



MEMOIRE

En vue de l'obtention du diplôme de :

Magister en Electrotechnique

Option: Entrainements Electriques

Thème :

**CONCEPTION D'UN CONTROLEUR D'ETAGE DE
PUISSANCE PAR FPGA**

Présenté par : M^r CHIBAH AREZKI

Soutenu le:

Devant le jury composé de:

PRESIDENT :

M^r. Chaibi Rachid : Professeur UMMTO

RAPPORTEUR:

M^r. Laghrouche Mourad : Maître de conférences A UMMTO

EXAMINATEURS:

M^r. Mohellebi Hacene : Professeur UMMTO

M^r. Melah Rabah : Maître de conférences A UMMTO

Remerciements

Au terme de ce modeste travail, je tiens à exprimer mes vifs remerciements et ma gratitude à :

M^r Laghrouche M., maître de conférences au département d'Electronique, faculté de Génie Electrique et d'Informatique à l'Université Mouloud Mammeri de Tizi-Ouzou, pour m'avoir encadré et soutenu au cours de la réalisation de ce projet.

M^r Auger F, maître de conférences hors classe, habilité à diriger des recherches au sein de l'Ecole Polytechnique de Nantes Atlantique, pour son orientation, sa disponibilité, son écoute, ses conseils et son soutien.

M^r Chaibi R, professeur au département d'Electrotechnique, faculté de Génie Electrique et d'Informatique à l'Université Mouloud Mammeri de Tizi-Ouzou, pour m'avoir fait l'honneur d'accepter la présidence du jury de soutenance de ce mémoire.

M^r Mohellebi H, professeur au département d'Electrotechnique, faculté de Génie Electrique et d'Informatique à l'Université Mouloud Mammeri de Tizi-Ouzou, pour m'avoir fait l'honneur d'examiner mon travail.

M^r Melah R, maître de conférences au département d'Automatique, faculté de Génie Electrique et d'Informatique à l'Université Mouloud Mammeri de Tizi-Ouzou, pour m'avoir fait l'honneur d'évaluer ce travail.

*J*adresse mes sincères remerciements à **M^{me} Mansouri. O**, pour ses encouragements, son aide et son soutien qui m'ont été très précieux.

Que tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail; trouvent ici ma profonde reconnaissance.

SOMMAIRE

SOMMAIRE

Introduction Générale	1
Chapitre I- Apprentissage-approfondissement des circuits logiques programmables, du langage VHDL et de la chaine de développement ISE-Weblink.	
I-Introduction	4
II-Circuits logiques programmables.....	4
II-1- Introduction.....	4
II-2 Définition des PLDs.....	4
II-3- Structure de base d'un PLD.....	5
II-4- Les différentes familles de PLD.....	6
II-4-1- Les PALs.....	6
II-4-2- Les GALs.....	7
II-4-3- Les EPLDs (Erasable Programmable Logic Device).....	8
II-4-4- Les CPLDs (Complex programmable logic device).....	9
II-4-5- Les FPGAs (Field Programmable Gate Array).....	9
II-5- Les FPGAs	10
II-5-1- Présentation.....	10
II-5-2- Classification des FPGAs.....	13
II-5-2-1- FPGAs à SRAM.....	13
II-5-2-2- FPGAs à Anti-fusible.....	14
II-5-3- Familles des FPGAs de XILINX.....	14
II-5-3-1- Famille SPARTAN 3E.....	15
II-5-3-2- Architecture interne d'un circuit SPARTAN 3E	15
III- Chaine de développement.....	19
IV - Langage de description VHDL.....	20
IV -1-Introduction.....	20
IV -2- Définition du VHDL.....	21
IV -3- Structure d'une description VHDL.....	21
IV-3-1- Entité.....	21
IV 3-2- Architecture.....	22
IV-3-2-1-Description comportementale.....	22
IV-3-2-2-Description structurelle.....	23
IV-3-2-3-Description Mixte.....	24

IV-3-2-4- Process.....	24
IV-4- Les Objets utilisés en VHDL.....	24
IV-4-1- Les constantes.....	25
IV-4-2- Les variables.....	25
IV-4-3- Les signaux.....	25
IV-5- Types.....	26
IV-6 Instructions concurrentes et séquentielles.....	26
IV-6-1- Les instructions concurrentes.....	26
IV-6-2- Les instructions séquentielles.....	27
IV-7- Les attributs.....	30
IV-8- Paquetage.....	30
V- Environnement ISE	31
VI-Conclusion.....	31

Chapitre II- Principe des techniques de modulation de largeur d'impulsion

I- Introduction	32
II- Modélisation de l'onduleur.....	32
II-1- Définition de l'onduleur.....	32
III--Stratégie de commande des onduleurs.....	35
III-1-Introduction.....	35
III-2-Commande MLI Naturelle ou Sinusoïdale.....	37
III-2-1-Introduction :.....	37
III-2-2-Définition de la commande MLI Naturelle	37
III-2-3-Caractéristique de la commande MLI sinusoïdale.....	38
III-3- MLI vectorielle	39
III-3-1-Acquisition de la consigne V_s	41
III-3-2-Détermination du secteur.....	42
III-3-3-Décomposition du vecteur de tension de référence.....	43
III-3-4- Elaboration des séquences de commande des interrupteurs.....	45
IV-Conclusion.....	47

Chapitre III- Analyse du besoin d'un circuit MLI adapté à la commande sans capteur de la MSAP

I-Introduction	48
II--Commande par autopilotage.....	48
III- Commande directe du couple (DTC).....	49
IV-Commande Vectorielle	51
IV-1- Introduction.....	51
IV-2- Principe de la commande vectorielle.....	52
IV-3- Commande vectorielle de la MSAP.....	54
IV-3-1- Commande vectorielle de la MSAP alimentée en tension.....	54
IV-4-Commande sans capteur de la MSAP.....	55
IV-4-1-Introduction.....	55
IV-4-2- Commande sans capteur mécanique par la méthode d'injection d'un signal haute fréquence (HF).....	56
IV-4-2-1-Principe.....	56
IV-4-2-2-Modèle de la MSAP à Haute fréquence	56
IV-4-2-3- Extraction de la séquence négative du courant à haute fréquence.....	59
IV-4-2-3-a-Extraction de la composante négative par un filtre passe bande suivi d'un filtre synchrone passe haut.....	59
IV-4-2-3-b-Extraction de la composante négative par deux filtres synchrones passe haut..	60
IV-4-2-3-c-Extraction de la composante négative par un filtre synchrone passe bas.....	61
IV-4-2-4-Estimation de la position du rotor.....	62
IV-4-2-4-a- Estimation par la fonction Arctg.....	62
IV-4-2-4-b-Estimation de la position du rotor par la boucle de verrouillage de phase.....	63
V- Conclusion.....	65

Chapitre IV- Réalisation et résultats expérimentaux

I-Introduction.....	66
II- Programmation du FPGA.....	66
II-1- Description de la plateforme Basys 2.....	68
II-2- Description du code VHDL.....	69
II-2-1- Amplitude	69
II-2-2- Amplitude-angle	69
II-2-3- Oscillateur.....	70

II-2-4- add-120	71
II-2-5- Amplitude_ckt.....	71
II-2-6- Adder	73
II-2-7- PWM.....	73
II-2-8-Three_Phase_Pwm.....	74
II-2-9-Clock-Divider.....	74
II-3-Sommaire du design.....	75
II-4- Résultats Expérimentaux.....	78
II-5-Interprétation des résultats.....	81
Conclusion Générale	82
Bibliographie	
Annexe 1	
Annexe 2	
Annexe 3	
Annexe 4	
Annexe 5	

Liste des Figures

Figure I-1- Structure de Base d'un PLD.....	5
Figure I-2- Physionomie d'un CPLD	9
Figure I-3- Physionomie d'un FPGA.....	12
Figure I-4- Structure générale d'un FPGA	12
Figure I-5- Routage d'un FPGA	13
Figure I- 6- Architecture d'une SPARTAN 3E.....	16
Figure I-7- Architecture d'un CLBs pour une SPARTAN 3E	17
Figure I-8- Disposition des CLBs dans un FPGA SPARTAN 3E	17
Figure I-9- Architecture d'une slice	18
Figure I-10-	19
Figure-I-11- Chaîne de développement des PLDs.....	
Figure-II-1 : Onduleur de tension triphasé alimentant une MSAP.....	33
Figure -II-2 : Stratégies de commande des onduleurs.....	36
Figure -II-3 : Principe de la MLI sinusoïdale.....	38
Figure -II-4- Tensions générées dans le plan (V_α , V_β) et définition des secteurs.....	40
Figure-II- 5- Algorithme de calcul de l'entier i.....	42
Figure-II-6- Principe de construction du vecteur de référence V_s	43
Figure-III-1- Schéma d'une machine synchrone autopilotée.....	48
Figure-III-2- Schéma structurel de la commande DTC appliquée à la MSAP.....	49
Figure-III-3- Schéma du principe de découplage pour la MSAP par analogie avec la MCC.....	51
Figure -III-4- Principe de la commande par orientation du flux.....	51
Figure -III-5- Schéma bloc d'une régulation de vitesse de la MSAP alimentée en tension et Commandée par 'orientation du flux.....	53
Figure-III-6- Schéma de principe de la commande sans capteur par injection d'un signal à haute fréquence.....	55
Figure-III-7- Représentation des vecteurs tournants du courant à haute fréquence.....	58
Figure-III-8- Schéma d'extraction des composantes inverses par un filtre analogique suivi par un filtre synchrone passe haut.....	59
Figure-III-9- Schéma d'extraction des composantes inverses par deux filtres synchrones passe haut.....	60

Figure-III-10 -Schéma d'extraction des composantes inverses par un filtre synchrone passe bas.....	61
Figure-III-11 -Estimation de la position du rotor par la fonction arctg.....	62
Figure-III-12 -Estimation de la position du rotor par la boucle de verrouillage de phase...	63
Figure VI-1 - Schéma de principe du dispositif expérimental.....	64
Figure-VI-2 -Schéma bloc du programme implanté sur le FPGA.....	65
Figure-VI-3 -Plateforme BASYS2.....	66
Figure-VI-4 - Sortie de l'oscillateur.....	68
Figure-VI-5 -Schéma de l'oscillateur.....	69
Figure-VI-6 -	70
Figure-VI-7 -Génération du signal PWM et du signal d'horloge de l'oscillateur.....	71
Figure-VI-8 -Forme du signal fondamental à 40Hz et du signal à haute fréquence généré par le code VHDL.....	77
Figure-VI-9 -Forme du signal modulé en largeur d'impulsion (PWM1) généré par le code VHDL.....	78
Figure-VI-10 -Forme du signal fondamental à 40Hz et du signal à haute fréquence généré par le code VHDL.....	78
Figure-VI-11 -Forme du signal modulé en largeur d'impulsion (PWM1) généré par le code VHDL.....	79

Liste des Tableaux

Tableau I-1- Tableau récapitulatif des différentes familles des PLDs.....	6
Tableau I-2- Tableau récapitulative des différentes familles de XILINX.....	15
Tableau-II-1- Tensions simples triphasées et diphasées	40
Tableau-II-2- Elaboration des séquences de commande.....	46

NOMENCLATURE

Sigles utilisés

PLD : circuits logiques programmables (programmable logic device).

EPLD : Erasable Programmable Logic Device.

CPLD : Complex programmable logic device.

FPGA : Field programmable Gate Array ou encore réseau de cellules logiques programmables.

CLB : Blocs logiques configurables.

IOB : Blocs d'entrées/sorties programmables.

LUT : Look-Up Table.

VHDL : Very High Speed Integrated Circuit Hardware Description Language.

MSAP : Moteur synchrone à aimant permanent

DTC : La commande directe du couple

SPWM : sinusoidal pulse with modulation (MLI sinusoidale).

SVM : space vector modulation (MLI vectorielle).

PI : Régulateur Proportionnel Intégrateur

P : Opérateur de Laplace

f : Fonction de connexion.

Repère

A, B, C : Axes liés aux enroulements triphasés.

d, q : Axes de référentiel de Park.

α , β : Axes de référentiel de concordia.

Grandeurs

U_{ab} , U_{bc} , U_{ca} : Tension composés entre deux phases statorique.

V_{as} , V_{bs} , V_{cs} : Tension statorique phase a, b, ou c.

V_{as}^* , V_{bs}^* et V_{cs}^* : Tension de phase référence.

U_{ao} , U_{bo} , U_{co} : Tension composée par rapport au point commun « O »

V_{dc}	: Tension du bus continu.
V_{ds}, V_{qs}	: Tension statorique sur l'axe d et l'axe q
I_{ds}, I_{qs}	: Tension statorique sur l'axe $\alpha \beta$,
I_a, I_b, I_c	: Courant statorique phase a, b et c
$I_{aref}, I_{bref}, I_{cref}$: Courants de phase de référence
I_{ds}, I_{qs}	: Courant statorique sur l'axe d et l'axe q.
I_{ds}^*, I_{qs}^*	: Courant de référence statorique sur l'axe d et l'axe q.
$I_{\alpha s}, I_{\beta s}$: Courant statorique sur l'axe $\beta\alpha$,
Ψ	: Déphasage du courant statorique par rapport à la F.E.M.
ϕ_{ds}, ϕ_{qs}	: Flux statorique suivant les axes d et q.
ϕ_f	: Flux crée par l'aimant permanent.
C_e	: Couple électromagnétique.
ω_r	: Vitesse de rotation électrique du rotor.
f_s	: Fréquence de l'alimentation fondamentale
f_r	: Fréquence de rotation du rotor
f_{HF}	: Fréquence du signal injecté
L	: Valeur moyenne des inductances statoriques.
ΔL	: Différence des inductances statoriques.
ϑ_r	: Position électrique du rotor.
$\hat{\vartheta}_r$: Position estimé du rotor.
v_{HF}	: Tension à haute fréquence injectée.
$V_{\alpha-HF}, V_{\beta-HF}$: valeurs de tension à haute fréquence suivant les axes α et β .
$I_{\alpha-HF}, I_{\beta-HF}$: valeurs de tension à haute fréquence suivant les axes α et β .
ϑ_H	: Angle entre la tension à haute fréquence injectée et l'axe α .

$i_{HF\alpha}, i_{HF\beta}$: Séquence positive et négative du courant à haute fréquence.

$I_{\alpha-nmHF}^{-\omega_H}, I_{\alpha-pnmHF}^{-\omega_H}$: Vecteur modèle suivant les axes α et β dans le repère $(-f_{HF})$.

ε : Erreur entre un signal de référence et un signal estimé

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Depuis l'avènement et le développement des matériaux de haute énergie tels que les aimants permanents, l'utilisation des machines synchrones à aimants permanents a connu un essor gigantesque. En effet, durant ces dernières années, un intérêt particulier et significatif leur est consacré du fait des nombreux avantages qu'elles présentent à savoir ; un rendement élevé, une construction robuste, un couple et une puissance massique importante.

La commande de ces machines nécessite la connaissance précise de la position et de la vitesse de l'arbre du rotor permettant leur autopilotage. Ce qui requière alors l'utilisation de capteur de position ou de vitesse placé sur l'arbre de la machine. Toutefois, ces résolveurs constituent un inconvénient majeur pour la machine, en effet, leurs emplois génèrent une augmentation non négligeable du coût et du volume, une diminution de sa fiabilité et une sensibilité aux perturbations électromagnétiques.

Tenant compte de toutes ces limites, de nombreuses études ont été faites pour supprimer ces capteurs mécaniques tout en conservant le bon fonctionnement de la machine. Ces études ont fait apparaître plusieurs méthodes de commande sans capteur. En effet, on peut distinguer dans la littérature trois techniques de base pour l'estimation de la position du rotor :

- Méthode basée sur la saturation locale du circuit magnétique.
- Méthode utilisant un observateur d'état.
- Méthode fondée sur l'estimation de la F.E.M.

La première exploite la variation des inductances statoriques due à la saturation locale du circuit magnétique créée par les aimants.

La seconde est basée sur l'utilisation d'observateurs à état qui sont capables de reconstruire des grandeurs non mesurées à partir d'un modèle dynamique du système et des mesures des grandeurs d'entrée et de sortie.

La dernière est basée sur l'estimation de la F.E.M. et elle est la seule grandeur électrique capable de fournir des informations instantanées sur les variables mécaniques à savoir la vitesse et la position. Cependant, cette méthode basée sur l'observation de la force électromotrice n'est pas adaptée à basse vitesse en raison de la proportionnalité de la F.E.M à la vitesse de rotation, en effet, il devient impossible d'extraire sa forme d'onde à basse vitesse car elle est noyée dans le bruit de mesure (la F.E.M est très petite).

Afin de s'affranchir de ce problème, plusieurs approches ont été proposées et parmi elles on note la commande sans capteur avec injection d'un signal à haute fréquence. C'est sur cette dernière que portera notre travail.

L'idée fondamentale de la commande sans capteur existait depuis plusieurs années, mais l'implantation de cette méthode avec des composants analogiques a limité son utilisation. Cependant, avec l'arrivée des solutions numériques, ce type de commande a subi des progrès significatifs [1]. En effet, l'utilisation de telles solutions a permis de lever les problèmes liés à l'utilisation des commandes analogiques et en plus, elles présentent un grand intérêt économique et une meilleure flexibilité de conception. Ces solutions peuvent être logicielles ou matérielles [1]. Dans le premier cas où elles sont à base de microprocesseur et de DSP (Digital Signal Processor), les conceptions qui leur sont associées sont conçues en utilisant des architectures prédéfinies, le concepteur ne peut agir sur la partie matérielle de la conception et n'a accès qu'à la partie logicielle. Quand au deuxième cas où elles sont à base de FPGA et d'ASIC, elles n'ont pas d'architecture prédéfinie ; c'est le concepteur lui-même qui assure sa conception. Ce degré de liberté permet de réaliser des architectures optimisées et adaptées au fonctionnement souhaité de la commande. Et c'est pour cette dernière solution qu'on optera pour la suite de notre travail.

L'objectif de ce travail est la réalisation du circuit de commande MLI adapté à la commande sans capteur de la MSAP et son implantation sur un FPGA (Field Programmable Gate Array).

Dans ce contexte, ce mémoire est scindé en 4 chapitres principaux :

- Le premier chapitre est divisé en trois parties : La première sera consacrée à des généralités sur les circuits logiques programmables où on s'intéressera plus particulièrement au FPGA, elle sera suivie d'une présentation détaillée du langage

VHDL et on clôturera ce chapitre par donner un aperçu sur le logiciel de développement ISE.

- Le deuxième portera sur les différentes stratégies MLI destinées à la commande des onduleurs de tension où on s'intéressera plus précisément à la MLI sinusoïdal et à la MLI vectoriel.
- Le troisième abordera l'analyse du besoin d'une commande MLI adaptée à la commande sans capteur de la MSAP.
- Enfin, dans le dernier chapitre on s'étalera sur la réalisation et l'implémentation sur un FPGA du circuit MLI de la commande sans capteur avec injection d'un signal à haute fréquence.

CHAPITRE I

**Apprentissage-approfondissement des circuits logiques
programmables, du langage VHDL et de la chaîne de
développement ISE-Weblink**

CHAPITRE I Apprentissage-approfondissement des circuits logiques programmables, du langage VHDL et de la chaîne de développement ISE-Weblink.

II-4- Les différentes familles de PLD

Vu la bataille commerciale féroce que se livre les constructeurs spécialisés pour emporter les parts de marché et les appellations choisies par certains d'entre eux qui ne sont pas toujours d'une logique rigoureuse, la classification des PLDs est un peu délicate et difficile. Néanmoins, on peut les classer suivant leurs structures internes à savoir ; le nombre d'entrées, de sorties, de connexions programmables et le niveau d'intégration [3]. Le tableau suivant représente certaines de ces familles :

Type	Nombre de porte intégré	Matrice ET	Matrice OU	Effaçable
PAL	10 à 100	Programmable	Fixe	Non
GAL	10 à 100	Programmable	Fixe	Electriquement
EPLD	100 à 3000	Programmable	Fixe	U-V
FPLA	2000à 3000	Programmable	Programmable	Electriquement
FPGA	Plus de 50.000	Programmable	Programmable	Electriquement

Tableau I-1- Tableau récapitulatif des différentes familles des PLDs [2].

II-4-1- Les PALs

Ce sont les premiers circuits programmables à être utilisés pour réaliser des fonctions logiques. Ce type de circuits a été développé, il y a de cela près de 20 ans, par le constructeur AMD. Ils possèdent des matrices « ET » programmables et des matrices « OU » fixes. La programmation de ces circuits s'effectue par destruction de fusibles et cela est obtenu par l'application, à leurs bornes, d'une tension de 11.5 V pendant 10 à 50 μ s [3]. Ces circuits sont caractérisés par le fait qu'une fois programmés ils ne peuvent plus être effacés.

Dans cette famille on peut distinguer quatre sous familles :

- Les PALs combinatoires ou PALs simples : Ils ne contiennent que des portes d'où l'appellation combinatoires.
- Les PALs à registres ou FLPS (Field programmable logic sequencer) : Ils sont constitués de logique combinatoire et séquentielle (registre).
- Les PALs asynchrones à registres : Ce type de PALs est une variante du dernier type évoqué. La première différence réside dans la méthode de distribution du signal d'horloge des bascules, en effet, dans ce type, chaque bascule dispose de son propre signal d'horloge (voir une combinaison de signaux) issu d'un terme de produit. Ces

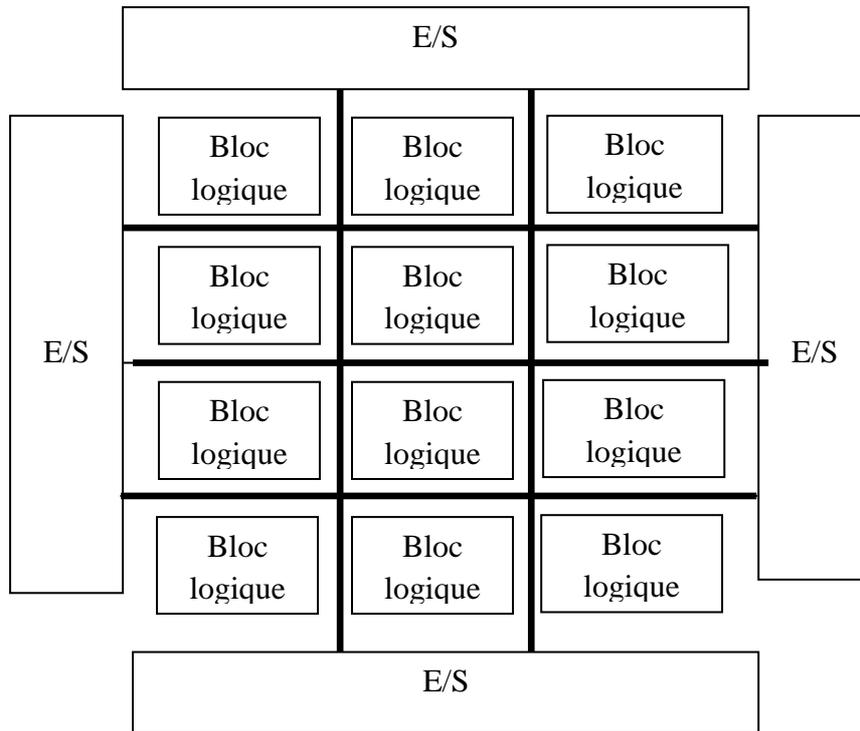


Figure I-3- Physionomie d'un FPGA [2].

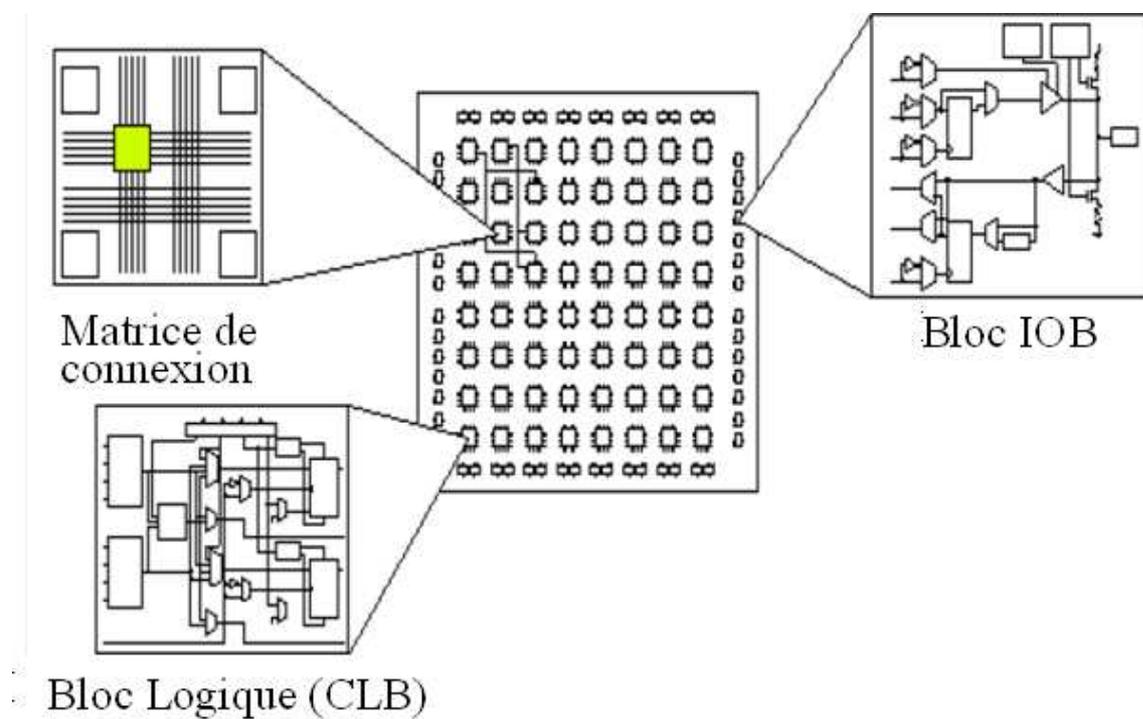


Figure I-4- Structure générale d'un FPGA [3].

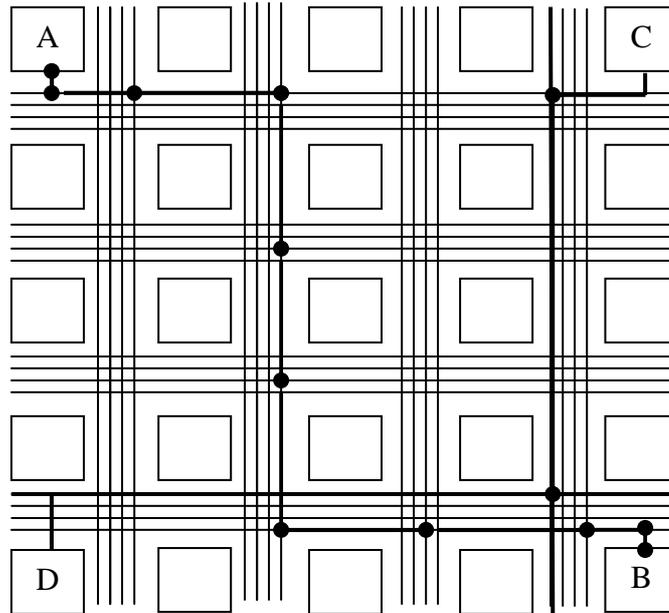


Figure I-5- Routage d'un FPGA [2].

II-5-2- Classification des FPGAs

Comme cité dans le paragraphe précédent, on distingue deux classes de FPGAs selon la technique de réalisation des PIP (points de connexion) :

II-5-2-1- FPGAs à SRAM

La structure de base d'un FPGA de type SRAM est très complexe. Le point de connexion entre les différentes cellules est un ensemble de transistors MOS de commutation commandés par des cellules de mémoire vive (RAM) [2].

Le FPGA à SRAM peut être considéré comme étant constitué de deux couches superposées :

- Une couche comportant toutes les ressources du FPGA (blocs d'entrées-sorties, CLB, réseaux de connexion).
- Une couche comportant la commande de chaque point de connexion. Le point de commande est une cellule de type mémoire vive (SRAM) et qui peut être assimilée à une bascule D. L'ensemble des points de commande est un grand registre à décalage.

Afin d'établir les connexions souhaitées et obtenir le système programmé, il suffit de charger le registre à décalage.

- Digital Clock Manager (DCM) Blocks

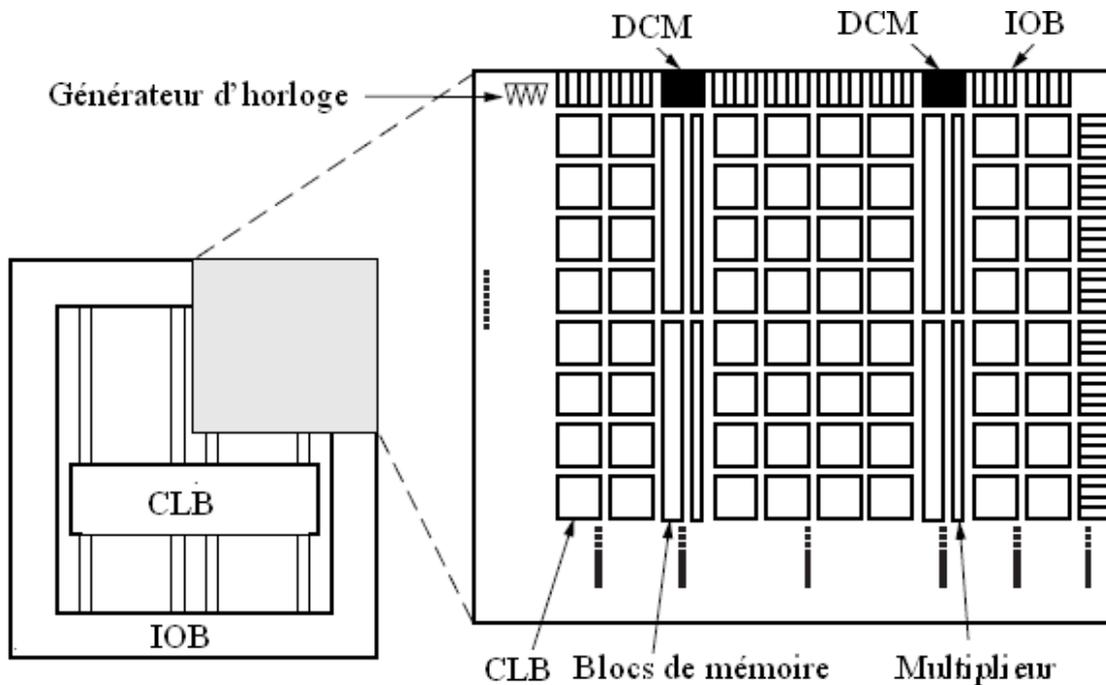


Figure I- 6- Architecture d'une SPARTAN 3E [7]

II-5-3-2-1-Structure d'un bloc CLB

Les blocs logiques programmables (CLBs) constituent les principales ressources logiques pour la mise en œuvre des circuits combinatoires et synchrones. Chaque CLB est composé de quatre cellules logiques (Slices) réparties en tranches identiques de deux slices chacune (paires) avec deux retenues indépendantes et une chaîne de décalage commune [7]. Ils sont disposés en un réseau régulier de lignes et de colonnes, comme le montre la figure (I-8).

Chaque slice contient essentiellement [7] :

- Un générateur de fonction à quatre entrées réalisé à l'aide d'une LUT (Look Up Table) à quatre entrées, d'une mémoire RAM à 16 bits ou d'un registre à décalage à 16 bits.
- Deux multiplexeurs F5MUX, FiMUX.
- Deux éléments de mémoire et de la logique arithmétique.

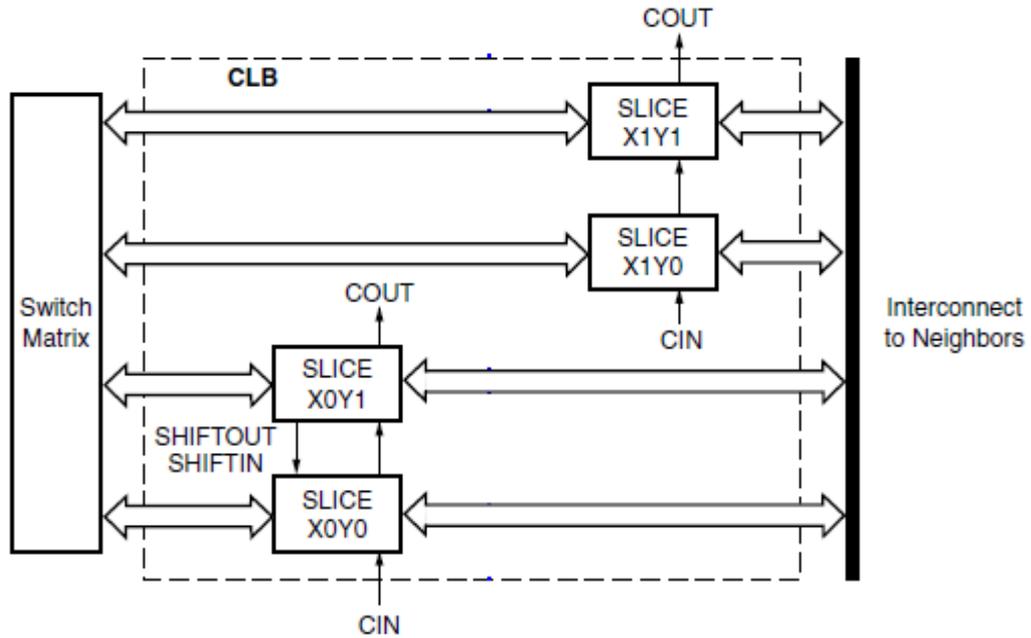


Figure I-7- Architecture d'un CLBs pour une SPARTAN 3E [7].

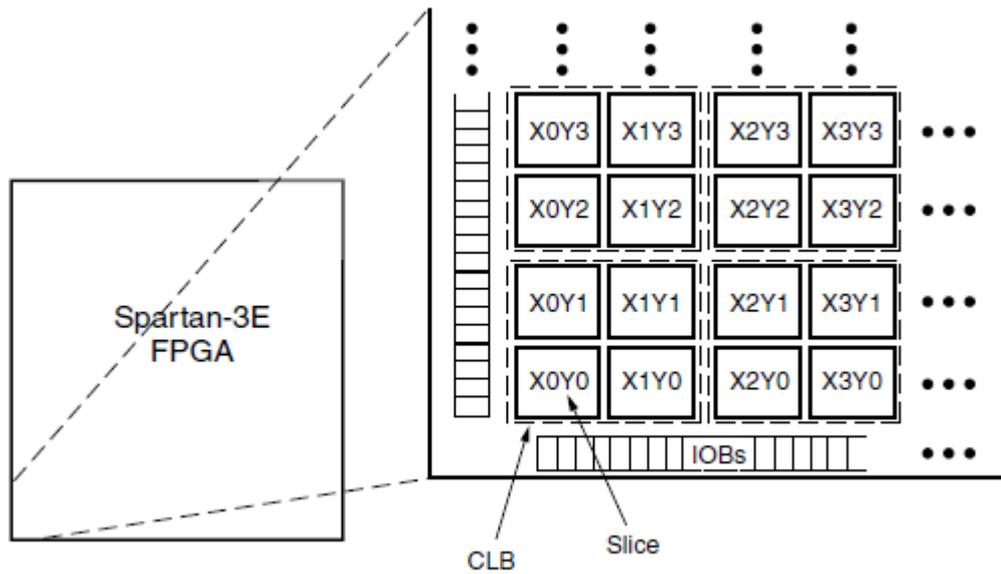


Figure I-8- Disposition des CLBs dans un FPGA SPARTAN 3E [7]

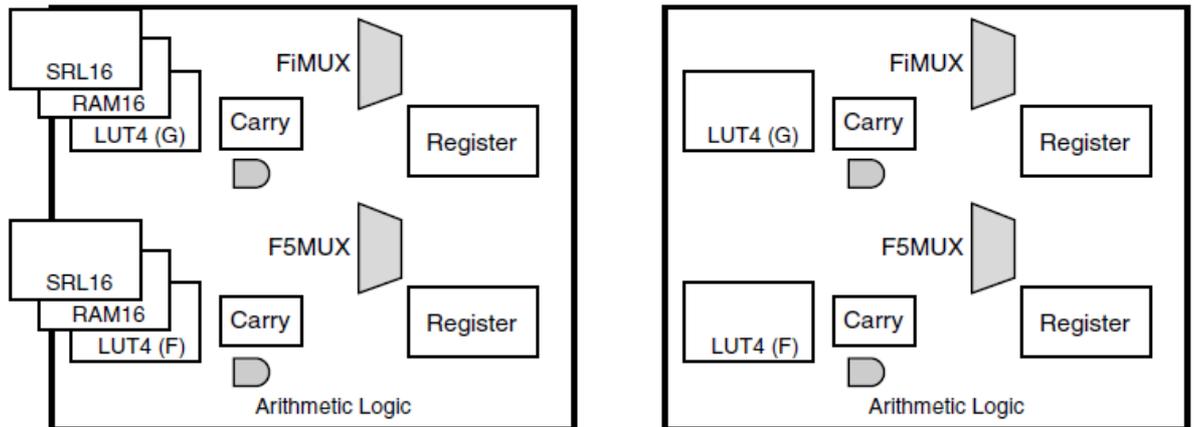


Figure I-9- Architecture d'une slice [7].

II-5-3-2-2- Les IOBs (Input/ Output Bloc)

Ils constituent l'interface entre les bornes du circuit et les CLB. Ils sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé. Chaque deux IOBs peuvent être utilisés comme une paire différentielle toujours connectée à la matrice de commutation, comme le montre la figure ci-dessous.

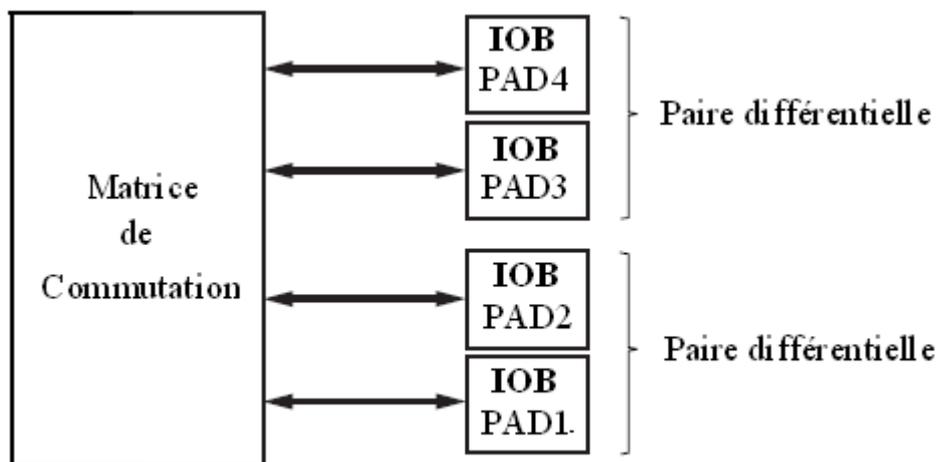


Figure I-10- Disposition des IOBs.

II-5-3-2-3- Les multiplieurs

Ce sont des multiplicateurs à chiffre signé. Chaque entrée est sur 18 bits et la sortie sur 32 bits.

Le circuit SPARTAN 3E possède aussi deux mémoires organisées en blocs de 4096 bits, situées de part et d’autres de la matrice CLB. Il offre également un bloc DCM qui permet de contrôler les décalages du signal d’horloge à l’intérieur du FPGA.

III- Chaîne de développement

Le processus de développement d’une application utilisant des circuits logiques programmables passe par un certain nombre d’étapes allant de la description de l’application à la programmation du composant. Cette chaîne de développement représente l’ensemble des outils logiciels nécessaires pour la réalisation des différentes étapes. Elle est représentée comme suit [2] :

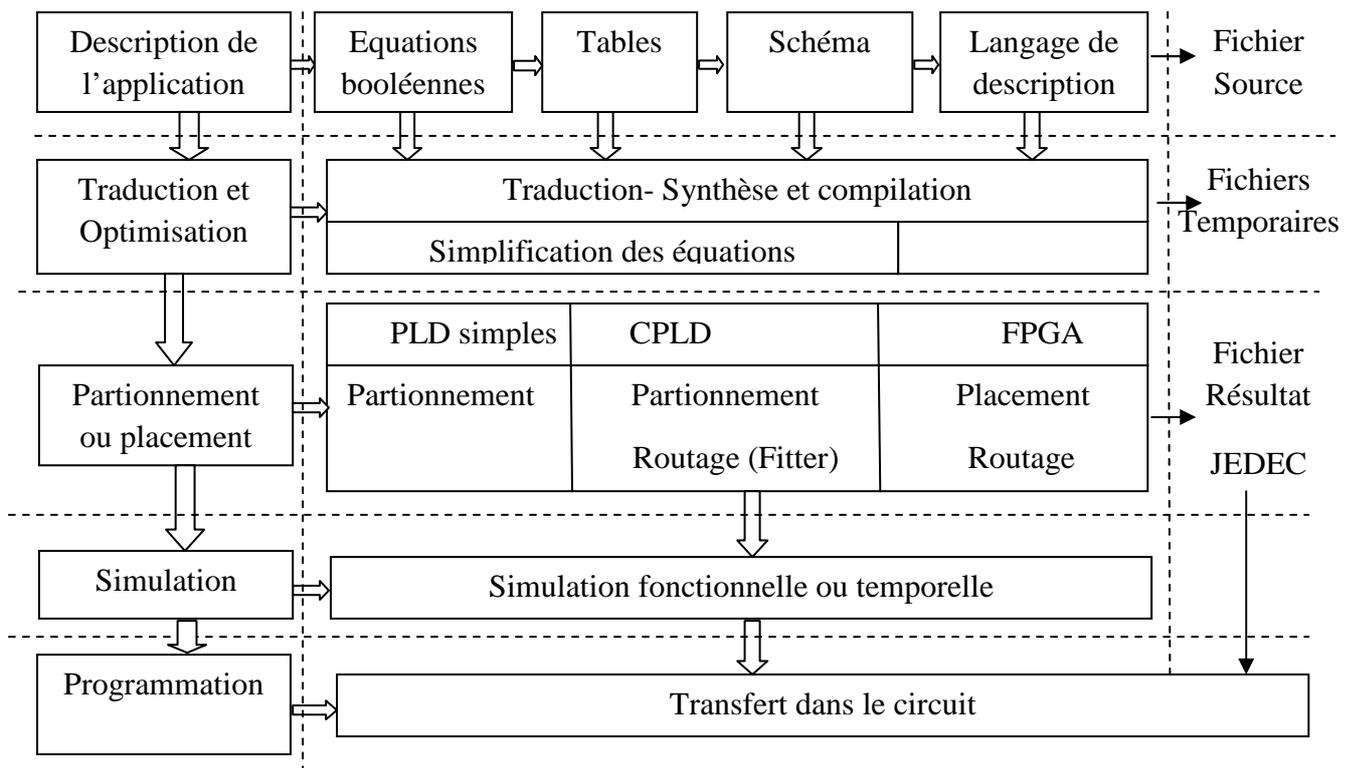


Figure-I-11- Chaîne de développement des PLDs [2]

CHAPITRE I Apprentissage-approfondissement des circuits logiques programmables, du langage VHDL et de la chaîne de développement ISE-Weblink.

Le mode peut être :

IN : Entrée, **OUT** : Sortie, **INTOUT** : Bidirectionnel (Entrée/Sortie), **Buffer**.

Le type peut être :

Boolean : Variable vrai ou faux, **Bit** : 0 ou 1, **Bit_vector** : Tableau de bits, **Integer**.

VI-3-2- Architecture

L'architecture décrit la vue interne du modèle ; elle consiste à définir les fonctionnalités du composant à réaliser. Ce fonctionnement peut être décrit de différentes façons ; structurelle, comportementale ou un mélange des deux (mixte). Comme pour l'étape « Entité », la syntaxe de l'architecture interne est comprise au sein d'un bloc encadré par les mots clés « ARCHITECTURE » et « END ». Le corps de l'architecture est inclus entre les deux mots clés « BEGIN » et « END ». La syntaxe générale d'une architecture est représentée comme suit :

ARCHITECTURE <Nom de l'architecture> **OF** <Nom de l'entité> **IS**

<Zone de déclaration optionnelle : type, constant, signaux locaux, composants>

BEGIN

<Description de la structure logique> < Instruction>

END Nom de l'architecture ;

IV-3-2-1-Description comportementale

Ce type de description spécifie le comportement du composant ou du circuit à réaliser au moyen d'instruction séquentielle ou sous forme de flow de données (DATAFLOW) constituant un process.

➤ **Sous forme de flow de données :**

Dans ce type de description comportementale, on modélise le circuit par un ensemble d'équations logiques et arithmétiques; c'est-à-dire on décrit chaque sortie par une équation en fonction des entrées.

CHAPITRE II

**Principe des techniques de modulation de largeur
d'impulsion**

I- Introduction

L'alimentation des machines électriques par convertisseurs statiques est très largement utilisée de nos jours surtout dans les systèmes d'entraînement industriel. En effet, l'évolution en matière de semi-conducteurs et dans les techniques numériques a permis la réalisation de convertisseurs robustes et de plus en plus performants, ainsi que l'exécution en temps réel des algorithmes complexes de contrôle de ces convertisseurs [8, 9]. Dans notre travail, on s'intéressera aux stratégies de commande des onduleurs, notamment aux techniques de modulation de largeur d'impulsion.

II- Modélisation de l'onduleur**II-1- Définition de l'onduleur**

Un onduleur est un convertisseur statique assurant la conversion continu-alternatif. Alimenté en continu et grâce à des composants semi-conducteurs, on relie chacune des bornes du récepteur tantôt à l'une, tantôt à l'autre des bornes d'entrée ; ce qui permet d'obtenir entre les bornes du récepteur une tension tantôt positive, tantôt négative. Pour une séquence adéquate de commande des semi-conducteurs, on obtient à la sortie du convertisseur une tension alternative de valeur moyenne nulle [10, 12, 13, 14].

On distingue plusieurs types d'onduleurs :

- **Selon la nature de la source :**
 - ❖ Onduleur de tension.
 - ❖ Commutateur de courant.
- **Selon le nombre de phases ;** monophasé, triphasé...etc.
- **Selon le nombre de niveau ;** 2 .3.....etc.

L'onduleur de tension est un convertisseur assurant la liaison entre une source de tension continue et une charge en courant alternatif. Il est formé de trois bras indépendant comportant chacun deux interrupteurs. Chaque interrupteur est constitué d'un composant semi-conducteur contrôlable et d'une diode en antiparallèle. Pour assurer la continuité des courants alternatifs et éviter le court-circuit de la source continue (bus continue V_{dc}), les interrupteurs d'un même bras doivent être contrôlés de manière complémentaire [10] [13, 14, 15].

Le schéma structurel d'un tel convertisseur statique alimentant le stator d'une MSAP est illustré par la figure (II-1).

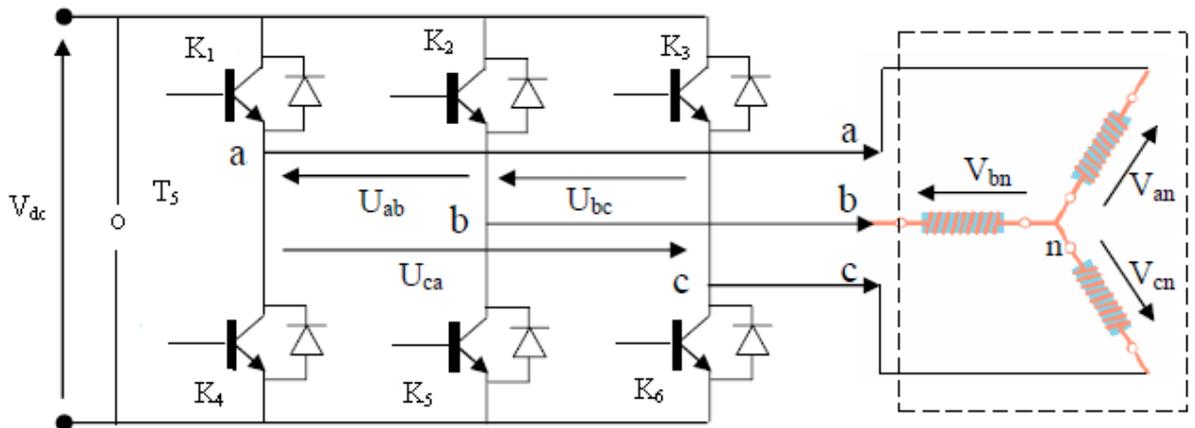


Figure-II-1-Onduleur de tension triphasée alimentant une MSAP.

Dans l'étude de la structure de l'onduleur, on utilisera la notion de la fonction de connexion qui explicite la nature des connexions réalisées par des interrupteurs entre deux points d'un circuit. Tout interrupteur idéalisé ou supposé parfait, peut être caractérisé par la fonction de connexion f tel que :

$$f = \begin{cases} 1 & \text{si l'interrupteur est passant} \\ 0 & \text{si l'interrupteur est ouvert} \end{cases}$$

En se basant sur la figure (II. 1) représentant la structure de l'onduleur de tension, les tensions composées délivrées par ce dernier sont données comme suit :

$$U_{ab} = V_{as} - V_{bs} \tag{II.1}$$

$$U_{bc} = V_{bs} - V_{cs} \tag{II.2}$$

$$U_{ca} = V_{cs} - V_{as} \tag{II.3}$$

Étant donné que le récepteur de sortie est équilibré, on peut passer des tensions simples aux tensions composées.

Les tensions simples forment un système de tensions triphasées équilibrées, alors :

$$V_{as} + V_{bs} + V_{cs} = 0 \tag{II.4}$$

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

$$V_{as} = \frac{1}{3}(U_{ab} - U_{ca}) \tag{II.5}$$

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

$$V_{bs} = \frac{1}{3}(U_{bc} - U_{ab}) \quad \text{II.6}$$

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

$$V_{cs} = \frac{1}{3}(U_{ca} - U_{bc}) \quad \text{II.7}$$

Les tensions simples peuvent aussi être données en fonction des tensions prises par rapport au point commun « O » :

$$V_{as} = \frac{1}{3}(2U_{ao} - U_{bo} - U_{co}) \quad \text{II.8}$$

$$V_{bs} = \frac{1}{3}(2U_{bo} - U_{ao} - U_{co}) \quad \text{II.9}$$

$$V_{cs} = \frac{1}{3}(2U_{co} - U_{ao} - U_{bo}) \quad \text{II.10}$$

Ces tensions peuvent être écrites sous la forme matricielle suivante :

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} U_{ao} \\ U_{bo} \\ U_{co} \end{bmatrix} \quad \text{II.11}$$

En utilisant la notion de fonction de connexion, le système (II.11) deviendra :

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad \text{II. 12}$$

Tel que :

$$U_{ao} = \begin{cases} V_{dc} & \text{si } K_1 \text{ conduit} \\ 0 & \text{si } K_4 \text{ conduit} \end{cases} \quad \text{donc : } U_{ao} = f_1 V_{dc}$$

$$U_{bo} = \begin{cases} V_{dc} & \text{si } K_2 \text{ conduit} \\ 0 & \text{si } K_5 \text{ conduit} \end{cases} \quad \text{donc : } U_{bo} = f_2 V_{dc}$$

$$U_{co} = \begin{cases} V_{dc} & \text{si } K_3 \text{ conduit} \\ 0 & \text{si } K_6 \text{ conduit} \end{cases} \quad \text{donc : } U_{co} = f_3 V_{dc}$$

Avec : f_1, f_2, f_3 ; fonctions de connexion déterminant l'état des interrupteurs.

$$f_1 = \begin{cases} 1 & \text{si } K_1 \text{ passant} \\ 0 & \text{si } K_1 \text{ ouvert} \end{cases}$$

$$f_2 = \begin{cases} 1 & \text{si } K_2 \text{ passant} \\ 0 & \text{si } K_2 \text{ ouvert} \end{cases}$$

$$f_3 = \begin{cases} 1 & \text{si } K_3 \text{ passant} \\ 0 & \text{si } K_3 \text{ ouvert} \end{cases}$$

La détermination des fonctions f_i dépend de la stratégie de commande adoptée.

III-- Stratégie de commande des onduleurs

III-1- Introduction

Plusieurs stratégies peuvent être considérées pour la commande d'un onduleur de tension alimentant une machine dans le cadre d'une application industrielle : les commandes en régime permanent et les commandes en régime transitoire [1, 11]. Dans le cas des commandes en régime permanent, les critères d'optimisation électrotechnique dominant : élimination d'harmoniques pour éviter les ondulations de couples dans les machines, minimisation des pertes dans les interrupteurs des onduleurs. Et dans le cas des commandes en régime transitoire, les critères de type automatiques dominant : rapidité, temps de réponse. Toutefois, il est souvent très difficile de répondre parfaitement et simultanément aux deux

types de critères. Ces stratégies de commande peuvent être regroupées par le diagramme suivant [13, 14] :

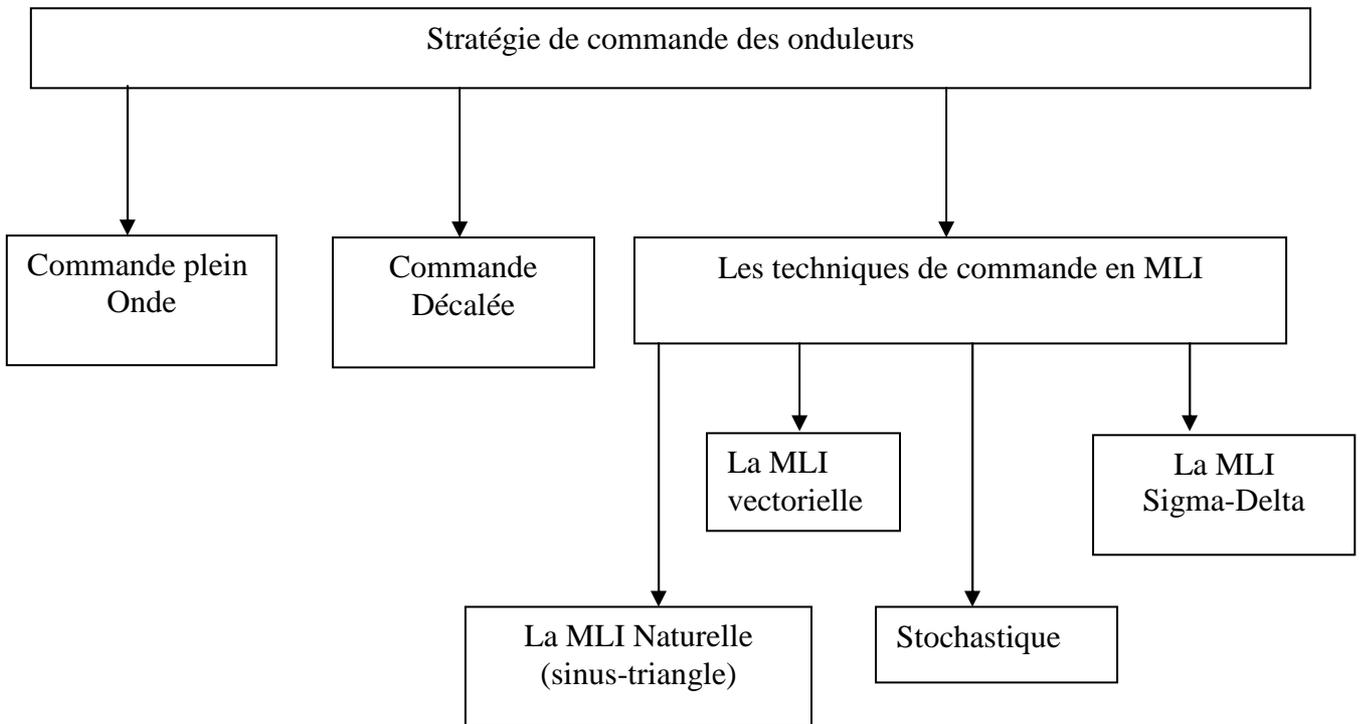


Figure -II-2- Stratégies de commande des onduleurs

Dans la suite de notre travail, on s'intéressera uniquement à deux variantes de la commande MLI ; à savoir la commande MLI sinusoïdale (SPWM, sinusoïdal pulse with modulation) et la commande MLI vectorielle (SVM, space vector modulation) qui peuvent être appliquées aussi bien en régime permanent qu'en régime transitoire.

III-2-Commande MLI Naturelle ou Sinusoïdale

III-2-1-Introduction :

Les onduleurs à une seule alternance ou à commande plein onde délivrent à leurs sorties des tensions riches en harmonique et leur filtrage est difficile et onéreux car le premier harmonique à éliminer (harmonique 3 ou 5) a une fréquence trop voisine de celle du fondamental. Afin de remédier à cela, on effectue la modulation par largeur d'impulsion qui consiste à former chaque alternance de tension ou de courant de sortie de plusieurs créneaux rectangulaires de largeurs convenables [14, 15, 16]. Cette multiplication du nombre d'impulsions formant chaque alternance offre la possibilité de moduler la forme de cette

tension de manière à avoir la forme sinusoïdale possible et elle présente deux avantages [10] [11, 12] [15] :

- La première est qu'elle repousse vers les fréquences les plus élevées les harmoniques de tension et de courant de sortie, ce qui facilite le filtrage.
- La seconde est qu'elle permet de faire varier la valeur du fondamental de la tension ou du courant.

Dans ce type de commande, les schémas des onduleurs de tension ou de courant restent les mêmes. C'est la commande des interrupteurs qui est modifiée ; la fréquence de commutation est supérieure à la fréquence des grandeurs de sortie [10, 11].

III-2-2-Définition de la commande MLI Naturelle :

La commande MLI consiste à découper la tension ou le courant de sortie en plusieurs créneaux de largeurs différentes par action sur les commutateurs de l'onduleur. Ces commutations sont déterminées par l'intersection entre deux signaux; le premier est l'onde de référence ou modulante représentant la tension de sortie désirée de fréquence f et le deuxième qui est un signal de haute fréquence par rapport au signal de référence est l'onde de modulation (porteuse) qui définit la cadence de la commutation des interrupteurs statiques de l'onduleur [10, 13, 14, 16].

Les coïncidences entre la référence et la porteuse déterminent les instants de commutation des sorties MLI. Lorsque le signal de référence est au dessus du signal de la porteuse, l'impulsion de sortie est 1; lorsqu'il est au dessous de la porteuse, l'impulsion de sortie est égale à 0.

Cette comparaison fournit trois signaux logiques f_1 , f_2 et f_3 qui valent 1 quand les interrupteurs du côté haut sont en conduction et ceux du côté bas sont bloqués et valent 0 dans le cas contraire.

La figure (II-3), illustre le principe de cette commande :

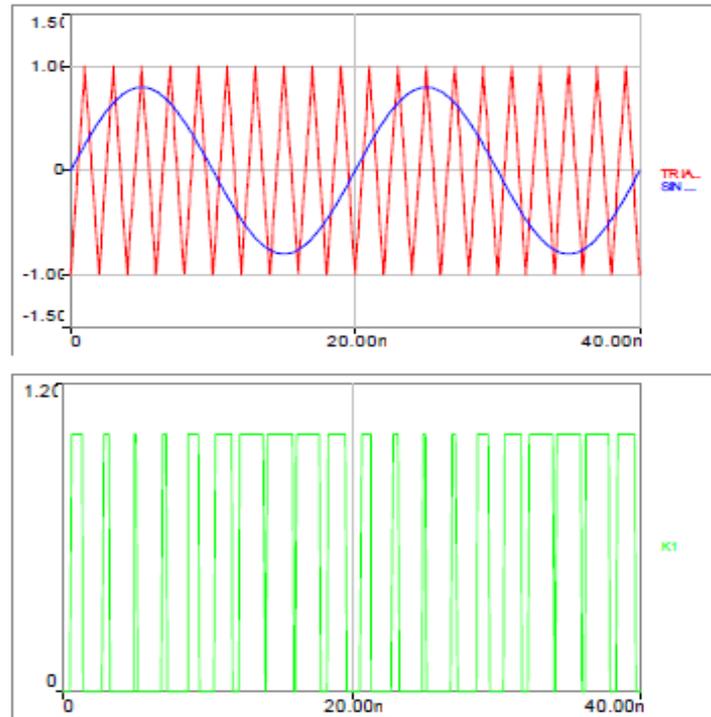


Figure -II-3 : Principe de la MLI sinusoidale

III-2-3- Caractéristique de la commande MLI sinusoidale

La commande MLI est caractérisée par plusieurs paramètres qui sont [13, 14] [16] :

- L'indice de modulation m qui est défini comme étant le rapport entre la fréquence de la porteuse (f') et la fréquence du signal de référence (f) :

$$m = \frac{f'}{f}$$

La valeur de m peut être un entier comme elle peut être aussi un non entier. Dans le premier cas, la fréquence de la porteuse est un multiple de la fréquence de l'onde de référence f et cette modulation est dite synchrone. Par contre dans le 2ème cas, la modulation est dite asynchrone.

Une valeur impaire de m permet d'obtenir pour la tension de sortie une alternance négative identique, au signe près, à son alternance positive, ce qui élimine les harmoniques pairs dans la tension de sortie [10, 12].

- Le coefficient de réglage de la tension r qui est défini comme étant le rapport entre l'amplitude de référence et la valeur crête de l'onde de modulation :

$$r = \frac{A_{ref}}{A_p}$$

- Le calage qui est dit optimal lorsque la position relative de la référence et de la modulante rend chaque alternance de la tension de sortie symétrique par rapport à son milieu.

III-3- MLI vectorielle

La modulation vectorielle est une variante de la commande par modulation de largeur d'impulsion et c'est une modulation en temps réel [10] [17]. Elle a pour principe la construction vectorielle de la référence de tension à partir des différents instants de commutation et en fonction des différentes possibilités de tension que peut générer un onduleur [12] [17, 18]. Afin de mettre en œuvre la stratégie de commande vectorielle et de simplifier les calculs, nous allons utiliser le repère (α, β) .

En utilisant la transformer de Concordia, on peut passer d'un système triphasé à un système biphasé (α, β) à l'aide de la matrice de Concordia donnée comme suit :

$$[C] = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}$$

Un onduleur triphasé possède six cellules de commutation donnant huit états de commutations possibles caractérisées par les fonctions de connexion (f_1, f_2 et f_3) [10] [13, 14] [17]. Ces huit états de commutation peuvent s'exprimer dans le plan (α, β) par 8 vecteurs de tensions (notés \bar{V}_0 à \bar{V}_7). Ceci est résumé dans le tableau suivant :

Vecteur d'espace	Etats des grandeurs booléennes de commande			Tensions simples triphasées			Tensions Diphasés	
	f_1	f_2	f_3	V_a	V_b	V_c	V_α	V_β
$(V)_i$	f_1	f_2	f_3	V_a	V_b	V_c	V_α	V_β
$(V)_0$	0	0	0	0	0	0	0	0
$(V)_1$	1	0	0	$\frac{.2V_{dc}}{3}$	$\frac{-.V_{dc}}{3}$	$\frac{-.V_{dc}}{3}$	$\sqrt{\frac{2}{3}} \cdot V_{dc}$	0
$(V)_2$	1	1	0	$\frac{.V_{dc}}{3}$	$\frac{.V_{dc}}{3}$	$\frac{-.2V_{dc}}{3}$	$\frac{1}{\sqrt{6}} \cdot V_{dc}$	$\frac{1}{\sqrt{2}} \cdot V_{dc}$

(V) ₃	0	1	0	$\frac{-V_{dc}}{3}$	$\frac{.2V_{dc}}{3}$	$\frac{-V_{dc}}{3}$	$-\frac{1}{\sqrt{6}}.V_{dc}$	$\frac{1}{\sqrt{2}}.V_{dc}$
(V) ₄	0	1	1	$\frac{-.2V_{dc}}{3}$	$\frac{.V_{dc}}{3}$	$\frac{.V_{dc}}{3}$	$-\sqrt{\frac{2}{3}}.V_{dc}$	0
(V) ₅	0	0	1	$\frac{-V_{dc}}{3}$	$\frac{-.V_{dc}}{3}$	$\frac{.2V_{dc}}{3}$	$\frac{1}{\sqrt{6}}.V_{dc}$	$-\frac{1}{\sqrt{2}}.V_{dc}$
(V) ₆	1	0	1	$\frac{.V_{dc}}{3}$	$\frac{-.2V_{dc}}{3}$	$\frac{.V_{dc}}{3}$	$-\frac{1}{\sqrt{6}}.V_{dc}$	$-\frac{1}{\sqrt{2}}.V_{dc}$
(V) ₇	1	1	1	0	0	0	0	0

Tableau-II-1 : Tensions simples triphasées et diphasées

Si on reporte ces tensions sur le plan (α , β) de Concordia, on obtient le schéma ci-dessous [10] [11] [17] [19, 20]:

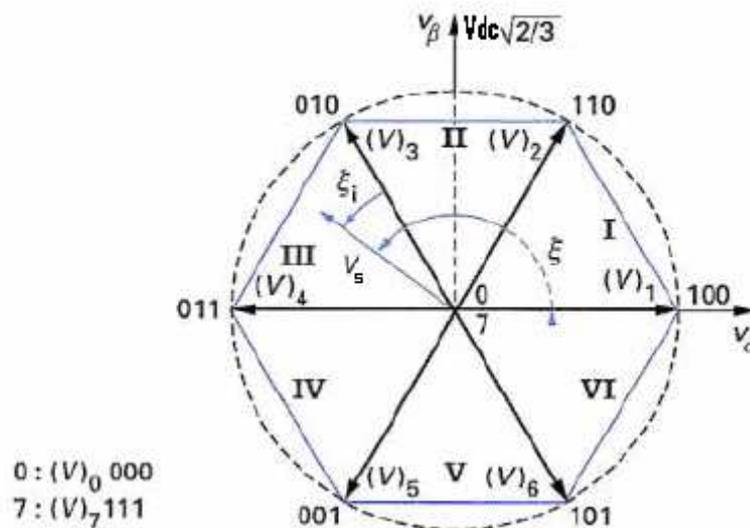


Figure -II-4-Tensions générées dans le plan (V_α , V_β) et définition des secteurs [11]

D'après le figure (II.4), on constate que parmi les huit vecteurs de tension générés, deux sont nuls et les autres sont équi-réparties tous les 60°.

Ces vecteurs sont appelés vecteurs d'espace et ils sont donnés par la relation suivante :

$$(V)_i = V_{dc} \cdot \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \cos[\frac{\pi}{3}(i-1)] \\ \sin[\frac{\pi}{3}(i-1)] \end{bmatrix} \quad \text{II. 13}$$

Avec : $i=1.....6$.

Le principe de la MLI vectorielle consiste à projeter le vecteur V_s désiré sur les deux vecteurs de tension adjacents correspondant à deux états de commutation de l'onduleur pendant une période d'échantillonnage T_e [11] [15] [21]:

Le fonctionnement de la MLI vectorielle se résume par les séquences de traitement suivantes : [10] [11]

- Acquisition d'une consigne V_s pour la période d'échantillonnage courante.
- Détermination du secteur i contenant le vecteur V_s .
- Décomposition du vecteur V_s suivant les deux vecteurs $(V)_i$ et $(V)_{i+1}$ délimitant le secteur i .
- Elaboration des séquences de commande des interrupteurs de l'onduleur.

III-3-1-Acquisition de la consigne V_s

Cette étape consiste à déterminer le vecteur de tension de référence $V_s = \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix}$. Cela s'obtient par application de la transformateur de Concordia à la tension de sortie triphasée souhaitée.

$$\begin{bmatrix} V_\alpha \\ V_\beta \\ V_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \sqrt{\frac{3}{2}} & -\sqrt{\frac{3}{2}} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad \text{II. 14}$$

III-3-2-Détermination du secteur

Le vecteur de référence V_s peut s'écrire sous forme polaire, de module V et d'argument ξ :

$$V_s = \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = V \cdot \begin{bmatrix} \cos \xi \\ \sin \xi \end{bmatrix} \quad \text{II. 15}$$

Afin de déterminer le secteur auquel appartient le vecteur de référence V_s , nous devons déterminer l'entier i appartenant à l'intervalle [1 6] et l'angle ξ_i appartenant à l'intervalle $[0 \frac{\pi}{3}]$.

Tel que :

$$\xi = \frac{\pi}{3}(i-1) + \xi_i \quad \text{II. 16}$$

Ce qui nous donne :

$$V_s = \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = V \cdot \begin{bmatrix} \cos[\frac{\pi}{3}(i-1) + \xi_i] \\ \sin[\frac{\pi}{3}(i-1) + \xi_i] \end{bmatrix} \quad \text{II. 17}$$

L'algorithme pour déterminer l'entier « i » est donné comme suit :

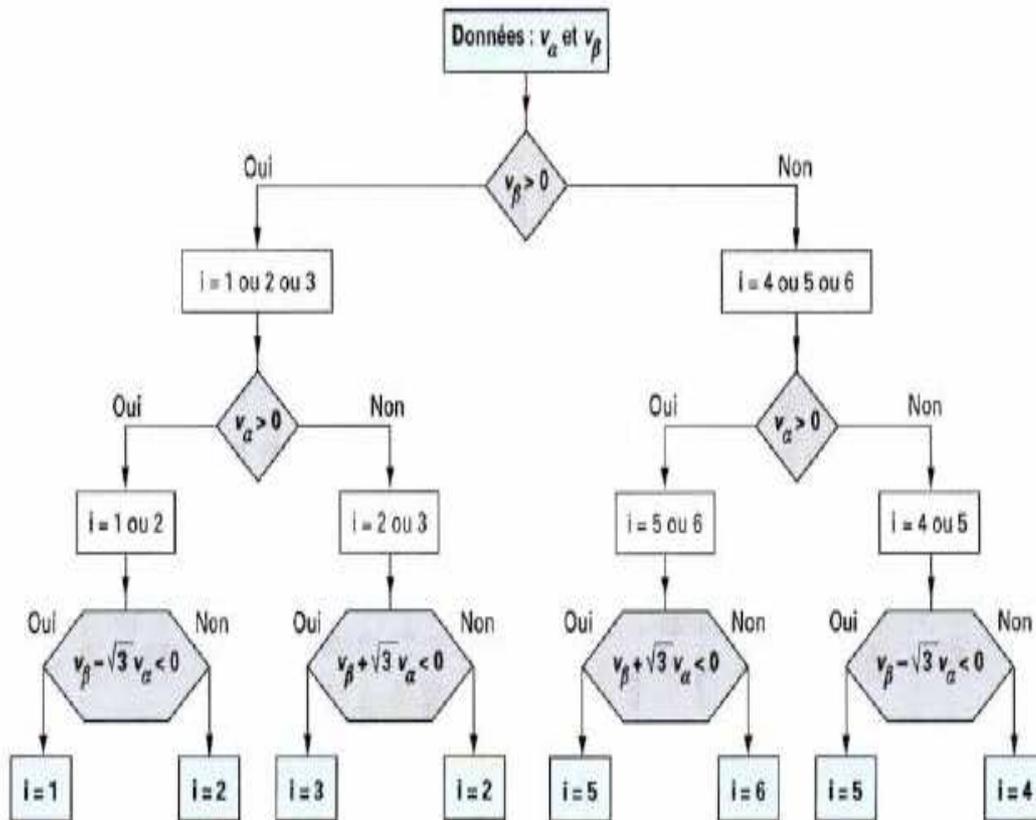


Figure-II- 5- Algorithme de calcul de l'entier i [11].

III-3-3-Décomposition du vecteur de tension de référence

L'onduleur ne peut fournir de façon exacte et instantané que des tensions de type (V)_i et on ne peut réaliser une tension quelconque qu'en valeur moyenne et sur une période d'échantillonnage T_e [10][15].

Puisque V_s ne s'obtient qu'en valeur moyenne, on doit appliquer des valeurs réalisables (V)_i pendant des durées adéquates sur l'intervalle T_e. Et afin de minimiser les ondulations de tension et par conséquent les harmoniques, on admet qu'il faut réaliser V_s avec les tensions les plus proches [16]. On pose donc :

$$\frac{1}{T_e} \int_{nT_e}^{(n+1)T_e} V_s dt = \frac{1}{T_e} [t_i \cdot (V)_i + t_{i+1} \cdot (V)_{i+1}] \quad \text{II. 18}$$

Avec :

t_i et t_{i+1} sont les durées pendant lesquelles on applique les tensions $(V)_i$ et $(V)_{i+1}$

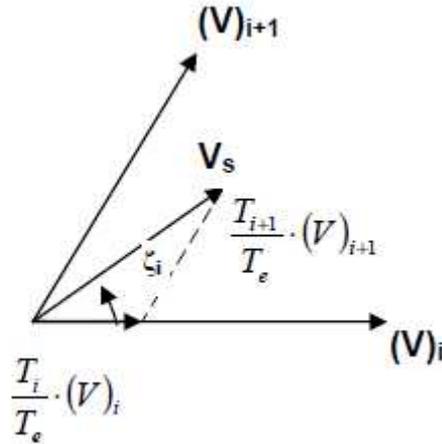


Figure-II-6- Principe de construction du vecteur de référence V_s .

En supposant que la période d'échantillonnage T_e est petite et le changement de V_s est relativement lent, on peut écrire alors :

$$V_s = \frac{1}{T_e} [t_i \cdot (V)_i + t_{i+1} \cdot (V)_{i+1}] \quad \text{II. 19}$$

Avec : $t_i + t_{i+1} \leq T_e$

Puisque la somme des temps t_i et t_{i+1} est inférieure au temps d'échantillonnage de l'onduleur et afin de maintenir la fréquence de commutation constante, un état nul de l'onduleur est appliqué durant une durée complémentaire à T_e [10] [15] [21].

En pratique, on choisit des durées égales pour l'application des vecteurs nuls :

$$t_0 = t_7 = \frac{T - t_i - t_{i+1}}{2} \quad \text{II. 20}$$

On aura donc :

$$V_s = \frac{1}{T_e} [t_i \cdot (V)_i + t_{i+1} \cdot (V)_{i+1} + \frac{t_0}{2} (V)_0 + \frac{t_7}{2} (V)_7] \quad \text{II. 21}$$

Une fois la décomposition de V_s suivant les deux vecteurs $(V)_i$ et $(V)_{i+1}$ faite, il ya lieu de déterminer les durées t_i et t_{i+1} . Cela s'effectue comme suit [11, 15, 16,17,18] :

A partir des expressions (II.16) et (II.19), on peut écrire :

$$V \cdot \begin{bmatrix} \cos[\frac{\pi}{3}(i-1) + \xi_i] \\ \sin[\frac{\pi}{3}(i-1) + \xi_i] \end{bmatrix} = \frac{1}{T_e} \left[t_i \cdot \sqrt{\frac{2}{3}} \cdot V_{dc} \begin{bmatrix} \cos[\frac{\pi}{3}(i-1)] \\ \sin[\frac{\pi}{3}(i-1)] \end{bmatrix} + t_{i+1} \cdot \sqrt{\frac{2}{3}} \cdot V_{dc} \begin{bmatrix} \cos[\frac{\pi}{3}(i)] \\ \sin[\frac{\pi}{3}(i)] \end{bmatrix} \right] \quad \text{II.22}$$

La résolution de cette équation matricielle nous donne :

$$t_i = [\sqrt{3} \cdot \frac{V}{V_{dc}} \sin(\frac{\pi}{3} - \xi_i)] T_e \quad \text{II.23}$$

$$t_{i+1} = [\sqrt{3} \cdot \frac{V}{V_{dc}} \sin(\xi_i)] T_e$$

Afin de simplifier les calculs, il est plus pratique de travailler avec des durées en per-unit qu'avec des durées en secondes. On définit alors des durées relatives τ_i, τ_{i+1} et τ_0 :

$$\begin{cases} \tau_i = \frac{t_i}{T_e} \\ \tau_{i+1} = \frac{t_{i+1}}{T_e} \\ \tau_0 = \frac{t_0}{T_e} \end{cases} \quad \text{II.24}$$

Le système (II.22) deviendra donc :

$$\tau_i = [\sqrt{3} \cdot \frac{V}{V_{dc}} \sin(\frac{\pi}{3} - \xi_i)]. \quad \text{II.25}$$

$$\tau_{i+1} = [\sqrt{3} \cdot \frac{V}{V_{dc}} \sin(\xi_i)]$$

L'expression (II.24) n'est pas pratique car elle est fonction de ξ_i , alors que nous, nous cherchons à avoir τ_i et τ_{i+1} en fonction de V_α et V_B .

Afin d'avoir τ_i et τ_{i+1} en fonction de V_α et V_B , on combine les expressions (II.16) et (II.25), on aura alors :

$$\tau_i = \frac{\sqrt{3}}{V_{dc}} [\sin(\frac{\pi}{3}(i)) \cdot V_\alpha - \cos(\frac{\pi}{3}(i)) \cdot V_B]. \quad \text{II.26}$$

$$\tau_{i+1} = \frac{\sqrt{3}}{V_{dc}} [(-\sin(\frac{\pi}{3}(1-i)).V_{\alpha} + \cos(\frac{\pi}{3}(1-i)).V_{\beta}]$$

III-3-4- Elaboration des séquences de commande des interrupteurs

Pour piloter l'onduleur, il est plus pratique de considérer des durées relatives D_1, D_2 et D_3 (rapports cycliques) pendant lesquelles les grandeurs booléennes de commande (f_1, f_2 et f_3) sont égales à 1 sur une période d'échantillonnage.

Les rapports cycliques (D_1, D_2 et D_3) sont fonctions de τ_i, τ_{i+1} et τ_0 et leurs expressions diffèrent suivant le secteur où on se trouve. Comme le nombre de secteurs est limité à 6, on peut déterminer les rapports cycliques D_1, D_2 et D_3 pour tous les secteurs. Tout cela est récapitulé dans le tableau (I .2).

La disposition respective des intervalles τ_i , et en particulier τ_0 et τ_7 peut donner lieu à de nombreux choix. Parmi les stratégies les plus fréquentes qui cherchent à minimiser le nombre de commutations et générer moins d'harmoniques, on cite la modulation asymétrique [11] et la modulation symétrique [21]. Dans notre travail, nous avons utilisé la modulation symétrique.

Secteur (i)	Durées relatives τ_i, τ_{i+1} et τ_0	Rapports cycliques D_1, D_2 et D_3	Séquence de commande (modulation symétrique)
1	$\tau_1 = \frac{3}{2V_{dc}} \left(V_{\alpha} - \frac{1}{\sqrt{3}} V_{\beta} \right)$ $\tau_2 = \frac{1}{V_{dc}} (\sqrt{3} V_{\beta})$ $\tau_0 = 1 - \tau_1 - \tau_2$	$D_1 = \tau_1 + \tau_2 + \frac{\tau_0}{2}$ $D_2 = \tau_2 + \frac{\tau_0}{2}$ $D_3 = \frac{\tau_0}{2}$	
2	$\tau_2 = \frac{3}{2V_{dc}} \left(V_{\alpha} - \frac{1}{\sqrt{3}} V_{\beta} \right)$ $\tau_3 = \frac{\sqrt{3}}{V_{dc}} \left(-\frac{\sqrt{3}}{2} V_{\alpha} + \frac{1}{2} V_{\beta} \right)$ $\tau_0 = 1 - \tau_2 - \tau_3$	$D_1 = \tau_2 + \frac{\tau_0}{2}$ $D_2 = \tau_2 + \tau_3 + \frac{\tau_0}{2}$ $D_3 = \frac{\tau_0}{2}$	

<p>3</p>	$\tau_3 = \frac{1}{V_{dc}} \cdot (\sqrt{3} \cdot V_\beta)$ $\tau_4 = \frac{\sqrt{3}}{2V_{dc}} \cdot (-\sqrt{3} \cdot V_\alpha - V_\beta)$ $\tau_0 = 1 - \tau_3 - \tau_4$	$D_1 = \frac{\tau_0}{2}$ $D_2 = \tau_3 + \tau_4 + \frac{\tau_0}{2}$ $D_3 = \tau_4 + \frac{\tau_0}{2}$	
<p>4</p>	$\tau_4 = \frac{3}{2V_{dc}} \cdot \left(V_\alpha - \frac{1}{\sqrt{3}} \cdot V_\beta \right)$ $\tau_5 = \frac{1}{V_{dc}} \cdot (\sqrt{3} \cdot V_\beta)$ $\tau_0 = 1 - \tau_4 - \tau_5$	$D_1 = \frac{\tau_0}{2}$ $D_2 = \tau_4 + \frac{\tau_0}{2}$ $D_3 = \tau_4 + \tau_5 + \frac{\tau_0}{2}$	
<p>5</p>	$\tau_5 = \frac{\sqrt{3}}{2V_{dc}} \cdot (-V_\alpha - 2 \cdot V_\beta)$ $\tau_6 = \frac{1}{V_{dc}} \cdot (V_\alpha - V_\beta)$ $\tau_0 = 1 - \tau_5 - \tau_6$	$D_1 = \tau_6 + \frac{\tau_0}{2}$ $D_2 = \frac{\tau_0}{2}$ $D_3 = \tau_5 + \tau_6 + \frac{\tau_0}{2}$	
<p>6</p>	$\tau_6 = \frac{3}{2V_{dc}} \cdot \left(V_\alpha - \frac{2\sqrt{3}}{3} \cdot V_\beta \right)$ $\tau_7 = \frac{\sqrt{3}}{2V_{dc}} \cdot (-\sqrt{3} \cdot V_\alpha + V_\beta)$ $\tau_0 = 1 - \tau_6 - \tau_7$	$D_1 = \tau_6 + \tau_7 + \frac{\tau_0}{2}$ $D_2 = \frac{\tau_0}{2}$ $D_3 = \tau_6 + \frac{\tau_0}{2}$	

Tableau-II-2- Elaboration des séquences de commande.

VI-Conclusion

Dans ce deuxième chapitre, nous avons donné un aperçu général sur les différentes stratégies de commande des onduleurs de tension et on s'est intéressé particulièrement aux techniques de modulation par largeur d'impulsion à savoir la MLI sinusoïdale et la MLI vectorielle.

CHAPITRE III

**Analyse du besoin d'un circuit MLI adapté à la commande
sans capteur de la MSAP**

I- Introduction

Durant ces vingt dernières années, le domaine de la commande des machines électriques a fait l'objet de mutations et d'évolutions continues. En effet, les méthodes classiques de variation de vitesse ont été peu à peu dépassées par des ensembles associant des convertisseurs statiques à des moteurs électriques. De nos jours, de nombreux actionneurs associant des machines à courant alternatif à des convertisseurs statiques offrent de nouvelles perspectives dans le domaine de l'entraînement à vitesse variable. Par ailleurs, l'évolution rapide des processeurs numériques a permis d'implanter des techniques de commande sophistiquées pour atteindre, ainsi, des performances élevées sur le plan de la rapidité et de la précision.

Dans la pratique, plusieurs méthodes sont utilisées pour le contrôle de la vitesse des machines électriques, et ce contrôle peut être réalisé de différentes manières et suivant différentes stratégies. Parmi elles, on peut citer la stratégie de commande par autopilotage, la stratégie de commande directe du couple (DTC) et en fin la stratégie de commande vectorielle. Dans le cadre de notre travail, on s'intéressera particulièrement à la dernière stratégie c'est-à-dire à la commande vectorielle et précisément à la commande sans capteur.

II--Commande par autopilotage

La commande par autopilotage est la méthode la plus simple, elle consiste à entraîner la MSAP en rotation en régulant trois courants sinusoïdaux triphasés [22]. La MSAP est alimentée par un onduleur de tension triphasée lui-même commandé par une structure particulière appelée « autopilotage », comme indiqué par la figure III-1.

Le principe de cette commande consiste à maintenir un déphasage Ψ constant entre le courant I_{Ψ} et la force électromotrice développé par la machine quelle que soit la position du rotor, en réalisant un asservissement des courants des trois phases de la machine. Tout cela est résumé de la manière suivante [22] :

- A partir de la connaissance de la position du rotor (à l'aide d'un capteur de position), de l'angle Ψ et de l'amplitude du courant $I_{réf}$ désirés, on synthétise les trois consignes de courant $(I_{\alpha\ réf}, I_{b\ réf}, I_{c\ réf})$.

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

- Chaque consigne du courant ($I_{a\ ref}, I_{b\ ref}, I_{c\ ref}$) est comparée à sa mesure respectivement (I_a, I_b, I_c). Ensuite, un correcteur par phase traite l'erreur d'asservissement et génère la commande de chaque bras de l'onduleur.

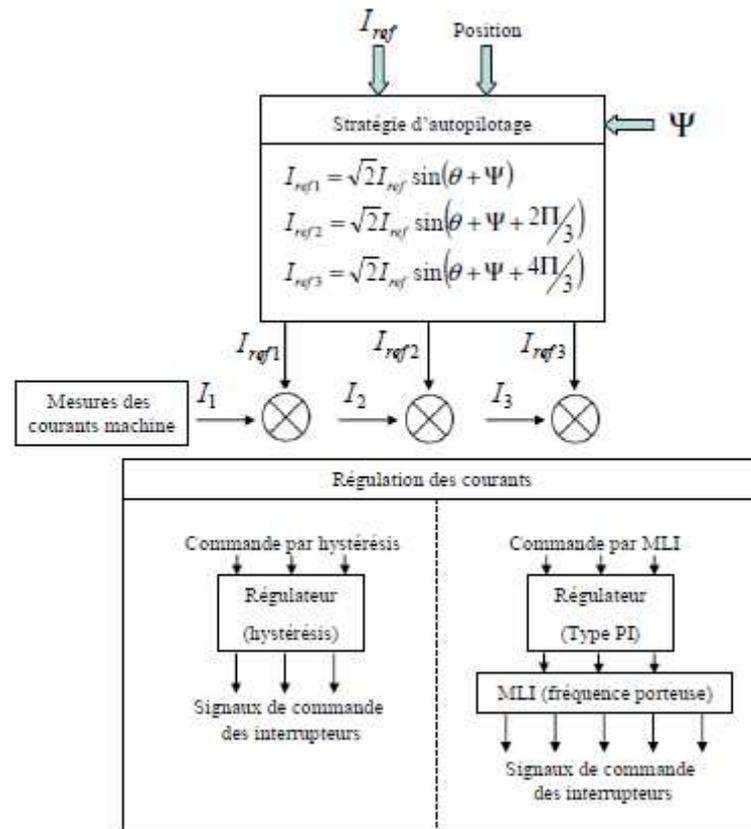


Figure.III.1 : Schéma de principe de la commande par autopilotage [22].

III- Commande directe du couple (DTC)

III-1-Principe généraux de la DTC

La méthode de contrôle direct du couple (DTC) « Direct Torque Control » à été introduite dans la deuxième moitié des années 1980 par takahashi et depenbrock comme alternative aux méthodes classiques. Elle a été conçue initialement pour la commande des machines à induction. Ensuite, plusieurs travaux ont permis le développement de la connaissance de cette méthode ainsi que son application aux machines synchrones [25].

Cette nouvelle stratégie de commande concurrentielle des méthodes classiques (commande par autopilotage et vectorielle) est basée sur l'orientation du flux statorique [9, 22, 25]. Elle repose sur la détermination directe de la séquence de commande appliquée aux

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

interrupteurs de l'onduleur de tension. Le choix de cette séquence est généralement basé sur l'utilisation de régulateurs à hystérésis qui ont pour rôle le contrôle de l'état du système à savoir ; l'amplitude du flux statorique et celle du couple électromagnétique. Cette stratégie est classée dans la catégorie des commandes en amplitude.

Cette technique a pour objectif la régulation directe du couple et du flux de la machine par application des différents vecteurs de tension générés par l'onduleur de tension [13]. Deux variables, le flux et le couple sont ainsi contrôlés généralement par l'intermédiaire de régulateur à hystérésis. La sortie de ces régulateurs détermine le vecteur de tension optimal à appliquer à chaque instant de commutation. Les intensités et les tensions sont donc commandées indirectement.

La commande DTC de la MSAP peut être schématisée par la figure suivante :

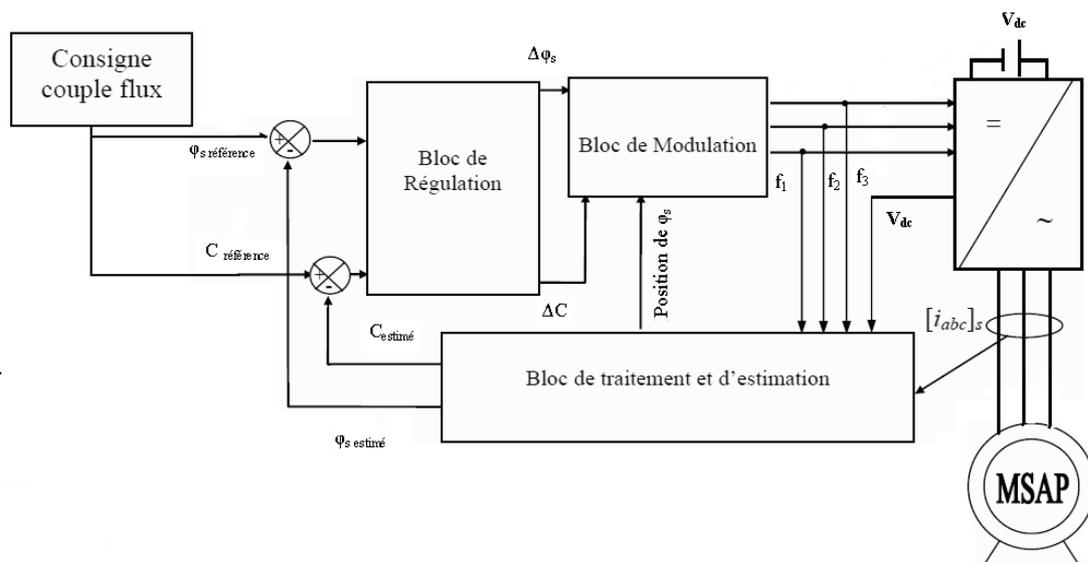


Figure .III.2 : Schéma structurel de la commande DTC appliquée à la MSAP [24].

III-2- Stratégie de commande directe du couple et du flux

A partir de la stratégie de base de la DTC proposée par takahashi, plusieurs stratégies dérivées se sont développées profitant des degrés de liberté offerts par la structure de l'onduleur de tension triphasé [24]. Dans notre cas, on s'intéressera seulement à la DTC originale de takahashi qui stipule que le contrôle directe du couple et du flux est basé sur l'algorithme suivant [25] :

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

- Le domaine temporel est divisé en deux périodes de durée T_e réduites ($T_e \leq 50$).
- Pour chaque coup d'horloge, on mesure les courants de ligne et les tensions par phase de la MSAP.
- On reconstitue les composantes du vecteur de flux statorique.
- On estime le couple électromagnétique de la MSAP grâce à l'estimation du flux statorique et à la mesure des courants de lignes.
- L'erreur entre le flux de référence et le flux estimé est introduite dans un régulateur hystérésis qui génère à sa sortie la variable binaire C_ϕ . Ainsi, $C_\phi = 1$ signifie qu'il faut augmenter l'amplitude du flux, et $C_\phi = 0$ signifie qu'il faut la réduire.
- L'erreur entre le flux de référence et le flux estimé est introduite dans un double régulateur hystérésis qui génère à sa sortie une variable logique à trois niveaux C_T . Avec, $C_T = 1$ signifie qu'il faut augmenter l'amplitude du couple et $C_T = -1$ signifie qu'il faut la diminuer et pour $C_T = 0$ il faut la maintenir constante.
- On détermine la séquence de fonctionnement de l'onduleur pour la commande du flux et du couple en se basant sur les variables logiques C_T et C_ϕ et selon l'angle angulaire θ dans lequel ce trouve le vecteur flux stator.

IV- Commande Vectorielle

IV-1- Introduction

Depuis plusieurs années, différentes recherches et études ont été menées et diverses solutions ont été proposées pour remédier au problème lié à la commande de la machine synchrone, dont la complexité réside dans le fait qu'il existe un couplage entre les variables d'entrées et de sorties et les variables internes de la machine comme le flux, le couple et la vitesse. Pour contrecarrer cette difficulté et pour obtenir une situation équivalente à celle de la machine à courant continu, Blaschke et Hasse ont proposé une technique de commande dite commande vectorielle (field oriented control) ou commande par orientation du flux [20]. L'idée fondamentale de cette stratégie est d'assimiler le comportement de la machine synchrone à celui d'une machine à courant continu; c'est-à-dire de régler le flux par une des deux composantes du courant et le couple par l'autre composante [10, 24, 25].

IV-2- Principe de la commande vectorielle

Cette méthode est basée sur la représentation des variables électriques de la MSAP dans un référentiel qui tourne avec le vecteur du flux rotorique [23, 24]. Afin de simplifier la

commande, il est nécessaire de faire un choix judicieux du référentiel, on se place donc dans un référentiel (d-q) lié au champ tournant tel que l'axe « d » coïncide avec la direction désirée du flux rotorique (**figure .II.3**).

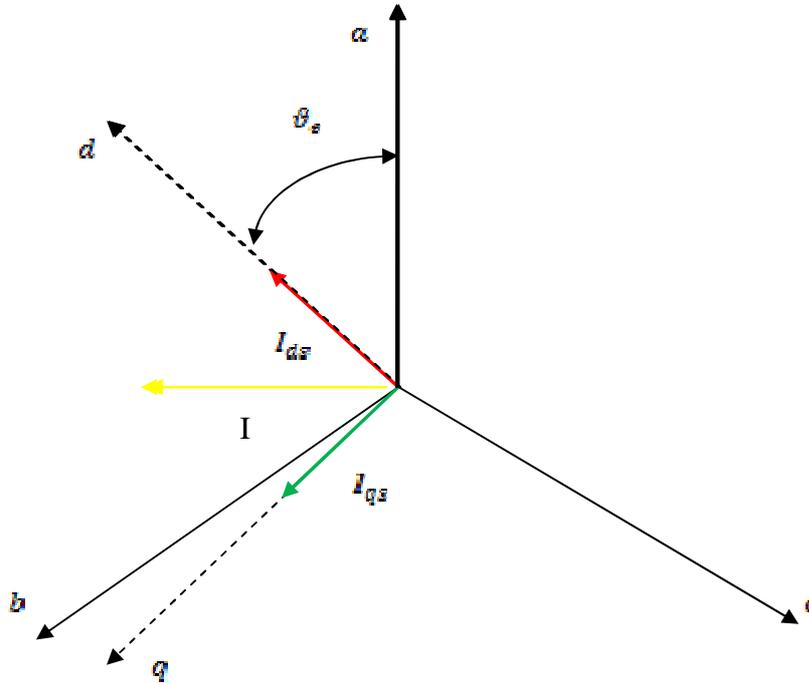


Figure.III.3 : Principe de la commande par orientation du flux.

D'après la figure (III.3), la position instantanée du rotor, et par conséquent le flux rotorique, se trouve à un angle θ_s par rapport à l'axe « a » du référentiel (abc) lié au stator. En choisissant d'orienter le flux rotorique suivant l'axe direct « d », on a la composante I_{qs} qui est en quadrature par rapport au flux rotorique. Si le courant I_{ds} est dans la même direction que le flux rotorique, le flux statorique suivant l'axe direct « d » s'ajoute au flux des aimants, ce qui engendre l'augmentation du flux d'entrefer. Par contre, si le courant I_{ds} est négatif, le flux statorique sera en opposition à celui du rotor, ce qui implique une diminution du flux d'entrefer (défluxage).

Pour un fonctionnement optimal de la MSAP, correspondant à un angle interne $\delta = \pm \frac{\pi}{2}$ [32], on a la composante directe du courant statorique I_{ds} qui est nulle, donc la composante du flux statorique suivant l'axe « d » est égale à : $\phi_d = \phi_f$ et le courant statorique se réduit à la seule composante en quadrature $I_{qs} = I_s$.

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

En tenant compte de ces deux conditions, l'expression du couple électromagnétique est :

$$C_e = P \phi_f I_{qs} \quad \text{II.1}$$

Comme le flux ϕ_f est constant, le couple électromagnétique est directement proportionnel à I_{qs} :

$$C_e = K \cdot I_{qs} \quad \text{II.2}$$

Nous pouvons remarquer, d'après les deux considérations précédentes, que le couple ne dépend que de la composante en quadrature I_{qs} si le flux rotorique est maintenu constant. Ainsi, nous avons réalisé la décomposition du courant statorique en deux termes qui correspondent respectivement au flux et au couple, et par conséquent, nous avons obtenu une structure semblable à celle d'une machine à courant continu.

IV-3- Commande vectorielle de la MSAP

Il existe deux méthodes différentes pour contrôler le courant [24, 25] :

- L'une ne nécessite pas la connaissance du modèle électrique de la machine et elle consiste à imposer directement les courants de phase dans une bande autour des courants de référence, c'est la méthode de contrôle par des régulateurs à hystérésis.
- L'autre exige la connaissance du modèle électrique de la machine et permet, en fonction de l'erreur des courants de phase par rapport à leurs références, de déterminer les références des tensions qui seront imposées aux bornes de la machine grâce à un onduleur de tension commandé en MLI. C'est sur cette méthode que se basera notre travail.

IV-3-1- Commande vectorielle de la MSAP alimentée en tension

La commande vectorielle porte généralement sur des machines alimentées en tension et régulées en courant sur les axes d et q. La figure (III.4) présente le schéma bloc d'une régulation en vitesse de la MSAP alimentée en tension et commandée par orientation de flux.

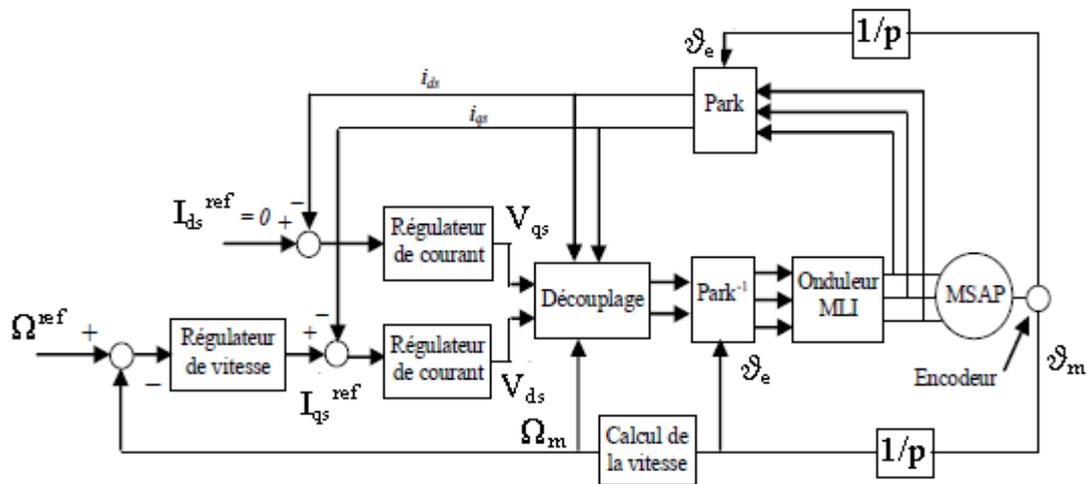


Figure.III.4 : Schéma bloc d'une régulation de vitesse de la MSAP alimentée en tension et Commandée par orientation du flux [32].

Comme le montre la figure III.4, la commande vectorielle est constituée de deux principales boucles à savoir la boucle de vitesse et la boucle des courants I_{ds} et I_{qs} .

La vitesse est régulée à travers la boucle externe du bloc. La sortie du régulateur de vitesse permet de générer le courant de référence I_{qs}^{ref} qui est comparé à la valeur du courant I_q issue de la mesure des courants réels et l'erreur sollicite le régulateur du courant I_{qs} .

En parallèle avec cette boucle, on trouve une boucle de régulation du courant I_d qui est maintenu à zéro. Les sorties des régulateurs de courant I_{ds} et I_{qs} sont appliquées à un bloc de découplage qui permet de générer les tensions de commande V_{ds} et V_{qs} et par transformation de Park, on obtient les tensions de la commande V_{as}, V_{bs}, V_{cs} de l'onduleur à commande MLI.

IV-4-Commande sans capteur de la MSAP

IV-4-1-Introduction

La méthode de commande vectorielle permet une amélioration du temps de réponse et une régulation rapide du couple [24]. Mais, elle nécessite l'utilisation de capteurs de position placés sur l'arbre de la machine qui permettent de donner la position précise du rotor afin d'assurer la régulation de la machine. Ces capteurs constituent l'inconvénient majeur de cette méthode. En effet, ils engendrent une augmentation du volume, du coût global du système et aussi une diminution de sa fiabilité [26, 27, 28]. De plus, cela provoque une difficulté de montage sur une machine de petite ou grande dimension. Ces différentes limites inhérentes à l'utilisation d'un capteur mécanique sont à l'origine de l'émergence des techniques de commande sans capteur des machines électriques.

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

Dans la littérature, trois approches principales sont employées pour estimer la position du rotor de la machine synchrone en fonction de la grandeur exploitée :

- Approche basée sur la saillance de la machine [26, 28, 29, 33, 36, 40]].
- Approche utilisant un observateur d'état [28, 30, 31].
- Approche fondée sur l'estimation de la F.E.M [28, 30, 32, 36, 40].

La première famille de méthodes est appropriée aux moteurs synchrones à forte saillance car elle exploite la variation de l'inductance en fonction de la position du rotor. Elle est souvent utilisée dans le cadre de l'estimation de la position initiale de la machine avant d'exécuter la procédure de démarrage de celle-ci. Cette méthode peut être complexe à mettre en œuvre en cas de saturation magnétique [33].

La seconde approche utilise des observateurs d'état qui sont capables de reconstruire des grandeurs non mesurées à partir d'un modèle dynamique du système et des mesures des grandeurs d'entrée et de sortie [28, 30, 31].

La dernière approche utilise les informations de vitesse et de position contenues dans l'expression de la F.E.M. En effet, la F.E.M est capable de fournir des informations instantanées sur les variables mécaniques. Celle-ci étant proportionnelle à la vitesse de rotation, il devient impossible d'extraire sa forme d'onde à basse vitesse car elle est noyée dans le bruit de mesure (la F.E.M est très petite) [27, 28, 34, 35, 36]. Afin de s'affranchir de ce problème, des méthodes spécifiques telles que la succession de rampes de fréquence ou l'injection d'un signal haute fréquence [33] ont vu le jour. C'est sur cette dernière méthode que portera la suite de notre travail.

IV-4-2- Commande sans capteur mécanique avec injection d'un signal à haute fréquence (HF)

IV-4-2-1-Principe

Cette méthode est basée sur l'injection d'une tension (ou d'un courant) de haute fréquence [34, 35, 36, 37, 38, 40]. Cette tension injectée est superposée à l'alimentation fondamentale de la machine (figure.III.5). En effet, il suffit d'ajouter les tensions $V_{abc\ HF}$ du signal à haute fréquence aux tensions V_{abc} de l'alimentation fondamentale à la sortie des régulateurs de courant. Ensuite, la tension résultante est appliquée aux bornes de la machine via un onduleur commandé en MLI. L'interaction entre la tension à haute fréquence et la saillance présentée par la machine produit un courant triphasé contenant des informations sur la position du rotor.

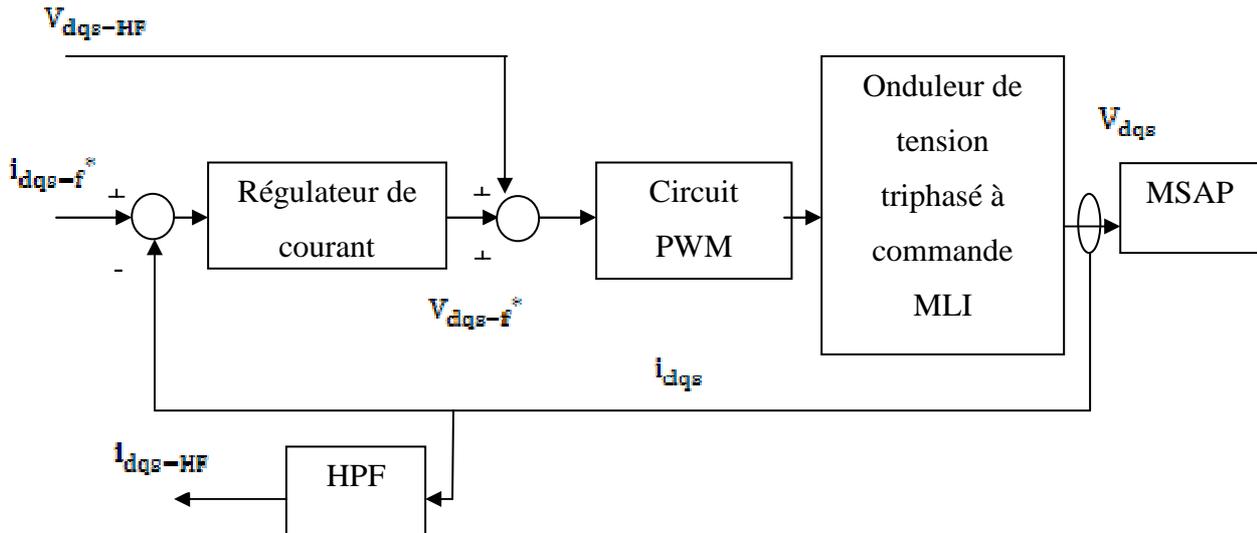


Figure.III.6 : Schéma de principe de la commande sans capteur par injection d'un signal à haute fréquence.

Le choix de l'amplitude et de la fréquence du signal injecté est dicté par plusieurs points à savoir [30] :

- La limite inférieure de la fréquence du signal injecté permettant une séparation spectrale suffisante entre la fréquence du signal injecté et celle de la tension de commande.
- La limite supérieure de la fréquence du signal injecté permettant de satisfaire le théorème de Shannon avec une résolution acceptable du signal, de l'autre côté.
- La limite inférieure de l'amplitude du signal injecté permettant d'avoir un rapport signal sur bruit suffisant.
- La limite supérieure de l'amplitude du signal injecté pour éviter d'accentuer la saturation et de perturber le fonctionnement de la machine.

IV-4-2-2-Modèle de la MSAP à Haute fréquence

Le modèle électrique de la machine synchrone à aimants permanents est représenté par le système d'équations suivant :

$$\begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} = \begin{bmatrix} r_s & 0 \\ 0 & r_s \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix} + \begin{bmatrix} s & -\omega_r \\ \omega_r & s \end{bmatrix} \begin{bmatrix} \phi_{ds} \\ \phi_{qs} \end{bmatrix} \quad \text{II.3}$$

$$\begin{bmatrix} \phi_{ds} \\ \phi_{qs} \end{bmatrix} = \begin{bmatrix} L_{ds} & 0 \\ 0 & L_{qs} \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix} + \begin{bmatrix} \phi_f \\ 0 \end{bmatrix} \quad \text{II.4}$$

Où :

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

$V_{d_s}, V_{q_s}, I_{d_s}, I_{q_s}$; représentent respectivement les tensions et les courants statoriques suivant les axes d et q.

$\phi_{d_s}, \phi_{q_s}, \phi_f$; représentent respectivement les flux au niveau des enroulements statoriques et le flux d'aimant.

ω_r : représente la pulsation électrique du rotor.

s : Opérateur de Laplace.

En transformant ces deux équations dans le repère statique (α, β) lié au champ statorique , on a :

$$\begin{bmatrix} V_{\alpha s} \\ V_{\beta s} \end{bmatrix} = \begin{bmatrix} r_s & 0 \\ 0 & r_s \end{bmatrix} \begin{bmatrix} I_{\alpha s} \\ I_{\beta s} \end{bmatrix} + \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \phi_{\alpha s} \\ \phi_{\beta s} \end{bmatrix} \quad \text{II.5}$$

$$\begin{bmatrix} \phi_{\alpha s} \\ \phi_{\beta s} \end{bmatrix} = \begin{bmatrix} L - \Delta L \cos(2\vartheta_r) & -\Delta L \sin(2\vartheta_r) \\ -\Delta L \sin(2\vartheta_r) & L + \Delta L \cos(2\vartheta_r) \end{bmatrix} \begin{bmatrix} I_{\alpha s} \\ I_{\beta s} \end{bmatrix} + \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \phi_f \cos(\vartheta_r) \\ \phi_f \sin(\vartheta_r) \end{bmatrix} \quad \text{II.6}$$

Où :

$L = \frac{L_{d_s} + L_{q_s}}{2}$: représente la valeur moyenne des inductances statoriques.

$L = \frac{L_{d_s} - L_{q_s}}{2}$: représente la différence des inductances statoriques.

Étant donné que la fréquence du signal injecté est beaucoup plus grande que la fréquence de la tension de commande, le modèle de la machine peut être simplifié en hautes fréquences. Ceci permet d'omettre les termes de couplage [34, 36, 37, 40].

A partir de toutes les hypothèses précédentes, le modèle de la MSAP en hautes fréquences se simplifié :

$$\begin{bmatrix} V_{\alpha s-HF} \\ V_{\beta s-HF} \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \phi_{\alpha s-HF} \\ \phi_{\beta s-HF} \end{bmatrix} \quad \text{II.7}$$

$$\begin{bmatrix} \phi_{\alpha s-HF} \\ \phi_{\beta s-HF} \end{bmatrix} = \begin{bmatrix} L - \Delta L \cos(2\vartheta_r) & -\Delta L \sin(2\vartheta_r) \\ -\Delta L \sin(2\vartheta_r) & L + \Delta L \cos(2\vartheta_r) \end{bmatrix} \begin{bmatrix} I_{\alpha s-HF} \\ I_{\beta s-HF} \end{bmatrix} + \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} \phi_f \cos(\vartheta_r) \\ \phi_f \sin(\vartheta_r) \end{bmatrix} \quad \text{II.8}$$

Où:

HF désigne le terme hautes fréquences.

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

$V_{\alpha-HF}, V_{\beta-HF}, I_{\alpha-HF}, I_{\beta-HF}$; désignent les valeurs de tension et de courant hautes fréquences suivant les axes α et β .

La vitesse angulaire du rotor a une faible valeur par rapport à la fréquence du signal en hautes fréquences [34, 36]. Cette conséquence nous permet de négliger le flux généré par l'aimant permanent par rapport à celui produit par le stator.

En prenant en compte l'hypothèse citée ci-dessus et en intégrant l'équation (II.7) suivant les deux axes (α, β), le courant à haute fréquence a pour forme [34, 35, 36, 37, 38] :

$$\begin{bmatrix} I_{\alpha s-HF} \\ I_{\beta s-HF} \end{bmatrix} = \frac{V_{HF}}{js(L^2 - \Delta L^2)} \begin{bmatrix} L \cos(\vartheta_H) + \Delta L \cos(2\vartheta_r - \vartheta_H) \\ L \sin(\vartheta_H - \frac{\pi}{2}) + \Delta L \sin(2\vartheta_r - \vartheta_H) \end{bmatrix} \quad \text{II.9}$$

$$\begin{bmatrix} I_{\alpha s-HF} \\ I_{\beta s-HF} \end{bmatrix} = \frac{V_{HF}}{s(L^2 - \Delta L^2)} \begin{bmatrix} L \cos(\vartheta_H - \frac{\pi}{2}) + \Delta L \cos(2\vartheta_r - \vartheta_H + \frac{\pi}{2}) \\ L \sin(\vartheta_H - \frac{\pi}{2}) + \Delta L \sin(2\vartheta_r - \vartheta_H + \frac{\pi}{2}) \end{bmatrix} \quad \text{II.10}$$

Où :

ϑ_H représente l'angle entre la tension à haute fréquence injectée et l'axe α du repère statique.

En notation exponentielle, le courant induit par l'injection d'une tension hautes fréquences est :

$$I_{\alpha\beta s-HF} = \frac{v_{HFL}}{\omega_H(L^2 - \Delta L^2)} e^{j(\vartheta_H - \frac{\pi}{2} + \varphi_{nH})} + \frac{v_{HF\Delta L}}{\omega_H(L^2 - \Delta L^2)} e^{j(2\vartheta_r - \vartheta_H + \frac{\pi}{2} + \varphi_{pH})} \quad \text{II.11}$$

Ou encore :

$$I_{\alpha\beta s-HF} = I_{HFp} e^{j(\vartheta_H - \frac{\pi}{2} + \varphi_{nH})} + I_{HFn} e^{j(2\vartheta_r - \vartheta_H + \frac{\pi}{2} + \varphi_{pH})} \quad \text{II.12}$$

Où:

I_{HFp} , I_{HFn} désignent respectivement la séquence positive et négative du courant haute fréquence.

D'après l'équation (II.12), on constate que le courant à haute fréquence induit est composé de deux termes. Le premier terme, qui est un vecteur tournant dans le même sens que la pulsation du signal injecté, a une amplitude proportionnelle à la valeur moyenne des inductances statoriques, à l'amplitude du signal injecté et inversement proportionnelle à la

fréquence du signal injecté. Cette composante ne contient pas d'informations sur la position du rotor. Le deuxième terme est un vecteur tournant dans le sens inverse de celui de la tension injectée et dont l'amplitude est proportionnelle à la différence des inductances statoriques, à l'amplitude du signal injecté et inversement proportionnelle à la fréquence du signal injecté. Cette composante contient des informations sur la position du rotor [33, 34, 35, 36, 37, 42].

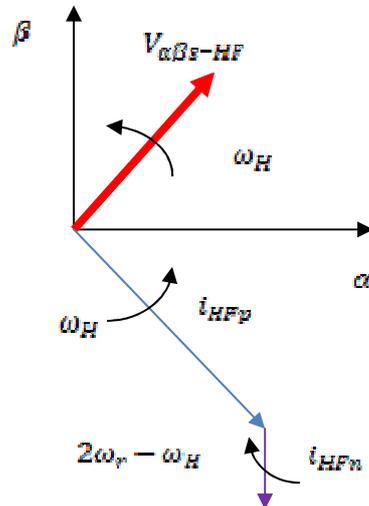


Figure.III.6 : Représentation des vecteurs tournants du courant à haute fréquence [33].

IV-4-2-3- Extraction de la séquence négative du courant à haute fréquence

En plus des deux composantes du courant à haute fréquence, le courant de la machine mesuré contient d'autres composantes à savoir ; la composante fondamentale et des composantes harmoniques qui sont dues à la commutation des composants semi-conducteurs de l'onduleur. L'extraction de la position du rotor à partir de cette mesure nécessite l'élimination de ces différents termes perturbateurs. Cela peut se faire à l'aide de différentes techniques. On trouve dans la littérature trois stratégies pour l'extraction de la séquence négative du courant à haute fréquence. Elles sont représentées comme suit :

IV-4-2-3-a-Extraction de la composante négative par un filtre passe bande suivi d'un filtre synchrone passe haut

Cette structure est constituée d'un filtre passe bande suivi d'un filtre synchrone passe haut [33, 37, 41]. Le filtre passe bande, dont la bande passante est choisie égale à la fréquence du signal à haute fréquence, permet d'éliminer la composante fondamentale [33]. Quant au filtre synchrone passe haut, il isole la composante négative du courant à HF. Ce filtre exploite la conséquence de rotation des deux vecteurs de courant à HF dans deux directions différentes [37]. En effet, il a pour principe le transfert du vecteur courant à haute fréquence dans un

référentiel en rotation synchrone avec la fréquence de la tension haute fréquence injectée. Ce passage permet de transformer la séquence positive du courant à HF en une composante continue facile à éliminer à l'aide d'un filtre passe haut. Ensuite, une transformation inverse est appliquée à la composante résultante pour transférer à nouveau le signal dans le repère statorique.

Le schéma d'extraction de la séquence négative du courant à haute fréquence est donné comme suit :

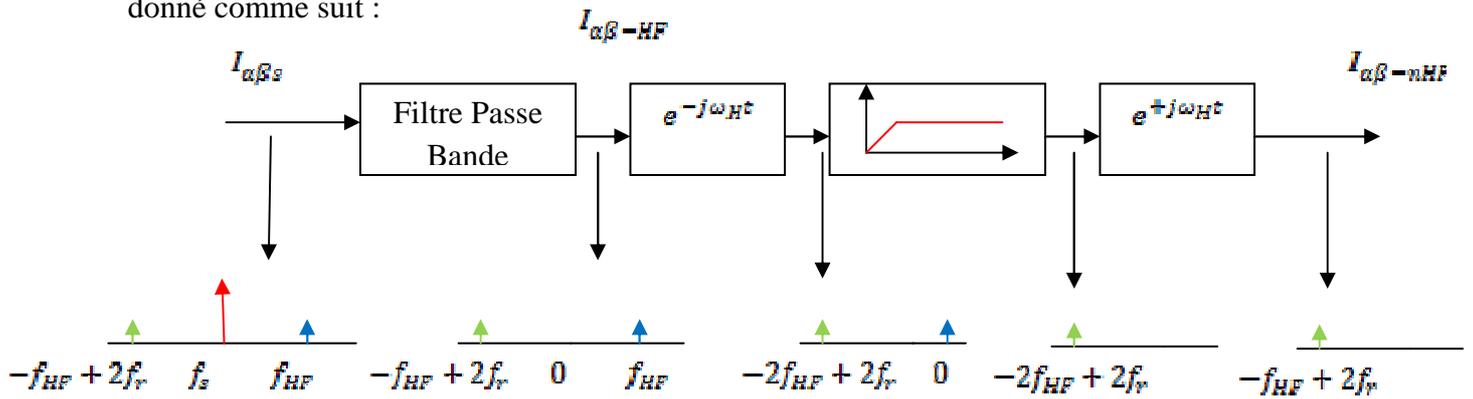


Figure.III.8 : Schéma d'extraction des composantes inverses par un filtre analogique suivi par un filtre synchrone passe haut [33].

Malgré la facilité d'implantation de cette structure, elle dispose de plusieurs inconvénients. En effet, elle introduit une asymétrie dans le signal filtré en raison de la différence entre les composants électroniques du filtre analogique utilisé pour les deux phases α et β [41]. Cette asymétrie engendre une séquence négative non modulée par la position du rotor [33]. En outre, elle crée d'autres composantes positives, ce qui complique davantage l'utilisation du filtre synchrone passe haut. L'autre inconvénient de cette stratégie réside dans le degré du filtre passe haut employer. En effet, l'utilisation d'un filtre de 1^{er} ordre accroît le temps de réponse, et cette conséquence dégrade l'estimation de la position du rotor durant les transitoires [33].

IV-4-2-3-b-Extraction de la composante négative par deux filtres synchrones passe haut

Cette structure est caractérisée par l'utilisation de deux filtre synchrone passe haut. Le premier élimine la composante fondamentale et le deuxième la composante positive [28, 33].

Le courant mesuré de la machine est transféré, en premier lieu, dans le référentiel lié à la pulsation de la tension d'excitation (f_s), ce qui ramène la composante fondamentale à une fréquence nulle (elle devient une composante continue). Ensuite, à travers un filtre passe haut, ce terme continu est éliminé. Le signal résultant de ce filtre est ramené dans un référentiel lié

à la pulsation de la tension haute fréquence injectée (f_{HF}), ce qui permet de transformer la séquence positive du courant à haute fréquence en une composante continue avant son élimination à travers un filtre passe haut. Par la suite, le signal est transféré vers un référentiel lié au champ statorique par transformation inverse.

Cette structure de filtrage est facile à implanter numériquement et elle permet d'éliminer d'une façon efficace et totale la composante fondamentale et la séquence positive du courant à haute fréquence [28]. Par ailleurs, elle n'induit pas de symétrie dans le signal filtré. Cependant, l'utilisation de filtre de 1^{er} ordre augmente le temps de réponse du système. D'autre part, cette topologie de filtrage présente une contrainte si le courant à filtrer contient d'autres composantes à savoir des termes harmoniques [33].

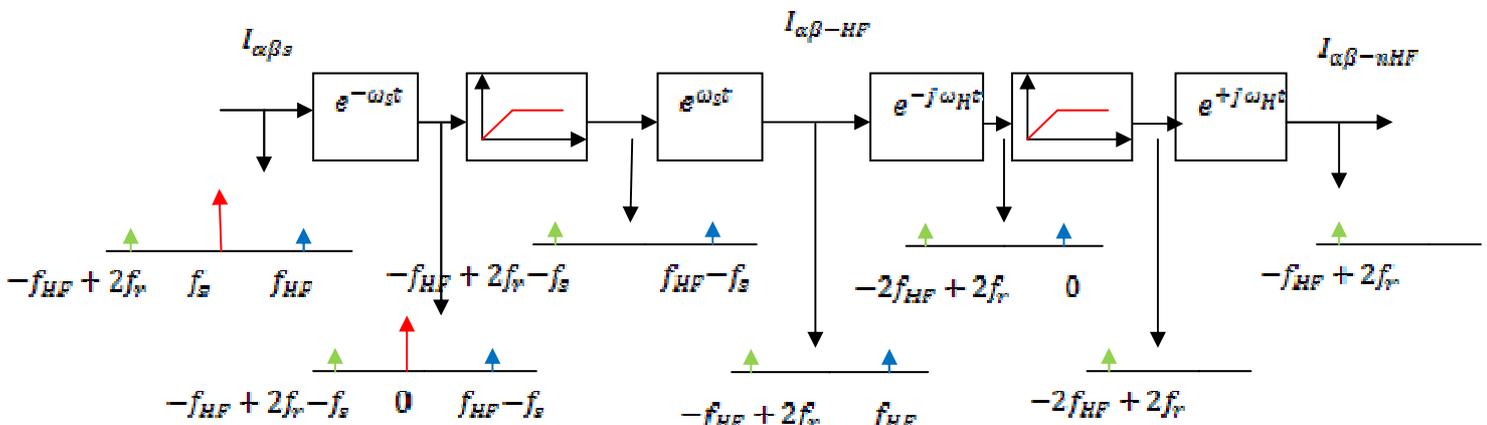


Figure III.9 : Schéma d'extraction des composantes inverses par deux filtres synchrones passe haut [33].

IV-4-2-3-c-Extraction de la composante négative par un filtre synchrone passe bas

Pour faire face aux différentes contraintes rencontrées par les stratégies d'extraction présentées ci-dessus, une autre technique a été mise au point. Celle-ci consiste à extraire la séquence négative du courant à haute fréquence par un filtre synchrone passe bas [33].

Le principe de ce filtrage est le suivant :

Le courant mesuré est transféré, en premier lieu, dans un référentiel lié à la pulsation de la tension injectée à haute fréquence ($-f_{HF}$). Ce changement de repère a pour conséquence de transférer la composante négative du courant à haute fréquence dans le domaine des basses fréquences et les composantes « fondamentale et positive » passent en haute fréquence. Par la suite, un filtre passe bas est employé pour éliminer les termes « fondamentale et positive ».

Par une transformation inverse du signal résultant du filtre passe bas, on trouve la composante négative du courant à haute fréquence.

Cette stratégie de filtrage est très efficace, elle élimine en une seule fois toutes les composantes inutiles à savoir la composante fondamentale, positive et harmonique dans le courant mesuré. Cependant, ce filtre introduit un déphasage important sur le signal filtré.

Le filtre passe bas utiliser est choisi d'ordre élevé afin d'éliminer la composante fondamentale ayant une amplitude très grande devant celle de la séquence négative [29].

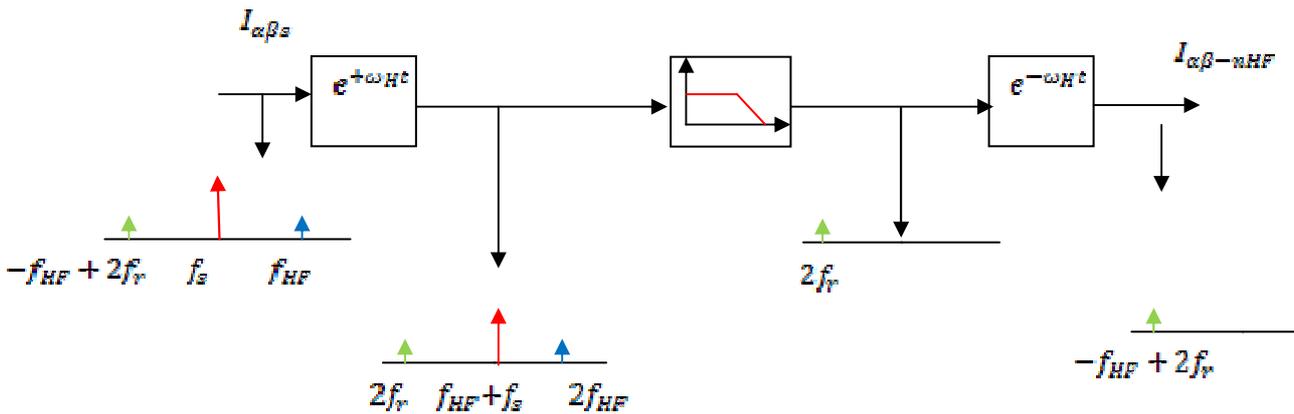


Figure.III.10 : Schéma d'extraction des composantes inverses par un filtre synchrone passe bas [33].

IV-4-2-4-Estimation de la position du rotor

Après l'extraction de la composante utile du courant mesuré contenant la position du rotor, il ne reste qu'à estimer cette position. Cela peut se faire suivant deux méthodes à savoir :

- Estimation de la position par la fonction Arctan [31, 41].
- Estimation de la position par une boucle à verrouillage de phase [28, 33, 40, 41, 42, 43, 44].

IV-4-2-4-a- Estimation par la fonction Arctan

Cette technique d'estimation se base sur la forme de la composante négative du courant à haute fréquence. En effet, à partir de l'équation représentant le terme négatif du courant à haute, on peut extraire la position du rotor.

CHAPITRE III Analyse du besoin d'un circuit MLI adapté la commande sans capteur

La composante négative du courant à haute fréquence dans le repère $(-f_H)$ est donnée par l'expression suivante :

$$I_{\alpha\beta-nHF}^{-\omega_H} = I_{nHF} e^{j(2\hat{\theta}_r + \frac{\pi}{2} + \varphi_{nH})} \quad \text{II.13}$$

Elle peut aussi s'écrire sous la forme :

$$I_{\alpha\beta-nHF}^{-\omega_H} = I_{\alpha-nHF}^{-\omega_H} \cos\left(2\hat{\theta}_r + \frac{\pi}{2} + \varphi_{nH}\right) + jI_{\beta-nHF}^{-\omega_H} \sin\left(2\hat{\theta}_r + \frac{\pi}{2} + \varphi_{nH}\right) \quad \text{II.14}$$

A partir de l'équation (II.14), on peut extraire la position du rotor :

$$2\hat{\theta}_r + \varphi_{nH} = \text{Arctan}\left(\frac{I_{\beta-nHF}^{-\omega_H}}{I_{\alpha-nHF}^{-\omega_H}}\right) + \frac{\pi}{2} \quad \text{II.15}$$

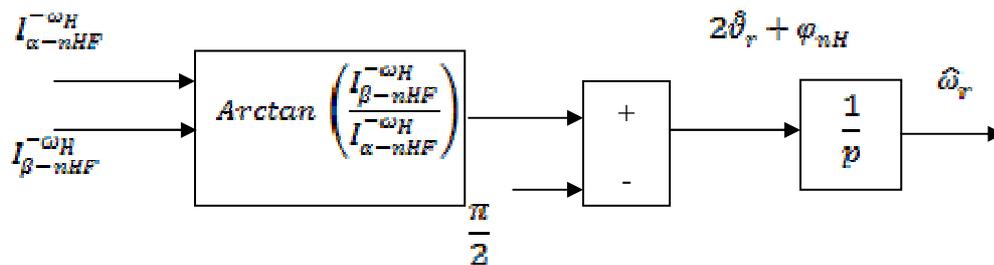


Figure.III.11 : Estimation de la position du rotor par la fonction arctg.

Cette stratégie a pour avantage sa réponse instantanée car elle ne se base pas sur un contrôleur ou sur un régime en boucle [33, 41]. Cependant, elle dispose d'un inconvénient qui réside dans la division par zéro, mais qui peut être résolu en utilisant la fonction « atan2 » qui calcule l'angle à partir des coordonnées dans le plan euclidien [33].

IV-4-2-4-a-Estimation de la position du rotor par la boucle de verrouillage de phase

Le principe de cette boucle est d'utiliser l'erreur résultant de la multiplication croisée entre un vecteur réel issu de la mesure et un vecteur modèle issu d'un modèle de référence, dans le but de forcer la convergence du vecteur modèle vers celle du vecteur réel [33, 40, 41, 43, 44].

Le principe de cette méthode est représenté par la figure ci-dessous :

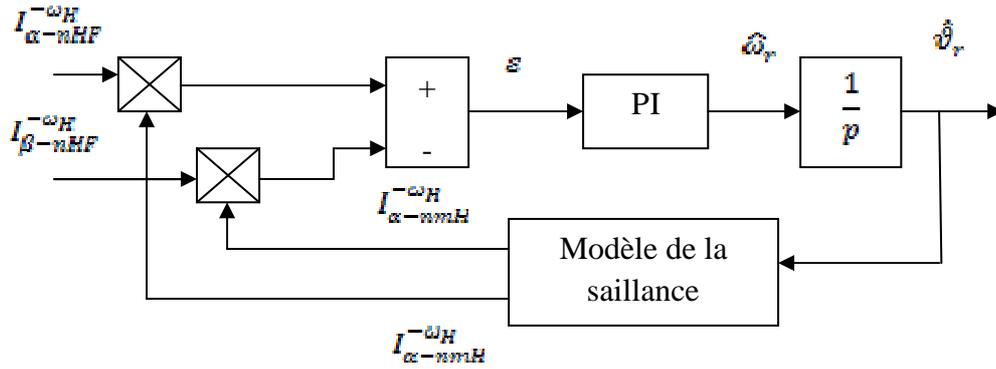


Figure.III.12: Estimation de la position du rotor par la boucle de verrouillage de phase.

Dans le cas d'une seule saillance [29], le vecteur modèle est donné par l'équation :

$$I_{\alpha\beta-nHF}^{-\omega_H} = I_{nHF} e^{j(2\hat{\vartheta}_r + \frac{\pi}{2} + \hat{\varphi}_{nH})} \quad \text{II.16}$$

L'erreur résultant de la multiplication croisée entre le vecteur modèle et le vecteur issu de la mesure est représenté par l'équation suivante:

$$\varepsilon = I_{\alpha-nHF}^{-\omega_H} * I_{\beta-nmHF}^{-\omega_H} - I_{\beta-nHF}^{-\omega_H} * I_{\alpha-nmHF}^{-\omega_H} = I_{nHF} \sin(2(\hat{\vartheta}_r - \hat{\vartheta}_r) + \varphi_{nH} - \hat{\varphi}_{nH}) \quad \text{II.17}$$

La boucle de verrouillage de phase par l'intermédiaire du régulateur PI va forcer l'erreur à converger vers zéro afin de faire coïncider le vecteur issu de la mesure et celui issu du modèle.

Pour des faibles valeurs de l'erreur, l'approximation suivante peut être faite [29] :

$$\varepsilon = I_{nHF} (2(\hat{\vartheta}_r - \hat{\vartheta}_r) + \varphi_{nH} - \hat{\varphi}_{nH}) \quad \text{II.18}$$

Cette erreur converge vers zéro si :

$$\hat{\vartheta}_r \rightarrow \hat{\vartheta}_r + \frac{\hat{\varphi}_{nH} - \varphi_{nH}}{2} \quad \text{II.19}$$

On peut ainsi estimer la position du rotor avec un déphasage de $\frac{\hat{\varphi}_{nH} - \varphi_{nH}}{2}$. Ce déphasage ne cause pas de problème dans le cas de l'estimation de la vitesse car $\hat{\varphi}_{nH}, \varphi_{nH}$ ne dépendent pas du temps.

V- Conclusion

Ce deuxième chapitre a été consacré à la présentation des différentes méthodes de commande des machines électriques, en particulier la commande vectorielle, et précisément la commande sans capteur avec injection d'un signal à haute fréquence qui nous permet de déterminer la position du rotor de la MSAP sans avoir à utiliser des capteurs de position ou de vitesse qui sont encombrants et qui présentent beaucoup d'inconvénients.

CHAPITRE VI

Réalisation et Résultats Expérimentaux

I- Introduction

Ce quatrième et dernier chapitre représente la partie expérimentale de notre projet qui consiste à la réalisation d'un circuit MLI-PWM adapté à la commande sans capteur des machines synchrones à aimant permanent. Ce contrôleur est réalisé à l'aide d'un Field programmable Gate Array (FPGA) qui accepte en entrée une fréquence et donne en sortie trois signaux modulés en largeur d'impulsion. Le FPGA est utilisé pour générer les trois signaux modulés en largeur d'impulsion destinés à la commande de l'onduleur triphasé. Le schéma de la commande réalisé est donné comme suit :

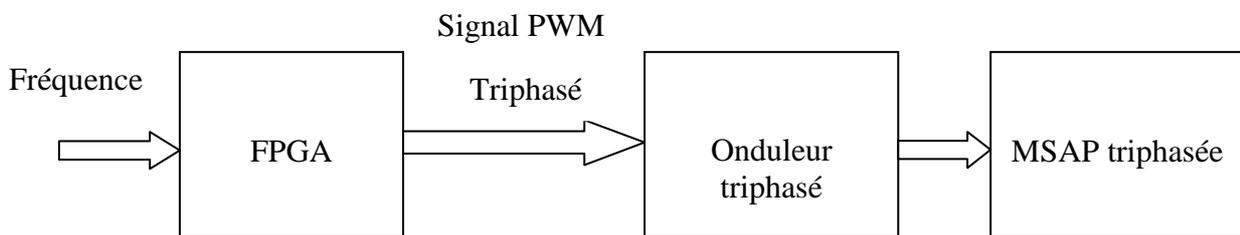


Figure VI-1- Schéma de principe du dispositif expérimental

II- Programmation du FPGA

La section suivante décrit la mise en œuvre de la PWM sur le circuit logique programmable FPGA ayant comme entrée la fréquence et donnant en sortie un signal triphasé modulé en largeur d'impulsion.

Dans notre travail, on a utilisé une plateforme Digilent Basys2 (figure VI-3) contenant une carte FPGA Xilinx Spartan 3E XC3 S250.CP 132 disposant d'un signal d'horloge de fréquence 50MHZ.

Pour la programmation du code VHDL, le logiciel Xilinx ISE 12.0 webpack a été utilisé. Ce logiciel permet de compiler, de synthétiser, de simuler et de générer le fichier VHDL (bistream). Le bistream a été programmé sur le FPGA via le câble USB en utilisant Digilent Adept 2.0 software.

Afin d'acquérir et de visualiser les différents signaux à la sortie de la plateforme, on a utilisé le convertisseur numérique-analogique (N/A) PmodDA2 (voir annexe 4).

Le schéma bloc du programme implanté sur le FPGA est donné comme suit :

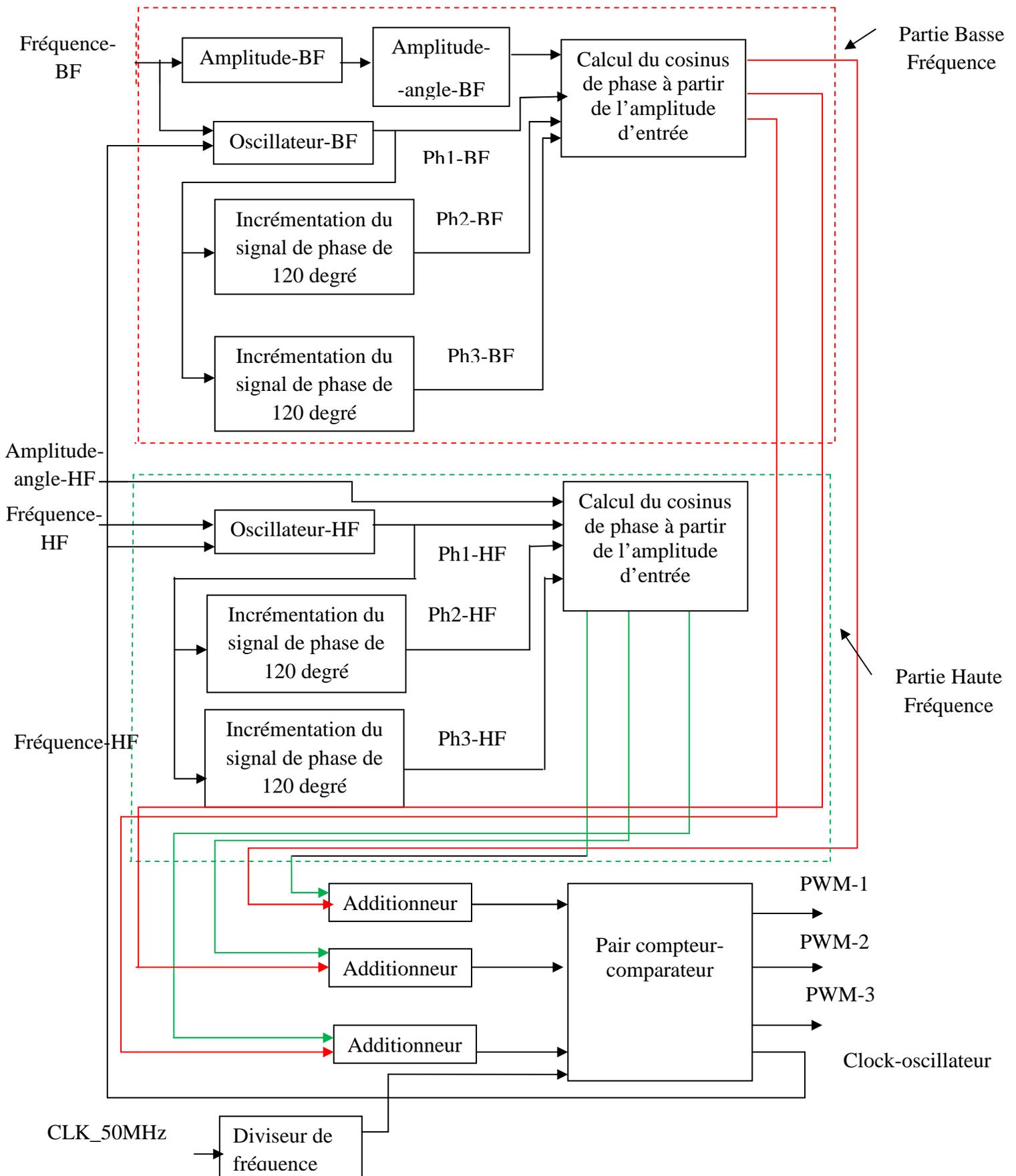


Figure-VI-2-Schéma bloc du programme implémenté sur le FPGA.



Figure-VI-3-Plateforme BASYS2

II-1- Description de la plateforme Basys 2

La carte Basys2 est une plateforme de conception et de mise en œuvre des circuits numériques. Elle est construite autour d'un FPGA SPARTAN 3E XC3 S250.CP 132 et d'un contrôleur USB AT90USB2 d'ATMEL. Cette plate forme est un matériel complet prêt à l'emploi convenant à l'hébergement des circuits allant des dispositifs logiques de base aux contrôleurs complexes. Elle est munie d'un large éventail de port E/S et de tous les supports FPGA nécessaires de sorte à pouvoir créer d'innombrables dessins et modèles sans avoir besoin d'autres composants. En outre, elle dispose de quatre connecteurs d'extension standards permettant de développer les conceptions au-delà de la carte Basys2 en utilisant des plaques d'essais, des cartes de circuit conçues par les utilisateurs ou des Pmods qui sont des modules d'E/S analogiques et numériques qui offrent la conversion analogique-numérique et numérique – analogique des signaux.

Cette plateforme peut être contrôlée et programmée via son port USB et en plus, elle est munie d'un câble USB qui est utilisé pour son alimentation.

Afin de transférer les informations de l'utilisateur vers le FPGA, Digilent offre le logiciel « ADEPT » qui détecte automatiquement la carte Basys2, et qui constitue une interface de programmation pour le FPGA et pour la plateforme flash ROM.

La carte Basys2 est conçue pour fonctionner avec toutes les versions de ISE Xilinx et même avec la version gratuite.

II-2- Description du code VHDL

Dans cette partie, on décrira le code VHDL programmé sur le FPGA.

Le code VHDL utilisé pour générer le signal triphasé modulé en largeur d'impulsion est constitué de trois parties principales. Les deux premières permettent de générer le signal à basse et à haute fréquence et elles sont subdivisées chacune en 6 entités. Quant à la troisième, elle est utilisée pour acquérir les trois signaux de commande de l'onduleur et elle est munie de trois entités. Toutes ces entités sont données comme suit :

II-2-1- Amplitude

Cette entité permet de donner la valeur de l'amplitude à partir de la valeur de fréquence en entrée. Elle accepte en entrée une fréquence à 8 bits et donne en sortie une amplitude à 8 bits. La plage de variation de cette amplitude est de 0 à 1. Lorsque elle est à 1, on atteint 100% de la valeur de l'amplitude correspondant en binaire ($2^8-1=255$) c'est-à-dire '1111111' et quand elle est égale à zéro, on atteint la valeur minimale de cette amplitude correspondant en binaire à '0000000'.

II-2-2- Amplitude-angle

Elle est utilisée pour coder la valeur de l'amplitude en un angle allant de 0 à 90 degrés. En entrée, elle dispose d'une amplitude à 8 bits et donne en aval un angle à 8 bits tel que '1111111' représente 90 degrés et '0000000' représente 0 degré. Cette conversion est effectuée suivant la relation suivante :

$$\text{Amplitude} - \text{angle} = \arccos(\text{Amplitude}) * \frac{2 * 255}{\pi}$$

II-2-3- Oscillateur

Cette entité est utilisée pour générer des valeurs de phase croissante de 0 à 360 degrés, puis de revenir à zéro, ce qui donne une forme ressemblant à une dent de scie avec une valeur minimale de zéro et une valeur maximale de 360 degrés (figure VI-4).

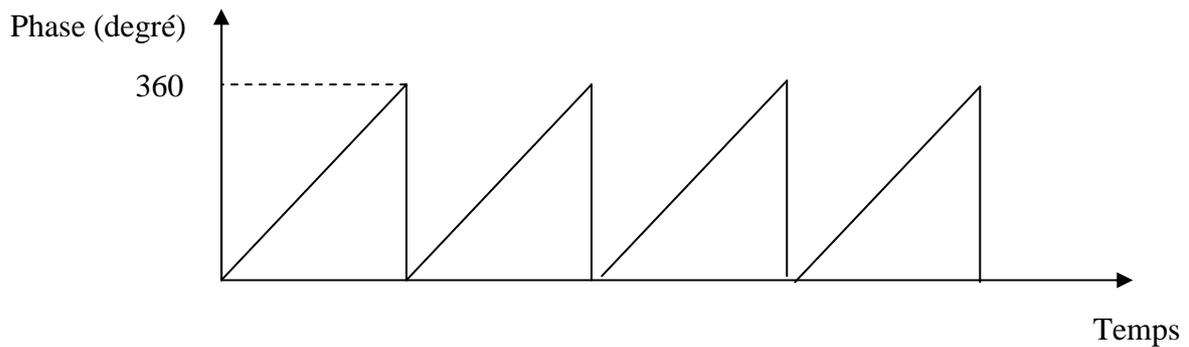


Figure-VI-4- Sortie de l'oscillateur

Elle prend en entrée des valeurs de fréquence à 8 bits, un signal d'horloge et donne en aval des valeurs de phase. La valeur de phase produite est à 10 bits tel que '000000000' présente 0 degré et '111111111' correspondant à $(2^{10}-1)$ présente 360 degrés. La valeur de 8 bits de phase présente l'incrément de phase qui est la valeur de la phase incrémentée par la valeur de la fréquence lors d'un cycle d'horloge. Et cet incrément de phase dépend de la valeur de la fréquence de sortie, de la fréquence d'échantillonnage et du nombre de bits de phase :

$$\text{Increment de phase} = \frac{f_{\text{out}}}{F_e} * 2^n$$

Avec :

f_{out} : Fréquence de sortie.

F_e : Fréquence d'échantillonnage.

n : Nombre de bit de phase.

L'oscillateur est constitué d'un additionneur suivi d'un registre fonctionnant à la fréquence d'horloge comme la figure ci-dessous l'indique :

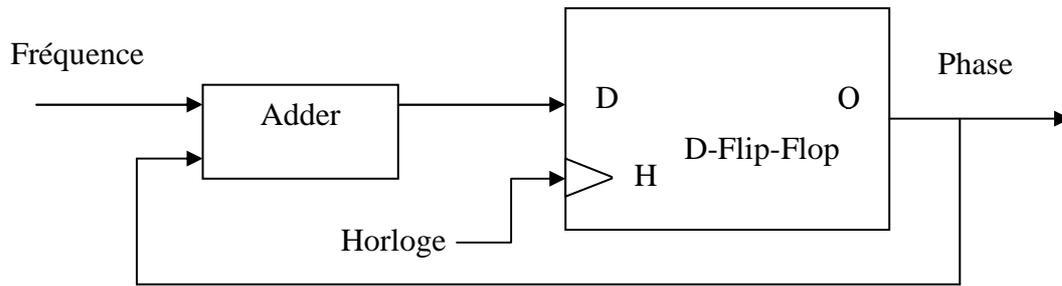


Figure-VI-5-Schéma de l'oscillateur

Le principe de fonctionnement de cet oscillateur est que pour chaque front montant de l'horloge, la valeur de phase est incrémentée par la valeur de l'incrément de phase.

La fréquence de l'horloge utilisée doit être (1/512) fois la fréquence utilisée par la paire compteur-comparateur.

II-2-4- add-120

Elle a en entrée une valeur de phase à 10 bits et elle incrémente de 120 degrés, ce qui donne en sortie une valeur de phase à 10 bits déphasée de $(2\pi/3)$ degré. Ainsi, quand une valeur de phase, par exemple P1 est incrémentée de 120 degrés, on aura P2, et si cette dernière est incrémentée aussi de 120 degrés, on aura P3. Ce qui nous donne en fin de compte un système de phase triphasé.

II-2-5- Amplitude_ckt

Cette entité est utilisée pour calculer la valeur du cosinus de phase en tenant compte de la valeur de l'amplitude en entrée. Elle dispose en entrée d'une valeur de phase à 10 bits, d'une amplitude à 8 bits et d'un signal d'horloge, et donne en sortie un cosinus à 8 bits.

Nous avons calculé la valeur de $1+A\cos\theta$, avec A ; la valeur de l'amplitude et θ ; la valeur de phase. Le calcul du cosinus se fait en utilisant une table de consultation du cosinus (LUT). Cependant, l'utilisation de multiplieurs pour multiplier la valeur d'amplitude engendre l'utilisation d'un nombre important de ressources du FPGA. Alors, on utilise une autre méthode plus simple.

Le principe de base de cette méthode est de réduire le nombre de ressources du FPGA à utiliser en effectuant une simple transformation trigonométrique. Cela est présenté comme suit :

La valeur de l'amplitude est codée en un angle allant de 0 à 90 degrés tel que '11111111' présente 90 degrés et '00000000' présente 0 degré. Le cosinus de cet angle est utilisé pour représenter la valeur de l'amplitude, ce qui nous donne alors : $\cos(A) \cdot \cos(\vartheta)$. Après une simple transformation trigonométrique, on aura :

$$1 + \frac{\cos(A - \vartheta) + \cos(A + \vartheta)}{2}$$

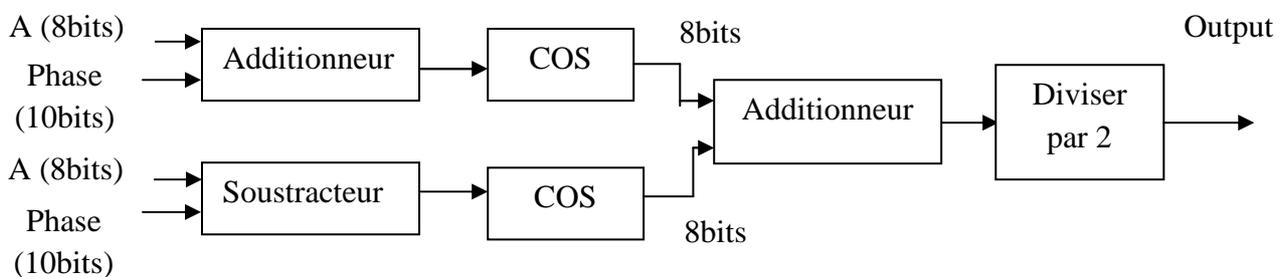


Figure-VI-6- Principe de la méthode de réduction des ressources du FPGA.

La table de consultation du cosinus (LUT) nous donne : $127.5 + 127.5\cos(\vartheta)$, avec ϑ : la valeur de phase à 10 bits et 127.5 représente 1 décimale qu'on ajoute à chaque fois afin d'avoir tous les résultats positifs à la sortie du LUT.

Le LUT utilise en entrée les huit bits les plus significatifs et nous donne en sortie un cosinus à 8 bits. Suite à la symétrie offerte par la courbe du cosinus, seulement la moitié des valeurs de phase a besoin d'une entrée dans la table de consultation du cosinus. Alors, afin que l'autre moitié des valeurs utilise la même table, un GenericSelectableInvert est utilisé. L'entrée du LUT est à 7 bits de large. Le bit le plus significatif est utilisé comme entrée du GenericSelectableInvert tel que ; quand il est à '1' les autres bits sont inversés et quand il est à '0', les bits restent inchangés. La sortie du GenericSelectableInvert est une entrée du LUT.

Étant donné qu'on doit réaliser un système triphasé, on doit utiliser la table de consultation 6 fois, 2 fois pour chaque phase. Cependant, pour un code plus efficace, le LUT n'est instancié (utilisé) qu'une seule fois. Ceci est fait comme suit :

Un compteur comptant de 0 à 11 est utilisé. Quand la valeur du compteur est à 0, la valeur de $\cos(A + \theta)$ de la première phase est utilisée comme entrée du LUT. Quand il est à 1, la sortie du LUT est considérée comme $\cos(A + \theta)$. Quand il est à 2, la valeur de $\cos(A - \theta)$ de la première phase est utilisée comme entrée du LUT. Quand il est à 3, la sortie du LUT est considérée comme $\cos(A - \theta)$. La même opération est effectuée pour les deux autres phases.

II-2-6- Adder :

Cette entité est un additionneur et il est utilisé pour effectuer la somme entre les signaux à basse fréquence et les signaux à haute fréquence. La somme de ces deux signaux est envoyée dans l'entité « PWM ».

II-2-7- PWM

Cette entité est une paire « compteur-comparateur ». Elle est utilisée pour générer les signaux modulés en largeur d'impulsion (PWM) à partir d'une entrée en cosinus issue de l'entité « Adder » (somme des signaux fondamentaux et à haute fréquence). Le principe de cette entité est de comparer la valeur de sortie d'un compteur synchrone à la valeur servie à moduler. Le compteur utilisé est un compteur modulo 8 qui doit normalement compter de 0 à 255 mais, afin d'avoir deux demi-onde et une symétrie quart onde, le compteur compte de 0 à 512. Le comparateur est à l'état bas (niveau logique zéro) si la valeur du compte est entre (256- valeur d'entrée) et (256+ valeur d'entrée) sinon, il est à l'état haut correspondant au niveau logique 1.

Cette entité génère aussi le signal d'horloge de l'oscillateur qui doit être (1/512) fois l'horloge utilisée par la paire « compteur-comparateur ».

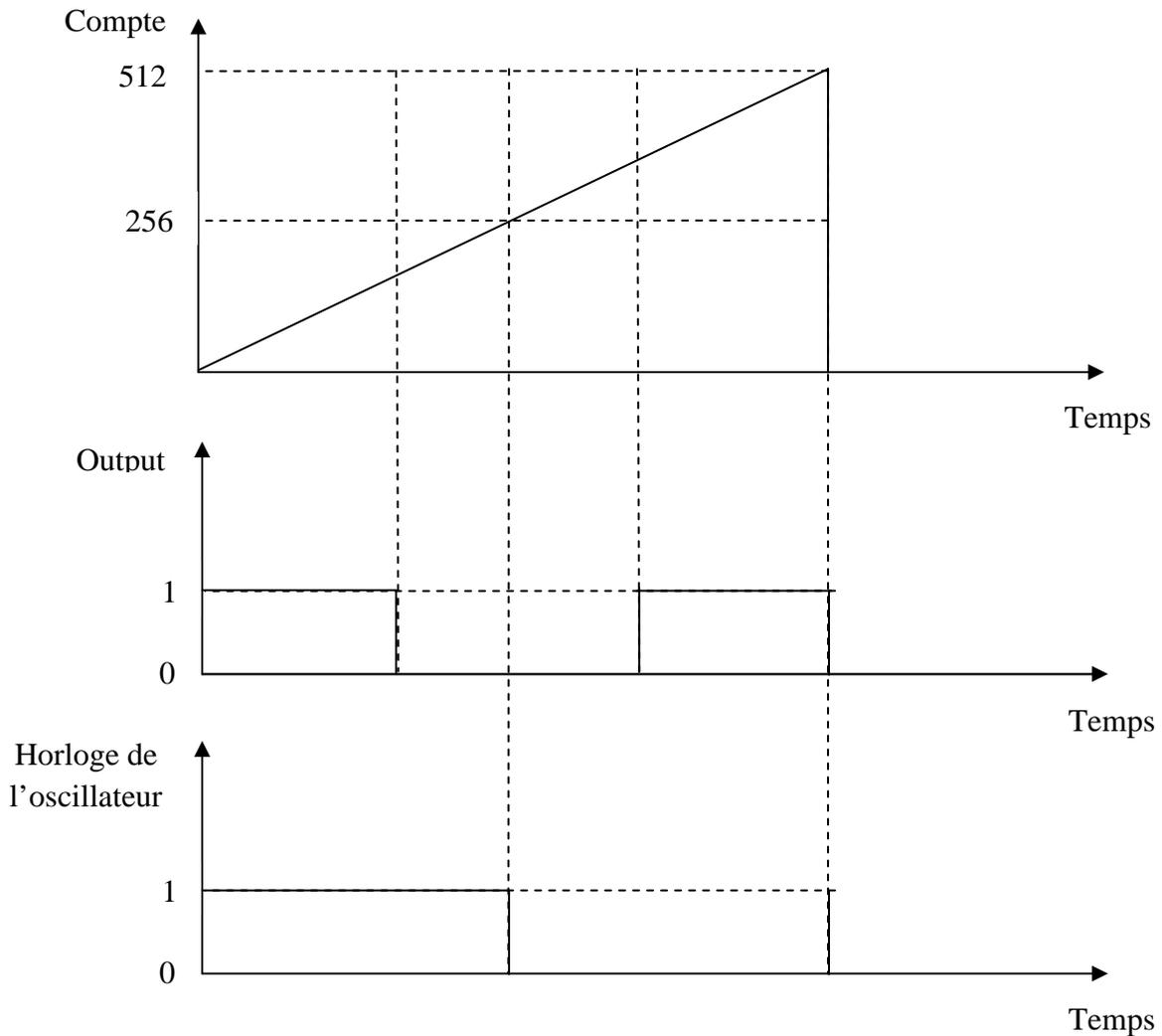


Figure-VI-7-Génération du signal PWM et du signal d'horloge de l'oscillateur.

II-2-8-Three_Phase_Pwm

C'est l'entité principale où toutes les autres entités sont instanciées. En d'autres termes, c'est l'entité qui prend les intrants dans le FPGA, à savoir, le signal fréquence à 8bits, le signal d'horloge et le signal de remise à zéro, et donne en aval les trois signaux PWM en utilisant des instances de chacune des entités mentionnées ci-dessus.

II-2-9-Clock-Divider

L'horloge interne de la carte Digilent Basys a une fréquence de 50 MHz, ce qui aboutirait à un signal PWM avec une fréquence de commutation d'environ 100 kHz, et ceci est trop élevé pour l'onduleur. Pour porter cette fréquence de commutation à environ 10 kHz, la fréquence d'horloge interne doit être divisée par 10. Cela se fait par un clock_divider, qui n'est autre qu'un compteur comptant de 0 à 9.

Ainsi, cette entité a un signal d'horloge en entrée et donne un signal d'horloge d'un dixième de la fréquence de l'entrée.

II-3-Sommaire du design

Le sommaire design suivant est généré par Xilinx ISE 12.0, après que le code VHDL soit synthétisé, simulé et implanté sur le FPGA :

three_phase_pwm_test Project Status (15/10/2010 - 12:04:34)			
Project File:	three_phase_pwm.xise	Parser Errors:	No Errors
Module Name:	three_phase_pwm_test	Implementation State:	Programming File Generated
Target Device:	xc3s250e-5cp132	• Errors:	No Errors
Product Version:	ISE 12.1	• Warnings:	44 Warnings (0 new, 0 filtered)
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constraints Met
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	230	4,896	4%	
Number of 4 input LUTs	469	4,896	9%	
Number of occupied Slices	359	2,448	14%	

Number of Slices containing only related logic	359	359	100%	
Number of Slices containing unrelated logic	0	359	0%	
Total Number of 4 input LUTs	562	4,896	11%	
Number used as logic	469			
Number used as a route-thru	93			
Number of bonded IOBs	16	92	17%	
Number of RAMB16s	2	12	16%	
Number of BUFGMUXs	4	24	16%	
Average Fanout of Non-Clock Nets	2.38			

Performance Summary				[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)		Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed		Clock Data:	Clock Report
Timing Constraints:	All Constraints Met			

Detailed Reports						[-]
Report Name	Status	Generated	Errors	Warnings	Infos	
Synthesis Report	Current	mar. 7. sept. 12:02:24 2010	0	43 Warnings (0 new, 0 filtered)	3 Infos (0 new, 0 filtered)	
Translation Report	Current	mar. 7. sept. 12:02:43 2010	0	0	0	

Map Report	Current	mar. 7. sept. 12:02:56 2010	0	0	2 Infos (0 new, 0 filtered)
Place and Route Report	Current	mar. 7. sept. 12:03:39 2010	0	1 Warning (0 new, 0 filtered)	4 Infos (0 new, 0 filtered)
Power Report					
Post-PAR Static Timing Report	Current	mar. 7. sept. 12:03:43 2010	0	0	5 Infos (0 new, 0 filtered)
Bitgen Report	Current	mar. 7. sept. 12:04:32 2010	0	0	0

Secondary Reports			[-]
Report Name	Status	Generated	
Post-Synthesis Simulation Model Report	Current	mar. 7. sept. 12:02:28 2010	
Post-Map Static Timing Report	Out of Date	lun. 5. juil. 17:56:41 2010	
Post-Map Simulation Model Report	Out of Date	lun. 5. juil. 17:56:53 2010	
WebTalk Report	Current	mar. 7. sept. 12:04:33 2010	
WebTalk Log File	Current	mar. 7. sept. 12:04:34 2010	

Date Generated: 15/10/2010 - 12:20:02

II-4- Résultats Expérimentaux

La description fonctionnelle de la PWM a été synthétisée en utilisant le langage VHDL et implantée sur le FPGA à l'aide de ISE 12.0 (voir annexe 5).

La fréquence de commutation de la PWM est de 5MHz. Quand à la fréquence et l'amplitude du signal à haute fréquence injecté, elles sont respectivement de 900Hz et de 10% de l'amplitude fondamentale.

La fréquence fondamentale (basse fréquence), la haute fréquence et l'amplitude à haute fréquence sont injectées dans le FPGA en utilisant respectivement les commutateurs « P11 », « L3 » et « K3 » de plateforme Basys2, telle que :

$$f_{BF} = \begin{pmatrix} \text{une certaine fréquence} & \text{si P11 est à l'état haut} \\ 0 & \text{si P11 est à l'état bas} \end{pmatrix}$$

$$f_{HF} = \begin{pmatrix} 900\text{Hz} & \text{si L3 est à l'état haut} \\ 0\text{Hz} & \text{si L3 est à l'état bas} \end{pmatrix}$$

$$\text{amplitude_angle}_{HF} = \begin{pmatrix} 10\% \text{ de l'amplitude_angle_bf} & \text{si K3 est à l'état haut} \\ 0 & \text{si K3 est à l'état bas} \end{pmatrix}$$

Les commutateurs « B4 », « G3 » sont utilisés pour envoyer les signaux `pwm_in_1_1`, `pwm_in_1_2` (correspondant respectivement aux signaux basse fréquence et haute fréquence générés par notre code), `pwm_1` et `pwm_2` (correspondant à la somme des deux signaux basse et haute fréquence) vers les entrées (`inter1` et `inter2`) du convertisseur numérique-analogique (PmodDA2) qui permet de convertir les signaux numériques en signaux analogiques. Les signaux issus de la sortie du convertisseur A/N (D1 et D2) sont envoyés vers un oscilloscope qui nous permet de visualiser leurs formes.

Le fonctionnement de ces commutateurs est représenté comme suit :

$$\text{inter}_1 = \begin{pmatrix} \text{pwm_in_1} & \text{si B4 est à l'état haut} \\ \text{pwm_in_1_1} & \text{si B4 est à l'état bas} \end{pmatrix}$$

$$\text{inter}_2 = \begin{pmatrix} \text{pwm_in_2} & \text{si G3 est à l'état haut} \\ \text{pwm_in_1_2} & \text{si G3 est à l'état bas} \end{pmatrix}$$

Les trois signaux modulés en largeur d'impulsion sont acquis et visualisés à l'aide d'un oscilloscope à partir des connecteurs d'extension présents sur la périphérie de la plateforme Basys2 (connecteurs de sortie « C6 », « C5 », « A9 »).

Pour différentes valeurs de la fréquence fondamentale (basse fréquence), on visualise les trois signaux modulés en largeur d'impulsion (PWM-1, PWM-2, PWM-3), `pwm_in_1_1`, `pwm_in_1_2` (correspondant respectivement aux signaux basse fréquence et haute fréquence générés par notre code) se trouvant sur les ports de sortie du convertisseur N/A.

Pour une basse fréquence de 40Hz (en binaire "01100110") correspondant à une amplitude_angle_bf de "10111100" on a :

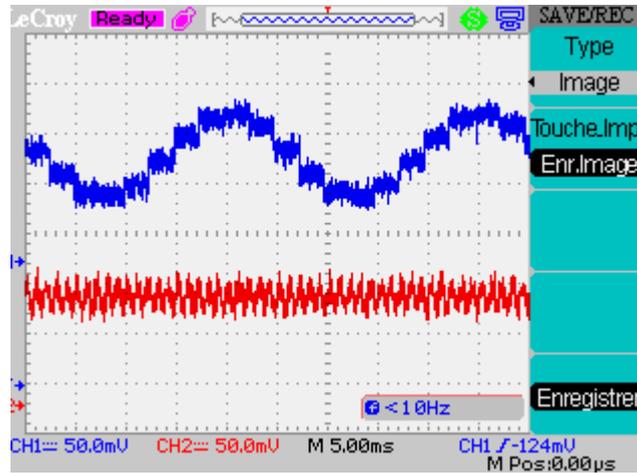


Figure-VI-8-Forme du signal fondamental à 40Hz et du signal à haute fréquence généré par le code VHDL

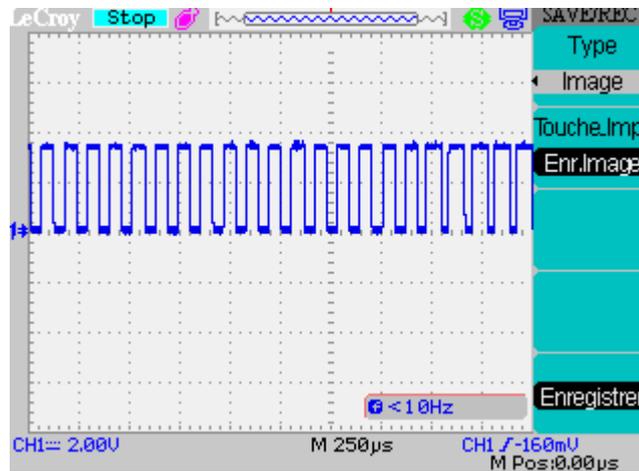


Figure-VI-9-Forme du signal modulé en largeur d'impulsion (PWM1) généré par le code VHDL

Pour une basse fréquence de 50Hz (en binaire "10000000") correspondant à une amplitude_angle_bf de "10101010" on a :

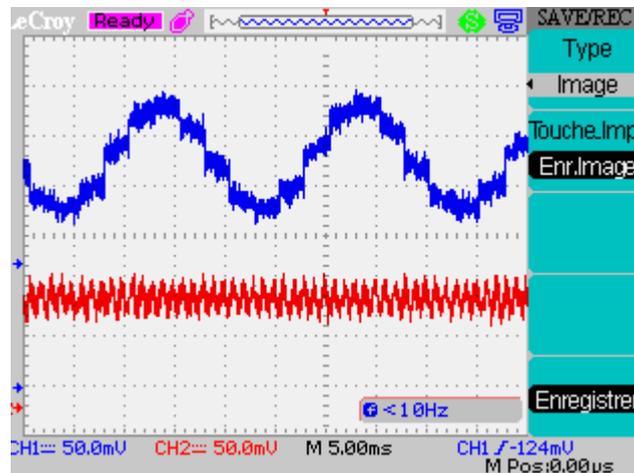


Figure-IV-11-Forme du signal fondamental à 50Hz et du signal à haute fréquence généré par le code VHDL

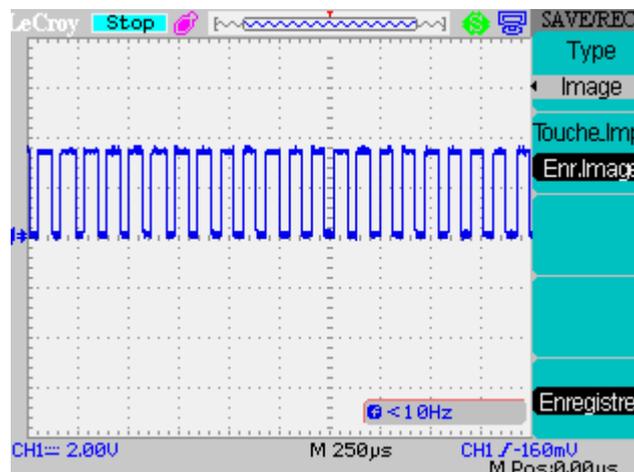


Figure-IV-11-Forme du signal modulé en largeur d'impulsion (PWM1) généré par le code VHDL

II-5-Interprétation des résultats

La manipulation réalisée au cours de notre travail consistait à l'analyse et à la mise en œuvre d'un circuit de commande PWM adapté à la commande sans capteur de la machine

synchrone à aimant permanent et son implantation sur un circuit logique programmable (FPGA).

L'analyse des résultats expérimentaux obtenus, nous a permis de montrer le bon fonctionnement du système mis en œuvre. Ainsi, d'après les figures (IV-8) et (IV-10) représentant l'évolution du signal fondamental et du signal à haute fréquence, on constate que ces deux signaux ont des fréquences et des amplitudes correspondantes à celle imposée à l'entrée de notre circuit. En outre, la somme de ces deux signaux appliqués à l'entrée de l'entité PWM, nous a permis d'obtenir les signaux modulés en largeur d'impulsion permettant le contrôle des instants de commutation des interrupteurs de l'onduleur de tension (figure IV-9) et (figure IV.11)

Par ailleurs, nous avons constaté que contrairement aux méthodes classiques de génération des signaux sinusoïdaux PWM, l'utilisation des FPGA offre de nombreux avantages. En effet, l'utilisation des FPGA réduit considérablement la taille du système réalisé ainsi que le nombre de composants utilisés. En outre, la caractéristique de reprogrammation à volonté des FPGA permet à la conception mise en œuvre d'être modifiée et améliorée à tout moment.

CONCLUSION GENERALE

CONCLUSION GENERALE

Notre travail a porté sur l'étude de la commande sans capteur de la machine synchrone à aimant permanent et en particulier sur la commande avec injection d'un signal à haute fréquence et son implémentation sur un circuit logique programmable FPGA .

Après un aperçu bibliographique sur les différentes familles des circuits logiques programmables et en particulier sur la famille des FPGAs , on s'est intéressé au langage de description VHDL, pour ensuite consacrer le deuxième chapitre aux différentes stratégies de commande des convertisseurs statiques continu/alternatif et particulièrement aux stratégies de commande MLI. Dans le troisième et avant dernier chapitre nous avons tenté de cerner les causes qui ont présidé à la naissance des techniques de commande sans capteur et c'est ainsi que nous avons abordé les différentes techniques de commande de la MSAP et précisément la technique de commande vectorielle. Cette structure de commande permet une amélioration du temps de réponse et un réglage rapide du couple, mais, elle nécessite l'utilisation de capteur de position ou de vitesse permettant de donner la position précise du rotor pour assurer son autopilotage. Ces capteurs placés sur l'arbre de la machine constituent l'inconvénient majeur de cette technique. En effet, ces derniers engendrent l'augmentation de la taille, du volume et du coût du système, ainsi qu'une diminution dans sa fiabilité. Ces différents inconvénients, sont à l'origine de l'apparition et de l'émergence des techniques de commande sans capteur qu'on retrouve dans la littérature sous trois formes différentes selon la grandeur exploitée. Dans notre travail, on s'est intéressé à la méthode fondée sur l'estimation de la F.E.M et particulièrement à la commande sans capteur par injection d'un signal à haute fréquence. Notre étude a été finalisée par la réalisation avec le langage VHDL du circuit MLI-PWM adapté à la commande sans capteur et son implémentation sur un FPGA.

Les résultats expérimentaux obtenus nous ont permis de démontrer le bon fonctionnement de notre circuit mis en œuvre et de saisir l'importance de l'utilisation des circuits logiques

programmables FPGA de par les nombreux avantages qu'elle offre (réduction de la taille du système réalisé, réduction du nombre de composants utilisés, reprogrammation à volonté des FPGA qui permet la modification et l'amélioration des conceptions réalisées).

REFERENCES BIBLIOGRAPHIQUES

Bibliographie

- [1] **M. W. NAOUAR, É. MONMASSON, I. S. BELKHODJA, A. A. NAASSANI** « Introduction à la commande numérique des machines électriques » Techniques d'ingénieur, Traité de Génie Electrique, D 2 900.
- [2] **A. NKESTA**, « Informatique Industrielle, circuits logiques programmables, mémoires, PLD, CPLD et FPGA » Editions Ellipses, 1998.
- [3] **TAVENIER, CHRISTIAN**, « Circuit logique programmable » Editions DUNOD, 1996.
- [4] **SHNEIDER, THIESSY**, « VHDL : Méthodologie de design et techniques avancées ; guide pratique du concepteur », Editions DUNOD, 2001.
- [5] **P. DARCHE**, « Architecture des ordinateurs ; fonctions booléennes, logiques combinatoires et séquentielles », Editions VUIBERT, 2002.
- [6] **R. AIRIAU, J.M.BERGE, V.OLIVE, J.ROUILLARD**, « VHDL, langage, modélisation, synthèse », Editions Presses Polytechniques et Universitaires ROMANDES, 2004.
- [7] **www. Xilinx.com**
- [8] **F. MILSANT**, « Cours d'électrotechnique Machines ; Electriques-Machines Synchrones et Asynchrones », Editions Ellipses, 2003
- [9] **J. CHATELAIN**, « Traité d'électricité, Electronique et Electrotechnique, machines électriques », Editions DUNOD, 1983.
- [10] **G. GRELLET, G. CLERC**, « Actionneurs Electriques », Editions EUROLLES, 2000.
- [11] **M. LAJOIE, P. VIAROUGE**, « Alimentation des machines synchrones », Techniques de l'ingénieur, Traité de Génie Electrique, D3630.
- [12] **J. BONAL, G. SEGUIER**, « Entraînement Electrique à vitesse variable », Editions TEC-DOC, 1998.

[13] **G. SEGUIER**, « Les convertisseurs de l'électronique de puissance », Volume 4. Editions Hermès, 1995.

[14] **G. SEGUIER**, « Electronique de puissance, structure ; fonctions de base principales applications, cours et exercices », Editions DUNOD, 2004

[15] **Zhou K., Wang D**, « Relationship Between Space-Vector Modulation and Three Phase Carrier-Based PWM: A Comprehensive Analysis », IEEE Transactions on Industrial Electronics, Vol. 49, No. 1, February 2002, page 186-196

[16] **Z. Yu, A. Mohammed, and I. Panahi**, « A review of three pwm techniques » , in Proc. Amer. Control Conf., 1997, pp. 257 – 261.

[17] **H. Van der Broeck, H. Skudelny, and G. Stanke**, « Analysis and realization of a pulse width modulator based on voltage space vectors» ,in Proc. IEEE Ind. Appl. Conf., 1986, pp.244-251.

[18] **P. A. Michael, Dr. N. Devarajan**, « FPGA Implementation of Multilevel Space Vector PWM Algorithms », International Journal of Engineering and Technology Vol. 1, No.3, August, 2009.

[19] **Y.Y. Tzou, H.J. Hsu**, « FPGA realization of space-vector PWM control IC for three-phase PWM inverters », IEEE Trans. Power Electron., vol.12, no.6, pp. 953-963, Nov. 1997.

[20] **Do-Hyun Jang, Duck-Yong Yoon**, « Space-vector PWM techniques for two phase inverter fed two phase induction motor», IEEE Trans, On industry applications, vol.39, No.2, pp.542-549,Mar./Apr. 2003.

[21] **Z. Zhou and T. Li, T. Takahashi and E. Ho**, « Design of a Universal Space Vector PWM Controller Based on FPGA», Applied Power Electronics Conference and Exposition, vol.3, page(s):1698-1720,2004.

[22] **Nicolas BERNARD**, « MACHINE SYNCHRONE : DE LA BOUCLE OUVERTE A L'AUTOPILOTAGE », Revue 3EI, n° 30, septembre 2002, pp. 24-39.

- [23] **B. TOUNSI**, « Etude comparative de groupes électrogènes embarqués à large gamme de vitesse variable associant machines à aimants permanents et conversion statique », Thèse de doctorat, Institut National Polytechnique de Toulouse, 30 Janvier 2006.
- [24] **F. KHATOUNIAN**, « Contribution à la modélisation, à l'identification et à la commande d'une interface haptique à un degré de liberté entraînée par une MSAP », Thèse de doctorat, Ecole Normale Supérieure de Cachan, 4 décembre 2006.
- [25] **C. Canudas de WI**, « Modélisation contrôle vectoriel et DTC ,commande des moteurs asynchrones», Editions Hermes, 2000.
- [26] **L. Wang. R. D. Lorenz**, « Rotor position estimation for permanent magnet synchronous motor using saliency-tracking self-sensing method », Industry applications conference, vol. 1, pp.445-450 Rome, Italy, 2000.
- [27] **T. Ané, L.Loron**, « Commande sans capteur de la MSAP par la méthode de Matsui couplée à des observateurs par mode glissant », Institut de Recherche en Électrotechnique et Électronique de Nantes Atlantique (IREENA).
- [28] **Michael W. Degner and Robert D. Lorenz**, « Using Multiple Saliencies for the Estimation of Flux, Position, and Velocity in AC Machines », IEEE transaction on industry applications, vol.34, No.5, septembre/octobre 1998.
- [29] **Z. ZHENG**, « Commande à haute performance et sans capteur mécanique du moteur synchrone à aimants permanents », Thèse de doctorat de l'Institut National polytechnique de Toulouse, 2008.
- [30] **B. N. MOBARAKEH**, « Commande vectorielle sans capteur mécanique des machines synchrones à aimants permanents : méthodes, convergence, robustesse, identification en ligne des paramètres », Thèse de Doctorat de l'Institut National Polytechnique de Lorraine, 2001.
- [31] **D. A Khaburi**, « Contribution à la commande vectorielle des machines synchrones à aimants permanents sans capteur mécaniques », Thèse de Doctorat de l'INPL, Nancy, décembre 1998.
- [32] **M. Sebba, A.Chaker, Y. Meslem, S. Hassaine**, « Commande en Vitesse du Moteur Synchrone à Aimants Permanents Dotée d'un Observateur d'Etat de LUENBERGER », 4th International Conference on Computer Integrated Manufacturing CIP'2007.

- [33] **L. Baghli, I. Al-Rouh, A. Rezzoug**, « Signal analysis and identification for induction motor sensorless control », Control Engineering Practice, Volume 14, Pages 1313-1324, November 2006.
- [34] **Ji-Hoon Jang, Seung-Ki Sul, Jung-Ik Ha, K. Ide, and M. Sawamura**, «Sensorless Drive of Surface-Mounted Permanent-Magnet Motor by High-Frequency Signal Injection Based on Magnetic Saliency », IEEE transaction on industry applications,vol.39, No.4, juillet/Aout 2003.
- [35] **M. Boussak**, « Implementation and Experimental Investigation of Sensorless Speed Control With Initial Rotor Position Estimation for Interior Permanent Magnet Synchronous Motor Drive », IEEE transaction on power electronics, vol.20, November 2005.
- [36] **X. Xiang, Y. He**, « Sensorless Vector Control Operation of a PMSM By Rotating High-Frequency Voltage Injection Approach », Proceeding of International Conference on Electrical Machines and Systems, Oct. 8~11, Seoul, Korea, 2007.
- [37] **Ji-Hoon Jang, Jung-Ik Ha, Motomichi Ohto, Kozo Ide, and Seung-Ki Sul**, «Analysis of Permanent-Magnet Machine for Sensorless Control Based on High-Frequency Signal Injection », IEEE transaction on industry applications,vol.40, No.6, novembre/decembre 2004.
- [38] **Zeroug, H.; Hocine, L** « Sensorless Control of a Surface Mounted Permanent Magnet Motor by Signal Injection Associated with Space Vector Modulation Technique » Proceeding of Electric Machines and Drives Conference, page(s): 1573 – 1580, 2009.
- [39] **C. Spiteri Staines, J. Cilia, B. Micallef et M. Apap**, «Sensroless vector control of surface mount PMSM using high frequency », Proceeding of international conference on Power Electronics, Machines and Drive, Page(s): 44 - 48, 2002.
- [40] **X. Xiang, Y. He**, « Sensorless Operation of PMSM Based on Hybrid Rotor Position Self-sensing Scheme », Proceeding of international conference on Electrical Machines and Systems, Page(s): 714 – 718, 2007.
- [41] **Teske; N.; Asher; G. M.; Sumner; M.; Bradley, K. J**, « Sensorless position estimation for symmetric cage induction motor under loaded conditions », Proceeding of IEEE Industry Applications Conference, vol. 3, pp. 1835 – 1841, 8 - 12 Oct. 2000.

[42] Degner, M.W.; Lorenz, R.D, « Position estimation in induction machines utilizing rotor bar slot harmonics and carrier frequency signal injection », Proceedings of the Power Conversion Conference, vol. 1, pp. 69 – 72, 3 – 6 Aug., Nagaoka, 1997.

[43] Ribeiro, L.A.S.; Degner, M.W.; Briz, F.; Lorenz, R.D, «Comparison of carrier signal voltage and current injection for the estimation of flux angle or rotor position », Proceedings of Industry Applications Conference, vol.1, Page(s): 452 – 459, 1998.

[44] Cilia; J.; Asher; G. M.; Bradley; K. J.; Sumner; M., «Sensorless position detection for vector-controlled induction motor drives using an asymmetric outer-section cage », IEEE Transactions on Industry Applications, vol. 33, no. 5, pp. 1162 – 1169, Sept./Oct. 1997.

ANNEXES

I-Types

Comme cité dans le premier chapitre, le langage VHDL est un langage fortement typé. Il possède quatre familles de types à savoir ; les types scalaires, les types composites, les types accès et les types fichiers

I-1- Les types scalaires

Le type scalaire représente les types énumérés, entiers, flottants et les types physiques.

La base de définition de tous types est l'énumération des éléments contenus. Toutes les déclarations commencent par le mot-clé « TYPE ». Afin de déclarer ce type de données, il suffit d'indiquer les valeurs symboliques (identificateurs ou caractères) qu'il peut prendre.

La syntaxe générale de déclaration est :

TYPE <nom_type> **IS** (val1, val2.....);

Par exemple :

TYPE BIT **IS** ('0', '1')

TYPE BOOLEAN **IS** ('FALSE', TRUE')

I-1-1- Types Entiers

Les types entiers peuvent être vus comme des types énumérés. En effet, ils représentent une suite énumérable. L'ensemble des opérations arithmétiques est défini sur tous ces types.

La déclaration d'un type entier se fait par indication de son intervalle de variation par le mot-clé « RANGE ». L'exemple suivant déclare un type énuméré entier variant de 1 à 12 bornes comprises.

TYPE profondeur_de_la_pile **IS RANGE** 1 to 12.

La syntaxe générale de déclaration est :

TYPE <nom_type> **IS RANGE** min to max.

I-1-2- Types physiques

Ils sont considérés comme des entiers associés à une unité physique.

I-1-3- Types flottants

De manière tout à fait identique aux types entiers, la déclaration d'un type flottant se fait en spécifiant ses bornes (flottantes aussi). L'exemple suivant déclare un type flottant variant de 1.89 à 6.07

TYPE <Nom_flottant> **IS RANGE** 1.89 to 6.07.

I-2- Les types composites

Comme tout langage de haut niveau, VHDL permet à l'utilisateur de structurer ses données au moyen des types composites constitués d'éléments scalaires ou eux-mêmes composites, tous de même type ou de types différents. Il existe deux catégories de types composites, les tableaux (array) et les enregistrements (records).

I-2- -1- Tableaux

Le tableau est une collection d'éléments, tous de même type, repérés par les valeurs d'indices.

I-2-2-Enregistrements

Ils permettent de rassembler les objets de types différents dans une même organisation et de repérer chaque élément par son nom.

La syntaxe de déclaration est :

TYPE <record_name > **IS RECORD.**

Elément_déclaration

END RECORD ;

I-3-Les types accès ou pointeurs

La syntaxe générale de déclaration est:

TYPE <nom_type> **IS ACCESS** <type_pointe> ;

I-4-Les types fichiers

Un fichier est une séquence d'objets, de même type, contenus dans un fichier du système hôte. Au cours de la déclaration d'un type fichier, des procédures d'accès sont déclarées implicitement : **FILE_OPEN** (.....) ; **FILE_CLOSE** (.....) ; **READ** (.....) ; **WRITE** (.....) et une fonction indique la fin d'un fichier, **END FILE**(...).

La syntaxe de déclaration est comme suit :

Type <nom_de_type> **IS FILE of** <type_des_éléments> ;

II- Les attributs

II-1- Attributs prédéfinis

Ces attributs sont classés suivant la nature de leur préfixe et la catégorie de l'attribut lui-même ; valeur, type, fonction ou signal.

II-1-1- Attributs de type ou de sous-type

➤ **Type de base**

T'base retourne le type de base de son préfixe. Pour un type donné, le type de base est le type lui-même. Pour un sous-type, le type de base est le type à l'origine du sous-type.

➤ **Bornes des types scalaires**

Ces attributs s'appliquent aux types ou aux sous-types scalaires et retournent une de leurs bornes limites. Ces attributs sont :

T'Left ; retourne la borne de gauche du type préfixe.

T'Right ; retourne la borne de droite du type préfixe.

T'Hight ; retourne la plus grande des deux limites du type ou du sous-type préfixe.

T'Low ; retourne la plus petite des deux limites du type ou du sous-type préfixe.

➤ **Conversion valeur-position**

Ils permettent la conversion entre variable de position d'un type et la valeur dans le type. Ils sont valables pour des types entiers, énumérés ou physiques.

T'Position (X); retourne la variable de position pour la valeur X.

T'Valeur (X) ; représente la fonction inverse. X représente une position et la fonction retourne la valeur selon la définition du type et du sous-type.

➤ **Déplacement de position**

Ces quatre attributs permettent d'incrémenter ou de décrémenter une position. C'est la valeur dans le type ou le sous-type qui est retournée.

T'Succ (X) ; retourne le successeur de X dans le type de base.

T'Pred(X) ; retour le prédécesseur de X dans le type de base.

T'LeftOf(X) ; retourne la valeur gauche de X.

T'RighOf (X) ; retourne la valeur droite de X.

II-1-2- Attributs de tableau

Dans ces attributs le préfixe doit être un objet de type tableau, un sous type tableau, un pointeur ou un alias. Ils possèdent un paramètre optionnel (N) permettant de faire porter l'attribut sur une dimension particulière du tableau.

➤ **Bornes d'index de tableau**

A'Left (N) ; retourne la borne gauche du Nème index de A.

A'Right (N) ; retourne la borne droite du Nème index de A.

A'Low (N) ; retourne la borne inférieur du Nème index de A.

➤ **Intervalle de variation d'index de tableau**

Ces attributs sont utilisés lors de l'écriture des boucles ou pour spécifier à partir d'un type tableau d'autres types ou sous-types.

A'Range (N) ; retourne l'intervalle de variation de la dimension N de A. Dans ce cas, l'ordre prédéfini est inchangé.

A'Reverse_Range (N) ; retourne l'intervalle de variation de la dimension N de A. L'ordre prédéfini est inversé.

A'Length (N) ; retourne le nombre d'éléments du Nème index de A.

II-1-3- Attributs de signal

Dans cette catégorie d'attributs, le préfixe doit être un signal. Le type du résultat peut être signal ou fonction.

➤ **Attribut signal**

Ces attributs retournent un signal. Les deux premiers sont sensibles à des événements et les deux autres aux transactions portant sur le signal préfixe. Les trois premiers possèdent un paramètre optionnel de type « Time » qui ne peut être négatif. Ces attributs ne peuvent être utilisés à l'intérieur de fonctions ou de procédures.

S'Delayed(T) ; est un signal de même type que S dont les valeurs sont retardées de T unités de temps.

S'Stable (T) ; est un signal de type booléen qui est vrai si S n'a pas changé de valeur dans l'intervalle de temps écoulé T.

S'Quiet (T) ; est un signal de type booléen qui est vrai si S n'a pas connu de transaction dans l'intervalle de temps écoulé T.

S'Transaction (T) ; est un signal de type bite qui change de valeur à chaque transaction sur S. Cet attribut permet aussi de transformer une transaction en événement

➤ **Attribut fonction**

Etant donné que les attributs de signal ne peuvent pas être placés dans un sous-programme, il existe des attributs de fonction.

S'Event ; est un booléen, vrai quand le signal auquel il est appliqué a changé depuis le dernier cycle de test.

S'Active ; est un booléen, vrai quand le signal auquel il est appliqué subit une transaction.

S'Last-Event ; Retourne le Laps de temps écoulé depuis le changement du signal S.

S'Last-Active ; Retourne le Laps de temps écoulé depuis la dernière transaction.

S'Last-Value ; Retourne la valeur qu'avait S juste avant le dernier événement le concernant.

II-1-4- Attributs de bloc

Le préfixe pour ce type d'attribut doit être le nom d'une architecture ou l'étiquette identifiant un bloc.

B'Behavior ; est un booléen vrai si et seulement si B ne contient d'instanciation de composant.

B' Structure ; est un booléen vrai pour une structure pure. Le bloc B ne doit contenir aucune affectation de signal concurrente ou dans un processus.

II-2- Attributs définis

L'utilisateur peut définir ses propres attributs sous réserve qu'ils correspondent à des constantes. La syntaxe générale de déclaration est :

Attribut-déclaration ::=

Attribute identifier :type_mark :

Une valeur lui est associée par une spécification d'attribut :

Attribute_specification ::=

Attribute identifier **of** entity_specification **is** expr :

Entity_specification ::=

Entity_name_list:entity_class;

I- Etape d'implémentation sous environnement ISE

L'implémentation d'un circuit numérique se fait suivant les étapes suivantes :

I-1- Lancement d'ISE 8.2 et création du projet

Cette étape consiste à lancer le Navigateur ISE et à créer le projet à réaliser. Cela se fait comme suit :

- On clique deux fois sur l'icône sur le bureau de travail où on choisit :
Start → Programs → XilinxISE → Project Navigator.
- Si des messages apparaissent, on sélectionne cancel
- On choisit **File → New Project.**
- On donne un nom pertinent au projet à réaliser.
- Etant donné qu'on désire réaliser un design à base d'un code HDL, on choisit **HDL** comme **Top_Level Source Type.**
- Ensuite on clique sur **Next.**
- On choisit les paramètres du circuit programmable utilisé à savoir :
Family ; permet de choisir la famille du circuit logique programmable.
Device ; permet de sélectionner le sous-type (nombre de portes logiques).
Package ; définit le type de boîtier.
- On appuie sur **Next.**
- Enfin, on appuie sur **Next** deux autres fois pour les deux autres fenêtres et sur **Finish.**

I-2- Description du circuit numérique à réaliser avec un fichier VHDL

Cette étape permet de réaliser un fichier source qui renferme les données descriptives ou de configuration du circuit à réaliser. La construction de ce fichier se fait suivant les instructions suivantes :

- On sélectionne le menu **Project → new Source.**
- On sélectionne **VHDL Module** comme source et on entre le mot du schéma à réaliser.
- On vérifie que l'option **add to Project** est cochée.
- On clique sur **Next.**
- On déclare les ports entrée/sortie du design à réaliser.

- On clique sur **Next** et ensuite sur **Finish**. Une ébauche du fichier apparaît avec la description de l'entité et de l'architecture.
- Dans la description de l'architecture, en dessous de l'énoncé begin, on insère les fonctions réalisant notre schéma ou circuit.

I- 3- Compilation

Une fois le fichier VHDL édité, il est conseillé de vérifier la syntaxe du design dans le but de trouver les erreurs de syntaxe ou de typographie :

- On vérifie que **synthesis / implementation** est sélectionné dans la liste déroulante de la fenêtre **sources**.
- On sélectionne le fichier VHDL **VHDL_didacticiel** pour afficher les processus liés dans la fenêtre **processes**.
- On clique sur le « + » à côté de **synthesize-XST**.
- On double-clique sur le processus **check syntax**. Si tout va bien, un crochet vert apparaît. Sinon :
- On consulte les messages d'erreurs dans la console au bas de l'écran.
- On corrige les erreurs s'il y a lieu, puis on ferme le fichier VHDL.

I- 4- Simulation du design

Cette étape permet de vérifier que le design décrit précédemment fonctionne de la façon prévue par les spécifications. Cette simulation est une simulation comportementale. Afin de réaliser cette dernière, il faut créer un banc d'essai contenant les stimuli d'entrée.

- On sélectionne le fichier à simuler (**VHDL_didacticiel**).
- On crée un nouveau banc d'essai en sélectionnant **Project_ New Source**.
- Dans la fenêtre qui s'ouvre, on sélectionne **Test Bench WaveForm** comme source, et on donne un nom au fichier.
- On clique sur **Next**.
- Une page apparaît montrant quel fichier source est associé au banc d'essai; il s'agit de notre fichier à simuler.
- On clique **Next**, puis **Finish**.
- Une fenêtre permettant d'effectuer une initialisation temporelle (détermination des paramètres de l'horloge et de certaines contraintes temporelles) apparaît.
- On clique enfin sur **Finish** pour terminer cette initialisation temporelle.

- Dans la fenêtre **sources**, on sélectionne **behavioral simulation**.
- On assure que le banc d'essai est bien sélectionné dans la fenêtre **sources**.
- Dans la fenêtre **processes**, on déroule l'outil **xilinx ise simulator**.
- On double-clique sur **Simulate Behavioral Model**.
- On corrige notre design si nous obtenons des erreurs.

I-5- Synthèse et implémentation du design

I-5-1- Description de la synthèse et de l'implémentation

La synthèse d'un circuit consiste à traduire la description du circuit à réaliser en blocs disponibles dans la technologie utilisée. Par exemple, pour un circuit décrit avec un code VHDL et qui doit être réalisé sur FPGA, le processus de synthèse convertit et regroupe les portes logiques en composantes réalisables sur le FPGA choisi.

L'implémentation du circuit se fait en quatre sous étapes :

- La transformation (**mapping**) : elle consiste à regrouper les composants obtenus lors de la synthèse dans des blocs spécifiques du FPGA.
- la disposition (**placement**) qui consiste à choisir des endroits spécifiques sur le FPGA où disposer les blocs utilisés, et choisir les pattes du FPGA correspondant aux ports d'entrée et de sortie;
- le routage (**routing**) : établir des connexions électriques entre les blocs utilisés.
- la configuration (**configuration**) : convertir toute cette information en un fichier pouvant être téléchargé sur le FPGA pour le programmer.

I-5-2- Ports d'entrée et de sortie

Lors de l'étape de disposition de l'implémentation, il faut assigner des pattes spécifiques du FPGA à des ports d'entrée et de sortie de son design.

L'assignation des ports se fait par l'entremise d'un fichier de contraintes avec l'extension «.ucf » (pour user constraints file).

- Dans la fenêtre sources, on sélectionne **Synthesis/Implementation**
- Dans la fenêtre **Processes**, on déroule le menu **User Constraints** puis on double-clique sur **Assign Package Pins**.

- Dans la fenêtre suivante, on clique sur **Yes** afin de créer automatiquement le fichier d'assignation (.ucf) des ports du FPGA.
- Une fenêtre d'assignation de pattes va s'ouvrir. On fait entrer, dans la section **Loc** du menu **Design Object List – I/O Pins**, les numéros des pattes reliant le FPGA au design.
- Afin de sauver les allocations effectuées, on sélectionne **File_Save**. Puis on sélectionne **XST Default** ; pour le type de sauvegarde. Et enfin, on clique sur **Ok**.
- On ferme la fenêtre d'assignation des pattes.
- Dans la fenêtre **Sources**, on sélectionne le fichier VHDL **VHDL_didacticiel– Behavioral (didactiel.vhdl)**.
- Dans la fenêtre **Processes**, on double-clique sur **Generate Programming File**. Après une attente de quelques minutes, on obtient un schéma indiquant que les étapes de synthèse ont bien été effectuées.

I- 6-Programmation du FPGA et tests

La démarche pour la programmation d'un circuit logique programmable est :

- On double-clique sur **Configure Device (IMPACT)**.
- Si un message d'avertissement s'affiche on clique sur **Cancel**.
- On sélectionne **Configure Devices using Boundary-Scan (JTAG)**.
- On clique sur **Finish**.
- On sélectionne le fichier de programmation créé (**VHDL_didactiel.bit** pour un design basé sur un fichier VHDL).
- On clique sur **Ok** si un message d'avertissement apparaît

I- Description de la plate-forme Basys 2

La carte Basys2 est une plate-forme de conception et de mise en œuvre des circuits numériques. Elle est construite autour d'un FPGA SPARTAN 3E Xilinx et d'un contrôleur USB AT90USB2 d'Atmel. Cette plate-forme est un matériel complet prêt à l'emploi convenant à l'hébergement des circuits allant des dispositifs logiques de base aux contrôleurs complexes. Elle est munie d'un large éventail de port E/S et de tous les supports FPGA nécessaires de sorte à pouvoir créer d'innombrables dessins et modèles sans avoir besoin d'autres composants. En outre, elle dispose de quatre connecteurs d'extension standards permettant de développer les conceptions au-delà de la carte Basys2 en utilisant des plaques d'essais, des cartes de circuit conçues par les utilisateurs ou des Pmods qui sont des modules d'E/S analogiques et numériques qui offrent la conversion analogique-numérique et numérique – analogique des signaux.

Les signaux présents sur les six connecteurs à 6 broches sont protégés contre les courts-circuits ; ce qui leur assurent une très longue vie dans tous les environnements.

Cette plate-forme peut être contrôlée et programmée via son port USB et elle est munie d'un câble USB qui est utilisé pour son alimentation.

Afin de transférer les informations de l'utilisateur vers la le FPGA, Digilent offre le logiciel « ADEPT » qui détecte automatiquement la carte Basys2 et qui constitue une interface de programmation pour le FPGA et pour la plate-forme flash ROM.

La carte Basys2 est conçue pour fonctionner avec toutes les versions de ISE Xilinx et même avec la version gratuite.

I- Outil de conversion Numérique/Analogique PmodDA2

I-1- Présentation du PmodDA2

Le pmodDA2 est un périphérique numérique qui permet d'effectuer la conversion d'un signal numérique en un signal analogique. Il est muni de deux petits convertisseurs numérique-analogique (DAC121S101) lesquels sont montés en parallèle de façon à recevoir les données en même temps. Il reçoit en entrée un signal numérique à 12 bits et donne en sortie un signal analogique.

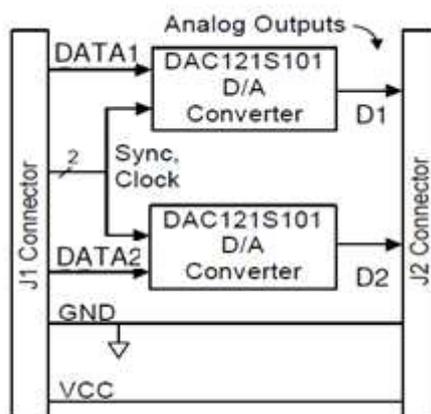


Figure.4.1. Bloc diagramme du PmodDA2

I-2-Constitution du PmodDA2

Il est constitué de cinq entrées et de cinq sorties (figure 1). En entrée, il dispose d'un signal « CLK » de 50MHz qui est utilisé comme horloge dans le processus du processus, d'un signal reset (RST) qui permet de réinitialiser le processus qui se déroule à l'intérieur du composant, de deux entrées de données qui sont des signaux à 12 bits (Data1, Data2) et enfin d'un signal « START » qui permet de déclencher la conversion. En sortie, il comprend un signal « clk_out » qui est le signal d'horloge divisé par 2 (25MHz), deux sorties série (D1 et D2) qui fournissent les données déplacées dans le PmodDA2, un signal « NSYNC » qui est utilisé pour verrouiller les données à l'intérieur du PmodDA2 après que ces dernières soient déplacées en sortie et un signal « DONE » qui indique quand la conversion est faite.

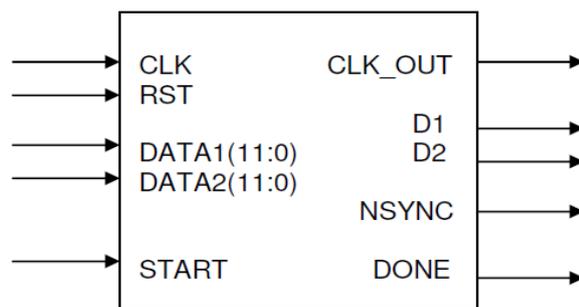


Figure.4.2.

I-3- Principe de fonctionnement du PmodDA2

Le fonctionnement de ce périphérique est représenté par trois machines à état (Idle, ShiftOut, SyncData) qui sont données comme suit :

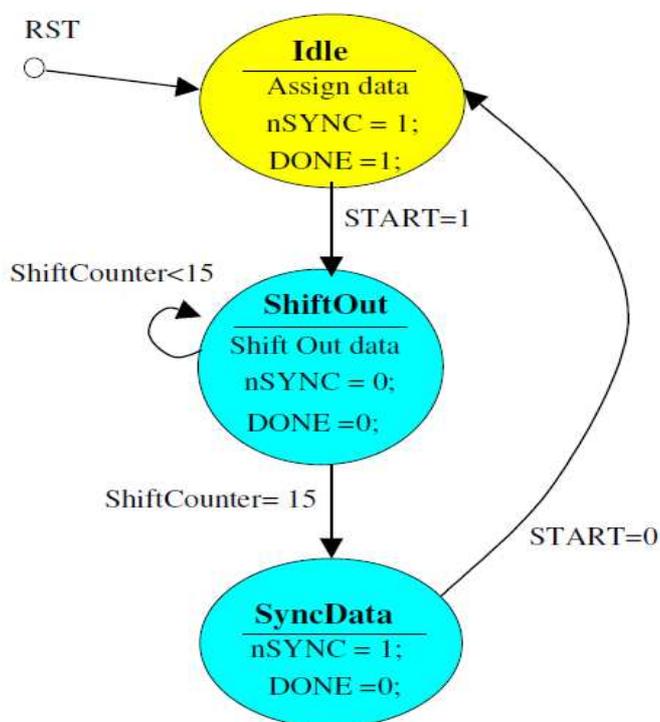


Figure.4.3. Principe de fonctionnement du PmodDA2

Au cours de l'état de repos, le vecteur de données à 12 bits est mis à jour à l'aide d'un registre de contrôle à 4 bits. Afin que la conversion se fasse, le signal de sortie « DONE » doit être à l'état « 1 ». Quand le signal « START » est à l'état haut, la machine à état passe à l'état « ShiftOut ». Dans cet état, les bits de données sont déplacés de la droite (MSB) vers la gauche (LSB) à l'aide d'un compteur modulo 16.

Quand le compteur atteint la valeur 15, la machine d'état passe à l'état « SyncData », au cours de laquelle les données sont verrouillées dans le Pmod. Quand « START » est à l'état zéro (0), on revient à l'état de repos (Idle).

Si le signal « RST » est à l'état haut, peu importe l'état où se trouve la machine à l'état, elle passe à l'état de repos (Idle).

I-4- Code VHDL du PmodDA2

Le code VHDL caractérisant le fonctionnement de ce périphérique est donné comme suit :

```

-----DA2RefComp-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity DA2RefComp is
  Port (

    --General usage
    CLK    : in std_logic;  -- System Clock (50MHz)
    RST    : in std_logic;
    START  : in std_logic;
    DATA1 : in std_logic_vector(11 downto 0);
    DATA2 : in std_logic_vector(11 downto 0);

    D1     : out std_logic;
    D2     : out std_logic;
    DONE   : out std_logic;
    nSYNC  : out std_logic;
    CLK_OUT : out std_logic);
end DA2RefComp ;

architecture DA2 of DA2RefComp is
-----
control constant : Normal Operation
-----
constant control : std_logic_vector(3 downto 0) := "0000";

```

Title : signal assignments

Description: The following signals are enumerated signals for the finite state machine, 2 temporary vectors to be shifted out to the DAC121S101 chips, a divided clock signal to drive the DAC121S101 chips, a counter to divide the internal 50 MHz clock signal, a 4-bit counter to be used to shift out the 16-bit register, and 2 enable signals for the parallel load and shift of the shift register.

```
type states is (Idle,ShiftOut,SyncData);
```

```
signal current_state : states;
```

```
signal next_state : states;
```

```
signal temp1 : std_logic_vector(15 downto 0);
```

```
signal temp2 : std_logic_vector(15 downto 0);
```

```
signal clk_div : std_logic;
```

```
signal clk_counter : std_logic_vector(27 downto 0);
```

```
signal shiftCounter : std_logic_vector(3 downto 0);
```

```
signal enShiftCounter: std_logic;
```

```
signal enParalelLoad : std_logic;
```

```
begin
```

Title : Clock Divider

Description: The following process takes a 50 MHz clock and divides it down to a 25 MHz clock signal by assigning the signals clk_out and clk_div to the 2nd bit of the clk_counter vector. clk_div is used by the Finite State Machine and clk_out is used by the DAC121S101 chips.

```
clock_divide : process(rst,clk)
```

```
begin
```

```
  if rst = '1' then
```

```
    clk_counter <= "00000000000000000000000000000000";
```

```
  elsif (clk = '1' and clk'event) then
```

```
    clk_counter <= clk_counter + '1';
```

```
  end if;
```

```
end process;
```

```

clk_div <= clk_counter(0);
clk_out <= clk_counter(0);

```

Title : counter

Description: This is the process where the temporary registers will be loaded and shifted. When the enParallelLoad signal is generated inside the state the temp1 and temp2 registers will be loaded with the 8 bits of control concatenated with the 8 bits of data. When the enShiftCounter is activated, the 16-bits of data inside the temporary registers will be shifted. A 4bit counter is used to keep shifting the data inside temp1 and temp 2 for 16 clock cycles.

```

counter : process(clk_div, enParallelLoad, enShiftCounter)
begin
  if (clk_div = '1' and clk_div'event) then
    if enParallelLoad = '1' then
      shiftCounter <= "0000";
      temp1 <= control & DATA1;
      temp2 <= control & DATA2;
    elsif (enShiftCounter = '1') then
      temp1 <= temp1(14 downto 0)&temp1(15);
      temp2 <= temp2(14 downto 0)&temp2(15);
      shiftCounter <= shiftCounter + '1';
    end if;
  end if;
end process;

```

```

D1 <= temp1(15);
D2 <= temp2(15);

```

Title : Finite State Machine

Description: This 3 processes represent the FSM that contains three states. First one is the Idle state in which the temporary registers are assigned the updated value of the input "DATA1" and "DATA2". The next state is the ShiftOut state which is the state where the 16-bits of temporary registers are shifted out left from the MSB to the two serial outputs, D1 and D2. Immediately following the second state is the third state SyncData. This state drives the output signal sync high for 2 clock signals telling the DAC121S101 to latch the 16-bit data it just received in the previous state.

Notes: The data will change on the upper edge of the clock signal. There is also an asynchronous reset that will reset all signals to their original state.

Title : SYNC_PROC

Description: This is the process where the states are changed synchronously. At reset the current state becomes Idle state.

```
-----
SYNC_PROC: process (clk_div, rst)
begin
  if (clk_div'event and clk_div = '1') then
    if (rst = '1') then
      current_state <= Idle;
    else
      current_state <= next_state;
    end if;
  end if;
end process;
```

Title : OUTPUT_DECODE

Description: This is the process where the output signals are generated asynchronously based on the state only (Moore State Machine).

```
-----
OUTPUT_DECODE: process (current_state)
begin
  if current_state = Idle then
    enShiftCounter <='0';
    DONE <='1';
    nSYNC <='1';
    enParallelLoad <= '1';
  elsif current_state = ShiftOut then
    enShiftCounter <='1';
    DONE <='0';
    nSYNC <='0';
    enParallelLoad <= '0';
  else --if current_state = SyncData then
    enShiftCounter <='0';
    DONE <='0';
    nSYNC <='1';
    enParallelLoad <= '0';
  end if;
end process;
```

Title : NEXT_STATE_DECODE

Description: This is the process where the next state logic is generated depending on the current state and the input signals.

```
NEXT_STATE_DECODE: process (current_state, START, shiftCounter)
```

```
begin
```

```
    next_state <= current_state; --default is to stay in current state
```

```
    case (current_state) is
```

```
        when Idle =>
```

```
            if START = '1' then
```

```
                next_state <= ShiftOut;
```

```
            end if;
```

```
        when ShiftOut =>
```

```
            if shiftCounter = x"F" then
```

```
                next_state <= SyncData;
```

```
            end if;
```

```
        when SyncData =>
```

```
            if START = '0' then
```

```
                next_state <= Idle;
```

```
            end if;
```

```
        when others =>
```

```
            next_state <= Idle;
```

```
    end case;
```

```
end process;
```

```
end DA2;
```

Résumé :

Les travaux de recherche présentés dans ce mémoire ont porté sur la réalisation d'un circuit de commande MLI adapté à la commande sans capteur par injection d'un signal à haute fréquence de la MSAP et son implantation sur un FPGA (Field Programmable Gate Array). Dans ce contexte, ce travail est scindé en 4 chapitres principaux :

- Le premier chapitre est divisé en trois parties : La première est consacrée à des généralités sur les circuits logiques programmables où on s'est intéressé plus particulièrement au FPGA, la seconde comprend la présentation détaillée du langage VHDL et la troisième un aperçu sur le logiciel de développement ISE.
- Le deuxième a porté sur les différentes stratégies MLI destinées à la commande des onduleurs de tension où on s'est intéressé plus précisément à la MLI sinusoïdale et à la MLI vectorielle.
- Le troisième a abordé l'analyse du besoin d'une commande MLI adaptée à la commande sans capteur de la MSAP.
- Enfin, dans le dernier chapitre on s'est étalé sur la réalisation et l'implémentation sur un FPGA du circuit MLI de la commande sans capteur avec injection d'un signal à haute fréquence.