REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université MOULOUD MAMMERI de TIZI-OUZOU

Faculté de génie électrique et d'informatique

Département d'Electronique

Mémoire de Magister en Electronique

Option : Microélectronique

Présenté par :

Mr KACI Yahia

Thème

Etude d'une nouvelle approche MLI (PWM) en temps réel basée sur le principe des réseaux de neurones destinée aux véhicules électriques

Soutenue le 15 Janvier 2015 devant le Jury :

Mr MEGHERBI Mohamed : Professeur	Président, UMMTO.
Mr BELKAID Mohamed Saïd : Professeur	Directeur de mémoire, UMMTO.
Mr HAMICHE Hamid : Maître de conférences A	Examinateur, UMMTO.
Mr MELLAH Rabah : Maître de conférences A	Examinateur, UMMTO.
Mr ZIRMI Rachid : Maître de conférences B	Examinateur, UMMTO.

Remerciement

Je remercie le Bon Dieu de m'avoir donné la volonté et la patience qui m'ont permis de mener à bien ce travail.

Je remercie vivement mon Promoteur, Monsieur M.S.BELKAID, Professeur à l'université Mouloud Mammeri de Tizi-Ouzou pour son suivi, son aide documentaire et son soutien matériel précieux, je voudrais aussi remercier Monsieur A. GUELLAL chercheur au niveau de laboratoire de l'équipe économie et maîtrise d'énergie au Centre de Recherche des Energies Renouvelables (CDER) de Bouzaréa pour son aide précieuse, son soutien durant la préparation de ce travail.

Je tiens à adresser mes sincères remerciements à tous les membres de jury qui ont accepté de juger ce travail et d'y apporter leur caution chargé d'examiner la soutenance de mon projet de magister.

Je voudrai exprimer mon profond respect à tous les Enseignants qui m'ont encadré durant mon étude.

Je ne saurai oublier de remercier toute personne qui, d'une manière ou d'une autre, m'a aidé dans l'élaboration de ce travail.

Dédicace

Je dédie ce travail à :

Mes parents,

Mes frères et mes soeurs,

Toute ma famille,

Ainsi qu'à tous mes amis.

Sommaire

Liste des tableaux

Listes des figures

1. Liste des acronymes

DSP : Digital Signal Processor.

NPC : Neutral-Point-Clamped.

MAS : Machine Asynchrone.

FEM : Force magnétomotrice.

MLI : Modulation à Largeur d'Impulsion.

PWM : Pulse Width Modulation.

ROM: Read Only Memory.

SPWM: Sinus Pulse Width Modulation.

SVM: Space Vector Modulation.

DDT : Direct Digital Technique.

PSIM : Simulateur des circuits électroniques.

THD : distorsion d'harmonique total.

MLP : Topologie multicouche.

tansig : tangent sigmoïde.

Purelin : Forme purement linéaire.

mapminmax : Fonction de normalisation directe.

mapminmax_reverse : Fonction de normalisation inverse.

FPGA :Field-Programmable Gate Array.

CLB: Configurable Logic Block.

IOB: Input Output Bloc.

PIP : Programme Interconnect Point.

ASIC : Application Specific Integrated Circuit.

LUT : Look Up Table.

SRAM : Static Random Access Memory.

MUX : Multiplexeurs.

RISC: Reduced Instruction Set Computer.

HDL :Hardware Description Langages.

VHDL: Very high speed integrated circuit Hardware Description Language.

VHSIC : Very High Speed Integrated Circuits.

IEEE : Institute of Electrical and Electronics Engineers.

DDR : Double Data Rate.

FBGA: Fine Ball Grid Array.

PROM**:** Programmable Read Only Memory.

CPLD : complex programmable logic device.

SDRAM: Synchronous Dynamic Random Access Memory.

LCD**:** liquid crystal display.

EEPROM: Electrically-Erasable Programmable Read-Only Memory.

LED **:** Light-Emitting Diode.

MOSFET : Metal–Oxide–Semiconductor Field-Effect Transistor.

IGBT :Insulated Gate Bipolaire Transistor. THD : Total Harmonic Distorsion.

MLP : Réseau multicouche.

2. Liste des nomenclatures

a,b,c :Indice correspondant aux trois phases de la machine.

d,q : Indice correspondant au référentiel lié au champ tournant.

α, β : Indice correspondant au référentiel fixe.

R_s ,R_r :Résistance du stator et du rotor.

L_s ,L_r :Inductances cycliques du stator et du rotor.

ϕ_s :Flux magnétique du stator.

ϕ_r :Flux magnétique du rotor.

M_{sr} : Inductance mutuelle cyclique entre le stator et le rotor.

I_s :Courant dans le stator.

I_r :Courant dans le rotor.

$T\max$:Couple maximum.

Lre : l'inductance du rotor ramenée au stator.

P: Nombre de paires de poles.

Ω_s : la vitesse de synchronisme en rad/s.

Ω : la vitesse mécanique en rad/s.

ω_s : Pulsation statorique.

ω_r : Pulsation rotorique.

g : Glissement.

θ : Position du référentiel par rapport au stator.

N_s : la vitesse de synchronisme en tr/mn.

N : la vitesse mécanique en tr/mn.

CEM : Couple électromagnétique.

Cr :Couple résistant.

T_{max} : Couple maximum.

V : Tension de sortie de l'onduleur.

f : Fréquence de signal de sortie de l'onduleur.

f_0 : Fréquence de signal d'alimentation.

f_s : Fréquence de synchronisme.

Id : courant continu.

m : Le nombre des niveaux de tension de sortie dans un onduleur en cascade.

s : nombre des sources des tensions continues.

r : Taux de modulation.

α_1, α_2 : Angles de commutation.

S : seuil.

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

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

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

J : fonction des moindre carré.

W : Matrice de poids.

K : Numéro de la couche.

$Err_i^{(k)}$: l'erreur du $i^{ème}$ neurone dans la $k^{ème}$ couche.

W1 : Matrice des poids de la couche cachée.

b1 : Matrice des seuils de la couche cachée.

W2 : Matrice des poids de la couche de sortie.

b2 : Matrice des seuils de la couche de sortie.

f 1 : Fonction tangente-sigmoïde (tansig).

f 2 : Fonction purement linéaire (purelin).

a : La sortie de la couche cachée.

r_n : Taux de modulation normalisée.

k1, k2 : Sorties de la couche de sortie.

t1, t2 : Les instants de commutations.

$\propto_{1exacte}$: La valeur exacte de l'angle de commutation $\propto_1$.

$\propto_{2exacte}$: La valeur exacte de l'angle de commutation $\propto_2$.

$\propto_{1approximé}$: La valeur approximé par le système neuronal de l'angle de commutation $\propto_1$.

$\propto_{2approximé}$: La valeur approximé par le système neuronal de l'angle de commutation $\propto_2$.

Err$\propto_1$: L'erreur commise sur le calcul de $\propto_1$.

Err$\propto_2$: L'erreur commise sur le calcul de $\propto_2$.

Introduction générale

La production d'énergie et les multiples problèmes qui y sont liés demeurent une préoccupation constante que ce suit au niveau technique qu'un niveau de sur utilisation.

Les énergies fossiles telles que le charbon, le pétrole et le gaz naturel (ce sont des énergies dont la durée de vie est tout de même limitée même si les spécialistes différent dans leur prévision quant à la durée prévue de leur épuisement.

De plus ces matières premières, présentent lors de leur utilisation, des problèmes qui seront difficiles à résoudre si l'on continue leur exploitation telle qu'elle est à l'état actuel. Elles sont, d'après les statistiques, à l'origine de l'ordre de 40% des émissions mondiales de CO_2. Le problème de l'effet de serre s'il n'est pas pris en charge de façon sérieuse et courageuse peut provoquer une élévation de la température de la terre qui provoquerait des problèmes importants sur la fonte des glace, le relèvement de niveaux des océans impliquant des problèmes majors.

De ce fait et depuis, notamment les dernières décennies, la prise de conscience de tous ces problèmes suscité des réflexions qui ont incité au niveau de la recherche scientifique à trouver des solutions qui permettront d'assurer l'avenir de l'approvisionnement énergétique. Plusieurs domaines de recherches sont aussi abordés et font l'objet de travaux intenses ; Ceux-ci sont liés principalement aux énergies renouvelables telles que la conversion solaire photovoltaïque, l'énergie thermique, l'éolien, la biomasse etc,…

C'est dans le cadre de ces applications diverses que nous nous fixons une contribution dans le domaine de l'électronique appliqué au fonctionnement des véhicules électriques.

En effet toute contribution consistant à utiliser une autre énergie que l'énergie fossile, comme l'utilisation d'une autre source d'énergie que l'essence et le gaz oïl dans le véhicule, peut être d'un apport positif.

Dans ce cadre, le systéme de propulsion électrique est le cœur du véhicule électrique(VE). Celui-ci est constitué d'un actionneur électrique, un dispositif de transmission, et des roues. L'entraînement, qui est l'ensemble du moteur électrique et des convertisseurs statiques, associé à une commande électronique, est le noyau du système de propulsion dans le VE

Le moteur asynchrone peut fonctionner sur une grande plage de variation de vitesse avec de faibles ondulations de couple s'il est associé à une commande adéquate. Parmi les techniques de commande des véhicules électriques, la commande vectorielle est actuellement parmi les plus employée. Dans ce cas, la commande du couple est appliquée au régime transitoire et permet d'avoir de meilleures performances dynamiques. La

commande scalaire est utilisée en régime permanent ; on distingue plusieurs stratégies de commande : la stratégie de commande pleine onde, la modulation tout ou rien, la modulation sigma-delta, la modulation de largeur d'impulsion MLI (PWM) triangulo-sinusoïdale, la modulation MLI calculée, etc…

Dans le cadre de notre travail, nous nous intéresserons à la technique MLI calculée.

Nous nous fixons pour objectif l'implémentation sur un circuit FPGA d'une commande MLI *'on-line'* avec asservissement du fondamental et élimination sélective des harmoniques, pour une application dans les véhicules électriques. Le principe de cette commande est inspiré de la technique de Patel et Hoft mais les angles de commutation sont calculés en se basant sur le principe des réseaux de neurones.

Nous présenterons le travail de notre présent mémoire selon quatre chapitres.

Dans le premier nous exposons les aspects théoriques des machines asynchrones et les différentes topologies des onduleurs de tension ainsi que les techniques de commandes des onduleurs de tension en relation avec le but que nous nous sommes assigné.

Le deuxième chapitre sera consacré à la commande d'un onduleur triphasé piloté par MLI à structure NPC (Neutral Point Clamping) multiniveaux ; nous traiterons la modélisation de l'onduleur triphasé cinq niveaux, nous procèderons ensuite à l'élaboration du modèle de simulation de ce dernier dans l'environnement MATLAB/SIMULINK et nous terminerons par la présentation des résultats obtenus par simulation.

Le troisième chapitre présente les aspects théoriques des réseaux de neurones ainsi que leurs caractéristiques principales; en nous basant sur ces derniers nous nous sommes attaché à élaborer l'architecture d'un réseau de neurones artificiel qui génère une commande MLI.

Le quatrième chapitre sera consacré à l'implémentation de la commande MLI neuronale *on-line* sur FPGA en commençant par la présentation des circuits FPGA et leurs caractéristiques principales ainsi que les langages de description matérielle HDL (Hardware Description Langages) ; le test des performances de la commande MLI *on-line* se fera au niveau du Laboratoire de l'équipe Economie et Maîtrise d'Energie au Centre de Recherche des Energies Renouvelables CDER de Bouzaréah (Alger).

S'ensuivra, après l'interprétation générale une conclusion et quelques perspectives relatives au travail que nous avons accompli.

Machines asynchrones et onduleurs de tension

1. Véhicules électriques

Une voiture électrique est une automobile (Figure 1.1), le cœur de ce véhicule est le moteur électrique, alimenté par une batterie d'accumulateurs. En l'état actuel de la technologie les batteries permettent difficilement d'assurer une autonomie suffisante, et nécessite des temps de recharge longs (quelques heures). Certains véhicules électriques sont donc munis de générateurs électriques internes : moteur thermique classique assurant selon la situation une partie de la traction ou une fonction de groupe électrogène (véhicule "hybride"), pile à combustible ou éventuellement des panneaux solaires intégrés à la carrosserie pour des véhicules spécialement économes en consommation énergétique [2].

.

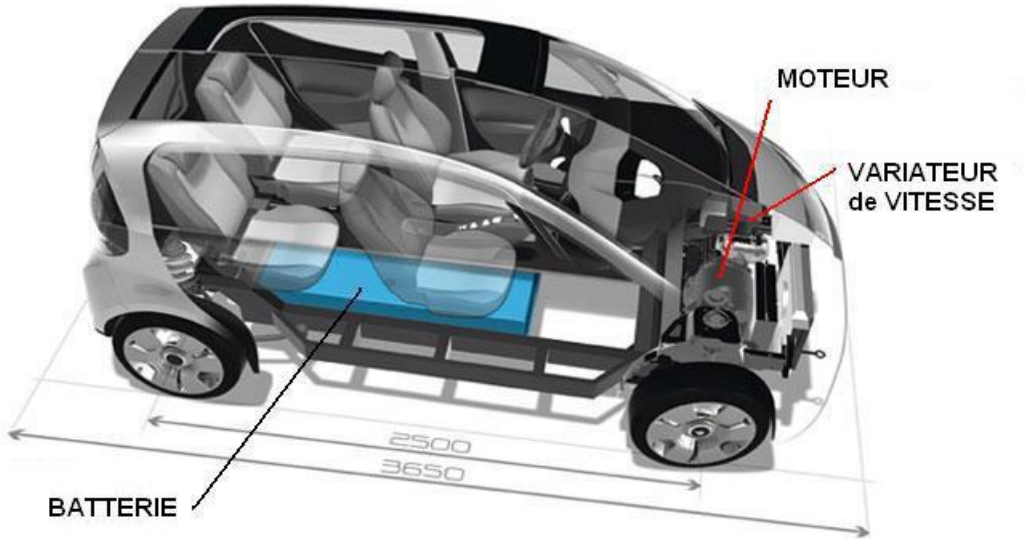


Figure 1.1. Représentation schématique d'un véhicule électrique

2. La machine asynchrone (MAS)

2.1. Introduction

La MAS a de nombreux avantages par rapport aux autres types de machines électriques tournantes, parmi lesquelles nous pouvons citer : la robustesse, le prix relativement bas et l'entretien moins fréquent. La MAS est aujourd'hui la plus utilisée dans les applications industrielles où la variation de vitesse, une haute précision de régulation et de hautes performances en couple sont requises. Cependant il faut noter que ces avantages ont longtemps été inhibés par la complexité de la commande due au couplage non linéaire existant entre le flux magnétique et le couple moteur. L'utilisation à grande échelle à l'heure actuelle est due à l'évolution technologique, notamment en matière de semi-conducteurs. Par

ailleurs, pour élaborer des approches de commande assurant les performances espérées, nous avons besoin d'un modèle reflétant le fonctionnement de la machine.

2.2. Constitution et principe de fonctionnement

La machine asynchrone est constituée de trois enroulements (bobines) parcourus par des courants alternatifs triphasés et possède p paires de pôles ; les courants alternatifs dans le stator créent un champ magnétique tournant à la pulsation de synchronisme :

$$\Omega_s = \frac{\omega}{p} \qquad \text{[rd/s]} \qquad (1.1)$$

Le rotor n'est relié à aucune alimentation. Il tourne à la vitesse de rotation Ω plus petite que la vitesse de synchronisme Ω_s .

$$\Omega = (1-g)\times\Omega_s \qquad \text{[rd/s]} \qquad (1.2)$$

$$\omega = 2\times\prod\times f \qquad \text{[rd/s]} \qquad (1.3)$$

On dit que le rotor glisse par rapport au champ tournant, ce glissement g va dépendre de la charge

$$g = \frac{N_s - N}{N_s} = \frac{\Omega_s - \Omega}{\Omega_s} \qquad (1.4)$$

2.3. Modélisation de la machine asynchrone

Dans le cadre de la modélisation de la machine asynchrone nous avons opté pour les hypothèses simplificatrices, conventionnellement utilisées suivantes [3] :

- L'entrefer est d'épaisseur uniforme et l'effet d'encochage est négligeable.
- La saturation du circuit magnétique, l'hystérésis, les courants de Foucault et l'effet de peau sont négligeables.
- Les résistances des enroulements ne varient pas avec la température.
- On admet que la force magnétomotrice (FEM) créée par chacune des phases des deux armatures est à répartition sinusoïdale.

La MAS triphasée, représentée schématiquement par la figure (Figure 1.2), est munie de six enroulements.

- Le stator de la machine est formé de trois enroulements fixes décalés de 120° dans l'espace et traversés par trois courants triphasés.
- Le rotor peut être modélise par trois enroulements identiques court-circuités dont la

tension aux bornes de chaque enroulement est nulle.

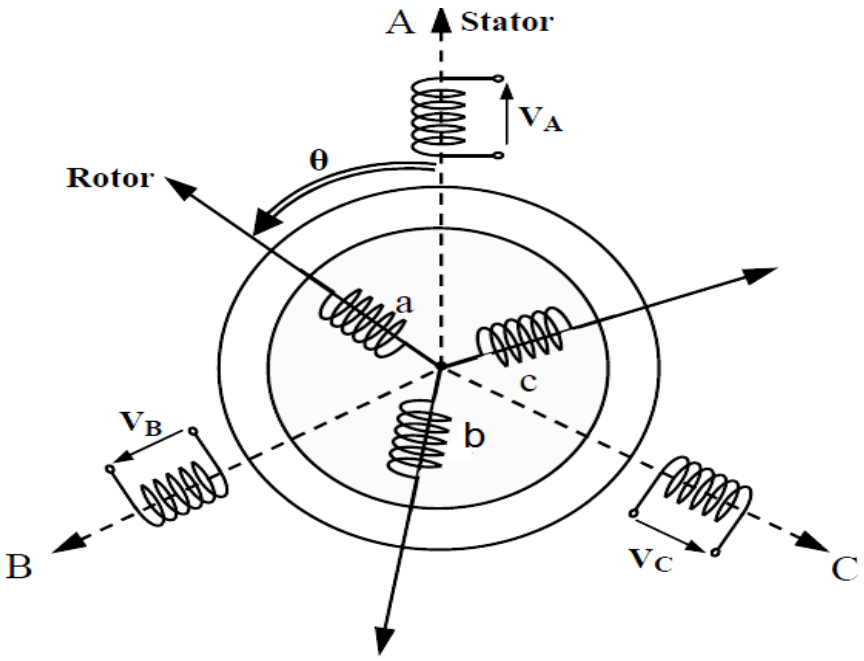


Figure 1.2. Représentation schématique d'une machine asynchrone triphasée

2.3.1. Equations électriques :

Par application de la loi de Faraday à chaque enroulement, on peut écrire :

$$[V_{sabc}] = [R_s] \times [I_{sabc}] + \frac{d}{dt}[\phi_{sabc}] \tag{1.5}$$

$$[V_{rabc}] = [R_r] \times [I_{rabc}] + \frac{d}{dt}[\phi_{rabc}] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Avec

$$[V_{sabc}] = [V_A \quad V_B \quad V_C]^T \; ; \; [I_{sabc}] = [I_A \quad I_B \quad I_C]^T \; ; \; [\phi_{sabc}] = [\phi_A \quad \phi_B \quad \phi_C]^T$$

$$[V_{rabc}] = [V_a \quad V_b \quad V_c]^T \; ; \; [I_{rabc}] = [I_a \quad I_b \quad I_c]^T \; ; \; [\phi_{rabc}] = [\phi_a \quad \phi_b \quad \phi_c]^T$$

Les matrices des résistances statoriques et rotoriques de la MAS sont données par :

$$[R_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} ; \qquad [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}^T$$

2.3.2 Equations magnétiques :

Les hypothèses que nous avons présentées conduisent à des relations linéaires entre les flux et les courants. Elles sont exprimées sous forme matricielle comme suit :

$$[\phi_{sabc}] = [L_s] \times [I_{sabc}] + [M_{sr}] \times [I_{rabc}]$$

$$[\phi_{rabc}] = [M_{rs}] \times [I_{sabc}] + [L_r] \times [I_{rabc}] \quad (1.6)$$

Les différentes matrices d'inductances s'écrivent :

$$[L_s] = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix}; \qquad [L_r] = \begin{bmatrix} l_r & M_r & M_r \\ M_r & l_r & M_r \\ M_r & M_r & l_r \end{bmatrix}$$

$$[M_{sr}] = [M_{rs}]^T = M \begin{bmatrix} \cos(\theta) & \cos(\theta + 2\Pi/3) & \cos(\theta + 4\Pi/3) \\ \cos(\theta + 4\Pi/3) & \cos(\theta) & \cos(\theta + 2\Pi/3) \\ \cos(\theta + 2\Pi/3) & \cos(\theta + 4\Pi/3) & \cos(\theta) \end{bmatrix}$$

On obtient finalement le modèle de la machine asynchrone triphasée suivant :

$$[V_{sabc}] = [R_s][I_{sabc}] + \frac{d}{dt}([L_s] \times [I_{sabc}] + [M_{sr}] \times [I_{rabc}]) \quad (1.7)$$

$$[V_{rabc}] = [R_r][I_{rabc}] + \frac{d}{dt}([M_{rs}] \times [I_{sabc}] + [L_r] \times [I_{rabc}]) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (1.8)$$

3. Commande des moteurs asynchrones

3.1. Introduction

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

La commande scalaire est basée sur le modèle en régime permanent, elle est simple à implanter avec une dynamique lente. Elle contrôle les grandeurs en amplitude.

La commande vectorielle est basée sur le modèle transitoire, elle est précise et rapide, elle permet le contrôle du couple à l'arrêt ; elle est cependant chère (encodeur incrémental ou estimateur de vitesse, DSP…). Elle contrôle les grandeurs en amplitude et en phase.

3.2. Commande scalaire

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

3.2.1. Contrôle en *V/f* de la machine asynchrone

Son principe est de maintenir *V/f* constant, ce qui signifie qu'il faut garder le flux maximal constant. Le contrôle du couple se fait par l'action sur le glissement.

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

$$T\max = \frac{3pV^2}{4\pi f\,(Rs + \sqrt{Rs^2 + [2\pi f\,(Ls + Lre)]^2})} \tag{1.9}$$

Avec Rs et Ls la résistance et l'inductance du stator, Lre l'inductance du rotor ramenée au stator, p le nombre de paires de pôles, V la tension efficace d'entrée du moteur (d'une phase) et f est la fréquence de la tension d'alimentation.

Puisque le couple maximum est constant, donc la caractéristique couple en fonction de la vitesse (figure 1.3) glisse de la gauche vers la droite avec l'augmentation de la vitesse, l'intersection du couple moteur avec le couple résistant Cr détermine le point de fonctionnement du moteur électrique.

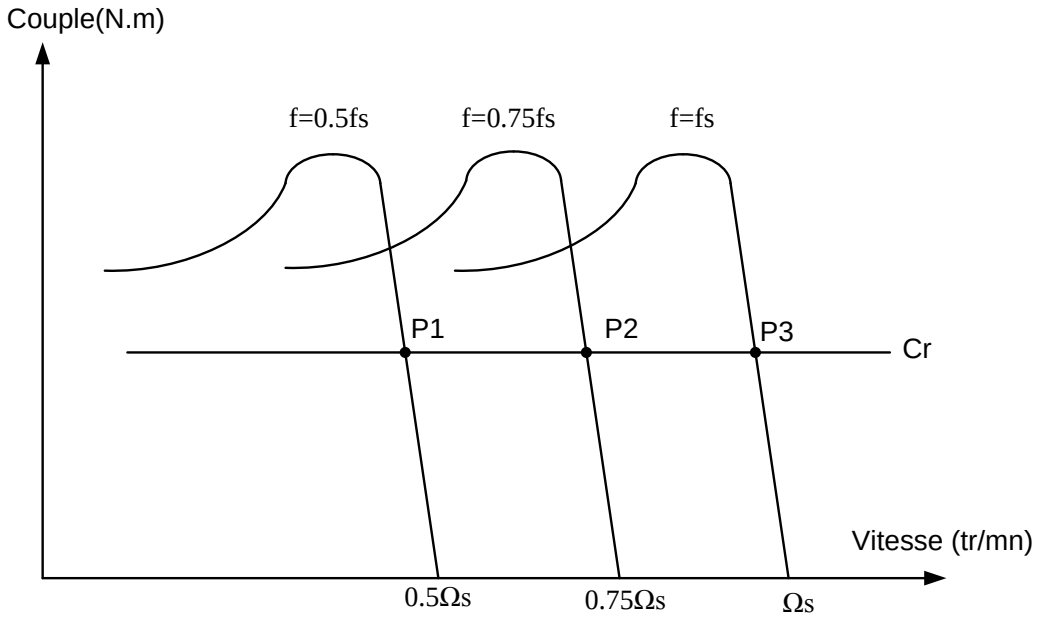


Figure 1.3. Caractéristique couple en fonction de la vitesse.

3.2.2. Contrôle scalaire du courant

La différence avec la commande précédente est que c'est un onduleur (commutateur) de courant qui est utilisé. On impose directement des courants dans les phases de la machine. La valeur du courant *Id* (courant continu) est égale, à une constante près, à la valeur efficace du courant imposé *Is*. Cette valeur est imposée par régulation à l'aide d'un pont redresseur contrôlé.

3.3. Commande vectorielle

La commande vectorielle a été introduite depuis longtemps. Cependant, elle n'a pu être implantée et utilisée réellement qu'avec les avancées en micro-électronique. En effet, elle nécessite des calculs de Transformé de Park, évaluation de fonctions trigonométriques, des intégrations, des régulations… ce qui ne pouvait pas se faire en pure analogique.

Le contrôle de la machine asynchrone requiert le contrôle du couple, de la vitesse ou même de la position.

4. Onduleur de tension

4.1. Définition et principe

Un onduleur est un dispositif permettant de transformer en alternatif une énergie électrique de type continue. Ils sont utilisés en électrotechnique pour :

- Soit fournir des tensions ou courants alternatifs de fréquence et amplitudes variables.

C'est le cas des onduleurs servant à alimenter des moteurs à courant alternatif devant tourner à vitesse variable par exemple (la vitesse est liée à la fréquence des courants qui traversent la machine).

- Soit fournir une ou des tensions alternatives de fréquence et d'amplitude fixes.

C'est le cas, en particulier, des alimentations de sécurité destinées à se substituer au réseau en cas de défaillance de celui-ci par exemple. L'énergie stockée dans les batteries de secours est restituée sous forme continue, l'onduleur est alors nécessaire pour recréer la forme de tension et fréquence du réseau. On distingue les onduleurs de tension et les onduleurs de courant, en fonction de la source d'entrée continue : source de tension ou source de courant. La technologie des onduleurs de tension est la plus maîtrisée et elle présente dans la plupart des systèmes industriels, dans toutes les gammes de puissance (quelques Watts à plusieurs MW).

4.2. Types d'onduleurs

4.2.1. Les Onduleurs monophasés de tension

Ce type d'onduleurs est destiné à alimenter des charges alternatives monophasées, on distingue deux configurations de base: en demi-pont ou en pont complet.

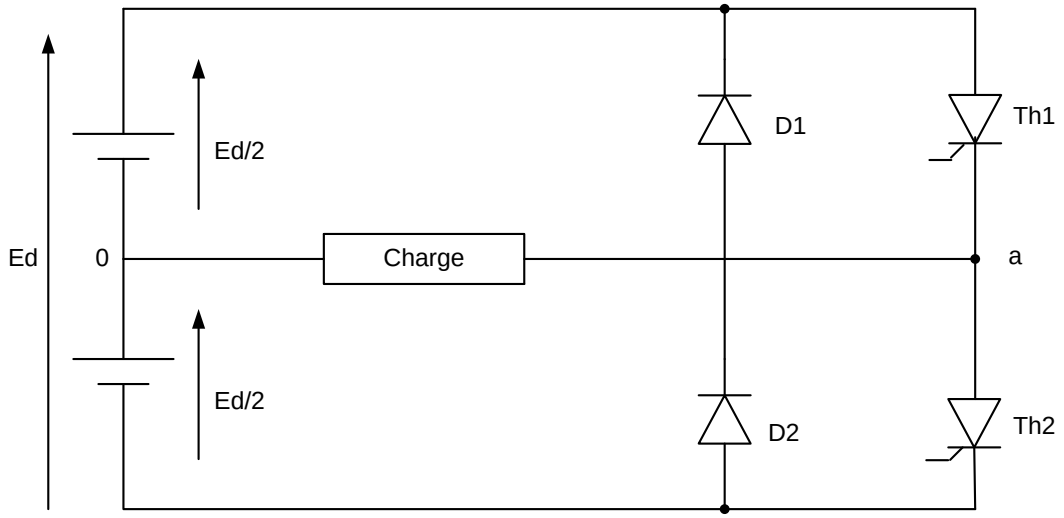


Figure 1.4. Schéma de l'onduleur demi-pont.

4.2.2. Les onduleurs triphasés

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

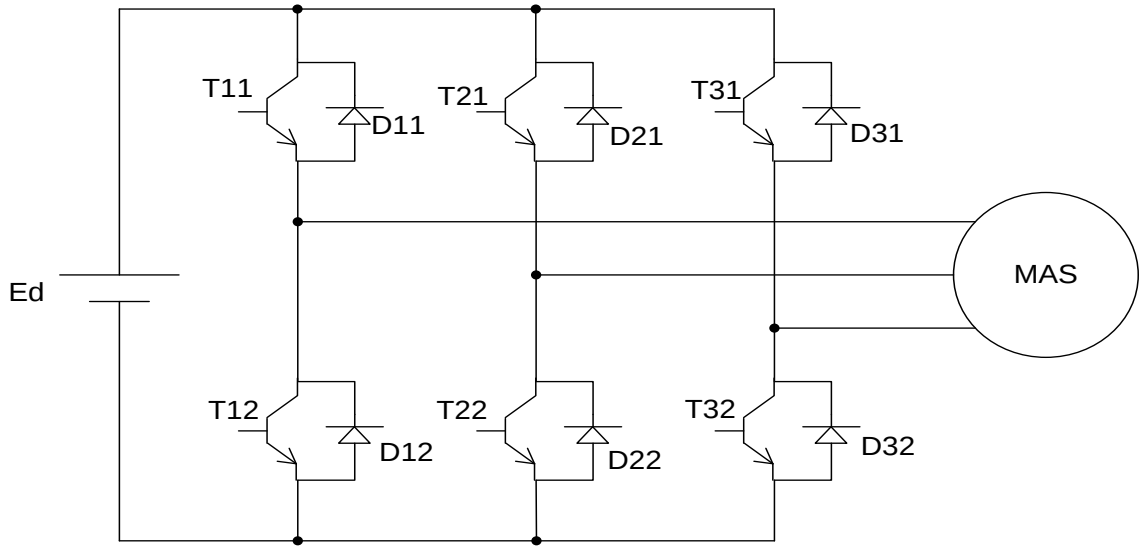


Figure 1.5. Schéma de principe d'un onduleur triphasé de tension

4.2.3. Onduleurs multi-niveaux

Par définition, l'onduleur de tension multi-niveaux possède trois ou plusieurs niveaux. L'objectif de cette partie est de donner une vue générale des trois topologies de base des onduleurs multi-niveaux: la topologie à diode de bouclage, la topologie au condensateur flotteur et la topologie en cascade.

4.2.3.1. Onduleur de tension à diode de bouclage (structure NPC)

La première topologie la plus pratique d'onduleur de tension multi-niveaux est le NPC (Neutral-Point-Clamped). Elle a été proposée, la première fois en 1981, par Nabae et al [4]. L'onduleur NPC à trois niveaux est donné par la (Figure 1.6).

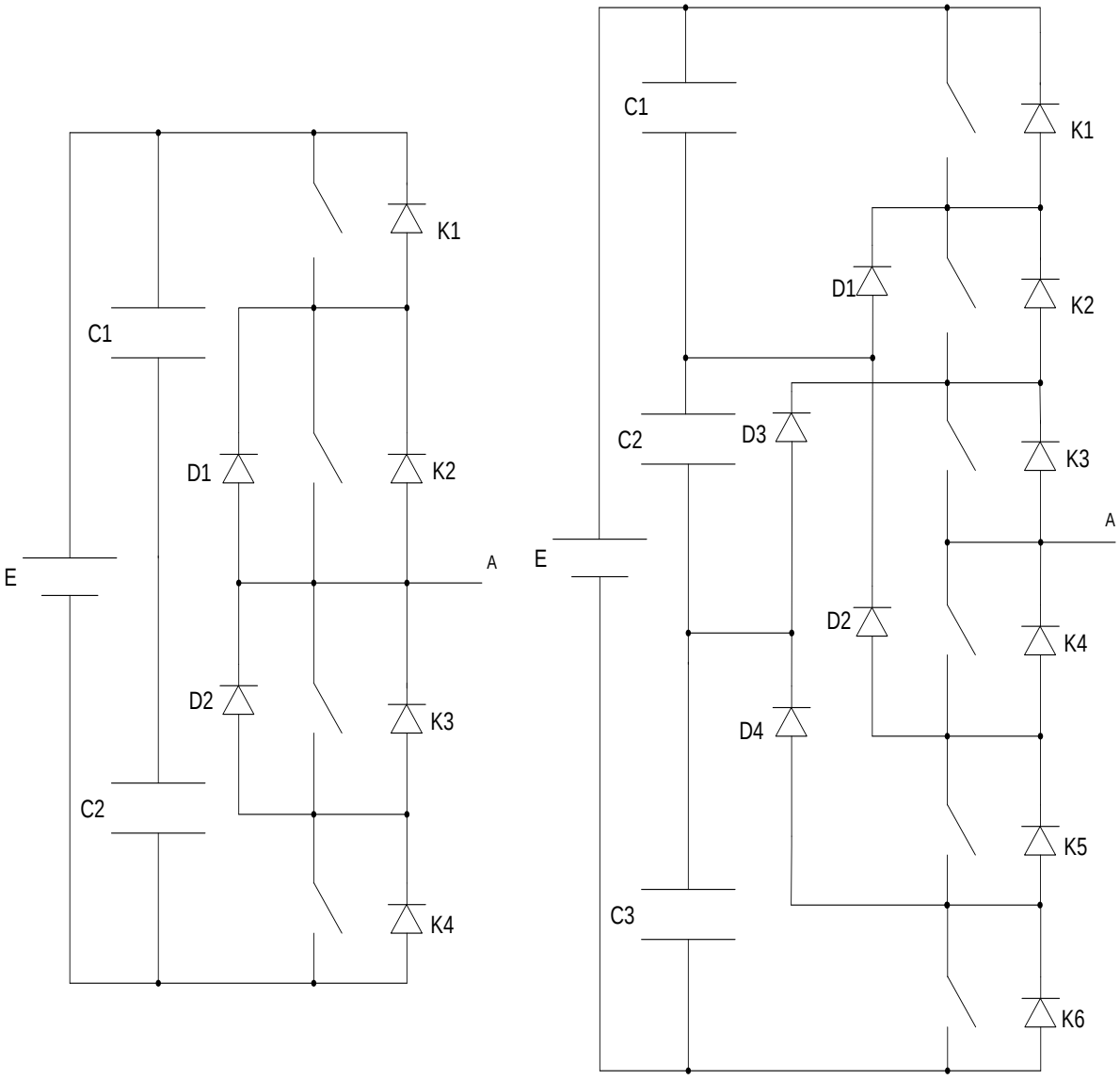


Figure 1.6. Onduleurs à structure NPC à trois et à quatre niveaux (phase A)

4.2.3.2. Onduleur de tension à condensateur flotteur

La topologie de l'onduleur multi-niveaux à condensateur flotteur (flying capacitor multilevel inverter), donnée par la Figure 1.7, a été proposée en 1992[4]. Elle est considérée comme l'alternative la plus sérieuse à la topologie de l'onduleur NPC. L'avantage de cette topologie est qu'elle élimine le problème des diodes de bouclage présent dans les topologies des onduleurs NPC multi-niveaux. En plus, cette topologie limite naturellement les contraintes en tension imposées aux composants de puissance (faible valeur de dv/dt aux bornes des composants) et introduit des états de commutation additionnelles qui peuvent être utilisés pour aider à maintenir l'équilibre des charges dans les condensateurs. La topologie de l' onduleur à condensateur flotteur a assez d'états de commutation pour contrôler l'équilibre des charges dans chaque bras d'onduleur ayant n'importe quel nombre de niveaux, ce qui n'est pas le cas dans l' onduleur NPC.

Actuellement il semble que cette topologie a quelques inconvénients. Néanmoins, quelques points faibles doivent toujours être explorés:

- ✓ le contrôleur de la charge du condensateur ajoute la complexité au contrôle du circuit entier;
- ✓ la topologie de l'onduleur à condensateur flotteur à multi-niveaux peut exiger plus de condensateurs que la topologie de l'onduleur NPC. De plus, il est évident que des courants de grandes valeurs efficaces circuleront à travers ces condensateurs;
- ✓ il y a un potentiel de résonance parasite entre les condensateurs découplés.

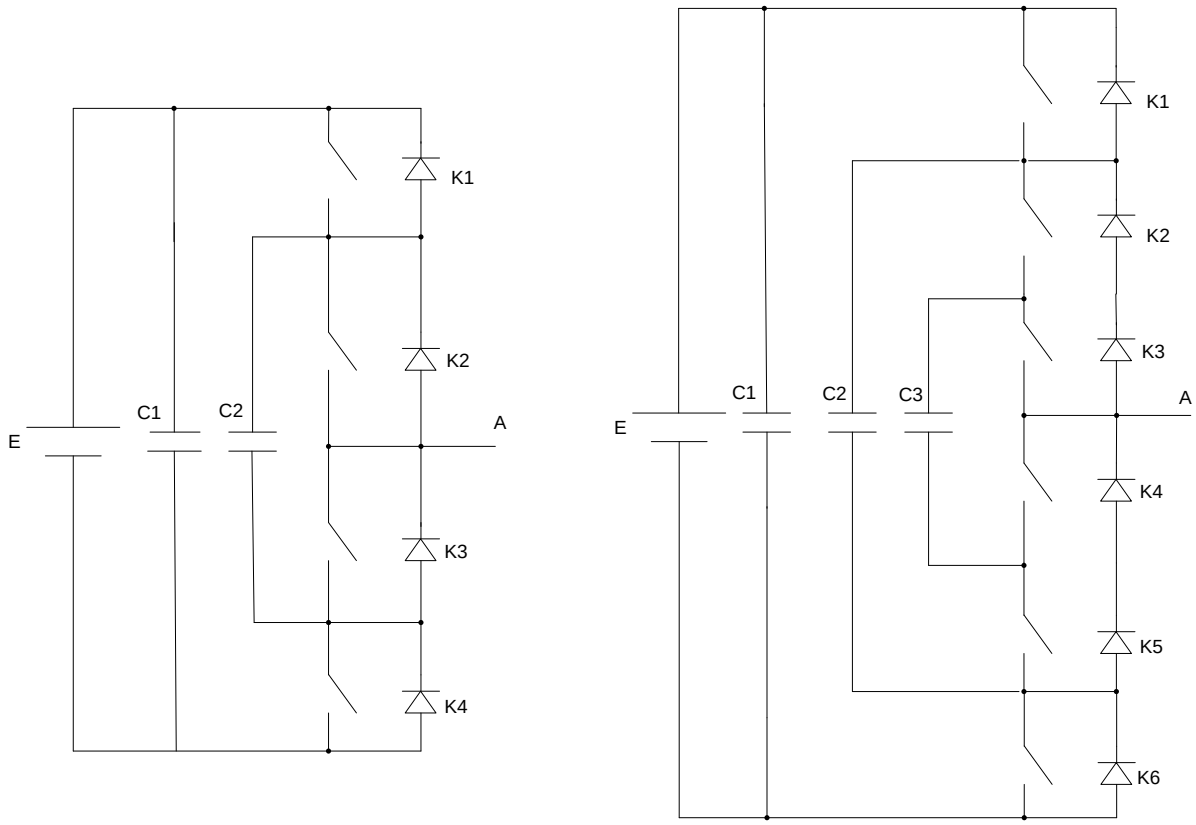


Figure 1.7. Onduleurs à condensateurs flotteurs à trois et à quatre niveaux (phase A)

4.2.3.3. Onduleur de tension en cascade (topologie en H)

Une des premières applications des connexions en série des topologies des convertisseurs monophasés en pont était pour la stabilisation de plasma en 1988. Cette approche modulaire a été étendue pour inclure aussi les systèmes triphasés. Sans conteste, les complications et le coût des sources isolées pour chaque pont n'est pas un inconvénient sérieux parce qu'il est compensé par les avantages de la construction modulaire. L'avantage principal de cette approche est que la topologie de ce type d'onduleur facilite la maintenance ; de plus, elle permet de donner une façon très pratique pour augmenter le nombre de niveaux dans le système. La Figure 1.8 représente un onduleur monophasé en cascade à cinq niveaux.

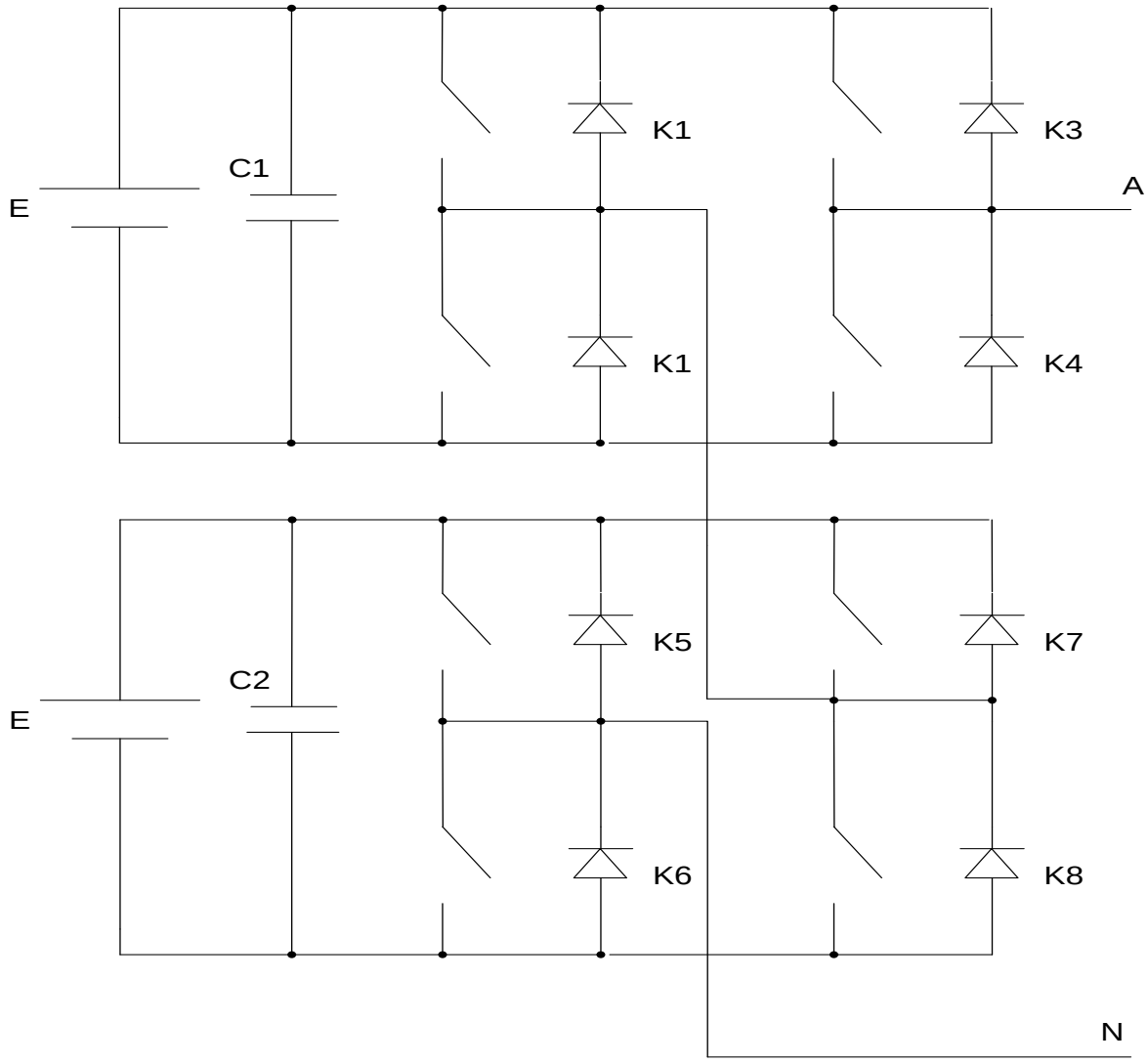


Figure 1.8. Onduleur en cascade de forme H à 5 niveaux (phase A).

Les sorties des onduleurs en pont sont connectées en série telle que l'onde de la tension synthétisée est la somme des tensions de sortie. Le nombre des niveaux de tension de sortie dans un onduleur en cascade est définie par:

$$m=2s+1 \tag{1.10}$$

où s est le nombre des sources des tensions continues.

L'avantage majeur de cette approche hybride est que le nombre de sortie peut être augmenté davantage sans aucun ajout de nouveaux composants. Il faut seulement des sources de tensions continues avec différents niveaux de tensions.

5. Types de commande des onduleurs

Le rôle de la fonction de modulation est de déterminer les instants de commutation et les ordres de commande logique des interrupteurs afin d'obtenir une séquence de commutation de ces derniers. Le choix d'une stratégie de modulation peut s'effectuer en fonction des performances souhaitées par l'utilisateur et toute les stratégies ont des avantages et des inconvénients et peuvent être réalisées par programmation logicielle ou matérielle.

L'ensemble de ces stratégies sont [5]:

a. **Stratégie de modulation Tout ou rien :** cette stratégie est basée sur le principe de comparaison de deux grandeurs (tension ou courant) par des comparateurs afin de générer les ordres de commande des bras de l'onduleur.

b. **Stratégie de modulation à pleine onde :** le principe de cette méthode est de commander les bras de l'onduleur tous les tiers de période.

c. **Stratégie de modulation à largeur d'impulsion MLI :** le principe de cette stratégie est de commander les bras de l'onduleur par une décision livrée par un algorithme au début de chaque période d'échantillonnage.

d. **Stratégie de modulation Sigma-Delta :** le principe de cette stratégie est de commander les bras de l'onduleur par une décision livrée par un algorithme durant chaque période d'échantillonnage.

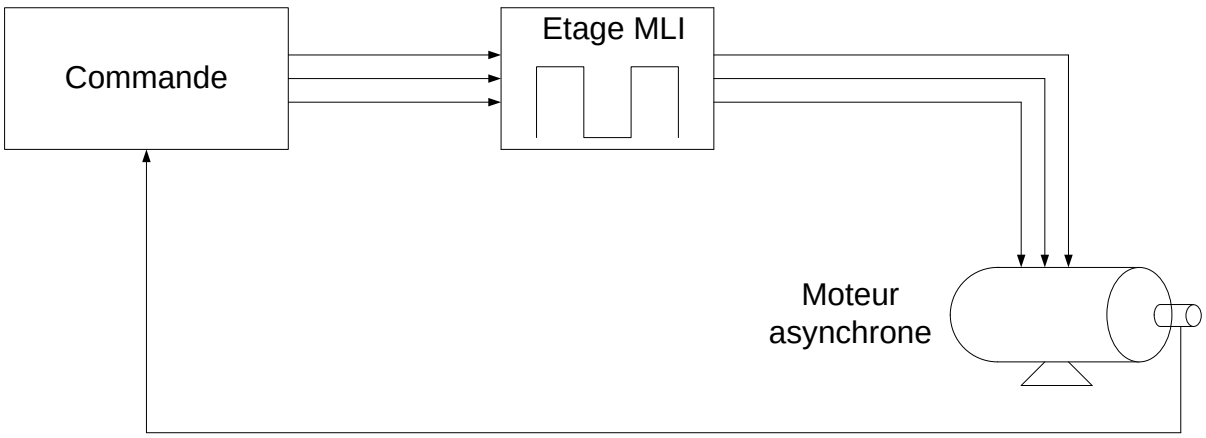


Figure 1.9. Schéma de position de l'étage MLI sur la chaine de régulation du moteur asynchrone.

5.1. Modulation à largeur d'impulsion MLI

La technique de modulation en largeur d'impulsion MLI (**M**odulation de **L**argeur d'**I**mpulsion ou PWM : **P**ulse **W**idth **M**odulation) est l'essor et le fruit du développement de l'électronique de puissance à la fin du siècle dernier. Elle est le cœur du contrôle des convertisseurs statiques. Le choix de la technique MLI pour contrôler l'onduleur de tension est d'avoir une réponse rapide et des performances élevées.

Le choix de la technique dépend du type de la machine à commander, du type des semi-conducteurs, de la puissance mise en jeu et la simplicité ou la complexité d'algorithmes à implanter. La MLI est composée d'impulsions dont la largeur dépend des choix effectués pour la stratégie de modulation [5].

Il existe plusieurs types de méthodes ou fonctions MLI. Une description non-exhaustive de l'ensemble de ces stratégies est résumée, ci-après, comme suit :

a. **MLI intersective (Modulation par porteuse) :**

C'est une stratégie MLI triphasée, simple à réaliser en analogique initialement conçue en monophasé et son implantation numérique est plus compliquée du fait qu'un grand nombre d'échantillons de la modulante doit être sauvegardé dans une mémoire ROM pour pouvoir obtenir une bonne précision du signal modulé. Son principe est facile avec une simple comparaison, pour chaque bras, entre un signal de référence (la modulante) et un signal triangulaire « dent de scie » de fréquence plus élevée (la porteuse). La fréquence de la porteuse définit la fréquence de découpage, et les points d'intersection entre la modulante et la porteuse correspondent aux instants de commutation au moment desquels l'onduleur change d'état. Parmi les variantes de la modulation MLI intersective, la plus populaire étant la modulation sinusoïdale « Modulation sinus-triangle SPWM (Sinusoïdal PWM) »

b. **MLI Vectorielle (Modulation poste calculer):**

La modulation vectorielle (Space Vector Modulation) est une technique numérique. Les ordres de commutations des interrupteurs sont déterminés par un algorithme et sont calculés analytiquement à travers des équations mathématiques avec un vecteur tension de contrôle qui est calculé globalement et approximé sur une période de modulation, par un vecteur tension moyen, puis les ordres de commandes adéquats sont appliqués aux interrupteurs.

Contrairement à d'autre méthodes, la MLI vectorielle ne s'appuie pas sur des calculs séparés des modulations pour chacun des bras de l'onduleur afin d'obtenir en valeur

moyenne une tension de référence à partir des états de commutation de l'onduleur et en fin les vecteurs à appliquer et les temps d'application de ces vecteurs.

c. **MLI pré-calculée (Modulation pré-calculée) :**

Le développement des technologies numériques permet le recours à des stratégies de modulation triphasées spécifiques, non déduites des techniques analogiques. Elle est appelée aussi la technique directe numérique (DDT- Direct Digital Technique) ou technique sans porteuse. Son principe est de générer des impulsions grâce à des séquences préalables calculées et stockées dans une mémoire [5].

Conclusion

Dans ce chapitre, on a présenté des rappels sur les véhicules électriques, les machines asynchrones, ainsi que les techniques de commande des machines asynchrones, en tenant compte des avantages et inconvénients qu'elle présentent la commande MLI pré-calculée (PWM),elle semble la plus adéquate pour une utilisation pratique adaptée.

Commande d'un Onduleur Triphasé piloté par MLI à Structure NPC Multiniveaux

Introduction

Pour les domaines de hautes tensions et fortes puissances, l'alimentation des machines à courant alternatif est souvent assurée par des groupements d'onduleurs à deux niveaux.
Pour remédier aux problèmes associés à ces groupements, nous proposons d'étudier dans ce mémoire un onduleur multiniveaux : onduleur triphasé à cinq niveaux à structure NPC (Neutral Point Clamping). Dans ce chapitre, nous étudierons la structure de l'onduleur à cinq niveaux à structure NPC. Pour ce faire, nous commencerons par élaborer son modèle de fonctionnement. Nous développerons ensuite un modèle de commande de ce convertisseur.

1. Modélisation de l'onduleur à cinq niveaux à structure NPC

1.1. Structure générale de l'onduleur à cinq niveaux

L'onduleur triphasé à cinq niveaux à structure NPC est une nouvelle structure de conversion utilisée pour alimenter, à tension et fréquence variables, des moteurs à courant alternatif de forte puissance. Plusieurs structures de l'onduleur à cinq niveaux sont possibles [6].
Dans le cadre de notre travail, nous présentons une structure de l'onduleur à cinq niveaux de type NPC comme l'indique la figure 2.1.Cette structure se compose de trois bras symétriques constitués chacun de six interrupteurs en série et deux autres en parallèles, plus deux diodes permettant l'obtention du zéro de la tension V_{km} notées DDK0 et DDK1, Chaque interrupteur est composé d'un interrupteur bicommandable (transistors MOSFET ou IGBT) et d'une diode montée en tête bêche.

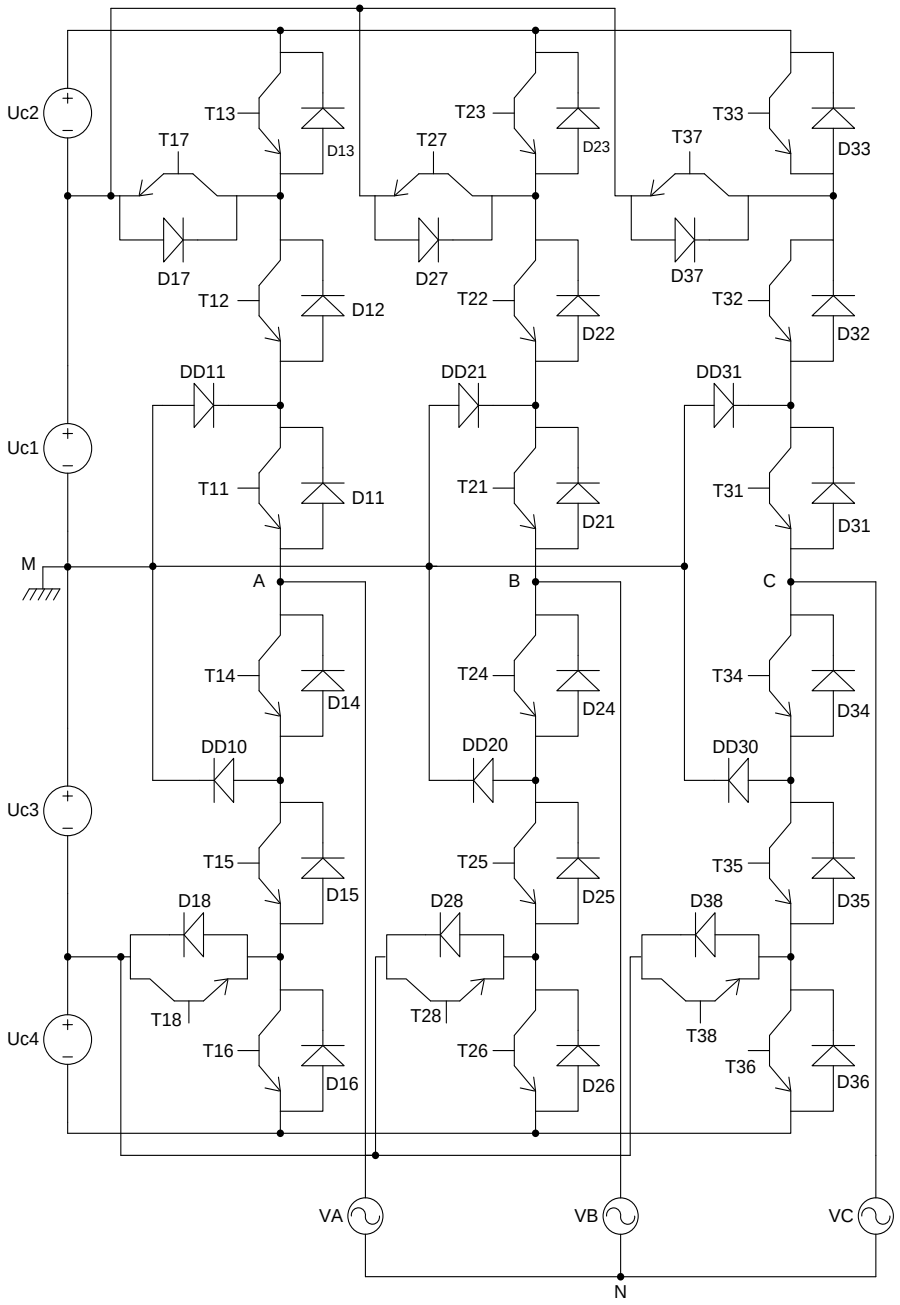


Figure 2.1. Schéma d'un onduleur triphasé à cinq niveaux à structure NPC.

1.2. Modélisation du fonctionnement de l'onduleur à cinq niveaux

Afin d'élaborer les différentes configurations de l'onduleur à cinq niveaux, sans apriori sur la commande, on considère les hypothèses suivantes:

- Chaque paire transistor–diode est représentée par un seul interrupteur bidirectionnel supposé idéal (Figure 2.2).
- Vue la symétrie de l'onduleur triphasé à cinq niveaux, la modélisation de ce dernier se fait par bras.

- Les tensions U_{c1}, U_{c2}, U_{c3}, U_{c4} sont des tensions continues supposées idéales (égales et constantes), $U_{c1} = U_{c2} = U_{c3} = U_{c4} = U_c$.

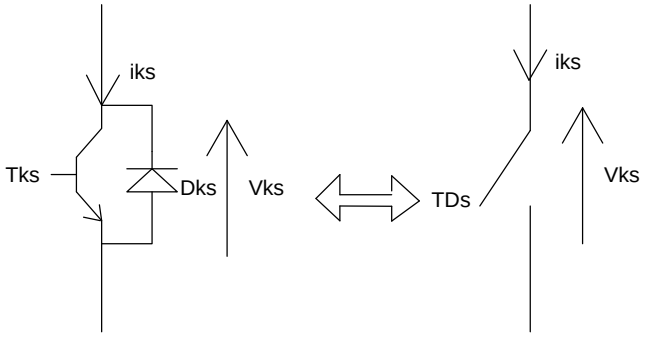


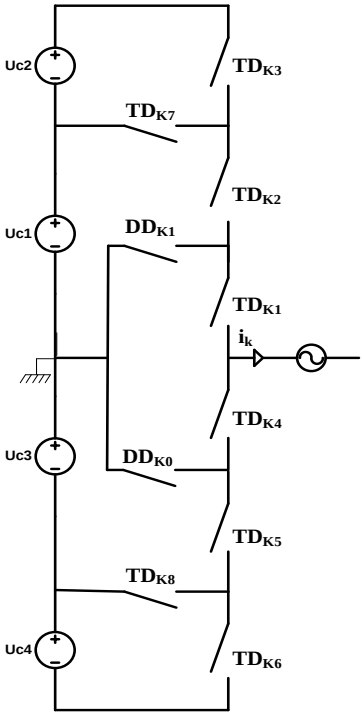
Figure 2.2. Interrupteur bidirectionnel équivalent au paire transistor–diode.

1.2.1. Différentes configurations d'un bras d'onduleur à cinq niveaux

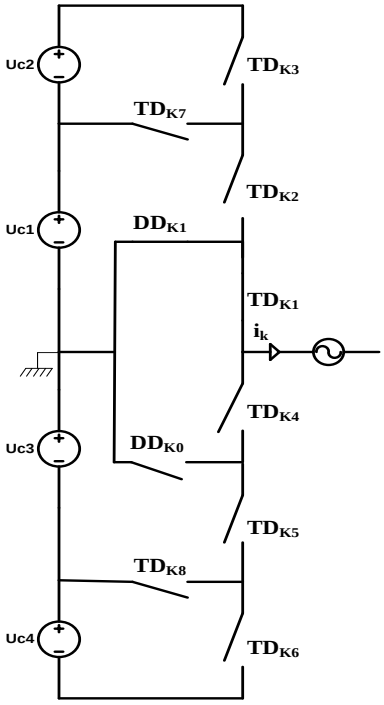
L'analyse topologique d'un bras de l'onduleur triphasé à cinq niveaux à structure NPC montre qu'il existe sept configurations possibles (figure 2.3). Les grandeurs électriques caractérisant chacune de ces configurations sont représentées dans le tableau 2.1 (avec M origine des potentiels et V le potentiel du nœud K du bras K).

Configuration	Grandeurs électriques
E_0	$I_k=0$
E_1	$V_k=0$
E_2	$V_k=U_{c1}= U_c$
E_3	$V_k= U_{c1}+ U_{c2}=2U_c$
E_4	$V_k= 0$
E_5	$V_k= -U_{c1}=-U_c$
E_6	$V_k= -U_{c1}-U_{c2}=-2U_c$

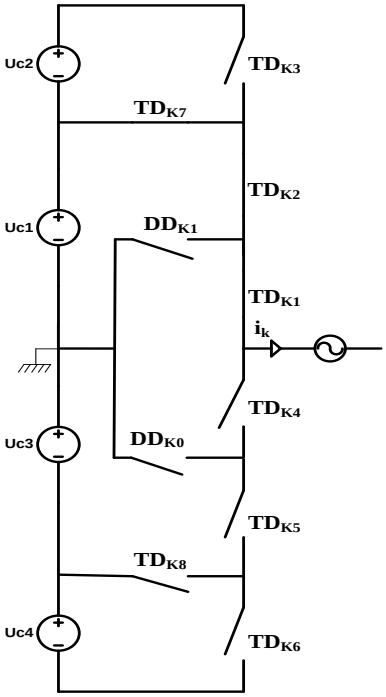
Tableau 2.1. Grandeurs électriques correspondantes à chacune des configurations d'un bras K d'onduleur à cinq niveaux à structure NPC.



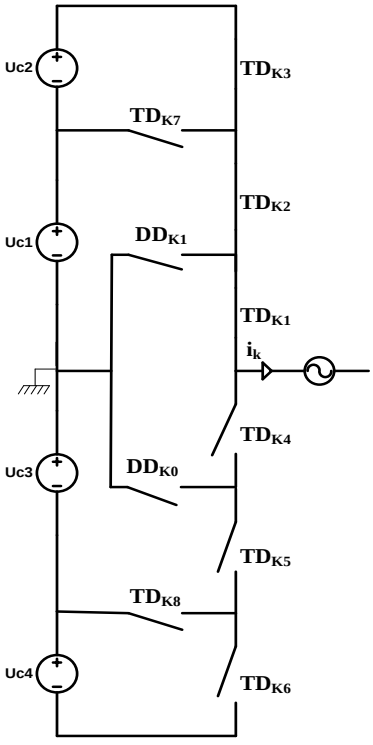
E_0



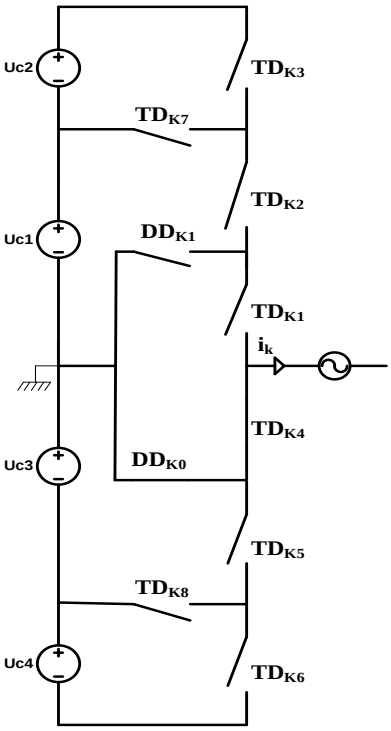
E_1



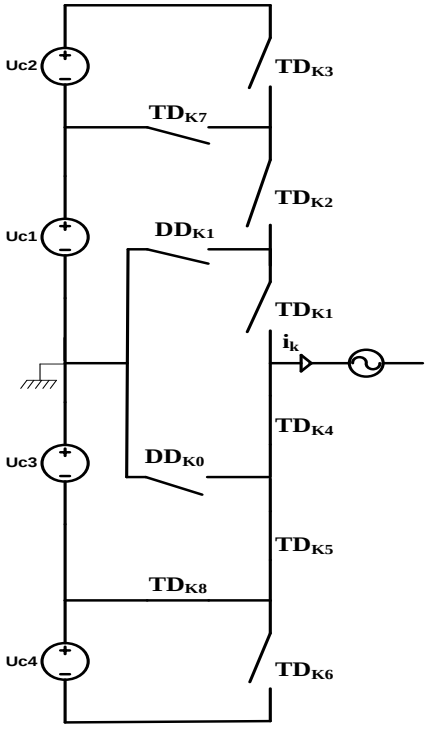
E_2



E_3



E_4



E_5

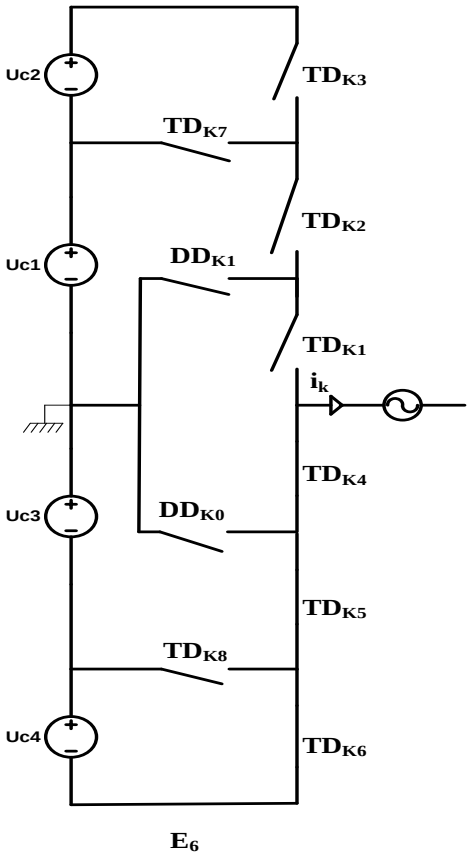


Figure 2.3. Différentes configurations du bras K de l'onduleur à cinq niveaux.

2. Commande de l'onduleur triphasé à cinq niveaux à structure NPC

Pour éviter la conduction simultanée des six interrupteurs d'un seul bras qui peut engendrer leur destruction par croissance du courant lors du court-circuit ou par une surtension dans le cas de l'ouverture de tous les interrupteurs, on définit une commande complémentaire des différents semi-conducteurs d'un bras, plusieurs commandes complémentaires sont possibles pour un onduleur à cinq niveaux, puisque l'onduleur triphasé à cinq niveaux est symétrique, donc l'étude se fera par bras et la commande la plus optimale est la suivante :

$$\begin{cases} B_{k1} = \overline{B}_{k5} \\ B_{k2} = \overline{B}_{k4} \\ B_{k3} = \overline{B}_{k6} \\ B_{k7} = B_{k1}B_{k2}\overline{B}_{k3} \\ B_{k8} = B_{k4}B_{k5}\overline{B}_{k6} \end{cases} \quad (2.1)$$

où B_{ks} désigne la commande de base du transistor T_{ks} et k indique le numéro du bras (1,2 ou 3), B_{ks} vaut 1 lorsque T_{ks} est fermée et elle vaut 0 dans le cas contraire.

$$B_{ks} = \begin{cases} 1 & si \quad T_{ks} \quad fermée \\ 0 & si \quad T_{ks} \quad ouvert \end{cases} \tag{2.2}$$

Les cinq niveaux de tension délivrés par l'onduleur ainsi que les états des interrupteurs correspondants sont indiqués au tableau (2.2).

V_{kM}	B_{k1}	B_{k2}	B_{k3}	B_{k4}	B_{k5}	B_{k6}	B_{k7}	B_{k8}
0	1	0	0	1	0	1	0	0
U_c	1	1	0	0	0	1	1	0
$2U_c$	1	1	1	0	0	0	0	0
$-U_c$	0	0	1	1	1	0	0	1
$-2U_c$	0	0	0	1	1	1	0	0

Tableau 2.2 Table d'excitation des interrupteurs d'un onduleur à cinq niveaux à structure NPC associée à la commande complémentaire.

D'après le tableau 2.2, on peut générer les signaux de commande des interrupteurs de l'onduleur mais il faut connaitre les instants de commutations t_1, t_2 ainsi que la période de commutation T.

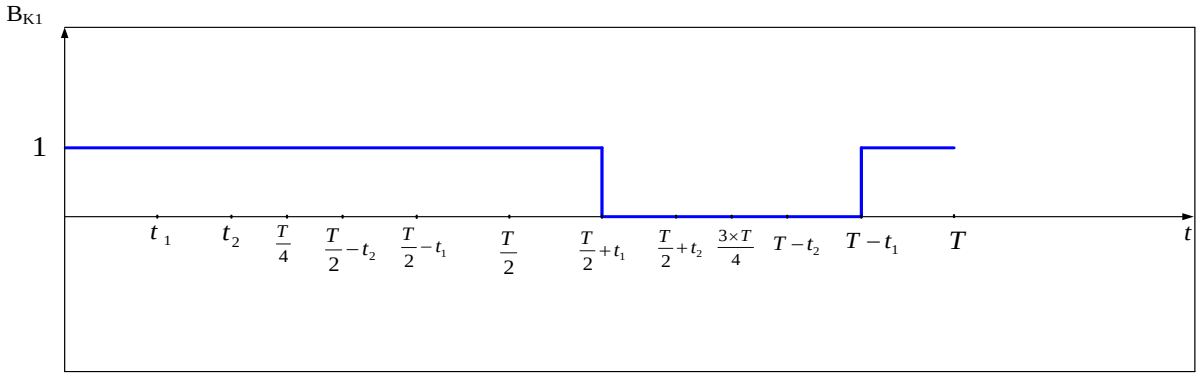


Figure 2.4. Signal de commande du transistor T11

Figure 2.5. Signal de commande du transistor T12

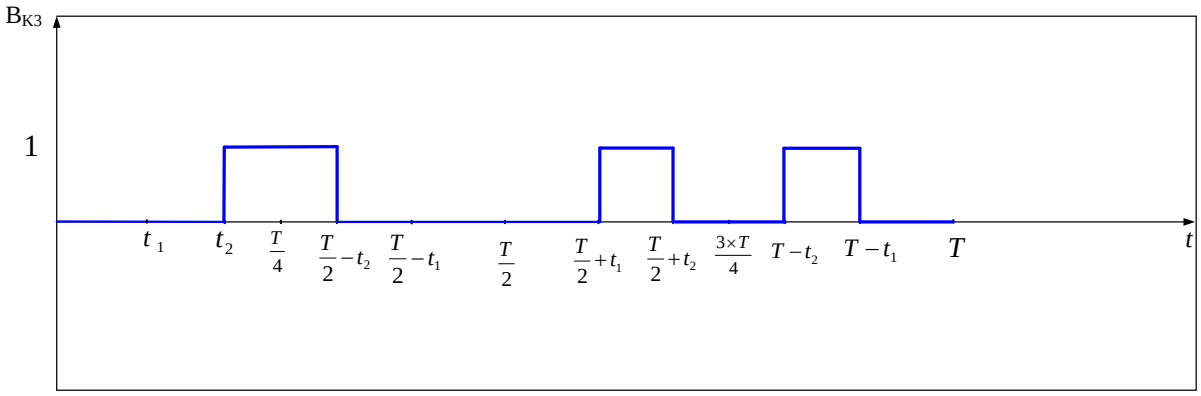


Figure 2.6. Signal de commande du transistor T13

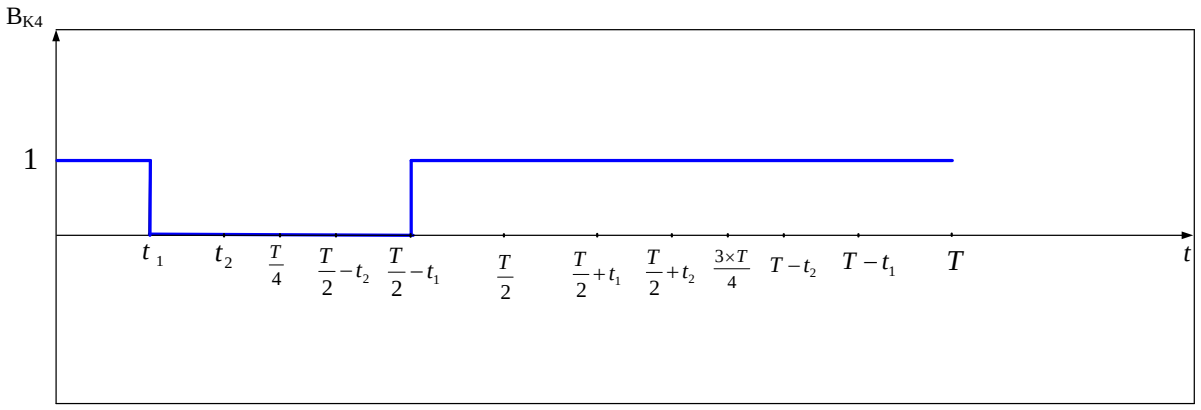


Figure 2.7. Signal de commande du transistor T14

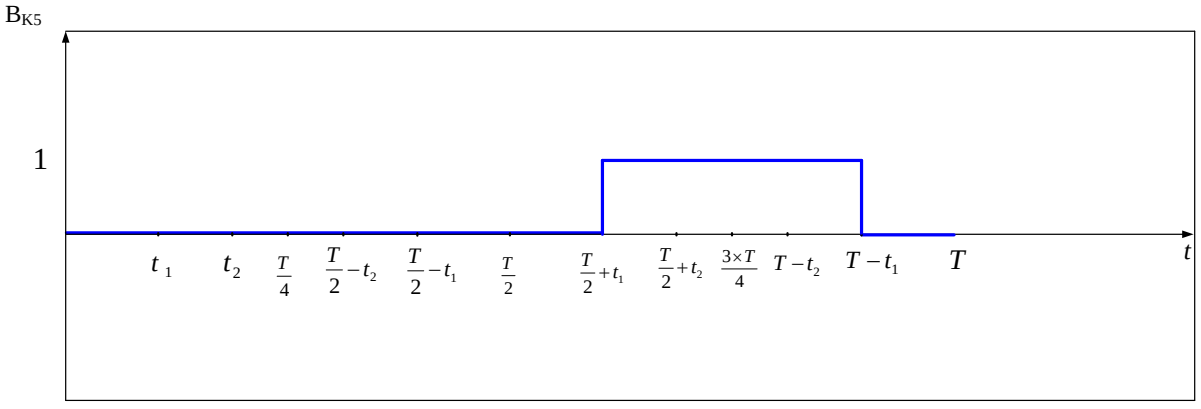


Figure 2.8. Signal de commande du transistor T15

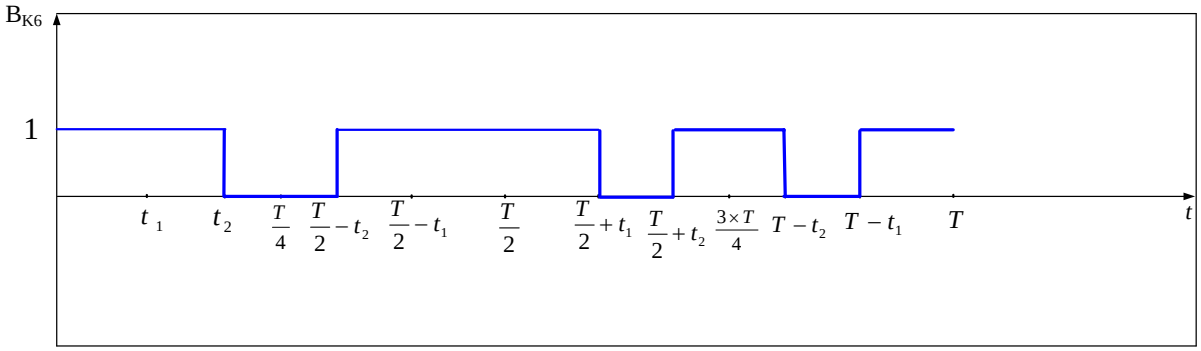


Figure 2.9. Signal de commande du transistor T16

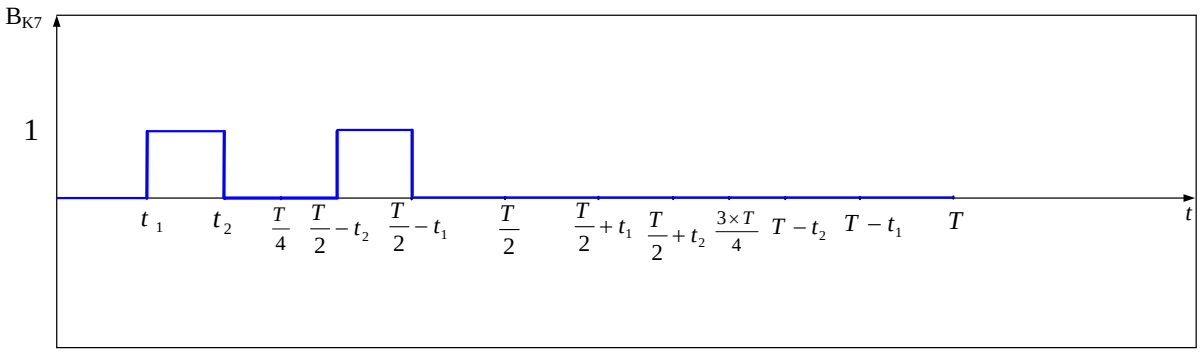


Figure 2.10. Signal de commande du transistor T17

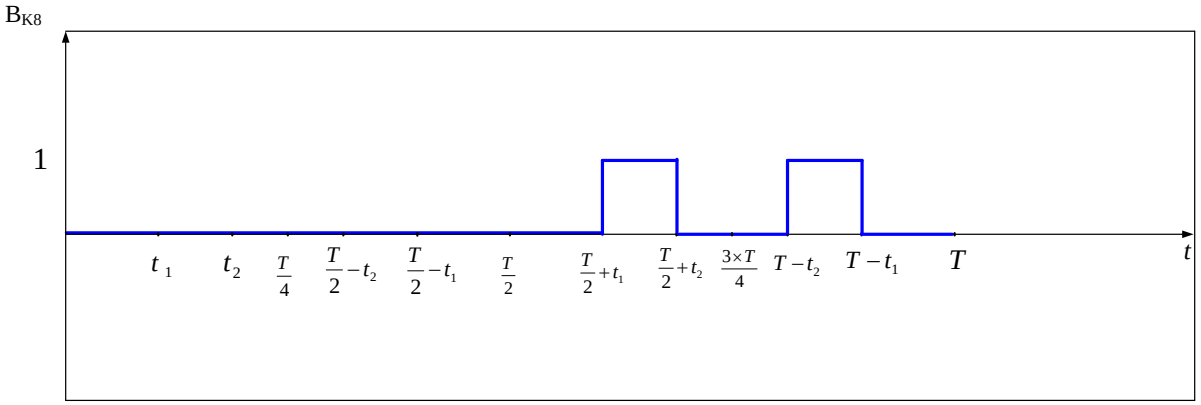


Figure 2.11. Signal de commande du transistor T18

La tension de sortie de l'onduleur triphasé à cinq niveaux est représentée dans la Figure 2.4. Les angles de commutation α_1, α_2 définissent les transitions entre les différents niveaux de tension. On suppose que l'amplitude de la tension de sortie est égale à l'unité (U_c=1).

2.1 La technique de PATEL et HOFT

En 1973, Patel et Hoft a généralisé la Procédé pour l'élimination sélective d'harmonique et avec asservissement du fondamental. Ils ont fourni la solution pour éliminer jusqu'à à cinq harmoniques. Alors que les autres harmoniques d'ordre supérieur peuvent être éliminés à l'aide de filtres [7]. Puisque la tension $f(\omega t) = f(\alpha)$ est périodique, donc elle est décomposable en série de Fourrier comme suit :

$$f(\alpha) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(n\alpha) + b_n \sin(n\alpha)) \tag{2.3}$$

Les coefficients a_n et b_n sont donnés par:

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

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \sin(n\alpha) d\alpha \qquad \text{n=1, 2, 3,4,…} \tag{2.4}$$

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

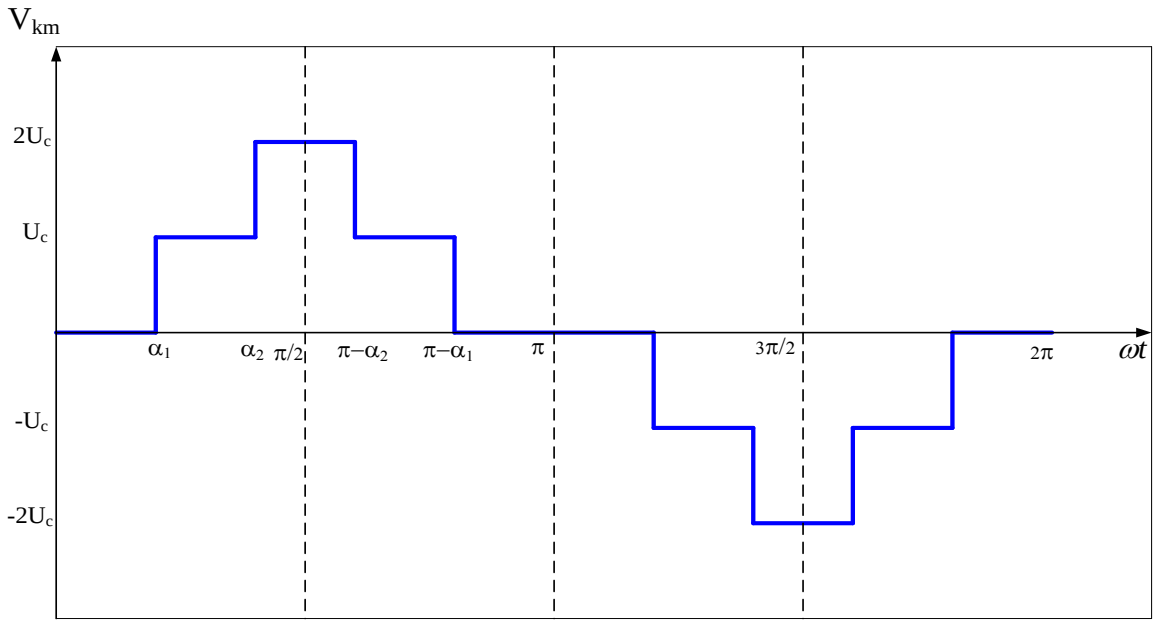


Figure 2.12. Motif de la tension de sortie du bras k d'un onduleur à cinq niveaux.

D'autre part comme $f(\omega t)$ (figure 2.12) présente une symétrie demi-onde et quart d'onde, la valeur moyenne a_0 et les coefficients a_n sont nuls et seulement les

harmoniques impaires des coefficients b_n existent, par conséquent, l'indice n prend les valeurs impaires 1, 3, 5, 7, 9,....

Puisque la charge est triphasée, dons les coefficients impairs multiples de trois sont nuls.

$$\begin{cases} b_n = 0 & si \quad n \quad pair \\ b_n = \dfrac{4}{\pi}\displaystyle\int_0^{\pi/2} f(\alpha)\sin(n\alpha)d\alpha & si \quad n \quad impair \end{cases} \tag{2.5}$$

Où n représente le rang de l'harmonique (1, 5, 7……)

L'expression (2.5) peut donc s'écrire de la manière suivante :

$$b_n = \frac{4}{\pi}\int_0^{\alpha_1} f(\alpha)\sin(n\alpha)d\alpha + \frac{4}{\pi}\int_{\alpha 1}^{\alpha_2} f(\alpha)\sin(n\alpha)d\alpha + \frac{4}{\pi}\int_{\alpha 2}^{\Pi/2} f(\alpha)\sin(n\alpha)d\alpha \tag{2.6}$$

Les paramètres α_1, α_2 représentent les angles de commutation à déterminer.

Sachant que $\cos(n\frac{\Pi}{2}) = 0$ pour n impair, l'équation (2.6) se réduit donc à :

$$b_n = \frac{4}{n\pi}\left[\cos(n\alpha_1) + \cos(n\alpha_2)\right] \tag{2.7}$$

En remplaçant n dans l'expression (2.7), on abouti à un système d'équations non linéaire suivant :

$$\begin{cases} \cos(\alpha_1) + \cos(\alpha_2) = \dfrac{\Pi r}{2} \\ \cos(5\alpha_1) + \cos(5\alpha_2) = 0 \end{cases} \tag{2.8}$$

Où r représente le taux de modulation.

Le système (2.8) contient 2 équations non-linéaires à 2 inconnues α_1, α_2 en fonction du taux de modulation r.

$$r = \frac{\text{Valeur crète du fondamental de la tension de charge}}{\text{Amplitude des crénaux de tension de sortie}} \tag{2.9}$$

2.2. Calcul des angles de commutation :

Le calcul des angles de commutation se fait à l'aide du Logiciel MATLAB, on exploite le principe de la théorie résultante et les polynômes symétriques qui consiste à transformer le système d'équations non-linéaire trigonométriques en un système polynomial algébrique, cette transformation est possible en effectuant les changements de variable appropriés, ensuite réduire le système de deux équations algébriques à une seule équation d'un seul inconnu mais de degré élevé en utilisant la méthode de substitution afin de réduire le degré de la résultante,

l'introduction du principe des polynômes symétriques sera utile en introduisant un deuxième changement de variable. Pour mieux illustrer l'approche mathématique (la théorie résultante et les polynômes symétriques) [8], on propose de résoudre le système d'équation (2.8) qui permettant d'éliminer l'harmoniques 5. Premièrement, on pose le changement de variable suivant :

$$\begin{aligned} x_1 &= \cos(\alpha_1) \\ x_2 &= \cos(\alpha_2) \end{aligned} \tag{2.10}$$

Les termes $\cos(5\alpha_1)$ et $\cos(5\alpha_2)$ est exprimé en fonction de $\cos(\alpha_1)$ et $\cos(\alpha_2)$ comme suit :

$$\cos(5\alpha_2) = 5\cos(\alpha_2) - 20\cos^3(\alpha_2) + 16\cos^5(\alpha_2) \tag{2.11}$$

$$\cos(5\alpha_1) = 5\cos(\alpha_1) - 20\cos^3(\alpha_1) + 16\cos^5(\alpha_1) \tag{2.12}$$

On remplaçant les équations (2.11) et (2.12) dans le système (2.8), on obtient :

$$\begin{cases} p_1(x_1, x_2) = x_1 + x_2 - \dfrac{\Pi r}{2} = 0 \\ p_5(x_1, x_2) = \sum_{i=1}^{2} (5x_i - 20x_i^3 + 16x_i^5) = 0 \end{cases} \quad \text{Avec } x_i = (x_1, x_2) \tag{2.13}$$

Les angles doivent vérifier la condition suivante :

$$0 \le \alpha_1 \le \alpha_2 \le \frac{\Pi}{2}, \qquad 0 \le x_2 \le x_1 \le 1 \tag{2.14}$$

La fonction evalin dans le logiciel MATLAB permet d'exploiter directement la théorie résultante des polynômes symétriques (2.15)

$$\begin{cases} q_1 = s_1 - x_1 - x_2 = 0 \quad avec \quad m = s_1 = \Pi * \dfrac{r}{2} \\ q_2 = s_2 - x_1 x_2 = 0 \end{cases} \tag{2.15}$$

L'application de la fonction MATLAB « evalin », qui permet d'effectué un changement de variable, ensuite on obtient l'équation (2.16).

$$p\ (m, s_2) = 16m^5 - 80m^3 s_2 - 20m^3 + 80m s_2^2 + 60m s_2 + 5m \tag{2.16}$$

Le polynôme (2.16) est de degré 2 par rapport à s_2, donc il possède deux solutions différentes, qui sont calculés par le logiciel MATLAB (équation 2.17).

$$\begin{cases} s_{21} = \dfrac{5^{1/2}m^2}{10} - \dfrac{5^{1/2}}{8} + \dfrac{m^2}{2} - \dfrac{3}{8} \\ s_{21} = \dfrac{5^{1/2}}{8} - \dfrac{5^{1/2}m^2}{10} + \dfrac{m^2}{2} - \dfrac{3}{8} \end{cases} \quad (2.17)$$

Le calcul de la résultante de q_1 et q_2 permet d'éliminer l'inconnu x_1 (équation 2.18)

$$res(q_1, q_2) = -x_2^2 + s_1 x_2 - s_2 \quad (2.18)$$

Par le logiciel MATLAB, on détermine les solutions (inconnu x_2) de l'équation 2.18, qui sont mentionné ci-dessous (équation 2.19)

$$\begin{cases} x_{21} = \dfrac{m}{2} + \dfrac{(m^2 - 4s_2)^{1/2}}{2} \\ x_{22} = \dfrac{m}{2} - \dfrac{(m^2 - 4s_2)^{1/2}}{2} \end{cases} \quad (2.19)$$

L'inconnu x_1 est donné par l'équation 2.20.

$$x_1 = s_1 - x_2 \quad (2.20)$$

Par ailleurs, les angles de commutations sont calculés comme suit (équation 2.21) :

$$\begin{cases} \alpha_1 = \cos^{-1}(x_1) \\ \alpha_2 = \cos^{-1}(x_2) \end{cases} \quad (2.21)$$

Le tableau 2.3 contient les angles de commutation en fonction du taux de modulation r.

r	α_1	α_2
0.3800	0.9374	1.5657
0.3900	0.9287	1.5570
0.4000	0.9199	1.5483
0.4100	0.9112	1.5395
0.4200	0.9024	1.5307
0.4300	0.8936	1.5219
0.4400	0.8847	1.5130
0.4500	0.8758	1.5042
0.4600	0.8669	1.4952

Tableau 2.3 Table des angles de commutation en fonction du taux de modulation.

On fait le changement de variable suivant : $\alpha_1 = a_1$ et $\alpha_2 = a_2$.

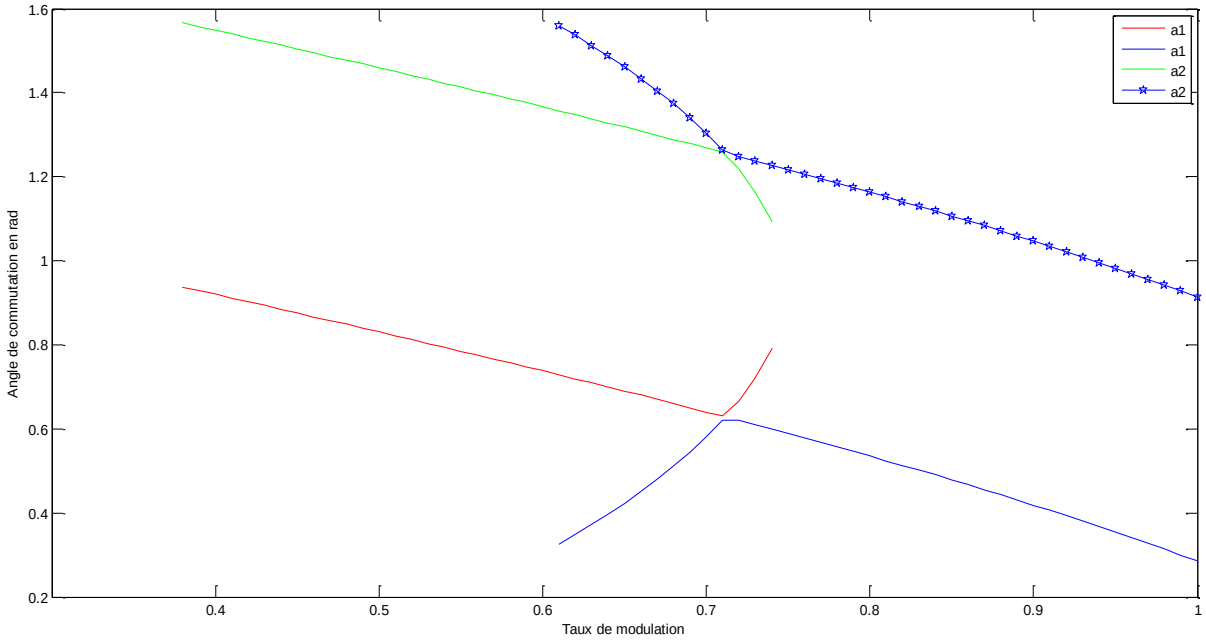


Figure 2.13. Courbe des angles de commutation en fonction du taux de modulation.

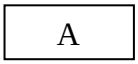
2.3. Interprétation des graphes

D'après la figure 2.13, le système d'équation 2.8 n'admet pas de solution lorsque $0 \le r \le 0.38$, il possède une seule solution dans l'intervalle $0.39 \le r \le 0.61$ ou $0.74 \le r \le 1$, elle possède deux solutions distinctes dans le cas $0.62 \le r \le 0.74$, pour chercher des solutions dans le premier intervalle, il faut utiliser d'autre méthodes comme la méthode itérative de Newton-Raphson.

3. Simulation de l'onduleur triphasé cinq niveaux

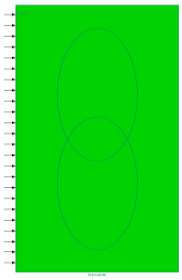
3.1. Schéma block :

La figure 2.14 représente un schéma block, qui est construit dans le Simulink MATLAB, les blocks fonctionnels utilisés sont détaillés ci-dessous :



From Workspace2 — Block qui permet de lire les signaux de commande depuis l'espace de travail MATLAB.

Transport Delay2 — Block qui assure le décalage temporelle entre les signaux de commande.



Block (SIMCOUPLER) qui représente l'interface entre

le Simulink MATLAB et le simulateur PSIM.

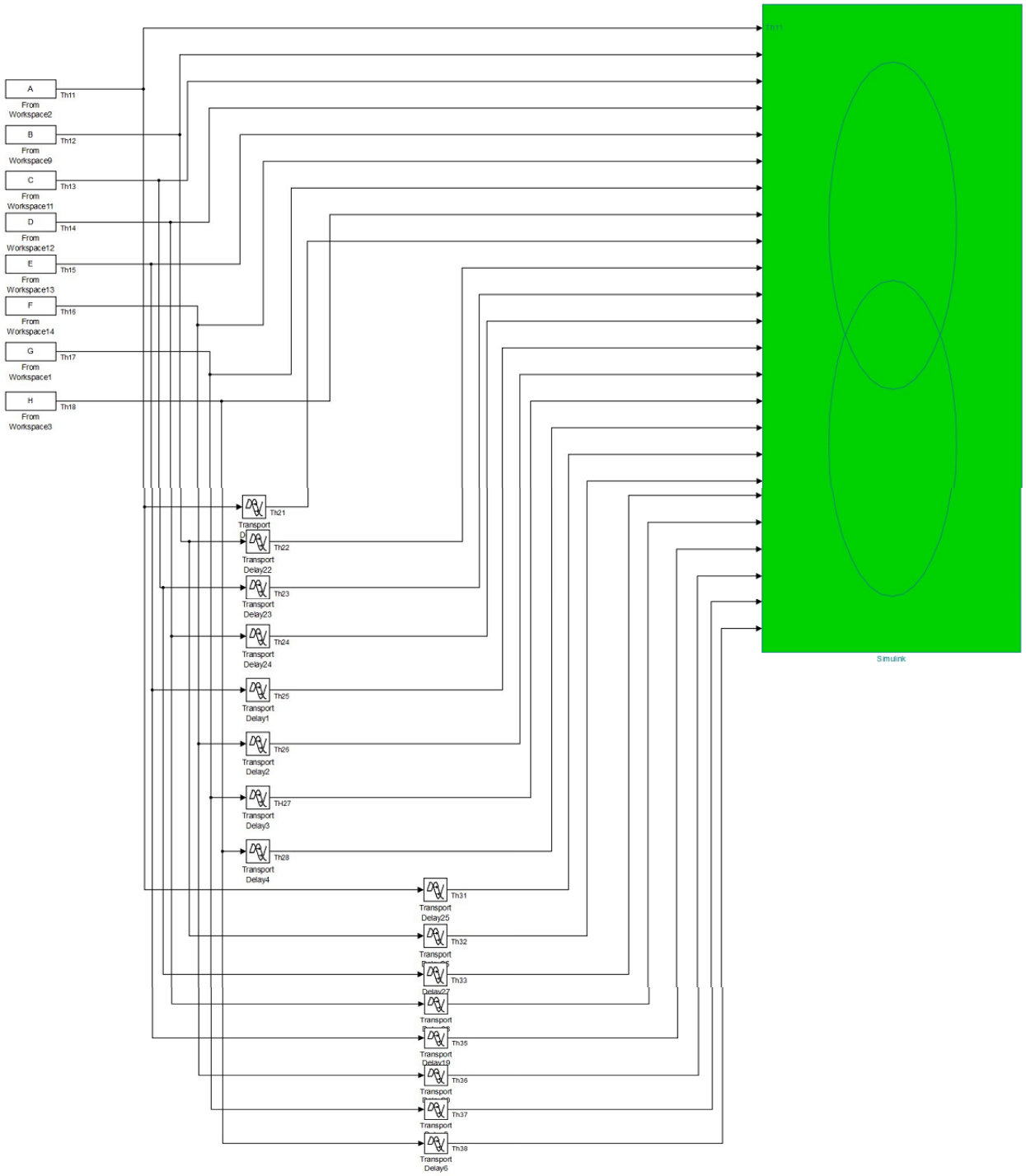


Figure 2.14. Schéma block de la commande MLI d'un onduleur triphasé.

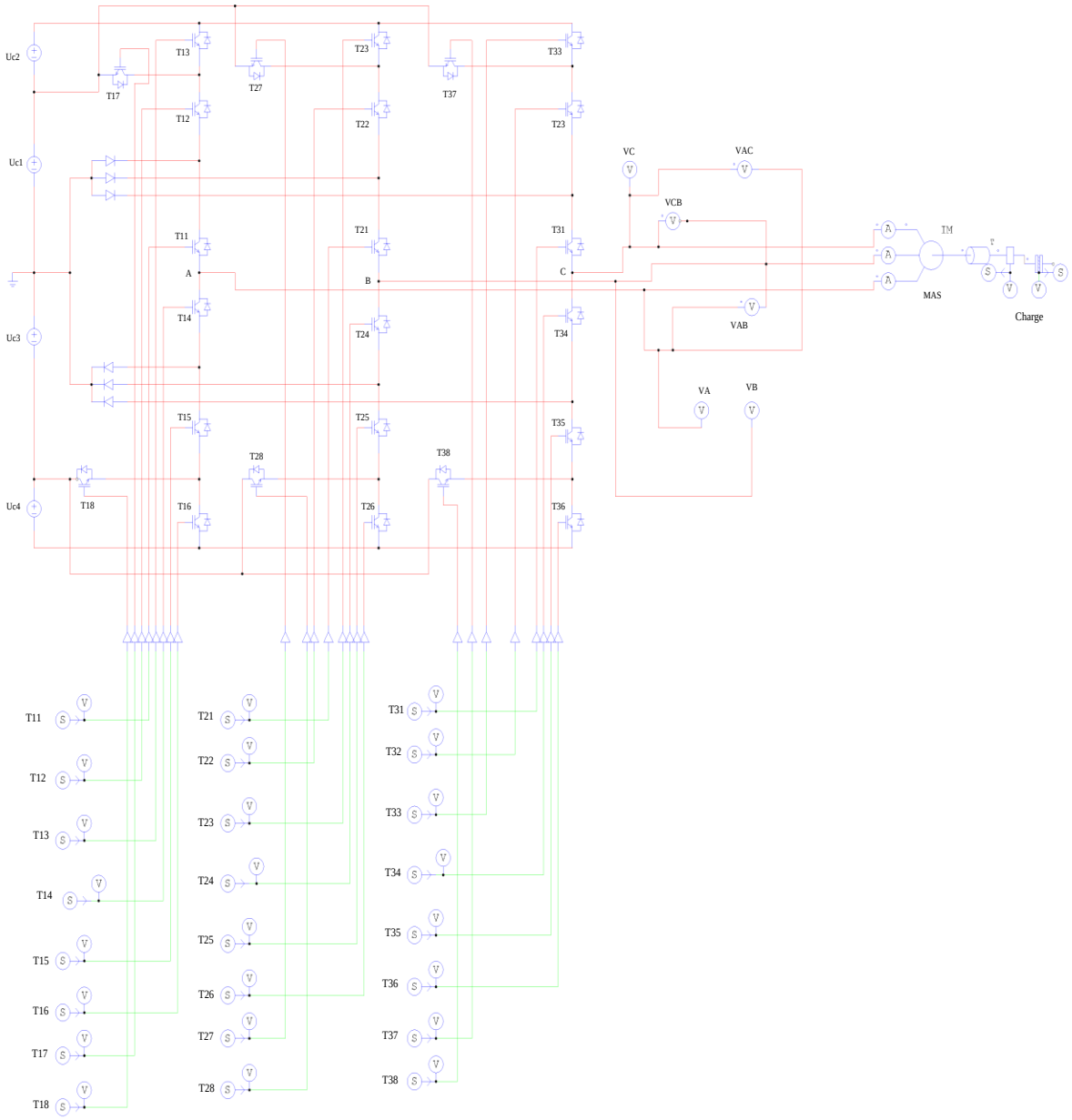


Figure 2.15. Schéma de la commande MLI de l'onduleur cinq niveaux dans le simulateur PISIM.

3.2. Distorsion d'harmonique total THD

Le taux de distorsion, encore appelé distorsion harmonique totale est défini comme le rapport de la valeur efficace globale des harmoniques (c'est-à-dire leur somme quadratique) à la valeur efficace de la composante fondamentale.

$$THD = \frac{\sqrt{H_2^2 + H_3^2 + \dots}}{H_1^2} \qquad (2.22)$$

3.3. Résultat de simulation :

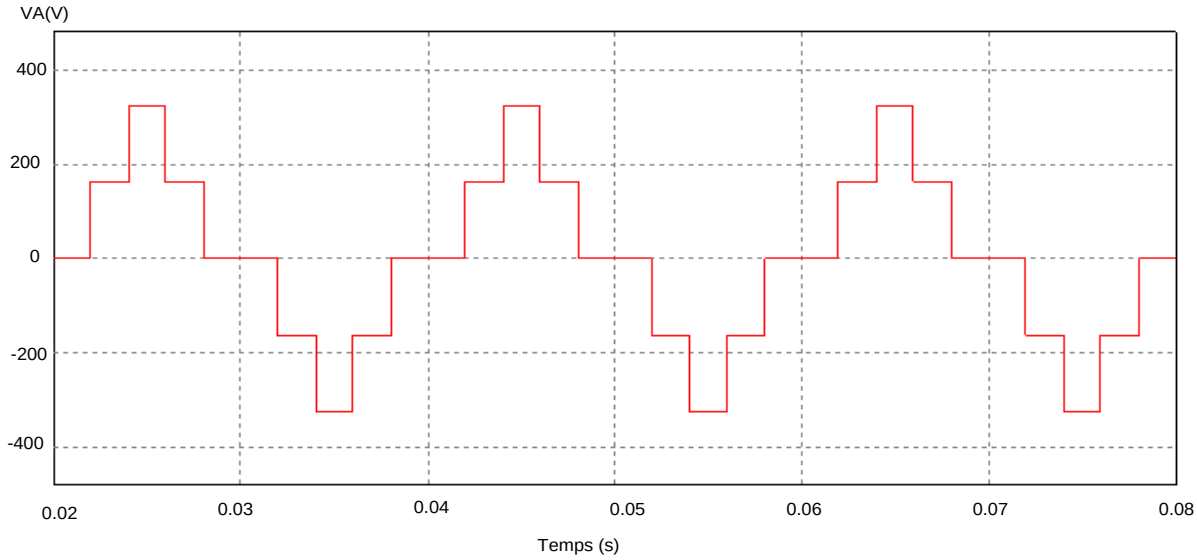


Figure 2.16. Allure de la tension simple de sortie de l'onduleur m=2, r=0.71, f=50Hz

$\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316°$.

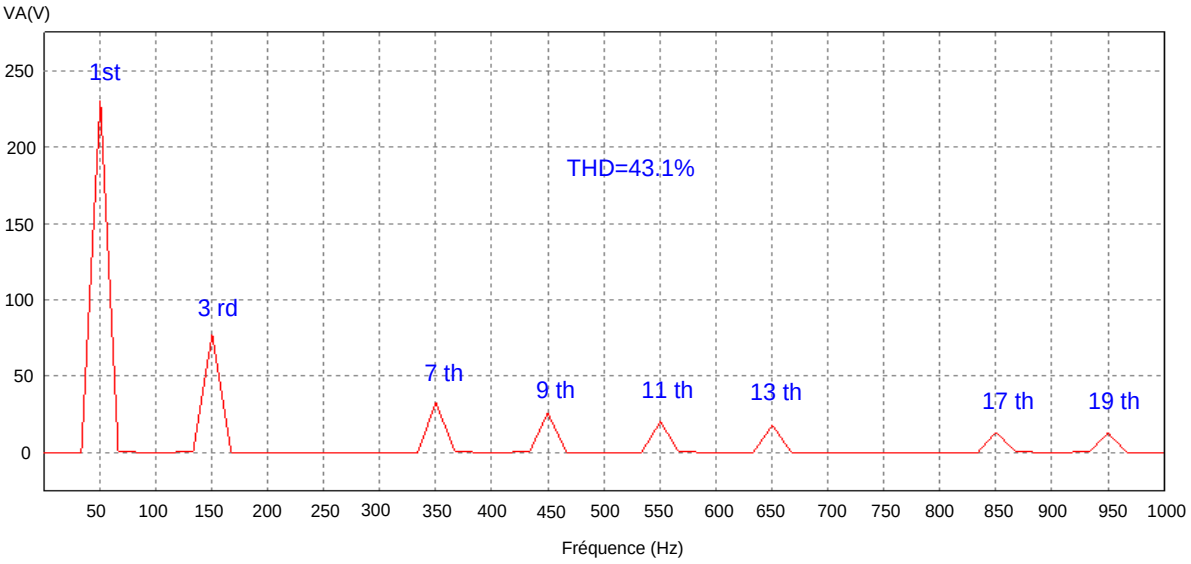


Figure 2.17. Spectre de la tension simple de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz,

$\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316°$.

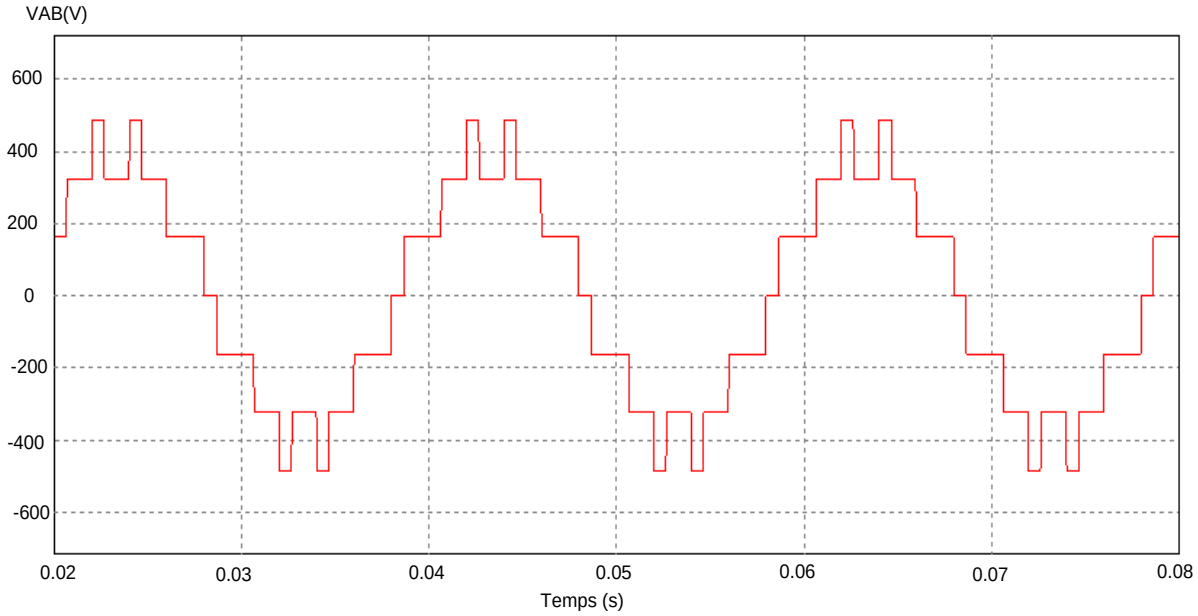


Figure 2.18. Allure de la tension composée de sortie de l'onduleur m=2, r=0.71, f=50Hz $\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316°$.

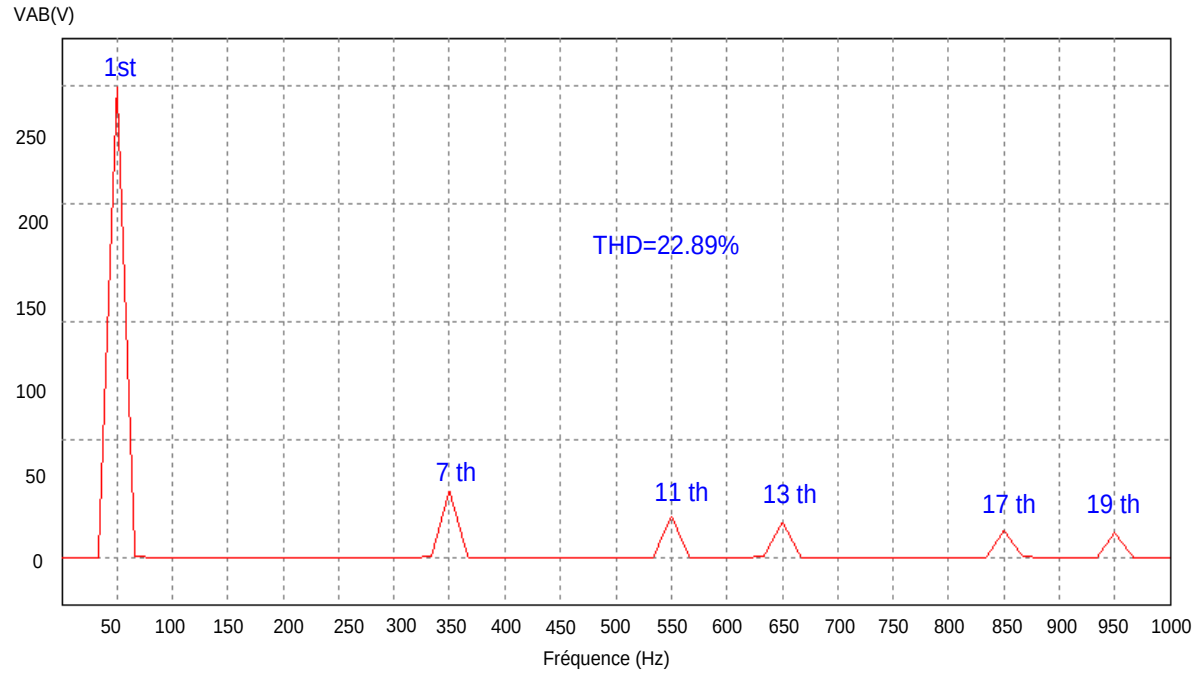


Figure 2.19. Spectre de la tension composée de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz $\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316°$.

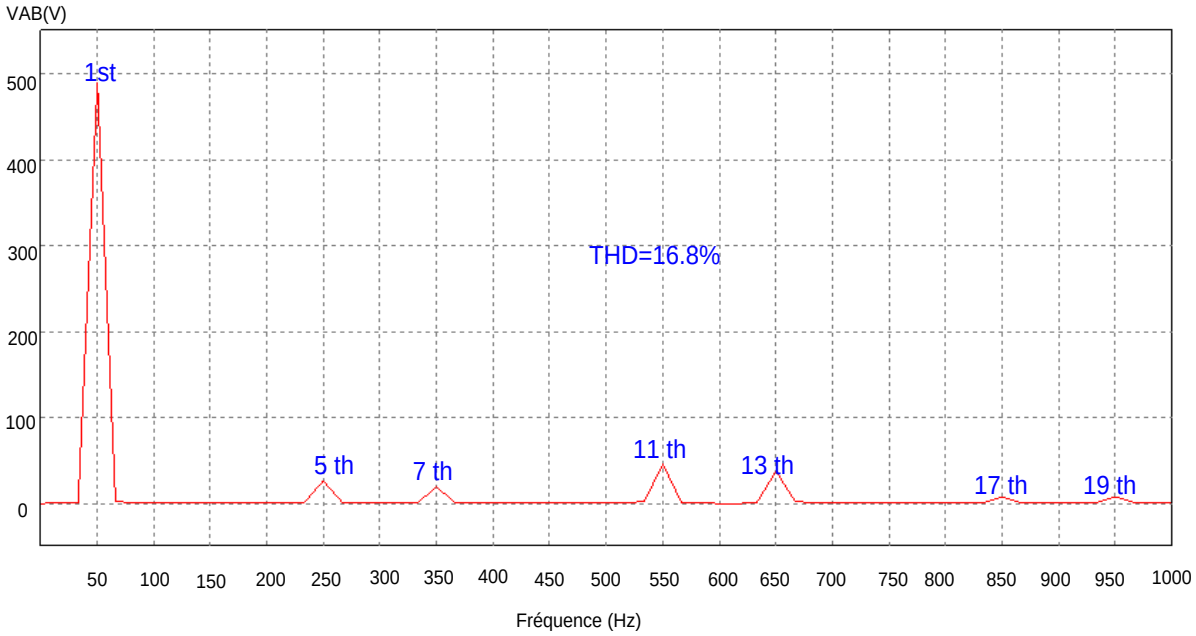


Figure 2.20. Spectre de la tension composée de l'onduleur cinq niveaux commandé par MLI calculée ($\alpha_1 = 30° / \alpha_2 = 60°$) pour m=2, f=50Hz.

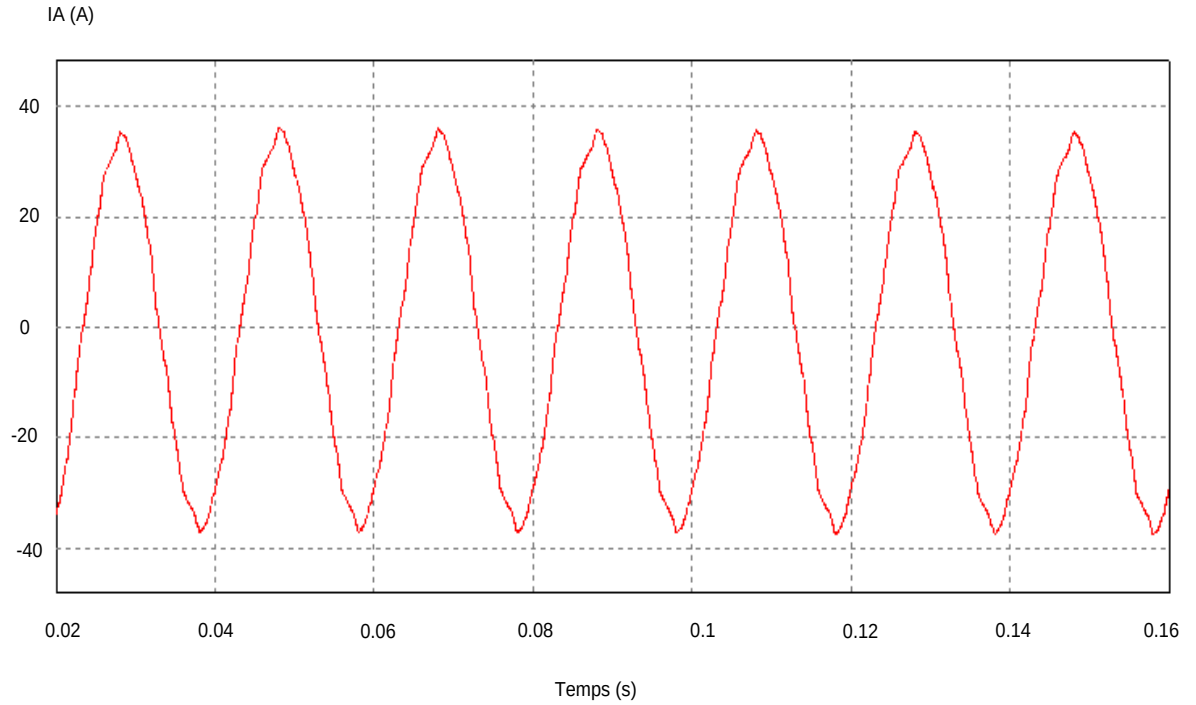


Figure 2.21. Allure du courant de sortie d'un bras de l'onduleur cinq niveaux dans la charge m=2, r=0.71, f=50Hz, $\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316°$.

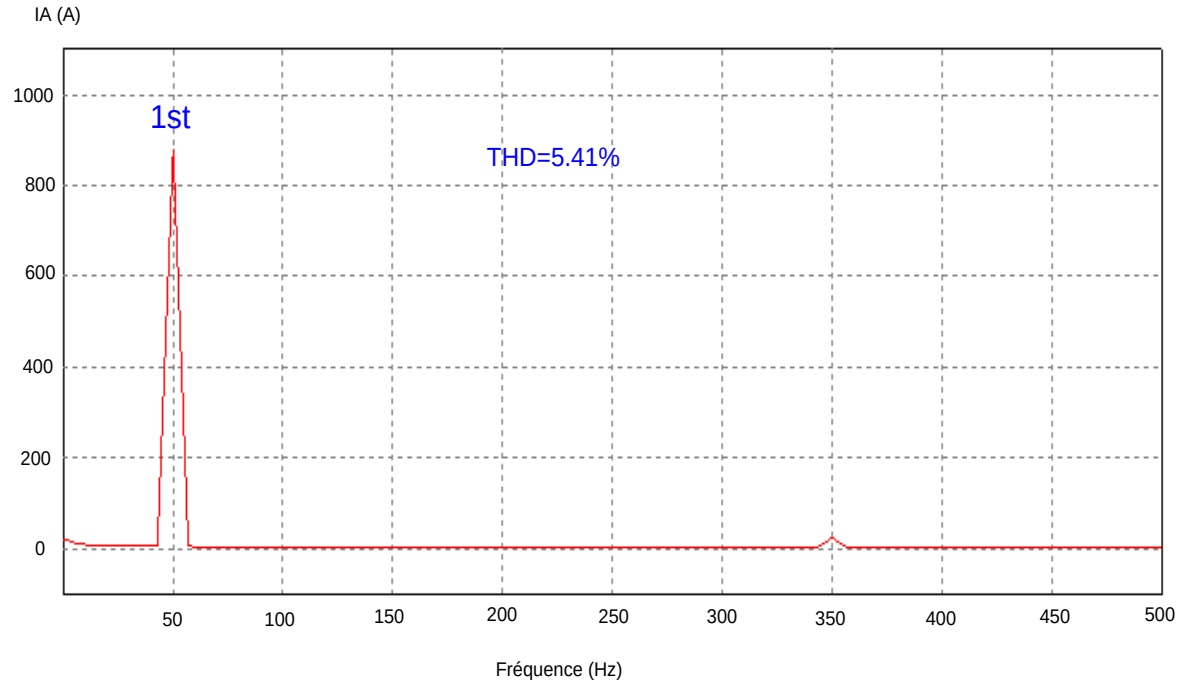


Figure 2.22. Spectre du courant de la charge de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz

$$\alpha_{1exacte} = 35.5683° et \alpha_{2exacte} = 72.4316° .$$

3.4. Interprétation des courbes :

Puisque le système (2.8) contient 2 inconnues α_1, α_2 en fonction du taux de modulation r, donc on peut éliminer juste l'harmonique n°5, cette dernière est située à la fréquence 250 Hz (r=0.71) et d'après la figure 2.19, cette harmonique est éliminée, par contre cette harmonique existe dans le cas d'une commande MLI ($\alpha_1 = 30° / \alpha_2 = 60°$), parce que ces deux angles ne sont pas des solution du système 2.8 ; on constate que l'amplitude des harmonique diminue lorsque la fréquence augmente, ce propriété est conforme à l'expression mathématique de l'amplitude (équation 2.7).

D'après la figure 2.19 et 2.20, les harmonique multiples de trois ne figure pas dans le spectre, ce qu'est confirme à la théorie des systèmes triphasé.

La forme du courant dans la charge est presque sinusoïdal (figure 2.21), le spectre du courant contient une seul harmonique qui est le fondamental (figure 2.22).

Conclusion

Dans ce chapitre on a étudié la commande MLI à structure NPC Multi-niveaux d'un onduleur de tension triphasé cinq niveaux, les angles de commutations qui annulent les harmoniques sont calculés à l'aide de la théorie résultante des polynômes symétriques, on a éliminé juste l'harmonique n°5 car le système 2.8 contient deux équations non-linéaire, pour éliminer les autres harmoniques, il faut calculer d'autres angles de commutation.

La commande MLI basée sur les réseaux de neurones artificiels

Introduction

Grâce aux résultats théoriques et pratiques obtenus au cours des dernières années, les réseaux de neurones sont devenus un outil de plus en plus utilisé dans divers domaines (industrie, banques, services, ingénierie, recherche et développement…). Ils demeurent encore un sujet d'un grand intérêt pour les chercheurs qui désirent améliorer les performances de ces réseaux et étendre leur champ d'applications. Cependant les réseaux de neurones appartiennent à l'intelligence artificielle, et comme tels, ils sont théoriquement capables de s'adapter et de permettre de réaliser diverses tâches.

Dans ce chapitre, nous présenterons les réseaux de neurones et ce dont ils sont capables. Nous étudierons surtout l'application des réseaux de neurones à la modélisation des systèmes non-linéaires. Nous essaierons aussi d'énumérer les contraintes et les limitations des réseaux de neurones dans un contexte d'utilisation en temps réel puis nous construirons l'algorithme MLI neuronal qu'estiment les angles de commutation en temps réels.

1. Choix d'une structure neuronale

Un problème qu'on doit impérativement résoudre avant d'utiliser un réseau de neurones est la définition de sa structure. Pour une topologie multicouche MLP, le nombre de neurones d'entrée/sortie du réseau est imposé par la structure de fonctionnement globale où il sera inséré, tandis que le nombre de couches cachées ainsi que le nombre de neurones correspondant à chaque couche, ne sont pas limités. L'objectif final étant une réalisation matérielle embarquable ; de ce fait, pour diminuer le temps de calcul, il est nécessaire de développer une architecture neuronale aussi petite que possible [9].

2. Le neurone biologique

Avant de commencer à énumérer les types de neurones, nous allons expliquer très brièvement les bases biologiques dont ils sont originaires.

Le neurone est le processeur élémentaire du cerveau, chaque neurone traite l'information qui lui parvient localement, puis transmet aux autres neurones qui lui sont connectés l'information qu'il a traitée ; ces cellules peuvent apprendre en changeant l'intensité de leurs connexions avec d'autres cellules ou détruire ou même créer de nouvelles connexions, elles peuvent aussi changer leurs règles de traitement de l'information. Ce processus de changement est appelé apprentissage et joue un rôle fondamental dans le comportement du neurone [10].

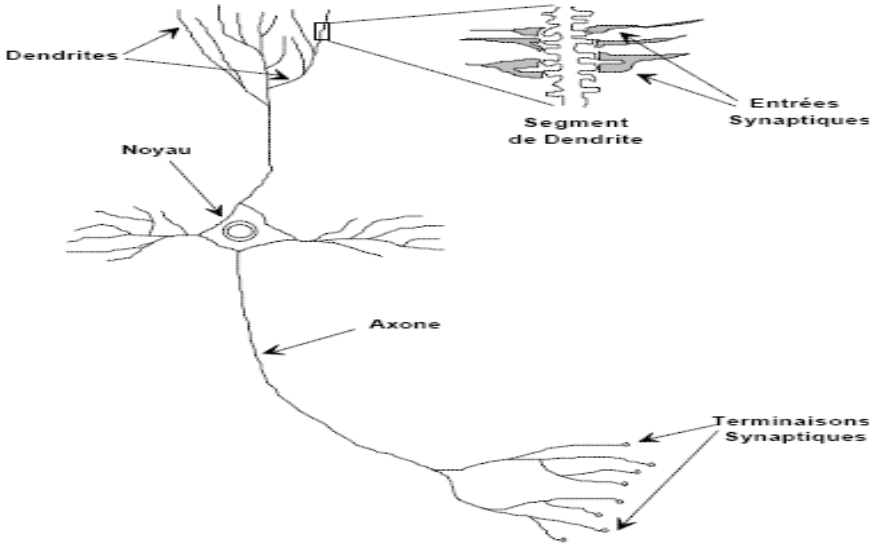


Figure 3.1. Représentation d'un neurone biologique [10]

D'après la figure 3.1, nous pouvons constater que le neurone est composé des parties suivantes :

a. **Corps cellulaire** : noyau du neurone.
b. **Dendrites :** Récepteurs principaux du neurone.
c. **Axone :** Fibre nerveuse de transport pour les signaux émis par le neurone.
d. **Synapse :** Connexion entre la dendrite et l'axone de deux neurones qui communique l'information, en la pondérant par un poids synaptique, à un autre neurone ; elle est essentielle dans le fonctionnement du système nerveux, chaque neurone réalise une opération très simple, qui est en fait une somme pondérée de ses entrées ; le résultat est comparé à un seuil et le neurone devient excité si ce seuil est dépassé. L'information contenue dans le cerveau est représentée par les poids donnés aux entrées de chaque neurone ; du fait du grand nombre de neurones et de leurs interconnections, ce système possède une propriété de tolérance aux fautes.

3. Modélisation pratique d'un neurone artificiel

Nous pouvons définir un neurone par les trois éléments suivants:

a. La fonction d'entrée totale qui définit le prétraitement effectué sur les entrées.
b. La fonction d'activation (ou d'état) du neurone qui définit son état interne en fonction de son entrée totale.

c. La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation.

La figure 3.1 montre la représentation graphique du modèle général d'un neurone.

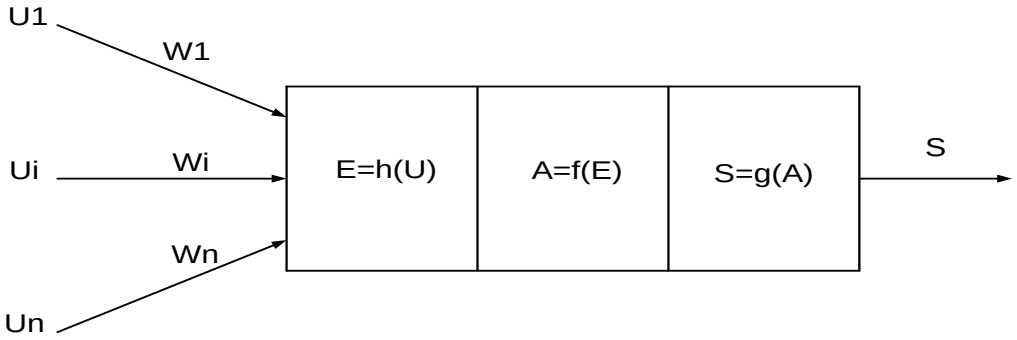


Figure 3.2. Model général d'un neurone

Nous adopterons les notations suivantes:

- U_i: Entrées du neurone (i=1..n).
- Wi : Coefficient de pondération de l'entrée i.
- h(.) : Fonction d'entrée totale.
- f(.) : Fonction d'activation.
- g(.) : Fonction de sortie.
- $E=h(u_1, ..,u_n)$: Entrée totale ou degré d'activation.
- A=f(E) : Etat du neurone.
- S=g(A) : Sortie.

Les entrées et les sorties peuvent être des grandeurs réelles ou binaires. La nature des différentes entrées, sorties et fonctions est explicitées ci-dessous.

La fonction d'entrée totale h(U) peut être une combinaison booléenne ou linéaire des entrées

$$h(u_1,..,u_n)=\sum_{1}^{n} w_i u_i - b \qquad ou \qquad h(u_1,..,u_n)=w^T u - b \tag{3.1}$$

Où b est le seuil.

La fonction d'activation f(E) prend généralement les formes suivantes:

• Une fonction binaire à seuil.

• Une fonction linéaire par morceaux.

• Une fonction sigmoïde.

$$A = f(E) = f(w^T U - b) \tag{3.2}$$

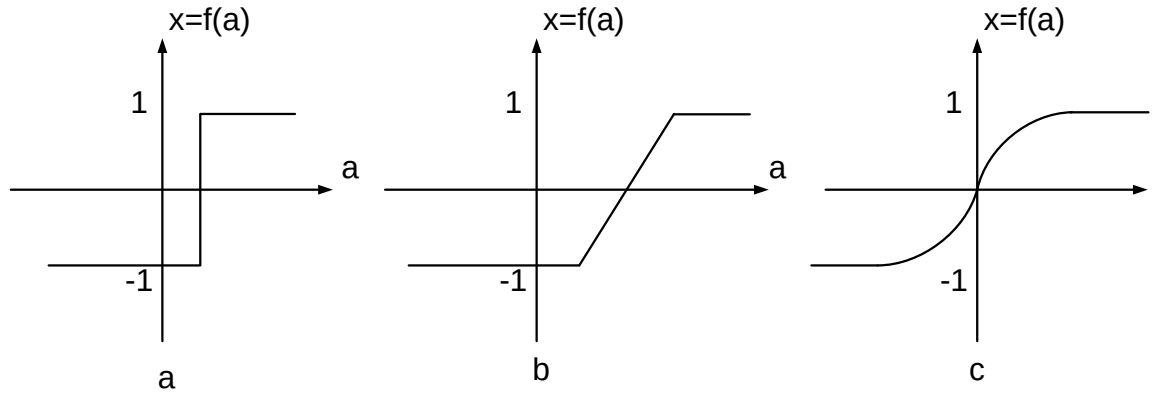


Figure 3.3. Différents types de la fonction d'activation pour le neurone artificiel

a : fonction à seuil, b : linéaire par morceaux, c : sigmoïde.

Nous constatons que les équations décrivant le comportement des neurones artificiels n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons affaire à des modèles à temps discret, synchrone, dont le comportement des composants ne varie pas dans le temps [11].

La fonction de sortie est souvent considérée comme la fonction identité :

$$S = g(A) = A = f(w^T U - b) \tag{3.3}$$

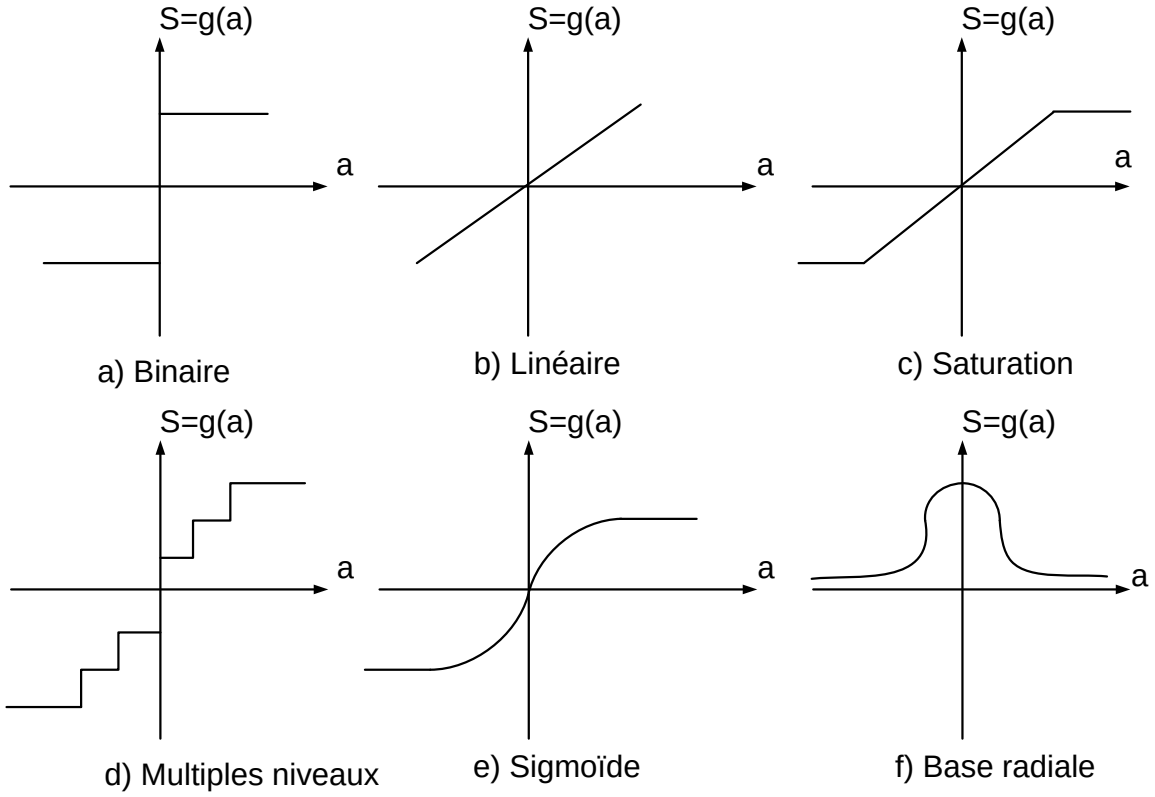


Figure 3.4. Fonction de sortie des neurones.

4. Différentes topologies neuronales

4.1. Réseau multicouche (MLP)

Les neurones sont arrangés par couches. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales (figure 3.5). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et avec celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées [11].

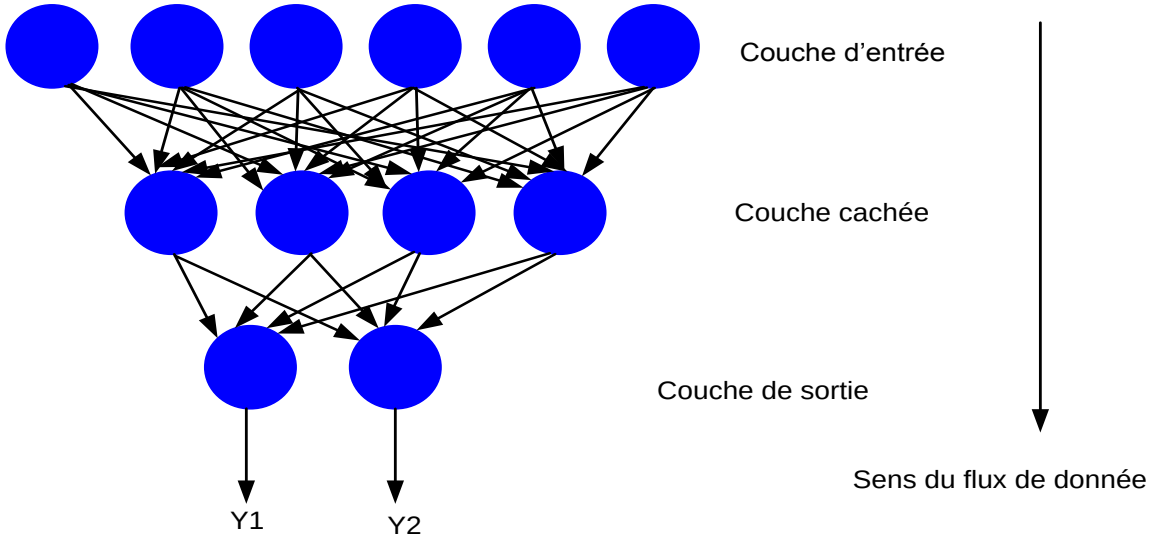


Figure 3.5. Topologie d'un réseau multicouche (MLP)

4.2. Réseau à connexions locales

Il s'agit d'une structure multicouche mais qui, à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (Figure 3.6). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

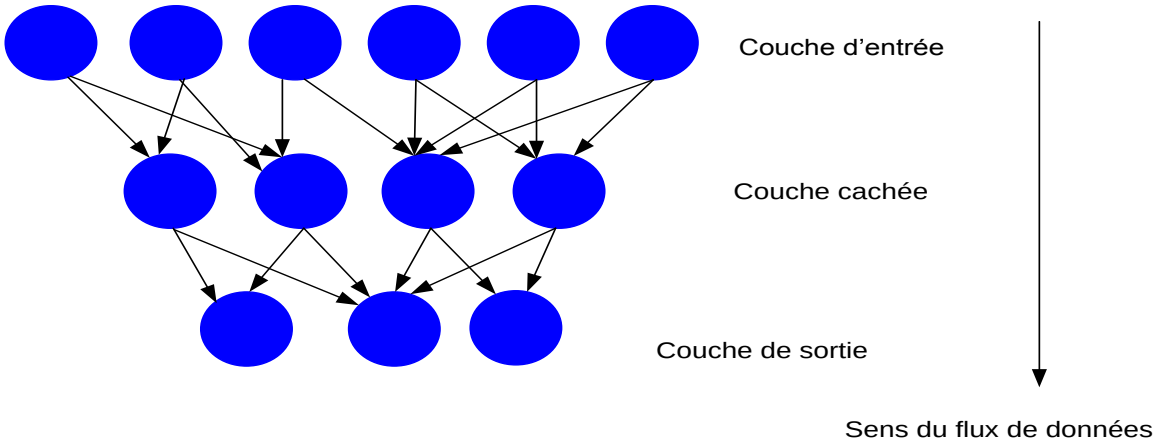


Figure 3.6. Réseau à connexion local.

4.3. Réseau à connexions récurrentes

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouches. Ces connexions sont le plus souvent locales (Figure 3.7).

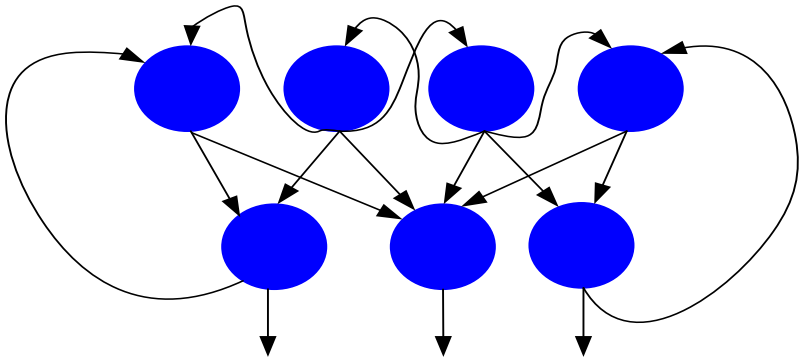


Figure 3.7. Réseau à connexions récurrentes.

4.4. Réseau à connexion complète

C'est la structure d'interconnexion la plus générale (Figure 3.8). Chaque neurone est connecté à tous les neurones du réseau (et à lui-même).

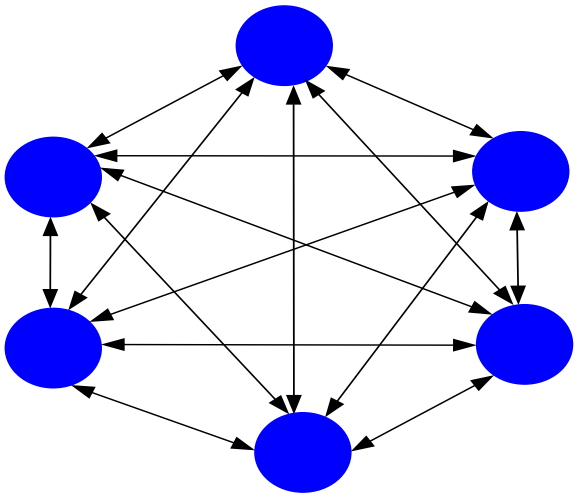


Figure 3.8. Réseau à connexion complète.

5. Fonctionnement d'un réseau de neurone

5.1. Le Perceptron (le neurone)

Avant d'aborder le comportement collectif d'un ensemble de neurones, nous allons présenter le Perceptron (un seul neurone) en phase d'utilisation. L'apprentissage ayant été réalisé, les poids sont fixes. Le neurone de la Figure 3.9 réalise une simple somme pondérée de ses entrées, compare une valeur de seuil, et fournit une réponse binaire en sortie. Par exemple, on peut interpréter sa décision comme classe 1 si la valeur de x est +1 et classe 2 si la valeur de x est -1. Les connexions des deux entrées e1 et e2 au neurone sont pondérées par les poids w1 et w2. La valeur de sortie du neurone est notée x. Elle est obtenue après une somme pondérée des entrées (a) et comparaison à une valeur de seuil S [10].

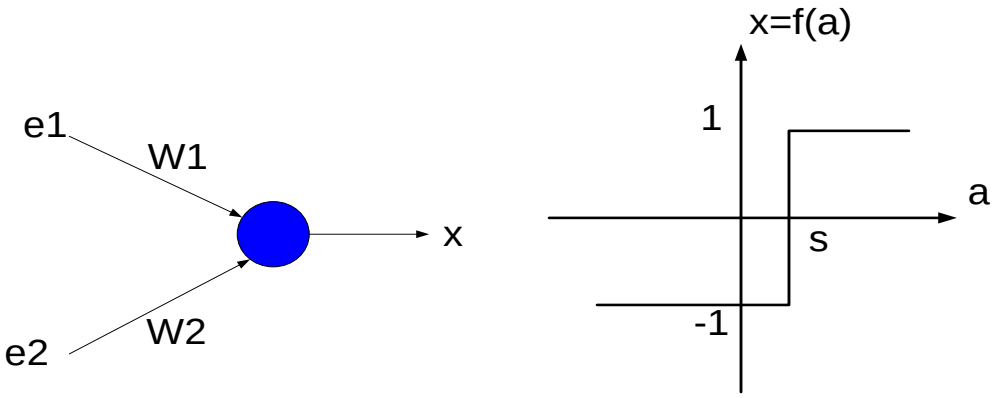


Figure 3.9. Le Perceptron : structure et comportement

5.2. Apprentissage

L'apprentissage est la propriété la plus intéressante des réseaux neuronaux, elle ne concerne cependant pas tous les modèles, mais les plus utilisés. L'apprentissage neuronal est

une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. Elle fait appel à des exemples de comportement. Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau afin d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider à priori des valeurs des poids des connexions d'un réseau pour une application donnée. Certains modèles de réseaux sont improprement dénommés à apprentissage permanent ; dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, cependant on peut toujours distinguer une phase d'apprentissage (en fait de remise à jour du comportement) et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage [10].

Des procédures d'apprentissage sont utilisées pour déterminer les paramètres du réseau pour qu'il se comporte correctement face à des entrées données. Selon le degré de supervision utilisé à l'apprentissage, les réseaux de neurones se classent en trois catégories :

a. **Apprentissage non supervisé**: l'ajustement des paramètres repose sur des critères internes au réseau, aucune information de référence n'est disponible, on ne dispose que des valeurs d'entrée.
b. **Apprentissage semi-supervisé ou apprentissage par renforcement**: l'utilisateur ne possède que des indications imprécises sur le comportement final désiré (correct/incorrect).
c. **Apprentissage supervisé** : les paramètres du réseau sont ajustés via un comportement de référence précis, les exemples sont des couples (Entrée, Sortie associée). Le réseau de neurones est entraîné de façon à reproduire le plus fidèlement possible les réponses désirées à un certain nombre de vecteurs d'entrée d'apprentissage.

L'apprentissage non supervisé est généralement utilisé pour la classification automatique des entrées, où le réseau apprend les caractéristiques des données d'entrée.

L'apprentissage semi-supervisé est généralement utilisé dans les systèmes de navigation autonome où un robot doit choisir un chemin selon un critère de type «bon/mauvais

chemin ». Cependant ce sont les réseaux à apprentissage supervisé qui sont les plus utilisés. Les domaines d'application de ces réseaux peuvent être classés essentiellement en deux catégories :

- **La reconnaissance et la généralisation :** dans les problèmes de reconnaissance, le réseau va apprendre à reproduire les sorties d'un ensemble de vecteurs d'apprentissage dont les entrées sont bruitées. Le réseau est censé générer les sorties adéquates bien que les entrées soient entachées d'erreur. Les applications de ces réseaux sont la reconnaissance de formes, de l'écriture, l'analyse des signaux, etc… Le réseau va reproduire une des sorties qu'il a déjà vues lors de l'apprentissage. Pour mémoriser toutes les entrées et sorties, le réseau va construire une hyper surface dans l'espace des vecteurs d'apprentissage et passant par tous les points d'apprentissage.
- **Les tâches d'approximation et d'interpolation :** le réseau va apprendre à reproduire le plus précisément possible les sorties d'un ensemble de vecteurs d'apprentissage. Il doit généraliser l'information acquise à des vecteurs qui n'ont pas été présentés lors de l'apprentissage. Le fait de faire apprendre un réseau à reproduire ou mémoriser exactement les informations données par les vecteurs d'apprentissage n'est pas désiré, parce que cela fait perdre sa performance de généralisation. Cette perte de généralisation est due au phénomène de sur-apprentissage. Les applications typiques de ces réseaux apparaissent surtout dans la robotique [11].

Il s'ensuit qu'un réseau de neurones se distingue par trois caractéristiques principales :

- L'architecture du réseau qui caractérise un chemin particulier par lequel les éléments du réseau sont connectés et qui définit la direction de propagation des informations.
- La fonction d'activation des neurones qui dicte leur comportement.
- L'algorithme d'apprentissage du réseau.

5.2.1. Caractéristique de l'algorithme d'apprentissage supervisé

Un réseau de neurones est conçu pour réaliser une tâche que le concepteur définit par un ensemble d'apprentissage **D** (base de données). Chaque élément de cet ensemble est appelé exemple d'apprentissage et se présente sous la forme d'un couple ***(x, y*)*** où **x** est une valeur d'entrée du réseau et ***y**** la valeur désirée correspondante pour les sorties des neurones de sortie.

L'architecture du réseau, la structure de ses connexions, ainsi que les fonctions d'activation, peuvent être fixées en fonction de la tâche que doit remplir le réseau.

Les valeurs des poids synaptiques sont, en général, déterminées par un processus algorithmique mettant en œuvre l'ensemble d'apprentissage.

Le but de l'apprentissage est donc de déterminer les valeurs ***W**** de la matrice ***W*** des poids des connexions du réseau de telle sorte que les sorties ***(y)*** soient proches des valeurs désirées ***y****.

W* est donc la solution d'un problème d'optimisation, consistant à minimiser une fonction de coût : ***E(W, D)*** [9]. Il existe plusieurs méthodes d'apprentissage, la plus utilisée est la méthode du gradient.

5.2.2. La méthode du gradient

La plupart des méthodes d'optimisation non linéaires sont basées sur la même stratégie. On choisit une valeur initiale ***W(0)*** de la matrice ***W***, puis on utilise un processus itératif dans lequel on tente d'optimiser la fonction ***E***. À chaque itération, cette optimisation implique deux étapes [9] **:**

- le choix de la direction dans laquelle on va chercher la valeur suivante ***W(t+1)***.
- le déplacement η le long de cette direction.

5.2.2.1 Centrage des données

Avant tout apprentissage, il est indispensable de normaliser et de centrer toutes les données de la base d'apprentissage, afin qu'ils soient actifs, en moyenne sur la partie linéaire de la fonction sigmoïde figure 3.10.

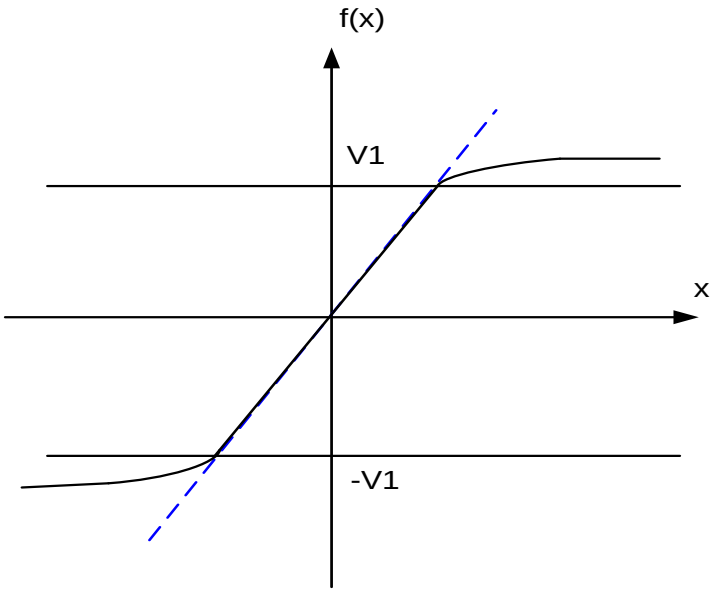


Figure 3.10. Centrage et normalisation des données de la base d'apprentissage.

En effet, si des entrées ont des grandeurs très différentes, celles qui sont « petites » n'ont pas d'influence sur l'apprentissage.

En pratique, il est donc recommandé, d'appliquer pour chaque vecteur d'entrée ***V*** la normalisation suivante :

Soit : *V*min, *V*max et *Vmoy* respectivement le minimum, le maximum et la moyenne de la variable, *V* considérée. On définit :

$$V_1 = Max\ (V_{moy} - V_{min}\ ,\ V_{max} - V_{moy}) \tag{3.4}$$

Les variables *V* sont alors centrées/réduites par

$$V = scal * \frac{(V - V_{moy})}{2 * V_{dev}} \tag{3.5}$$

5.2.2.2 Rétropropagation du gradient

L'algorithme de la rétro propagation du gradient est très connu et le plus utilisé dans les applications des réseaux de neurones. À chaque itération, on retire un exemple d'apprentissage ***(x, y*)*** et on calcule une nouvelle estimation de ***W(t)***. Cette itération est réalisée en deux phases :

a- Propagation

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

b- Rétro propagation

Cette étape est similaire à la précédente. Cependant, les calculs s'effectuent dans le sens inverse. À la sortie du réseau, on forme le critère de performance **J** en fonction de la sortie réelle du système et sa valeur désirée. Puis, on évalue le gradient de **J** par rapport aux différents poids en commençant par la couche de sortie et en remontant vers la couche d'entrée.

5.2.2.3 Calcul du gradient

Pour un exemple ***i*** d'un ensemble d'observation ***E***, la fonction de coût des moindres carrés est égale à la somme sur les ***N2*** valeurs de l'ensemble d'observation, de carrés des écarts entre la sortie du modèle (sortie du réseau de neurones = y_i) et la sortie désirée (Grandeur mesurée notée y_i^{de}). On cherche à minimiser, à chaque étape de mise à jour, le critère suivant :

$$J = \frac{1}{2}\sum_{K=1}^{N_2}(y_i^{des} - y_i)^2 \tag{3.6}$$

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

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

Le problème consiste à déterminer les poids ***W*** de toutes les couches qui minimisent le critère de performance **J.**

La mise à jour de ***W*** se fait selon la règle de delta :

$$\Delta W = -\eta \frac{\partial J}{\partial W} \tag{3.7}$$

$$W_{ij}^{(K)}(t+1) = W_{ij}^{(K)}(t) - \Delta W \tag{3.8}$$

Ce qui revient à déterminer les variations du critère de performance **J** par rapport aux variations des poids [12].

5.2.2.4 Evaluation de la couche de sortie

Calcul de $\frac{\partial J}{\partial W_{ij}}$ selon la figure 3.11

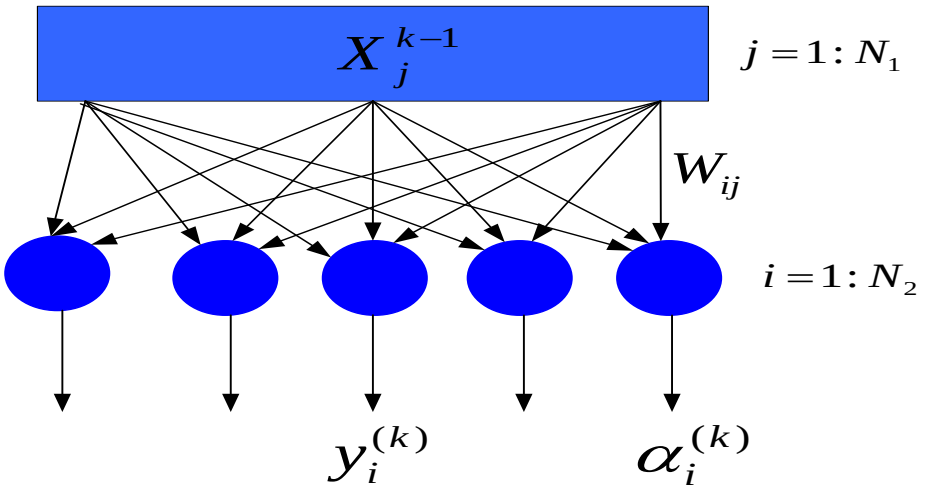


Figure 3.11. Représentation de la couche de sortie d'un réseau de neurones

Où N2 est le nombre de neurones dans la couche K (couche de sortie), **N1** le nombre de neurones dans la couche (K -1) (la dernière couche cachée).

$$\frac{\partial J}{\partial W_{ij}^{(K)}} = \frac{\partial J}{\partial y_i^{(K)}} \frac{\partial y_i^{(K)}}{\partial \alpha_i^{(K)}} \frac{\partial \alpha_i^{(K)}}{\partial W_{ij}^{(K)}} \tag{3.9}$$

$$J = \frac{1}{2}\sum_{i=1}^{N_2}(y_i^{des} - y_i)^2 \Rightarrow \frac{\partial J}{\partial y_i^{(K)}} = -(y_i^{des} - y_i^{(K)}) \tag{3.10}$$

$$y_i^{(K)} = g(\alpha_i^{(K)}) \Rightarrow \frac{\partial y_i^{(K)}}{\partial \alpha_i^{(K)}} = g'(\alpha_i^{(K)}) \tag{3.11}$$

$$\alpha_i^{(k)} = \sum_{j=1}^{N_1} W_{ij}.X_j^{(k-1)} \quad \Rightarrow \quad \frac{\partial \alpha_i^{(k)}}{\partial W_{ij}} = X_j^{(k-1)} \tag{3.12}$$

De l'équation (3.10) et (3.11) :

$$Err_i^{(K)} = \frac{\partial J}{\partial \alpha_i^{(K)}} = -(y_i^{des} - y_i^{(K)}).g'(\alpha_i^{(K)}) \tag{3.13}$$

Où $Err_i^{(k)}$ est l'erreur du $i^{ème}$ neurone dans la $k^{ème}$ couche.

Et de l'équation (3.12) et (3.13) : $\frac{\partial J}{\partial W_{ij}^{(K)}} = Err_i^{(K)}.X_j^{(K-1)}$

5.2.2.5 Evaluation des couches cachées

Calcul de $\frac{\partial J}{\partial W_{ij}}$ selon la figure 3.12.

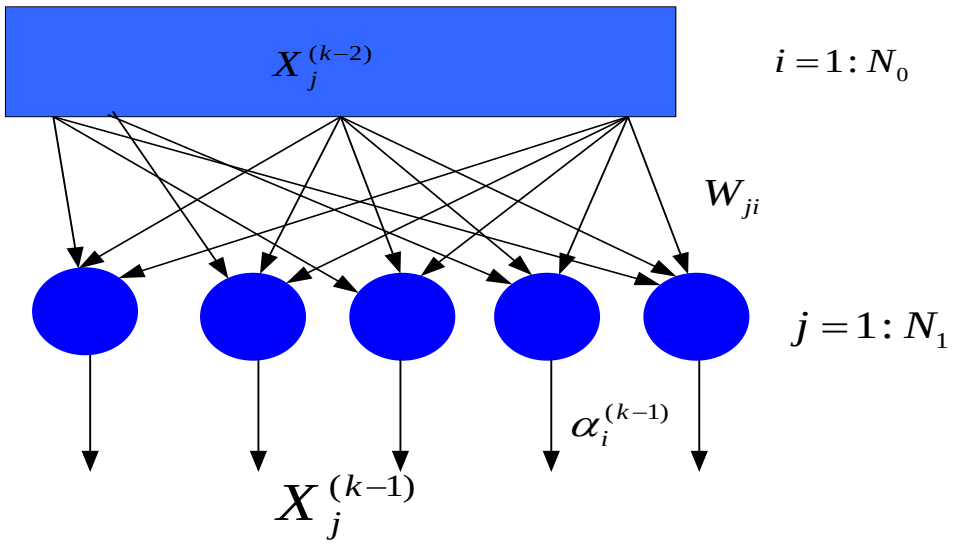


Figure 3.12. Représentation d'une couche cachée d'un réseau de neurones.

Où **N0** est le nombre de neurones de la couche (k-2).

$$\frac{\partial J}{\partial W_{ji}^{(K-1)}} = \frac{\partial J}{\partial X_{ji}^{(K-1)}} \frac{\partial X_j^{(K-1)}}{\partial \alpha_j^{(K-1)}} \frac{\partial \alpha_j^{(K-1)}}{\partial W_{ji}^{(K-1)}} \tag{3.14}$$

$$\frac{\partial J}{\partial W_{ji}^{(K-1)}} = \sum_{i=1}^{N_2} \frac{\partial J}{\partial \alpha_j^{(K)}} \frac{\partial \alpha_i^{(K)}}{\partial X_j^{(K-1)}} \Rightarrow \frac{\partial J}{\partial X_j^{(K-1)}} = \sum_{i=1}^{N_2} Err_i^{(K)}.W_{ji}^{(K)} \tag{3.15}$$

$$X_j^{(K-1)} = g(\alpha_j^{(K-1)}) \Rightarrow \frac{\partial J}{\partial \alpha_j^{(K-1)}} = g'(\alpha_j^{(K-1)}) \tag{3.16}$$

$$\alpha_j^{(K-1)} = \sum_{i=1}^{N_0} W_{ji}^{(K-1)}.X_i^{(K-2)} \Rightarrow \frac{\partial \alpha_j^{(K-1)}}{\partial W_{ji}^{(K-1)}} = X_i^{(K-2)} \tag{3.17}$$

De l'équation (3.15) et (3.16) on aura :

$$Err_j^{(K-1)} \equiv \frac{\partial J}{\partial \alpha_j^{(K-1)}} = \left[\sum_{i=1}^{N_0} Err_i^{(K)}.W_{Ji}^{(K)}\right] g'(\alpha_j^{(k-1)}) \Rightarrow \frac{\partial \alpha_j^{(K-1)}}{\partial W_{ji}^{(K-1)}} = X_i^{(K-2)} \tag{3.18}$$

$$\frac{\partial J}{\partial W_{ji}^{(K-1)}} = Err_j^{(K-1)}.X_i^{(K-2)} \tag{3.19}$$

5.2.2.6 Modification des paramètres du réseau de J en fonction du gradient

Dans l'étude précédente, nous avons vu comment évaluer le gradient simple de la fonction de coût ***J*** par rapport aux paramètres du réseau de neurones à chaque itération du processus d'apprentissage. Une fois que l'on dispose de cette évaluation, on effectue une modification des poids selon l'équation (3.3), afin d'approcher d'un minimum de la fonction de coût ***J*** dans l'espace des poids, pour cela il faut que la condition suivante soit vérifiée

$$\frac{\partial J}{\partial W_{ij}} = 0 \qquad (\forall i, j)$$

Le paramètre η dans l'équation (3.2) est appelé pas du gradient ou pas d'apprentissage

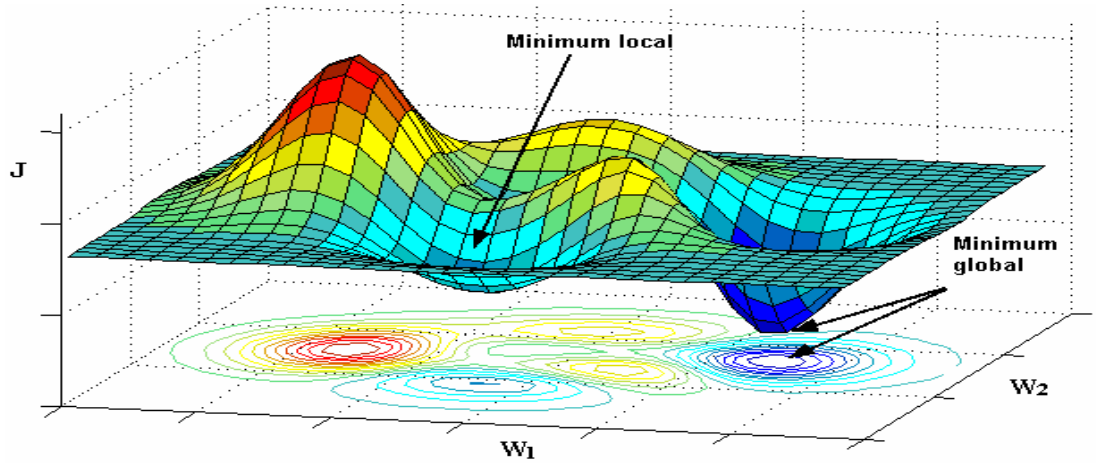


Figure 3.13. Représentation de la fonction de coût J d'un neurone à deux entrées pondérées W1 et W2 [11].

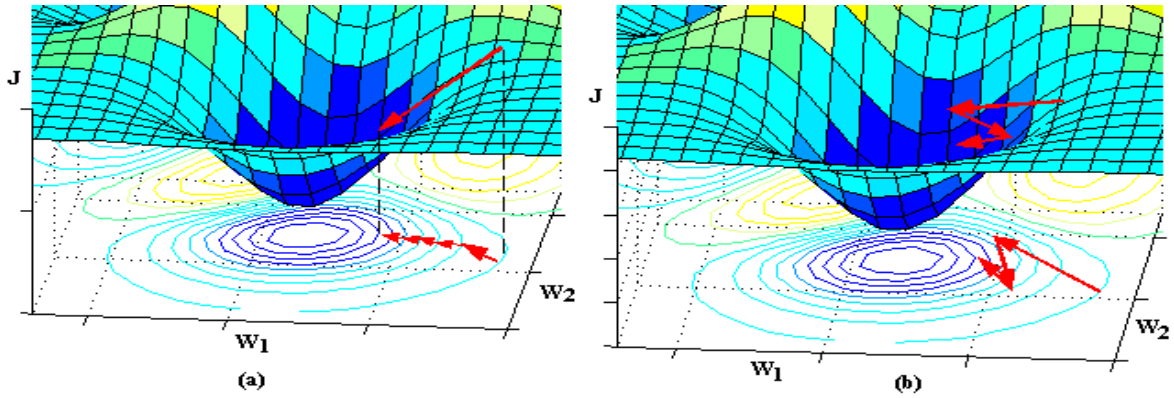


Figure 3.14. Minimisation de la fonction de coût J par la méthode du gradient

(a) Le pas du gradient est petit, convergence lente, le minimum global peut être atteint.

(b) Le pas du gradient est grand, convergence rapide, le minimum global est rarement atteint [9].

Cette méthode du gradient de premier ordre est simple, mais elle présente de nombreux inconvénients, entres autres :

A. Si le pas du gradient est trop petit, la décroissance du coût ***J*** est très lente figure 3.14(a). Si le pas est trop grand, le coût ***J*** peut augmenter ou osciller, cette situation est illustrée sur la figure 3.14 (b), qui représente une forme 3D d'une fonction de coût ***J*** ayant des minimum locaux et un minimum global, où sa projection sur le plan de base formé par deux variables (poids ***W1*** et ***W2***) donne un contour de plusieurs niveaux de la fonction de coût.

B. Au voisinage d'un minimum de la fonction de coût, la pente de la descente est faible, ce qui revient à dire que le gradient de cette fonction tend vers zéros c.-à-d $\frac{\partial J}{\partial W_{ji}} \approx 0$; à ce niveau, multiplier cette grandeur par le pas d'apprentissage η, n'améliorera pas considérablement le résultat, et donc l'évolution du vecteur des paramètres du réseau lors de la mise à jour par l'équation (3.8) devient très lente.

C. Si la courbure de la surface de coût varie beaucoup, la direction du gradient peut être très différente de la direction qui mènerait vers le minimum ; c'est le cas si le minimum recherché se trouve dans une « vallée » longue et étroite (les courbes de niveau sont des ellipsoïdes allongés au voisinage du minimum (figure 3.14) [9].

6. La commande MLI neuronal

Introduction

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

6.1 Architecture du système

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

Pour la couche cachée, on prend un seul neurone parce que le système possède une seul entrée, la fonction d'activation est de type tangent sigmoïde (tansig), concernant la couche de sortie, elle est constituée de deux neurones et la fonction d'activation a une forme linéaire (Purelin).

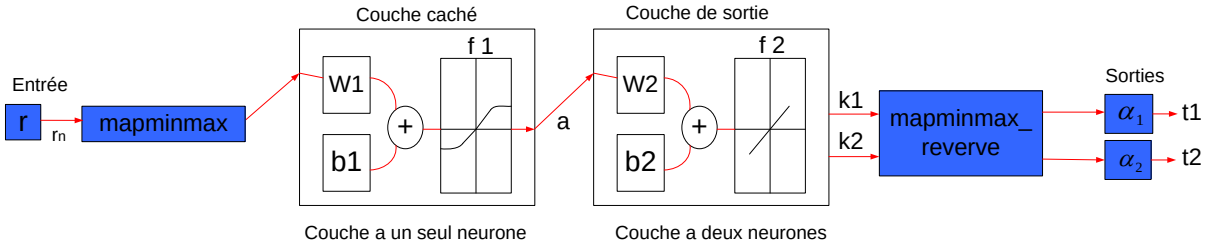


Figure 3.15. Architecture de la commande MLI neuronal.

D'après la figure 3.12, le taux de modulation r qui représente l'entrée de notre réseau varie entre 0.61 et 0.71, cet intervalle de variation correspond à la partie non-linéaire qu'on doit estimer. Les paramètres du système neuronal sont cités ci-dessous :

r : Vecteur d'entrée du réseau de neurone (taux de modulation).

mapminmax : Fonction de normalisation directe, elle permet de générer la matrice dans l'intervalle [-1, +1], l'expression de la fonction de normalisation appliqué sur la fonction y=f(x) est donnée par la formule suivante :

$$y = (y_{max}-y_{min})*(x-x_{min})/(x_{max}-x_{min}) + y_{min} \quad (3.20)$$

W1 : Matrice des poids de la couche cachée.

b1 : Matrice des seuils de la couche cachée.

W2 : Matrice des poids de la couche de sortie.

b2 : Matrice des seuils de la couche de sortie.

mapminmax_reverse : Fonction de normalisation inverse, elle permet de retourner à l'état initial (elle donne une estimation de la matrice d'apprentissage), l'expression de la fonction de

normalisation inverse appliqué sur la fonction f(x) est donnée par la formule suivante :

$$x=f^{-1}(y) = (y-y_{min})*(x_{max}-x_{min})/ (y_{max}-y_{min}) + x_{mi} \quad (3.21)$$

Les valeurs des angles de commutations ($\alpha_1 et \alpha_2$), qui représentent la sortie du réseau de neurones dépendent du taux de modulation r.

f 1 : Fonction tangente-sigmoïde (tansig).

f 2 : Fonction purement linéaire (purelin).

a : La sortie de la couche cachée.

r_n : Taux de modulation normalisée.

k1, k2 : Sorties de la couche de sortie.

t1, t2 : Les instants de commutations.

6.2 Etablissement de l'expression des instants de commutations et période du signal de sortie :

L'expression des paramètres intermédiaires est donnée ci-dessous :

$$-1 \le r_n \le 1 \quad (3.22)$$

$$r_n = 20*r - 13.2 \quad (3.23)$$

$$0.61 \le r \le 0.71 \quad (3.24)$$

$$1.0677 \le w1*r_n + b1 \le 1.3419 \quad (3.25)$$

L'approximation de la fonction tangente-sigmoïde est donnée par l'expression suivante :

$$f1 = 0.3045*x + 0.4635 \quad (3.26)$$

$$a = -0.0417*r_n + 0.8303 \quad (3.27)$$

$$k_1 = f_2(a*w_2^1 + b_2^1) = a*w_2^1 + b_2^1 = 0.9672*r_n - 0.0303 \quad (3.28)$$

$$k_2 = f_2(a*w_2^2 + b_2^2) = a*w_2^2 + b_2^2 = -0.9672*r_n + 0.0303 \quad (3.29)$$

$$\alpha_1 = 169.7250*r - 84.9393 \quad (3.30)$$

$$\alpha_2 = 169.7250*r - 31.1029 \quad (3.31)$$

$$r_1 = 1000*r \quad (3.32)$$

$$610 \le r_1 \le 710 \quad (3.33)$$

$$t_1(\mu s) = 4714 - (2359000 / r_1) \quad (3.34)$$

$$t_2(\mu s) = 4714 - (863970 / r_1) \quad (3.35)$$

$$2*r = f / f_0 \quad (3.36)$$

$$T(\mu s) = 1/(2*r*f_0) = 10^7 / r_1 \quad (3.37)$$

6.3. Bases de données :

Pour déterminer les paramètres du système neuronal (poids et seuils) de notre réseau, on écrit un programme dans le logiciel MATLAB, qui permet de résoudre le système d'équations non-linéaire 2.8 en utilisant la méthode « théorie résultante des polynômes symétriques » afin générer la base de données, qui sera utilisée dans l'étape d'apprentissage.

Le Tableau 3.1. montre la base des données des angles exacts de commutation calculés par notre programme, lorsque r varie de 0.61 à 0.71.

r	$\propto_{1exacte}$	$\propto_{2exacte}$
0.61	18.5954	89.4045
0.62	19.9391	88.0608
0.63	21.3312	86.6687
0.64	22.7782	85.2217
0.65	24.2882	83.7117
0.66	25.8713	82.1286
0.67	27.5410	80.4589
0.68	29.3147	78.6852
0.69	31.2169	76.7830
0.70	33.2830	74.7169
0.71	35.5683	72.4316

Tableau 3.1 Base des données des angles exacts de commutation

6.4. Etablissement des paramètres du système

Dans la phase d'apprentissage de notre réseau de neurones, nous exploitons la base de données calculées précédemment afin de déterminer les paramètres (poids et seuil) du système.

Le programme d'apprentissage est basé sur la méthode du gradient, il possède deux conditions d'arrêt : la performance et le nombre d'itération ; on lance l'apprentissage jusqu'à ce que l'une de ces conditions soit vérifiée.

Le tableau 3.2 donne un exemple des poids et seuils calculé par le programme lorsque r varie entre 0.61 et 0.71.

W1	W2	b1	b2
-0.1371	-23.1704	1.2048	19.2094
	23.1704		-19.2094

Tableau 3.2 Les poids et les seuils pour m=2.

6.5. Simulation

Les résultats de simulation obtenus sont mentionnés ci-dessous dans le tableau 3.3.

r	$\propto_{1exacte}$	$\propto_{1approximé}$	$Err\propto_1$	$\propto_{2exacte}$	$\propto_{2approximé}$	$Err\propto_2$
0.61	18.5954	18.6086	-0.013148	89.4045	89.3914	0.013148
0.62	19.9391	19.9310	0.008218	88.0608	88.0690	-0.008218
0.63	21.3312	21.3176	0.013625	86.6687	86.6824	-0.013625
0.64	22.7782	22.7712	0.007057	85.2217	85.2288	-0.007057
0.65	24.2882	24.2942	-0.005986	83.7117	83.7058	0.005986
0.66	25.8713	25.8893	-0.017918	82.1286	82.1107	0.017918
0.67	27.5410	27.5592	-0.018174	80.4589	80.4408	0.018174
0.68	29.3147	29.3065	0.008188	78.6852	78.6935	-0.008188
0.69	31.2169	31.1340	0.082866	76.7830	76.8660	-0.082866
0.70	33.2830	33.0444	0.238630	74.7169	74.9556	-0.238630
0.71	35.5683	35.0404	0.527970	72.4316	72.9596	-0.527970

Tableau 3.3 Angles exacts et approximées pour m=2.

Les symboles utilisés dans le tableau sont précisés ci-dessous :

$\propto_{1exacte}= a_{1exacte}$: La valeur exacte de l'angle de commutation $\propto_1$.

$\propto_{2exacte}=a_{2exacte}$: La valeur exacte de l'angle de commutation $\propto_2$.

$\propto_{1approximé}=a_{1approximé}$: La valeur approximé par le système neuronal de l'angle de commutation $\propto_1$.

$\propto_{2approximé}=\propto_{2approximé}$: La valeur approximé par le système neuronal de l'angle de commutation $\propto_2$.

$Err\propto_1$: L'erreur commise sur le calcul de $\propto_1$.

Err$\propto_2$: L'erreur commise sur le calcul de $\propto_2$.

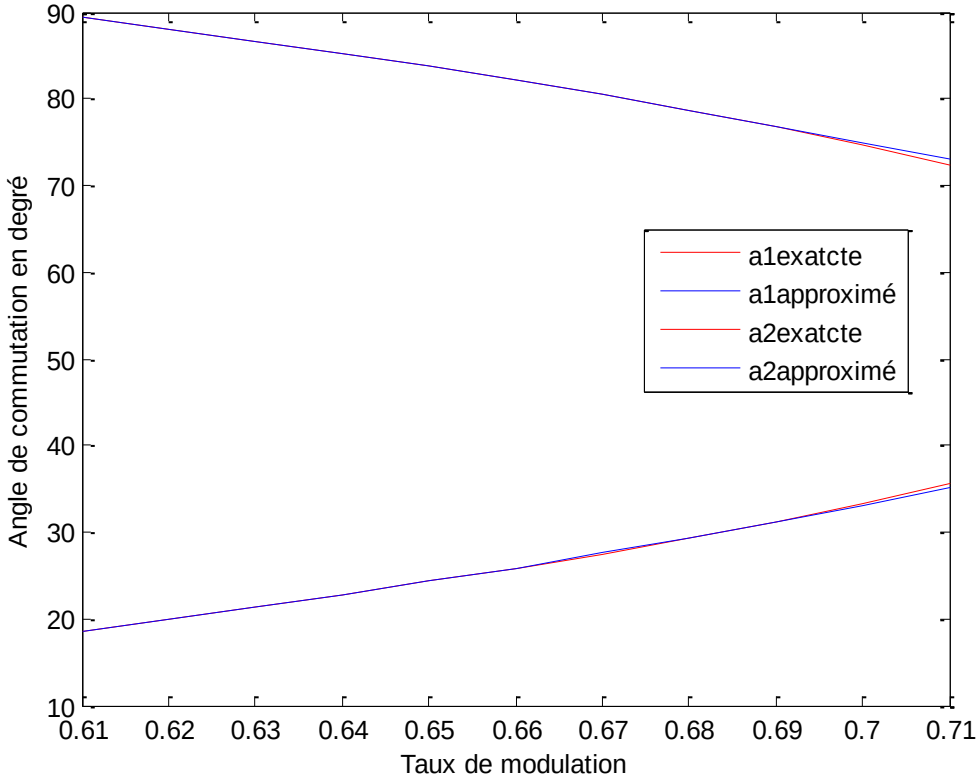


Figure 3.16. Allure des angles exacts et angles approximés.

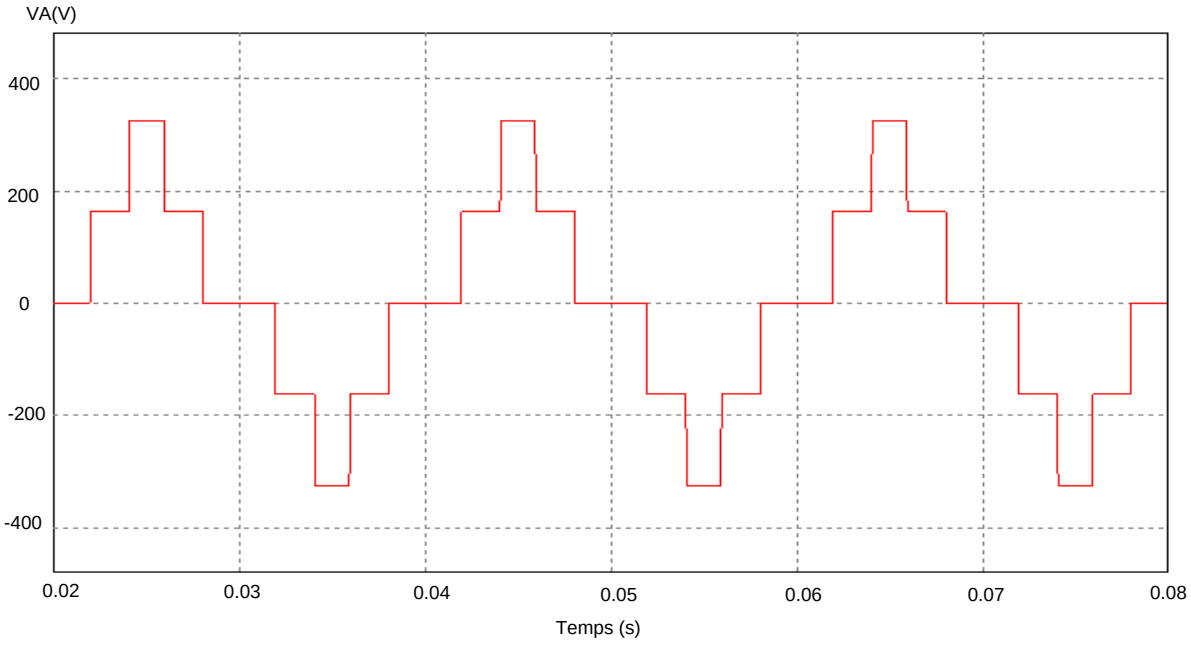


Figure 3.17. Allure de la tension simple de sortie de l'onduleur

$$\alpha_{1approximé} = 35.0404° et \alpha_{2approximé} = 72.9596° .$$

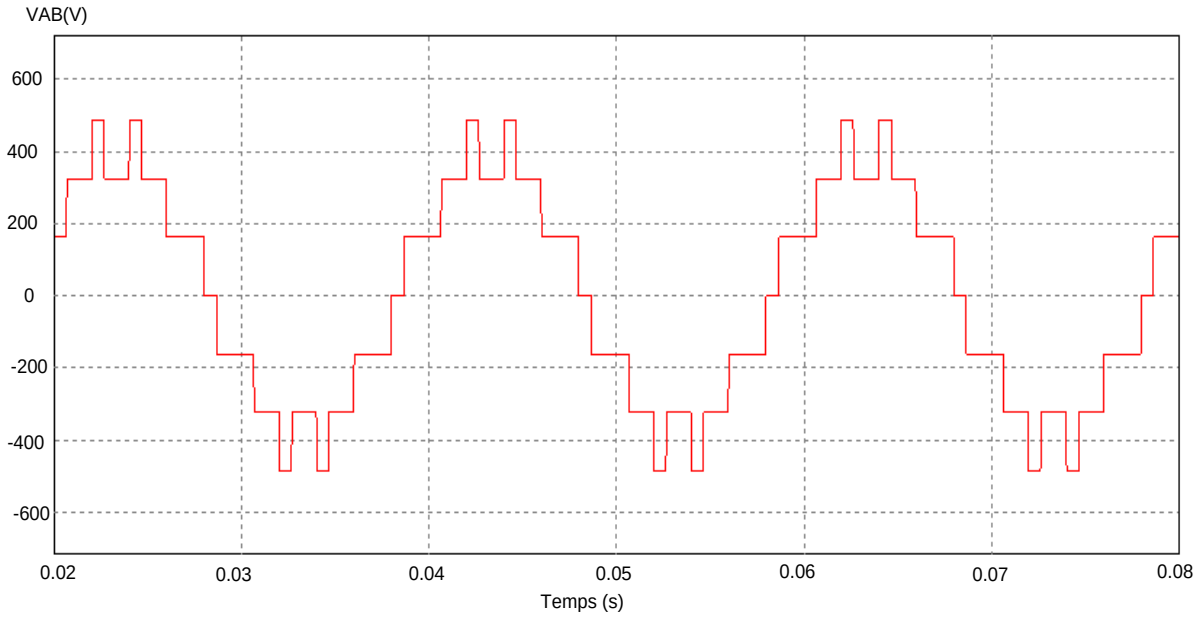


Figure 3.18. Allure de la tension composée de sortie de l'onduleur

$\alpha_{1approximé} = 35.0404° et \alpha_{2approximé} = 72.9596°$.

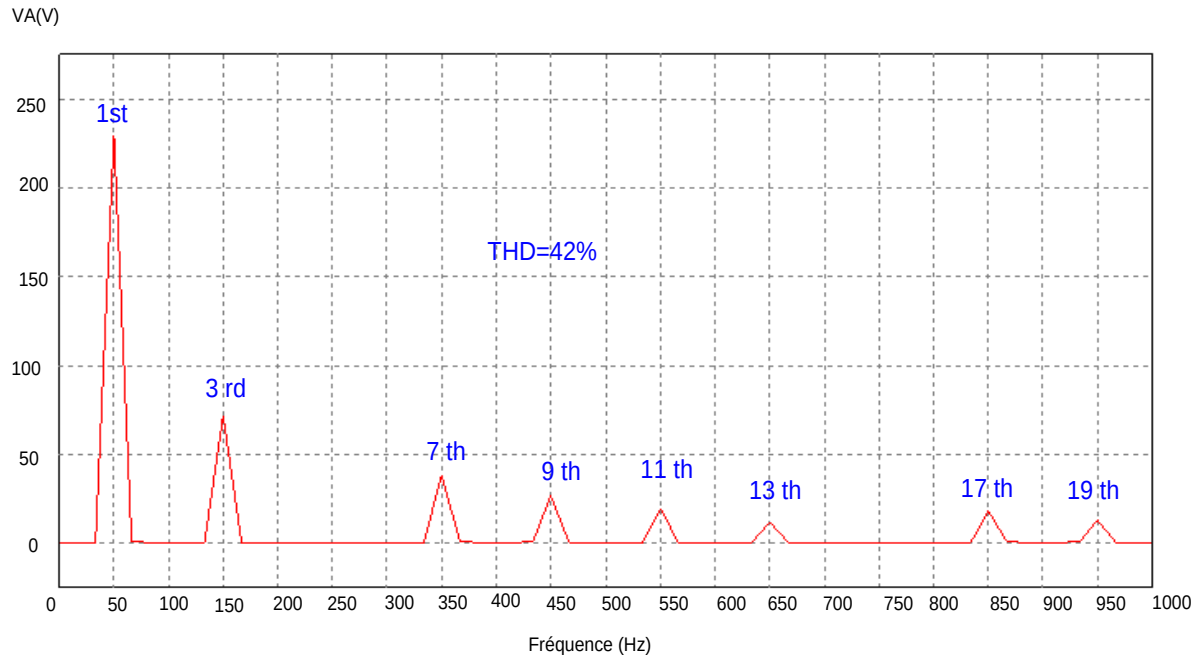


Figure 3.19. Spectre de la tension simple de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz,

$\alpha_{1approximé} = 35.0404° et \alpha_{2approximé} = 72.9596°$.

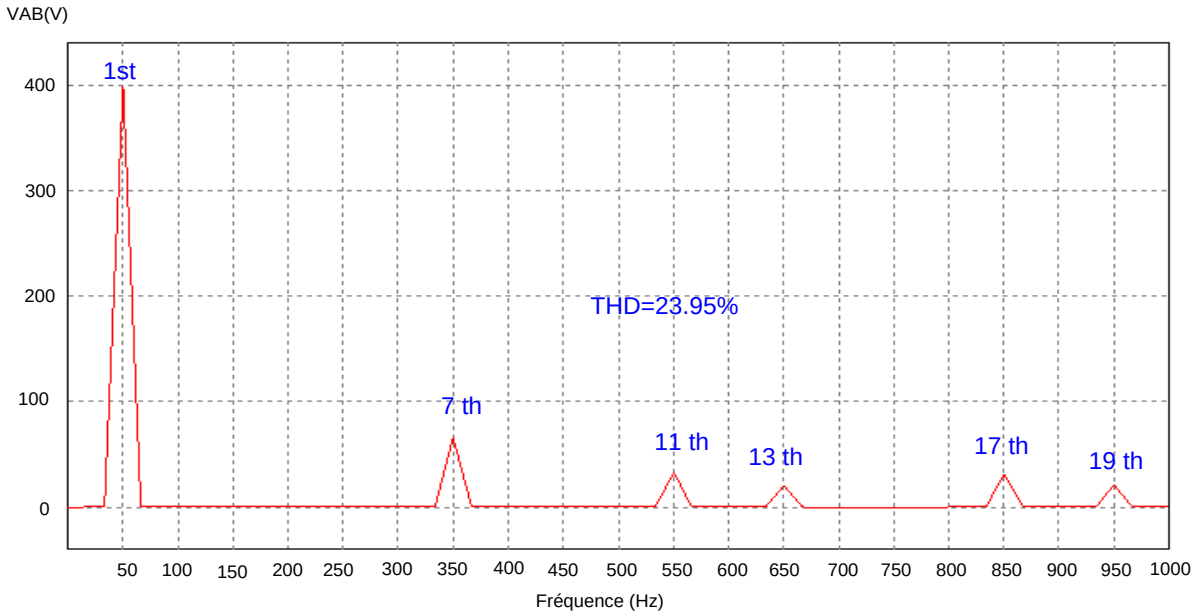


Figure 3.20. Spectre de la tension composée de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz,

$\alpha_{1approximé} = 35.0404° et \alpha_{2approximé} = 72.9596°$.

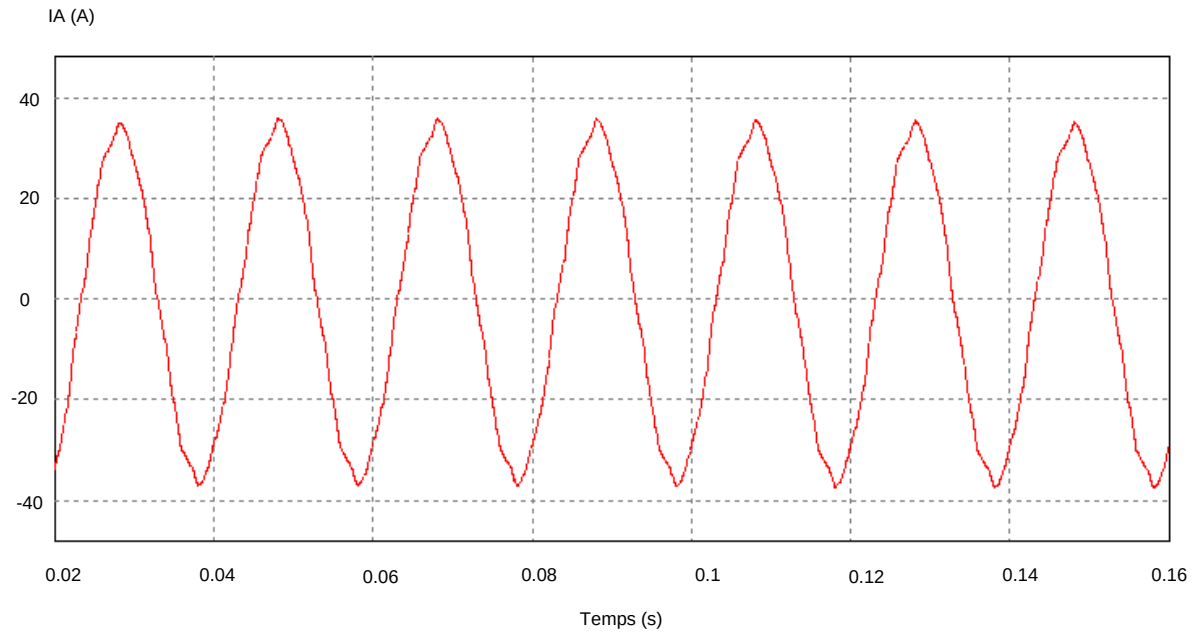


Figure 3.21. Allure du courant de sortie d'un bras de l'onduleur cinq niveaux dans la charge

$\alpha_{1approximé} = 35.0404° et \alpha_{2approximé} = 72.9596°$.

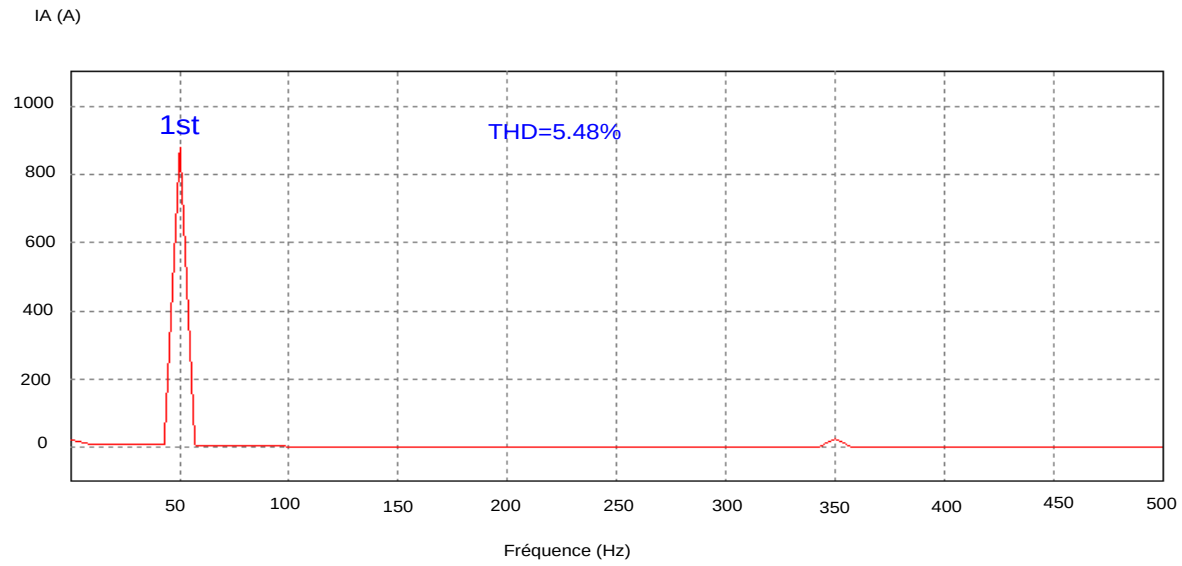


Figure 3.22. Spectre du courant de la charge de l'onduleur cinq niveaux commandé par l'algorithme MLI calculée pour m=2, r=0.71, f=50Hz

$\alpha_{1approximé} = 35.0404°$ *et* $\alpha_{2approximé} = 72.9596°$.

6.6. Interprétation des résultats :

D'après le Tableau 3.3 et la figure 3.16, on constate que les angles de commutation approximés par notre réseau de neurones sont très proches des angles exacts et l'erreur maximale est inférieure à 0.53 degrés.

De la figure 3.20, on remarque que l'harmonique n°5 est éliminée, donc l'estimation des angles de commutation $\alpha_1 et \alpha_2$ est précise.

La figure 3.21 montre que la forme du courant dans la charge est très proche d'une forme sinusoïdale ; de la figure 3.22, on constate que toutes les harmoniques du courant sont éliminées sauf le fondamental.

Conclusion

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

Ces résultats montrent que l'algorithme MLI neuronal que nous avons proposé présente une bonne précision dans l'estimation des angles de commutation, ainsi qu'une efficacité dans l'élimination des harmoniques désirées.

Implémentation de la commande MLI neuronale on-line sur FPGA

1. Les Circuits FPGA et langage VHDL

1.1. Les Circuits FPGA

1.1.1. Introduction

Les composants logiques programmables sont des circuits composés de nombreuses cellules logiques élémentaires librement assemblables, celles-ci sont connectées d'une manière définitive ou réversible par programmation, afin de réaliser les fonctions logiques désirées. Un circuit FPGA (Field-Programmable Gate Array) est un circuit intégré composé d'éléments logiques programmables CLB (Configurable Logic Block) et IOB (Input Output Bloc), ils sont répartis régulièrement et reliés entre eux grâce à des connections qui forment une matrice de routage programmable afin d'obtenir un comportement spécialisé du circuit dans sa globalité.

1.1.2. Avantages et inconvénients des FPGA

Avantages	Inconvénients
Technologie facile à maîtriser. Temps de développement réduit. Reprogrammable. Idéal pour le prototypage. Coûts peu élevés. Parallélisme de traitement. Flexibilité et possibilité de réduire fortement les délais de développement et de commercialisation. Reconfiguration (parfois) en temps réel.	Performances non optimisées. Temps de réponse long par rapport aux ASIC.

1.1.3. Architecture interne des FPGA :

L'architecture interne des FPGA est différente d'un fabricant à un autre, mais leurs ressemblances peuvent être représentées dans le schéma de la figure suivante [5] :

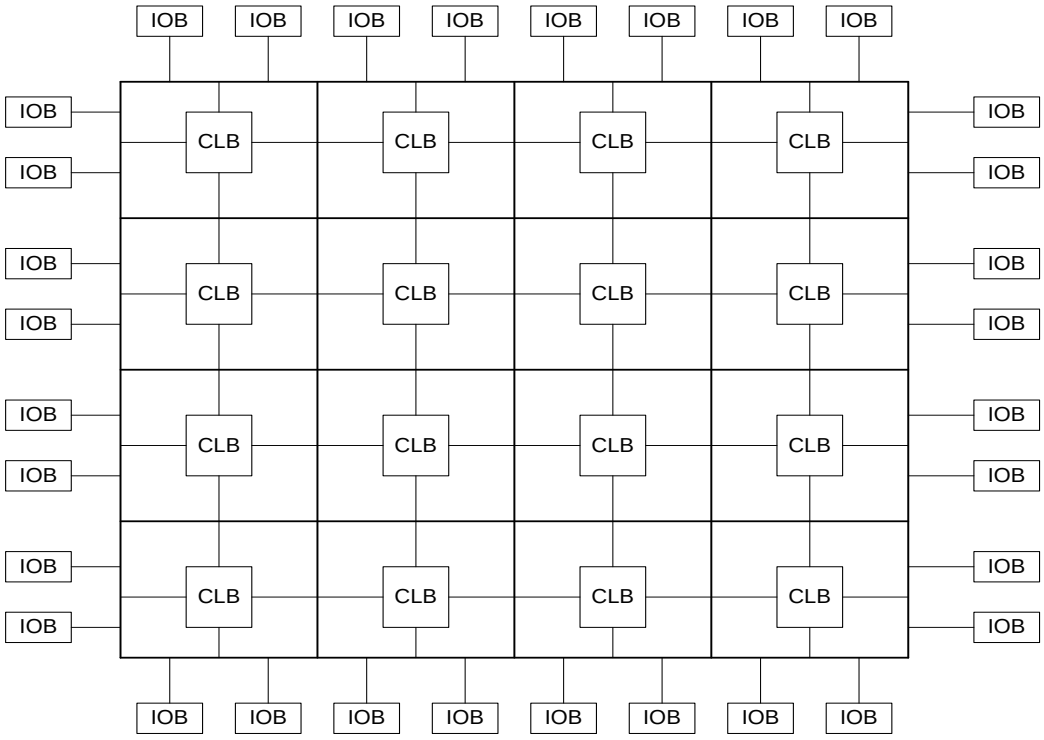


Figure 4.1. Architecture interne d'un FPGA.

Les macro-cellules internes sont appelées :

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

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

a. Structure des CLB

Les CLB représentent la ressource principale à implémenter pour la logique séquentielle tout comme pour la logique combinatoire, chaque CLB est constitué de SLICES interconnectées, ces slices sont disposées en paires et chaque paire est disposée en colonne.
La paire de gauche est appelé SLICE1et celle de droite SLICE0.

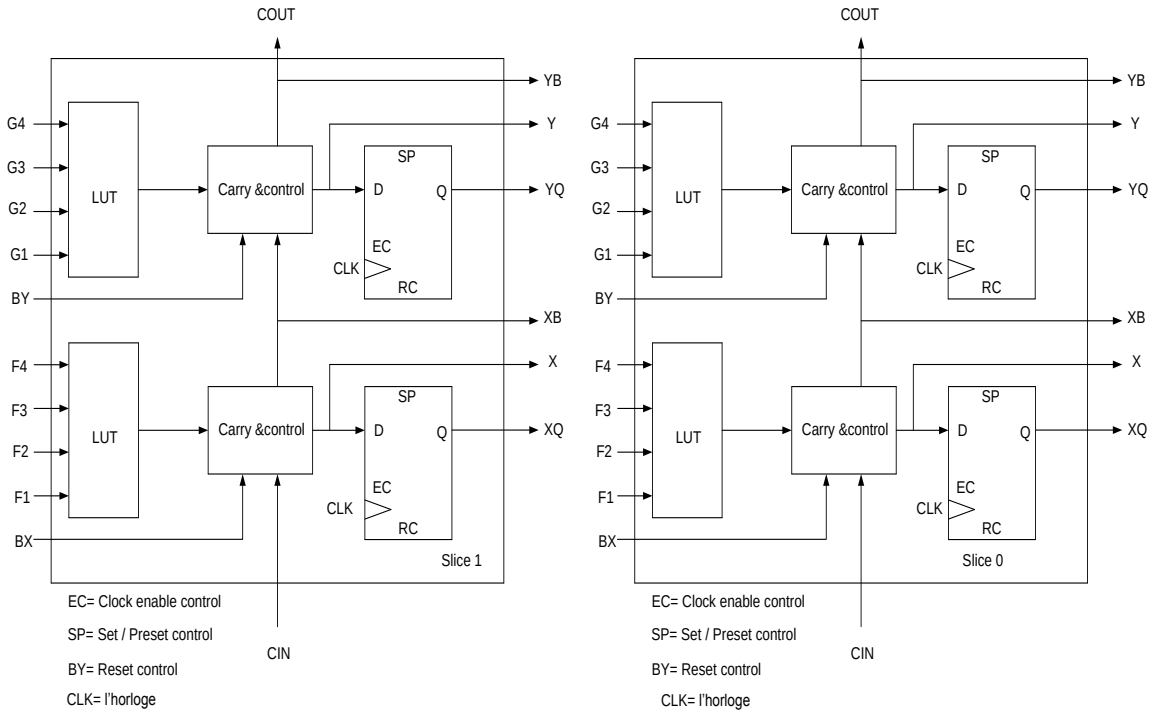


Figure 4.2. Architecture d'un CLB de famille Virtex [5].

L'architecture d'un SLICE est comme suit :

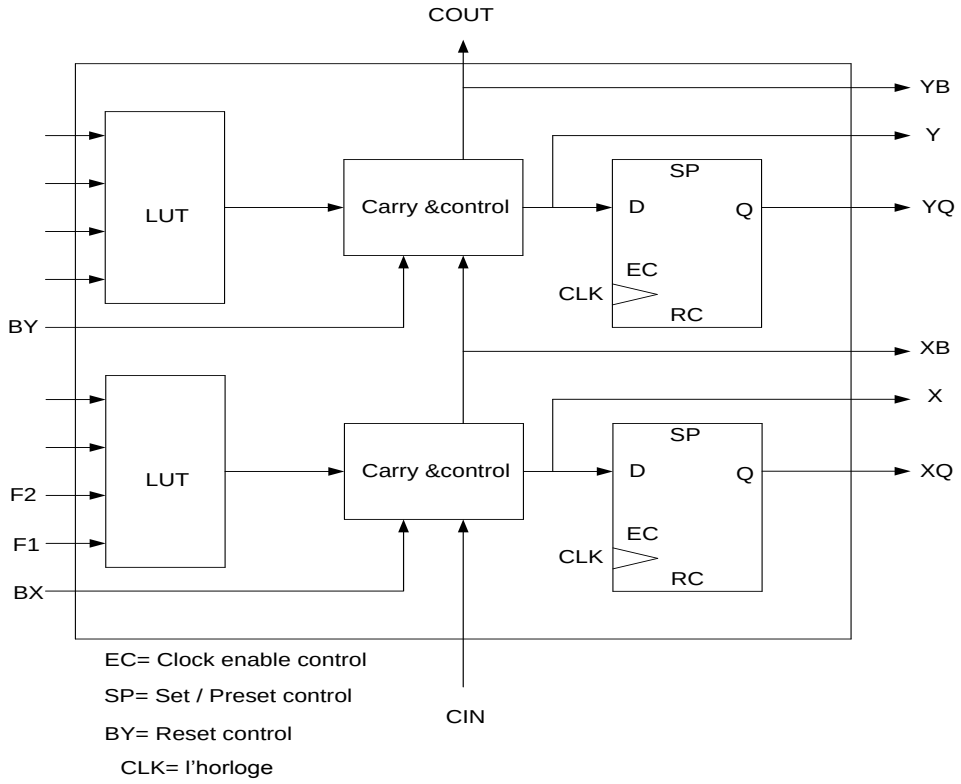


Figure 4.3. Architecture d'un SLICE de famille Virtex [5].

b. Structure des IOB

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

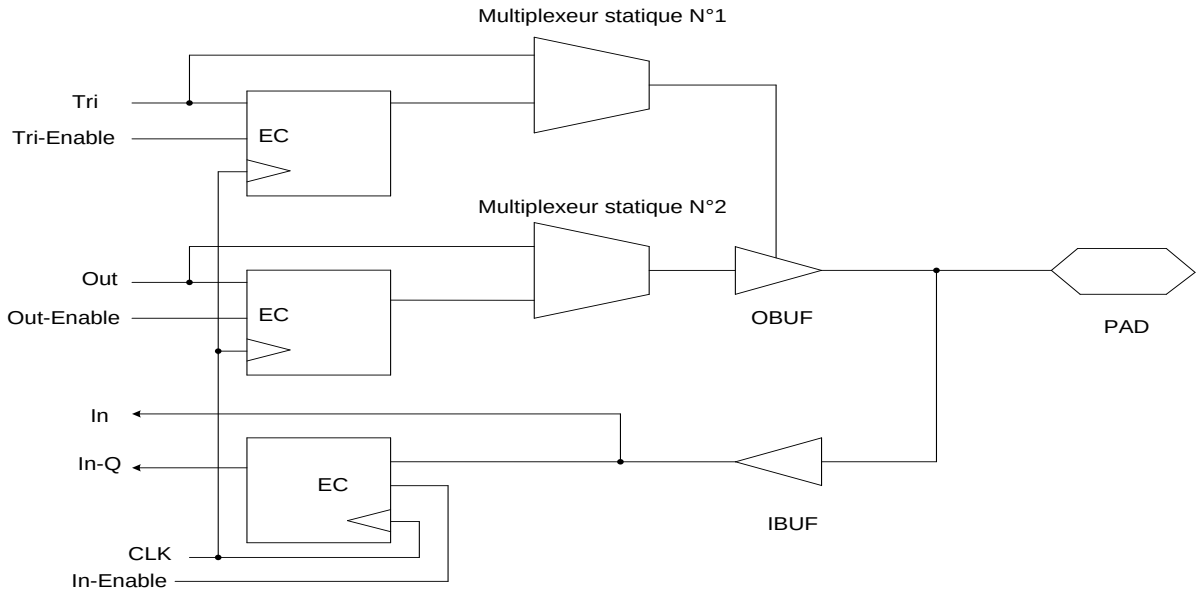


Figure 4.4. Architecture d'un IOB de famille Virtex [5].

1.2. Langage VHDL

1.2.1. Les langages de description matérielle HDL (Hardware Description Langages)

La description matérielle a connu une évolution au fur et à mesure que les moyens informatiques et les technologies d'intégration des transistors (de plus en plus denses) évoluaient. Les langages de description Matérielle ont connu une évolution en quatre étapes pour arriver à maturité [12] :

a. Dessin au micron

Au début, la conception de circuits intégrés était manuelle. On dessinait les composants (transistors) à la main, sur un papier spécial (Mylar) avec des crayons de couleurs. C'est le dessin au micron. Une telle technique limitait la complexité des dispositifs conçus.

b. Les langages de description

Plus la technologie microelectronique évoluait vers l'intégration de nombres plus important de transistors, plus la necessité de nouveaux outils de conception s'imposait ; c'est ainsi que les langages de déscription matérielle (HDL) ont fait leur apparition, ils avaient pour but de modéliser, donc de simuler, mais aussi de concevoir. Des outils informatiques (placeurs-routeurs) permettant de traduire automatiquement une description textuelle en

dessins. Le concepteur aborde les problèmes à un niveau d'abstraction plus élevé. Il manipule des objets élémentaires de l'ordre de la dizaine de transistors (les portes logiques).

c. Les schémas

L'apparition des interfaces graphiques et donc des éditeurs de schémas simplifie le travail de conception des circuits intégrés. En effet , il est en général beaucoup plus facile de lire et de comprendre un schéma qu'une déscription textuelle. La déscription textuelle est remplacée par une description schématique.

D. L'abstraction fonctionnelle

Les langages fonctionnels (ou comportementaux) de description de matériel, grâce aux nouvelles possibilités de description à un niveau d'abstraction plus élevé, ont répondu à des besoins fondamentaux des concepteurs de circuits intégrés :

1. La réduction des temps de conception.
2. L'accélération des simulations qui devenaient prohibitives avec l'accroissement de la complexité des dispositifs.
3. La normalisation des échanges : des langages normalisés et universellement de la complexité des dispositifs, entre partenaires industriels, entre fournisseurs et clients. Verilog, VHDL et System C sont des exemples.
4. L'anticipation : grâce aux modèles HDL, il est possible de concevoir un système alors que ses composants ne sont pas encore disponibles.
5. La fiabilité : les langages HDL sont conçus pour limiter en principe les risques d'erreur.
6. La maintenabilité et la réutilisabilité : les modifications et adaptations sont rendues plus simples donc moins risquées et moins couteuses.

L'augmentation du nombre de portes intégrées (plusieurs millions de portes) sur une même puce fait apparaitre la nécessité de développer des outils encore plus puissants : les synthétiseurs logiques. Le dispositif à modéliser ou à concevoir sera représenté par sa fonction et non plus par sa structure. C'est le synthétiseur logique qui déterminera la structure automatiquement à partir de la fonction.

1.2.2. Utilité des HDL

Les langages HDL offrent deux avantages majeurs en plus de la simplicité alors de la conception :

a. Simulation : Le but de la modélisation, c'est la simulation, il éxiste deux points importants pour la simulation :

- La fidélité : un modèle se doit d'être aussi précis que possible dans son champ d'application.
- L'éfficacité : le modèle doit pouvoir être simulé le plus rapidement possible et doit être portable, réutilisable et facile à maintenir.

Pour simuler un modèle, il faut disposer du modèle, bien sûr, mais aussi de stimuli, c'est-à-dire de la description des signaux d'entées du modèle au cours du temps. Ces stimuli sont appelés les TESTBENCHES (en Anglais).

Les langages fonctionnels de description de matériel permettent de simplifier la conception en analysant automatiquement les résultats en cours de simulation.

b. Synthèse : Les langages fonctionnels de description matérielle servent aussi à concevoir. Il ne s'agit plus de modéliser en vue de la simulation, mais de décrire les objets qui seront véritablement fabriqués. Si les considérations de vitesse d'exécution en simulation existent toujours (la description sera simulée avant d'alimenter le synthétiseur, afin de vérifier que la fonction décrite est bien la fonction désirée) elles ne sont plus prioritaires. En effet, ceci est pour que ce dernier produise la description structurelle la plus économique possible (et donc la surface de silicium la plus petite possible).

1.2.3. Structure d'un programme VHDL

A l'origine, avant même la mise en place du VHDL, le programme VHSIC (Very High Speed Integrated Circuits), impulsé par le département de la défence des Etats Unis dans les années 1970-1980, a donné naissance à un langage VHSIC-HDL ou VHDL. Deux normes successives (IEEE-1976-1987 et 1993) en ont stabilisé la définition, complétée par des extensions plus récentes ou à venir. L'évolution prévue est la création d'un langage commun analogique-numérique, dont le VHDL, sous sa forme actuelle, constituerait la facette numérique.

Un opérateur élémentaire, un circuit intégré, une carte électronique ou un système complet, est complètement défini par des signaux d'entrées et de sorties et par la fonction réalisée de façon interne. Les concepteurs du VHDL ont adopté l'approche suivante : n'importe quel système est considéré comme une boite noire, cette dernière a des entrées et des sorties. Ils ont appelé la boîte noire « ENTITY ». N'importe quel système électronique effectue des opérations sur le signal d'entrée pour donner le résultat du traitement en sortie. Ces opérations sont le contenu de la boite noire, ce contenu est appelé « ARCHITECTURE »

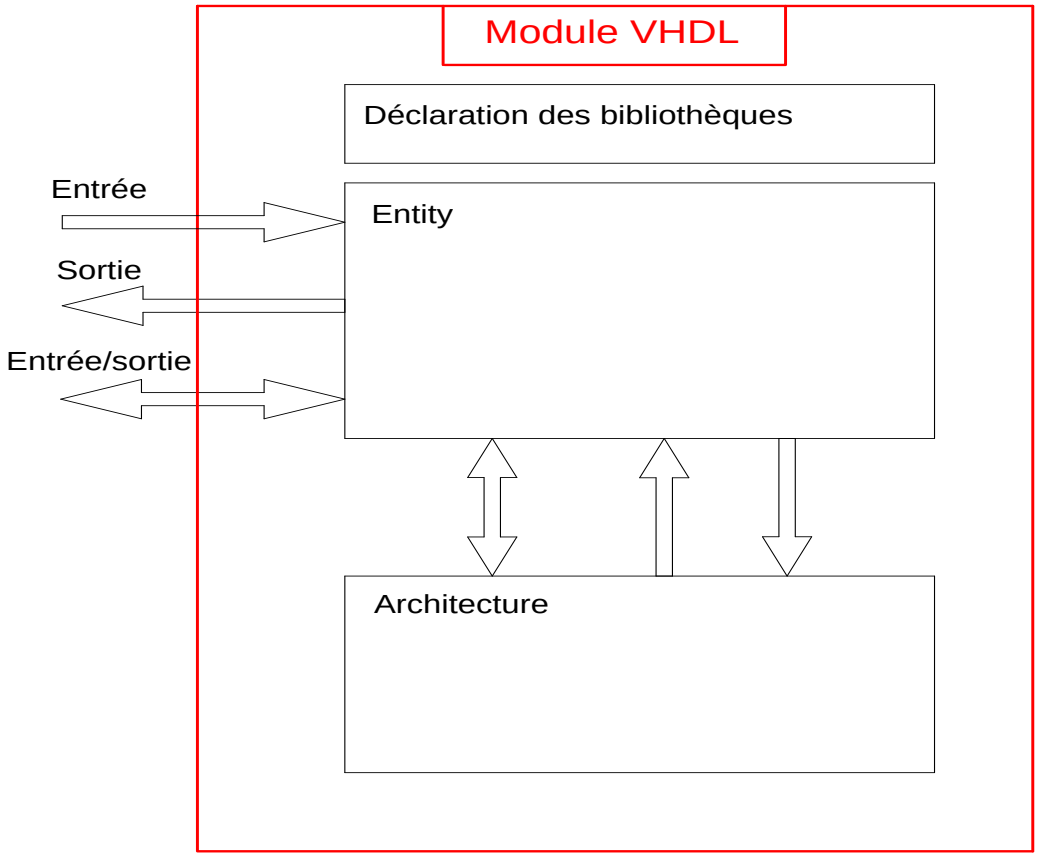


Figure 4.5. Structure de base d'un module sous VHDL.

Une description VHDL est composée de deux parties essentielles :

a) Déclaration des bibliothèques **:** toute description VHDL utilisée pour la synthèse a besoin de bibliothèques. L'IEEE (Institut of Electrical and Electronics Engineers) les a normalisées et plus particulièrement la bibliothèque IEEE1164. Elles contiennent les définitions des types de signaux électroniques, des fonctions et sous -programmes permettant de réaliser des opérations arithmétiques et logiques, etc...

Syntaxe:

Library ieee;

Use ieee.std_logic_1164.all;

Use ieee.numeric_std.all;

Use ieee.std_logic_unsigned.all;

La directive **Use** permet de sélectionner les bibliothèques à utiliser. L'entité (ENTITY) : elle définit les entrées et les sorties.

Syntaxe:

entity nom de l'entité is

port (description des signaux d'entrées /sorties …) ;
end nom de l'entité;
b) L'architecture: elle contient les instructions VHDL permettant de réaliser le fonctionnement attendu.
Syntaxe:
architecture « nom de l'architecture » of « nom de la fonction » is
begin
end « nom de l'architecture » ;

1.2.4. Les deux modes de travail en VHDL

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

a. Le mode combinatoire

En mode combinatoire (ou concurrent), toutes les instructions d'une description VHDL sont évaluées et affectent les signaux de sortie en même temps, l'ordre dans lequel les instructions sont écrites n'a donc aucune importance.
Avec VHDL il faut essayer de penser à la structure qui va être générée par le synthétiseur pour écrire une bonne description, cela n'est pas toujours évident. En effet la description génère des structures électroniques, c'est la grande différence entre une description VHDL et un langage informatique classique.

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

b. Le mode séquentiel

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

• **Le mode séquentiel asynchrone :** dans lequel les changements d'état des sorties peuvent se produire à des instants difficilement prédictibles (retards formés par le basculement d'un nombre souvent inconnu de fonctions),

• **Le mode séquentiel synchrone :** dans lequel les changements d'état des sorties se produisent tous à des instants quasiment connus (un temps de retard après un front actif de l'horloge).

1.2.5. Etapes nécessaires de développement d'un projet sur FPGA

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

Avec les outils actuels, il est possible de disposer de fichiers VHDL à chaque étape. Le même fichier de simulation (test-bench) est ainsi utilisable pour vérifier le fonctionnement de la description à chaque étape de la procédure de développement. La figure 4.6 donne les différentes étapes nécessaires au développement d'un projet sur circuit FPGA.

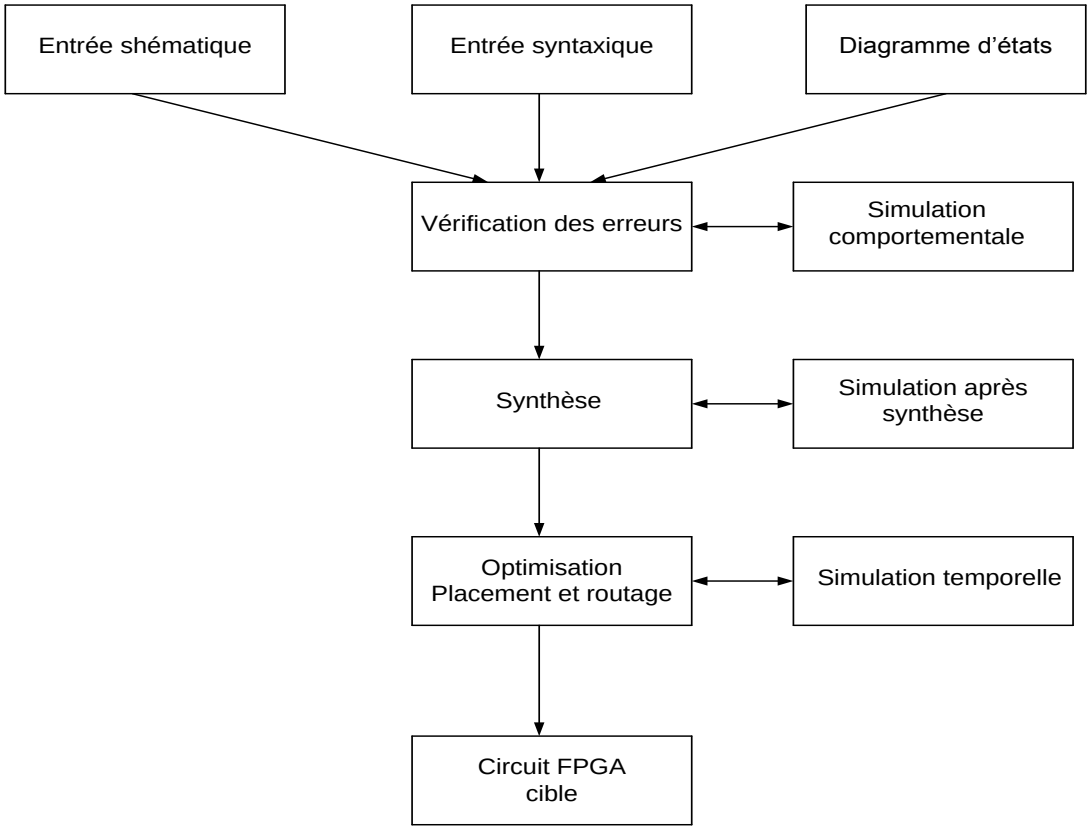


Figure 4.6. Organisation fonctionnelle de développement d'un projet sur un circuit FPGA.

a. Saisie du texte VHDL

La saisie du texte VHDL s'effectue dans le logiciel « ISE Xilinx Project Navigator ». Ce logiciel propose une palette d'outils permettant d'effectuer toutes les étapes nécessaires au développement d'un projet sur circuit FPGA. Il possède également des outils permettant de mettre au point une entrée schématique ou de créer des diagrammes d'état, qui peuvent être utilisés comme entrée au lieu du texte VHDL.

La figure 4.7 montre comment se présente le logiciel «ISE Xilinx Project Navigator».

La saisie du texte se fait sur la partie droite de l'écran, on voit en haut à gauche la hiérarchie du projet, et en bas à gauche les nombreux outils nécessaires tout au long du développement du projet.

Il faut commencer par créer un projet, ensuite inclure des fichiers sources dans lesquels il faut saisir le texte VHDL désiré. On peut inclure autant de sources qu'on veut dans un projet.

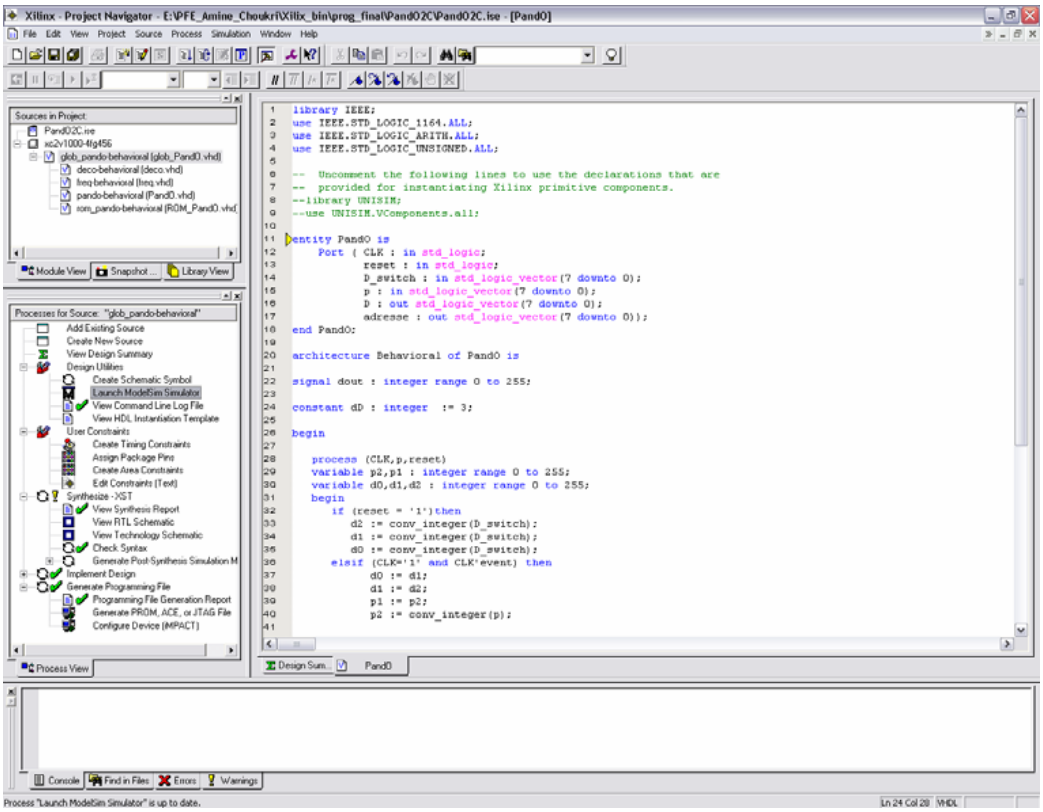


Figure 4.7. Vue d'ensemble du logiciel « Xilinx Project Navigator »

b. Vérification des erreurs

Cette étape est effectuée en appuyant sur le bouton « check syntax ». Elle permet de vérifier les erreurs (errors) de syntaxe du texte VHDL et d'afficher les différentes alarmes « warnings » liées au programme, par exemple des signaux déclarés mais non utilisés dans le programme. S'il y'a des erreurs dans le programme, il ne peut pas être synthétisé, mais la présence d'alarmes n'empêche pas de poursuivre normalement les autres étapes du développement.

Cette étape permet donc de valider la syntaxe du programme et de générer la « netlist », qui est un fichier contenant la description de l'application sous forme d'équations logiques.

c. Synthèse

La synthèse permet de réaliser l'implémentation physique d'un projet. Le synthétiseur a pour rôle de convertir le projet, en fonction du type du circuit FPGA cible utilisé, en portes logiques et bascules de base. L'outil « View RTL Schematic » permet de visualiser les schémas électroniques équivalents générés par le synthétiseur (figure 4.8).

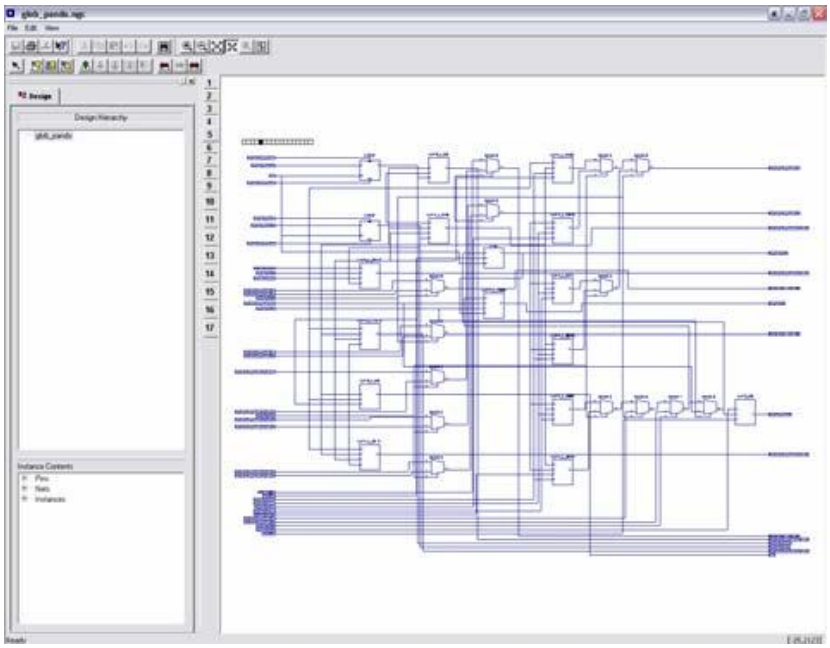


Figure 4.8. Aperçu de l'outil « View RTL Schematic »

De plus, le synthétiseur permet à l'utilisateur d'imposer des contraintes de technologie (User constraints) : par exemple fixer la vitesse de fonctionnement (Create Timing Constraints), délimiter la zone du circuit FPGA dans laquelle le routage doit se faire (Create Area constraints) ou affecter les broches d'entrées/sorties (Assign Package Pins).

La figure 4.9 montre un aperçu de l'outil d'assignation des broches d'entrées/sorties.

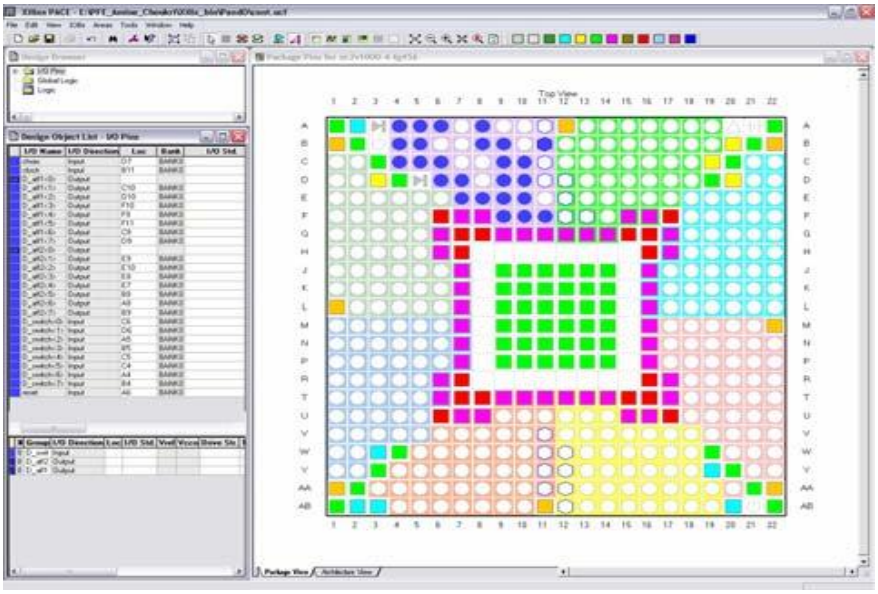


Figure 4.9. Aperçu de l'outil d'affectation des broches d'entrées/sorties

d. Simulation

Le simulateur utilisé est le « ModelSim Simulator » (figure 4.10). La simulation permet de vérifier le comportement d'un design avant ou après implémentation dans le composant cible. Elle représente une étape essentielle qui nous fera gagner du temps lors de la mise au point sur la carte. Il faut juste noter qu'un projet peut être simulé même s'il n'est pas synthétisable.

Lors de l'étape de simulation comportementale, on valide l'application indépendamment de l'architecture et des temps de propagation du futur circuit cible. La phase de simulation après synthèse valide l'application sur l'architecture du circuit cible, et enfin la simulation temporelle prend en compte les temps de propagation des signaux à l'intérieur du circuit FPGA cible.

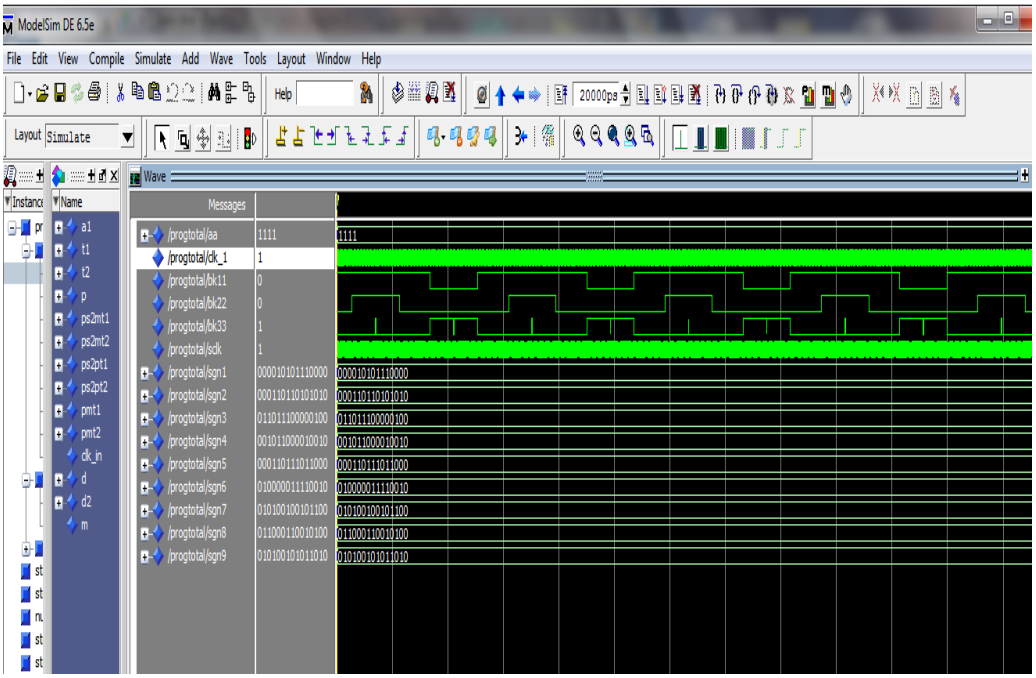


Figure 4.10. Présentation du simulateur « ModelSim Simulator »

2. Implémentation de la commande MLI neuronal sur FPGA

2.1. Description du programme sous VHDL

On développe un programme sous VHDL qui calcule les instants de commutations t_1, t_2 et la période T de la tension de sortie. D'après l'expression mathématique de t_1, t_2 et T (équation 3.34, 3.35 et 3.37 respectivement) ; on doit programmer la division binaire sous VHDL afin de les calculés, on n'utilise pas la division sous-VHDL parce que on ne peut pas implémenter le programme ; donc on fait des rappels théorique sur la division binaire dans ce qui ce suit.

La division est la plus complexe des quatre opérations arithmétiques. Le principe de la division décimale classique que l'on apprend à l'école est basé sur le principe d'essais successifs. Par exemple, pour diviser 11 par 4, on doit d'abord s'apercevoir que 11 est supérieur à 4 et se demander combien de fois 4 va dans 11. Si l'on fait un essai initial avec 3, la multiplication 3x4=12 nous indique que le choix est mauvais (12>11). Il faut recommencer avec 2 etc...

Compte tenu du moins grand nombre de possibilité d'essais, ce principe d'essais successifs et en fait beaucoup plus simple dans la division binaire.

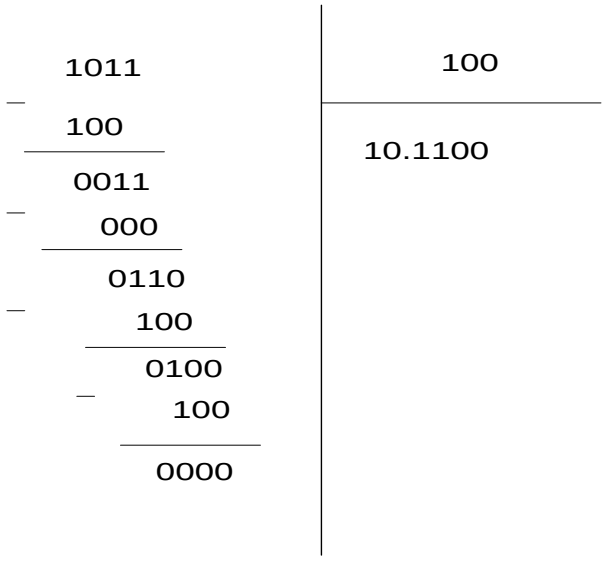


Figure 4.11. Exemple d'une division binaire

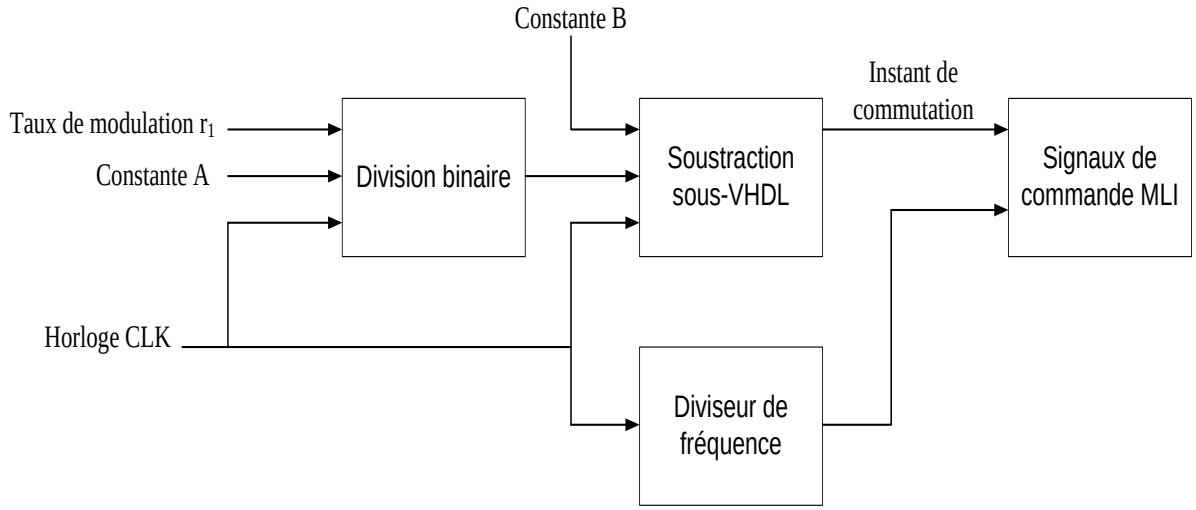


Figure 4.12. Schéma synoptique du programme

La figure 4.12 présente un schéma synoptique du programme développé sous-VHDL, il est composé de plusieurs bloc :

- ✓ Bloc division binaire : il réalise la division de A sur r_1
- ✓ Bloc soustraction sous-VHDL : il réalise la soustraction B-A/r_1
- ✓ Bloc diviseur de fréquence : permet de générer une fréquence d'horloge de 1 MHz à partir d'un signal d'horloge de 50 MHz
- ✓ Bloc signaux de commande : permet de générer les signaux de commande à partir des instants de commutation calculée.

On commence par la déclaration de toutes les bibliothèques utilisées ; puis on déclare les variables d'entrées/sorties ; ensuite on décret l'architecture du programme. Une fois le programme est finalisé ; on va faire la synthèse pour la vérification des erreurs, puis la simulation du programme sous Model Sim pour visualiser les signaux de sortie, enfin l'implémentation du programme sur le circuit FPGA.

2.2. Résultats de simulation sous Model Sim

L'outil de simulation utilisé dans notre mémoire est le simulateur Model Sim pour la visualisation des signaux de sorties; la figure 4.12 représente les signaux de commande de T11, T12 et T13 obtenus par simulation pour r = 0.61et la figure 4.13 représente les signaux de commande de T11, T12 et T13 obtenus par simulation pour r = 0.71 et m=2

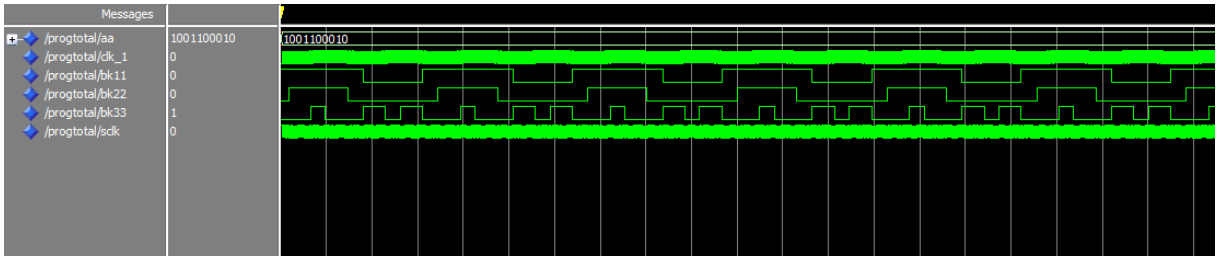


Figure 4.13. Les signaux de commande de T11, T12 et T13 obtenus par simulation pour r = 0.61 et m = 2

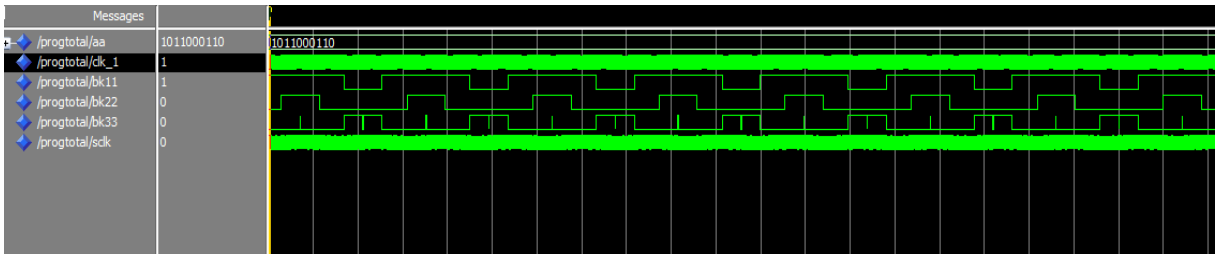


Figure 4.14. Les signaux de commande de T11, T12 et T13 obtenus par simulation pour r = 0.71 et m = 2

2.3. Interprétation

On constate que les signaux de commande de T11, T12 et T13 obtenus par simulation pour r = 0.61 et m = 2 (Figure 4.13) sont identiques à ceux obtenus par Simulink MATLAB (Figure 2.4, Figure 2.5 et Figure 2.6), il en est de même pour r = 0.71et m = 2 (Figure 4.13).

2.4. Résultats expérimentant

2.4.1. Description du ban d'essai

Pour vérifier les résultats de simulation obtenus par le simulateur Model Sim ; on implémente le programme écrit sous VHDL sur la carte de développement FPGA Spartan-3E XC3S500E.

La figure 4.15 présente la plateforme d'essai qui est composée d'un oscilloscope Tektronix MSO2024B afin de visualiser les signaux de sorties, la carte FPGA sur laquelle le programme est chargé et un PC portable pour la programmation sous VHDL. Les Figure 4.17 et Figure 4.18 présentent les signaux commande de T11, T12 et T13 obtenus par l'implémentation pour r = 0.61 et r = 0.71 respectivement et m=2.

Le kit de développement Spartan-3E Starter, qu'on a utilisé pour développer notre application, fournit une solution complète de développement d'applications sur la famille Spartan-3E de Xilinx [13].

Les caractéristiques de la carte FPGA (Figure 4.16) Xilinx Spartan-3E XC3S500E sont citées en annexe.

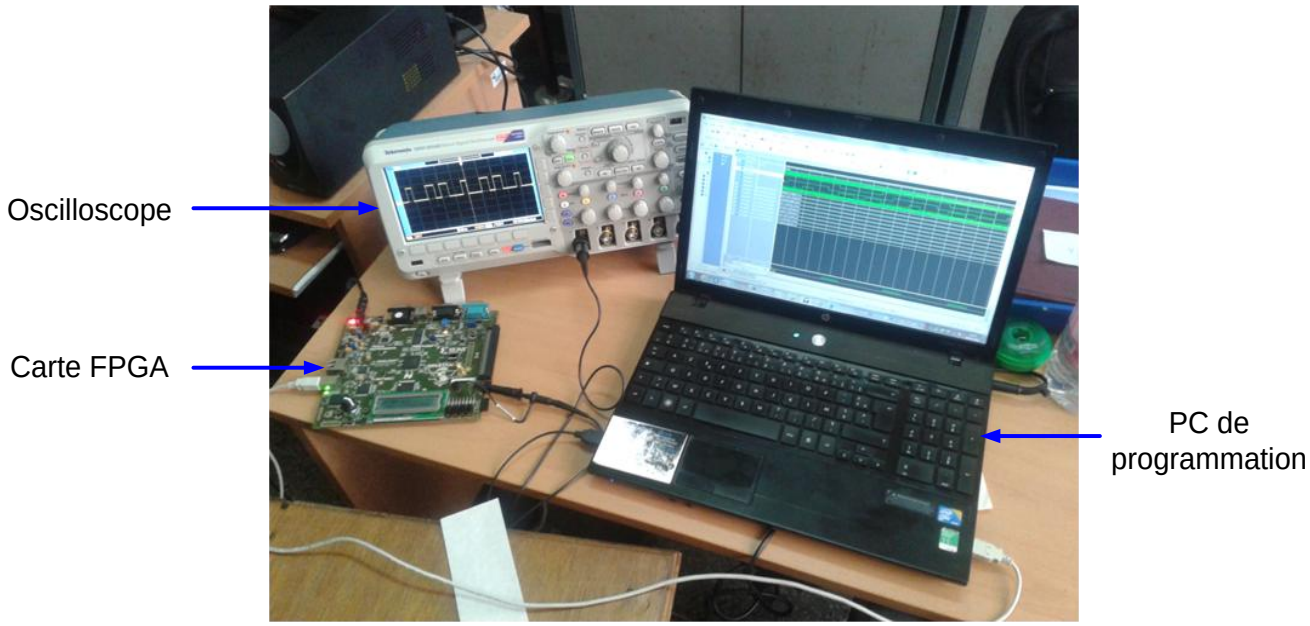


Figure 4.15. Photographie du ban d'essai

Figure 4.16. Photographie de la carte de développement FPGA (Spartan-3E XC3S500E)

2.4.2. Implémentation sur la carte de développement :

Nous avons présenté dans cette partie la procédure d'implémentation d'un programme sur un circuit FPGA ; l'implémentation est faite au niveau de laboratoire de l'équipe économie et maîtrise d'énergie au Centre de Recherche des Energies Renouvelables (CDER) sis à Bouzaréa.

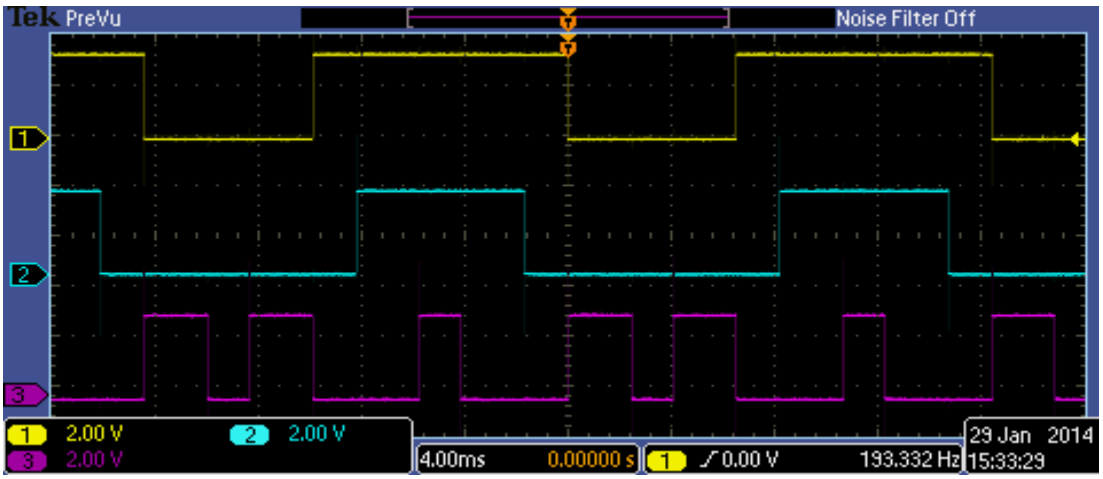


Figure 4.17. Les signaux commande de T11, T12 et T13 obtenus par l'implémentation pour r = 0.61 et m = 2

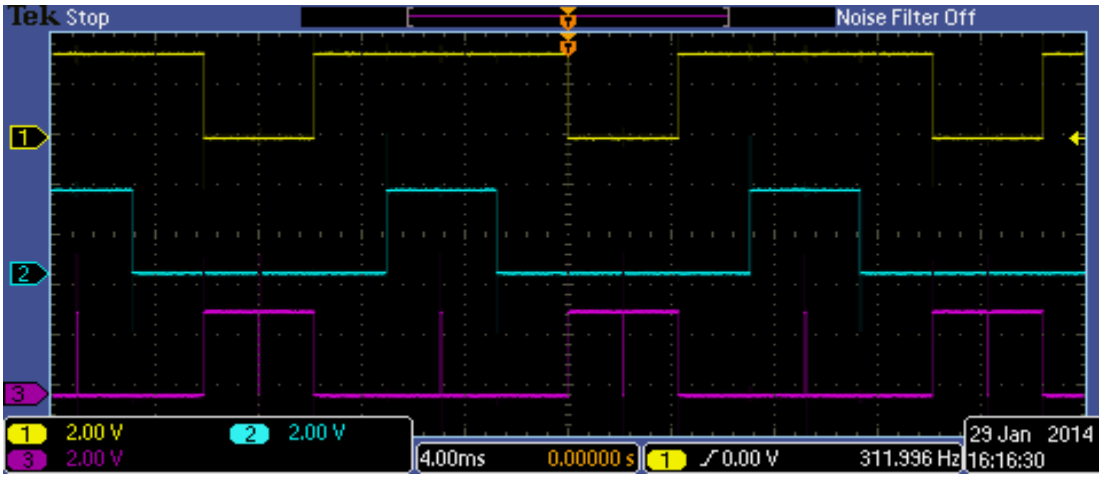


Figure 4.18. Les signaux commande de T11, T12 et T13 obtenus par l'implémentation pour r = 0.71 et m = 2

2.4.3. Interprétation des résultats :

On constate que les signaux de commande de T11, T12 et T13 obtenus par implémentation pour r = 0.61 et m = 2 (Figure 4.17) sont identiques à ceux obtenus par simulation (Figure 4.13), il en est de même pour r = 0.71 et m = 2 (Figures 4.14 et 4.18).

On constate encore une similitude entre les signaux de commande de T11, T12 et T13 obtenus par implémentation pour r = 0.61 et m = 2 (Figure 4.17) et ceux obtenus par Simulink MATLAB (Figure 2.4, Figure 2.5 et Figure 2.6).

L'implémentation du programme sur le circuit FPGA permet de visualiser les signaux de commandes sur l'oscilloscope, ils montrent la superposition entre les signaux de commande réels et ceux obtenus par simulation.

Conclusion générale

Dans ce travail, nous avons étudié une commande MLI (PWM) à élimination sélectives d'harmoniques et asservissement du fondamental en se basant sur la théorie de réseau de neurones dans le but de commander un onduleur triphasé à cinq niveaux à structure NPC multiniveaux déstinée aux véhicules électriques.

On a modélisé la machine asynchrone triphasée alimentée par un onduleur ; ensuite les types des onduleurs ont été présentés; on a cité les différentes stratégies de commande de ces onduleurs ; d'après les avantages et les inconvénients de la commande MLI pré-calculée (PWM), elle appariât la plus adéquate pour varier la vitesse des machines asynchrone.

On a résolue le système d'équation non-linéaire sous logiciel MATLAB à l'aide de la théorie résultante des polynôme symétrique afin de calculer les angles de commutation des interrupteurs de l'onduleur; les résultats de simulations dans le simulateur PSIM montrent l'efficacité de la commande dans l'élimination des harmoniques désirés (l'harmonique n°5 dans le cas étudié). L'inconvénient de cette technique est le temps de calcul des angles de commutations qui la rendre une technique de commande « off-line ».

Les réseaux de neurones artificiels on été exploités pour élaborer une technique « on-line » afin de générer les angles de commutation des interrupteurs en temps réel. Les simulations de cette technique sous Simulink MATLAB montrent la précision de calcul des angles de commutations des interrupteurs et l'efficacité dans l'élimination des harmoniques désirés ; les harmoniques multiples de trois sont éliminées automatiquement dans le cas de la tension entre phase.

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

Comme perspectives, nous pouvons proposer les points suivants :

-Augmenter le nombre de niveaux de l'onduleur pour éliminer plus d'harmoniques.

-Appliquer la même technique à d'autres types d'onduleur et faire une comparaison.

-Détermination des angles de commutation dans tout l'intervalle de variation de taux de modulation.

-Utiliser d'autres techniques permettant le calcul des angles en temps réel.

L’utilisation de la même technique dans des applications industrielles nécessitant la variation de vitesse des moteurs asynchrones.

Bibliographie:

[1] : F. BRIHMAT « L'étude conceptuelle d'un système de conditionnement de puissance pour une central hybride PV/Eolienne » Mémoire de magister UMMTO Tizi-ouzou Juillet 2012.

[2] : A. Djerdir, K. Elkadri et A. Miraoui « Alimentation par biberonnage solaire photovoltaïque d'une chaîne de motorisation électrique » Revue des Energies Renouvelables Vol. 9 N°2 63 – 74 Juin 2006.

[3] : Les voitures électriques le kart bimoteur de l'iut de Troyes, Synthèse par Denis Hoang du travail collectif de l'équipe « Kart jaune ».Livre délivrée par l'Association E-KART-TROYE S, d départements GEII et GMP de l'IUT de Troyes, 2011.

[4] :A. Djamila « Commande d'une machine asynchrone sans capteur mécanique, à l'aide de régulateurs fractionnaires » Mémoire de magister UMMTO Tizi-ouzou 2011.

[5] : Z.Ait wali « Application des FPGA à la commande d'un moteur asynchrone » Mémoire de magister UMMTO Tizi-ouzou.

[6] B.HOUSSEINI « Prototypage rapide à base d'FPGA d'un algorithme de contrôle avancé pour le moteur a induction » mémoire présenté à l'université du Québec à Trois-Rivières décembre 2010.

[7] : A. TALHA « Etude de différentes cascades de l'onduleur à sept niveaux à structure NPC. Application a la conduite d'une machine synchrone à aimants permanents » Thèse de doctorat Décembre 2004.

[8] : K.Imarazene, H.Chekireb, E.M.Berkouk « Application de la théorie résultante a la commande par élimination d'harmoniques d'un onduleurs sept niveaux » International Conference On Industrial Engineering and Manufacturing ICIEM'10, May, 9-10, 2010, Batna, Algeria.

[9] E. Olasagasti « application des réseaux de neurones a l'identification d'un axe de machine-outil » THESE pour obtenir le grade de DOCTEUR DE L'INPG Spécialité: « Génie électrique» Préparée au Laboratoire d'Electrotechnique de Grenoble dans le cadre de l'Ecole Doctorale « Electronique, Electrotechnique, Automatique, Télécommunication, Signal ». Novembre 2002

[10] C. TOUZET «Les réseaux de neurones artificiels introduction au connexionnisme cours, exercices et travaux pratiques ». Livre publié Juillet 1992.

[11] A .Guellal « Implémentation sur FPGA d'une commande MLI on-line basée sur le principe des réseaux de neurones » Mémoire de magister ENP d'Alger 2009.

[12] B. Ashok, A.Rajendran Selective harmonic elimination of multilevel inverter using SHEPWM technique, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May May 2013.

[13] N. Toufik « Implémentation d'une instrumentation sur un circuit FPGA » Mémoire de magister UMMTO Tizi-ouzou. Novembre 2011

[14] Guide d'utilisation de la carte Spartan-3E FPGA Starter Juin 2011.

1. Les angles exacts calculés par le programme MATLAB

r	α_1	α_2
0.010000000000000	1.248378802570974	1.876697333288932
0.020000000000000	1.240119980435132	1.868438511153091
0.030000000000000	1.231860031411629	1.860178562129587
0.040000000000000	1.223598391191380	1.851916921909338
0.050000000000000	1.215334494425341	1.843653025143299
0.060000000000000	1.207067774375094	1.835386305093053
0.070000000000000	1.198797662561049	1.827116193279008
0.080000000000000	1.190523588407655	1.818842119125614
0.090000000000000	1.182244978885006	1.810563509602965
0.100000000000000	1.173961258146196	1.802279788864154
0.110000000000000	1.165671847159789	1.793990377877748
0.120000000000000	1.157376163336737	1.785694694054695
0.130000000000000	1.149073620151054	1.777392150869013
0.140000000000000	1.140763626753559	1.769082157471518
0.150000000000000	1.132445587577943	1.760764118295902
0.160000000000000	1.124118901938413	1.752437432656372
0.170000000000000	1.115782963618126	1.744101494336085
0.180000000000000	1.107437160447595	1.735755691165554
0.190000000000000	1.099080873872210	1.727399404590169
0.200000000000000	1.090713478507977	1.719032009225935
0.210000000000000	1.082334341684547	1.710652872402506
0.220000000000000	1.073942822974538	1.702261353692497
0.230000000000000	1.065538273708113	1.693856804426071
0.240000000000000	1.057120036471726	1.685438567189684
0.250000000000000	1.048687444589874	1.677005975307833
0.260000000000000	1.040239821588634	1.668558352306593
0.270000000000000	1.031776480639683	1.660095011357642
0.280000000000000	1.023296723983432	1.651615254701391
0.290000000000000	1.014799842329805	1.643118373047763
0.300000000000000	1.006285114235110	1.634603644953069

0.310000000000000	0.997751805453341	1.626070336171300
0.320000000000000	0.989199168260139	1.617517698978098
0.330000000000000	0.980626440747519	1.608944971465478
0.340000000000000	0.972032846087346	1.600351376805305
0.350000000000000	0.963417591761383	1.591736122479342
0.360000000000000	0.954779868755590	1.583098399473549
0.370000000000000	0.946118850716179	1.574437381434137
0.380000000000000	0.937433693064738	1.565752223782697
0.390000000000000	0.928723532069545	1.557042062787504
0.400000000000000	0.919987483869951	1.548306014587910
0.410000000000000	0.911224643450481	1.539543174168440
0.420000000000000	0.902434083561043	1.530752614279002
0.430000000000000	0.893614853579306	1.521933384297265
0.440000000000000	0.884765978311032	1.513084509028991
0.450000000000000	0.875886456723756	1.504204987441715
0.460000000000000	0.866975260608844	1.495293791326802
0.470000000000000	0.858031333166511	1.486349863884469
0.480000000000000	0.849053587507931	1.477372118225890
0.490000000000000	0.840040905068025	1.468359435785984
0.500000000000000	0.830992133921954	1.459310664639913
0.510000000000000	0.821906086997711	1.450224617715670
0.520000000000000	0.812781540176482	1.441100070894440
0.530000000000000	0.803617230271676	1.431935760989635
0.540000000000000	0.794411852876658	1.422730383594617
0.550000000000000	0.785164060070235	1.413482590788193
0.560000000000000	0.775872457967884	1.404190988685842
0.570000000000000	0.766535604105501	1.394854134823460
0.580000000000000	0.757152004641099	1.385470535359057
0.590000000000000	0.747720111358382	1.376038642076341
0.600000000000000	0.738238318454448	1.366556849172406
0.610000000000000	0.728704959091928	1.357023489809887
0.620000000000000	0.719118301693816	1.347436832411774

0.630000000000000	0.709476545956720	1.337795076674678
0.640000000000000	0.699777818555673	1.328096349273630
0.650000000000000	0.690020168510440	1.318338699228402
0.660000000000000	0.680201562179864	1.308520092897822
0.670000000000000	0.670319877846696	1.298638408564659
0.680000000000000	0.660372899851022	1.288691430568967
0.690000000000000	0.650358312224892	1.278676842942858
0.700000000000000	0.640273691775369	1.268592222493346
0.710000000000000	0.630116500555755	1.258435031273607
0.720000000000000	0.666148565109377	1.218807027044499
0.730000000000000	0.720325027917312	1.164630564236565
0.740000000000000	0.792586939304900	1.092368652848979
0.830000000000000	0.860734739001424	0.860734739001424
0.930000000000000	0.751859215044166	0.751859215044166

Table des angles de commutation en fonction du taux de modulation.

2. Caractéristiques de la carte FPGA Xilinx Spartan-3E XC3S500E

• configuration parallèle Flash NOR

• configuration de Flash série SPI

• le développement intégré

• MicroBlaze ™ processeur RISC embarqués 32 bits

• Contrôleur PicoBlaze ™ 8 bits intégré

• Interfaces de mémoire DDR

• Jusqu'à 232 broches d'E/S

• Forfait 320 broches FBGA

• Plus de 10 000 cellules logiques

• Xilinx 4 Mbit Flash Platform configuration PROM

• Xilinx 64 macrocellulaire XC2C64A CoolRunner ™ CPLD

• 64 Mo (512 Mbit) de DDR SDRAM, interface de données de x16, 100 MHz

• 16 Mo (128 Mbit) de parallèle Flash NOR (Intel Strataflash)

• Code MicroBlaze observation

• 2 lignes, écran LCD 16 caractères

• souris PS / 2 ou port clavier

• Port d'affichage TVGA

• 10/100 PHY Ethernet (nécessite Ethernet MAC dans FPGA)

• Deux ports RS-232 à 9 broches (DTE-DCE et de style)

• FPGA / CPLD télécharger / Interface de débogage de bord basé sur l'USB

• 50 MHz oscillateur d'horloge

• SHA-1 1-wire EEPROM série pour bitstream protection contre la copie

• connecteur d'extension Hirose FX2

• Trois Digilent connecteurs d'extension à 6 broches

• Quatre-production, basé SPI numérique-analogique (DAC)

• Le port de débogage ChipScope ™ SoftTouch

• Rotary-codeur avec arbre à bouton-poussoir

• Huit LED discrètes

• Quatre interrupteurs à glissière

• Quatre boutons-poussoirs

• Connecteur SMA entrée d'horloge

• Prise DIP 8 broches pour auxiliaire oscillateur d'horloge

Résumé :

La commande MLI (PWM) calculée avec élimination sélective d'harmonique et asservissement de la fondamentale joue un rôle important dans le domaine industriel, en particulier dans la régulation des procédés industriels, par contre l'exploitation de cette technique souffre de plusieurs problèmes ; en premier la commande en temps réel ; l'objectif est d'implémenter sur un circuit FPGA une commande MLI 'on-line ' appliquée sur un onduleur triphasé cinq niveaux à structure NPC multi-niveaux destinée aux véhicules électrique.

Mot clés : FPGA, VHDL, MLI, Onduleur triphasé cinq niveaux, topologie NPC, Théorie résultante et polynôme symétrique, véhicule électrique.