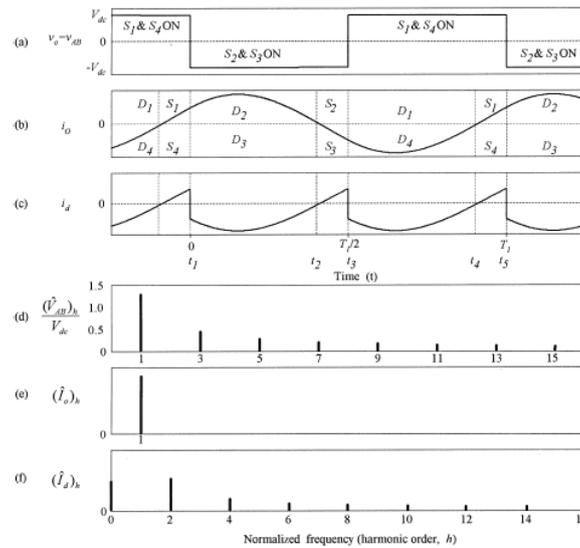


Fig. I.6. Tensions et leurs spectres

I.4.3 Les onduleurs triphasés

Les onduleurs monophasés sont utilisés pour des applications de faible puissance, alors que les onduleurs triphasés couvrent la gamme de moyenne et de forte puissance. L'objectif de cette topologie est de fournir une source de tension triphasée, dont l'amplitude, la phase et la fréquence sont contrôlables [21].

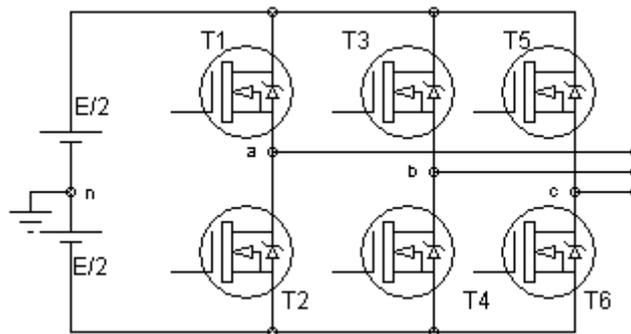


Fig. I.7. Schéma de principe d'un onduleur triphasé de tension

La figure I.7 montre un onduleur de tension triphasé constitué de trois cellules de base. Une charge triphasée est raccordée aux points milieu a, b et c. La tension d'entrée est généralement fournie par un redresseur.

Si on commande les interrupteurs des trois cellules avec un déphasage d'un tiers de période fondamentale, les trois tensions simples V_{an} , V_{bn} et V_{cn} (vis-à-vis du point milieu fictif n à l'entrée) sont également déphasés d'un tiers de période fondamentale les unes par rapport aux autres. Il en va de même des trois tensions composées, par exemple $V_{ab} = V_{an} - V_{bn}$. Ceci est illustré par la figure I.8 pour une onde carrée.

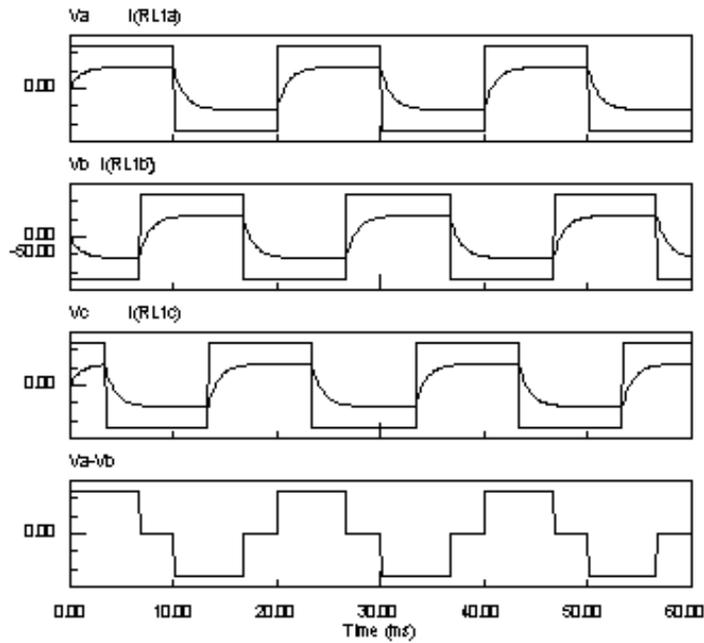


Fig. I.8. Signaux de sortie pour un onduleur triphasé

I.4.4 Domaines d'application

Les deux grands domaines d'applications classiques des onduleurs de tension sont les alimentations de secours et la commande des moteurs alternatifs. Ils sont également caractéristiques de deux grandes familles, respectivement celle des systèmes à fréquence fixe et celle des systèmes à fréquence variable [5] [6].

I.4.4.1 Domaine des fréquences fixes

Dans ce premier domaine, l'onduleur de tension est destiné à palier les défaillances ou même l'absence d'un réseau alternatif en recréant la tension correspondante à partir d'une batterie. Le schéma de principe d'une telle application est donné par la figure I.9

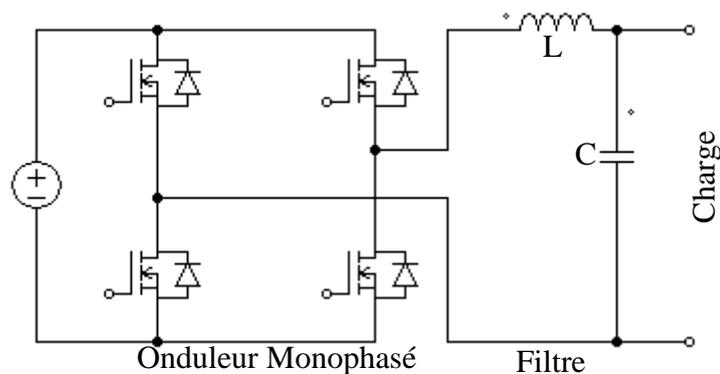


Fig. I.9. Schéma de principe d'un onduleur pour alimentation de secours

Cette application requiert généralement la fourniture d'une tension de sortie très pure, donc sans les harmoniques de découpage, d'où la nécessité d'un filtre de sortie.

I.4.4.2 Domaine des fréquences variables

Dans ce deuxième domaine, l'onduleur est généralement alimenté par l'intermédiaire d'un réseau alternatif. La source continue est donc un redresseur suivi d'un filtre d'entrée qui peut avoir une double fonction, à savoir éliminer les composantes harmoniques de courant issues de l'onduleur, et les composantes harmoniques de tension dues au redressement.

Une autre différence notable par rapport au cas précédent réside dans le domaine de fonctionnement beaucoup plus étendu tant en fréquence fondamentale (de quelques hertz à quelques centaines de hertz) qu'en amplitude.

Le domaine d'applications des onduleurs de tension le plus connu est sans doute celui de la variation de vitesse des machines à courant alternatif. Le schéma de principe d'une telle application est donné par la figure I.10

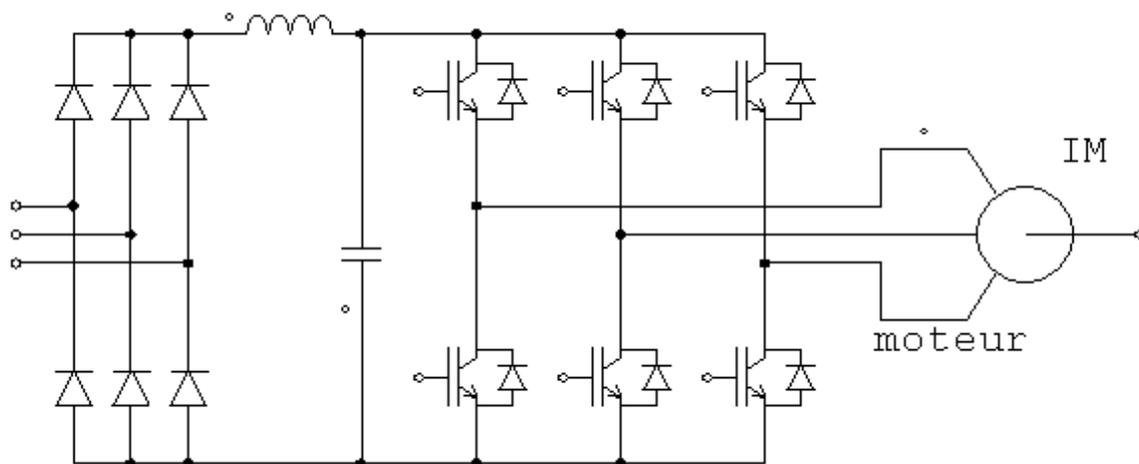


Fig. I.10. Schéma de principe d'un onduleur pour la conduite d'un moteur asynchrone

I.4.5 La commande des onduleurs

Plusieurs stratégies de commande des onduleurs ont été développées dans la littérature dont les principes consistent soit:

- Analogique : Consiste en la génération des signaux de commande des interrupteurs de puissance par l'asservissement de la tension de sortie de l'onduleur à une référence de tension sinusoïdale: c'est la commande dite implicite, technique analogique telle que principalement la MLI engendrée, la Modulation Delta, la commande à hystérésis.

- Numérique : Consiste en la détermination des instants de commutation des composants de puissance formant l'onduleur par le biais du développement en série de Fourier des formes d'onde souhaitées en sortie répondant à des critères bien définies (taux d'harmoniques, valeur du terme fondamental,..) : c'est la commande dite explicite où la commande des interrupteurs se fait d'une façon numérique telle que la technique de la sortie sinusoïdale synthétisée ou la modulation programmée [7].

I.4.5.1 Technique MLI engendrée

La commande MLI Triangulo-sinusoïdale consiste dans la comparaison d'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 [22].

L'étude de la MLI se base sur deux paramètres qui caractérisent la commande:

L'indice de modulation m qui est l'image du rapport des fréquences de la porteuse F_p sur la référence F_r . Si m est entier la modulation est synchrone. Elle est asynchrone dans le cas contraire.

Le taux de la modulation r qui est l'image du rapport des amplitudes des tensions de la référence V_m sur celle de la porteuse V_{pm} .

$$m = \frac{F_p}{F_r} \quad (I.4)$$

$$r = \frac{V_m}{U_{pm}} \quad (I.5)$$

Pour éliminer les harmoniques de rang pair et les harmoniques de rang 3, l'indice de modulation m doit être impair et multiple de 3.

La porteuse la plus adaptée aux convertisseurs à deux niveaux est la triangulaire bipolaire [4]. La figure I.11 représente le principe de génération des impulsions de commande d'un bras d'un onduleur de tension à deux niveaux.

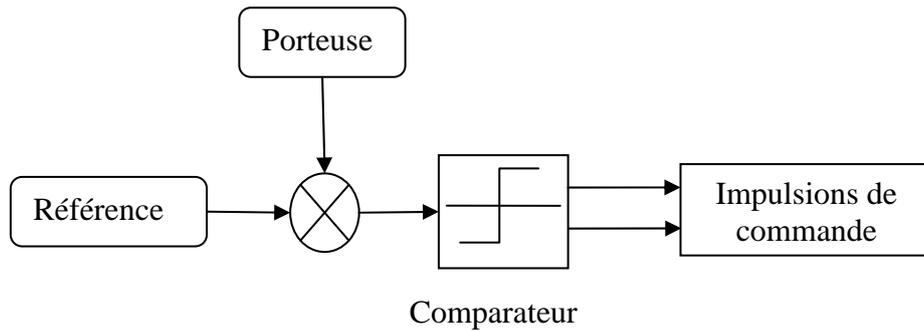


Fig. I.11. Principe de la MLI Sinus- Triangle pour onduleur monophasé

Pour un onduleur monophasé de la figure I.5 l'application de cette stratégie de commande, donne les résultats cités ci- dessous.

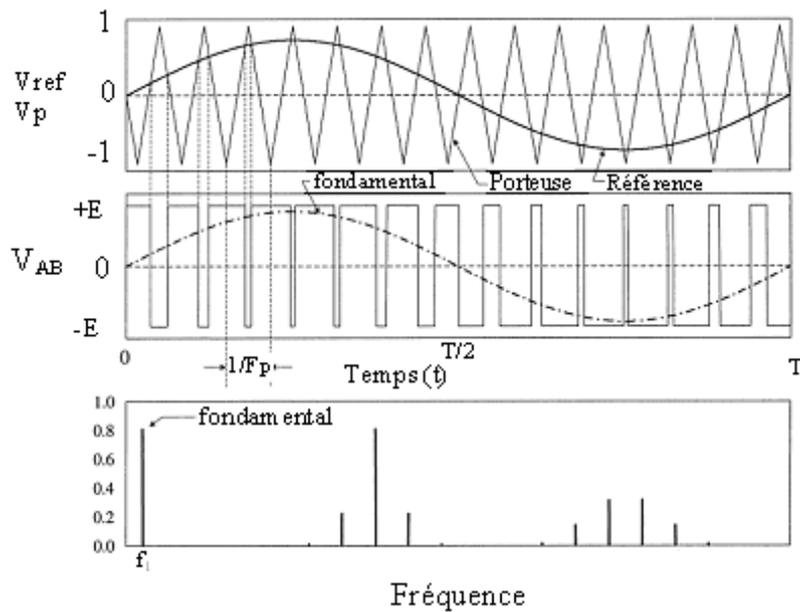


Fig. I.12. Forme des signaux de commande, la tension de sortie et son spectre

PWM à deux niveaux pour un onduleur monophasé en pont

La référence sinusoïdale, la porteuse est triangulaire bipolaire de fréquence $F_P=15F_r$ $r=0.8$.

La tension de sortie V_{AB} et son spectre fréquentiel normalisé.

Ce principe de modulation s'applique aussi pour un montage triphasé de la figure I.7 :

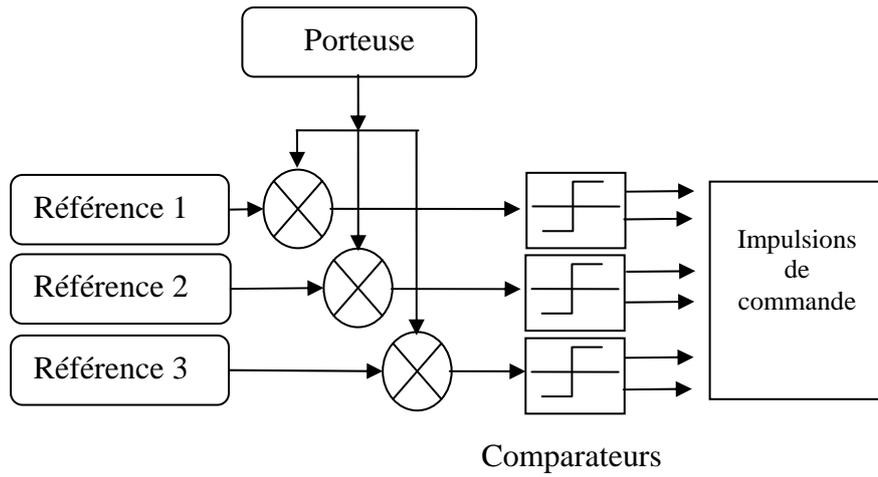


Fig. I.13. Principe de la MLI Sinus- Triangle pour onduleur triphasé

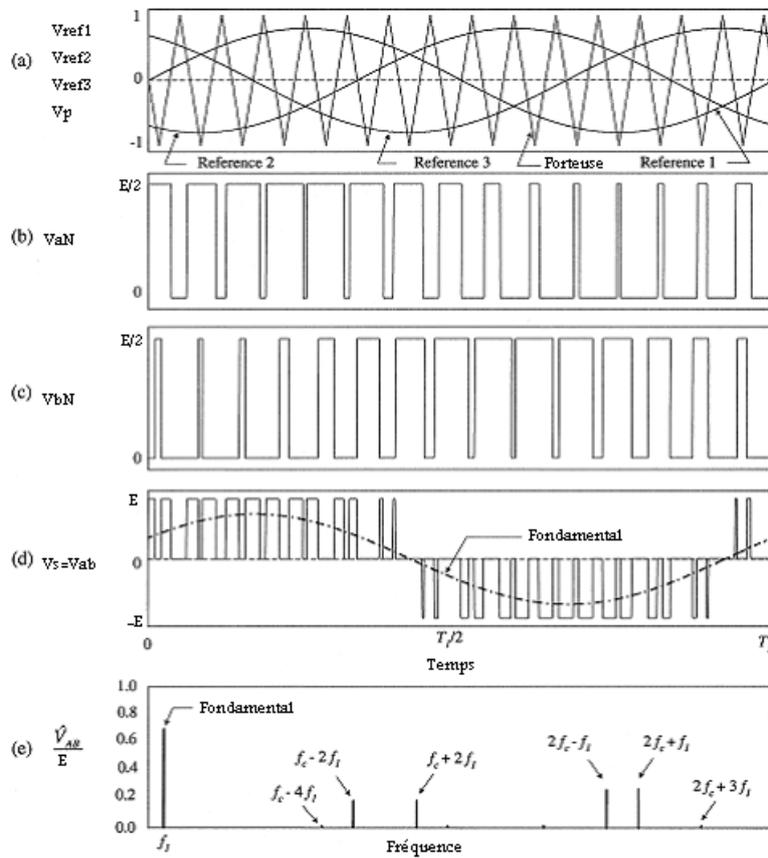


Fig. I.14. Forme des signaux de commande, la tension de sortie et son spectre

PWM pour onduleur triphasé

- a: La porteuse $F_p=15 F_r$ et $m=0.8$
- b,c : Les tension de sorties V_{an} V_{bn}
- d: La tension composée V_{ab}
- e: Spectres de fréquence de la tension V_{ab}

I.4.5.2 Technique MLI calculée (programmée)

Cette technique consiste à calculer les instants de commutation (séquences de fonctionnement) 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 d'harmoniques dans une bande de fréquences spécifiée [5] [9].

Conclusion

Dans ce chapitre nous avons présenté brièvement le moteur asynchrone ainsi que ses deux types de commande qui sont la commande scalaire et la commande vectorielle. La partie commande des onduleurs de tension présente une part décisive, elle permet de modifier voire améliorer la qualité des signaux de sortie.

La technique MLI présente l'une des techniques les plus utilisées, elle permet d'obtenir un bon spectre de sortie en augmentant l'indice de modulation mais ça nécessite un espace mémoire considérable pour une implémentation numérique.

La technique MLI programmée avec asservissement du fondamental et élimination d'harmoniques objet de ce mémoire est présentée au chapitre suivant.

CHAPITRE II

ALGORITHME MLI CALCULEE ON-LINE

La qualité et les performances des onduleurs de tension dépendent largement de la stratégie de commande utilisée. Les différentes stratégies MLI se divisent en deux grandes catégories: la MLI engendrée (triangulo-sinusoidale) et la MLI calculée ou programmée [9].

Les techniques de modulation dites triangulo-sinusoidales permettent de déterminer les instants de commutation par l'intersection (comparaison) de deux ondes, l'une de référence et l'autre porteuse. Analytiquement, ces techniques présentent des équations transcendantes et difficiles à résoudre en temps réel. Néanmoins, elles se prêtent bien à une implémentation analogique. Leur numérisation est coûteuse en temps de calcul et en espace mémoire.

La MLI programmée optimise une fonction particulière, comme par exemple la minimisation des pertes, la réduction des fluctuations du couple ou l'élimination sélective d'harmoniques [5] [9].

II.1 Avantages de la MLI calculée

Les techniques MLI programmée rencontrent une difficulté concernant les valeurs des angles de commutation, mais malgré ces difficultés de calcul, la MLI calculée présente plusieurs avantages par rapport à la MLI engendrée à modulation sinusoidale [9] [11] [12].

1- Réduction d'environ 50% dans la fréquence de commutation est atteinte en comparaison avec la MLI engendrée à référence sinusoidale, ce qui contribue à une réduction des pertes à la commutation, et permet l'utilisation des interrupteurs GTO pour les convertisseurs haute puissance.

2- Un gain élevé en tension dû à la surmodulation possible (la valeur du fondamental).

3- La MLI calculée permet d'obtenir un bon spectre dans les grandeurs de sortie : tension et courant et par conséquent les ondulations dans le courant continu -DC- d'entrée seront réduites, le filtre d'entrée à son tour sera aussi réduit.

II.2 Principe de la MLI Calculée de PATEL et HOFT

Le principe de la technique de modulation par élimination d'harmoniques a été introduit pour la première fois par Turnbull en 1964 puis développé par Patel et Hoft [9] [11].

Cette technique consiste à former l'onde de sortie d'une succession de créneaux de largeurs variables et contrôlables. Les angles de commutation sont déterminés de façon à éliminer certains harmoniques gênants dans l'onde de sortie. Améliorant ainsi le rendement du système onduleur-machine par la réduction des ondulations du couple, ainsi que des pointes de courant et des pertes dans la machine [8].

Pour un onduleur monophasé en demi-pont de la figure (II.1) l'onde de sortie par cette technique est représentée sur la figure (II.2), c'est une onde à double symétrie par rapport au quart et à la demi-période.

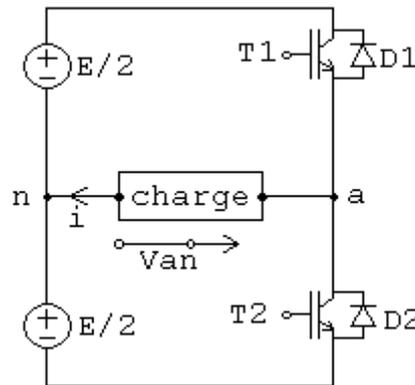


Fig. II.1. Onduleur monophasé en demi-pont

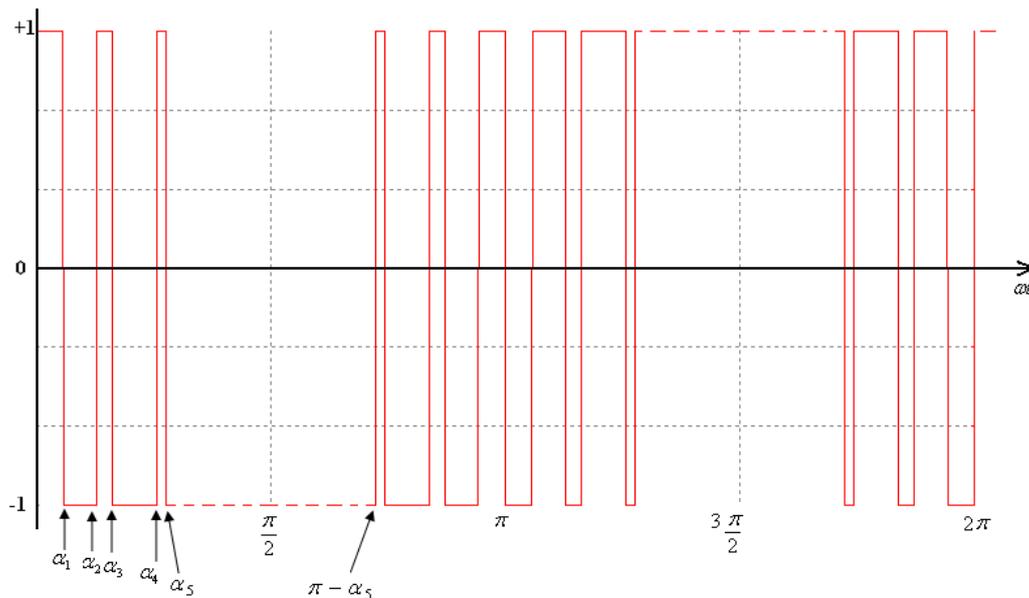


Fig. II.2. Forme d'onde MLI pour un onduleur monophasé en demi-pont

Par décomposition en série de Fourier on trouve

$$Van = f(\alpha) = a_0 + \sum_{n=1}^{\infty} (a_n \sin(n\alpha) + b_n \cos(n\alpha)) \tag{II.1}$$

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

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \sin(n\alpha) d\alpha \tag{II.2}$$

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

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).

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(n\alpha_k) \right] \tag{II.3}$$

En fixant la valeur du fondamental à im et en mettant les harmoniques qu'on désire éliminer à zéro. Pour un montage triphasé l'harmonique trois et ses multiples sont éliminés par connexion Y du moteur [19]. On obtient le système d'équations (II.4) [8].

$$\left[\begin{array}{l} a_1 = \frac{4}{n} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(\alpha_k) \right] = im \\ a_5 = \frac{4}{5\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(5\alpha_k) \right] = 0 \\ a_7 = \frac{4}{7\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(7\alpha_k) \right] = 0 \\ a_{11} = \frac{4}{11\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(11\alpha_k) \right] = 0 \\ \dots \\ \dots \\ a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(n\alpha_k) \right] = 0 \end{array} \right] \tag{II.4}$$

Ce système d'équations possède plusieurs solutions, qui doivent satisfaire la condition

$$\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \dots < \alpha_N < \frac{\pi}{2} \tag{II.5}$$

Le premier harmonique non éliminé est donné par la formule générale

$$rang = 3.m + 2 \tag{II.6}$$

m est le nombre des angles de commutation.

Il n'y a pas de théorie générale en ce qui concerne la résolution des équations non-linéaires (N-L) et encore moins pour les systèmes d'équations N-L. Pour cette raison on se penche vers les méthodes numériques basées sur des itérations qui, sous certaines conditions, convergent vers la solution exacte.

D'innombrables méthodes ont été élaborées pour la résolution des équations et des systèmes d'équations N-L, pour ces derniers la solution est plus délicate.

Les méthodes les plus rencontrées sont celles des substitutions successives, de Newton et d'optimisation – méthode de gradient [20].

L'algorithme de Newton-Raphson est l'un des plus connus et des plus puissants. Basé sur le développement de Taylor au premier ordre, il converge rapidement vers la solution exacte si on a un bon estimé au départ [5].

II.3 Différentes structures

La forme d'onde MLI en sortie de l'onduleur dépend de la charge (monophasée ou triphasée) et du type de la structure de l'onduleur (à deux niveaux ou à trois niveaux) d'autre terme en demi-pont ou en pont complet. Pour chaque variante découle un système d'équations correspondant. Les figures (II.3), (II.4) et (II.5) donnent une classification [9].

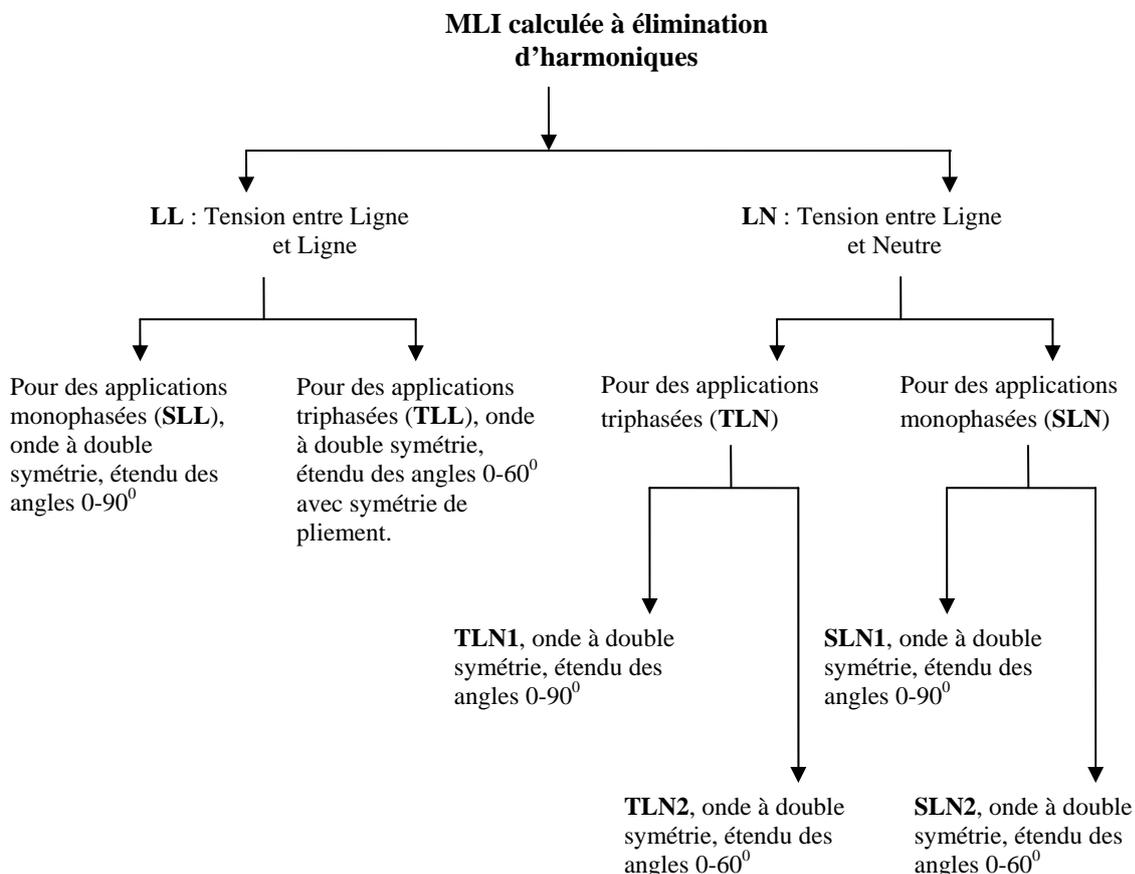


Fig. II.3. Différentes techniques MLI calculée à élimination d'harmoniques

On peut aussi classer ces techniques selon le type l'onduleur: monophasé ou triphasé, comme il est montré sur les figures (II.4) et (II.5).

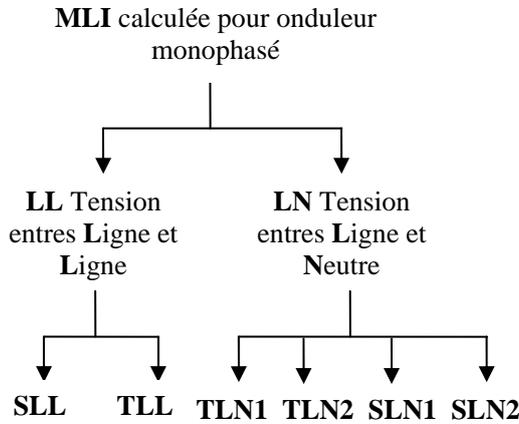


Fig. II.4. MLI calculée pour onduleur monophasé

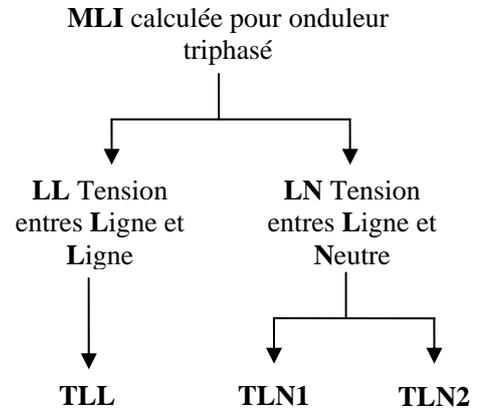


Fig. II.5. MLI calculée pour onduleur triphasé

Dans la technique TLN2 la tension est prise entre ligne et neutre, tous les harmoniques impaires non multiples de trois sont éliminés jusqu'au rang désiré. C'est une onde à double symétrie avec un étendu des angles de 0^0 à 60^0 . Son développement en série de fourrier donne

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(n\alpha_k) \right] \quad (II.7)$$

Avec m est un nombre pair, l'indice n prend les valeur 1,5,7,11, ...

Dans la technique TLL la tension est prise entre ligne et ligne, elle est appliquée dans les onduleurs à trois niveaux. Le coefficient de fourrier est donné par la formule II.18

$$a_n = \frac{4}{n\pi} \left[\sum_{k=1}^{2m} (-1)^k \cos(n\alpha_k) \right] \quad (II.8)$$

Les angles a_{m+1} à a_{2m} sont obtenus par symétrie. Par exemple pour m=6, $\alpha_7 = 120 - \alpha_6$.

La technique SLN1 est utilisée dans le cas des onduleurs monophasé, c'est une onde à double symétrie par rapport au quart et à la demi période, l'étendu des angles est de 0^0 à 90^0 . Le développement en série de Fourier est le même que l'équation (II.3).

Pour la technique SLL le coefficient a_n de Fourier est donné par

$$a_n = \frac{4}{n\pi} \left[\sum_{k=1}^m (-1)^{k+1} \cos(n\alpha_k) \right] \quad (II.9)$$

II.4 Algorithmes MLI on-line

Les équations utilisées pour calculer les angles de commutation – celles du système (II.4) – sont non linéaires et transcendantes, donc leur solution ne peut pas être faite on-line par un microprocesseur.

Ces angles sont calculés off-line par des méthodes numériques bien connues, ensuite stockés dans des mémoires; mais avec une large gamme de variation de l'indice de modulation m et l'interpolation requise demande une énorme puissance de calcul et rend le système plus coûteux [11].

Ce problème a poussé les chercheurs d'élaborer des algorithmes permettant un calcul on-line de ces angles. Un de ces algorithmes c'est celui publié par TAUFIK, MELLITT et GOODMAN [15]. Plusieurs travaux ont été faits par BOWES [13] [17] [18]. D'autres travaux existent mais ils sont dans la plupart basés sur un calcul off-line et un stockage dans des mémoires (LUT), c'est la MLI précalculée.

Dans les travaux [10] [11] [14] un algorithme MLI on-line a été proposé. Il est basé sur la méthode d'ajustement des courbes et la méthode des approximations quadratiques à partir des courbes des solutions exactes. Ils utilisent la technique TLN1 qui est destinée aux onduleurs triphasés.

Les équations élaborées sont les suivantes:

Pour les valeurs paires de k

$$\alpha_k = \frac{60^0(k+1)}{m+1} - \left[\frac{2 \times 60^0}{m+1} \times \frac{\Delta_k \times V_1}{0.8} \right] \quad (\text{II.10})$$

avec

$$\Delta_k = -\frac{0.21}{m^2} \left[k - \frac{m+1}{2} \right]^2 + 0.4025 \quad (\text{II.11})$$

Et pour les valeurs impaires de k

$$\alpha_k = \frac{60^0 \times k}{m+1} + \left[\frac{2 \times 60^0}{m+1} \times \frac{\Delta_k \times V_1}{0.8} \right] \quad (\text{II.12})$$

avec

$$\Delta_k = -\frac{0.082}{(m-1)^2} [k - 2.482(m-1)]^2 + 0.505 - \frac{k}{m^3} \quad (\text{II.13})$$

Les quatre équations précédentes sont utilisées pour des valeurs de V_1 inférieures à 0.8; pour les valeurs supérieures il faut ajouter un terme correctif ΔD_k .

La valeur corrigée sera de la forme

$$\alpha_{k(corr)} = \alpha_k - \Delta D_k \tag{II.14}$$

ΔD_k est donné par

$$\Delta D_k = \frac{(V_1 - 0.8)^2}{0.09} \times \left[-\frac{52}{m} \left[\frac{k}{m+5} - 0.5 \right]^2 + \frac{13}{m} \right] \tag{II.15}$$

k pair

$$\Delta D_k = \frac{(V_1 - 0.8)^2}{0.09} \times \left[-\frac{52}{m} \left[\frac{k}{m+3} - 0.5 \right]^2 + \frac{13}{m} \right] \tag{II.16}$$

k impair

Dans les paragraphes qui suivent on va élaborer et simuler un algorithme MLI calculée on-line, c'est un algorithme basé sur la méthode des moindres carrés.

II.5 Algorithme MLI on-line proposé

II.5.1 Approximation des angles exacts

Les valeurs des angles exacts sont calculées par un système d'équations non linéaires; les valeurs approximées sont calculées par le nouvel algorithme [16].

La figure (II.10) montre des courbes représentant les valeurs exactes des angles de commutation α_k en fonction de l'indice de modulation im (im c'est la valeur du fondamental), pour les valeurs du nombre de commutations $m=3$ et $m=5$ par quart d'onde (ou nombre de découpages par demi-onde).

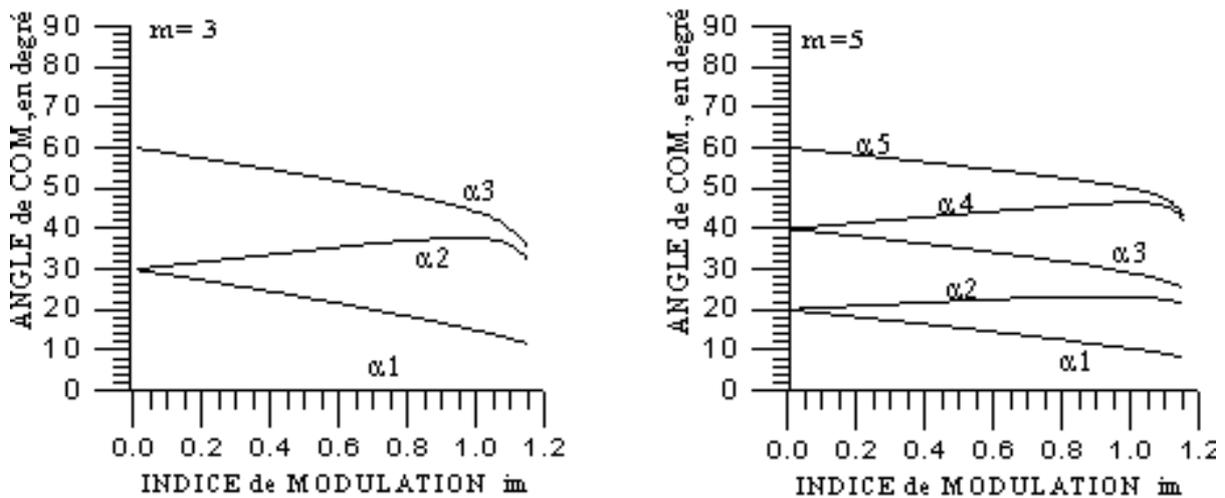


Fig. II.6. Valeurs exactes des angles de commutation en fonction de im pour $m=3$ et 5

Pour un indice de modulation im égal à zéro, l'angle α_s , appelée angle de séparation, est donné par la formule approchée:

$$\alpha_s = \frac{2 \times 60^0}{m+1} \text{ m impair} \tag{II.17}$$

Pour m fixé, les courbes de rang k impair présentent une pente négative et sont parallèles dans la majeure partie de l'intervalle de variation de l'indice de modulation im , à l'exception

des valeurs extrêmes de cet intervalle. De même les courbes de rang k pair présentent cette fois une pente positive et sont 'presque' parallèles dans les mêmes conditions de variation de l'indice de modulation im que précédemment. Cette caractéristique donne l'idée à approximer les courbes correspondant à k impair différemment de celles correspondant à k pair. De plus la forme linéaire de ces courbes laisse penser que les fonctions pour approximer les courbes exactes devraient être la combinaison d'une fonction linéaire et d'une autre fonction non linéaire de l'indice im et dont l'expression reste à déterminer.

L'idée consiste à approximer la valeur exacte de l'angle de commutation en approxinant chaque courbe exacte en deux étapes.

Dans une première étape on approxime la partie entière de la valeur exacte de l'angle de commutation par une fonction linéaire de l'indice im .

Dans une seconde étape on approxime soit la partie fractionnaire de la valeur exacte, soit la différence entre la valeur exacte et la valeur approximative, trouvée dans la première étape.

La mise en oeuvre de ces deux étapes d'approximation nécessite la conception et la mise au point d'un programme d'approximation basé sur la méthode des moindres carrés. Celui-ci calcule les coefficients A_0, A_1, \dots, A_p ($p < n$) de la fonction d'approximation suivante:

$$\Phi(x) = A_0 \cdot \phi_0(x) + \dots + A_p \cdot \phi_p(x) \quad (\text{II.18})$$

permettant d'approximer la fonction $f(x)$ connue empiriquement en $(n+1)$ points et qui prend les valeurs b_0, b_1, \dots, b_n aux abscisses a_0, a_1, \dots, a_n

Les fonctions élémentaires d'approximation $\phi_0, \phi_1, \dots, \phi_p$ sont choisies à l'avance.

II.5.2 Cas k impair

Soit m fixé. Considérons les valeurs exactes des angles de commutation α_k :

* Première étape d'approximation de α_k :

On cherche une fonction d'approximation linéaire de la forme:

$$\Phi(im) = A_0 + A_1 * im \quad (\text{II.19})$$

Pour approximer la partie entière des valeurs exactes de chaque courbe $\alpha_k(im)$ avec im variant de 0 à 1.15 et k impair. On a:

$$A_0 = \alpha_k(0) = \alpha_s = (k+1) \frac{60^\circ}{(m+1)} \quad (\text{II.20})$$

$$\text{On suppose: } A_1' = A_1 \times \frac{60^\circ}{(m+1)} \quad (\text{II.21})$$

On remplace dans l'équation (II.19) On trouve la nouvelle fonction d'approximation:

$$\Phi(im) = \frac{60^\circ}{(m+1)}(k+1) + A_1 \times \left(\frac{60^\circ}{(m+1)}im\right) \quad (\text{II.22})$$

Notons $[\alpha_k]$, la partie entière de α_k . Pour calculer A_1 , il faut mettre l'équation (II.22) sous la forme:

$$\Phi'(im) = [\alpha_k] - \frac{60^\circ}{(m+1)}(k+1) = A_1 \times \left(\frac{60^\circ}{(m+1)}im\right) \quad (\text{II.23})$$

On cherche donc à approximer la différence suivante:

$$[\alpha_k] - \frac{60^\circ}{(m+1)}(k+1) \quad (\text{II.24})$$

par la fonction d'approximation suivante:

$$\Phi'(im) = A_1 \times \left(\frac{60^\circ}{(m+1)} \times im\right) \quad (\text{II.25})$$

La fonction d'approximation élémentaire est donc:

$$\phi_0(im) = \frac{60^\circ}{(m+1)}im \quad (\text{II.26})$$

Pour déterminer la valeur du coefficient A_1 , on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.1. Pour chaque valeur de l'indice im , on calcule la différence (II.24). Le coefficient A_1 va nous permettre de construire la fonction d'approximation (II.25) qui va approximer l'ensemble des données numériques citées ci-dessus.

Le coefficient A_1 est pratiquement constant et égal à moins un (-1) lorsque k varie de 1 à m , par valeurs impaires, quelque soit la valeur de m . L'équation (II.22) devient:

$$\Phi(im) = \frac{60^\circ}{(m+1)}(k+1) + (-1) \times \frac{60^\circ}{(m+1)} \times im \quad (\text{II.27})$$

* Seconde étape d'approximation de α_k :

Après avoir approximé la partie entière de l'angle 'exact' de commutation, dans la première étape d'approximation, on va approximer maintenant la différence $d(im)$ entre la valeur exacte et la valeur approximée, donnée par l'équation (II.27), de la partie entière de l'angle de commutation. La différence $d(im)$ est définie par:

$$d(im) = \alpha_k(im) - \Phi(im) = \alpha_k(im) - \left[\frac{60^\circ}{m+1}(k+1) + (-1) \times \frac{60^\circ}{m+1} \times im \right] \quad (\text{II.28})$$

Pour étudier la forme de $d(im)$, on va calculer $d(im)$ pour m égal à 9, par exemple, et pour k variant de 1 à 9 par valeurs impaires en faisant varier im de 0.1 à 1.1 par pas de 0.1. La figure II.7 représente les variations de $d(im)$ en fonction de im pour plusieurs valeurs de k .

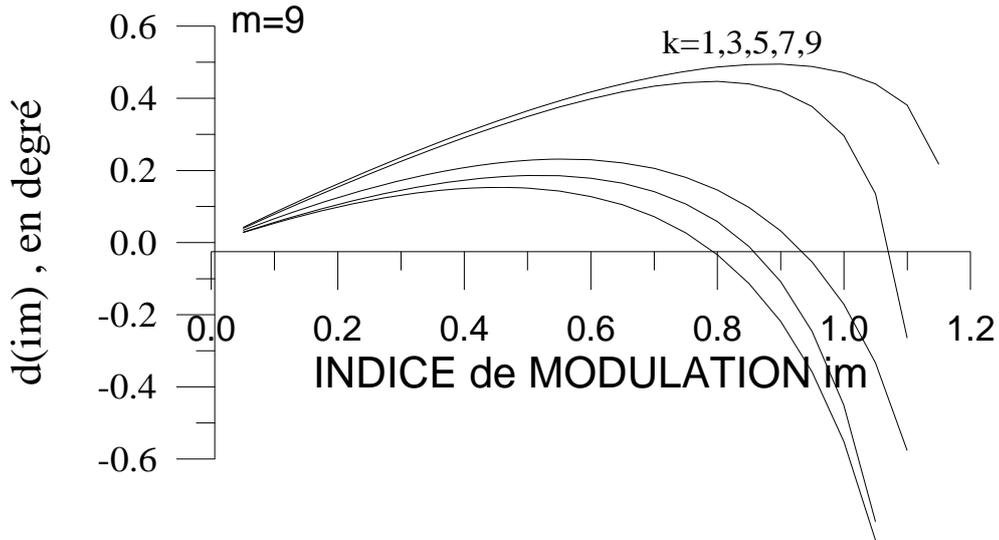


Figure II.7 Courbes représentant $d(im)$ en fonction de im , avec m égal à 9 et k variant de 1 à m . Cependant en observant ces courbes on constate qu'elles ont chacune une forme linéaire en début d'intervalle, passent par un maximum puis décroissent rapidement en fin d'intervalle. Pour approximer les courbes de la figure II.11, considérons les fonctions suivantes:

$$f(x) = a_1 \cdot x \quad a_1 > 0 \quad x \in [0, 1.15]$$

$$g(x) = a_2 \cdot x^p \quad a_2 > 0 \quad p > 0 \quad x \in [0, 1.15]$$

Considérons maintenant la différence $d(x)$ définie par:

$$d(x) = f(x) - g(x) = (a_1 \cdot x - a_2 \cdot x^p) \tag{II.29}$$

Calculons $d(x)$ pour x variant dans l'intervalle $[0, 1.15]$ et traçons les courbes correspondantes avec les paramètres suivants:

$$a_1 = 0.5, 1, 1.5 \quad a_2 = 1 \quad p = 4$$

Comparons les courbes des figures II.7 et II.8, elles présentent une grande similitude entre elles.

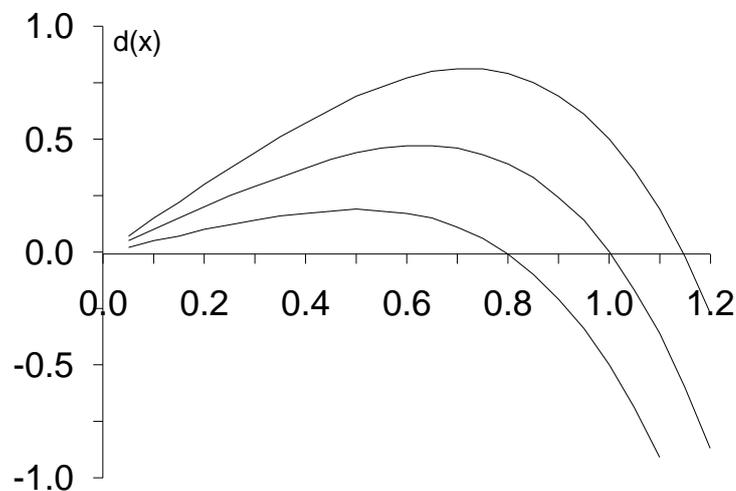


Figure II.8 Courbes représentant la fonction $d(x)$ pour: $a_2 = 1$ $p = 4$ $a_1 = 0.5, 1, 1.5$

Donc la fonction d'approximation de la fonction $d(im)$ sera une fonction de la forme :

$$\Phi(im) = A_0 \cdot im - A_1 (im)^p \quad \text{Avec } A_0, A_1, p \text{ réels} \quad (II.30)$$

Les fonctions élémentaires d'approximation sont donc:

$$\phi_0(im) = im \quad (II.31)$$

$$\phi_1(im) = -(im)^p \quad (II.32)$$

Le problème maintenant est de déterminer les trois coefficients A_0 , A_1 et p pour chacune des 65 courbes représentant les valeurs exactes des angles de commutation en fonction de m et k impair. On fixe p et pour chaque valeur de m et k on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.1. Pour chaque valeur de l'indice im , on calcule la différence (II.28).

On a donc 11 données numériques par courbe à approximer en utilisant les fonctions élémentaires d'approximation (II.32) (II.31).

On a donc:

$$d(im) = \alpha_k(im) - \left[\frac{60^\circ}{m+1} [(k+1) - im] \right]$$

$$\Phi(im) = A_0 \cdot im - A_1 (im)^p \quad \text{avec } p > 0$$

$$d(im) \cong \Phi(im)$$

Des trois équations précédentes on déduit l'équation générale d'approximation des valeurs exactes des angles de commutation $\alpha_k(im)$ pour k impair:

$$\alpha_k(im) \cong \frac{60^\circ}{(m+1)}(k+1) + \left[A_0 - \frac{60^\circ}{(m+1)} \right] \times im - A_1 \times (im)^p \quad (II.33)$$

$$m=3,5,\dots,21 \quad k=1,3,\dots,m \quad im \in [0, 1.15]$$

II.5.3 Cas k pair

De l'équation (II.33) on déduit que l'équation générale d'approximation des angles de commutation 'exacts' pour k pair aura la même forme c'est à dire:

$$m=3,5,\dots,21 \quad k=2,4,\dots,m-1 \quad im \in [0,1.15]$$

$$\alpha_k(im) = \frac{60^\circ}{(m+1)} \times k + A_0 \times im - A_1 \times im^p \quad (II.34)$$

De l'équation (II.34) on déduit la différence:

$$\alpha_k(im) - \frac{60^\circ}{(m+1)}(k+1) \cong A_0 \times im - A_1 \times (im)^p \quad (II.35)$$

Les fonctions élémentaires d'approximation sont donc:

$$\phi_0(im) = im \quad (\text{II.36})$$

$$\phi_1(im) = -(im)^p \quad (\text{II.37})$$

De la même façon il faut déterminer les trois coefficients A_0 , A_1 et p pour chacune des courbes représentant les valeurs exactes des angles de commutation en fonction de m et k pair. On fixe p et pour chaque valeur de m et k on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.1. Pour chaque valeur de l'indice im , on calcule la différence (II.35). On a donc 11 données numériques par courbe à approximer en utilisant les fonctions élémentaires d'approximation (II.37) et (II.36).

Le coefficient p est choisit de façon que l'erreur moyenne entre $d(im)$ et $\Phi(im)$ soit minimale. On répète le calcul en faisant varier m de 3 à 11 par valeurs impaires et pour chaque valeur de m on fait varier k de 1 à m par valeurs paires.

Les résultats donnés montrent que l'erreur moyenne, notée $emoy$, entre $d(im)$ et la valeur approchée $\Phi(im)$ est de l'ordre du centième de degré ($1/100^\circ$).

Donc les valeurs des angles de commutation sont données par les équations (II.33) pour les valeurs impaires de k , et (II.34) pour k pair.

$$\alpha_k(im) \cong \frac{60^\circ}{(m+1)}(k+1) + \left[A_0 - \frac{60^\circ}{(m+1)} \right] \times im - A_1 \times (im)^p \quad (\text{II.33})$$

Avec $m=3,5,\dots,21$; $k=1,3,\dots,m$ et $im \in [0,1.15]$.

$$\alpha_k(im) = \frac{60^\circ}{(m+1)} \times k + A_0 \times im - A_1 \times im^p \quad (\text{II.34})$$

Avec $m=3,5,\dots,21$; $k=2,4,\dots, m-1$ et $im \in [0,1.15]$.

II.6 Simulation de l'algorithme MLI on-line

L'objectif de cette partie est la simulation de cet algorithme MLI on-line, les équations (II.33) et (II.34) sont utilisées pour générer un signal MLI par l'outil Simulink de Matlab.

La partie puissance du système, constitué de l'onduleur triphasé et la charge, est simulée avec le logiciel PSIM de Power sim spécialisé dans l'électronique de puissance. Nous avons fait une cosimulation PSIM/SIMULINK en utilisant l'outil simcoupler du logiciel PSIM.

II.6.1 Schéma de simulation sous PSIM /SIMULINK

La figure II.9 donne le schéma de l'étage de puissance sous PSIM, il est constitué d'un onduleur triphasé alimentant un moteur asynchrone. Les signaux de commande sont générés par Matlab est représentée par la figure II.10.

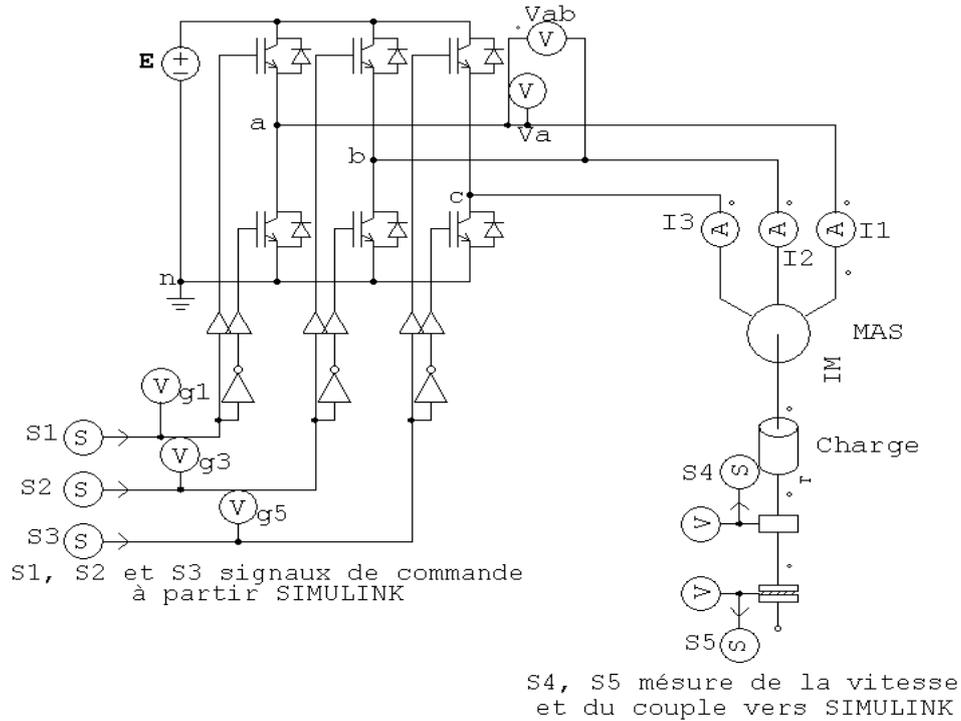


Fig. II.9. Partie puissance, l'onduleur triphasé et la charge sous PSIM

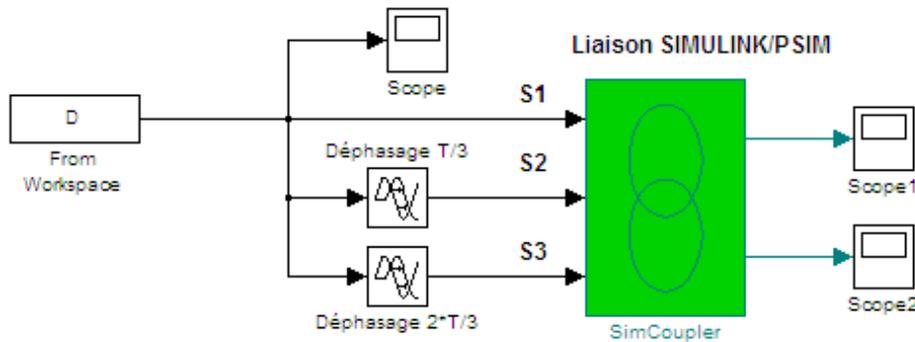


Fig. II.10. Partie de la commande MLI sous SIMULINK

II.6.2 Résultats de simulation

Pour la simulation nous avons choisi de varier la valeur du i_m de 0.1 jusqu'à sa valeur maximale $i_m=1$, et pour chaque cas on a associé un nombre m des angles de commutation.

Pour $m=1$ on a choisi 5 angles de commutation, on est dans la fréquence de 50Hz. Le nombre m augmente en diminuant la valeur de m . Pour chaque valeur de m et de m on a tracé les formes des tensions V_{an} et V_{ab} ainsi que leurs spectres fréquentiels. Les trois signaux de commande pour $m=1$ et $m=5$ sont donnés par la figure II.11 alors que la figure II.12 donne le signal de commande g_1 sur une période.

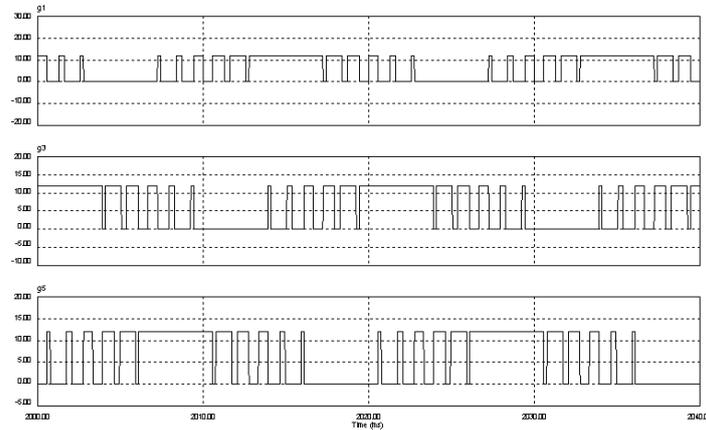


Fig. II.11. Les 3 signaux de commande g_1 , g_2 , g_3

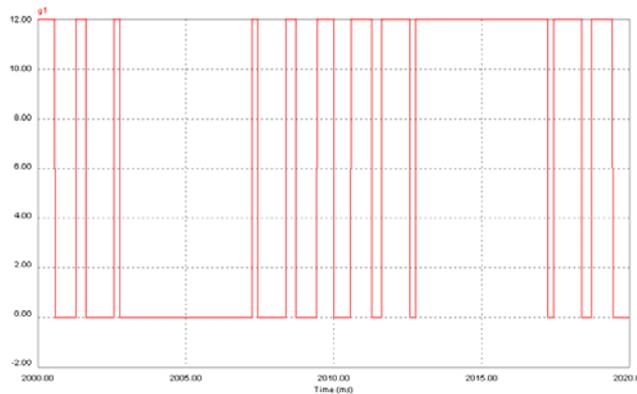


Fig. II.12. Forme du signal de commande g_1 sur une période.

Comme il est clair sur la figure II.12 le signal de commande présente 5 commutations par quart de période, avec une double symétrie par rapport au quart et à la demi-période.

Pour vérifier l'élimination des harmoniques désirés, nous avons tracé la tension simple et son spectre de fréquence, Figure II.13 et II.14 respectivement.

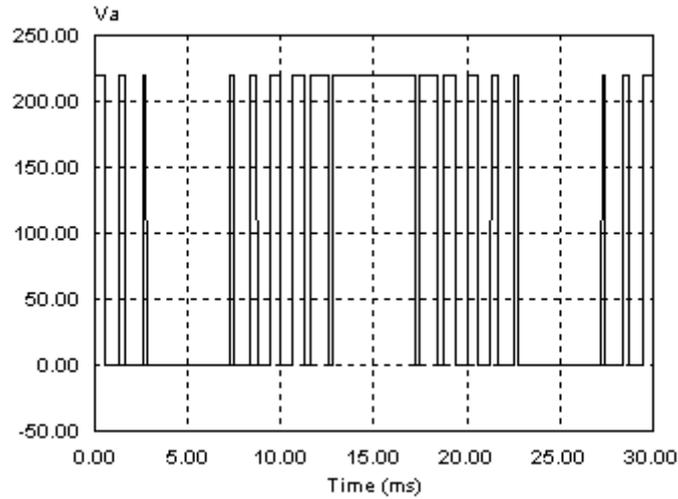


Fig. II.13. Tension simple V_{an} pour $im=1$ et $m=5$.

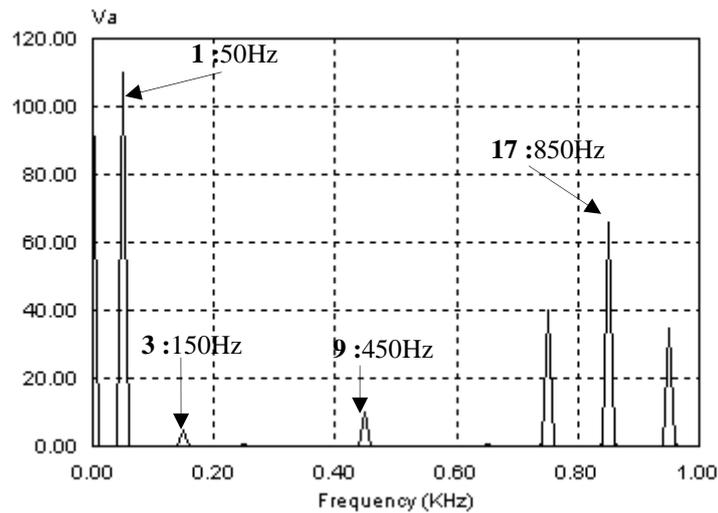


Fig. II.14. Spectre de la tension simple V_{an} pour $im=1$ et $m=5$

On remarque bien l'élimination des harmonique 5,7,11 et 13, le premier harmonique non éliminé c'est celui du rang 17 (850Hz). Comme il est clair sur la figure II.18 le fondamental est d'amplitude de 100% qui correspond à 110 V, il à une fréquence 50Hz. L'harmonique trois et ses multiples sont absents dans les spectres des tensions composées.

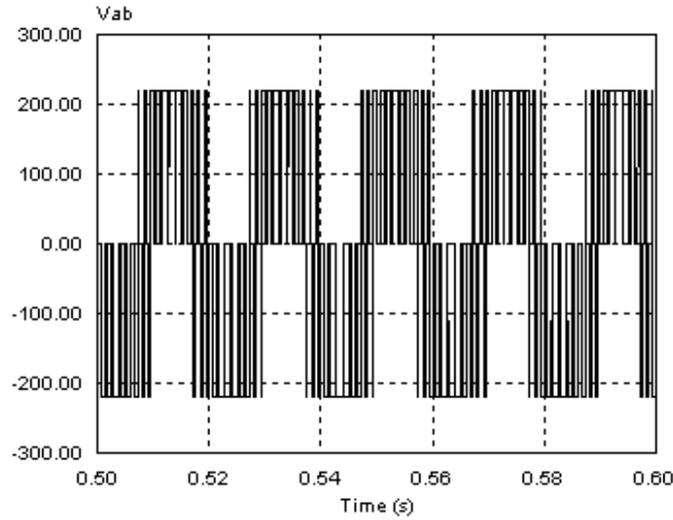


Fig. II.15. Tension composée de l'onduleur triphasé pour $i_m=1$ et $m=5$

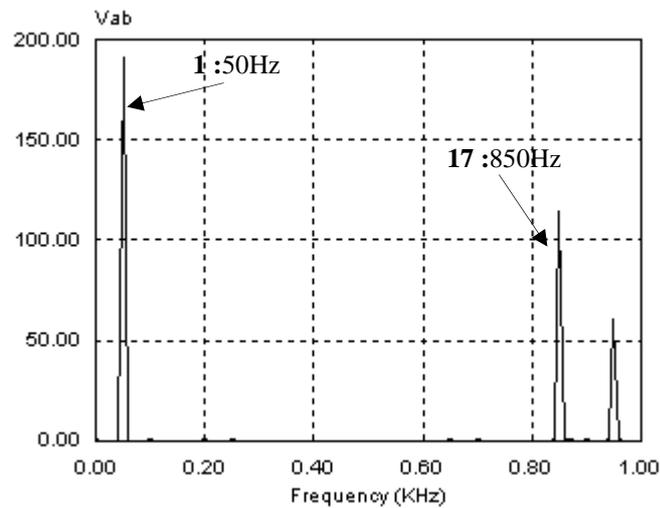


Fig. II.16. Spectre fréquentiel de la tension composée pour $i_m=1$ et $m=5$

La figure II.15 représente la tension composée V_{ab} pour les mêmes valeurs de i_m et de m ($m=5$ et $i_m=1$). Dans le spectre fréquentiel de cette dernière tous les harmoniques sont éliminés y compris l'harmonique trois et ses multiples. Le premier harmonique non éliminé est celui du rang 17 (850 Hz). La figure II.27 représente la forme du courant I_1 et son spectre fréquentiel.

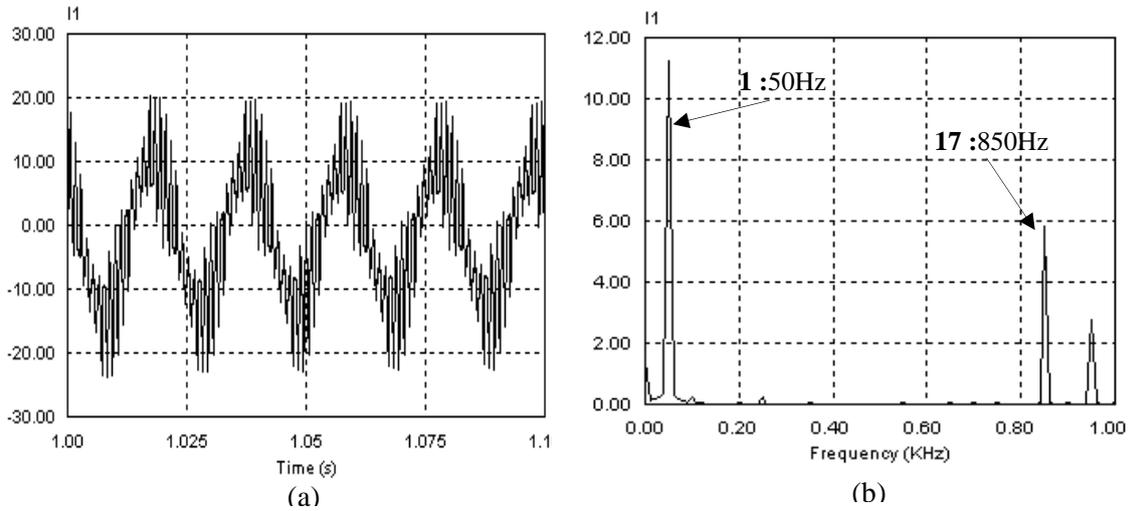


Fig. II.17. Forme du courants I_1 (a) et son spectre (b), pour $im=1$ et $m=5$

Pour $im=0.6$ on a choisi $m=7$, le fondamental est de valeur de 60% qui correspond à 66 V, la tension générée est de fréquence de $50*0.6=30\text{Hz}$. La figure II.18 donne la forme de la tension V_{an} et son spectre de fréquence. Pour $m=7$ on doit éliminer les harmoniques 5,7,11,13,17 et 19 comme il est clair dans la figure II.18 (b). Les harmoniques du rang multiple de trois sont éliminés dans le spectre de la tension composée, qui est représenté par la figure II.19 (a). Le premier harmonique non éliminé est donné par la formule générale $3*m+2$, pour $m=7$ c'est 23 ($23*30=690\text{Hz}$). La figure II.19 (b) donne le spectre du courant I_1 .

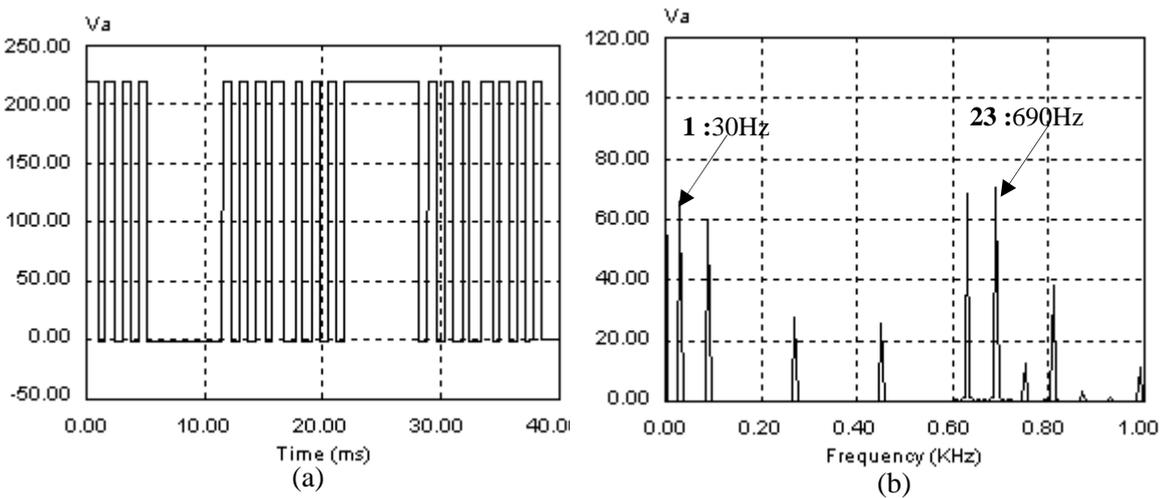


Fig. II.18. Tension simple V_{an} (a) et son spectre (b) pour $im=0.6$ et $m=7$

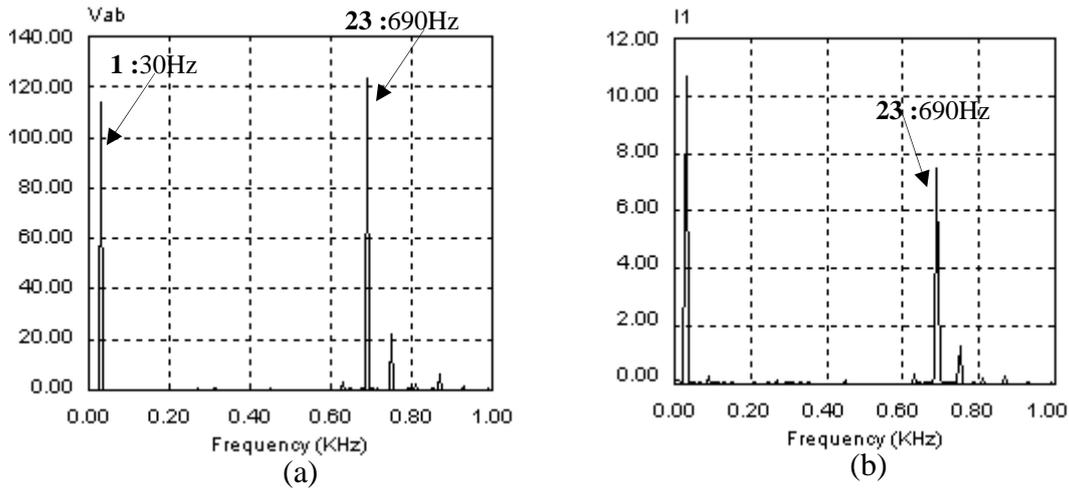


Fig. II.19. Spectre fréquentiel pour $i_m=0.6$ et $m=7$.

(a) : Tension composée V_{ab} . (b) : Courant I_1

La dernière valeur choisie est $i_m=0.1$, la tension délivrée par l'onduleur est de fréquence de $50 \cdot 0.1 = 5$ Hz, la période correspondante est $T=200$ ms. Le fondamental a une amplitude de 11V qui représente 10%. Les figures ci-dessous donnent les formes des différentes grandeurs.

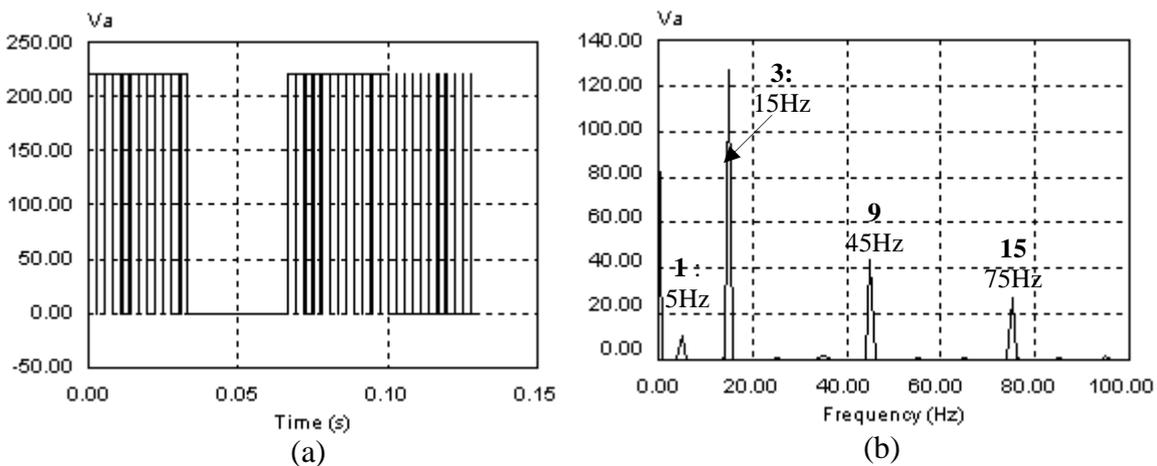


Fig. II.20. Tension simple V_{an} (a) et son spectre (b) pour $i_m=0.1$ et $m=23$.

Comme il est clair sur la figure II.20 (b), le spectre de la tension simple ne contient que le fondamental qui est de fréquence de $50 \cdot 0.1 = 5$ Hz et d'amplitude de $10\% = 11$ V, et aussi les harmonique du rang multiple de trois. .

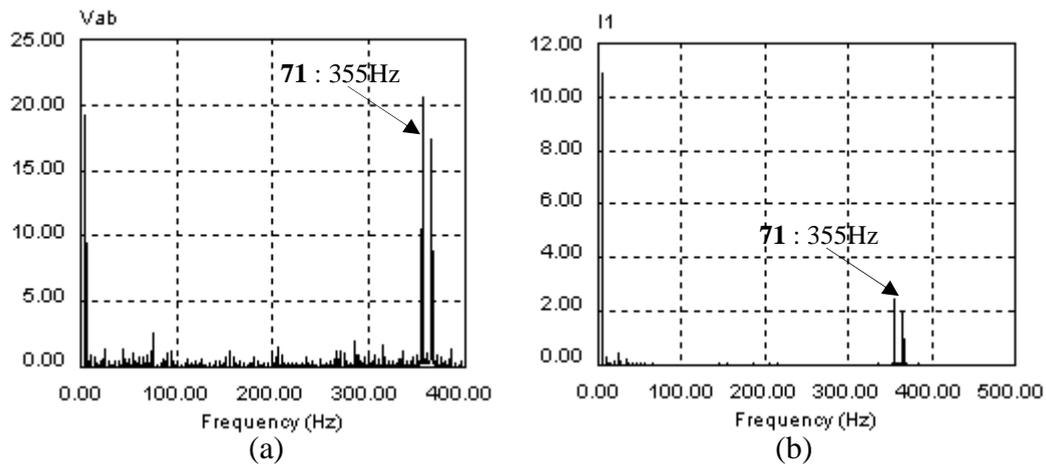


Fig. II.21. Spectre de fréquentiel pour $i_m=0.1$ et $m=23$.

(a) : Tension composée V_{ab} . (b) : Courant I_1

Conclusion

L'algorithme MLI on-line élaboré présente une grande précision dans le calcul des angles de commutation, il donne une efficacité dans l'élimination des harmoniques voulus. La cosimulation de cet algorithme dans PSIM et SIMULINK a prouvé son efficacité dans la commande de la vitesse des moteurs à induction.

Pour valider pratiquement l'efficacité de cet algorithme nous allons l'implémenter sur une architecture hardware de type FPGA.

CHAPITRE III

IMPLEMENTATION DE L'ALGORITHME MLI ON-LINE SUR CIRCUIT FPGA

Les composants programmables sont disponibles depuis les années 1970, mais leur utilisation est restée limitée essentiellement pour des raisons technologiques.

Le premier composant XILINX a été fabriqué en 1984, constitué de 85000 transistors, l'équivalent de 1000 portes logiques, fabriqué en 2 μ m. Actuellement, les FPGAs récents sont fabriqués avec une technologie de 65 nm, leur densité est d'environ 10 millions portes logiques [23]. Lors de la conception des systèmes électroniques industriels (circuits) plusieurs critères doivent être pris en considération: le prix, la consommation de l'énergie (surtout dans le cas des systèmes embarqués), les performances demandées par l'application et avant tout est ce que le matériel est bien choisi et correspond à l'algorithme à implémenter. Actuellement les deux moyens pour implémenter un contrôleur sont les DSPs et les FPGAs. Selon la nature de l'algorithme le concepteur peut faire un choix entre ces deux possibilités. La figure III.1 illustre ce principe.

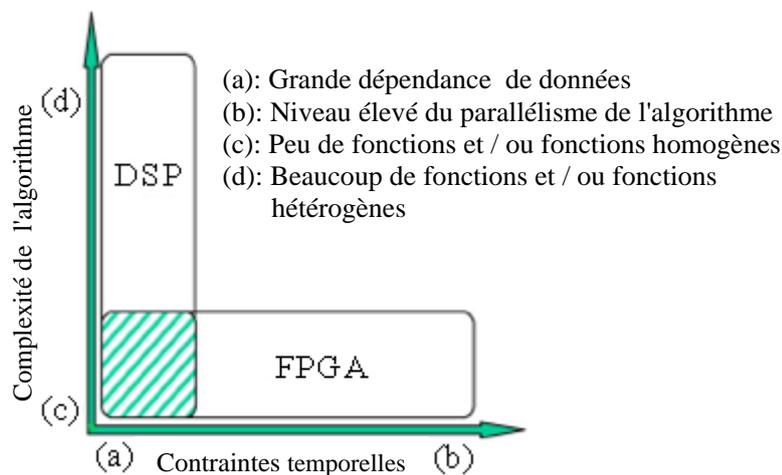


Fig. III.1. Domaines d'utilisation des DSPs et des FPGAs

L'axe X de ce graphe représente les contraintes temporelles de l'algorithme, alors que l'axe Y représente la complexité de l'algorithme.

Dans le chapitre précédent nous avons vu quelques techniques de commande pour les onduleurs de tension, et nous avons élaboré un algorithme MLI on-line permettant la commande de la vitesse d'un moteur asynchrone triphasé. Pour vérifier l'algorithme et valider expérimentalement les résultats de simulation nous avons choisi d'implémenter cet algorithme sur un circuit reconfigurable de type FPGA.

III.1 Les circuits FPGA

III.1.1 Structure générale des circuits FPGAs

Les circuits logiques programmables sont des circuits intégrés qui permettent la réalisation de circuits logiques à grand nombre de variables d'entrée et de sortie, selon la spécification de l'utilisateur et en utilisant un même support matériel. Dans la littérature anglo-saxonne on rencontre deux appellations différentes pour de tels circuits intégrés: le Programmable Logic Device – PLD et le Field Programmable Device – FPD. La différence entre ces deux appellations est liée à la quantité et la complexité de la logique réalisable au sein de ces circuits, en général beaucoup plus importantes dans le cas des circuits FPD [29].

Les circuits FPGA - Field Programmable Gate Arrays – sont des circuits FPD, construits autour d'une matrice régulière (array) constituée d'un grand nombre de cellules logiques élémentaires programmables, d'un réseau dense d'interconnexion, lui aussi programmable et d'un certain nombre de bornes d'entrées et de sorties [23] [27] [32].

La disposition des différents éléments qui constituent un circuit FPGA est représentée de façon schématique dans la Figure III.2.

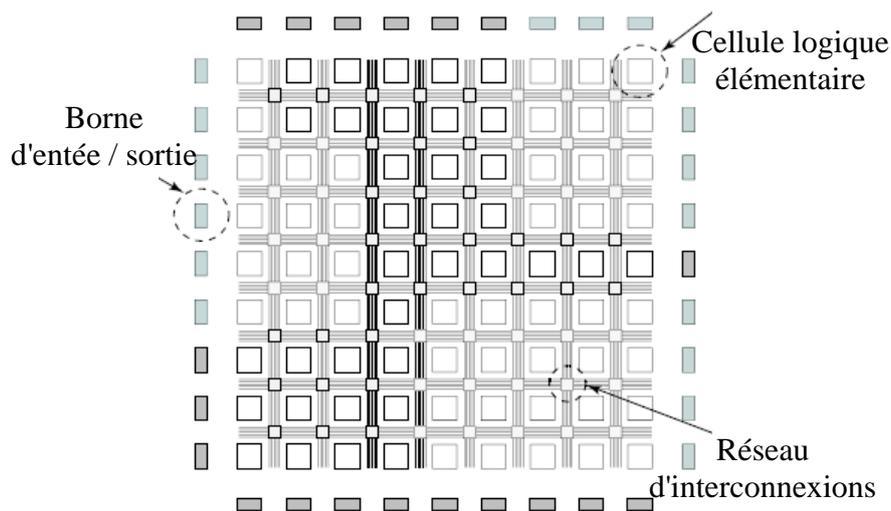


Fig. III.2. Architecture des circuits FPGAs.

L'architecture d'une cellule logique élémentaire varie d'un concepteur et d'une famille de FPGA à l'autre. Quelque soit le producteur ou la famille considérée, une cellule logique

élémentaire est toujours composée de quelques mémoires de mots d'un bit, généralement de quelques bits d'adresse. Ces mémoires sont destinées à la sauvegarde des différentes tables de vérité des fonctions logiques définies par le circuit à réaliser. Chaque cellule logique est également dotée de bi-stables programmables permettant la réalisation des circuits synchrones [29].

Un bloc logique de base dans les FPGAs XILINX est constitué d'une table LUT à quatre entrées, qui sert à implémenter des fonctions logiques combinatoires; un registre pour mémoriser la sortie du générateur logique. La figure ci-dessous illustre ce principe [27].

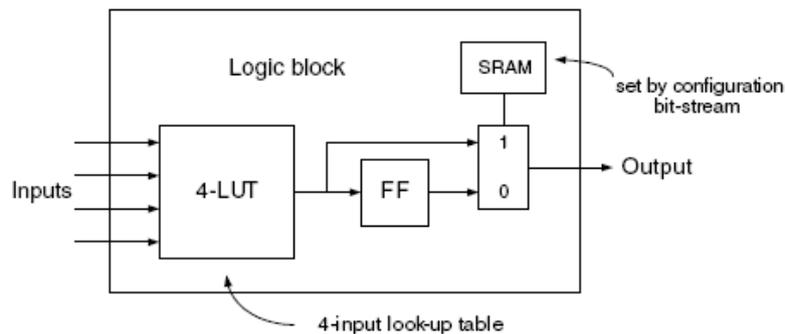


Fig. III.3. Structure d'une cellule logique élémentaire.

III.1.2 Méthodes de programmation

Il existe trois méthodes pour programmer un circuit FPGA [27] [30].

- **SRAM:** Les connexions sont réalisées en rendant les transistors passants. L'avantage de cette technologie est qu'elle permet une reconfiguration rapide au sein même du circuit. Le principal inconvénient est la surface nécessaire pour la SRAM. Elle nécessite l'utilisation d'une mémoire standard chargée à l'initialisation.

XILINX et ALTERA utilisent cette technologie. Le composant peut être programmé infiniment.

- **La technologie anti-Fuse:** Un état anti-fuse réside en un état d'une haute impédance. Il peut être programmé dans un état de faible impédance ou état "fused". Il s'agit d'une technologie moins chère que la SRAM, elle permet d'atteindre des vitesses plus élevées et occupe moins de place sur le circuit. Par contre, un tel FPGA ne peut être programmé qu'une seule fois. Les performances électriques sont supérieures à la technologie SRAM (minimisation des effets RC due à la faible surface). Actel et Quicklogic utilisent cette technologie.

- **EPROM/EEPROM:** Cette méthode est la même que celle utilisée dans les mémoires EPROM. L'EEPROM est reprogrammable. La puce fonctionne seule. La surface moyenne et les caractéristiques électriques sont semblables à la SRAM.

III.1.3 Ressources typiques des FPGA

La quantité des ressources logiques disponibles dans un circuit FPGA dépend du degré d'intégration et de la taille du support utilisé pour sa construction. Elle sera également influencée par le choix concernant l'architecture de la cellule logique élémentaire: taille et le nombre des mémoires, leur agencement et le type d'interconnexions, présence ou non d'une quelconque logique et/ou de circuits arithmétiques dédiés, etc... La quantité des ressources logiques disponibles est le plus souvent exprimée par le nombre de cellules élémentaires et/ou par le nombre équivalent de portes logiques [29].

Le tableau indique la quantité des ressources logiques qui caractérisent quelques circuits FPGAs de faible, moyenne et haute densité produits par les trois principaux producteurs de circuits logiques programmables: Xilinx, Actel et Altera. L'architecture d'une cellule logique élémentaire (CLE) varie fortement d'un producteur à l'autre. Ainsi chez Xilinx une seule CLE possède 8 mémoires à 4 bits d'adresse (16 mots d'un bit), huit bistables programmables et une logique dédiée par mémoire pour des opérations arithmétiques. Chez Actel, une CLE est composée d'une mémoire à 5 bits d'adresse (32 mots d'un bit) et d'un bistable programmable toutes les deux mémoires. Enfin une CLE de chez Altera est une mémoire à 4 bits d'adresse (16 mots d'un bit), associée à un bit bistable programmable et une logique dédiée pour l'addition.

Producteur	Famille	Circuit	CLE	Np	Nes
XILINX	Virtex-II	XC2V250	64	0.250	200
		XC2V1500	1920	1.5	528
		XC2V8000	11648	8	1108
	Virtex-II-Pro	XC2VP2	1408		204
		XC2VP30	13696		644
		XC2VP125	55616		1200
Actel	Axcelerator	AX125	1344	0.125	168
		AX500	5376	0.5	336
		AX2000	21504	2	684
Altera	Stratix	EP1S10	10570		426
		EP1S30	32470		726
		EP1S80	79040		1238

Tab. III.1. Ressources typiques dans les circuits FPGA

Np est le nombre de portes en million, Nes est le nombre des entrées/sorties.

III.1.4 Applications des FPGAs

L'augmentation de la densité des circuits FPGAs, combinée avec le développement de langages évolués de description de matériel et des outils permettant de passer de façon automatique de la description à la réalisation matérielle, a ouvert la voie à l'application. Les domaines abordés sont très variés et touchent la cryptographie, le traitement des images numériques, la 3D,....la liste annoncée des applications est loin d'être exhaustive [23].

III.1.5 Architecture des circuits FPGA Virtex II de Xilinx

Comme tout circuit FPGA, les FPGAs de la famille Virtex II de Xilinx sont construits autour de trois éléments essentiels: les cellules logiques élémentaires, le réseau d'interconnexion et les bornes d'entrées/sorties. En plus de ces ressources, les FPGAs de cette famille possèdent un certain nombre de mémoires RAMs et des circuits multiplicateurs dédiés. Les circuits FPGAs de la famille Virtex II-Pro sont même dotés de microprocesseurs embarqués, par exemple, l'unité FPGA XC2VP125 possède pas moins de 4 processeurs PowerPC-405 [27] [29].

III.1.5.1 Cellule logique élémentaire

III.1.5.1.A Les CLB (configurable logic bloc):

La figure ci-dessous, nous montre le schéma d'un CLB. Dans la terminologie Xilinx une cellule logique élémentaire est appelée Configurable Logic Block - CLB. Bien que ce nom soit resté identique tout au long de l'évolution des différentes familles des FPGAs, l'architecture qu'il désigne a fortement changé depuis l'introduction des circuits de la famille Virtex.

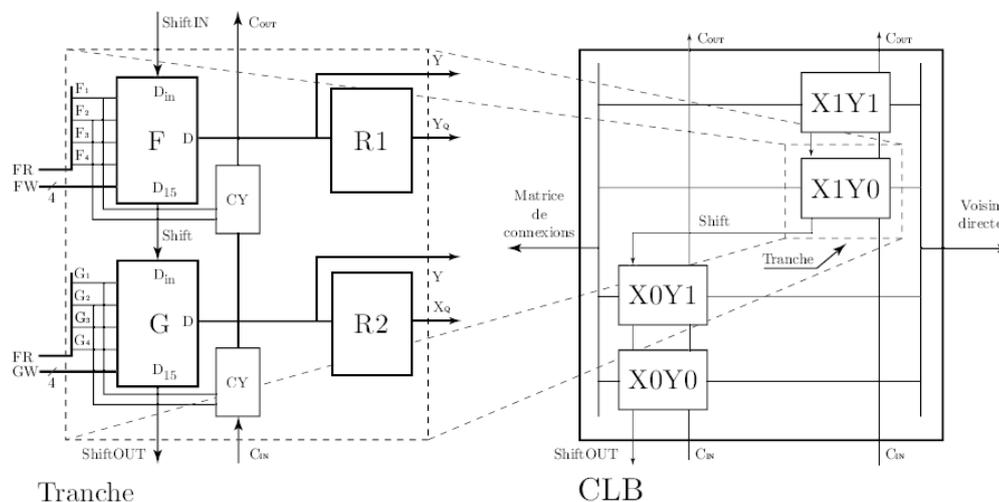


Fig. III.4. Architecture d'un CLB des circuits FPGA de la famille Virtex.

A partir de la première génération de la famille Virtex, un CLB est composé de quatre sous cellules identiques appelées tranches (slices) représentées schématiquement à la Figure III.4.

Chacune des tranches d'un CLB possède un certain nombre de ressources logiques. Nous décrivons les plus importantes d'entre elles :

- **Mémoires F et G** : Il s'agit des mémoires de 16 bits que l'on peut adresser avec deux bus d'entrée indépendants de quatre bits chacun. Ces mémoires possèdent deux sorties :

- Une sortie D qui donne le contenu de la mémoire selon l'adresse présente sur le bus FR (ou GR).

- Une sortie D15, le dernier bit adressé par la mémoire.

Chaque mémoire possède également une entrée supplémentaire Din. Cette entrée, combinée avec la sortie D15, est utilisée pour la construction des registres à décalage à plus grand nombre de bits à partir de plusieurs mémoires F (ou G) mises en série.

- **Logique dédiée d'addition CY**: La logique supplémentaire par tranche permet d'effectuer efficacement l'opération d'addition (ou de soustraction). A cette logique est associée une connexion spécifique permettant de tenir compte d'un éventuel report de l'étage précédant et de propager le report calculé aux tranches situées immédiatement au dessus de la tranche considérée (connexions Cin et Cout). Il est ainsi possible de construire des additionneurs de mots d'un plus grand nombre de bits.

- **Bi-stables programmables R1 et R2** : La sortie D de la mémoire F (ou G) peut être dirigée vers deux bi-stables programmables R1 et R2. Ces bi-stables peuvent être utilisés pour la réalisation des latch ou des flip-flops habituels lors de réalisation des circuits synchrones.

- **Multiplexeurs** - L'ensemble des multiplexeurs (non représentés à la Figure III.4) permettent la réalisation de multiplexeurs à grand nombre de canaux.

Dans chaque tranche, les sorties D des mémoires F et G peuvent être :

- soit dirigées directement vers d'autres circuits combinatoires et/ou séquentiels (sorties X, Y).

- soit mémorisées dans les deux éléments mémoire R1 et R2 dans le cas de la réalisation d'un circuit synchrone (sorties XQ, Y Q).

Dans les deux cas, la connexion des sorties avec d'autres CLBs se fait soit via des connexions dédiées pour des tranches adjacentes au CLB considéré, soit via la matrice des connexions à des CLBs situés plus loin dans le FPGA.

III.1.5.1.B Modes de fonctionnement d'un CLB

Un CLB se présente donc comme une ressource logique polyvalente. En fonction de la configuration attribuée, il peut adopter l'un de plusieurs modes de fonctionnement distincts, dont nous présentons les plus importants :

- **Générateur des fonctions logiques**

Les mémoires des CLBs (en fait les mémoires des tranches) contiennent des tables de vérité permettant le calcul des fonctions logiques. Un CLB peut donc réaliser au maximum 8 fonctions logiques différentes, chacune sur 4 variables d'entrées différentes.

- **Registres**

Chaque mémoire (F ou G) peut fonctionner comme un registre à décalage de 16 bits à écriture synchrone mais à lecture asynchrone. Les deux éléments mémoire peuvent cependant être utilisés pour une lecture synchrone depuis F et G. La lecture d'un poids particulier d'un registre fait à l'aide des mémoires est effectuée par l'intermédiaire des variables d'entrée, en fait les 4 bits d'adresses des mémoires F ou G. Indépendamment de ce registre à décalage, et donc des mémoires F et G, les R1 et R2 peuvent être utilisés pour la réalisation de différents types de registres à lecture parallèle. Un registre de 8 bits nécessitera donc pour sa réalisation un seul CLB.

- **Mémoire**

Comme chaque CLB possède 8 mémoires de 16 bits, chacune des mémoires peut être organisée pour contenir des mots de 1, 2, 4 et 8 bits à simple accès, ou en des mémoires de mots de 1, 2, 4 bits en double accès. L'adressage double, dans le cas d'un cycle d'écriture/lecture simultanées aux adresses différentes, est fait à l'aide des deux bus supplémentaires WF et WG de mémoires F et G.

- **Multiplexeurs**

Les mémoires F et G ainsi que les différents multiplexeurs associés au CLB permettent la réalisation des multiplexeurs de 4, 8, 16 et 32 canaux à l'aide de 1, 2, 4 et 8 tranches respectivement.

- **Logique dédiée pour l'arithmétique**

Chaque CLB ayant des ressources logiques spécifiques pour le calcul et la propagation du report (le bloc CY à la Figure III.4) permet une implémentation très efficace des circuits arithmétiques tels que les additionneurs ou les soustracteurs. Il est ainsi possible de réaliser un additionneur ou un soustracteur complet sur deux mots d'un bit en tenant compte d'un

éventuel report de l'étage précédent (Cin) à l'aide de seulement une tranche de CLB. Le report généré est directement transmis à la tranche supérieure Coût pour la construction des additionneurs ou des soustracteurs de mots codés sur plusieurs bits.

- **Calcul de SdP**

Chaque tranche possède une porte logique OU généralisée (notée ORCY) ayant comme entrée le signal Cout de la tranche considérée et le même signal provenant de la tranche située immédiatement à sa gauche . Les sorties des différentes portes ORCY provenant des tranches adjacentes peuvent être mises en série. Il est ainsi possible de calculer aisément des expressions logiques de type somme des produits - SdP, sans pour autant consommer de CLBs supplémentaires pour calculer la somme après avoir calculé les produits.

- **Bloc de mémoire RAM**

Il s'agit de mémoires RAM d'une capacité de 18Kbits qui peuvent être exploitées en simple ou en double accès (cycles écriture/lecture simultanées sur des adresses différentes).

Ces mémoires sont configurables et peuvent être organisées en mémoires de 0.5K, 1K, 2K, 4K, 8K et 16K mots de 36, 18, 9, 4, 2 et 1 bits respectivement.

- **Multiplicateur**

A chaque bloc de mémoire RAM est associé un circuit multiplicateur de deux mots de 18 bits. Plusieurs de ces circuits peuvent calculer de façon simultanée la multiplication sur des données sauvegardées ailleurs dans le FPGA, mais sont optimisés pour les opérations sur des opérands provenant des blocs de mémoire RAM. L'utilisation des circuits de multiplication s'avère particulièrement utile lors de l'implémentation des applications de traitement de signal, caractérisées par un nombre important de cycles lecture - multiplication - accumulation de résultats.

Le nombre de circuits multiplicateurs (également des blocs de mémoire RAM) dépend de la taille et de la densité du circuit FPGA considéré. Ainsi nous disposons d'un multiplicateur et d'un bloc de mémoire RAM toutes les 4 lignes de CLBs et ceci toutes les 2, 4 ou 6 colonnes, en fonction de la densité du FPGA (6 colonnes pour les FPGAs les plus denses).

- **Digital Clock Manager - DCM**

L'architecture des circuits FPGAs Virtex II offre une ressource particulière permettant une gestion efficace des différents signaux d'horloge (Digital Clock Manager - DCMs) nécessaires au fonctionnement optimal d'un circuit logique réalisé dans un FPGA. Le rôle principal de cette ressource consiste à générer des signaux d'horloge internes au FPGA mais parfaitement

en phase avec les horloges sources venant de l'extérieur. Ainsi les délais de distribution du signal d'horloge dans les endroits éloignés de la source seront complètement éliminés. Grâce à ces unités il est également possible de générer un grand nombre de fréquences d'horloge différentes, à travers un ensemble des diviseurs/multiplicateurs de fréquence.

Comme pour les blocs de mémoire RAM et le multiplicateur, le nombre de DCMs varie en fonction de la taille de FPGA. On compte deux ressources DCM toutes les 2, 4 ou 6 colonnes des CLBs, pour les FPGAs les plus denses.

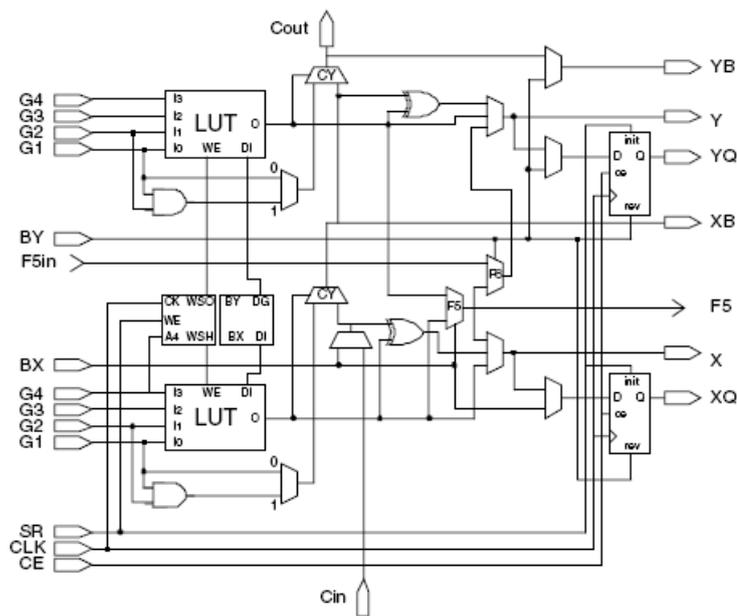


Fig. III.5. Cellule logique élémentaire (CLB)

III.1.5.2 Les IOB (input output bloc)

Le rôle principal des blocs d'entrée/sortie (Input/Output Block - IOB) est d'assurer la communication entre la logique interne au FPGA et le monde extérieur au circuit et ceci pour l'ensemble des standards électriques les plus répandus.

Chaque IOB peut être programmé de façon à ce que la patte de circuit qui lui est associée puisse être utilisée soit comme une entrée, soit comme une sortie, soit les deux (une patte bidirectionnelle). Chaque IOB dispose également de 6 éléments mémoires configurables en tant que bascule D ou en tant que latch. Pour un mode de fonctionnement particulier de l'IOB (entrée, sortie ou bidirectionnelle) les deux éléments de mémoire permettent de doubler le débit de données entrantes/sortantes puisque leur synchronisation peut être faite avec deux signaux d'horloge de même période mais déphasés de 180° .

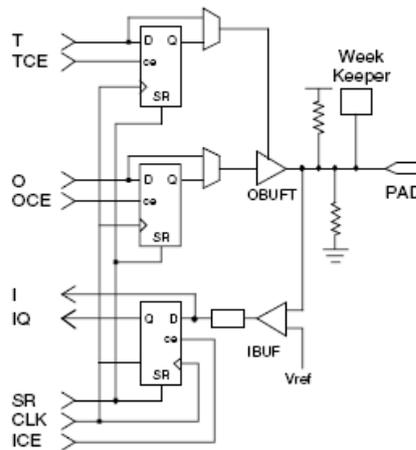


Fig. III.6. Input Output Block (IOB)

III.1.5.3 Les différents types d'interconnexions

Le réseau d'interconnexions permet la réalisation des connexions (i.e. le routage) entre les différentes tranches au sein d'un même CLB, entre les différents CLBs, les DCMs, les multiplicateurs, la mémoire RAM et les IOBs utilisés pour la construction d'un circuit donné.

Au sein d'un même CLB, les différentes tranches peuvent communiquer via des connexions rapides. Une matrice des connexions programmables (Programmable Switch Matrix - PSM) est associée à chaque CLB, IOB, DCM, à la mémoire RAM et au circuit multiplicateur.

Le rôle de cette matrice est de réaliser une connexion programmée entre les conducteurs "entrants" et "sortants". La programmation de la connexion est établie à l'aide d'un mot de mémoire défini par la configuration. Grâce aux PSMs toute ressource peut donc accéder à toute autre ressource, quelque soit l'endroit où elle se trouve dans le circuit FPGA.

Les conducteurs "entrants" et "sortants" d'une matrice PSM sont disposés entre les lignes et les colonnes des CLBs.

III.1.6 Performance des FPGAs Virtex

La performance globale d'un circuit réalisé au sein d'un FPGA dépendra :

- Des performances respectives de tous les éléments individuels du FPGA; ce que nous désignerons comme la performance brute du circuit

- Des ressources effectivement utilisées pour la construction du circuit définitif.

En ce qui concerne la performance brute des FPGAs, la famille Virtex II propose pour chaque circuit particulier trois classes de performance notées : -4, -5, -6 ; -6 étant l'indice qui désigne les circuits les plus performants.

Pour un circuit donné, la performance va dépendre de sa complexité : le nombre et la position des blocs IOB, le temps de transmission des signaux depuis les entrées jusqu'à la logique

combinatoire et/ou séquentielle, la vitesse avec laquelle celle-ci calculera les fonctions logiques réalisées à l'aide d'un ensemble des mémoires et le temps de transmission des signaux vers les sorties.

Circuits	CLBs	Tranches	Multiplic	RAM	DCMs	Max E/S
XC2V40	8x8	256	4	72	4	88
XC2V80	16x8	512	8	144	4	120
XC2V250	24x16	1.536	24	432	8	200
XC2V500	32x24	3.072	32	576	8	264
XC2V1000	40x32	5.120	40	720	8	432
XC2V1500	48x40	7.200	48	864	8	528
XC2V2000	56x48	10.752	56	1,008	8	624
XC2V3000	64x56	14.336	96	1,728	12	720
XC2V4000	80x72	23.040	120	2,160	12	912
XC2V6000	96x88	33.792	144	2,592	12	1104
XC2V8000	112x104	46.592	168	3,024	12	1108

Tab. III.2. Ressources de circuits FPGAs de la famille Virtex II de XILINX

III.2 Le VHDL

III.2.1 Historique

Le VHDL - VHSIC (Very High Speed Integrated Circuit Hardware Description Language), été demandé par le DOD (Département de la défense américain) pour décrire les circuits complexes, de manière à établir un langage commun avec ses fournisseurs. C'est un langage, standard IEEE 1076 depuis 1987, qui aurait du assurer la portabilité du code pour différents outils de travail (simulation, synthèse pour tous les circuits et tous les fabricants). Une mise à jour du langage VHDL s'est faite en 1993 (IEEE 1164) et en 1996, la norme 1076.3 a permis de standardiser la synthèse VHDL [25] [28].

III.2.2 Utilité du VHDL

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 et autant que possible optimisés pour leurs propres circuits.

- **Spécification**

Etabli en premier lieu pour de la spécification, c'est dans ce domaine que la norme est actuellement la mieux établie. Il est tout-à-fait possible de décrire un circuit en un VHDL standard pour qu'il soit lisible de tous. Certains fabricants (de circuits ou de CAO) adaptent ce langage pour donner à l'utilisateur quelques facilités supplémentaires, au détriment de la portabilité du code. Heureusement, il y a une nette tendance de la part des fabricants à revoir leurs positions et à uniformiser le VHDL. Le rapprochement se fait, comme précité, autour du VHDL de Synopsys. Il est donc probable que l'on s'approche d'un vrai standard VHDL et non plus d'un standard théorique. Il y aura toujours des ajouts de la part des fabricants, mais il ne s'agira plus d'une modification du langage, mais de macros offertes à l'utilisateur pour optimiser le code VHDL en fonction du circuit cible (en vue de la synthèse).

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.

- **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.

- **Conception**

Le VHDL permet la conception de circuits avec une grande quantité de portes. Les circuits actuels comprennent, pour les FPGA par exemple, entre 500 et 100'000 portes et ce nombre augmente très rapidement. 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. Cependant, ce langage étant conçu en premier lieu pour de la spécification, certaines variantes du langage ne sont pas pour l'instant utilisables pour la conception.

Il faut noter que lorsqu'il s'agit de concevoir quelque chose en VHDL, il faut le faire avec une étude au préalable. 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. Donc avant de concevoir une ALU, un processeur RISC, une interface PCI ou d'autres éléments de cette complexité, il peut être judicieux de choisir un circuit cible en fonction des besoins et d'acheter la macro offerte par le constructeur. Il est bien évident qu'il faudra évaluer les besoins (performance du code nécessaire, quantité de pièces à produire) et le coût d'une telle macro.

III.3 Etapes de réalisation d'un projet autour d'un circuit FPGA

L'expansion du monde des circuits logiques programmables et en particulier des circuits FPGAs est étroitement liée au développement de la technologie de fabrication des circuits intégrés permettant la fabrication d'unités de plus en plus denses. Mais cette expansion n'aurait jamais pu avoir une telle ampleur sans le développement, en parallèle, d'outils performants permettant de faciliter le processus de conception des circuits logiques avec de la logique programmable.

Ces outils assurent d'une part la description aisée des circuits logiques complexes et d'autre part la réalisation automatique de circuit à partir de ces descriptions.

La réalisation d'un circuit logique, en fait le passage de sa description à l'information nécessaire pour la réalisation d'une implémentation physique du circuit dans un FPGA (i.e la configuration), est dans la littérature anglo-saxonne appelée design flow. Ce processus est habituellement décomposé en plusieurs étapes distinctes. Actuellement, grâce aux outils informatisés, certaines étapes de ce processus peuvent être réalisées de façon entièrement automatique (notamment la synthèse, l'implémentation ainsi que la génération de la configuration pour le FPGA cible) [29] [30].

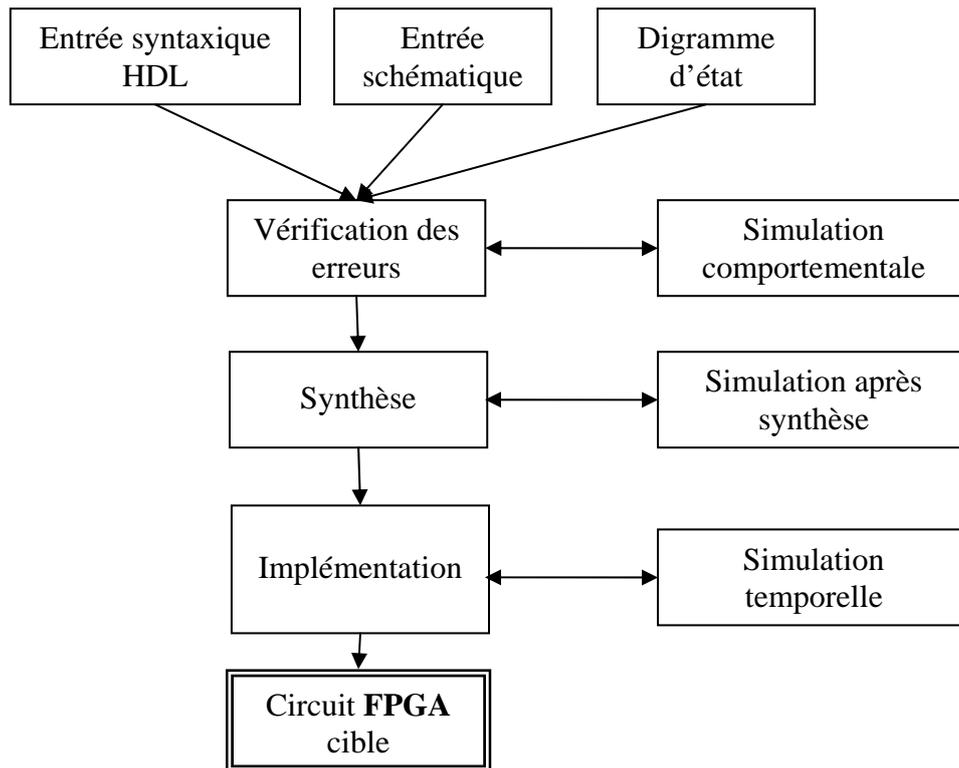


Fig. III.7. Etapes de réalisation d'un projet sur circuit FPGA (design flow)

III.3.1 Description

La première étape du design flow comprend la description formelle du circuit à partir des spécifications. Aujourd'hui la description des circuits logiques est faite la plupart de temps à l'aide des langages de description de matériel (Hardware Description Language - HDL) à haut niveau d'abstraction tels que VHDL ou Verilog. Souvent, lorsque l'utilisateur souhaite rester proche du matériel afin d'obtenir la meilleure performance possible de son circuit, la description est faite de façon schématique. Il existe d'autres formalismes permettant de décrire un circuit logique, souvent proposés par un producteur de FPGA particulier, comme par exemple l'éditeur graphique des machines à états finis (FSM) chez Xilinx. La description d'un circuit peut être faite exclusivement à l'aide d'un formalisme bien particulier, ou combiner des formalismes différents. Les différentes parties d'un même circuit logique complexe peuvent donc être décrites de façons différentes avant d'être traduites en une configuration particulière pour un (ou plusieurs) FPGA cible.

III.3.2 Vérification des erreurs

La vérification fonctionnelle de la description se fait à l'aide d'un simulateur, en l'occurrence le logiciel ModelSim. A la description de circuit sont associés les différents stimuli sous forme des bancs d'essai, décrits également en langage VHDL. La description du

circuit et le fichier de stimulus sont ensuite passés au logiciel de simulation, permettant d'effectuer une vérification fonctionnelle de la description.

III.3.3 Synthèse

Lors de l'étape de la synthèse, les différentes descriptions établies lors de la première étape sont ramenées à une description du circuit unique, optimisée pour une performance maximale ou une occupation minimale des ressources. Après l'étape de la synthèse il est possible d'effectuer une analyse fonctionnelle du circuit (vérification de la description à travers la simulation) et de se rendre compte d'éventuelles erreurs, avant de passer à l'étape de l'implémentation. L'analyse fonctionnelle n'est qu'une vérification partielle car elle ne peut tenir compte des délais introduits par le circuit réel, celui-ci n'étant pas encore réalisé. Lors d'une analyse fonctionnelle tous les délais du circuit obtenu après la synthèse sont considérés comme nuls.

Après la vérification de la description, la synthèse du circuit peut être effectuée à l'aide du programme XST - Xilinx Synthesis Technology. Outre la possibilité de visualiser le circuit sous forme d'un schéma RTL, nous pouvons disposer d'une première estimation de la performance pour le circuit décrit. Bien entendu la performance donnée ne sera qu'une indication, car la performance réelle ne pourra être obtenue qu'après le placement et le routage définitifs [26] [32].

III.3.4 Implémentation

L'étape de synthèse fournit au processus d'implémentation un fichier contenant la description physique de circuit, sous forme d'un ensemble de primitives (briques de base) des circuits FPGAs Xilinx (il s'agit d'un fichier portant une extension NGD, Native Generic Database). Mis à part la description du circuit, le processus d'implémentation proprement dit, en fait les trois sous-étapes : translate, map et place and route.

Le mapping consiste à traduire le circuit logique obtenu par le processus de synthèse en mémoires des cellules logiques élémentaires d'un circuit FPGA particulier.

Lors de la sous-étape de placement, une cellule logique existante, avec sa position unique au sein du FPGA cible, est attribuée à chaque cellule logique nécessaire pour la réalisation du circuit. Ceci est bien entendu fait sous la contrainte des longueurs minimales des connexions entre les différentes cellules.

Enfin la dernière sous-étape, le routage, consiste à établir les différentes connexions physiques entre les cellules élémentaires placées.

Ces étapes nécessitent la définition d'un ensemble de paramètres permettant de contrôler leur déroulement. Ces paramètres peuvent être classés en deux groupes distincts : les paramètres des programmes d'implémentation et les contraintes de la performance.

III.3.4.1 Paramètres des programmes d'implémentation

Les paramètres liés aux différents programmes d'implémentation sont introduits via l'interface graphique de la plate-forme de développement ISE. Il s'agit principalement : de l'effort que le programme de placement et de routage va fournir afin d'obtenir la performance du circuit souhaité et le choix d'une des 100 différentes tables de coûts utilisée pour le placement. L'influence de choix d'une table de coût sur la performance du circuit définitif peut être significative car chaque table produira un placement particulier. Les expériences ont montré qu'une différence en performance de 30 % est possible, pour une même description de circuit mais pour les différentes tables de coût. Lorsque l'on souhaite pousser la performance d'un circuit au maximum, alors l'implémentation peut être faite pour un certain nombre de tables de coûts (ou toutes) dans une implémentation dite Multi-Pass Place and Route - MPR. La table de coût offrant la meilleure performance serait alors retenue pour le placement et le routage définitifs.

III.3.4.2 Les contraintes

Les contraintes sur la performance de circuit que l'on souhaite obtenir sont introduites dans le processus d'implémentation à l'aide des applications graphiques Constraint Editor et/ou PACE, ou tout simplement à la main, à l'aide d'un fichier texte contenant les descriptions des contraintes avec une syntaxe particulière. Pour les circuits synchrones, la spécification des contraintes peut être faite de manière globale à l'aide de trois types de contraintes différentes :

- **Période minimale (fréquence maximale) :** Cette contrainte agit sur le délai maximal qui peut y avoir entre tout élément synchrone source et tout élément synchrone destination. C'est donc la fréquence maximale avec laquelle le circuit synchrone peut fonctionner dans le FPGA.
- **Contrainte sur les entrées (Pad to setup time) :** Comme le circuit FPGA reçoit les données du monde extérieur, il faut également tenir compte de délais liés au transit de données depuis la frontière physique du FPGA (les pattes d'entrées) jusqu'au premier élément synchrone. Même si le premier élément synchrone se trouve dans un IOB du FPGA, ces délais sont néanmoins à prendre en compte.

➤ **Contrainte sur les sorties (Clock to Pad) :** Le même raisonnement est d'application pour les sorties du FPGA.

III.3.5 Configuration

Le processus de design flow se termine par la génération du fichier de configuration correspondant au circuit réalisé. Ce fichier peut être directement transféré dans le circuit FPGA cible depuis la plate-forme du développement (en utilisant l'interface JTAG) ou depuis une mémoire programmable (ISP PROM) dans le cas d'une solution embarquée. Le circuit conçu devient alors opérationnel [28].

III.4 Implémentation de l'algorithme MLI on-line

III.4.1 Schéma block

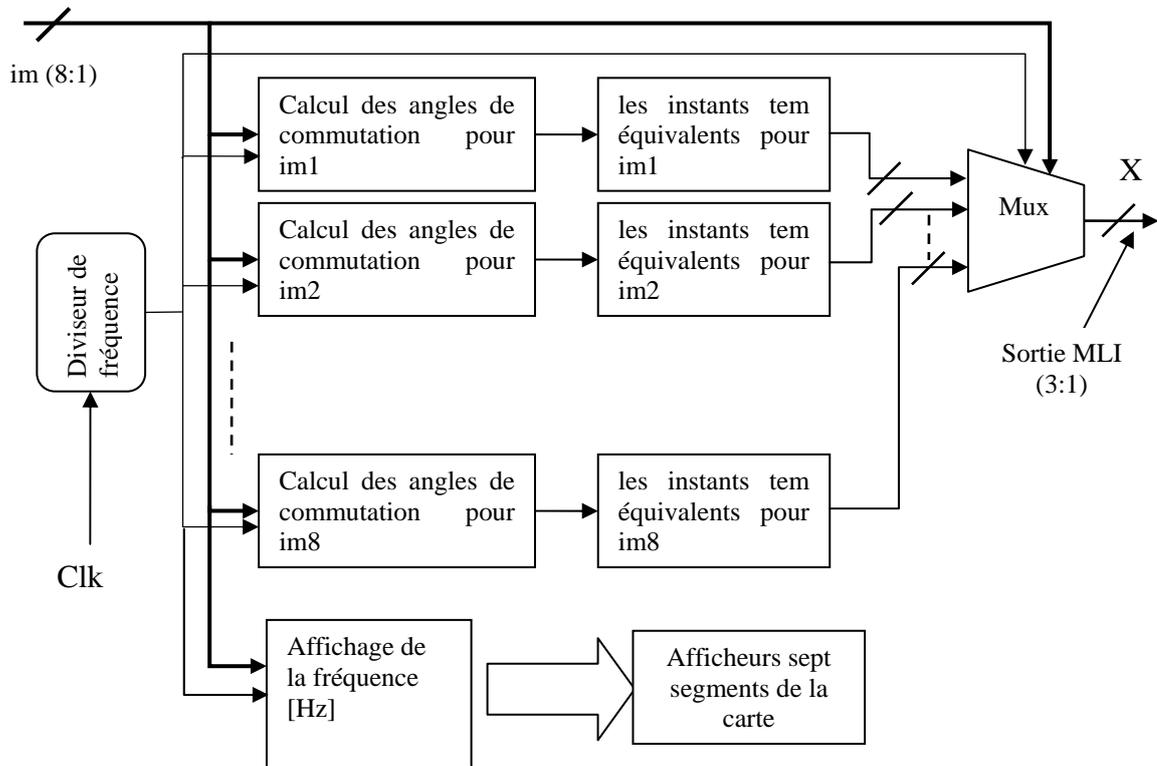


Fig. III.8. Synoptique de l'application MLI on-line proposé

III.4.2 Développement du code VHDL

Nous avons élaboré plusieurs codes VHDL, chacun pour une partie du système.

- Le premier programme est une routine contenant les différents types numériques dont on a besoin, et la définition des opérations nécessaires pour l'implémentation de notre algorithme.
- Le deuxième programme a été consacré au calcul des angles de commutation dans un quart de période (pour im donné) puis les instants temporels équivalents, et la génération du signal principal MLI, enfin la déduction de deux autres signaux de commande dans le cas d'un système triphasé.
- Le troisième programme, c'est l'affichage de la vitesse/fréquence sur les afficheurs disponibles sur la carte.
- Le quatrième programme est un diviseur de fréquence, la carte que nous avons utilisée dispose de deux générateurs d'horloge internes (100 et 24 MHz),

L'outil utilisé pour le développement du code VHDL est Xilinx ISE 7.1.

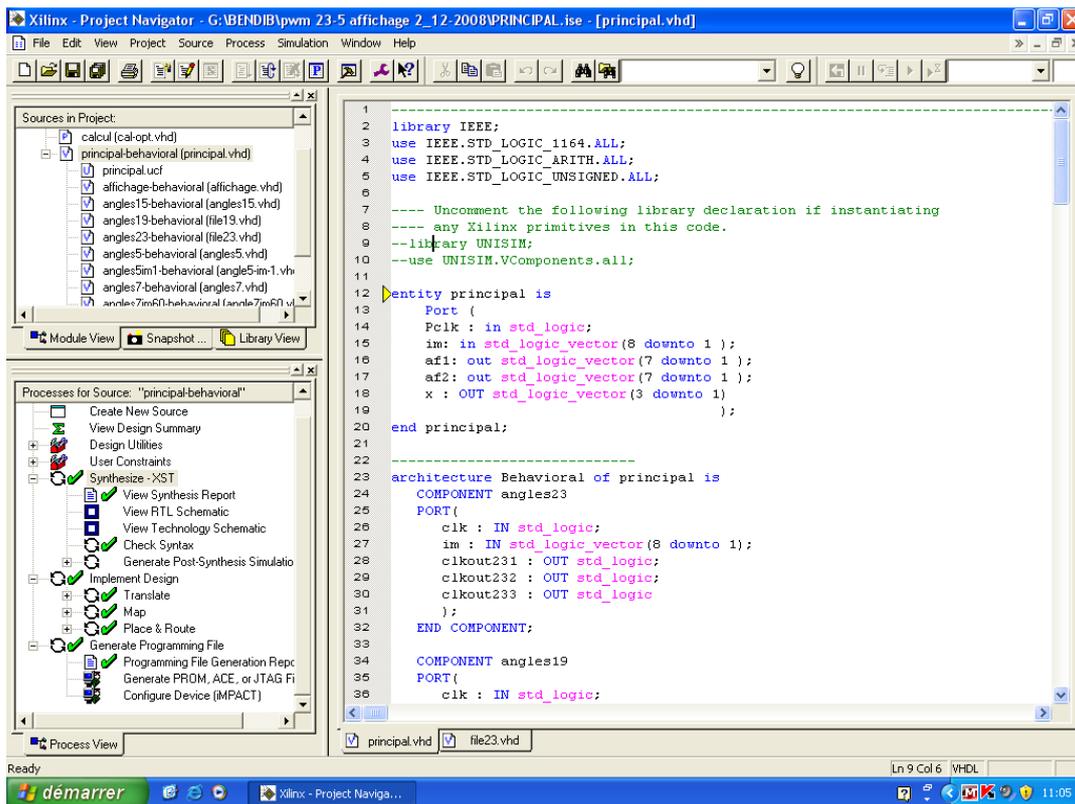


Fig. III.9. Différents programmes VHDL utilisés

III.4.3 Résultats de simulation

La deuxième étape est la simulation fonctionnelle de l'algorithme, l'outil utilisé est MODELSIM ; la figure ci-dessous donne quelques exemples des résultats obtenus.

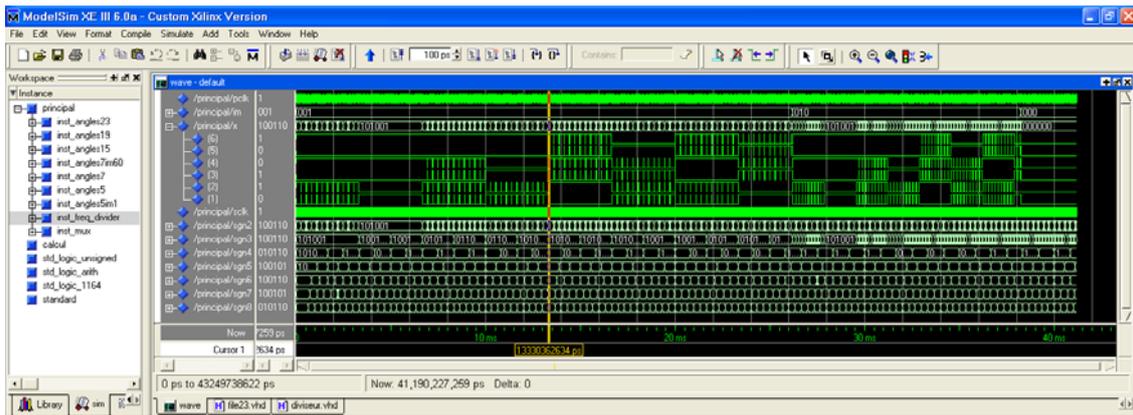


Fig.III.10. Résultats de simulation pour différentes valeurs de im et m

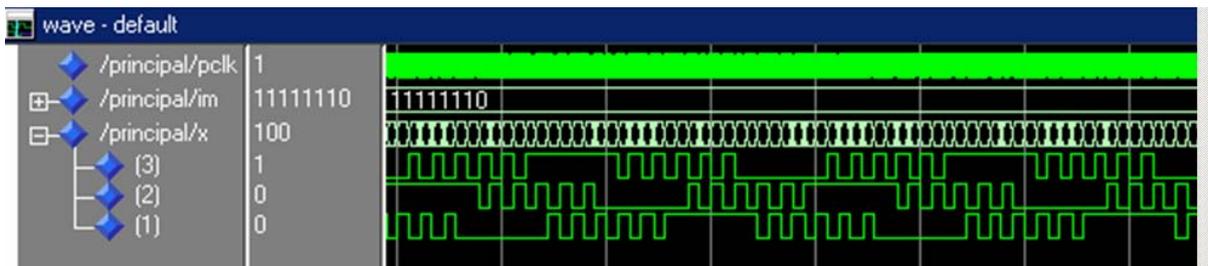


Fig. III.11. Résultats de simulation fonctionnelle pour $im=1$ et $m=5$

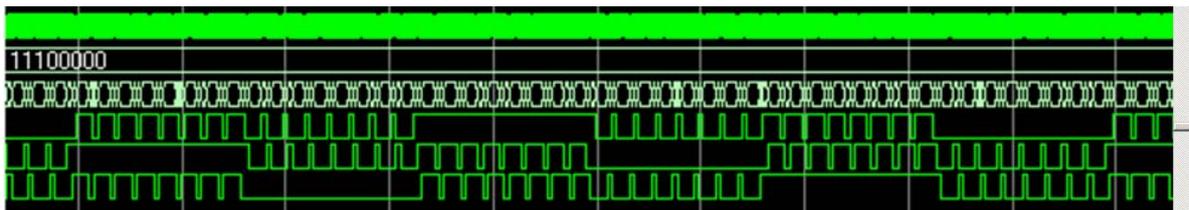


Fig. III.12. Résultats de simulation fonctionnelle pour $im=0.4$ et $m=15$

III.4.4 La synthèse

L'étape la plus critique est la synthèse car le VHDL est standardisé alors que la synthèse ne l'est pas. Par d'autres termes il y'a des instructions VHDL synthétisables d'autres ils ne le sont pas. [27]

- **Les conditions initiales**

Le VHDL supporte les conditions initiales soit pour les signaux ou les variables, mais on ne peut pas les réaliser physiquement.

Dans la pratique les conditions initiales sont synthétisées d'une manière aléatoire donc dans un système pratique il faut ajouter une reset utilisant un pin extérieur. [27] [31]

- **Les types numériques:**

La synthèse est possible seulement pour les nombres ayant un rang fini; un nombre entier dont le rang n'est pas spécifié n'est pas supporté par les softwares de synthèse.

Les vecteurs doivent avoir une taille constante.

Les nombres flottants (réels) en général ne sont pas synthétisables, sauf s'ils disposent des bibliothèques spéciales [27] [32].

• **Les boucles:**

La boucle FOR est un cas particulier du mécanisme général des boucles en VHDL, elle doit avoir un rang constant. Donc une boucle FOR avec un rang variable n'est pas synthétisable [27].

Pour notre algorithme nous avons utilisé le programme XST - Xilinx Synthesis Technology, l'outil View RTL Schematic nous donne le résultat suivant pour le système global (Fig III.11)



Fig. III.13. Résultat de l'outil view RTL schematic pour l'algorithme MLI on-line

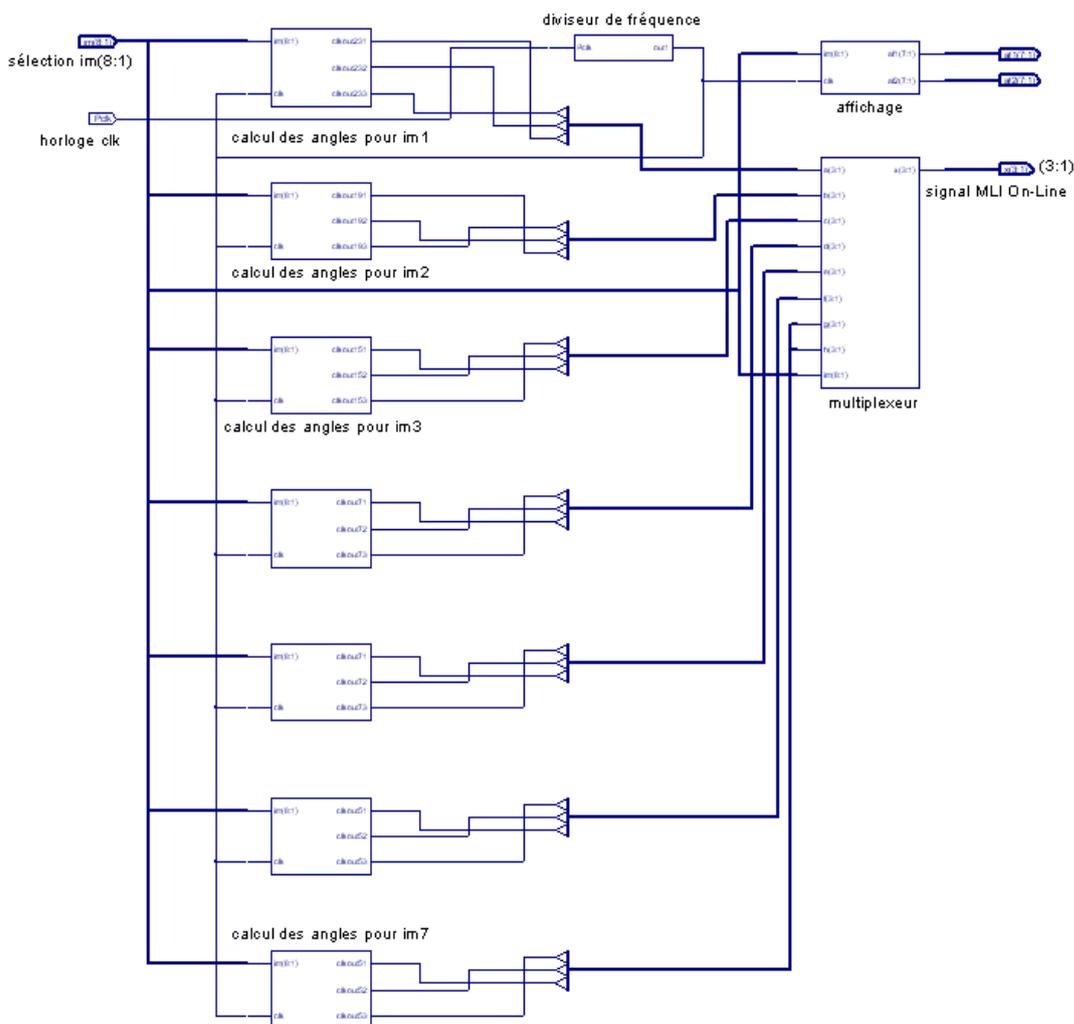


Fig. III.14. Resultat de l'outil view technology schematic pour l'algorithme MLI on-line

Les figures ci-dessus montrent les schémas équivalents générés par l'outil de synthèse pour chaque partie de notre programme ; le diviseur de fréquence, le multiplexeur, le bloc d'affichage.

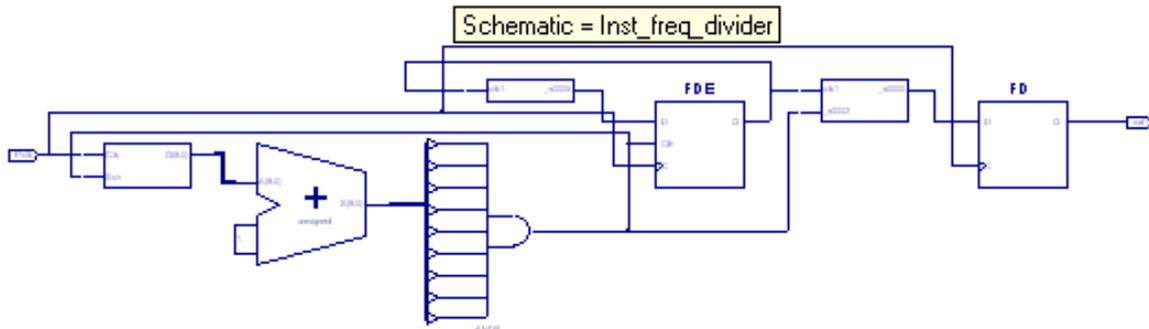


Fig III.15. Diviseur de fréquence

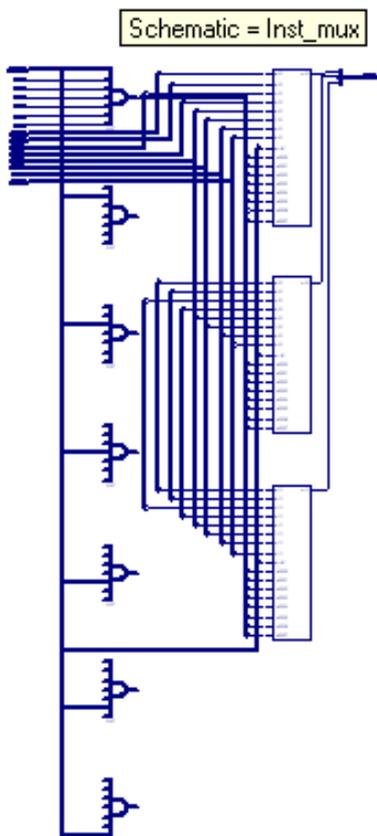


Fig. II.16. Multiplexeur

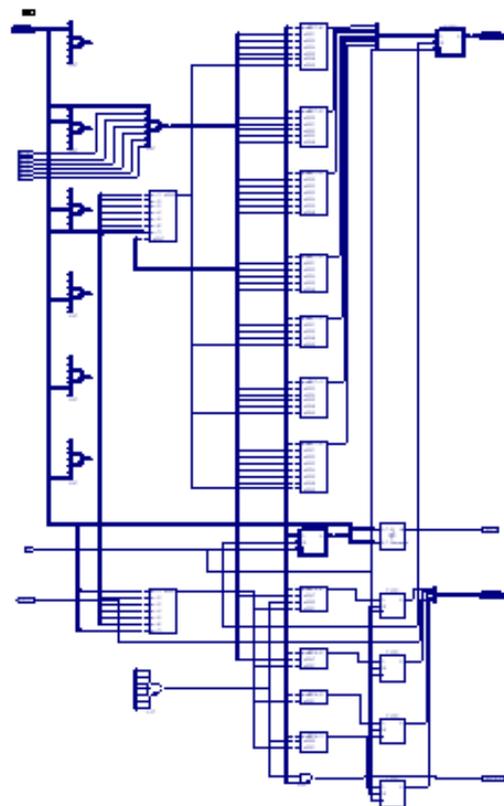


Fig. III.17. Affichage

III.4.5 Phase d'implémentation (placement routage)

Avant l'implémentation du programme dans la carte cible, il faut choisir les contraintes, la figure III.18 montre l'outil 'Assigne Package Pins'. La figure III.19 montre la quantité des ressources utilisées dans la carte par notre programme .

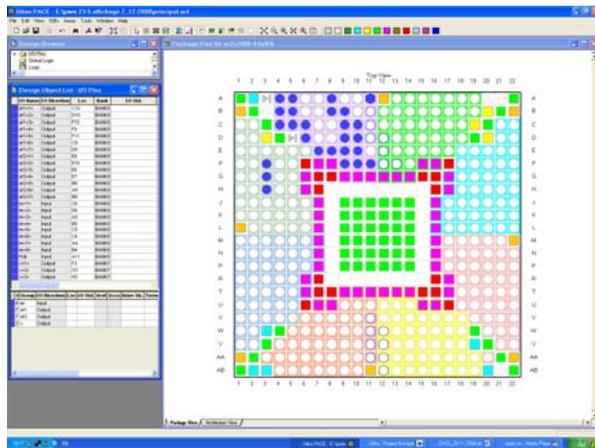


Fig. III.18. Choix des contraintes

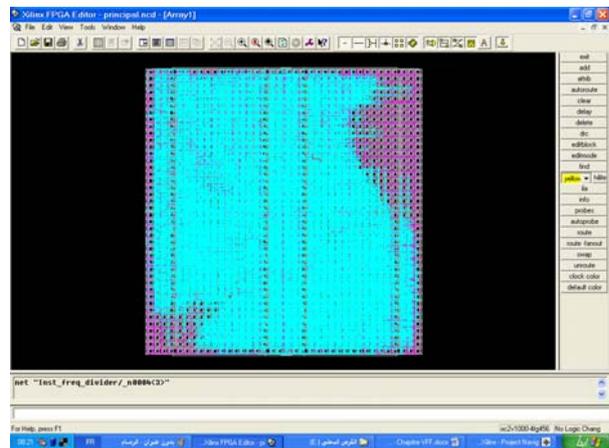


Figure II.19. Ressources utilisées

III.4.6 Génération du fichier .bit chargement et vérification

Le tableau suivant donne la quantité et le pourcentage des ressources utilisées dans l'implémentation de la commande MLI on-line

Ressource	utilisé	Disponible	Taux
Slice Flip Flops:	412	10,240	4%
4 entrées LUTs:	8,058	10,240	78%
Distribution Logique:			
Slices occupées:	4,418	5,120	86%
Slices contenant logique dépendant	4,418	4,418	100%
Slices contenant seulement logique indépendant	0	4,418	0%
Nombre total 4 entrées LUTs:	8,375	10,240	81%
Nombre des IOBs	26	324	8%
Nombre des GCLKs:	2	16	12%

Tableau III.3. Ressources utilisées

III.4.7. Test des signaux de sortie

Nous avons exploité les broches du port LVDS pour sortir nos trois signaux MLI on-line, se sont les pins F3, G3 et H3. L'horloge utilisée est une horloge de 24 Mhz qui est disponible sur la carte (pin A11). La sélection de im (vitesse ou fréquence) est faite par les dip switch. La visualisation des signaux MLI se fait par un oscilloscope numérique. La figure III.20 donne le schéma de la carte utilisée. Les figures III.21 à III.23 montrent quelques exemples des signaux MLI on-line générés.

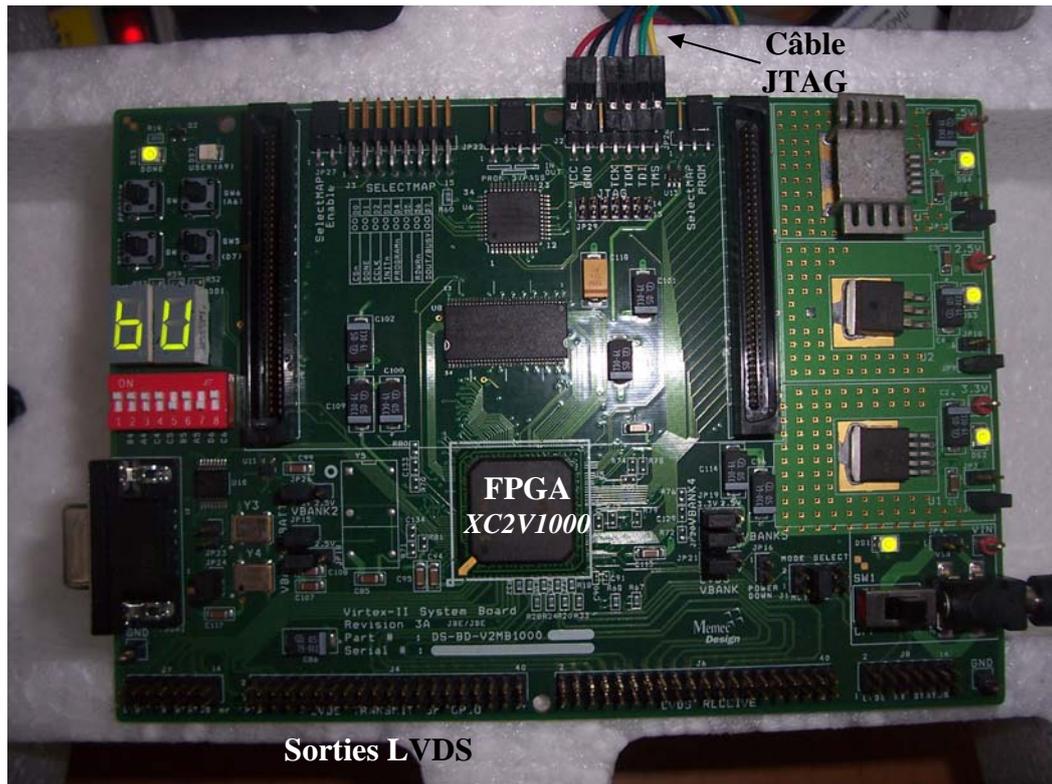


Fig. III.20. Carte de développement utilisée (V2MB1000)

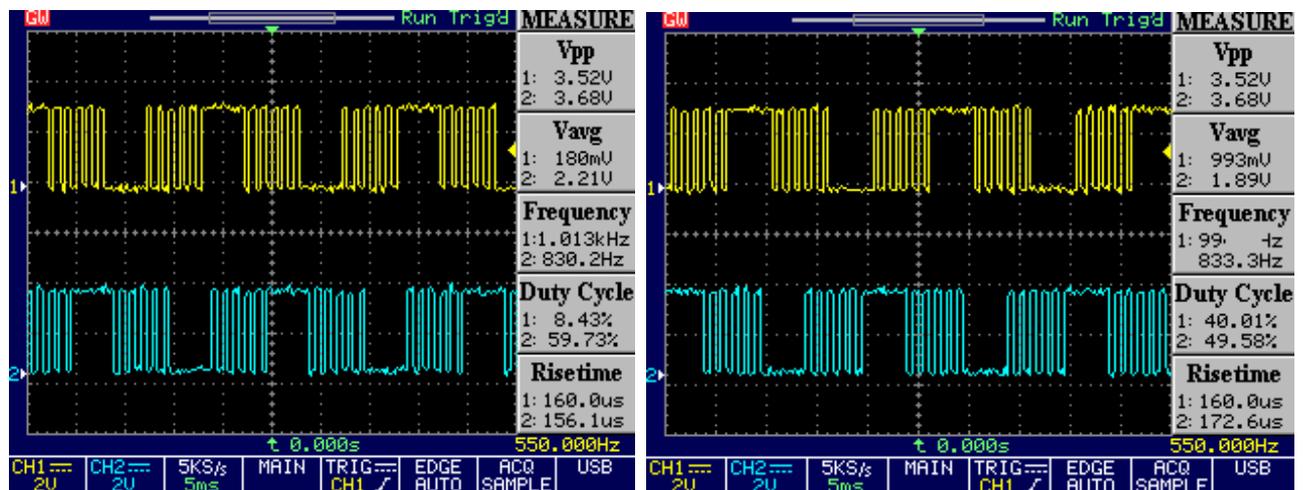
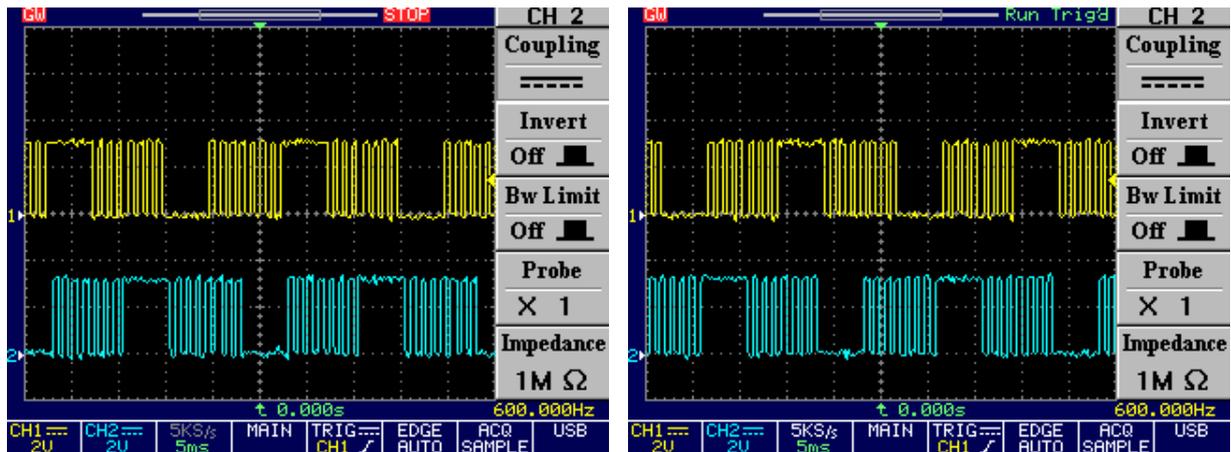
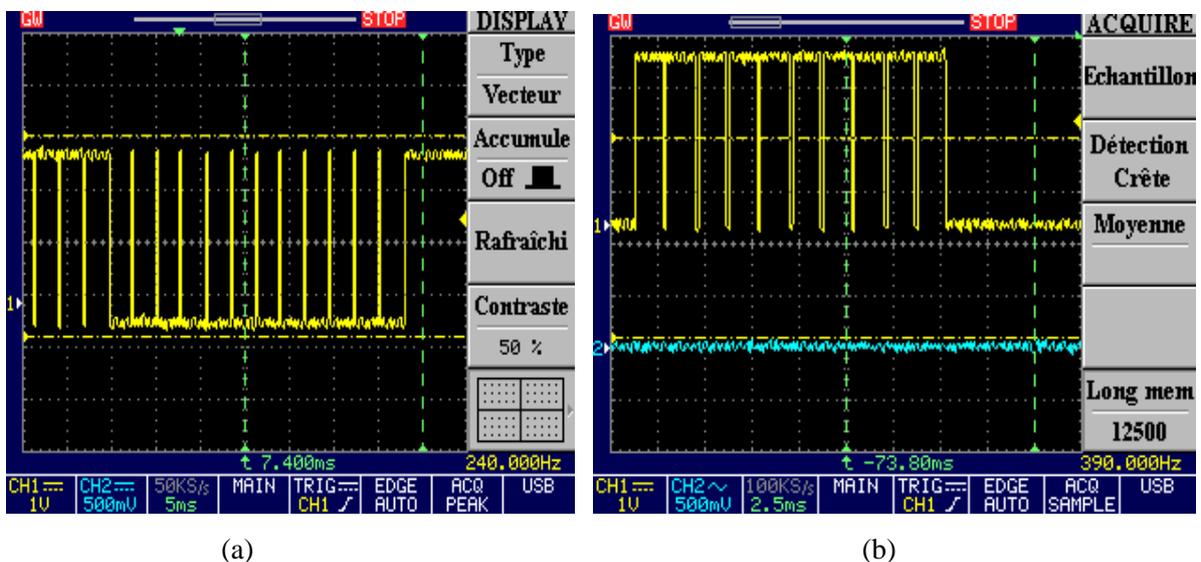


Fig. III.21. Signaux MLI pour im=1 et m=5

Fig. III.22. Signaux MLI pour $i_m=0.8$ et $m=7$ Fig. III.23. Signal généré pour : (a) $m=23$ et $i_m=0.1$; (b) : $m=19$ et $i_m=0.2$

Conclusion

L'algorithme MLI implémenté permet d'éliminer les harmoniques et d'asservir la valeur du fondamental. Les signaux obtenus sont très proches de ceux de la simulation. Il est clair que chaque quart de période contient m angles de commutation. Le résultat de synthèse nous a donné des schémas compliqués qui sont, sans doute irréalisables en analogique.

Un problème délicat que nous avons rencontré, est le passage du VHDL à la synthèse. En d'autres termes travailler avec un VHDL synthétisable va diminuer le nombre d'instructions et rend l'intervalle de travail étroit. Une de ces limitations est les nombres flottants et les boucles à rang variable.

Après l'implémentation, il paraît que la quantité de ressources est un peu élevée (86 % de tranches sont occupées), car la taille du circuit utilisée est un peu faible par rapport à ce qui est disponible sur le marché. Afin de valider expérimentalement cet algorithme nous allons l'utiliser dans un variateur de vitesse à base d'un onduleur triphasé. C'est l'objectif du chapitre suivant.

CHAPITRE IV

REALISATION D'UN VARIATEUR DE VITESSE POUR MOTEUR ASYNCHRONE TRIPHASE

Ce chapitre concerne la conception et réalisation d'un variateur de vitesse pour moteur asynchrone triphasé, à base d'un onduleur de tension, piloté par une commande MLI on-line. Après la description détaillée de l'algorithme et la simulation du fonctionnement du système à réaliser, nous avons implémenté cet algorithme sur un circuit FPGA Virtex de la famille XILINX en utilisant la carte de développement Memec Design V2MB1000.

L'étape suivante est la réalisation de l'application globale y compris la carte de commande, circuits de conditionnement et l'étage de puissance. La dernière partie est consacrée aux relevés des résultats expérimentaux et leur interprétation.

IV.1 Description générale de l'application

L'application comporte principalement trois parties, une partie commande un étage de puissance et une charge.

Etage de commande: C'est la partie responsable de la génération des signaux de commande (commande éloignée); nous avons utilisé le composant FPGA Virtex II de la compagnie XILINX avec la carte de développement Memec Design V2MB1000. Ces signaux sont ensuite conditionnés et isolés à l'aide des optocoupleurs. Pour la génération de la commande rapprochée nous avons utilisé le driver IR2111.

Etage de puissance: Cette partie contient une alimentation continue issue du secteur, redressée et filtrée. Des transistors MOSFET et des diodes rapides formant un onduleur triphasé en pont à deux niveaux.

Charge: La charge utilisée est un moteur asynchrone triphasé.

La stratégie de commande utilisée est de faire varier la vitesse du moteur, en utilisant une loi de commande MLI avec un rapport V/f constant. Nous commençons par des petites vitesses –petites fréquences- on prend $m=23$, puis on augmente la fréquence et par conséquent la vitesse jusqu'à la vitesse nominale correspondante à la fréquence nominale 50Hz, $m=5$.

La figure IV.1 donne le schéma synoptique du variateur de vitesse réalisé.

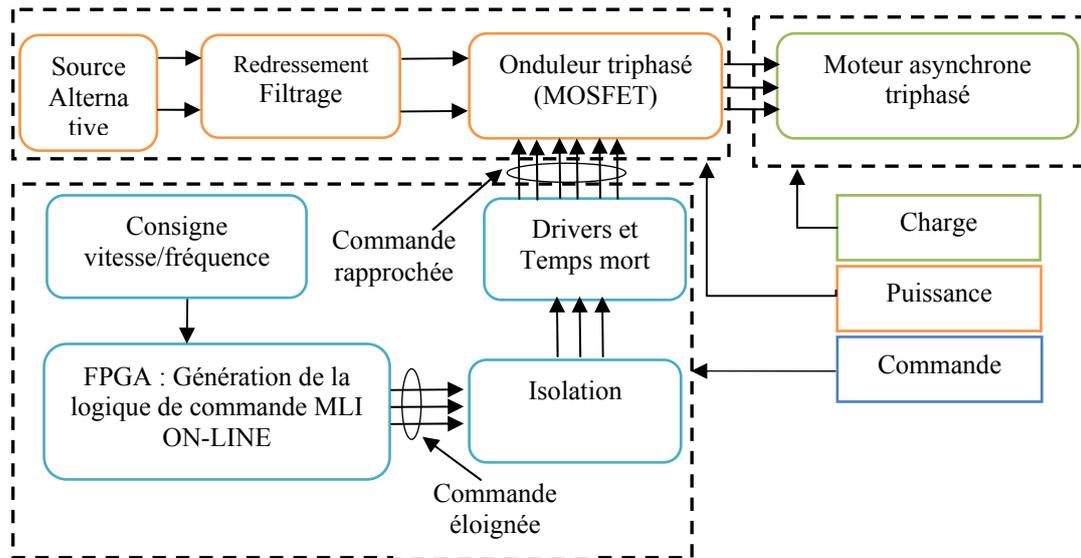


Fig. IV.1. Schéma synoptique du variateur de vitesse réalisé

IV.2 Commande des transistors MOSFET

IV.2.1 Généralités sur les transistors MOSFET

Le transistor MOSFET est un composant 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 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 [40]. La figure IV.2 donne la place du MOSFET et son domaine d'utilisation.

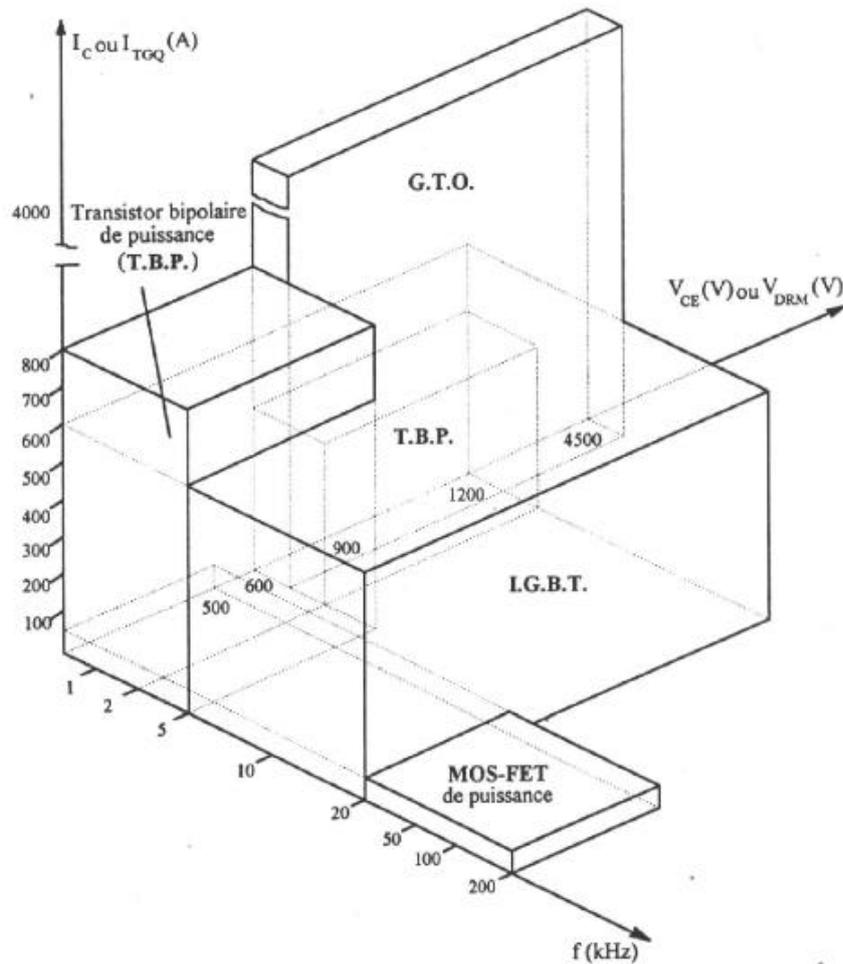


Fig. IV.2. Domaines d'utilisation du MOSFET

IV.2.2 Commande des MOSFET low side

Les MOSFET sont beaucoup plus faciles à commander que les transistors bipolaires et sont très bien adaptés pour être contrôlés par des portes logiques.

Il y a lieu simplement de tenir compte de quelques règles pour assurer la fiabilité du composant et minimiser sa dissipation.

La figure IV.3 rappelle les techniques utilisées lorsqu'on a une charge inductive :

- La diode de roue libre (qui peut être une simple 1N4001-4005) mais qui devra être du type Schottky si le MOS devra être coupé rapidement.
- La diode zener qui elle, répond en quelques picosecondes et protège bien le MOS contre les surtensions transitoires. Ici aussi, la résistance série est optionnelle.
- Le snubber RC est souvent utilisé pour limiter les pics causés par les inductances de fuite et les inductances parasites de câblage. Les valeurs des composants sont déterminées de façon empirique.

La relation $\frac{1}{2}LI^2 = \frac{1}{2}CV^2$ conduit à $V = I\sqrt{\frac{L}{C}}$

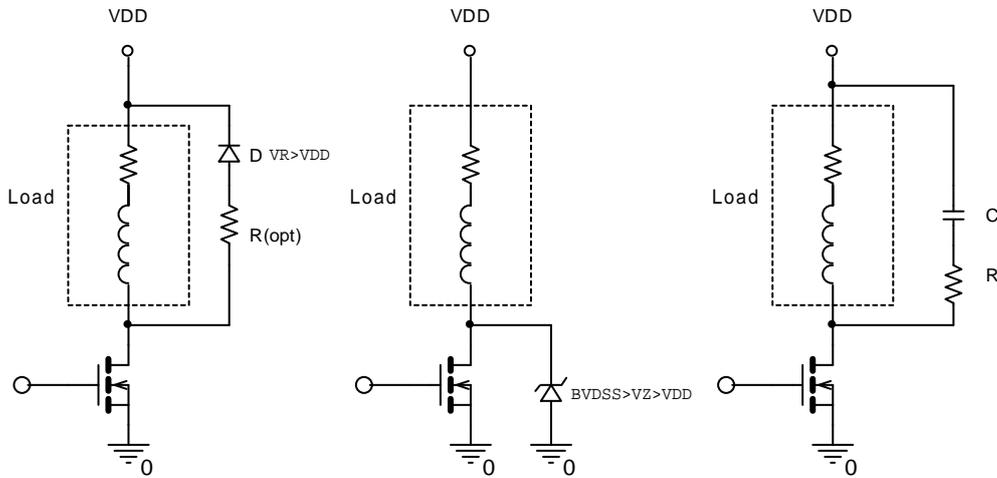


Fig. IV.3. Méthodes de limitation des pics de tension

La commande à partir des portes logiques est très simple. La figure IV.4 montre l'utilisation de portes à collecteur ouvert pour appliquer une tension supérieure à 5V à la grille. Pour accélérer la commutation et réduire la dissipation, on utilise aussi le driver « Totem Pole » qui charge plus vite le condensateur C_{iss} [42].

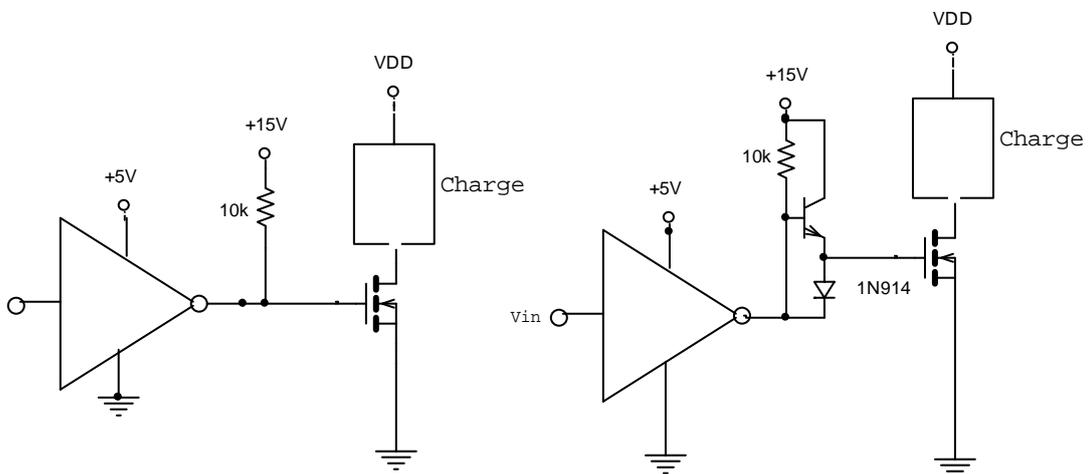


Fig. IV.4. Commande de MOSFET par portes logiques

IV.2.3 Transistor high side

Afin de maintenir le transistor high side enclenché, il faut apporter à sa grille un potentiel supérieur à celui de la source, qui n'est pas reliée à la masse comme il est montré par la figure IV.5.

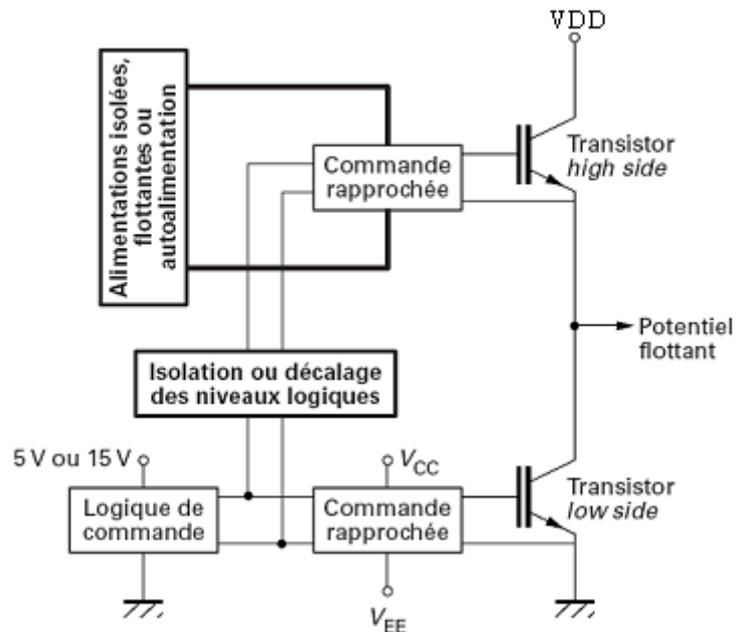


Fig. IV.5. Principe de commande d'un bras de pont

Dans de nombreux cas, une isolation entre les circuits logiques de commande et le circuit de commande rapprochée est nécessaire. Notamment, la commande d'un bras d'onduleur soulève le problème de la commande de l'interrupteur high side, la source de ce dernier se trouvant à un potentiel flottant qui varie entre 0 et la tension du bus continu. Il faut donc créer une alimentation du circuit de commande high side pouvant suivre ces variations de potentiel très rapides, et également isoler ou décaler le potentiel des ordres logiques de commande au potentiel du circuit de commande rapprochée de l'interrupteur high side [34] [40].

IV.2.3.1. Transformateur d'impulsion

Le transformateur d'impulsion peut permettre le décalage des niveaux de la partie logique vers le circuit de commande rapprochée de l'interrupteur high side. Cette solution simple permet des temps de transit extrêmement brefs, mais ne peut être intégrée économiquement dans des circuits intégrés spécialisés, exception faite de circuits hybrides de commande, dédiés aux commandes de transistors de forte puissance.

Toutefois, certains fabricants proposent notamment des circuits intégrés de commande dédiés à la commande de bras d'onduleur avec isolation des impulsions de commande par transformateurs d'impulsion spécifiés [34].

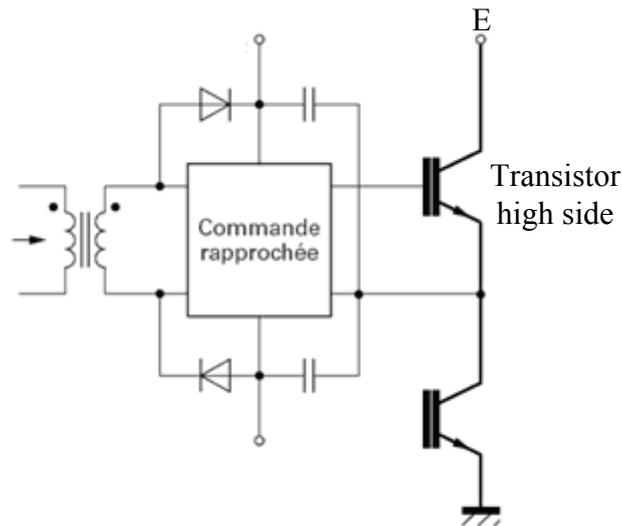


Fig. IV.6. Utilisation du transformateur d'impulsions pour la commande des MOSFET

Cette technique est simple et à faible coût, mais elle est limitée par la durée d'impulsion. Pour utiliser des rapports cycliques élevés, ainsi que pour commander des MOSFET, la circuiterie devient plus complexe. De plus, la taille du transformateur augmente avec la diminution de la fréquence de découpage et les perturbations générées limitent la fréquence maximale de fonctionnement [43].

IV.2.3.2 Isolation par optocoupleur

L'isolation optique permet d'isoler la logique de commande, et ainsi autoriser le décalage de niveau des impulsions pour la commande des interrupteurs high side. Certains optocoupleurs dédiés aux applications de commande de composants à semi-conducteurs de puissance, sont équipés d'écrans électrostatiques qui leur confèrent une grande immunité aux perturbations du mode commun (réduction de la capacité parasite entre l'entrée et la sortie). La plupart des dispositifs optoélectroniques disponibles sont capables de supporter des dV/dt de l'ordre de $15 \text{ kV}/\mu\text{s}$, et ont des tensions d'isolation supérieures à 1500 V . Les temps de propagation vont de quelques 100 ns , pour les composants les plus rapides jusqu'à $1 \mu\text{s}$ environ.

Les optocoupleurs transmettent l'information de commande et non l'énergie. L'énergie de commande est prélevée sur des alimentations auxiliaires qui doivent être isolées ou recrées à partir de la tension de collecteur ou de l'alimentation de puissance. La figure IV.7 montre le schéma de principe d'un circuit de commande isolé à partir d'un optocoupleur.

Enfin, de nombreux circuits à base d'optocoupleurs intègrent des fonctions particulières d'amplification ou de protection. L'amplification intégrée permet à un optocoupleur de

commander directement le composant à grille isolée sans circuit d'amplification supplémentaire [39]. On peut noter l'existence de circuits dédiés à la commande de bras intégrant dans un même boîtier deux circuits optocoupleurs et les amplificateurs associés, tel le circuit HCPL314J (Agilent).

La protection contre les courts-circuits peut être également intégrée, avec une détection généralement effectuée par mesure de désaturation.

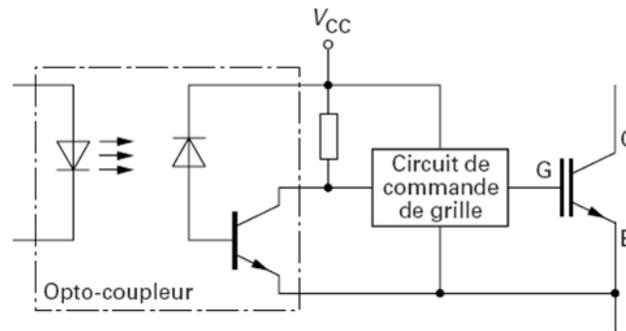


Fig. IV.7. Utilisation d'un optocoupleur pour la commande d'un transistor IGBT

IV.2.3.3 Alimentations isolées

Les alimentations du transistor high side peuvent être isolées par un transformateur (alimentations à découpage). Cette solution permet d'obtenir une grande immunité aux perturbations à conditions que les transformateurs utilisés aient été dimensionnés et conçus dans cet objectif. Il existe des alimentations dédiées aux commandes isolées de transistors MOSFET ou IGBT. Le schéma de principe est donné par la figure IV.8.

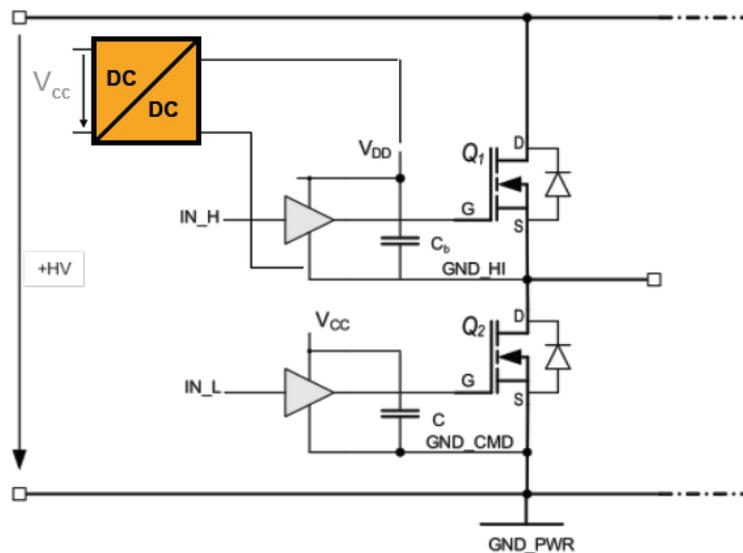


Fig. IV.8. Commande d'un MOFET par alimentation isolée

Cette solution consiste à avoir une alimentation flottante, donc galvaniquement découplée entre VDD et GND_HI.

Un des grands avantages de cette technique est la liberté totale de commande en durée et fréquence. Par contre elle est coûteuse, car chaque MOSFET high-side demande une alimentation flottante.

La commande doit être translatée d'où la nécessité d'utiliser des transformateurs ou des optocoupleurs.

IV.2.3.4 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 [39]. Le principe (Figure IV.9) consiste à profiter de la conduction de l'interrupteur low side pour charger une capacité de bootstrap C_B sous la tension V_{CC} à travers la diode D_B . Lorsque l'interrupteur low side se bloque et que la tension réapparaît à ses bornes, la diode D_B se bloque à son tour, et elle doit donc être capable de supporter la haute tension E . La charge alors portée par C_B 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.

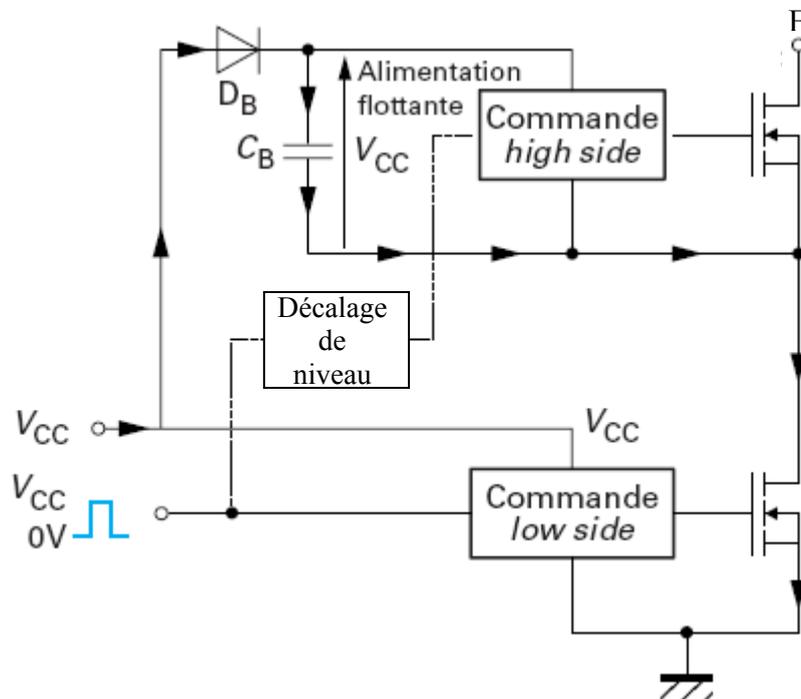


Fig. IV.9. Principe de la technique de bootstrap

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.

Pour notre réalisation, afin d'accomplir l'isolation nous avons choisi d'utiliser des optocoupleurs rapides et de faible courant d'entrée HCPL2200.

Le driver IR2111 de IR a été choisi pour la technique de bootstrap. Les paragraphes qui suivent donneront le détail de chaque étage avec les résultats expérimentaux.

IV.3 Conception et réalisation de la carte de commande

IV.3.1 Etage d'isolation

Les signaux délivrés par notre carte FPGA sont de types LVDS, sont montrés sur la figure ci-dessous.

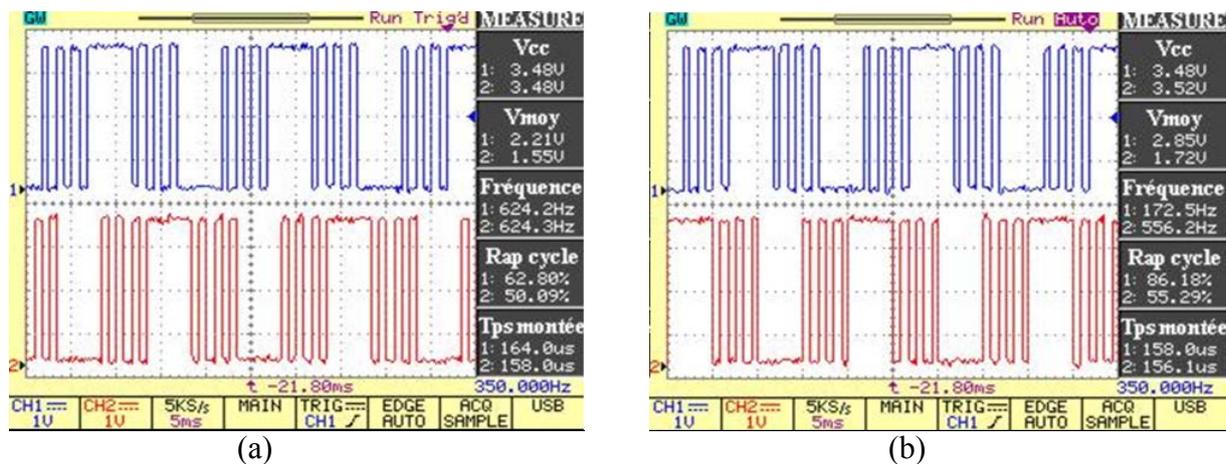


Fig. IV.10. Signaux de commande délivrés par le circuit FPGA

(a) : CH1 représente le signal de commande attaquant la première phase g1, CH2 la 2^{ème} phase.
 (b) : CH1 représente le signal de commande attaquant la première phase g1, CH2 la 3^{ème} phase.

Afin de pouvoir utiliser ces signaux, qui sont de faible amplitude, et de protéger la carte on est obligé de concevoir un étage d'isolation entre cette partie et l'étage de puissance. La solution est d'utiliser des optocoupleurs, qui permettent la transmission optique de la commande éloignée. La particularité ici est de trouver des optocoupleurs faible courant et compatibles avec les signaux LVDS, à condition qu'ils puissent délivrer un courant suffisant aux drivers.

Le circuit utilisé pour notre cas est le HCPL2200 de la société HEWLETT PACKARD, ce circuit a les caractéristiques suivantes:

Compatible avec les signaux LSTTL, TTL, et CMOS Logique.

Un faible courant d'entrée (1.6 mA).

Temps de montée (rise time) $t_r=55ns$

Temps de décroissance (fall time) $t_f=15 ns$

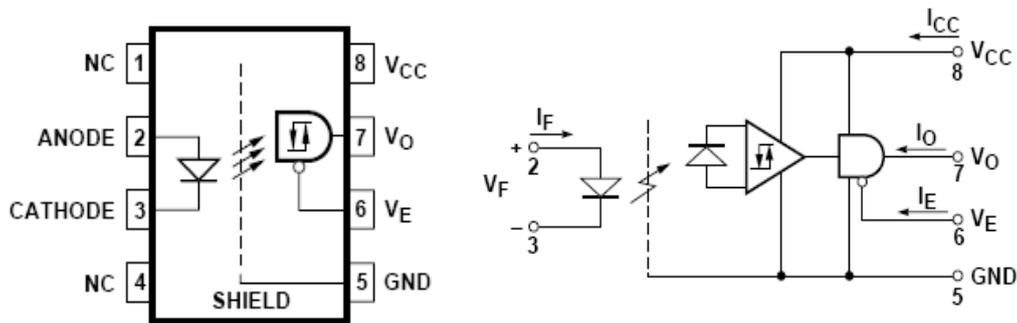


Fig. IV.11. L'optocoupleur HCPL2200 et son schéma interne.

Une résistance de 470Ω est ajoutée à l'entrée 2 de l'optocoupleur afin de limiter le courant d'entrée et de le protéger. D'après le constructeur un condensateur de $0.1\mu F$ doit être placé aux bornes de l'alimentation de ce dernier (entre les pins 5 et 8) [35]. Afin de tester le bon fonctionnement nous avons réalisé le circuit de la figure IV.12, les résultats sont visualisés sur oscilloscope numérique, le résultat pour $i_m=0.8$ et $m=7$ est donné par la figure IV.13.

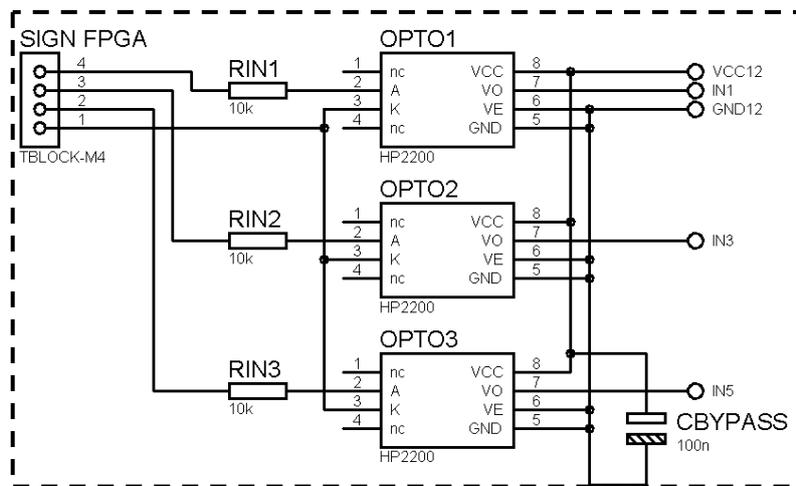


Fig. IV.12. Circuit d'isolation

Sortie de l'optocoupleur

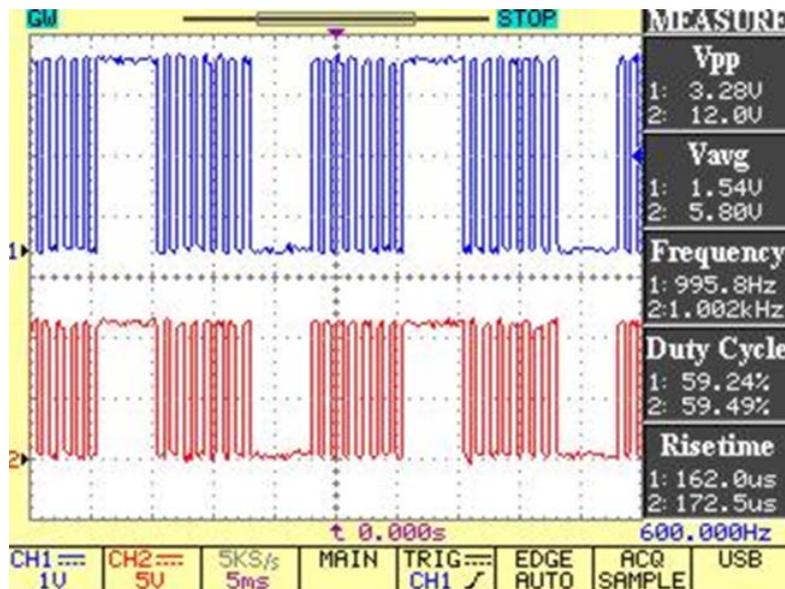


Fig. IV.13. Sortie du FPGA (CH2) et la sortie de l'optocoupleur (CH1)

IV.3.2 Le driver IR2111

La technique de bootstrap est utilisée pour commander les transistors high side, pour cela nous avons utilisé la solution intégrée de International Rectifieir.

Le composant choisi est le driver IR2111, qui permet de commander un demi-pont, avec un temps mort intégré de 650 ns entre la sortie HO et la sortie LO, le schéma de connexion est donné par la figure IV.14. La figure IV.5 donne son schéma interne.

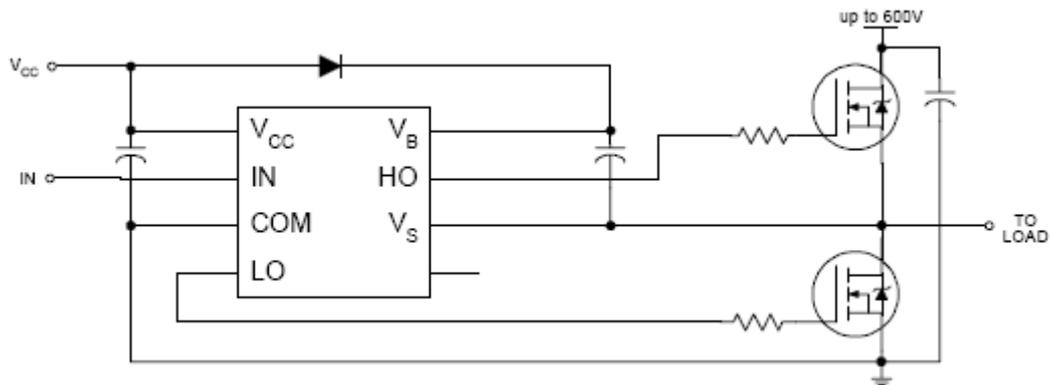


Fig. IV.14. Connexion typique du circuit IR2111

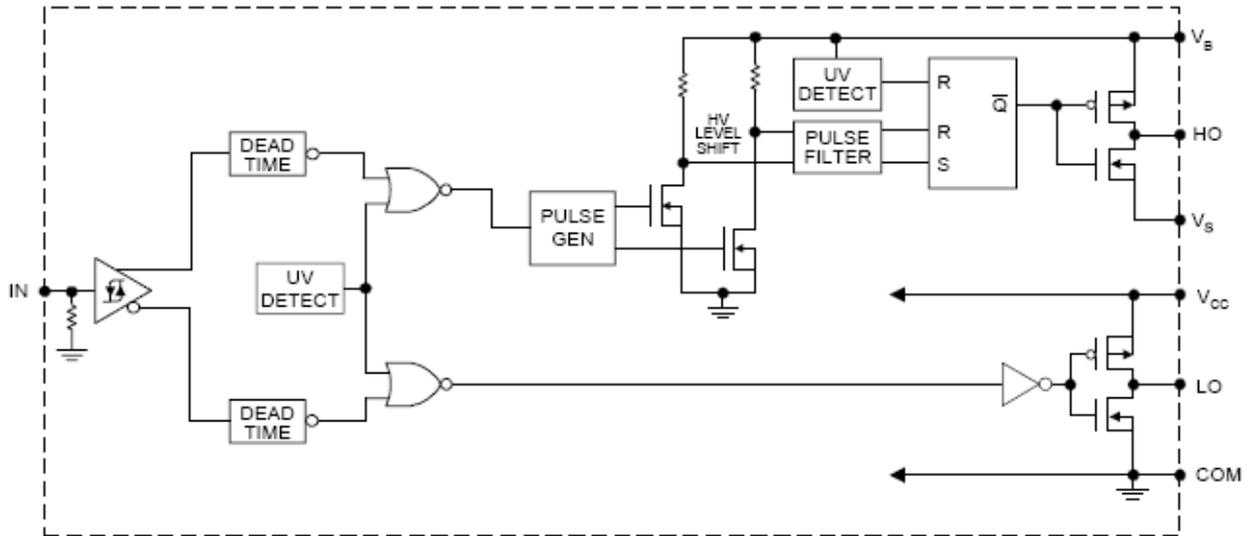


Fig. IV.15. Schéma interne du driver IR2111

IV.3.3 Sélection des composants bootstrap

Seulement la diode D_B et le condensateur C_B sont les deux composants externes nécessaires pour le fonctionnement du driver avec une commande MLI standard [33] [41].

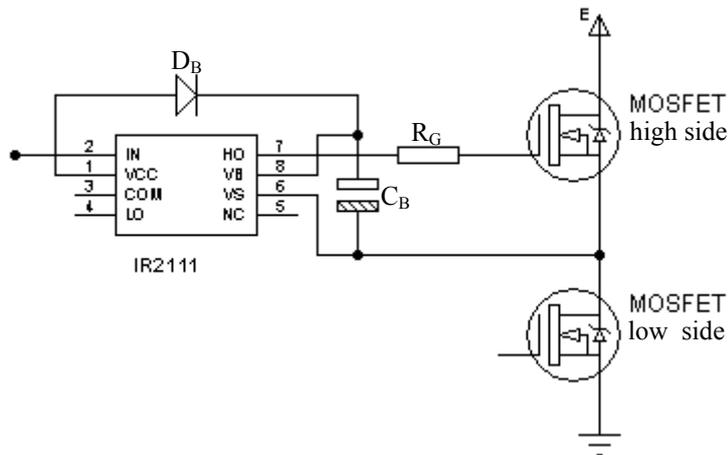


Fig. IV.16. Composants de la technique de bootstrap

IV.3.2.1 La diode de bootstrap D_B

La tension maximale aux bornes de la diode est celle de l'alimentation de l'onduleur, donc elle doit supporter une tension supérieure à celle du bus continu (E). De plus elle doit être rapide (de type Schottky) [41].

IV.3.2.2 Choix du condensateur de bootstrap C_B

La tension aux bornes du condensateur de bootstrap est celle de l'alimentation des drivers, sa valeur est définie en prenant en considération les contraintes suivantes [41] [33]:

- La tension nécessaire à la grille pour mettre en conduction le transistor high side.
- Le courant I_{QBS} , qui est le courant de repos de la partie supérieure du driver IR2111.
- Le courant interne du translateur de niveau dans le circuit intégré.
- Courant de fuite Grille –Source.
- Courant de fuite dans le condensateur.

Ce dernier facteur est suivi seulement sur le condensateur utilisé de type électrolyte.

La valeur minimale du condensateur s'exprime de la manière suivante:

$$C_B \geq \frac{2 \left[2Q_g + \frac{I_{qb(max)}}{f} + Q_{ls} + \frac{I_{c_{bs(leak)}}}{f} \right]}{V_{CC} - V_f - V_{LS} - V_{Min}}$$

Q_g : Charge de la grille du MOSFET de la partie haut (high side)

f: Fréquence de fonctionnement.

$I_{c_{bs(leak)}}$: Courant de fuite dans le condensateur C_B .

$I_{qbs(max)}$: Courant maximal de repos.

V_{CC} : Alimentation du driver.

V_f : Tension direct aux bornes de la diode de bootstrap D_B .

V_{LS} : La tension aux borne du MOSFET de la partie bas (low side)

V_{MIN} : La tension minimale entre V_b et V_s .

Q_{LS} : La charge nécessaire par cycle pour le décalage du niveau (typiquement 5nC pour 500V/600V et 20 nCb pour 1200V).

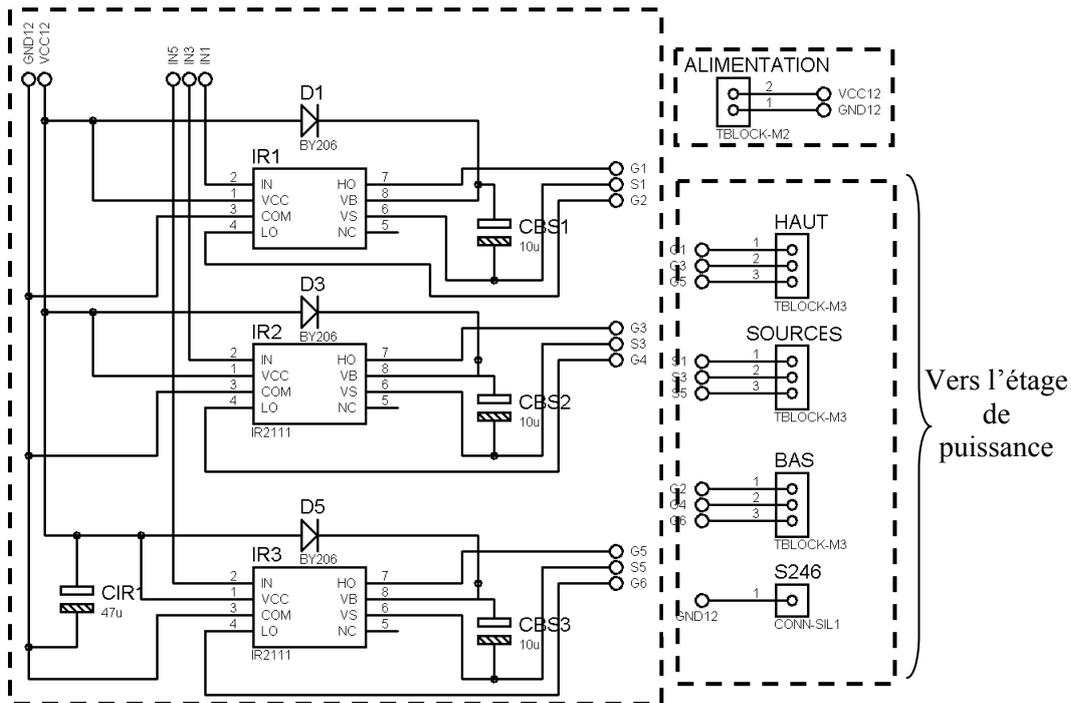


Fig. IV.17 Circuit des drivers IR2111

La réalisation du circuit de la figure IV.17 et après son test on a obtenu les résultats suivants:

Pour $m=1$ on a choisi $m=5$, la figure ci-dessous donne le signal de commande éloignée (généré par le circuit FPGA) sur CH1, et le signal de commande rapprochée qui va attaquer la grille du transistor low side.

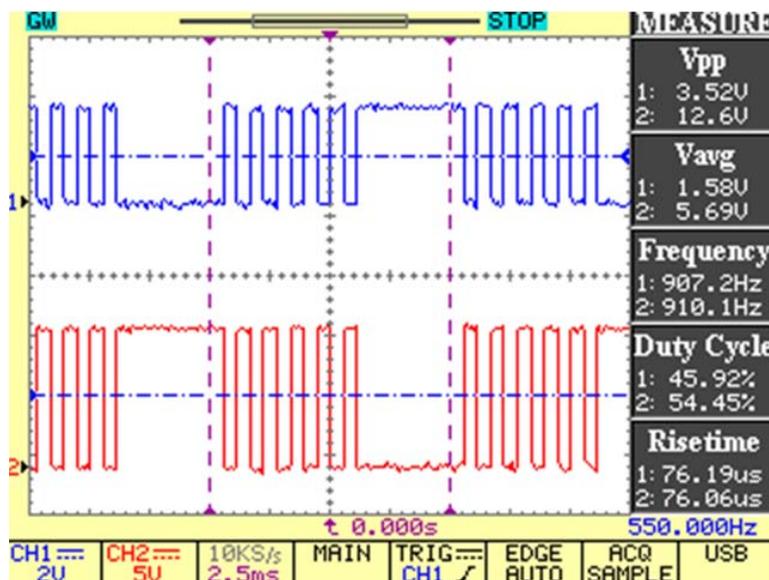


Fig. IV.18. CH1: Signal FPGA; CH2: Sortie LO du driver IR2111

La figure IV.19 donne les mêmes signaux, en mettant $i_m=0.2$ avec $m=19$ angles de commutation par quart de période (la période est de 100ms).

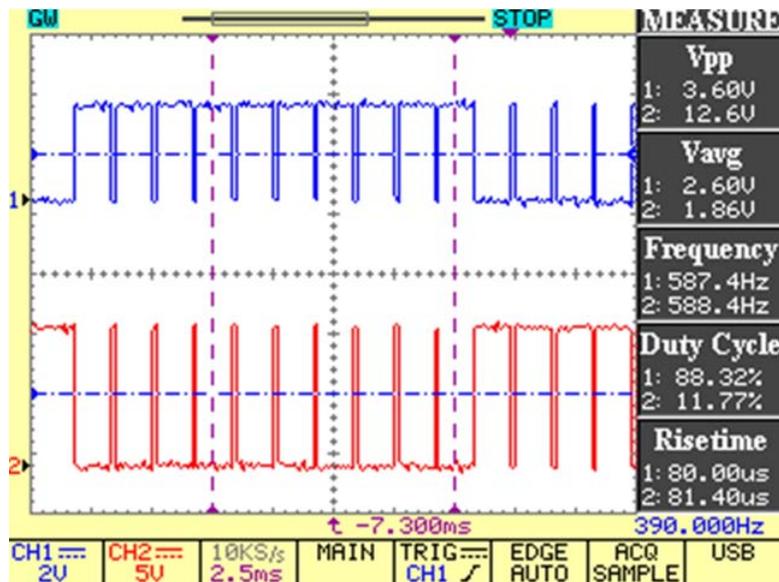


Fig. IV.19. CH1: Signal FPGA, CH2: Sortie LO du driver IR2111

IV.4 Conception et réalisation de l'étage de puissance

L'étage de puissance est formé de trois bras, chacun comporte deux interrupteurs comme on le montre sur la figure IV.21. Les interrupteurs utilisés dans cette réalisation sont de type MOSFET IRFP450 placés sur un dissipateur de chaleur surdimensionné pour dégager la chaleur.

IV.4.1 Le MOSFET IRFP450

Les principales caractéristiques sans les suivantes:

$$V_{DSS}=500V, R_{DS(on)}=0.33 \Omega, I_D=14 A.$$

Pour la commutation à l'ouverture le IRFP450 possède un temps de retard à la croissance t_d (dealy time) de 24 ns et un temps de montée (rise time) t_r de 14 ns, donc le temps d'ouverture t_{on} est de l'ordre de 38ns.

Pour la fermeture :

$t_{r(voff)}= 15ns$, temps de décroissance (fall time) $t_f=25ns$, temps de stockage t_c (cross-over time) est de 35ns, donc le temps de coupure (turn-off time) t_{off} est de l'ordre de 75ns.

IV.4.2 Choix de la résistance Rg

Les temps de commutation sont fortement dépendants du courant I_D et de la résistance R_G placée en série avec la grille. Les fabricants les plus consciencieux donnent des courbes tenant compte de ces deux paramètres [43].

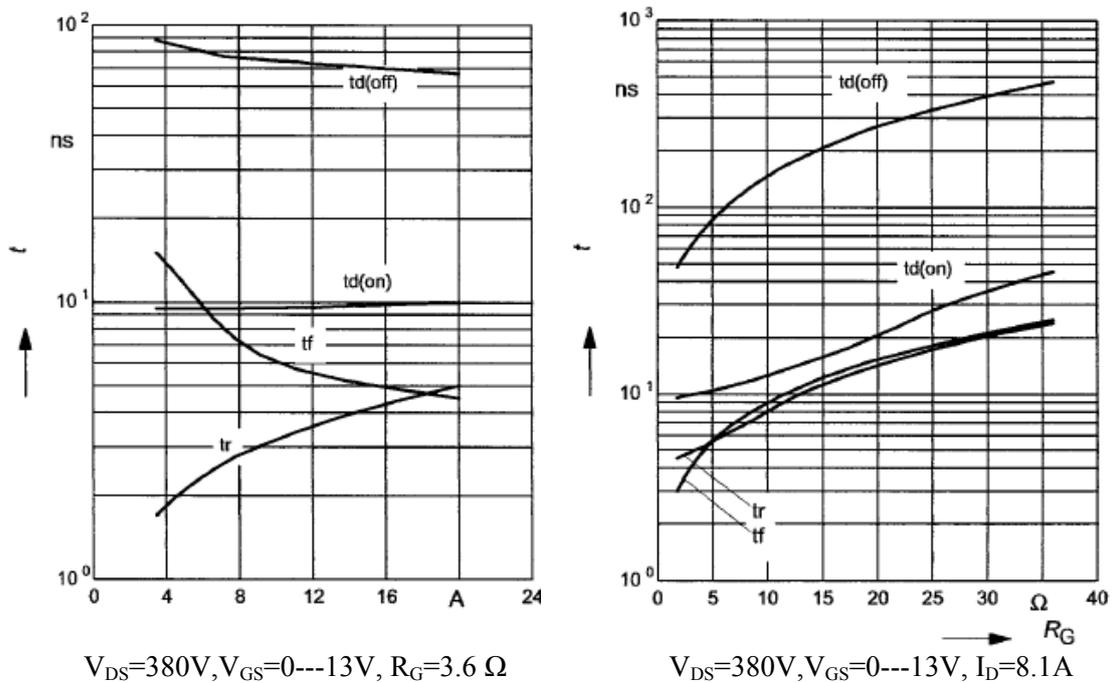


Fig.IV.20 Temps de commutation

La résistance R_G n'est pas choisie arbitrairement, elle permet de diminuer les temps de commutation, en augmentant le temps de montée et par conséquent la diminution du risque dv/dt . On utilise une résistance de $4.7\ \Omega$ en série avec chaque MOSFET [36] [37]. Le schéma global est donné par la figure IV.21

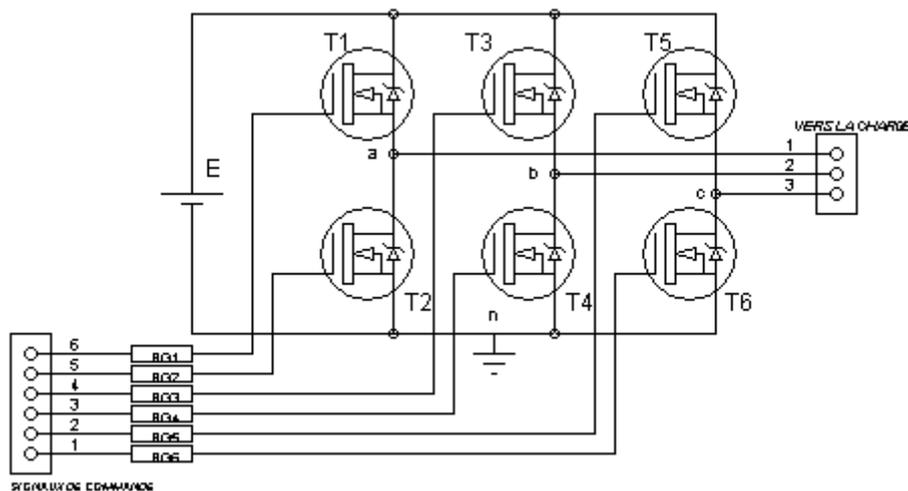


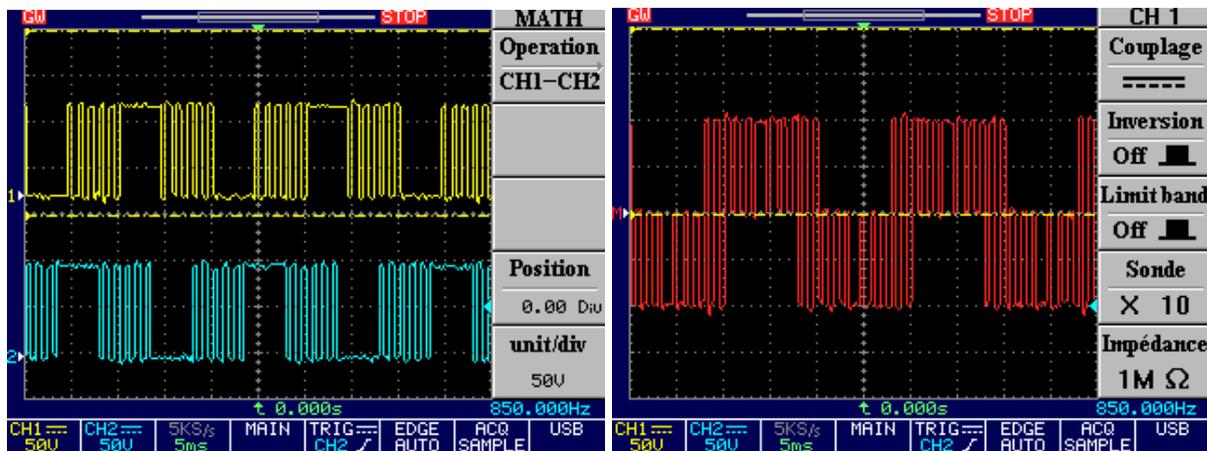
Fig. IV.21. Schéma de l'étage de puissance.

IV.5 Résultats expérimentaux

Après l'implémentation de la commande MLI on-line sur FPGA, un étage d'isolation et d'adaptation a été réalisé à base des optocoupleurs HCPL2200 et des drivers IR2111 afin de générer les signaux de commande rapprochée.

Les résultats expérimentaux sont visualisés sur oscilloscope numérique, puis récupérés sur ordinateur par une carte d'acquisition Instrunet, afin de les interpréter et les comparer avec les résultats obtenus en simulation. Plusieurs résultats sont présentés pour différentes valeur de i_m (consigne fréquence et par conséquent vitesse), et m (nombre des angles de commutation et par conséquent le nombre d'harmoniques à éliminer).

Pour la valeur 1 de i_m on a une valeur de 100% du fondamental, c'est la vitesse maximale du moteur, (correspondante à une fréquence de 50Hz), m est égale à 5.



(a): Tensions simples V_{an} (CH1), V_{bn} (CH2)

(b): Tension composée V_{ab}

Fig. IV.22. Tensions de sortie de l'onduleur pour $i_m=1$ et $m=5$

Dans la partie (a) de la figure IV.22 les tensions V_{an} et V_{bn} sont représentées, (b) est la tension composée V_{ab} .

Le spectre de la tension de phase est donné par la figure IV.24, on remarque bien l'élimination des harmoniques sauf ceux du troisième rang et ses multiples qui, comme nous allons voir, sont éliminés dans le spectre de la tension composée. Le rang du premier harmonique non éliminé est donné par la relation générale $3*m+2$, pour notre cas c'est $17 = (3*5 + 2)$. Le spectre de la tension composée est représenté sur la figure IV.25

Dans la pratique nous avons utilisé une alimentation de 97 V, et pour la mesure nous avons effectué une division par 21. Pour cela le signal obtenu a une amplitude de 4.61V

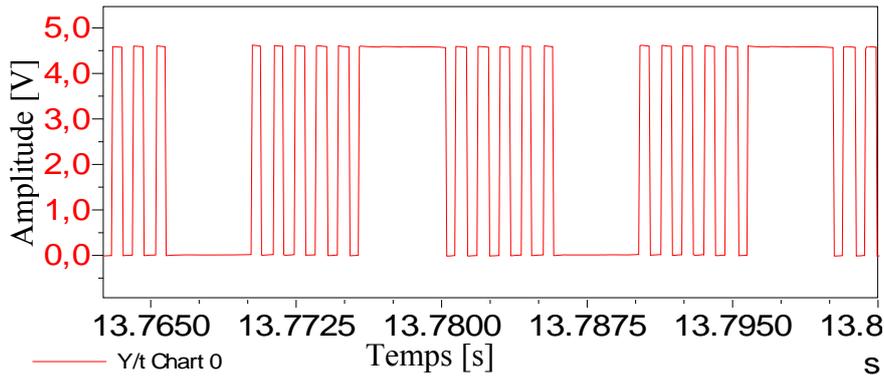


Fig. IV.23. Tension simple pour $im=1$ et $m=5$

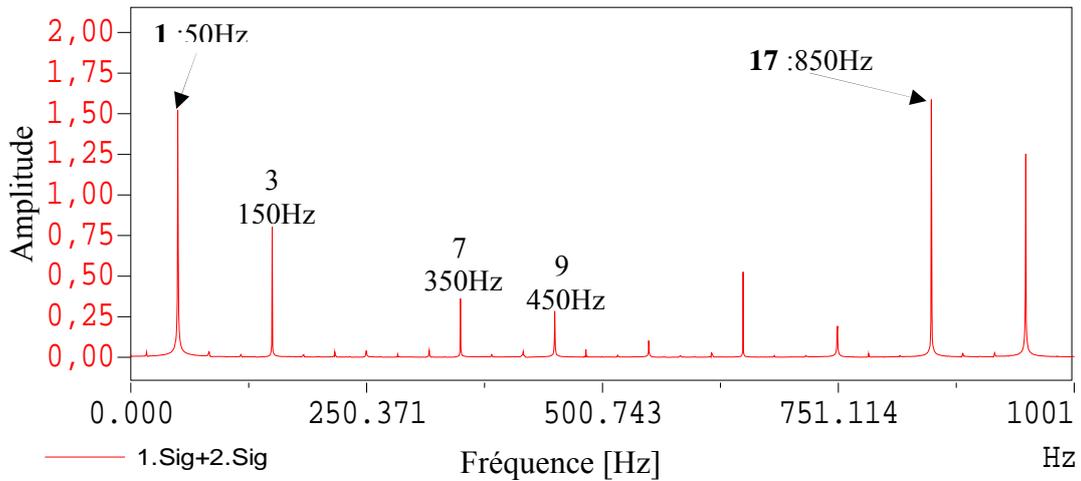


Fig.IV.24 Spectre de la tension simple pour $im=1$ et $m=5$

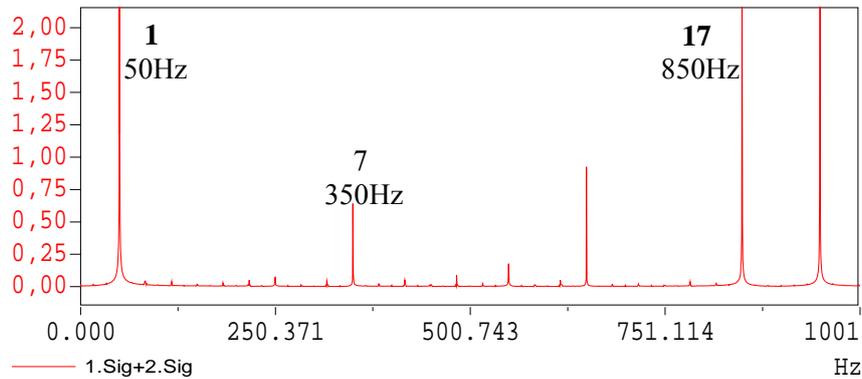


Fig.IV.25. Spectre de la tension composée pour $im=1$ et $m=5$

Pour im 0.9 ($m=5$ et $f=50*0.9$) la tension de sortie a une fréquence de 45 Hz et un fondamental d'amplitude de 90%. La figure IV.26 donne la forme de la tension V_{an} entre le point neutre n et la phase a , alors que les figures IV.27 et IV.28 représentent les spectres de la tension simple V_{an} et la tension composée V_{ab} .

Le premier harmonique non éliminé est celui du rang 17 (765Hz).

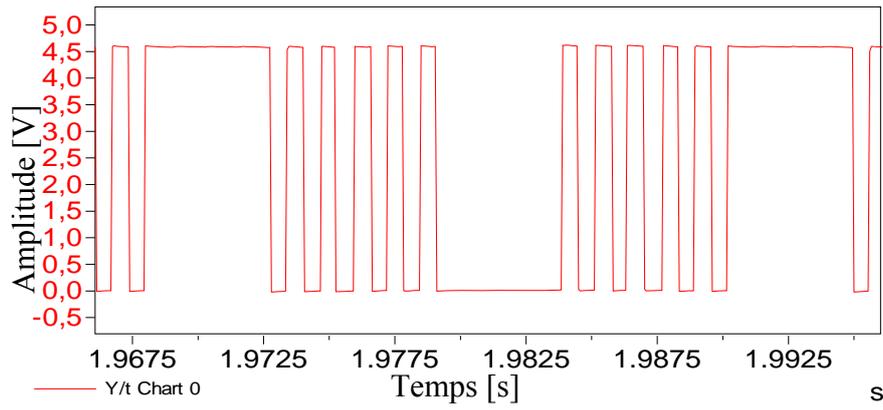


Fig.IV.26. Forme de la tension Van pour $i_m=0.9$ et $m=5$

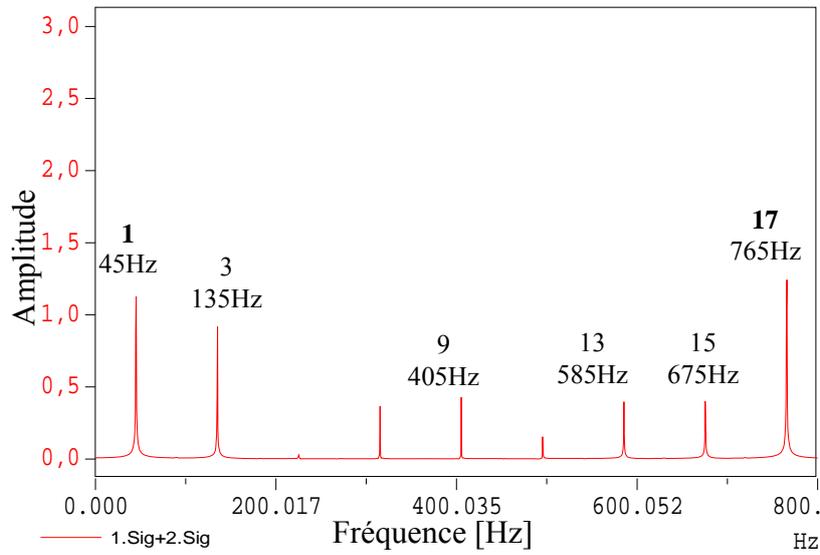


Fig.IV.27. Spectre de la tension simple Van pour $i_m=0.9$ et $m=5$

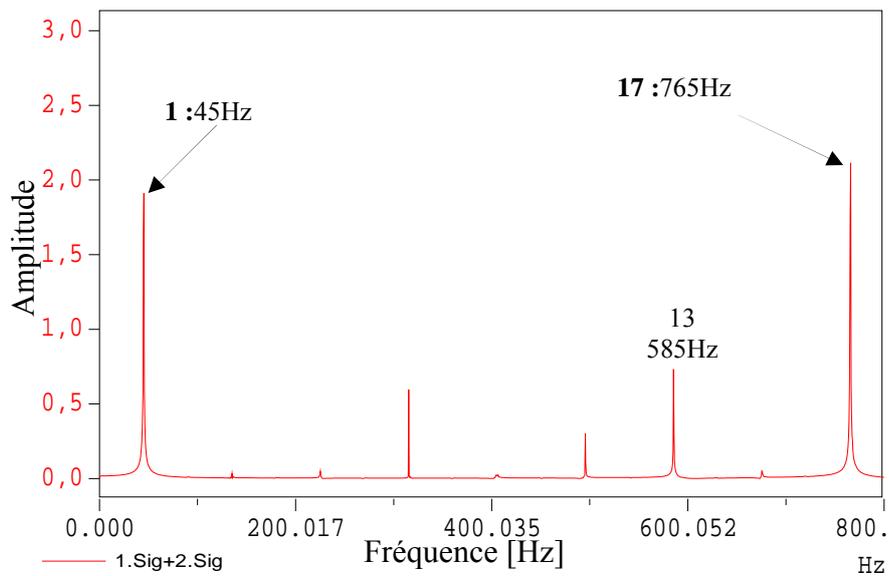


Fig.IV.28. Spectre de la tension composée Vab pour $i_m=0.9$ et $m=5$

Pour la valeur $i_m=0.8$ on a choisi 7 angles de commutation ($m=7$), les résultats sont donnés par les figures IV.29 et IV.30.

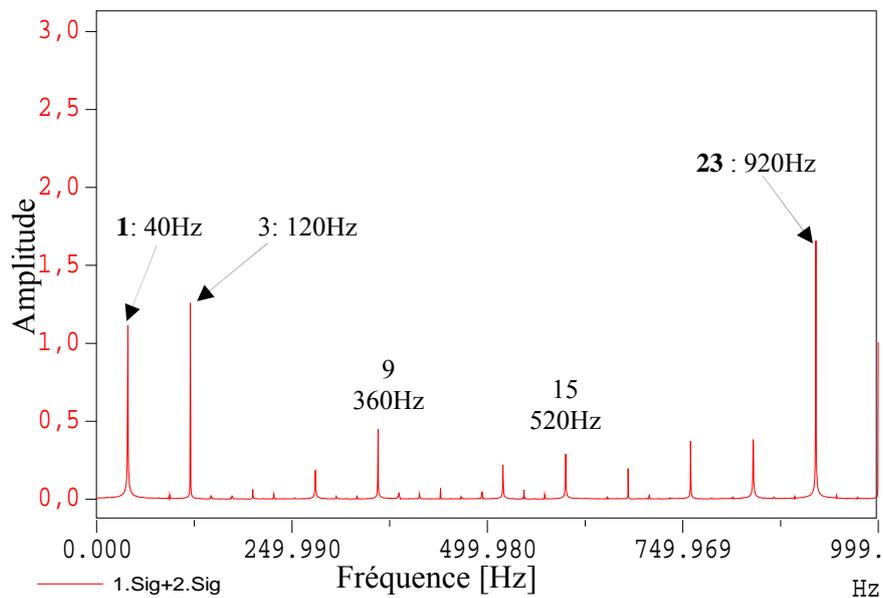


Fig.IV.29. Spectre de la tension simple Van pour $i_m=0.8$ $m=7$

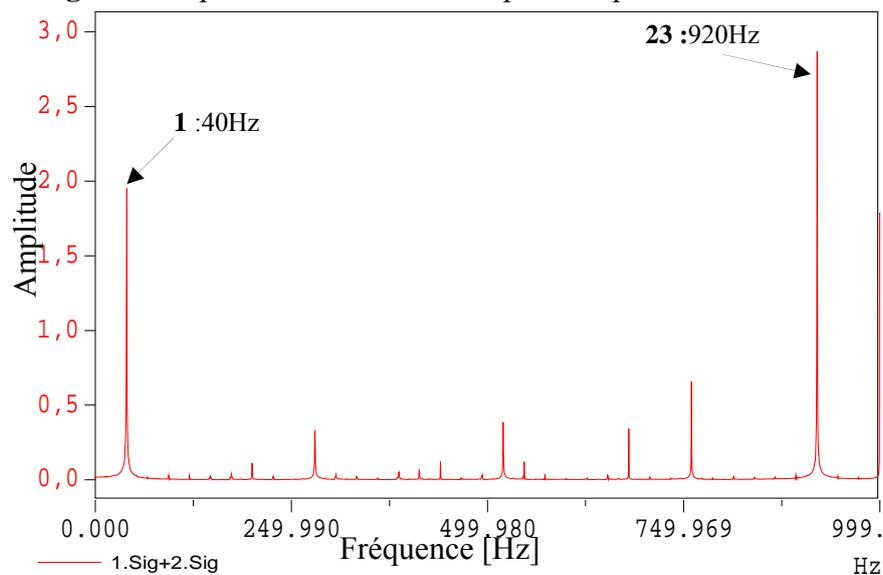


Fig.IV.30. Spectre de la tension composée Vab pour $i_m=0.8$ $m=7$

Pour $i_m=0.2$ le nombre m est égale à 19, donc le premier harmonique non éliminé est celui du rang $3*19+2$ (59), ce qui correspond à une valeur de 590Hz (la fréquence est de $10\text{Hz}=50*0.2$). La forme de la tension Van est donnée par la figure IV.31 et les spectres de la tension Van et la tension Vab sont représentés respectivement par les figures IV.32 et IV.33.

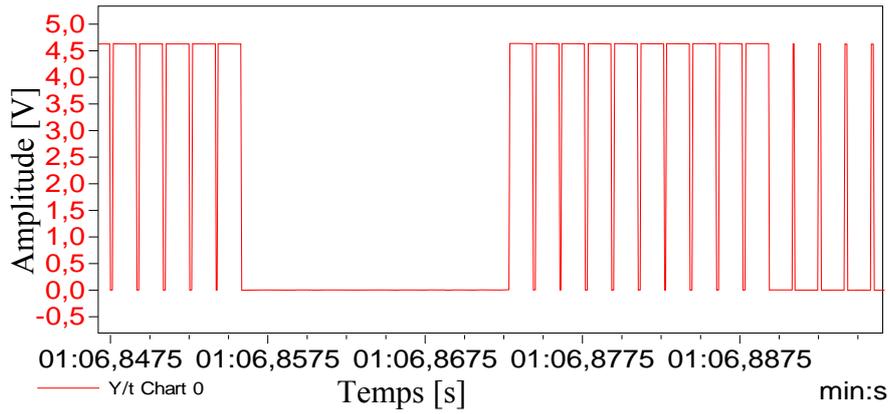


Fig.IV.31. Forme de la tension simple Van pour $i_m=0.2$ $m=19$

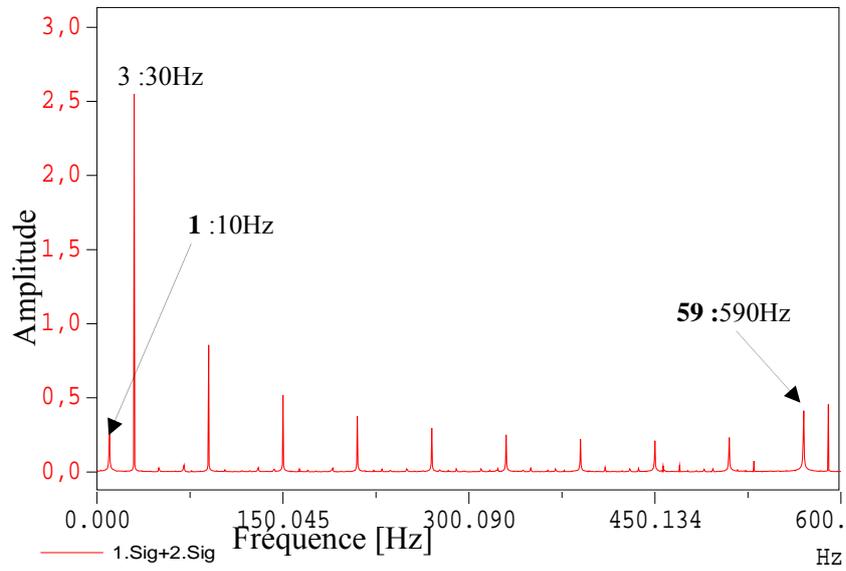


Fig.IV.32. Spectre de la tension simple Van pour $i_m=0.2$ $m=19$

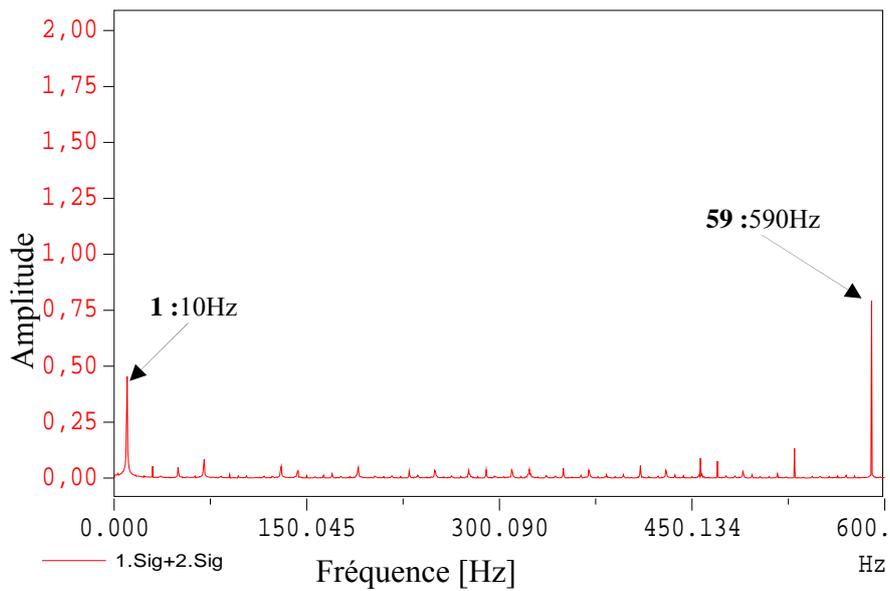


Fig.IV.33. Spectre de la tension composée Vab pour $i_m=0.2$ $m=19$

La petite valeur de i_m que nous avons prise est 0.1, ce qui correspond à une fréquence de 5Hz. Nous avons choisi 23 angles de commutation, le premier harmonique non éliminé est de rang 71 (Figure IV.34). Le fondamental est d'amplitude de 10%. La figure IV.35 donne le spectre de la tension composée.

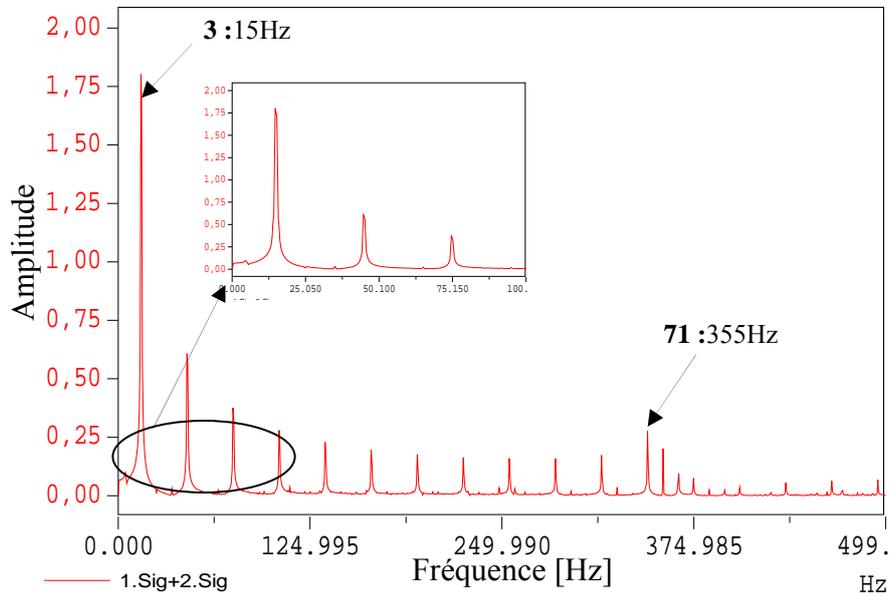


Fig.IV.34. Spectre de la tension simple Van pour $i_m=0.1$ $m=23$

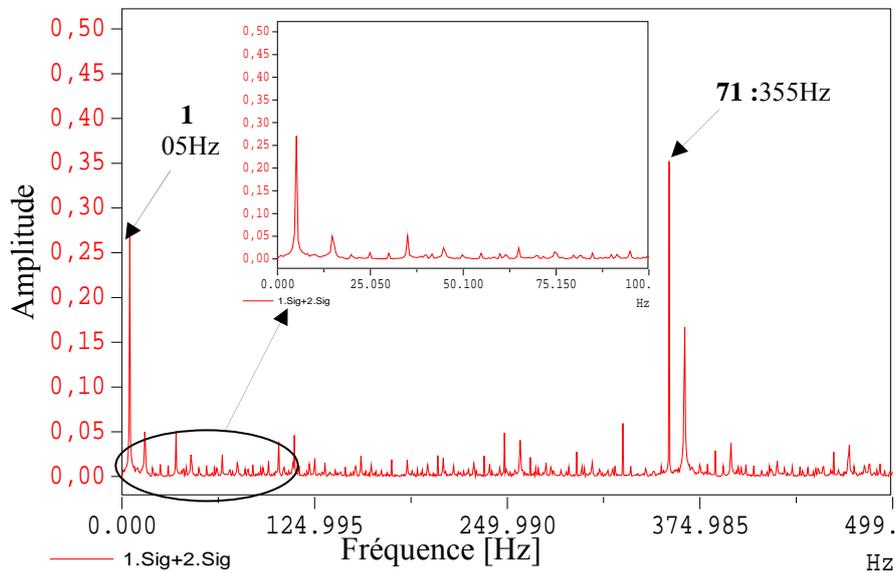


Fig.IV.35. Spectre de la tension composée Vab pour $i_m=0.1$ $m=23$

IV.6 Interprétation et discussion des résultats

Dans la pratique nous avons utilisé une alimentation de 97 V comme source continue à l'entrée de l'onduleur, et pour la mesure nous avons effectué une division par 21. Pour cela le signal obtenu a une amplitude de 4.61V. Les figures ci-dessous donnent une comparaison entre les signaux obtenus en pratique et les signaux obtenus par la simulation.

Les figures IV.36 et IV.37 donnent une comparaison entre les spectres des tensions de sortie obtenus en simulation et les résultats obtenus pratiquement pour $im=1$ et $m=5$. On est dans la valeur maximale du fondamentale, et par conséquent la fréquence, puisque nous avons utilisé un rapport V/f constant, la vitesse à son tour est maximale.

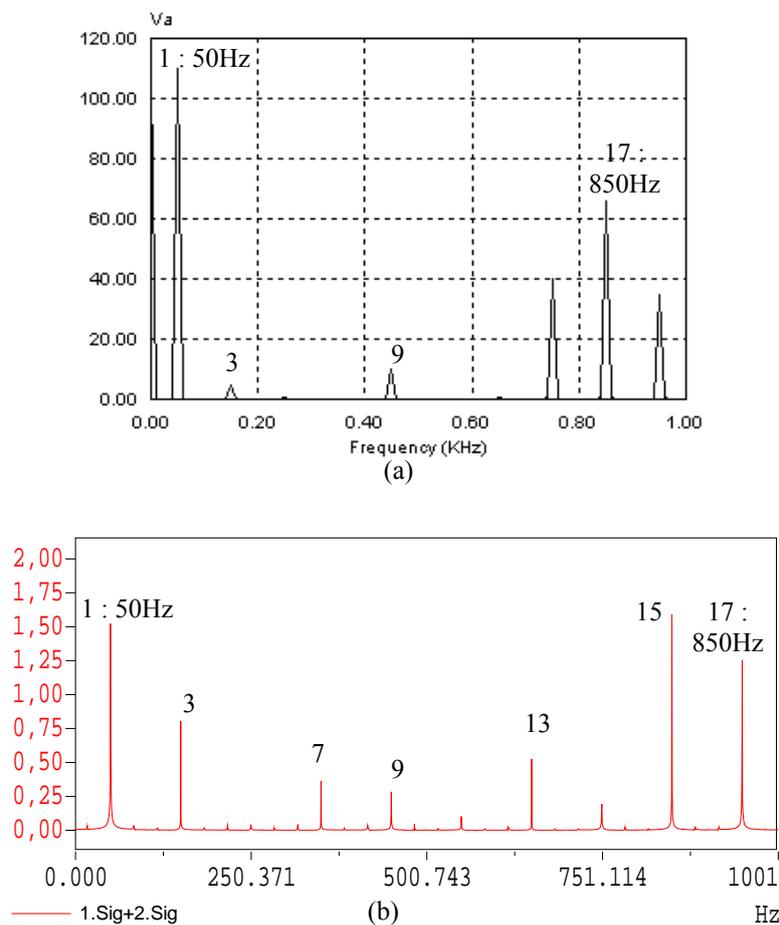


Fig.IV.36. Spectre de la tension simple V_{an} pour $im=1$ $m=5$

(a) : Simulation, (b) : Pratique

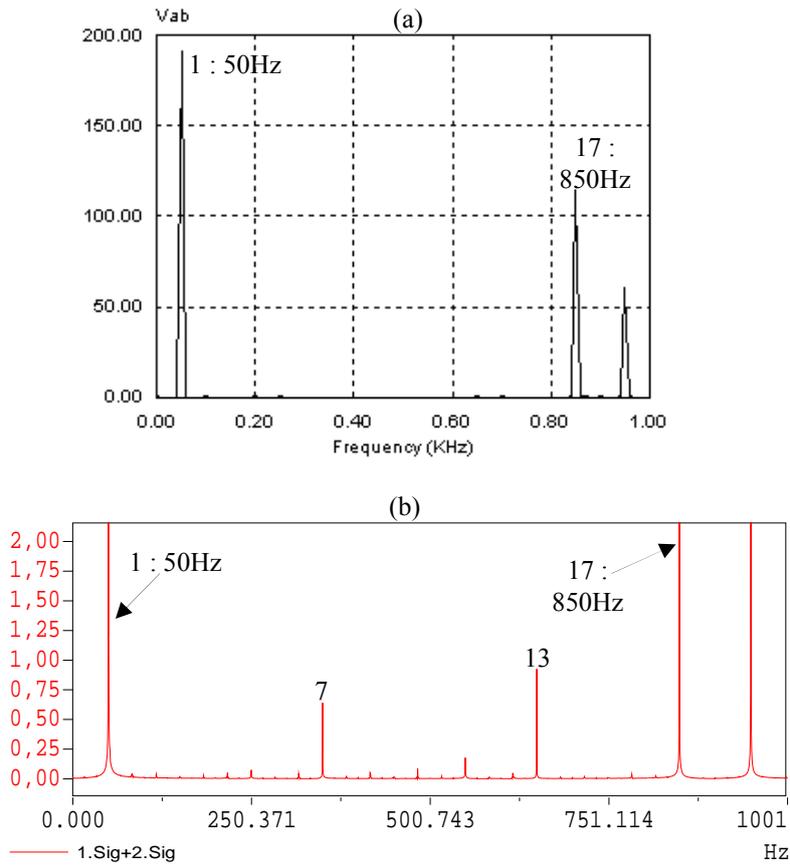


Fig.IV.37. Spectre de la tension V_{ab} pour $i_m=1$ $m=5$
 (a) : Simulation, (b) : Pratique

Pour $i_m=0.6$ et $m=7$ les résultats obtenus sont donnés par les figures IV.38 et IV.38.

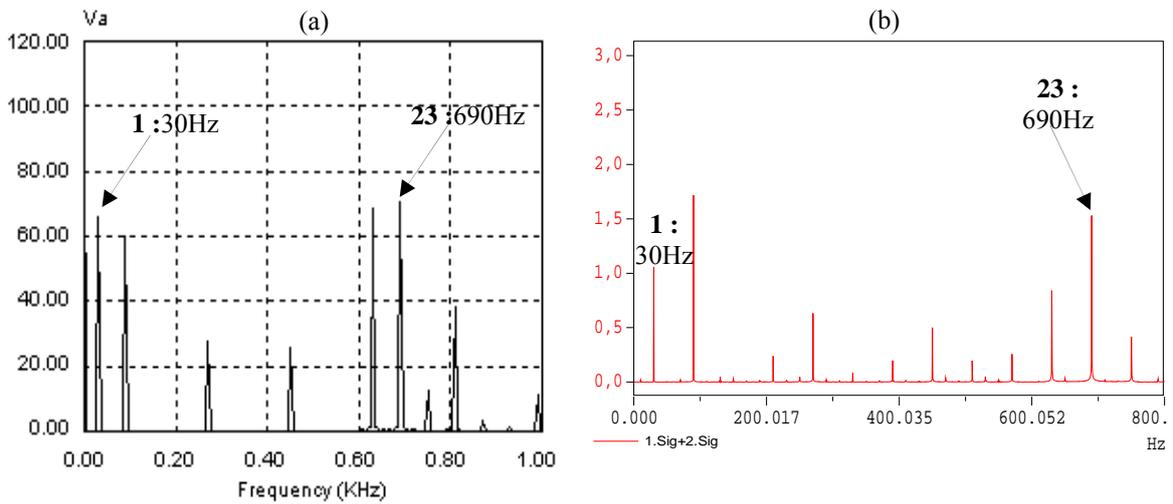


Fig.IV.38. Spectre de la tension V_{an} pour $i_m=0.6$ $m=7$
 (a) : Simulation, (b) : Pratique

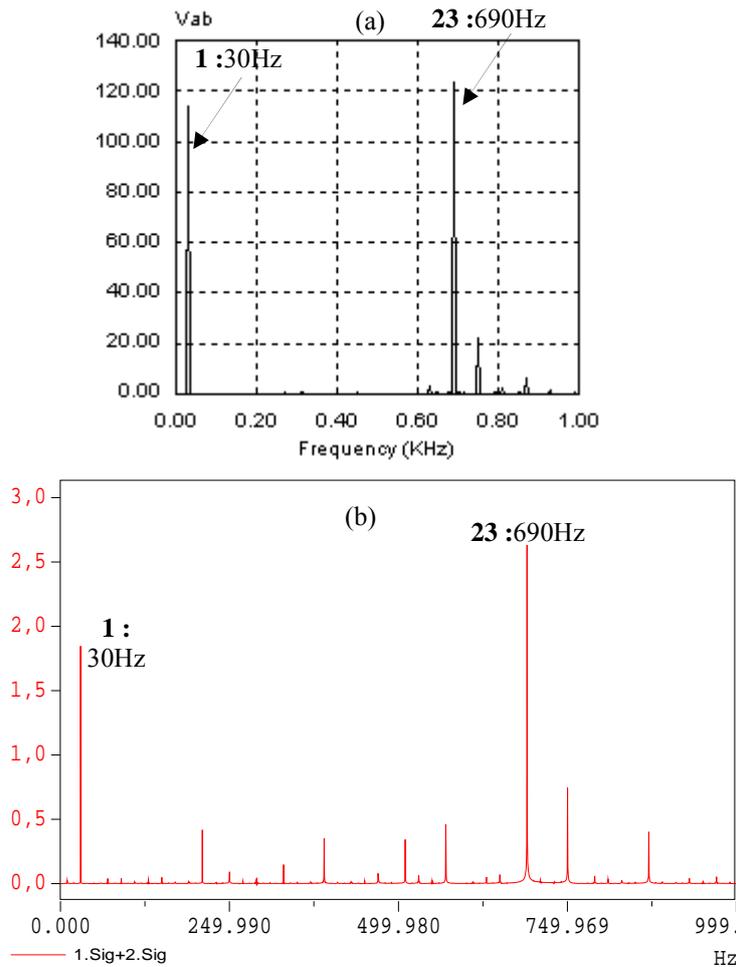


Fig.IV.39. Spectre de la tension V_{ab} pour $i_m=0.6$ et $m=7$
 (a) : Simulation, (b) : Pratique

Pour $i_m=0.1$ et $m=23$ angles de commutation les résultats sont présentés ci-dessous.

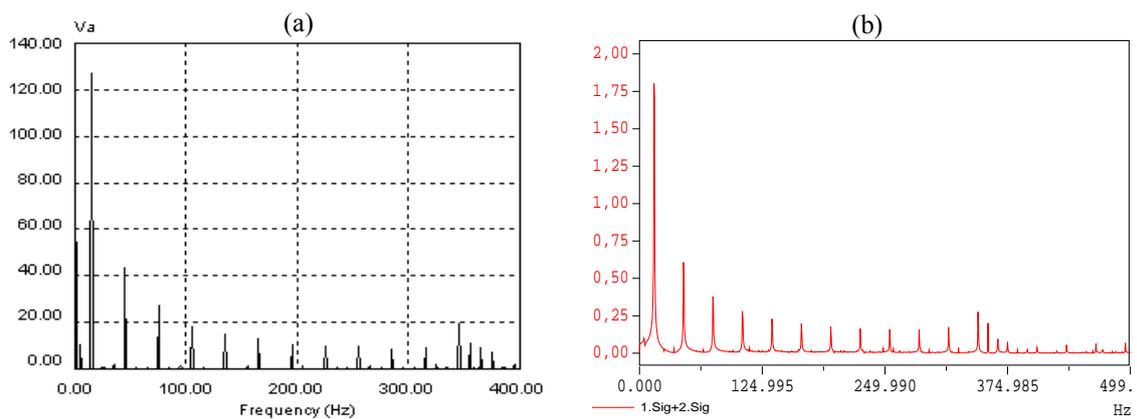


Fig.IV.40. Spectre de la tension V_a pour $i_m=0.1$ et $m=23$ (simulation)
 (a) : Simulation, (b) : Pratique

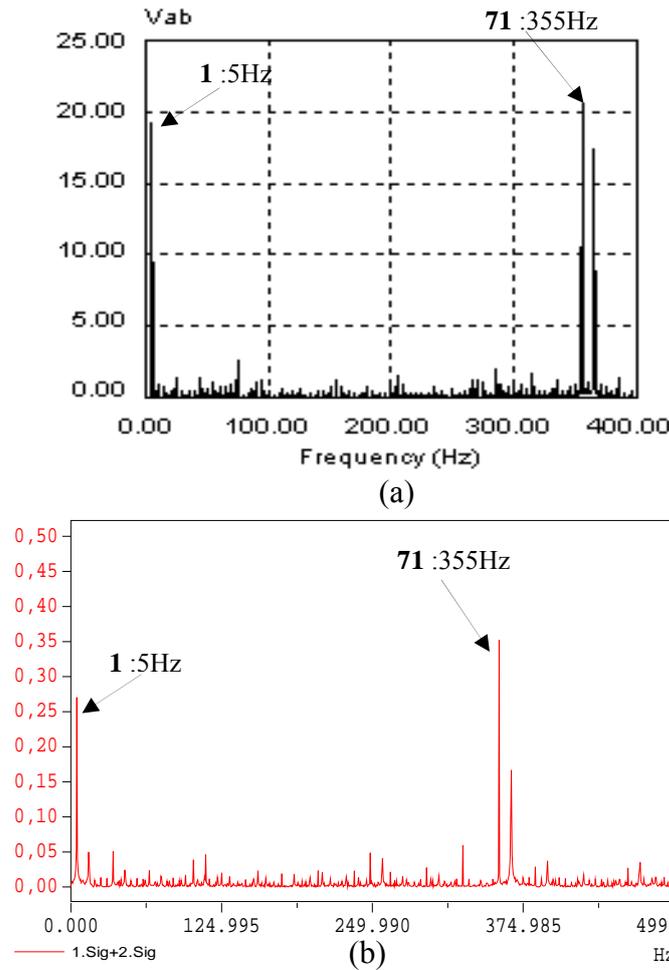


Fig.IV.41. Spectre de la tension V_{an} pour $i_m=0.1$ et $m=23$
 (a) : simulation (b) : pratique

En comparant les résultats obtenus dans la pratique avec ceux obtenus par simulation on peut dire que l’algorithme implémenté donne de très bons résultats dans les basses vitesses, c’est le cas de $i_m=0.1$ et $m=23$ figures IV.40 et 41. On a pu éliminer tous les harmoniques gênants jusqu’au rang désiré qui est 71. Le premier harmonique non éliminé est de fréquence $71*5=355\text{Hz}$.

La vitesse est fonction directe de la fréquence, nous faisons une commande en boucle ouverte qui est une caractéristique de la commande scalaire V/f . En augmentant la vitesse presque à la moitié où $i_m=0.6$ la période diminue, on remarque une légère différence entre les signaux réels et les signaux de simulation ; mais on a toujours le premier harmonique non éliminé de grande amplitude. Il est de rang $3*m+2=23$, ce qui correspond à une fréquence de $23*0.6*50=690\text{ Hz}$ (figures IV.39 et 40).

Pour $m=1$ c'est la vitesse et la fréquence maximales, c'est la période minimale ici deux harmonique n'ont pas été éliminés mais leur valeur peut être négligeable par rapport au premier harmonique non éliminé (figures IV.36, 37, 38 et 39).

Cette diminution de précision qui a causé cette différence est due essentiellement à la taille du circuit FPGA utilisé où on a rencontré un problème très délicat dans la manipulation des nombres en virgule flottante. Nous avons été face à un compromis précision : ressources, mais nous avons choisi de dégrader un peu la précision car on est dans une application d'un variateur de vitesse d'où la nécessité de travailler en régime de fréquence variable.

Une autre chose qui pourrait être cause c'est le temps mort introduit par les driver car celui-ci change un peu la forme des ondes obtenues à la sortie de l'onduleur.

Conclusion

Les simulations effectuées donnent de bons résultats. De même l'algorithme MLI on-line a été correctement implémenté sur le circuit FPGA. A la fin de cette réalisation nous avons déduit que les MOSFET sont bien adaptés à fonctionner dans des fréquences élevées, le driver IR2111 nous a résolu facilement le problème du transistor high side.

Les résultats obtenus en pratique sont acceptables en comparaison avec ceux obtenus en simulation Il est clair que les harmoniques sont éliminés et la valeur du fondamental est contrôlable. Coté précision, dans les basses vitesses nous avons effectivement éliminé les harmoniques désirés et avec une grande précision.

Le problème est dans les grandes vitesses où la période est petite et la précision diminue. Cette diminution est fortement liée aux approximations faites dans le programme de calcul des angles, car on n'a pas disposé des bibliothèques traitant les nombres en virgule flottante. Une telle bibliothèque pourrait résoudre ce problème, mais on détriment de la quantité de ressources utilisées par le circuit FPGA.

CONCLUSION GENERALE

Au cours de ce travail nous avons réalisé un variateur de vitesse pour moteur asynchrone triphasé. Pour cela nous avons commencé par introduire des notions de base sur ce type des moteurs et sur les onduleurs de tension.

Ensuite nous avons entamé la technique MLI calculée à élimination d'harmonique, en donnant son principe et ses avantages par rapport à la MLI classique à porteuse sinusoïdale. Cette technique permet l'élimination des harmoniques et l'asservissement du fondamental tout en maintenant un rapport V/f constant. Ce qui permet d'obtenir un bon spectre dans les grandeurs de sortie de l'onduleur et d'avoir un couple maximal et constant. Les harmoniques de rang élevé peuvent être facilement filtrés. Alors que son point faible réside dans la difficulté d'un calcul on-line des angles de commutation, ce qui empêche une commande en temps réel de la vitesse d'un moteur asynchrone.

Afin de résoudre ce problème, nous avons proposé un algorithme qui permet un calcul on-line des angles de commutation, nous l'avons simulé, de bons résultats ont été obtenus.

Lors du passage à l'implémentation nous avons choisi d'utiliser des architectures reconfigurables FPGA, qui donnent de bonnes performances pour les algorithmes avec des contraintes temporelles. L'absence des bibliothèques traitants les nombres flottants nous a conduits à faire quelques approximations, malgré cela, les résultats obtenus sont acceptables.

L'utilisation des MOSFET dans notre application a prouvé qu'ils peuvent fonctionner à des fréquences élevées. La résolution du problème du transistor high side nous a pris un temps considérable.

Pour la commande V/f utilisée elle a permis d'éliminer les harmoniques, et de contrôler la valeur du fondamental, elle permet de varier la vitesse en variant la fréquence et l'amplitude en même temps, de zéro à la vitesse nominale du moteur.

En perspectives, nous proposons l'amélioration de la précision de cet algorithme soit en utilisant des cartes qui incluent à la fois des DSP et des FPGA ; ou l'amélioration

du code développé en lui ajoutant des bibliothèques traitant les nombres en virgule flottante, mais ceci influe sur la quantité de ressources utilisées.

Nous espérons que cette étude sera suivie par l'étude d'autres techniques MLI calculée afin de les comparer, les bien évaluer voire les améliorer.

Pour l'étage de puissance des protections doivent être envisagées comme par exemple, une protection contre les courts circuits et les surtensions. Ces améliorations rendent le système réalisé un banc de test facilitant ainsi le travail des chercheurs dans notre laboratoire.

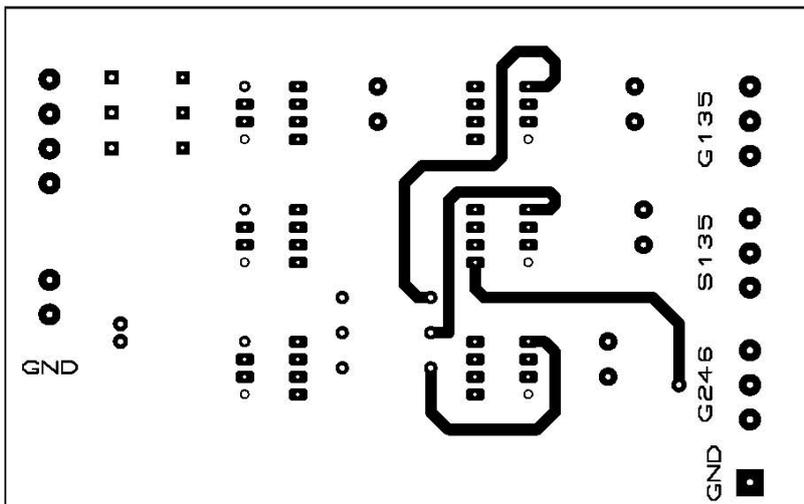
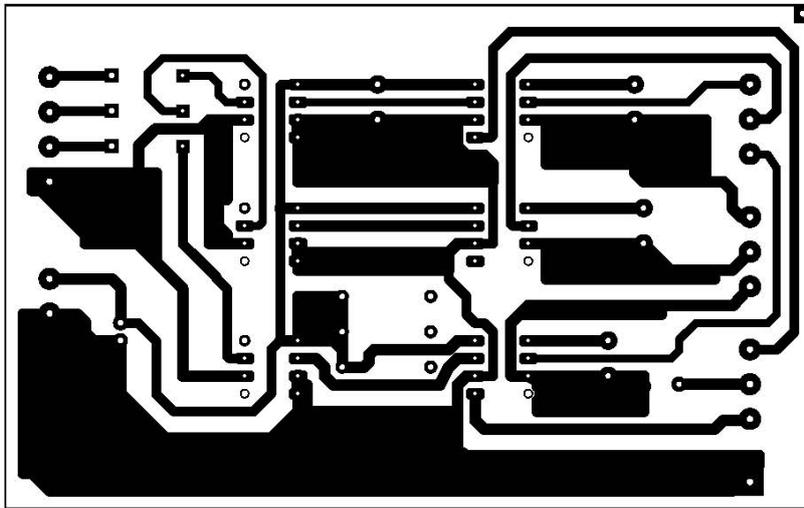
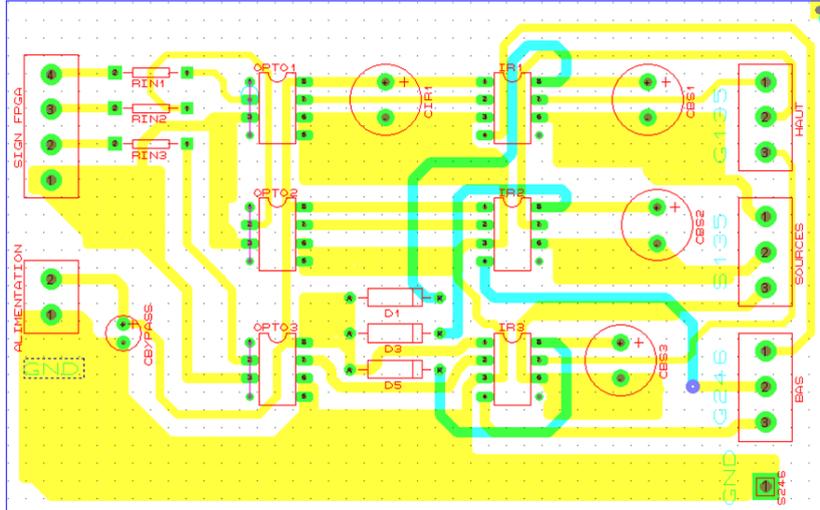
BIBLIOGRAPHIE

- [1] M'hamed BOUNEKHLA, " Contribution à l'identification paramétrique de la machine asynchrone ", Thèse de Doctorat d'Etat ès-Sciences ENP 2004.
- [2] Lotfi BAGHLI, "Modélisation et commande de la machine asynchrone", Notes de cours, IUFM de Lorraine, UHP 2005.
- [3] Theodor WILDI, "L'électrotechnique", 3^{ème} édition, éditions de Boek Université, Bruxelles 2000, 1196p.
- [4] E.Acha, V.G.Agelidis,O.Anaya-Lara,"Power Electronics Control in Electrical Systems" Newnes 2002
- [5] R.KABOUCHE,"Etude et élaboration d'une technique MLI à élimination d'harmoniques optimale", Mémoire de Magister en Electronique, Ecole Nationale Polytechnique, Alger, 2002
- [6] H.Foch, F.Forest et T.Meynard, "Onduleurs de tension. Structures. Principes. Applications" Techniques de l'ingénieur, vol. D31, N^o D3176, Novembre 1998.
- [7] F.Cherfa, " étude et réalisation d'une centrale photovoltaïque connectée au réseau de distribution électrique bt", Mémoire de magister en électronique, Ecole Nationale Polytechnique, Alger, 2004
- [8] H.S.Patel and R.G.Hoft,"Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part I- Harmonic Elimination, IEEE Trans.Ind.App., vol.1A-9, pp. 310-317 NO. 3,May/June 1973
- [9] P.N.Enjeti, P.D.Ziogas and J.F.Lindsay,"Programmed PWM Techniques to Eliminate Harmonics: A Critical Evaluation", IEEE Trans.Ind.App.,vol.26,NO 2, pp. 302-316, MARCH/APRIL 1990.
- [10] Z .Salam S.I.Safie L.J.Yiap "An On-Line Harmonics Elimination PWM Scheme for Three-phase Voltage Source Inverters" Fifth International Conference Power Electronics and Drive Systems IEEE PEDS, Vol.1, 17-20 Nov 2003.
- [11] Z.Salam and C.T.Lynn, "Algorithm for Near Optimal Harmonics Elimination PWM Based on Quadratic Approximation Method" IECON'02, Seville, Spain, Nov 2002
- [12] N.A.Azli and L.Y.Chan , "Development of Equations Through Trajectories Linearization for an HEPWM Inverter ", First international PECon 2006
- [13] S.R.Bowes and S.Grewal, "Novel Harmonic PWM Control Strategies for Three-Phase PWM Inverters Using Space Vector Technique" IEE Proc-Electr. Power Appl. Vol 146, N05, September 1999 pp 495-502
- [14] Z Salam and C. T. LYNN," A new near optimal harmonics elimination PWM algorithm for ac traction drives"Jurnal Teknologi 36(D),pp.111–128 Universiti Teknologi Malaysia Jun 2002
- [15] Taufiq, J. A., B. Mellitt and C. J. Goodman. 1986. Novel Algorithm for Generating Near Optimal PWM Waveforms for AC Traction Drives. IEE Proceedings. Vol 133, Pt. B, No. 2pp 85-94
- [16] Khider M, " commande de vitesse en temps réel d'un moteur asynchrone triphasé " mémoire de magister, Ecole Nationale Polytechnique, Alger, 2003
- [17] S. R. Bowes and P. R. Clark "Simple microprocessor implementation of new regular-sampled harmonic-elimination PWM techniques," IEEE Trans. on Ind. Appl., Vol. 28, No. 1, pp. 89-95,1992.
- [18] S. R. Bowes, and P. R. Clark, "Regular-Sampled Harmonic Elimination PWM control of inverter drives," IEEE Trans. on Power Electronics, Vol. 10, No. 5, pp. 521-531, 1995.
- [19] JQ.Wang et L.Chang, "PWM control strategies for wind turbine inverters"IEEE Conf 1998, pp 309-312
- [20] J.Vignes, "Algorithmes numériques, analyse et mise en œuvre tome 2: Equations et systèmes non linéaires " Technip, Paris,1980, 302p (Colloque langage et algorithmes informatiques

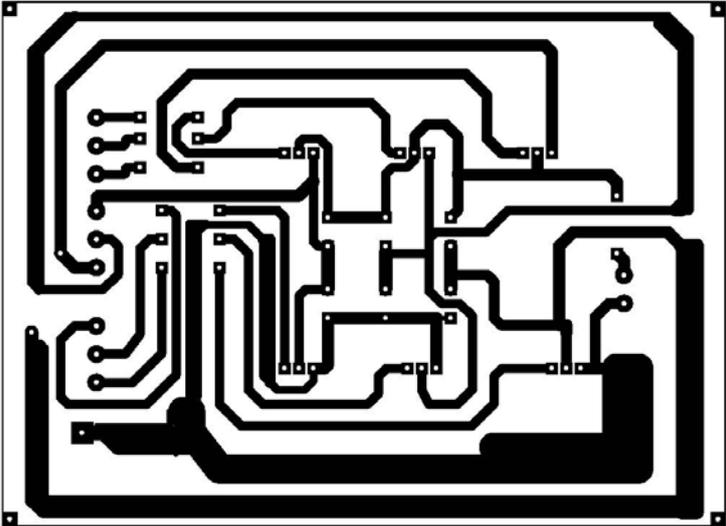
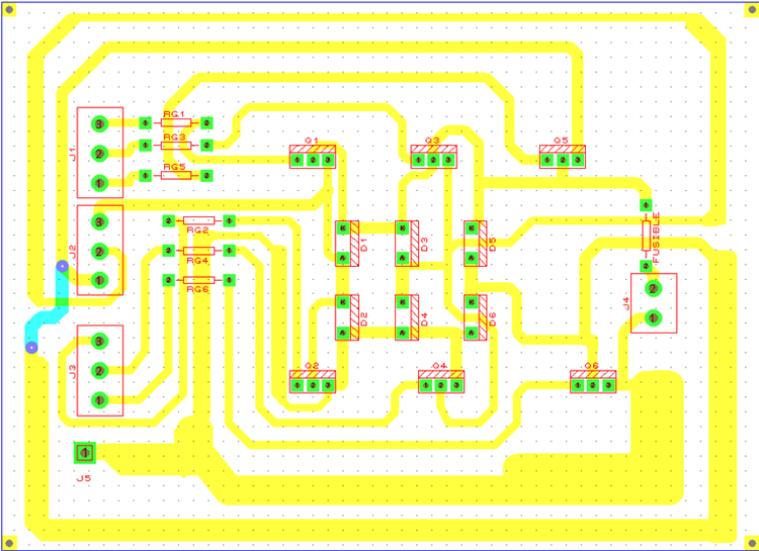
- [21] J.R. Espinoza, "Inveters" In: M.H.Rashid, "Power Electronic Handbook", Academic Press USA 2001, pp 225-269
- [22] E.M.BERKOUK, "Contribution à la conduite des machines asynchrones monophasées et triphasées alimentées par des convertisseurs directs et indirects .Application gradateurs et onduleurs multiniveaux."Thèse de doctorat. C.N.A.M., Paris 1992
- [23] Juan J- Rodriquez-Andina, "Features, Design Tools, and Application Domains of FPGAs", IEEE Trans. Ind. Electron., vol.54,NO.4, Aout 2007
- [24] Eric Monmasson, Marcian N. Cirstea, "FPGA Design Methodology for Industrial Control Systems- A Review", IEEE Trans. Ind. Electron., vol. 54,no. 4,Aout 2007.
- [25] Claude Guex Introduction au VHDL Ecole d'ingénieurs de Canton de Vaud Juin 98
- [26] S.Berto, A.Paccagnella, M.Ceschia, "Potentials and Pitfalls of FPGA Application in Inverter Drives - a Case Study" in Proc. IEEE ICIT 2003, pp.500-505
- [27] J P Deschamps, G J Antoine Bioul, "Synthesis of Arithmetic Circuits FPGA, ASIC, And Embedded Systems" JOHN WILEY & SONS, 2006.
- [28] Volnei A Pedroni, "Circuit Design With VHDL" MIT Press 2004.
- [29] Dragomir Milojevic, "Implémentation des filtres non-linéaires de rang sur des architectures universelles et reconfigurables " These de Doctorat en Sciences Appliquées, Université Libre de Bruxelles, 2004
- [30] Bob Zeidman, "Designing with FPGAs and CPLDs" ELSEVIER 2002
- [31] P Wilson, "Design recipes for FPGAs" Newnes 2007
- [32] PONG P. CHU. "RTL Hardware esign Using VHDL" USA:John Wiley & Sons 2006, 669p.
- [33] International Rectifier, " HV Floating MOS-Gate Driver ICs", Appl. Notes 978.
- [34] S.Lefebvre et B.Multon, "MOSFET et IGBT : circuits de commande", Techniques de l'ingénieur D3 233, 16p
- [35] DATASHEET optocoupleur HCPL 2200
- [36] FAIRCHILD Semi.Conductorm, "Suppressing MOSFET Gate Ringing in Converters: Selection of a Gate Resistor", Application Bulletin AB-9.
- [37] R.Torres, R.S.Murphy and S. Decoutere, " MOSFET gate resistance determination", Electronics letters 23rd January 2003 Vol. 39 No. 2 pp 248-250
- [38] Omar M F Muhialdin, Ahmad Nahas and Takyin Chan , "The Design and Implementation of a Three Phase Power Converter in the Power Electronics and Drives Subject" Converter Project (AC/DC/AC) Report” School of Electrical Engineering Victoria University Australia
- [39] SCHULTZ (W.). – Drive techniques for high side N channel MOSFET. Revue PCIM, Juin 1987, pp. 34-40.
- [40] N.Mohan, M.Tore and P.William, "Power electronics. Converters, Applications and Design", 2nd Ed. New York, JOHN WILEY & SONS, 2003 802p.
- [41] International Rectifier. -Bootstrap Component Selection For Control IC's. Note d'applications DT 98-2a
- [42] M.Haddadi, "Semi conducteurs de puissance", cours PG ENP Alger 2000.
- [43] M.Correvon, " Les semi-conducteurs de puissance 02: le MOSFET", haute Ecole d'Ingénierie et de Gestion du Canton du Vaud, pp83.
- [44] Blasckhe,F., "The principle of field oriented as rotationg-field machines," Siemens Review XXXIX, n^o 5, pp. 217-220, 1972

ANNEXE

1. Carte de commande



2. Partie puissance de l'onduleur



3. Photo du variateur de vitesse réalisé

