

3/03

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département ELECTRONIQUE



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES
En vue de l'obtention du diplôme

D'INGENIEUR D'ETAT EN ELECTRONIQUE

Thème

**Implémentation sur PIC16F876 d'un
algorithme de génération du signal MLI**

Proposé par :
Mr.C.LARBES
Mr.M. HADDADI

Etudié par :
Mr.S.GHERBI

Dirigé par :
Mr.C.LARBES
Mr. M.HADDADI

PROMOTION JUIN 2003
ENP, 10 Avenue Hassen Badi EL-Harrach

إن تقنية تغيير عرض الدفع المبرمجة لبائل و هوفت ذات ضبط المركب الأصلي للموجة و حذف مركباتها التوافقية تمثل حلا مغريا في تطبيق التحكم في سرعة المحرك اللامتزامن. لكن لن تلقى هذه التقنية انتشارا واسعا بسبب استحالة حساب زوايا التبديل في حالة اشتغال المحرك. و لتفادي هذه السلبيات. عرضت أعمال كثيرة و طرق متعددة لحساب زوايا التبديل لحضيا و بقل حساب ممكن. قمنا في هذا العمل بعرض احدى الطرق المقدمه، حيث تسمح لنا في التحكم في المركب الأصلي مع حذف مركباته التوافقية و ذلك حسب رغبتنا نحن، كما لها - هذه التقنية- نفس الدقة مع تقنية باتل و هوفت. إن استعمال هذه التقنية تسمح في التحكم في سرعة المحرك في كل مجال تغييرها، هذا يؤدي إلى اقتصاد في الطاقة المستهلكة من طرف المحرك و انخفاض حرارته مع تقادي التغيرات العشوائية للعزم. قمنا بمحاكاة هذه الطريقة على مستوى الحاسوب و تأكدنا من صحة نتائج هذه الطريقة. ثم قمنا ببرمجة هذه الطريقة على ميكروكونترولر (16F876)، و قد تحصلنا على نتائج عملية تعدنا بأفق مشرق، حيث تحصلنا على إشارة (MLI)، تسمح لنا التحكم في إحدى أذرع المومج الثلاثي الذي يعتبر مغذي الطاقة للمحرك.

المفاتيح: تغيير عرض الدفع، المركب الأصلي للموجة، زوايا التبديل، المركبات التوافقية، الميكروكونترولر.

Résumé

La technique de modulation de largeur d'impulsion (MLI) calculée de Patel et Hoft, avec asservissement du fondamental et élimination des harmoniques sélective, considéré comme une alternative très attirante pour la commande en vitesse du moteur asynchrone triphasé. Mais son utilisation est limitée par le fait que les angles de commutation ne peuvent être calculé en fonctionnement dans des applications temps réel.

Pour palier cette inconvénient, plusieurs algorithmes on était proposé pour arriver a une application pratique de cette méthode. Dans ce travail en présente un algorithmme qui nous permet de diminuer le calcule de cette méthode, ainsi la capacité mémoire.

Cette algorithmme présente une précision égal a celle de la technique de Patel et Hoft permettant ainsi une élimination totale des premier harmoniques est un asservissement du fondamental.

La simulation de ce nouvel algorithmme avec MATLAB nous a donnée des résultants très précis.

Enfin, en a implémenter cette algorithmme sur un système a base d'un microcontrôleur 16F876, et en a obtenu des résultats très encourageant qui nous en permis de commander un seul bras d'un onduleur triphasé.

Mot clés : onduleur triphasé, modulation de largeur d'impulsion, angles de commutation, microcontrôleur, angles de commutation.

Abstract

The calculated pulse width modulation (PWM) technique of harmonic elimination and voltage control, proposed by Patel and Hoft, is an attractive alternative for speed control of an induction motor. However, its application has been limited by the fact that the switching angles cannot be calculated on line in real-time applications.

To overcome this problem, a lot off algorithms has been proposed, in this work, we have studied on of these algorithms. This latter gives the same accuracy as Patel and Hoft's algorithm leading to the elimination of the low order harmonics and to the control of the fundamental voltage.

The simulation of the algorithm with MATLAB has given good results as like as the Patel and Hoft's algorithm.

We have applied this algorithm in practice with microcontroller PIC16F876, and we have obtained good results for leading only one leg of the triphase inverter.

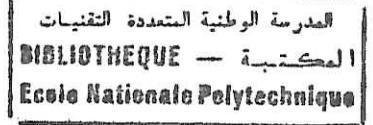
Key words: pulse width modulation, harmonic elimination, switching angles, microcontroller, triphase inverter.

Dédicace

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédie ce travail à ma mère et à mon père, pour tous ceux qui ont sacrifié pour moi, à toute ma famille, et à tous mes amis.

Remerciements



Ce travail a été fait au laboratoire de l'énergie solaire du département d'électronique de l'école nationale polytechnique d'Alger, sous la direction de **M.C.LARBES** ET **M.M.HADDADI**.

Je remercie **Dieu** pour tous ce qui m'a donné.

Je tiens à adresser mes sincères remerciements à mon promoteur **M.C.LARBES** et à mon co-promoteur **M.M.HADDADI** qui ont suivi mon travail durant tout un semestre pour leur aide et leur soutien matériel précieux.

Je remercie également, **M..M.S.AIT-CHIEKH** qui a voulu bien présider le jury de soutenance de ma thèse.

Nous n'oublions pas de remercier les autres membres de jury **E.M.BERKOUK** pour leur évaluation de mon travail.

Ainsi que tous les autres enseignants qui m'ont aidé pendant tout mon travail.

Je voudrais exprimer mon profond respect à tous les enseignants qui m'ont encadré durant ma formation d'ingénieur à l'école nationale polytechnique.

Enfin je remercie tout le personnel de la bibliothèque pour leur aide et leur patience.

Je n'oublie pas de remercier aussi tout le personnel administratif de l'école nationale polytechnique.

SOMMAIRE

Introduction.....1

Chapitre 1

Le moteur asynchrone.....4

 1. Introduction..... 4

 2. Moteur asynchrone.....4

 2.1 Le stator.....4

 2.2 Le rotor.....5

 2.3 principe de fonctionnement.....5

 2.4 point de fonctionnement.....6

 2.4.1 Définition.....6

 2.4.2 méthode de calcul du point de fonctionnement.....6

 2.4.3 remarque sur la stabilité.....7

 3. variation de vitesse d'un moteur asynchrone.....7

 4. démarrage du moteur asynchrone.....8

 5. inversion du sens de rotation du moteur.....8

 6. le rapport V/f constant.....8

 7. avantages et inconvénients.....10

Chapitre 2

La conversion continu – alternatif.....11

 1. introduction.....11

 2. principe de l'onduleur de tension.....12

 2.1 Onduleur en demi – pont.....12

 2.2 Onduleur monophasé en pont.....13

 2.3 Onduleur triphasé.....14

 3. Réglage de tension par modulation de largeur d'impulsion.....15

 3.1 Technique triangulo – sinusoïdale.....15

 3.2 Les techniques à échantillonnage19

 3.3 Méthode de réglage par élimination d'harmoniques.....21

Chapitre 3

Technique MLI programmée.....	23
1. Description de la méthode de Patel et Hoft	23
1.1 remarque sur les angles de commutations.....	26
1.2 Onduleur demi-pont à transistors.....	27
2. Calcule des valeurs exactes des angles de commutation par la méthode de Newton-Raphson	27
2.1 Estimation initiale de la solution.....	29
2.2 Résolution du système non linéaire par la méthode de Newton-Raphson	30
2.3 Résolution du système linéaire par l'algorithme de Gauss	34
3. Conclusion.....	35

Chapitre 4

Nouvel algorithme 'on-line' pour le calcul des angles de commutations.....	36
1. Approximation des angles exacts	36
1.1 Cas k impair.....	37
1.2 Cas k pair.....	43
2. Précision de l'algorithme.....	44
2.1 Angles calculés.....	44
2.2 Calcul des harmoniques	45
3. Simulation du nouvel algorithme 'on-line'	45
3.1 Introduction.....	45
3.2 Générateur d'onde MLI programmée.....	46
3.3 Calcul des instants de commutation sur une période (déphasage 0°).....	46
3.4 Calcule des instants de commutation T (i) sur une période.....	47
3.5 Calcul de la valeur S (i) de la tension MLI à l'instant T (i).....	48
4. Conclusion.....	52

Chapitre 5

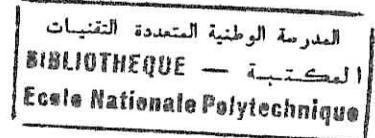
Application à un système a base d'un PIC16F876	53
1. Le PIC 16F876.....	53
1.1 Qu'est-ce qu'un PIC	53
1.2 Les différentes familles des PICs.....	53
1.3 Le PIC16F876	54

1.3.1 La mémoire programme.....	54
1.3.2 La RAM.....	55
2. Réalisation de la carte pour la génération du signal MLI.....	57
3. Le programme de génération du signal MLI.....	58
4. Simulation du programme.....	62
5. Les résultats pratiques.....	63
6. Conclusion	66
Conclusion générale.....	67
Bibliographie.....	68

Annexe

Annexe A.....	69
Annexe B	72
Annexe C.....	75
Annexe D.....	76

Introduction générale



Le travail a toujours fatigué l'homme, celui-ci n'a eu de cesse et de quête plus persévérante, depuis l'aube des temps, que de trouver des moyens de ce faire remplacer pour tous les pénibles travaux nécessaires à sa vie, à son épanouissement et à sa soif de conquête de l'univers qu'il habite.

Il a d'abord trouvé les autres hommes, les esclaves et les animaux de tout les sortes qu'il à su apprivoiser. Il a inventé les outils qui utilisaient encore sa force motrice, puis les machines de plus en plus puissantes, de plus en plus rapides, de plus en plus intelligentes en les dotant de mémoire et de capacité d'apprentissage, voire d'autonomie. Les machines ont été d'abord purement mécaniques, à base de leviers et de roues, puis hydrauliques et thermiques et enfin électriques.

Depuis 1970 on assiste à une explosion des applications des servomécanismes électriques (machines, outils de commande numérique, robots, manipulateurs, vérins, etc.) dans tous les domaines, y compris l'aéronautique, très exigeante, et le domaine militaire. Cette percée est due à l'apparition de nouveaux actionneurs caractérisés principalement par :

- Des constantes de temps électriques et mécaniques beaucoup plus petites que celles des machines classiques par utilisation de structures nouvelles (entrefer axiaux, conducteurs libres, rotors cloches, etc.) ;
- Des puissances massiques accrues d'un facteur 10 à 100 ;
- Une utilisation croissante d'aimants permanents puissants, comme source d'excitation sans pertes (ferrites puis terres rares) ;
- Des alimentations statiques à tension et fréquence variables avec des temps de réponse extrêmement brefs ;
- Des méthodes de commande nouvelles et numériques grâce aux microprocesseurs et cultures divers qui apportent la versatilité voulue.

Un actionneur électromécanique est un transducteur qui assure une conversion d'énergie et d'information de la forme électrique à la forme mécanique.

Dans le domaine de l'industrie et du transport, on cherche de plus en plus la vitesse variable comme solution aux exigences d'économie d'énergie, de souplesse d'exploitation, d'automatisation ou de contrôle du processus lui-même.

Par exemple, en matière de contrôle de processus, la vitesse variable permet d'adapter la vitesse du moteur à la demande du processus. Dans ce cas la puissance prélevée au réseau industriel est égale à la puissance consommée par le processus, aux pertes près. Il en résulte une économie d'énergie.

La vitesse variable est assurée généralement par le moteur à courant continu. Mais celui-ci présente plusieurs inconvénients : coût élevé, présence d'un collecteur, entretien fréquent. Pour éviter ces derniers, on utilise de plus en plus, dans la gamme des faibles et moyennes puissances, le moteur asynchrone triphasé d'induction à cage qui présente des avantages intéressants (coût réduit, robustesse, simplicité, entretien réduit, encombrement faible). Alimenté à partir d'un onduleur de tension autonome (variable en tension et en fréquence, avec le rapport V/f constant) à modulation de largeur d'impulsion (MLI), il offre la possibilité de fonctionnement dans une large gamme de vitesses.

Mais ce fonctionnement est accompagné de pulsation de couple et d'échauffement du moteur dus à la présence d'harmoniques dans le signal de sortie de l'onduleur de tension MLI. Ces pulsations de couple deviennent gênantes aux faibles vitesses.

En résumé, le moteur d'induction à cage, alimenté par un onduleur autonome à transistors (variable en tension et en fréquence, avec le rapport V/f constant), selon une technique MLI éliminant correctement les harmoniques, présente des performances meilleures que celles du moteur à courant continu.

Au cours de cette étude, on donnera d'abord une brève description des caractéristiques du moteur d'induction à cage.

Ensuite, après avoir rappelé succinctement les différentes techniques de génération du signal MLI, on va détailler la technique MLI programmée, avec élimination harmonique sélective et asservissement du fondamental (méthode de Patel et Hof).

On va présenter un algorithme qui approximer les valeurs 'exactes' des angles de commutation à l'aide d'un algorithme basé sur la méthode des moindres carrés, l'erreur d'approximation de l'algorithme est acceptable.

Cet algorithme nous permet de réduire le calcul des angles considérablement et permet une implémentation de la méthode d'élimination des harmoniques sur un système à base d'un microprocesseur.

En dernier on va faire une application sur un système à base d'un microcontrôleur 16F876.

Chapitre 1

Le moteur asynchrone

1. Introduction

Le moteur électrique est une machine électrique qui transforme l'énergie électrique en énergie mécanique.

Il existe différents types de moteurs électriques :

- Moteur à courant continu : il est utilisé en vitesse variable dans une large gamme de puissances.
- Moteur synchrone : il est utilisé dans la gamme grandes puissances.
- Moteur asynchrone : il est utilisé en moyennes et faibles puissances.

Donc le domaine d'application nous détermine le choix du type du moteur.

2. Moteur asynchrone

Un moteur asynchrone est une machine à $2p$ pôles, alimentée à partir d'un réseau alternatif de fréquence f et qui ne tourne pas exactement à la vitesse synchrone N définie par :

$$N = f / p = \omega / 2\pi p \quad (1.1)$$

Le moteur asynchrone est formé d'un stator, relié à la source triphasée, et d'un rotor constitué d'un enroulement polyphasé en court circuit.

On distingue deux types de moteurs asynchrones :

- Moteur à collecteur : l'armature tournante est reliée au réseau par un collecteur.
- Moteur d'induction : une des armatures, le rotor d'ordinaire, n'est pas alimentée ; les courants qui y circulent sont induits par l'autre armature, le stator.

2.1 Le stator

Comme pour la machine synchrone, le stator d'un MAS est constitué d'un bobinage triphasé à p pair de pôles.

L'ensemble de ce bobinage est formé à l'aide de 3 enroulements que l'on relie au réseau (triphase de fréquence f) par un couplage étoile ou triangle.

2.2 Le rotor

a- Rotor en cage d'écureuil (la grande majorité)

Constitué de barres métalliques parallèles reliées par 2 couronnes (résistance faible).

b- Rotor bobiné

Les barres métalliques forment un enroulement triphasé. Les extrémités de ces enroulements sont soudées à 3 bagues. Des balais frottant sur ces bagues permettent d'accéder au rotor.

Dans ce travail, on s'intéressera au moteur asynchrone d'induction à cage.

2.3 Principe de fonctionnement

Les 3 enroulements statoriques alimentés par un réseau triphasé équilibré créent dans l'entrefer un champ magnétique tournant à la fréquence de rotation de synchronisme n_s . Les conducteurs du rotor sont soumis à ce champ tournant. Ils sont alors traversés par des courants de Foucault induits. D'après la loi de Lenz (" les courants induits s'opposent par leurs effets à la cause qui leur donnent naissance "). Les enroulements du rotor étant en court-circuit, la circulation des courants est alors possible. Les forces de Laplace qui en résultent exercent des moments sur le rotor. Le rotor tourne alors à la fréquence de rotation n . De par son principe la fréquence de rotation du rotor est inférieure à la fréquence de synchronisme n_s : $n < n_s$

Le stator crée un flux tournant à la vitesse angulaire synchrone :

$$\Omega = \omega / p. \quad (1.2)$$

Ce flux tournant balaye les enroulements du rotor et y induit des courants. L'interaction entre le flux statorique et les courants induits rotoriques crée le couple de rotation du rotor. Si le rotor tournait à la même vitesse que le flux tournant, le flux à travers les enroulements du rotor ne varierait plus, il n'y aurait plus de courants induits dans le rotor, donc il n'y aurait pas de couple.

Le rotor tourne à une vitesse Ω' plus petite que Ω . L'écart entre Ω' et Ω augmente lorsque le couple résistant sur l'arbre du rotor augmente.

Le glissement

On appelle glissement l'écart des vitesses angulaires synchrones Ω et réelle Ω' rapporté à la vitesse synchrone Ω :

$$g = \frac{\Omega - \Omega'}{\Omega} = \frac{(\omega - \omega')}{\omega} = \frac{(N - N')}{N} \quad (1.3)$$

$$\text{Avec : } N = \Omega / 2\pi \quad \text{et } N' = \Omega' / 2\pi \quad (\text{t/s}) \quad (1.4)$$

On l'exprime en %, ordre de grandeur 3 à 5 % (pour le nominal).

2.4 Point de fonctionnement

2.4.1 Définition

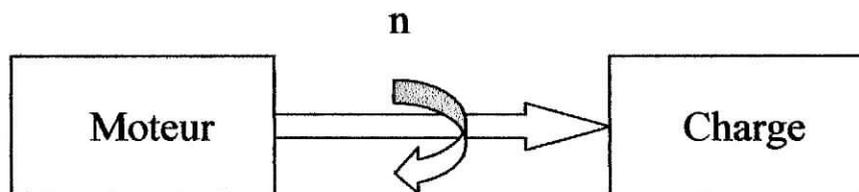


figure1.1 liaison moteur-charge

La charge impose un couple résistant de moment T_r sur l'arbre qui tourne à la vitesse Ω . Le moteur doit fournir un couple utile de moment T_u , à l'équilibre $T_u = T_r$. Le point de fonctionnement du groupe moteur-charge se situe à l'intersection des caractéristiques mécaniques $T_u = f(\Omega)$ et $T_r = f(\Omega)$ du moteur et la charge.

2.4.2 Méthodes de calcul du point de fonctionnement

Il en existe deux méthodes :

1. Si l'on dispose des deux caractéristiques mécaniques, on les trace sur le même repère et on lit les coordonnées de leur point d'intersection.

2. Si l'on connaît les relations mathématiques $T_u = f(\Omega)$ et $T_r = f(\Omega)$, on résout l'équation $T_u = T_r$ (s'il existe plusieurs solutions, on ne conserve que celle qui à un sens physique).

2.4.3 Remarque sur la stabilité

Définition :

Un système est stable s'il revient à sa position d'origine après une perturbation.

Dans le cas suivant:

Il y a deux points de fonctionnement :

- le point A est instable
- le point B est stable

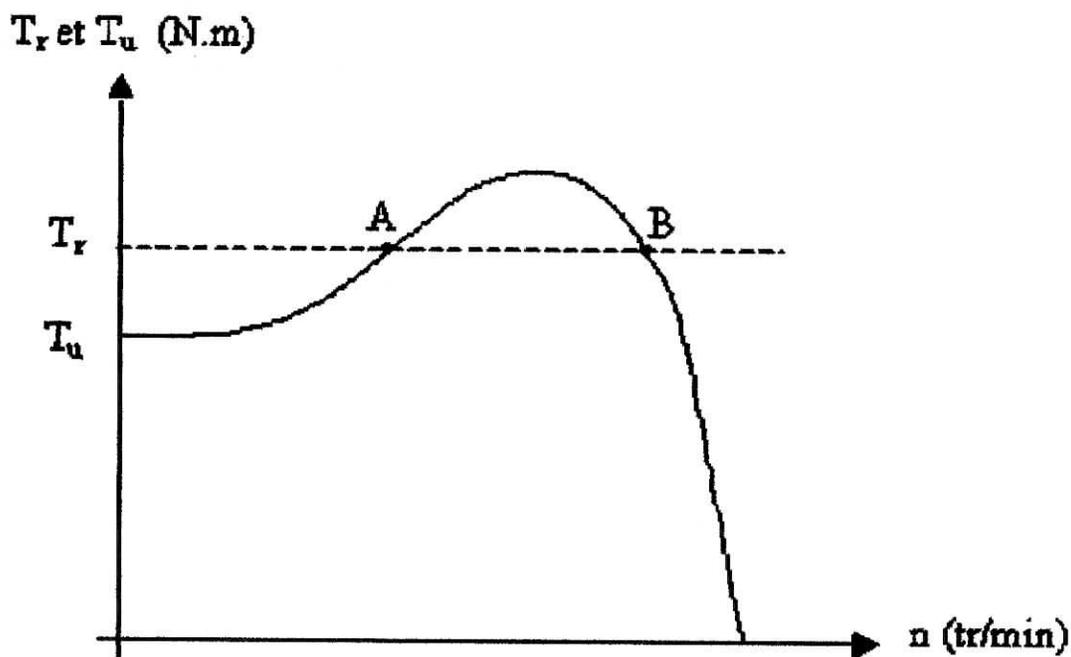


figure1.2 couples T_u et T_r en fonction de la vitesse du moteur

3. Variation de vitesse d'un moteur asynchrone

La technique la plus utilisée pour faire varier la vitesse d'un moteur asynchrone est d'alimenter ce dernier à l'aide d'un onduleur autonome.

On étudiera ceci dans le chapitre sur l'onduleur.

4. Démarrage du moteur asynchrone

Au démarrage du moteur asynchrone on constate que :

Le couple de démarrage est 2 à 3 fois supérieur au couple nominal.

Conséquences : - manque de "confort" mécanique (démarrage brutal).

- courant de démarrage 3 à 5 fois supérieur au courant nominal

Il faut donc un système de protection électrique adapté.

5. Inversion du sens de la rotation du moteur

Le sens de rotation du moteur asynchrone triphasé est varié en inversant le sens de rotation du flux tournant statorique. Il suffit pour cela de permuter les entrées de deux phases. L'inversion du sens du flux tournant pendant la rotation du moteur entraîne d'abord le freinage puis l'arrêt et enfin la rotation en sens inverse du moteur.

6. Le rapport V/f constant

On se propose de fournir un couple maximal et constant dans toute la gamme de fonctionnement du moteur, des faibles vitesses à la vitesse nominale.

Considérons la relation suivante :

$$T_{\max} = \frac{3pV^2}{4\pi f (R_s + \sqrt{R_s^2 + [2\pi f(L_s + L_{re})]^2})^2} \quad (1.5)$$

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

Sachant que, dans l'équation (1.5), les paramètres R_s , L_s , p et L_{re} sont constants, alors le couple T_{\max} est fonction seulement des variations de V et f .

En technique MLI, la tension d'alimentation et la fréquence doivent varier simultanément. Pour atteindre le but proposé ci-dessus, le rapport V/f doit varier en fonction de la fréquence, de la façon suivante :

- Pour des fréquences en dessous de la fréquence minimale f_r , le rapport V/f doit être maintenu constant pour que le flux totalisé reste approximativement constant. Dans ce

cas on obtient un couple de sortie maximum T_m constant (figure 1.3). On peut le voir à partir de l'équation (1.5) en supposant :

$$R_s \ll 2\pi f(L_s + L_{re}) \quad (1.6)$$

La tension d'entrée moteur V devrait être à la valeur maximale V_r quand f égal à la fréquence nominale f_r .

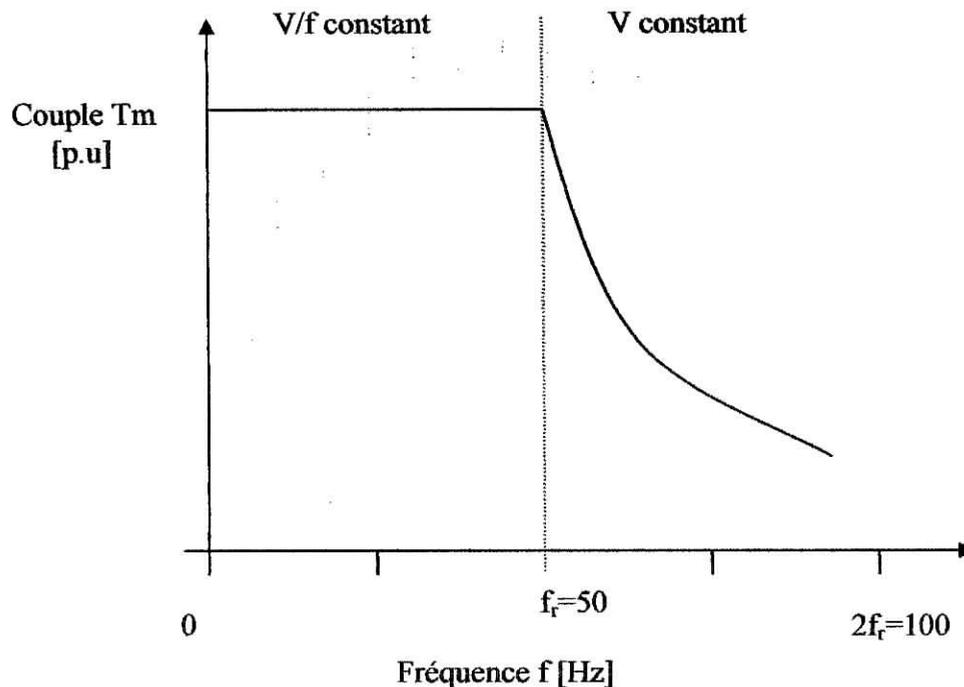


Figure 1.3 Le couple moteur T_m en fonction de la fréquence (Avec le rapport V/f constant puis avec V constant et égal à V_r)

- Pour des valeurs faibles de f , l'équation (1.6) n'est plus valable car R_s prend des valeurs relativement importantes. On doit donc diminuer V moins vite que f .
- Pour des fréquences supérieures à la fréquence nominale f_r , la tension V est maintenue constante à la valeur nominale V_r . Dans ce cas, le couple est proportionnel à $1/f^2$ et la puissance est proportionnelle à $1/f$ (figure).

En fait dans notre étude, on s'intéresse surtout à la région située en dessous de la fréquence nominale f_r .

7. Avantages et inconvénients

Avantage	Inconvénients
<ul style="list-style-type: none">- faible coût d'achat- faible coût d'entretien	<ul style="list-style-type: none">- variation de vitesse (nécessité d'un variateur de vitesse)- la vitesse dépend de la charge.

Chapitre 2

La conversion continue - alternative

1. Introduction

La conversion continue – alternative a pour rôle la transformation d'un signal de tension continu en un signal de tension alternative d'amplitude et de fréquence désirés. Cette transformation s'opère dans un convertisseur appelé onduleur (figure2.1).

L'amplitude et la fréquence de la tension peuvent être fixes ou variables. La variation en amplitude de la tension peut être obtenue, soit par la variation de la tension continue quant le gain du convertisseur est constant, soit par la variation du gain de l'onduleur lorsque la tension continue est constante. La modification du gain de l'onduleur est normalement accomplie par la commande de l'onduleur en utilisant la technique de modulation en largeur d'impulsions (MLI).

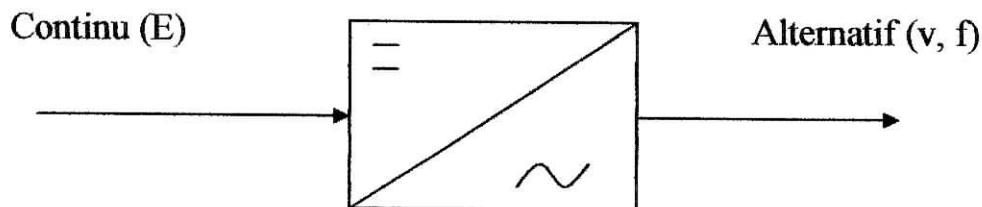


Figure2.1 Schéma bloc de la conversion continue - alternative

Le gain de l'onduleur peut être défini comme étant le quotient de la valeur efficace de la tension alternative sur la tension continue.

Un onduleur idéal devrait fournir une tension alternative sinusoïdale. Cependant, les onduleurs réels produisent une onde de tension non sinusoïdale et contient donc un certain nombre d'harmoniques. Pour les applications à faible et à moyenne puissance, une tension alternative à onde carrée ou à onde quasi-carrée est acceptable ; cependant pour les puissances élevées est requise une onde de tension sinusoïdale de faible distorsion. Grâce à la

disponibilité de nouveaux composants à semi-conducteurs rapides et de fortes puissances, il est possible de minimiser ou de réduire les harmoniques que contient l'onde de tension.

Les onduleurs sont largement utilisés dans plusieurs branches de l'industrie. Dans notre application, l'onduleur est utilisé comme variateur de vitesse pour moteur asynchrone. La tension d'entrée peut être fournie par une batterie, un générateur photovoltaïqueetc.

Les onduleurs sont classés en fonction du nombre de phases (monophasé ou triphasé) présentes à la sortie, en fonction de la grandeur à transformer et en fonction du mode de commutation (MLI, résonance,...) des interrupteurs. Par exemple on doit pour caractériser un onduleur (onduleur triphasé de tension à MLI).

2. Principe de l'onduleur de tension

2.1 Onduleur en demi - pont

Le principe de l'onduleur monophasé en demi - pont peut être expliqué à partir de la figure3.2. On dispose d'une source de tension continue à point milieu et de 2 interrupteurs (Q1, Q2) ; le récepteur est une résistance. Quand seul le transistor Q1 est fermé pendant $T/2$, la tension $v(t)$ prend la valeur $E_s/2$. Quand seul le transistor Q2 est fermé pendant $T/2$, la tension $v(t)$ prend la valeur $-E_s/2$, les deux transistors Q1 et Q2 ne doivent jamais conduire en même temps.

La tension $v(t)$ ne dépend que de la commande des interrupteurs, elle prend sur chaque demi - période successivement ($E_s/2, -E_s/2$).

Le courant $i(t)$ dépend de la forme de la tension $v(t)$ et du type du récepteur ; les diodes D1, D2 assurent un retour d'énergie à la source continue, elles sont alors appelées diodes de récupération.

Cet onduleur nécessite une source continue à point milieu, quand un transistor est bloqué, sa tension inverse est $v_{inv}=E_s$.

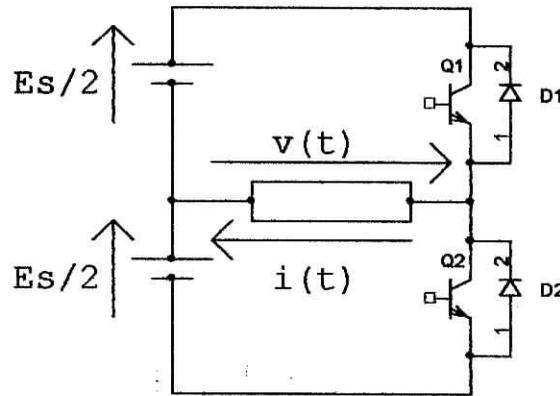


Figure2.2 Onduleur demi - pont

2.2 Onduleur monophasé en pont

Pour éviter la nécessité d'une source à point milieu et doubler la tension de sortie $v(t)$, on utilise un onduleur en pont (figure2.3). Dans ce cas, on commande sur une demi - période les transistors Q1, Q2 et les transistors Q3, Q4 sur le reste de la période.

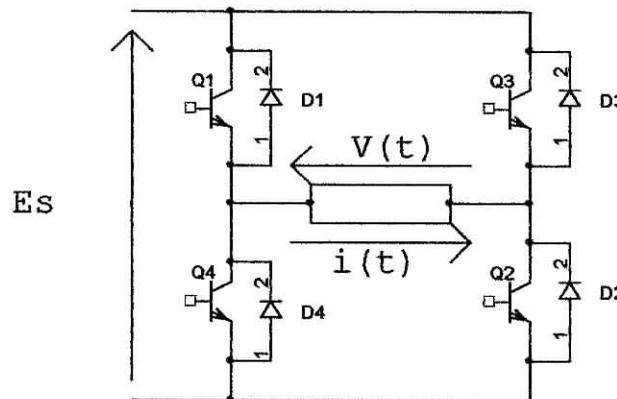


Figure2.3 Onduleur monophasé en pont

Remarque

La tension inverse maximale de blocage de chaque transistor et les performances de la tension alternative, pour l'onduleur en demi - pont et en pont, sont les mêmes. Cependant pour l'onduleur en pont avec une charge résistive, la puissance de sortie est 4 fois plus grande et le fondamental est doublé.

2.3 Onduleur triphasé

Les onduleurs triphasés sont normalement utilisés pour les applications de grande puissance. Trois onduleurs monophasés en demi - pont (ou en pont) peuvent être connectés comme indiqué à la figure 3.4 pour constituer un onduleur triphasé. Les impulsions d'allumage des onduleurs monophasés doivent être avancés ou retardés de 120° les uns par rapport aux autres afin d'obtenir 3 tensions triphasées équilibrées (pour le fondamental). Le récepteur étant couplé en étoile, les tensions de phase (v_{na} , v_{nb} , v_{nc}) sont liées aux tensions de sortie de l'onduleur (v_{oa} , v_{ob} , v_{oc}) par :

$$V_{oa} = V_{on} + V_{na}$$

$$V_{ob} = V_{on} + V_{nb}$$

$$V_{oc} = V_{on} + V_{nc}$$

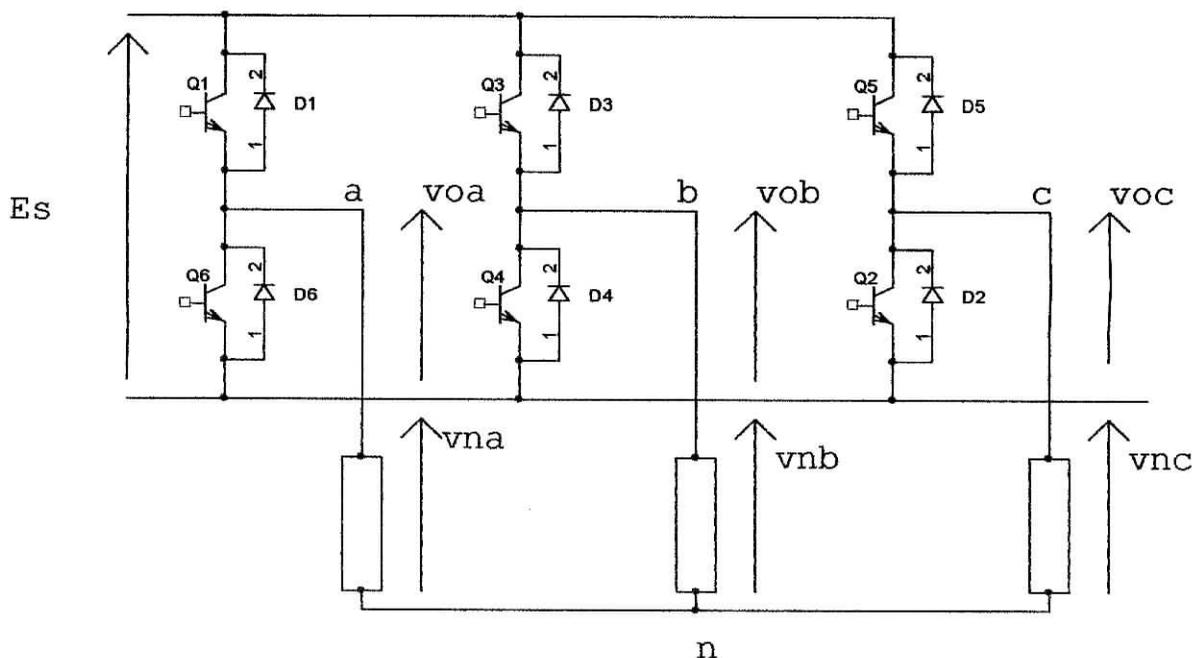


Figure 2.4 Onduleur triphasé

3. Réglage de tension par modulation de largeur d'impulsion

Elle consiste à générer, par alternance de la tension alternative, une tension composée de plusieurs créneaux de largeurs variables. Elle permet ainsi, d'obtenir un fondamental de tension variable en amplitude et en fréquence.

3.1 Technique triangulo - sinusoïdale

La simplicité d'implantation de cette technique par des moyens analogiques lui a permis d'être la plus utilisée jusqu'à l'apparition récente des microprocesseurs qui a rendu possible l'implantation d'autres techniques compétitives donnant ainsi un large choix.

Le principe de cette technique consiste à comparer un signal sinusoïdal d'amplitude et de fréquence variable appelé référence à un signal triangulaire appelé porteuse (figure2.5). L'intersection de ces signaux donne les instants de commutation des interrupteurs.

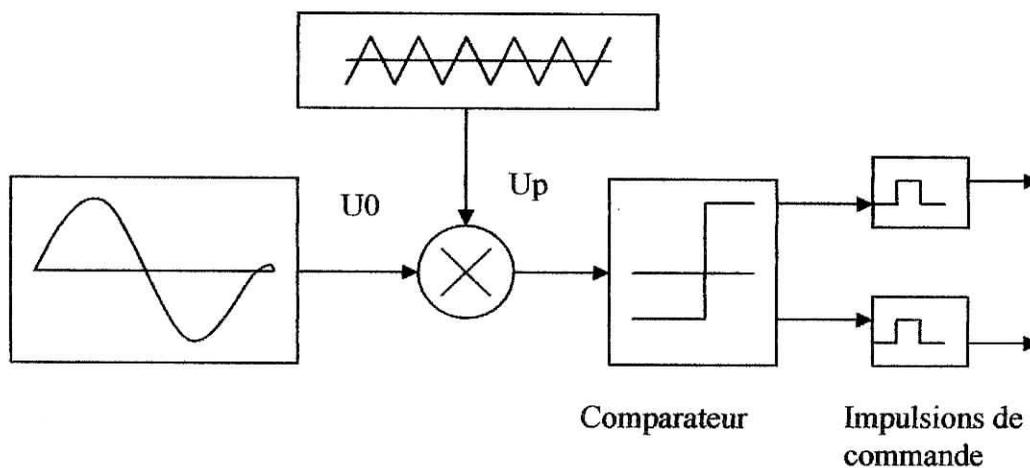


Figure2.5 Schéma synoptique de la modulation triangulo - sinusoïdale

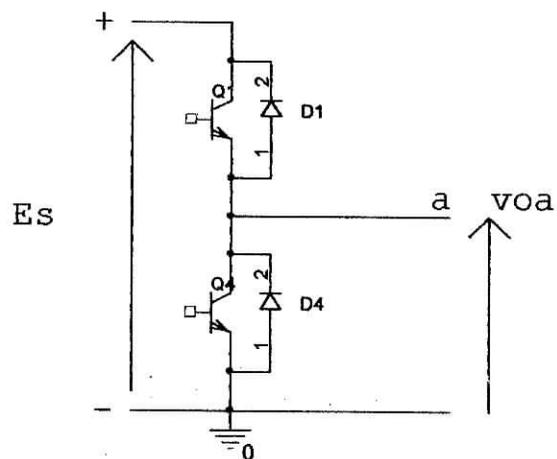


Figure2.6 : bras onduleur

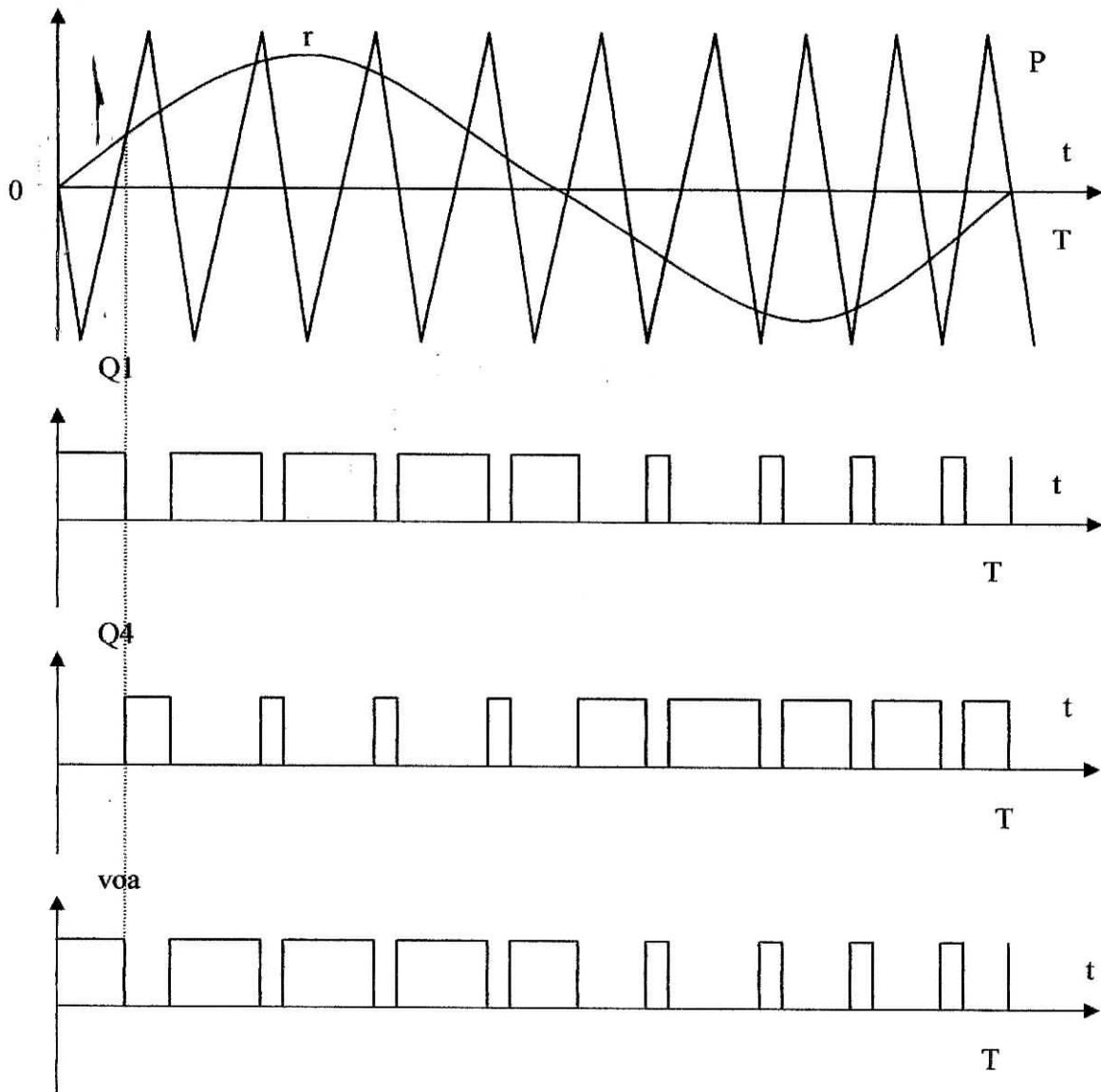


figure2.7 Principe de génération du signal MLI par la méthode triangulo-sinusoidale

Ainsi pour un bras d'onduleur, les signaux de commande des transistors Q1, Q4 et la forme de la tension de sortie voa sont tels qu'indiqués à la figure2.7. Il est à noter que le signal de commande de Q4 est complémentaire du signal de commande de Q1. On caractérise cette technique par 2 paramètres :

-L'indice de modulation M égal au rapport de la fréquence de la porteuse (f_p) par celle de la fréquence (f_{rs})

$$M = \frac{f_p}{f_{rs}} \quad (2.1)$$

-Le coefficient de réglage r, égal au rapport de l'amplitude de la référence $V_{rs}(\max)$ sur l'amplitude de la porteuse $V_p(\max)$

$$r = \frac{V_{rs}(\max)}{V_p(\max)} \quad (2.2)$$

La modulation est dite synchrone si M est entier, elle est dite asynchrone s'il ne l'est pas. En modulation synchrone, si le maximum de la référence correspond à un pic de la porteuse, on dit que le calage est optimal et la tension alternative bénéficie d'une double symétrie par rapport au quart et au milieu de la période. L'exemple est donné à la figure2.7, illustrant la forme des tensions pour M=9 et r=0.8.

a. Cas du pont monophasé

Un onduleur en pont monophasé (figure2.3) peut être commandé soit pour obtenir un signal de sortie bipolaire, soit du type unipolaire. Dans le premier cas, les transistors Q1, Q2 reçoivent les mêmes signaux de commande et les transistors Q3, Q4 reçoivent les signaux complémentaires aux premières cibles.

Pour obtenir une tension de sortie unipolaire en MLI, il suffit de commander les transistors Q1, Q4 en MLI avec la commande de Q4 égale au complément de Q1, de plus il faut commander les transistors Q3, Q2 en créneau de durée égales. La commande de Q3 étant complémentaires de celle de Q2.

b- Cas de l'onduleur en pont triphasé

En MLI, pour obtenir les tensions de sortie triphasées, il faut disposer de trois signaux de référence qui forment un système triphasé. La comparaison de chacune de ces références à la porteuse triangulaire fournit les signaux de commande des 2 interrupteurs du même bras.

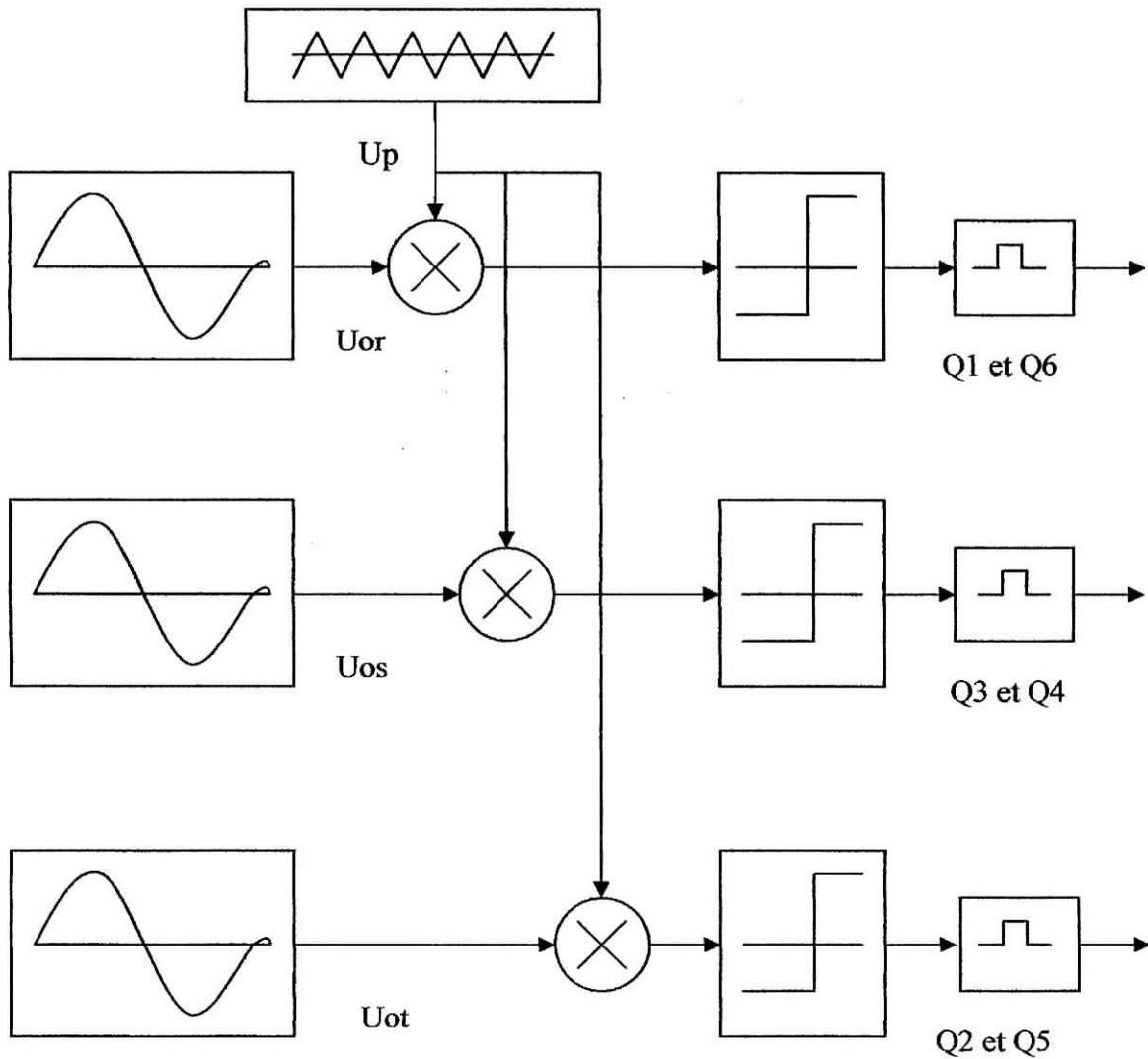


Figure2.8 Schéma synoptique de la modulation triangulo – sinusoidale

3.2 Les techniques à échantillonnage

Les techniques à échantillonnage ont l'avantage de présenter des équations simples pour le calcul des instants de commutation. Ils existent, la technique à échantillonnage régulière, celle à échantillonnage régulière asymétrique et enfin à échantillonnage régulière modifié.

Nous ne présentons que la technique à échantillonnage régulier symétrique. Elle consiste à comparer une porteuse – triangulaire à une version discrétisée de la référence sinusoidale. La référence sinusoidale (r_s) est échantillonnée, aux instants des pics positifs de la porteuse (P), pour obtenir la référence discrétisée (r_d).

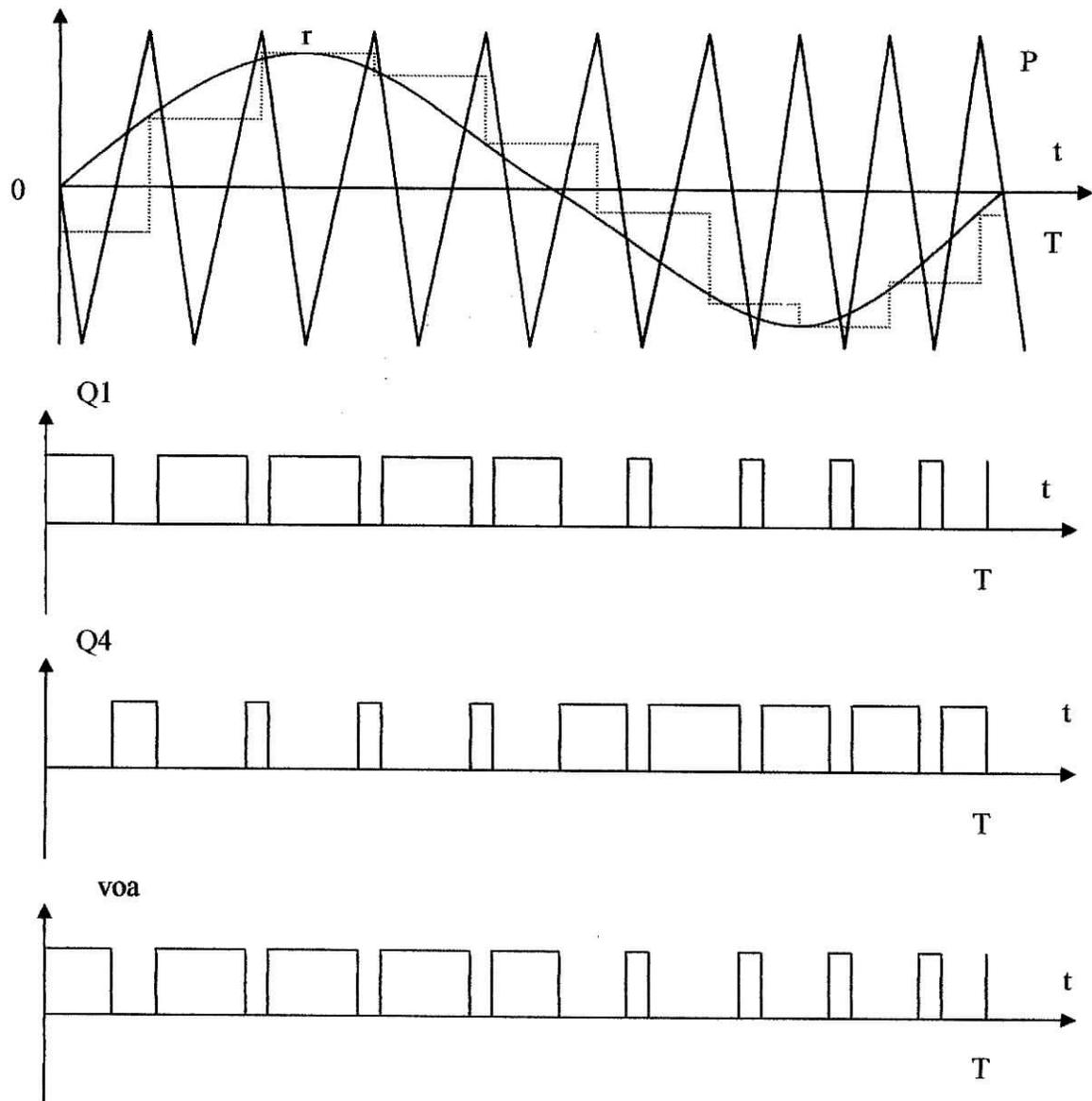


Figure2.9 Principe des techniques d'échantillonnages

Pour un bras d'onduleur, les commandes des transistors et la tension de sortie de l'onduleur sont indiqués à la figure2.9.

3.3 Méthode de réglage par élimination d'harmoniques

Les harmoniques indésirables de l'onde en créneaux peuvent être éliminés et le fondamental peut être contrôlé par la méthode d'élimination d'harmonique avec asservissement du fondamental. Dans cette méthode, on produit des trous dans l'onde carrée à des angles prédéterminés (comme indiqué sur la figure2.10).

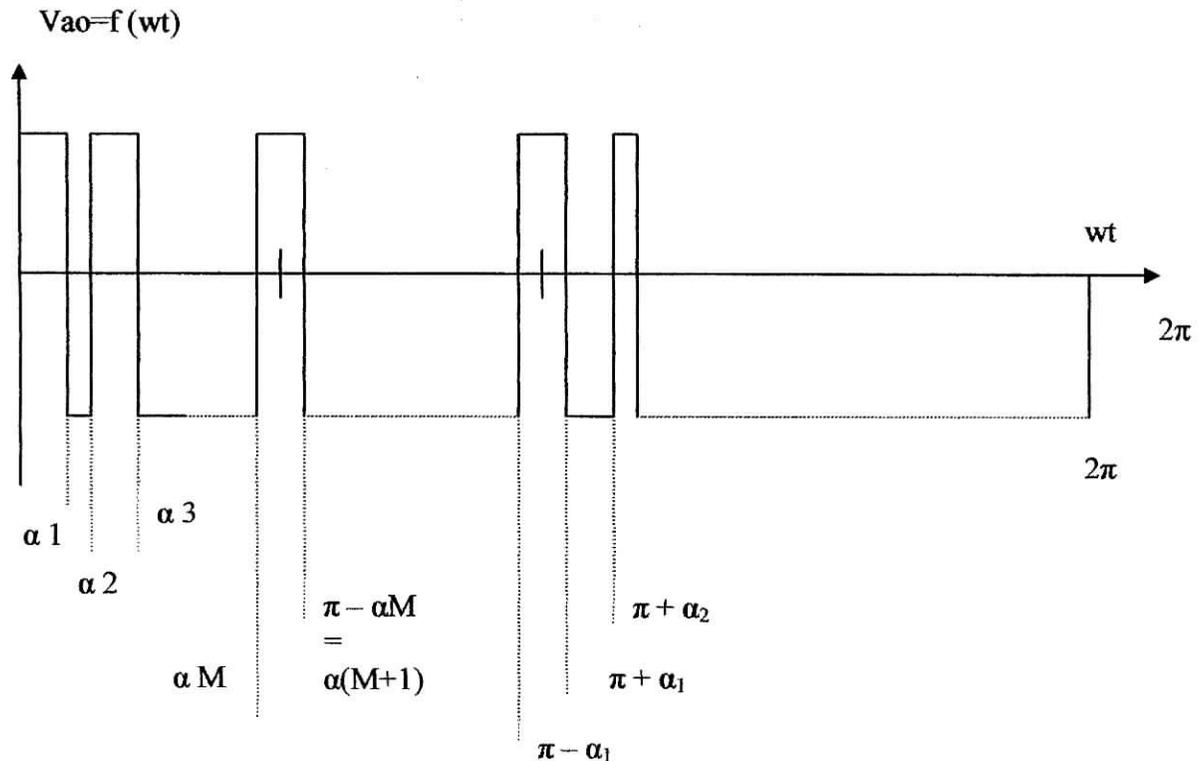


Figure2.10 Principe de la méthode de réglage par élimination d'harmoniques

Il est montré que les angles α_1 , α_2 , α_3 et α_4 où se positionnent les 4 trous peuvent être commandés pour éliminer 3 harmoniques du spectre de la tension et varier la tension du fondamentale.

Un grand nombre d'harmoniques peuvent être éliminés si l'onde de tension peut s'accorder de trous additionnels. Dans le cas général, pour éliminer $(k-1)$ harmoniques et commander le fondamental, il faut disposer de k positions angulaires $(\alpha_1, \alpha_2, \dots, \alpha_k)$ pour décrire l'onde de tension.

Pour calculer les k positions angulaires des trous, il faut résoudre un système de k équations algébriques non – linéaires de la forme :

$$bn = \frac{4}{n\pi} \left(1 + 2 \sum_{i=1}^k (-1)^i \cos(n\alpha_i) \right)$$

Si par exemple, il faut éliminer le 5^{ième} et le 7^{ième}, le 3^{ième} harmonique et ses multiples sont à ignorer si le récepteur est couplé en étoile avec neutre isolé. Donc il suffit de prendre $k=3$ et les 3 équations nécessaires pour déterminer $(\alpha_1, \alpha_2, \alpha_3)$ sont telle que :

Le fondamental,
$$b1 = \frac{4}{\pi} (1 - 2 \cos \alpha_1 + 2 \cos \alpha_2 - 2 \cos \alpha_3)$$

L'harmonique 5,

$$b5 = \frac{4}{5\pi} (1 - 2 \cos(5\alpha_1) + 2 \cos(5\alpha_2) - 2 \cos(5\alpha_3))$$

L'harmonique 7,
$$b7 = \frac{4}{7\pi} (1 - 2 \cos(7\alpha_1) + 2 \cos(7\alpha_2) - 2 \cos(7\alpha_3))$$

Les trois équations algébriques non – linéaires sont résolues numériquement pour une amplitude spécifiée du fondamental et ainsi les trois angles sont entièrement déterminés.

Chapitre 3

Technique MLI programmée

Dans ce chapitre, on va étudier la méthode de génération du signal MLI programmé à commande de tension et élimination des harmoniques, et on va calculer les valeurs exactes des angles de commutation avec asservissement du fondamental et élimination des harmoniques, telle que définie par la technique de Patel et Hoft.

1. Description de la méthode de Patel et Hoft

Considérons un onduleur monophasé en demi pont, soit la tension de sortie à deux états de l'onduleur demi pont, les angles de commutations impaires $\alpha_1, \alpha_3, \dots$ définissent des transitions négatives, tandis que les angles de commutations paires $\alpha_2, \alpha_4, \dots$ définissent des transitions positives, on suppose la tension de sortie périodique d'amplitude unité. Dans ce cas, la tension de sortie V_{ao} peut s'écrire en série de Fourier :

$$V_{ao} = f(\alpha) = a_0 + \sum_{n=1}^{\infty} (a_n \sin(n\alpha) + b_n \cos(n\alpha)) \quad (3.1)$$

Les coefficients a_n et b_n sont donnés par :

$$\begin{aligned} a_0 &= \frac{1}{2\pi} \int_0^{2\pi} f(\alpha) d\alpha \\ a_n &= \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \sin(n\alpha) d\alpha \\ b_n &= \frac{1}{\pi} \int_0^{2\pi} f(\alpha) \cos(n\alpha) d\alpha \end{aligned} \quad (3.2)$$

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

D'autre part comme $f(\alpha)$ présente une symétrie demi onde i.e

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

La valeur moyenne a_0 est nulle et seulement les harmoniques impairs existent. Par conséquent, l'indice prend les valeurs impaires 1, 3, 5, 7, 9,

Les coefficients a_n et b_n sont alors donnés par :

$$\begin{aligned}
 a_0 &= 0 \\
 a_n &= \frac{2}{\pi} \int_0^\pi f(\alpha) \sin(n\alpha) d\alpha \\
 b_n &= \frac{2}{\pi} \int_0^{2\pi} f(\alpha) \cos(n\alpha) d\alpha
 \end{aligned} \tag{3.3}$$

Remplaçons $f(\alpha)$ par sa valeur dans l'équation (3.3) :

$$a_n = \frac{2}{\pi} \left[\int_{\alpha_0}^{\alpha_1} (-1)^0 \sin(n\alpha) d\alpha \right] + \dots + \frac{2}{\pi} \left[\int_{\alpha_{2M}}^{\alpha_{(2M+1)}} (-1)^{2M} \sin(n\alpha) d\alpha \right] \quad n=1, 3, 5, \dots$$

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

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

Avec $\alpha_{(2M+1)} = \pi$ et $\alpha_0 < \alpha_1 < \alpha_2 < \alpha_3 \dots < \alpha_{(2M+1)}$.

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

Comme :

$$\alpha_0 = 0$$

$$\alpha_{2M+1} = \pi$$

On déduit :

$$\cos(n\alpha_0) = 1$$

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

D'où :

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

De même pour les coefficients b_n , on trouve après simplification le résultat suivant :

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

Comme n doit être impair, on peut écrire :

$$a_n = \frac{4}{n\pi} \left[1 + \sum_{k=1}^{2M} (-1)^k \cos(n\alpha_k) \right] \quad n \text{ impair} \quad (3.4a)$$

$$b_n = \frac{4}{n\pi} \left[- \sum_{k=1}^{2M} (-1)^k \sin(n\alpha_k) \right] \quad n \text{ impair} \quad (3.4b)$$

D'autre part la forme d'onde $f(\omega t) = V_{ao}(t)$ présente une symétrie quart d'onde i.e

$$f(\alpha) = f(\pi - \alpha)$$

et d'après la figure de l'onde MLI on a :

$$\alpha_k = \pi - \alpha_{2M-k+1} \quad k=1,2,\dots, M$$

d'où :

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

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

Pour n impair on a :

$$\sin(n\pi) = 0 \quad \cos(n\pi) = -1$$

D'où :

$$\sin(n\alpha_k) = \sin(n\alpha_{2M-k+1}) \quad k=1, 2, 3, \dots, M \quad (3.5)$$

Remplaçons (3.5) dans (3.4b) :

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

D'autre part :

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

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

D'où :

$$\cos(n\alpha_k) = -\cos(n\alpha_{2M-k+1}) \quad (3.6)$$

Remplaçons (3.6) dans (3.4a), on obtient :

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^M (-1)^k \cos(n\alpha_k) \right] \quad (3.7)$$

Avec n impair et différent d'un multiple de 3.

On considère une alimentation unité, ie $E_d/2=1$.

Le coefficient a_n est l'amplitude de l'harmonique de rang n du signal suivant :

$$Va_0(t) = f(\omega t) = \sum_{n=1}^{\infty} a_n \sin(n\omega t) \quad (3.8)$$

Les équations (3.7) possèdent m variables inconnues $\alpha_1, \alpha_2, \dots, \alpha_M$ appelées angles de commutations exacts. Le problème est de calculer les valeurs de celles-ci qui permettent :

1. d'annuler les amplitudes a_n des $(m-1)$ premiers harmoniques f_1 :

$$f_n(\omega t) = a_n \sin(n\omega t) \quad n \neq 1$$

2. d'assigner une valeur déterminée au fondamental f_1 :

$$f_1(\omega t) = a_1 \sin(\omega t)$$

Ces équations sont non linéaires. On utilise la méthode de Newton-Raphson pour résoudre ce système de m équations non linéaires à m inconnues.

1.1 Remarque sur les angles de commutation :

Considérons un bras d'onduleur à transistor, on a :

$\alpha = \alpha_0 = 0$, Th1 conduit, Th2 bloqué

$\alpha = \alpha_1$, Th2 conduit, Th1 bloqué

$\alpha = \alpha_2$, Th1 conduit, Th2 bloqué

.....

$\alpha = \alpha_M$, Th2 conduit, Th1 bloqué

Donc lorsque Th1 conduit (et Th2 bloqué) on a $f(\omega t) = +1$, et lorsque Th2 conduit (et Th1 bloqué), on a $f(\omega t) = -1$, autrement dit, les thyristors Th1 et Th2 conduisent alternativement pour connecter le point a aux phases positives et négatives.

1.2 Onduleur demi-pont à transistors

L'onduleur demi-pont à transistors offre une solution simple et économique pour les faibles et moyennes puissances (inférieur à 30 KW).

L'onduleur MLI est intéressant pour les applications ne supportant pas les pulsations de couple: Traction électrique à partir d'une alimentation continue, l'industrie textile et métallurgie.

Le fonctionnement des transistors à des fréquences (plusieurs de KHZ) plus élevées que celles des thyristors permettent d'augmenter le rang des harmoniques éliminés (m élevé). Les circuits de filtrage se trouvent donc simplifiés.

2. Calcule des valeurs exactes des angles de commutation par la méthode de Newton-Raphson :

La relation (3.7) est un système de m équations non linéaires à m inconnues $\alpha_1, \alpha_2, \dots, \alpha_M$. On assigne un valeur déterminée m , appelée indice de modulation, à amplitude a_1 du fondamental et on annule les amplitudes a_n des $(m-1)$ premiers harmoniques.

On résout ce système par la méthode itérative de Newton-Raphson. Celle-ci converge bien (quadratiquement) si l'on possède un bon estimé initial de la solution.

Cet estimé peut être obtenu par la méthode du gradient. Mais, celle-ci étant lente (convergence linéaire), on utilisera plutôt l'algorithme 'on-line' de Taufik, Mellitt et Goudman pour estimer rapidement les valeurs initiales de la solution du système non linéaire.

Pour les montages triphasés, les harmoniques de rang 3 et multiple de 3 sont inopérants. Pour cette raison les triplettes ne sont pas éliminées dans cette étude.

D'autre part, il faut éliminer deux harmoniques de tension pour éliminer un harmonique de courant. Comme l'amplitude du fondamental doit être fixée à une valeur déterminée, ceci fixe la première valeur de m à 3 (m étant le nombre de commutations par quart d'onde ou nombre de découpages par demi-onde). Par conséquent, lorsque m augmente successivement par pas égal à 2, le nombre d'harmoniques de courants qui seront éliminés augmente par pas égal à 1. Finalement on obtient un système de m équations non linéaires de la forme :

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

avec $m=5$ donc, $n=1,5,7,11,13$, pour m égal à 7, n prend les valeurs 1,5,7,11,13,17,19 etc.....

Le système (3.7) s'écrit encore :

$$\begin{aligned}
 a_1 &= \frac{4}{\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(\alpha_k) \right] = -im \\
 a_5 &= \frac{4}{5\pi} \left[1 + 2 \sum (-1)^k \cos(5\alpha_5) \right] = 0 \\
 a_7 &= \frac{4}{7\pi} \left[1 + 2 \sum (-1)^k \cos(7\alpha_k) \right] = 0 \\
 a_{11} &= \frac{4}{11\pi} \left[1 + 2 \sum (-1)^k \cos(11\alpha_k) \right] = 0 \\
 &\dots\dots\dots \\
 a_n &= \frac{4}{n\pi} \left[1 + 2 \sum (-1)^k \cos(n\alpha_k) \right] = 0
 \end{aligned} \tag{3.9}$$

Ces amplitudes sont normalisées i.e que la tension d'alimentation continue est supposée égale à l'unité.

On doit signaler que la valeur de l'indice de modulation im assignée au fondamental est un indice sans dimension variant de 0 à 1.15. Pour obtenir la valeur correspondante en volt, il faut multiplier im par $Ed/2$, la tension d'alimentation continue de l'onduleur demi-pont.

D'autre part la méthode itérative de Newton – Raphson ne converge pas pour une valeur positive de im . C'est pourquoi on assigne une valeur négative ($-im$) au fondamental. Ce qui correspond à un déphasage de π du fondamental. Ce déphasage est sans effet sur le moteur.

En résumé, on a un système de forme générale.

$$f_1(\alpha_1, \alpha_2, \dots, \alpha_m) = \frac{4}{\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(\alpha_k) \right] + im = 0$$

$$f_2(\alpha_1, \alpha_2, \dots, \alpha_m) = \frac{4}{5\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(5\alpha_k) \right] = 0$$

$$f_3(\alpha_1, \alpha_2, \dots, \alpha_m) = \frac{4}{7\pi} \left[1 + 2 \sum_{k=1}^m 1 + 2 \sum_{k=1}^m (-1)^k \cos(7\alpha_k) \right] = 0 \quad (3.10)$$

.....

$$f_m(\alpha_1, \alpha_2, \dots, \alpha_m) = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^m (-1)^k \cos(n\alpha_k) \right] = 0$$

2.1 Estimation initiale de la solution

Pour assurer la convergence de la méthode de Newton-Raphson, on doit obtenir un bon estimé initial de la solution 'exacte' recherchée. On peut utiliser la méthode du gradient mais celle-ci est lente. On va utiliser l'algorithme de Taufik, Mlelitt et Goodman, noter TMG.

Description de l'algorithme TMG :

K impair :

$$\alpha(k) = \frac{60(k+1)}{(m+1)} - \left[\frac{2 \times 60}{(m+1)} \times \frac{\Delta_k}{0.8} \times NP1 \right] \quad k=1, 3, 5, \dots, m \quad (3.11)$$

Avec :

$$\Delta_k = 0.4025 \sin \left[\frac{59.184^\circ(k-0.5)}{m} + 60.408 \right] \quad (3.12)$$

avec NP1 l'indice de modulation (identique à l'indice im) variant de 0 à 0.8.

K pair :

$$\alpha(k) = \frac{60k}{(m+1)} + \left[\frac{2 \times 60}{(m+1)} \times \frac{\Delta_k}{0.8} \times NP1 \right] \quad k=2, 4, \dots, (m-1) \quad (3.13)$$

avec :

$$\Delta_k = 0.381 \sin \left[\frac{k}{(m-1)} \left(58.558 - \frac{1.135 \times 11}{(m-2)} \right) \right] \quad (3.14)$$

avec NP1 l'indice de modulation (identique à l'indice im) variant de 0 à 0.8.

Pour NP1 supérieur à 0.8, on utilisera les facteurs de corrections suivantes :

k impair

$$\alpha(k)_{\text{précis}} = \alpha(k)_{\text{appro}} - \frac{(NP1 - 0.8)^2}{0.09} \times \frac{13}{m} \times \sin\left[\frac{180k}{m+5}\right] \quad (3.15)$$

k pair :

$$\alpha(k)_{\text{précis}} = \alpha(k)_{\text{approx}} - \frac{(NP1 - 0.8)^2}{0.09} \times \frac{14}{m} \times \sin\left[\frac{180(k-1.5)}{m}\right] \quad (3.16)$$

2.2 Résolution du système non linéaire par la méthode de Newton-Raphson

Algorithme :

Notons :

$$\alpha' = (\alpha_1^*, \alpha_2^*, \dots, \alpha_m^*)$$

Le vecteur solution du système non linéaire (3.10) :

$$f_i(\alpha) = 0 \quad i=1, m$$

avec $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$

Si chaque fonction f_i est continue différentiable, alors on peut la développer en série de Taylor dans le voisinage d'un estimé $\alpha^{(k)}$ obtenue à la $k^{\text{ième}}$ itération proche de α^* .

On obtient :

$$\begin{aligned} f_i(\alpha^*) &= f_i(\alpha^{(k)} + (\alpha^* - \alpha^{(k)})) \\ &= f_i(\alpha^{(k)}) + \sum_{j=1}^m \left[\frac{\partial f_i(\alpha)}{\partial \alpha_j} \right]_{\alpha=\alpha^{(k)}} (\alpha_j^* - \alpha_j^{(k)}) + \dots \\ &\dots + \frac{1}{2!} \sum_{j=1}^m \sum_{r=1}^m (\alpha_j^* - \alpha_j^{(k)}) (\alpha_r^{(k)}) \left[\frac{\partial^2 f_i(\alpha)}{\partial \alpha_j \partial \alpha_r} \right]_{\alpha=\alpha^{(k)}} + \dots = 0 \end{aligned} \quad (3.17)$$

Pour $i=1, \dots, m$

Si $\alpha^{(k)}$ est un estimé proche de α^* , les éléments $(\alpha_j^* - \alpha_j^{(k)})^2$ sont négligeable ainsi que les termes de degré supérieur à 2.

Le système (3.17) s'écrit donc :

$$\left\{ \sum_{j=1}^m \frac{\partial f_i(\alpha)}{\partial \alpha_j} \Big|_{\alpha=\alpha^{(k)}} (\alpha_j^* - \alpha_j^{(k)}) = -f_i(\alpha^{(k)}) \right. \quad (3.18)$$

avec : $i=1, \dots, m$.

Définissons la matrice des dérivées premières :

$$E^{(k)} = (E_{ij}^{(k)})$$

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

D'où :

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

avec m impair et n impair et différent d'un multiple de 3.

Définissons l'erreur :

$$\Delta \alpha^{(k)} = [\Delta \alpha_1^{(k)}, \Delta \alpha_2^{(k)}, \dots, \Delta \alpha_m^{(k)}] \quad (3.19)$$

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

Soit le vecteur :

$$F^{(k)} = [F_1^{(k)}, F_2^{(k)}, \dots, F_m^{(k)}]$$

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

Alors le système (3.18) s'écrit sous la forme matricielle suivante :

$$E^{(k)} \cdot \Delta \alpha^{(k)} = F^{(k)} \quad (3.20)$$

ou $\Delta\alpha^{(k)}$ est le vecteur inconnu.

Le système (3.20) est un système linéaire que l'on peut résoudre par l'algorithme de Gauss. Celui-ci sera décrit en détail plus loin.

Une fois le vecteur $\Delta\alpha^{(k)}$ déterminé, on obtient un meilleur estimé $\alpha^{(k+1)}$ de α^* par la relation :

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

On continue jusqu'à ce que $|\alpha^* - \alpha^{(k)}| \rightarrow 0$

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

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

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

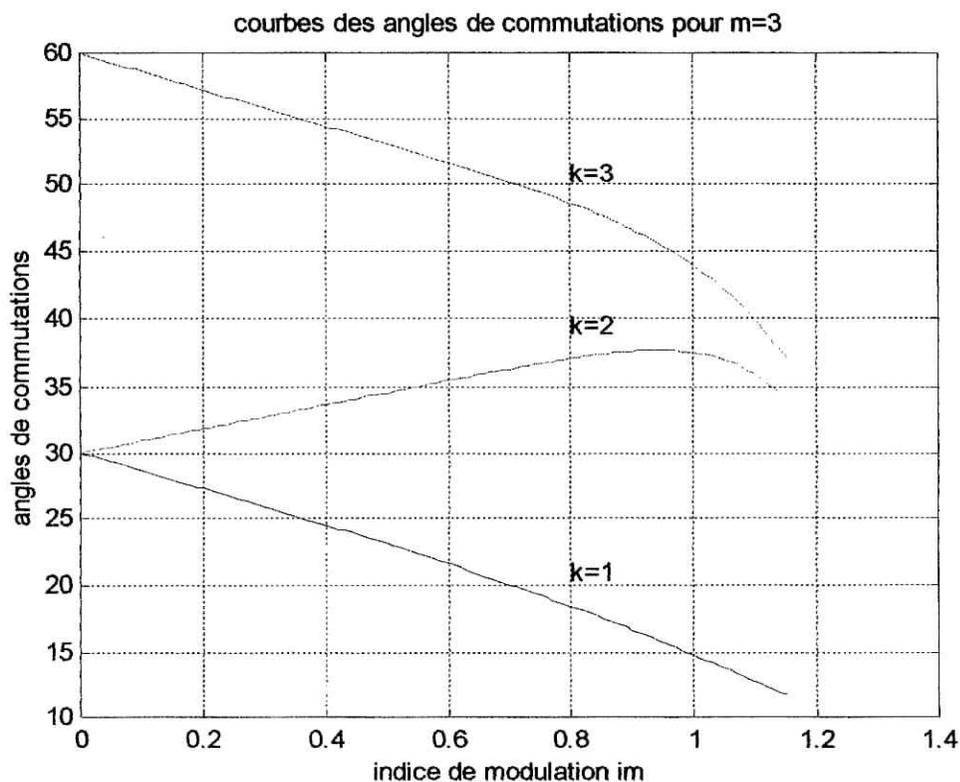


Figure 3.1 Courbes des angles exacts de commutation pour m=3

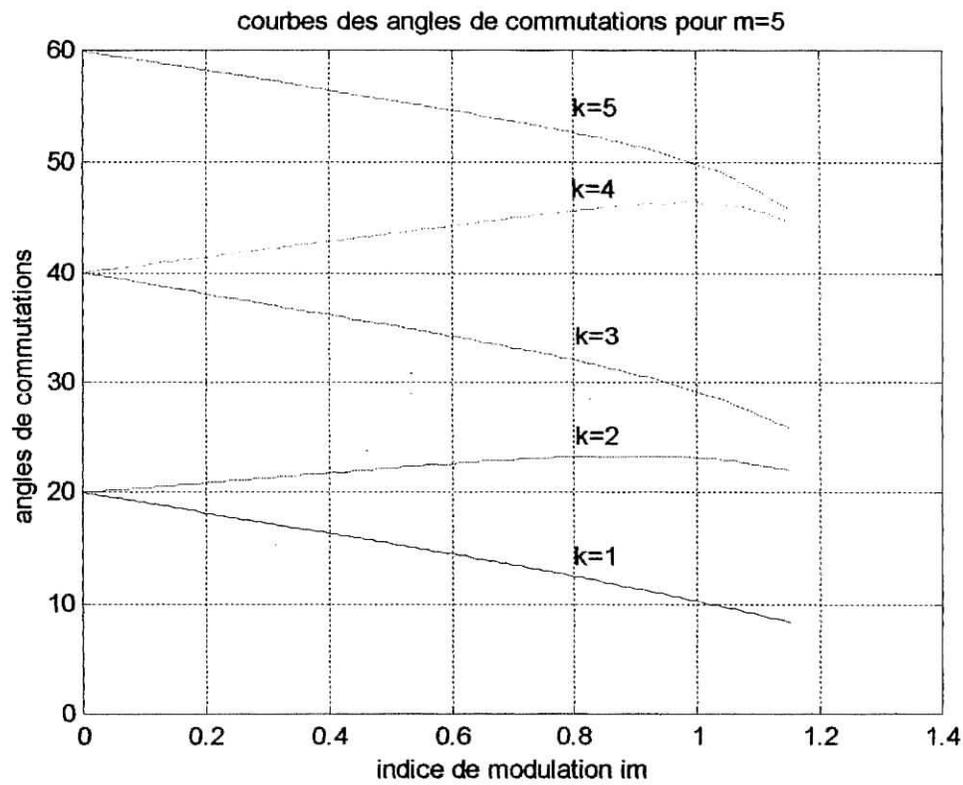


Figure 3.2 Courbes des angles exacts de commutation pour $m=5$

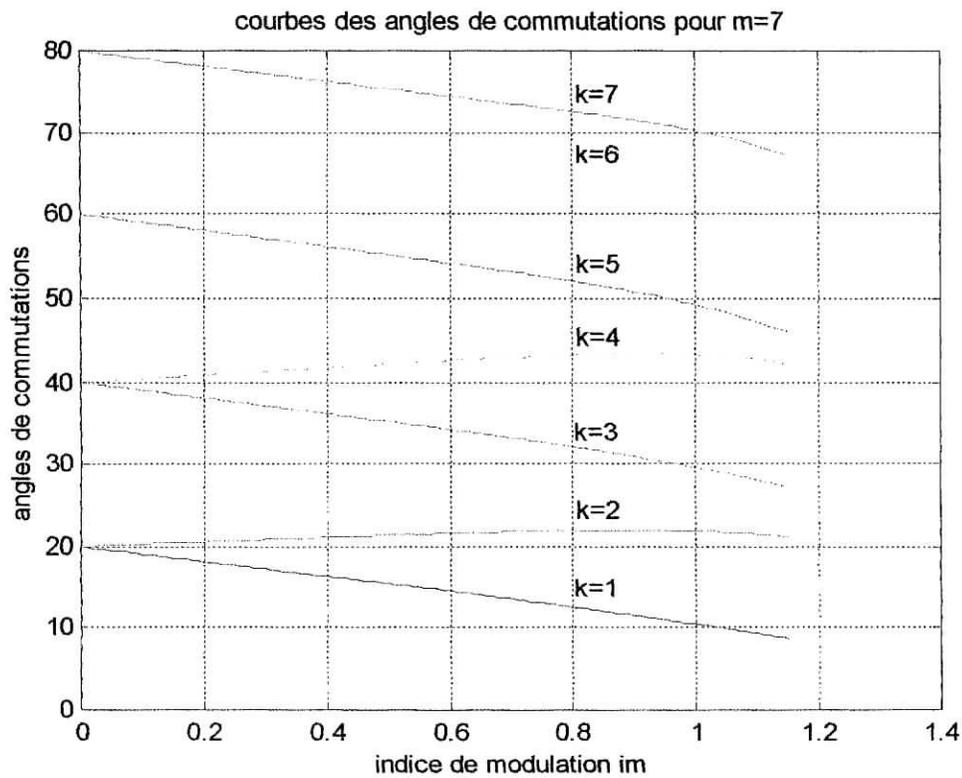


Figure 3.3 Courbes des angles exacts de commutation pour $m=7$

2.3 Résolution du système linéaire par l'algorithme de Gauss :

La méthode de Gauss consiste à transformer le système linéaire matricielle $AX=B$, à matrice A quelconque, en un système équivalent $A'X=B'$, où A' est une matrice triangulaire supérieure ; la résolution de ce dernier système est immédiate. Ceci est valable si les 'pivots' $a(k, k)$ sont non nuls.

Pour un système linéaire d'ordre n, l'algorithme de Gauss est le suivant :

1-Verification du pivot non nul :

Au cours de la $(k-1)^{ième}$ étape, on veut annuler les termes sous diagonaux de la $(k-1)^{ième}$ colonne :

$$a_{k+1,k}^{(k-1)} = a_{k+2,k}^{(k-1)} = \dots = a_{n,k}^{(k-1)} = 0$$

Si $a_{k,k}^{(k-1)} = 0$ et il existe l tel que ($l > k$ et $a_{l,k}^{(k-1)} \neq 0$) alors on permute les lignes k et l de la matrice $[A, b]^{(k-1)}$ de façon à avoir un pivot non nul.

D'autre part pour minimiser l'erreur d'arrondi, on évite la division par de petits pivots. Pour cette raison, on recherche le plus grand pivot soit par le procédé du pivot partiel ou total.

2-Triangularisation :

$$w = a_{ik} / a_{kk}$$

$$a_{ij} = a_{ij} - w \cdot a_{kj}$$

Avec $j = k+1, n+1$; $i = k+1, n$; $k = 1, n-1$.

3-Résolution du système résultant :

$$x_i = (b_i - \sum_{j=i+1}^n a_{ij}) / a_{ii} \quad \text{avec } i = n, n-1, \dots, 1$$

Pour adapter cet algorithme à notre problème, il faut remplacer la matrice A par la matrice $E^{(k)}$, le vecteur colonne inconnu X par le vecteur colonne $\Delta\alpha^{(k)}$ et le vecteur colonne B par $F^{(k)}$.

Après avoir calculé $\Delta\alpha^{(k)}$, on calcule un meilleur estimé $\alpha^{(k+1)}$ de α^* par la relation :

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

3. Conclusion

Dans ce chapitre, on a montré que le calcul des valeurs exactes des angles de commutation d'une tension MLI, avec asservissement du fondamental et élimination des harmoniques, exige la résolution d'un système de m équations non linéaires à m inconnues, m étant le nombre de commutations par quart d'onde.

Le calcul des angles à l'aide de cet algorithme nécessite un temps de calcul très élevé ce qui empêche une commande de vitesse temps réel. La technique de commutation est d'abord calculée sur ordinateur ensuite ces valeurs sont stockées en mémoire pour pouvoir être utilisées dans une commande de vitesse d'un moteur. Un tel procédé requiert une capacité mémoire élevée qui augmente le coût de l'application.

Dans le chapitre suivant, on se propose de décrire un nouvel algorithme 'on-line', basé sur une approximation des angles exactes par la méthode des moindres carrés, ce qui calcule les angles de commutation avec pratiquement la même précision que celle de la technique 'off-line' de Patel et Hoft et un temps de calcul plus faible que les autres techniques 'on-line' connues.

Une faible capacité mémoire est requise pour des coefficients les équations de l'algorithme proposé.

Chapitre 4

Le nouveau algorithme 'on-line' pour le calcul des angles de commutations

Ce chapitre sera divisé en deux parties, On va présenter dans la première partie un algorithme qui nous permet de calculer les angles de commutation $\alpha_k(im)$, cet algorithme nous permet de réduire la capacité mémoire nécessaire pour générer un signal MLI, au lieu de mémoriser tout les angles exacts $\alpha_k(im)$ calculé par ordinateur pour $im \in [0,1.15]$, ce qui nécessite une grande capacité mémoire. Il suffit de trouver des formules qui nous donnent des valeurs approximatives, et réduit la capacité mémoire nécessaire pour générer le signal MLI. Dans la deuxième partie, on va simuler cet algorithme avec le logiciel de simulation MATLAB.

1. Approximation des angles exacts

On entend par 'exacte' les valeurs exactes des angles de commutation obtenu par la résolution du système non linéaire des α_k , et par 'approximé' les angles de commutations calculés en utilisant le nouvel algorithme 'on-line'. Les 3 figures de (Figure3.1, Figure3.2, Figure3.3) montre une courbe représentant des valeurs exactes des angles de commutations α_k en fonction de l'indice de modulation im , pour un indice $m=3, 5, 7$.

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

$$\alpha_s = \frac{2 \times 60^\circ}{(m+1)} \quad \text{Avec } m \text{ impaire} \quad (4.1)$$

Pour m fixé, les courbes de rang k impair présentent une pente négative et sont presque parallèles, dans la majeure partie de l'intervalle de variation $[0,1.15]$ de l'indice de modulation im , à l'exception des valeurs extrêmes de cet intervalle. De même les courbes de rang k pair présentent cette fois une pente positive et sont 'presque' parallèles dans les mêmes conditions de variation de l'indice de modulation que précédemment. Cette caractéristique

nous amène à approximer les courbes correspondant à k impair différemment de k pair. De plus la forme de ces courbes laisse penser que les fonctions qui approximent les courbes exactes devraient être la combinaison d'une fonction linéaire et d'une autre fonction non linéaire d'indice im et dont l'expression reste à déterminer.

L'idée donc consiste à approximer les valeurs exactes des angles de commutation en approximant chaque courbe en deux étapes :

- Dans la première étape, on approxime la partie entière de la valeur exacte de l'angle de commutation par une fonction linéaire d'indice im .
- Dans une seconde étape, on cherche à approximer soit la partie fractionnaire de la valeur exacte et la valeur approximative (trouvée dans la première étape).

La mise en service de ces deux étapes d'approximation nécessite la conception et la mise au point d'un programme d'approximation basé sur la méthode des moindres carrés (référence [2] paragraphe 8.2). Ce programme calcule les coefficients A_0, A_1, \dots, A_p ($p < n$) de la fonction d'approximation suivante :

$$\Phi(x) = A_0 \cdot \phi_0(x) + \dots + A_p \cdot \phi_p(x) \quad (4.2)$$

Permettant l'approximation de la fonction $f(x)$ connue empiriquement en $(n+1)$ points et qui prend les valeurs b_0, b_1, \dots, b_n aux abscisses a_0, a_1, \dots, a_n .

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

1.1 Cas k impair

Soit m fixé et soit α_k la valeur exacte de l'angle de commutation :

- **Première étape d'approximation de α_k :**

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

$$\Phi(im) = A_0 + A_1 \cdot im \quad (4.3)$$

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

$$A_0 = \alpha_k(0) = \alpha_s = (k+1) \frac{60^\circ}{(m+1)} \quad (4.4)$$

On suppose:
$$A_1 = A_1 \times \frac{60^\circ}{(m+1)}$$

On remplace A_0 et A_1 dans l'équation (5.3). On trouve la nouvelle fonction d'approximation :

$$\Phi(im) = \frac{60^\circ}{(m+1)}(k+1) + A_1 \times \left(\frac{60^\circ}{(m+1)}im\right) \quad (4.6)$$

Notons $[\alpha_k]$, la partie entière de α_k . Pour calculer A_1 , il faut mettre l'équation (4.6) sous la forme :

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

On cherche donc à approximer la différence suivante :

$$[\alpha_k] - \frac{60^\circ}{(m+1)}(k+1) \quad (4.7)$$

Par la fonction d'approximation suivante :

$$\Phi(im) = A_1 \times \left(\frac{60^\circ}{(m+1)} \times im\right) \quad (4.8)$$

La fonction d'approximation élémentaire est donc :

$$\phi_0(im) = \frac{60^\circ}{(m+1)}im \quad (4.9)$$

Le problème maintenant est de déterminer le coefficient A_1 pour les courbes représentant les valeurs exactes des angles de commutation en fonction de m et k impairs.

Pour chaque valeur de m et de k , on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.01.

Pour chaque valeur de l'indice im , on calculera la différence (4.7).

On a donc 115 données numériques à approximer en utilisant la fonction d'approximation (4.9). Par programmation, on peut faire un programme qui calcule A_1 pour chaque courbe définie par m . Le coefficient A_1 va donc nous permet de construire la fonction d'approximation (4.8) qui va approximer l'ensemble des 115 données numériques citées ci-dessus.

On répète le calcul en faisant varier m de 3 à 11 par valeurs impaires et pour chaque valeur de m on fait varier k de 1 à m par valeurs impaires.

Le coefficient A_1 est pratiquement et égale (-1) lorsque k varie de 1 à m , par valeur impairs, quelques soit la valeur de m . L'équation (4.6) devient :

$$\Phi(im) = \frac{60^\circ}{(m+1)}(k+1) + (-1) \times \frac{60^\circ}{(m+1)} \times im \quad (4.10)$$

- **Seconde étape d'approximation de α_k :**

Après approximation de la partie entière des l'angles 'exacts' de commutation , dans la première étape d'approximation , on va effectuer maintenant l'approximation de la différence $d(im)$ entre la valeur exacte et la valeur approximée, donnée par l'équation (4.10) , de la partie entière de l'angle de commutation . La différence $d(im)$ est définie par :

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

(4.11) Pour étudier la forme de $d(im)$, on va calculer $d(im)$ pour m égale à 7, par exemple, et pour k variant de 1 à 7 par valeur impaires en faisant varier im de 0.1 à 1.1 par pas de 0.01. La figure 4.2 trace la fonction $d(im)$ pour $k=1,3,5,7$.

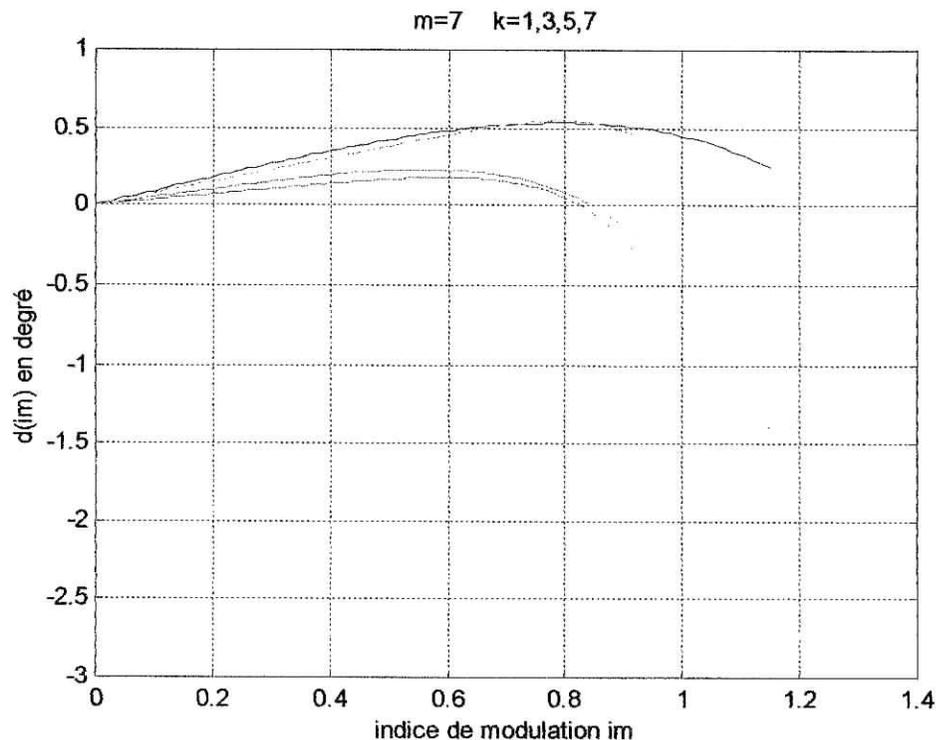


Figure 4.1 courbes représentant $d(im)$ en fonction de im , avec m égal à 7 et k variant de 1 à m .

Cependant en observant ces courbes, on constate qu'elles ont chacune une forme linéaire en début de l'intervalle, passent par un maximum puis décroissent rapidement en fin de l'intervalle.

Le problème maintenant est de chercher comment effectuer l'approximation des courbes de la figure 5.1 à l'aide de fonctions simples et avec une plus grande précision ?

Pour répondre à cette question, considérons les fonctions suivantes :

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

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

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

$$d(x) = f(x) - g(x) = (a_1 \cdot x - a_2 \cdot x^p) \quad (4.12)$$

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

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

Comparons les courbes des figures 4.1 et 4.2. Elles présentent une grande similitude entre elles. Il est tout à fait naturel d'utiliser une fonction de la forme $d(x)$, donnée par l'équation (4.12), pour faire l'approximation avec une bonne précision, de la fonction $d(im)$ donnée par l'équation (4.11).

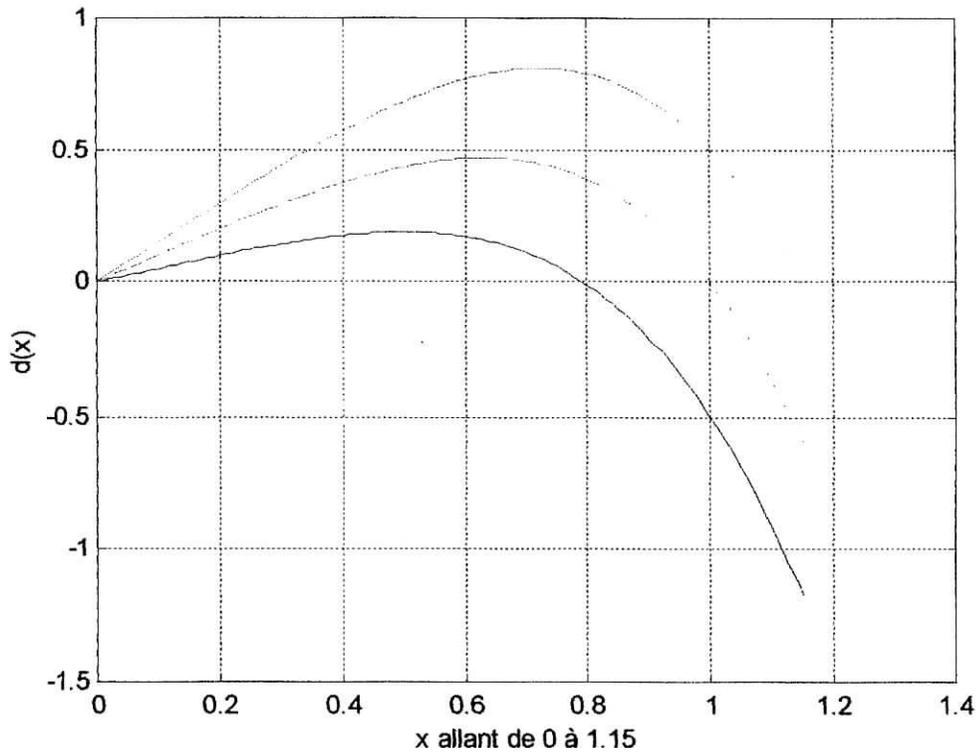


Figure 4.2 Courbes représentant la fonction $d(x)$ pour : $a_2=1, p=4$ et $a_1=0.5, 1, 1.5$

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

$$\Phi(im) = A_0 - A_1(im)^P \quad (4.13)$$

Les fonctions d'approximations élémentaires d'approximation sont donc :

$$\phi_0(im) = im \quad (4.14)$$

$$\phi_1(im) = -(im)^P \quad (4.15)$$

Le problème maintenant est de déterminer les trois coefficients A_0 , A_1 et P pour chacune des courbes représentant les valeurs exactes des angles de commutation en fonction de m et k impair. On fixe P et pour chaque valeur de m et k , on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.1. Pour chaque valeur de l'indice im , on calcule la différence (4.11). On a donc 11 données numériques à approximer par courbe et les fonctions élémentaires d'approximation sont (4.14) et (4.15). Le coefficient p est choisi de façon que l'erreur moyenne entre $d(im)$ et $\Phi(im)$ soit minimale. Ces coefficients vont nous permettre de

construire la fonction d'approximations (4.13) qui va approximer l'ensemble des 11 données numériques citées ci-dessus.

On répète le calcul en faisant varier m de 3 à 11 par valeurs impaires et pour chaque valeur de m , on fait varier k de 1 à m par valeurs impaires. Quelques résultats sont donnés, à titre d'exemple, dans le tableau 4.1.

m	k	P	A ₀	A ₁	emax	emoy
3	1	4	1.1659	1.3675	0.054	0.029
3	3	8	1.2320	2.2860	0.150	0.096
5	1	4	1.0514	0.7118	0.027	0.015
5	3	6	0.4619	1.4311	0.076	0.045
5	5	9	0.9653	1.1071	0.091	0.054
7	1	4	0.9042	0.4530	0.018	0.009
7	3	5	0.5112	1.0020	0.045	0.023
7	5	7	0.3514	1.1006	0.063	0.040
7	7	10	0.7667	0.5854	0.054	0.033

Tableau 4.1 Exemple de coefficients A₀, A₁, P de l'équation (4.16) pour k impair et m égal à 3, 5 et 7

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

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

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

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

$$\alpha_k(im) \cong \frac{60^\circ}{(m+1)} (k+1) + \left[A_0 - \frac{60^\circ}{(m+1)} \right] \times im - A_1 \times (im)^p \quad (4.16)$$

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

1.2 Cas k pair

De l'équation (4.16) on déduit la forme générale de **l'équation générale d'approximation des angles de commutation 'exact's** pour k pair :

$$\alpha_k(im) = \frac{60^\circ}{(m+1)} \times k + A_0 \times im - A_1 \times im^p \quad (4.17)$$

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

De l'équation (5.17), on déduit la différence :

$$\alpha_k(im) - \frac{60^\circ}{(m+1)} \times (k+1) \cong A_0 \times im - A_1 \times im^p \quad (4.18)$$

Les fonctions élémentaires d'approximation sont donc :

$$\phi_0(im) = im \quad (4.19)$$

$$\phi_1(im) = -(im)^p \quad (4.20)$$

Le problème maintenant est de déterminer les trois coefficients A_0 , A_1 et P pour chacune des courbes représentant les valeurs exactes des angles de commutations en fonction de m et k pair. On fixe p et pour chaque valeur de m et k , on fait varier l'indice de modulation im de 0.1 à 1.1 par pas de 0.1. Pour chaque valeur de l'indice im , on calcule la différence (4.18). On a donc 11 données numériques par courbe à approximer. Le coefficient p est choisi de façon que l'erreur moyenne entre $d(im)$ et $\Phi(im)$ soit minimale. Ces coefficients vont nous permettre de construire la fonction d'approximations (4.13) qui va approximer l'ensemble des 11 données numériques citées ci-dessus.

On répète le calcul en faisant varier m de 3 à 11 par valeurs impaires et pour chaque valeur de m , on fait varier k de 1 à m par valeurs paires.

Les résultats du tableau 4.2 montrent que l'erreur moyenne, notée $emoy$, est de l'ordre de centième de degré ($1/100^\circ$). Donc, la technique d'approximation utilisée donne une plus grande précision de calcul par rapport aux algorithmes MLI 'on line' connus aussi bien pour les angles pairs qu'impairs.

Le calcul des coefficients A_0 , A_1 et p des équations (4.16) et (4.17) est effectué par un programme basé sur la méthode des moindres carrés, quel que soit la valeur de m .

m	k	p	A_0	A_1	emax	emoy
3	2	10	9.0872	1.5766	0.093	0.038
5	2	6	4.4472	1.3392	0.084	0.026
5	4	11	7.1389	0.7694	0.062	0.027
7	2	6	2.5970	0.8099	0.028	0.017
7	4	7	4.3641	1.0538	0.074	0.028
7	6	12	5.6821	0.4038	0.037	0.018
9	2	6	1.7134	0.5410	0.020	0.012
9	4	6	2.9615	0.8662	0.051	0.020
9	6	8	3.8996	0.7525	0.051	0.026
9	8	12	4.6968	0.2573	0.024	0.012

Tableau 4.2 Exemple de coefficients A_0, A_1 , p de l'équation (4.17) pour k pair et m égal à 3, 5, 7 et 9

2. Précision de l'algorithme

2.1 Angles calculés

Le calcul de l'erreur en valeur absolue, en degré, entre la valeur exacte de l'angle de commutation et sa valeur approximée par l'algorithme on-line pour m égal à 5 et 15, dans tout l'intervalle de variation [0, 1.15] de l'indice im nous montre que l'erreur moyenne est de l'ordre de trois centièmes (3/100) de degré pour m égal à 5 et qu'elle diminue lorsque m augmente et atteint un centième (1/100) de degré pour m égal à 15.

2.2 Calcul des harmoniques :

Pour déterminer de façon précise, les performances de l'algorithme, on calcule les amplitudes du fondamental et des harmoniques 'éliminés' en utilisant les angles approximatés par l'algorithme, pour différentes valeurs de m et pour ω variant dans tout l'intervalle $[0,1.15]$. Ensuite on représente l'erreur entre la valeur exacte et la valeur approximatée de l'amplitude de l'harmonique de rang r , pour m égal à 5 et 15.

L'erreur moyenne est de l'ordre 0.002%.

Pour m égal à 5, le taux d'harmonique est inférieur à 0.3% en début de l'intervalle. Il décroît et prend la valeur moyenne 0.05% dans le sous intervalle $[0.6,1.1]$. En fin il s'accroît à partir de la valeur 1.1.

Pour m égal à 15, le taux d'harmoniques est inférieur à 0.16% en début de l'intervalle. Il décroît et prend la valeur moyenne 0.03% dans le sous intervalle $[0.55,1.1]$. En fin il s'accroît à partir de la valeur 1.1.

En conclusion, le taux d'harmoniques décroît de 0.3% à 0.03% lorsque m croit de 5 à 15.

3- Simulation du nouvel algorithme 'on-line'

3.1 Introduction

Dans cette seconde partie, nous allons simuler le nouvel algorithme 'on-line' avec le logiciel de simulation MATLAB. On va calculer en premier lieu les angles approximatés puis générer le signal MLI. Ceux-ci sont utilisés dans la génération d'une tension de commande d'un onduleur de tension MLI alimentant un moteur asynchrone triphasé.

Cette technique présente de nombreux avantages tels que la variation de la tension du fondamental et l'élimination des $(m-1)$ premiers harmoniques. En se propose donc de vérifier les résultats théoriques obtenus avec le nouvel algorithme en effectuant la simulation d'une commande de vitesse d'un moteur asynchrone triphasé alimenté par une tension MLI calculée par le nouvel algorithme proposé. Cette simulation nous permettra également d'évaluer les performances de cet algorithme supposé être très précis et ayant temps de calcul réduit, donnant ainsi la possibilité d'une implémentation d'une commande par élimination des harmoniques sur un système à base de microprocesseur.

3.2 Générateur d'onde MLI programmée

L'indice i_m correspond à la valeur normalisée du fondamental de la tension MLI. Pour obtenir la tension réelle correspondante, on doit multiplier l'indice i_m par la tension nominale simple V_n (phase - neutre), égal à 127V.

Les formules (4.16) et (4.17) nous donnent les angles de commutations approximées en degrés, donc on doit les convertir en instants de commutations calculés en secondes ou en milli secondes.

3.3 Calcul des instants de commutation sur une période (déphasage 0°)

On se propose de générer une tension MLI simple, sur une période, pour commander une phase de l'onduleur de tension MLI. Cette tension MLI prend alternativement les valeurs +127V et -127V.

Le programme `mli.m` calcule les angles de commutations α_k , $k=1, \dots, m$ et i_m variant de 0 à 1.15 par pas de 0.01. Ces angles de commutations correspondent aux instants de commutation du premier quart d'onde de la tension MLI.

Donc, pour une demi période de la tension MLI, on aura $2m$ commutations auxquelles il faut ajouter une commutation à chaque demi période, soit en tout $(2m+1)$.

On a donc sur une période $2(2m+1)$, soit $4m+2$. En ajoutant une commutation à l'origine, on obtint finalement $(4m+3)$ commutations pour la première période et $4m+2$ commutations pour les périodes suivantes de la tension MLI.

Les angles de commutations étant calculés, par le programme `mli.m`, sur un quart d'onde de la période, il faut donc déduire leurs valeurs pour les trois quarts d'ondes restantes. Autrement dit, il faut calculer les angles de commutation sur tout la période.

Calculons les angles de commutation $\alpha(i)$ sur une période, donc pour i variant de 1 à $(4m+2)$.

L'angle de commutation à l'origine est nul.

Sur la première demi période :

Les angles $\alpha(i)$ sont calculés par les formules (4.16) et (4.17)

Pour i variant de $(m+1)$ à $2m$:

$$\alpha(i) = 180 - \alpha(j) \quad j \text{ décrois de } m \text{ à } 1 \text{ par pas de } 1$$

Pour i égal à $2m+1$:

$$\alpha (i)=180^\circ$$

Sur la second demi période :

Pour i variant de (2m+2) à (4m+2) :

$$\alpha (i)=180+\alpha (j) \quad \text{avec } j=1,\dots,2m+1.$$

Les angles de commutation sur une période quelconque de la tension MLI sont :

$$\alpha (1),\alpha (2),\dots,\alpha (4m+2).$$

Les angles de commutation sur la première période sont :

$$0, \alpha (1),\dots,\alpha (4m+2).$$

Les valeurs des angles sont calculées en degrés par le programme. Il faut les convertir en radians en utilisant l'équation suivante :

$$\alpha (i)[\text{rad}]=\alpha (i)^\circ * \pi/180 [\text{rad}] \quad i=1,\dots,4m+2$$

Maintenant, on doit les convertir en valeurs temporelles T(i) pour une fréquence déterminée.

3.4 Calcule des instants de commutation T (i) sur une période

On se propose de calculer les valeurs temporelles T (i) des angles de commutation.

$$\alpha(i) = (2 \times \pi \times f \times T(i)) \times 180 / \pi \quad \text{avec } \alpha \text{ en degrés}$$

donc
$$\alpha(i) = 360 \times f \times T(i)$$

$$T(i) = \alpha(i) / (360 \times f) \tag{4.21}$$

Pour avoir le rapport V/f constant, on prend:

$$V/V_n = i_m$$

D'où
$$V/f = V_n/f_n = (V_n * i_m) / (f_n * i_m) = \text{cte}$$

Avec
$$f_n = 50 \text{ Hz.}$$

On déduit que:

$$f = V * (f_n / V_n) = (i_m * V_n) * (f_n / V_n) = i_m * f_n$$

L'équation (6.1) devient alors:

$$T(i) = \alpha(i) / (360 \times i_m \times f_n) \tag{4.22}$$

On remplace la valeur de f_n dans l'équation, on trouve la valeur des instants de commutation en secondes et uniquement en fonction de l'indice de modulation i_m et des coefficients A₀, A₁ et P :

Avec T (i) en secondes, f_n égal à 50 Hz et i_m l'indice de modulation.

Donc, les instants de commutations sur une période sont donnés par l'équation (4.22) à T seconds près, T étant la période temporelle.

3.5 Calcul de la valeur $S(i)$ de la tension MLI à l'instant $T(i)$

Si i impair alors $S(i)$ est égal à $-V_n$. On a alors une tension négative de la tension MLI.

Si i pair alors $S(i)$ est égal à $+V_n$. On a alors une tension positive de la tension MLI.

On obtient alors 2 vecteurs, un vecteur décrit les angles de commutations et l'autre décrit les tensions de commutations. Le premier décrit les valeurs temporelles, en milli secondes des instants de commutations, le seconde décrit les valeurs de la tension, en volt, mais on va prendre un signal unité variant entre 1 et -1, mais dans la réalité ce signal est multiplié par la tension nominale d'une phase V_n .

Les figures ci-dessous donnent les résultats de simulation du nouvel algorithme pour différents indices de modulations i_m compris entre 0 et 1.15 variant par de 0.01, aussi pour différentes valeurs de m variant de 1 à une valeur indéterminé et prenant des valeurs impaires.

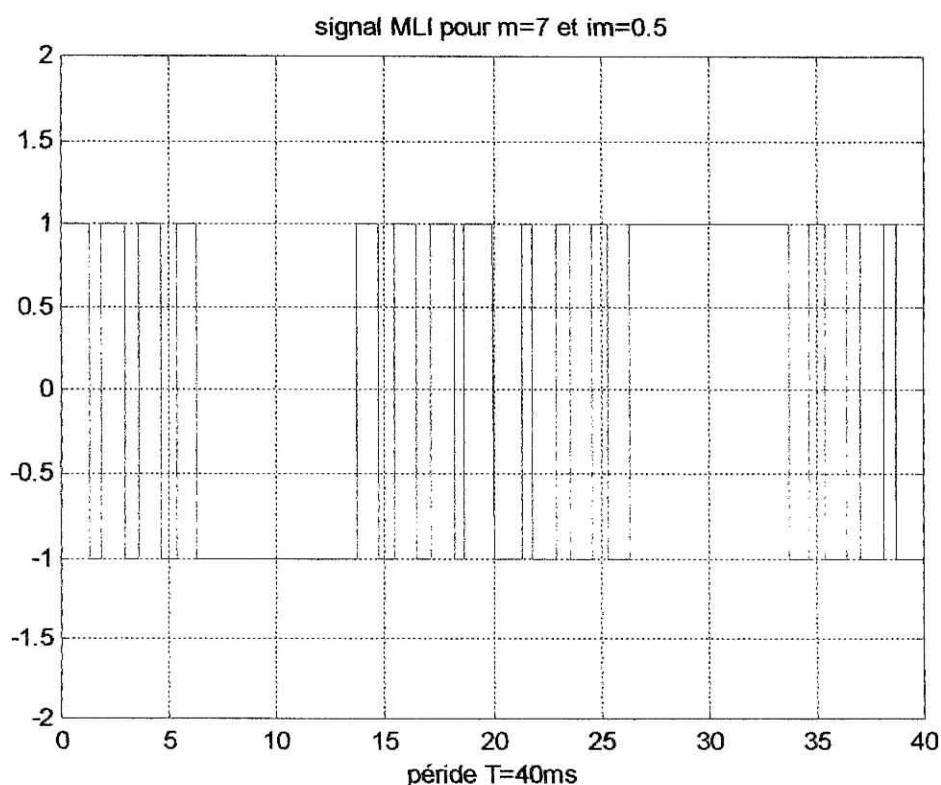


Figure4.3 Le signal MLI pour $m=7$ et $i_m=0.5$

La figure ci dessus trace le signal MLI pour $m=7$ et $im=0.5$, ce qui nous donne une période de 40ms.

On remarque que les angles de commutations dans le premier quart d'onde sont distantes avec des distances presque égales, ce qui est logique, car si on regarde la figure 3.3 qui trace les courbes exactes des angles de commutations pour $m=7$, on remarque que pour $im=0.5$ les distances entre les courbes sont presque égales, ce qui correspond aux angles de commutations pour cette valeur de l'indice de modulation im .

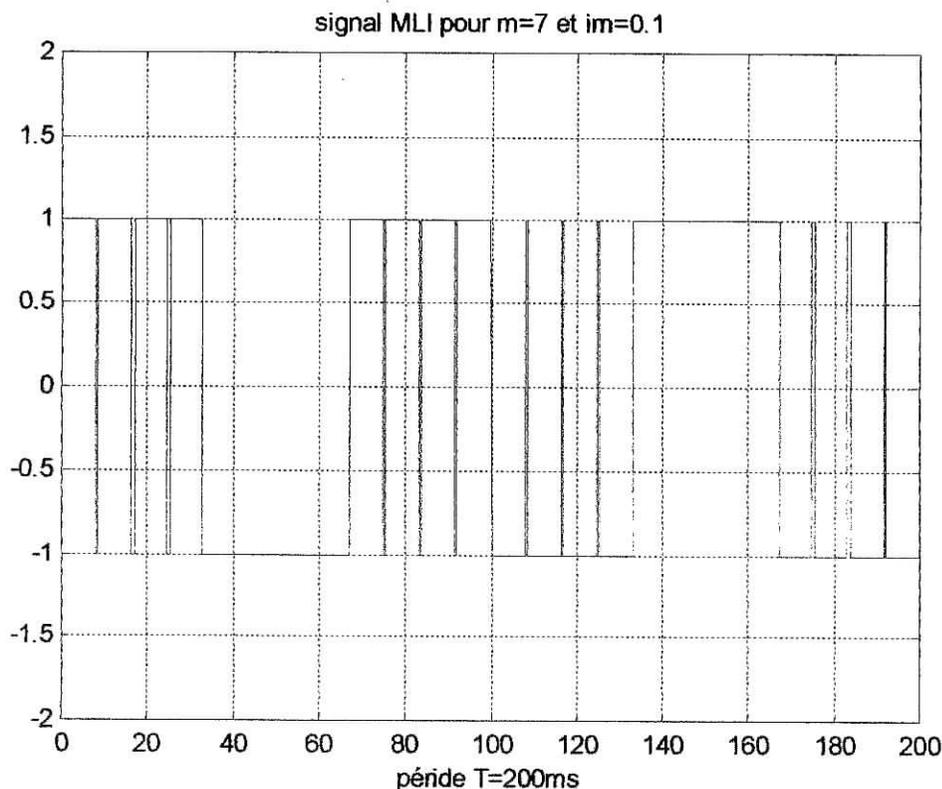


Figure4.4 Signal MLI pour $m=7$ et $im=0.1$

La figure ci-dessus trace le signal MLI pour $m=7$ et un indice de modulation $im=0.1$, ce qui nous donne une période de 200 ms.

On remarque que les angles de commutation adjacents impairs et pairs avec k croissant dans cette ordre $(\alpha_1, \alpha_2), (\alpha_3, \alpha_4), \dots$, ont presque la même valeur, ce qui explique que le signal MLI prend un rapport cyclique élevé dans le premier quart d'onde. Les autres quart d'ondes sont déduits directement par symétrie quart d'onde. Pour le deuxième quart d'onde, la symétrie est par rapport à $T/4$, et pour la deuxième demi période, la symétrie est par rapport à $T/2$.

Ce résultat peut être justifié par l'allure des courbes des angles exacts. Si on trace une droite verticale d'équation $x=0.1$, cette droite coupe les courbes des angles de commutations exacts pour un indice de modulation $im=0.1$, et on remarque que les angles de commutations adjacents impaires et paires sont très proches l'une de l'autre, d'où l'allure du signal MLI précédent.

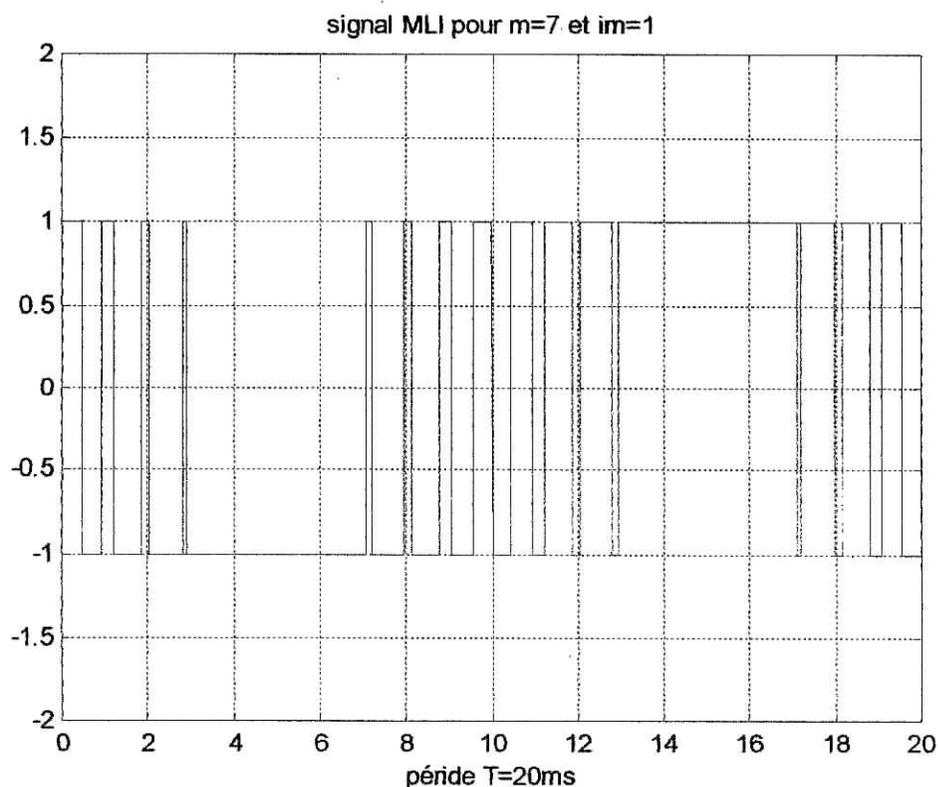


Figure4.5 Signal MLI pour $m=7$ et $im=1$

La figure ci-dessus trace un signal MLI pour $m=7$ et un indice de modulation $im=1$, ce qui nous donne une période de 20 ms.

On remarque les angles adjacents pairs et impairs dans cette ordre $(\alpha_2, \alpha_3), (\alpha_4, \alpha_5), \dots$ sont proches l'un de l'autre, ceci aussi peut être expliqué en analysant les courbes des angles de commutations exacts. On remarque que les courbes qui génèrent ces angles rapprochent de plus en plus quand im augmente.

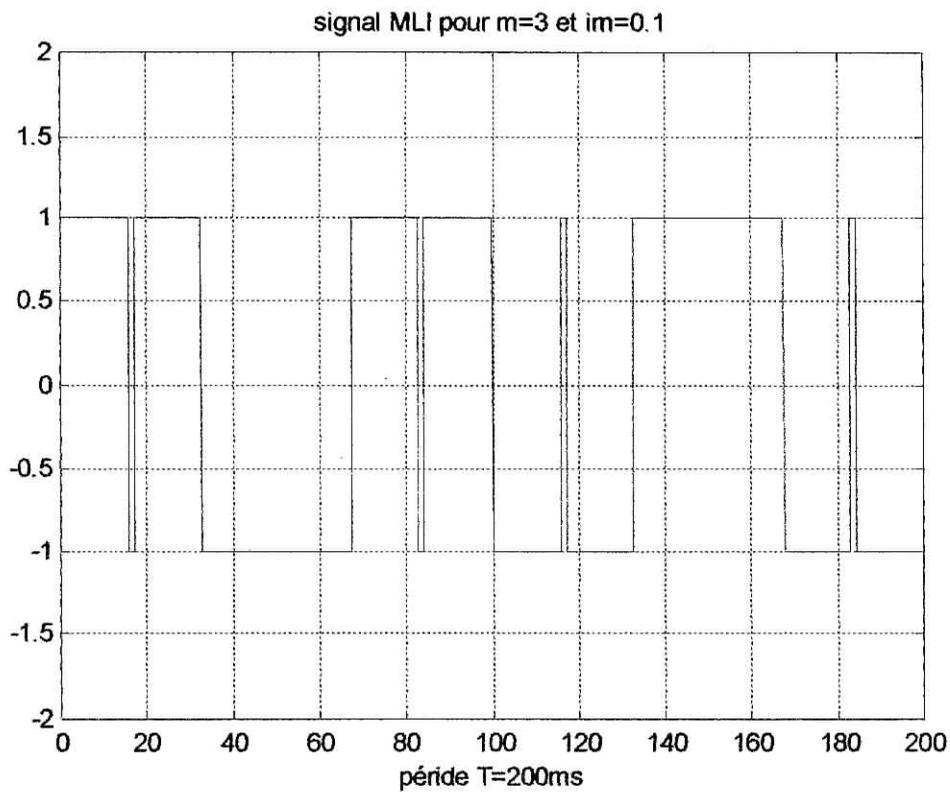


Figure4.6 Signal MLI pour $m=3$ et $im=0.1$

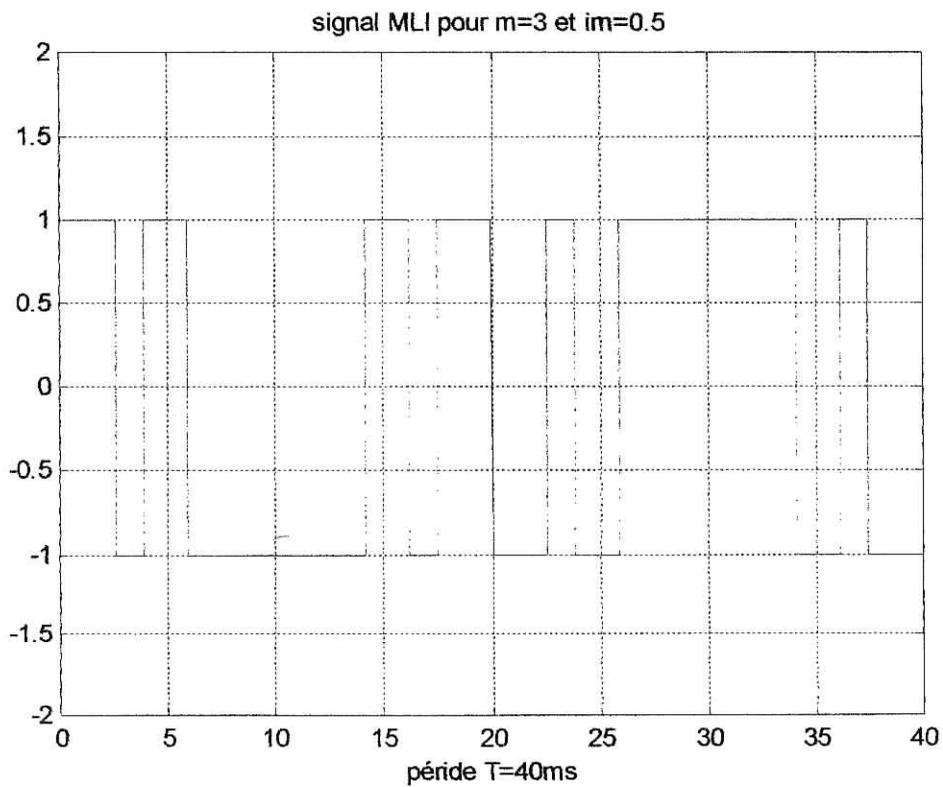


Figure4.7 Signal MLI pour $m=3$ et $im=0.5$

Les mêmes remarques faites pour les figures tracées pour $m=7$ peuvent être refaites pour les courbes ci-dessus, tracer pour $m=3$ et $im=0.1$ et 0.5 , c'est-à-dire que les angles impairs et pairs adjacents sont très proches l'un de l'autre pour les valeurs de m au voisinage de 0, donc un rapport cyclique élevé pour le premier quart d'onde, et les angles adjacents pairs et impairs dans cet ordre sont très proches l'un de l'autre pour les valeurs de l'indice de modulation voisin de 1.15, ce qui correspond à un rapport cyclique très petit.

Les remarques sur les distances entre les instants de commutation adjacents sont très importantes par la suite pour la réalisation d'un système à base d'un microprocesseur, car si le temps entre deux instants de commutation est très petit, ce temps sera proche des limites d'exécution du programme de génération du signal MLI implémenté dans un microprocesseur. Donc si nous ne disposent pas d'un microprocesseur très rapide, on peut prévoir des problèmes d'exécution du programme dans le système à base de microprocesseur.

4. Conclusion

Dans ce chapitre, nous avons proposé un nouvel algorithme qui calcule, en temps réel, les angles de commutation pour la génération d'une tension MLI avec asservissement du fondamental et élimination d'harmoniques.

Cet algorithme nous a permis de palier aux inconvénients de la méthode de Patel et Hoft, qui nécessite un temps de calcul élevé et une grande capacité mémoire, ce qui la rend une méthode loin d'une application temps réel (c'est une méthode 'off-line').

Pour chaque rang K des angles de commutations α_k , le calcul des angles nécessite de stocker trois coefficients A_0 , A_1 et p , ce qui réduit la capacité mémoire d'une manière considérable.

Chapitre 5

Application à un système à base d'un PIC16F876

Dans ce chapitre, on se propose d'implémenter le nouvel algorithme proposé dans le chapitre 5 à un système à base d'un microcontrôleur PIC 16F876, pour la génération d'un signal MLI pour commander un bras d'un onduleur triphasé décrit dans le chapitre 3, et évaluer ainsi les performances de l'algorithme proposé. Les autres bras seront commandés par le même signal retardé de $T/3$ pour le deuxième bras, et de $2T/3$ pour le troisième bras. Donc, la commande complète de l'onduleur triphasé nécessite des circuits de retards en plus de la carte qu'on va réaliser.

1. Le PIC 16F876

1.1 Qu'est-ce qu'un PIC ?

Un PIC n'est rien d'autre qu'un microcontrôleur, c'est à dire une unité de traitement d'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants externes.

Les PICs sont des composants dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instructions réduit, car plus on réduit le nombre d'instructions, plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne.

Tous les PICs Mid-Range ont un jeu de 35 instructions, stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en 1 cycle. On atteint donc de très grandes vitesses, et les instructions sont de plus très rapidement assimilées. L'exécution en un seul cycle est typique des composants RISC.

1.2 Les différentes familles des PICs

La famille des PICs est subdivisée à l'heure actuelle en 3 grandes familles : La famille Base-Line, qui utilise des mots d'instructions de 12 bits, la famille Mid-Range, qui utilise des mots de

14 bits (et dont font partie les 16F84 et 16F876), et la famille High-End, qui utilise des mots de 16 bits.

1.3 Le PIC16F876

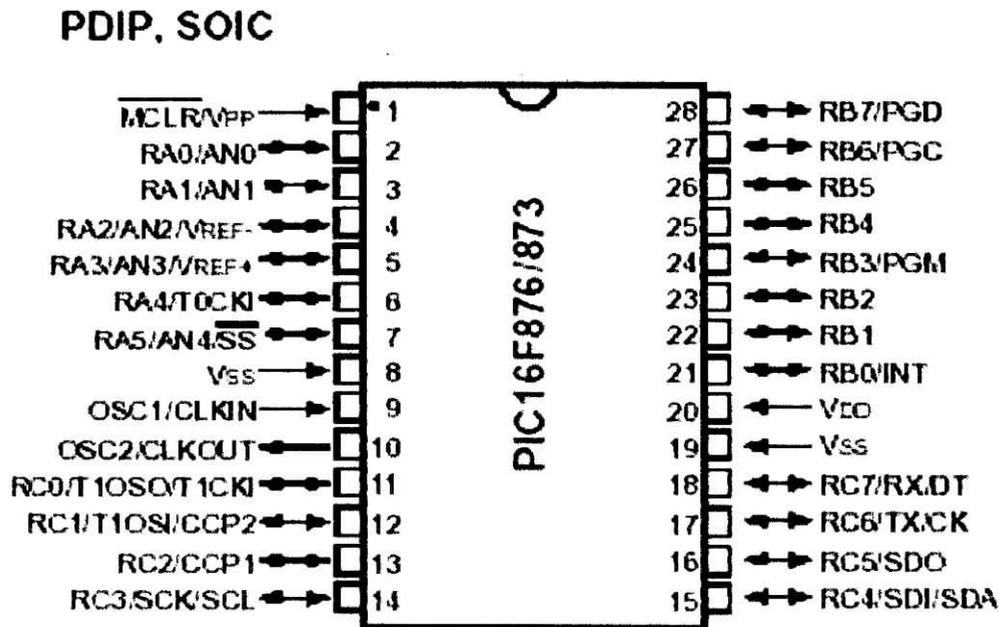


Figure5.1 Le brochage du PIC16F876

Le 16F876 fait partie de la sous-famille des 16F87x. Cette branche fait partie intégrante de la grande famille des PICs Mid-Range. Le PIC16F876 à 8k de mémoire programme qui est une mémoire flash, 368 octets de RAM, 256 octets eeprom, 22 E/S, 5 entrées A/D et un port série USART/MSSP. Le numéro du composant peut être suivi d'un «A» et d'un «XX» pour indiquer la fréquence d'horloge maximale supportée par le PIC.

1.3.1 La mémoire programme

La mémoire programme est constituée de 8K mots de 14 bits. En effet, il faut 2 octets pour coder 14 bits. Ceci explique également pourquoi, on a 0x3FFF dans un PIC vierge. Cela donne en

binaire B'11111111111111', soit 14 bits. Une instruction est codée sur 1 mot. Donc, 8K donne 8 milles d'instructions possibles.

1.3.2 La RAM

La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant, c'est dans la RAM que nos données seront stockées après apprêt leurs calculs, donc les angles de commutation seront perdus lors de coupure de l'alimentation.

La mémoire RAM disponible du 16F876 est de 368 octets. Elle est répartie de la manière suivante :

- 1) 80 octets en banque 0, adresses 0x20 à 0x6F
- 2) 80 octets en banque 1, adresses 0xA0 à 0XEF
- 3) 96 octets en banque 2, adresses 0x110 à 0x16F
- 4) 96 octets en banque 3, adresses 0x190 à 0x1EF
- 5) 16 octets communs aux 4 banques, soit 0x70 à 0x7F = 0xF0 à 0xFF = 0x170 à 0x17F = 0x1F0 à 0x1FF.

C'est dans la RAM qu'on trouve les REGISTRES SPECIAUX et cases mémoires « libres » dont nous allons utiliser pour nos variables (les instants de commutations). La figure5.2 montre tous les registres spéciaux utilisés dans le PIC.

Chaque registre provoque un fonctionnement spécial du PIC ou la mise en service d'une fonction particulière. Certains registres sont identiques dans les 2 banques (FSR par exemple). Cela signifie qu'y accéder depuis la banque 0 ou 1 ne fait pas de différence.

La banque 0 utilise les adresses de 0x00 à 0x7F, la banque 1 allant de 0x80 à 0xFF. L'emplacement 0x00 est un emplacement auquel on ne peut pas accéder. Pour la grande majorité des registres, chaque bit a une fonction spéciale.

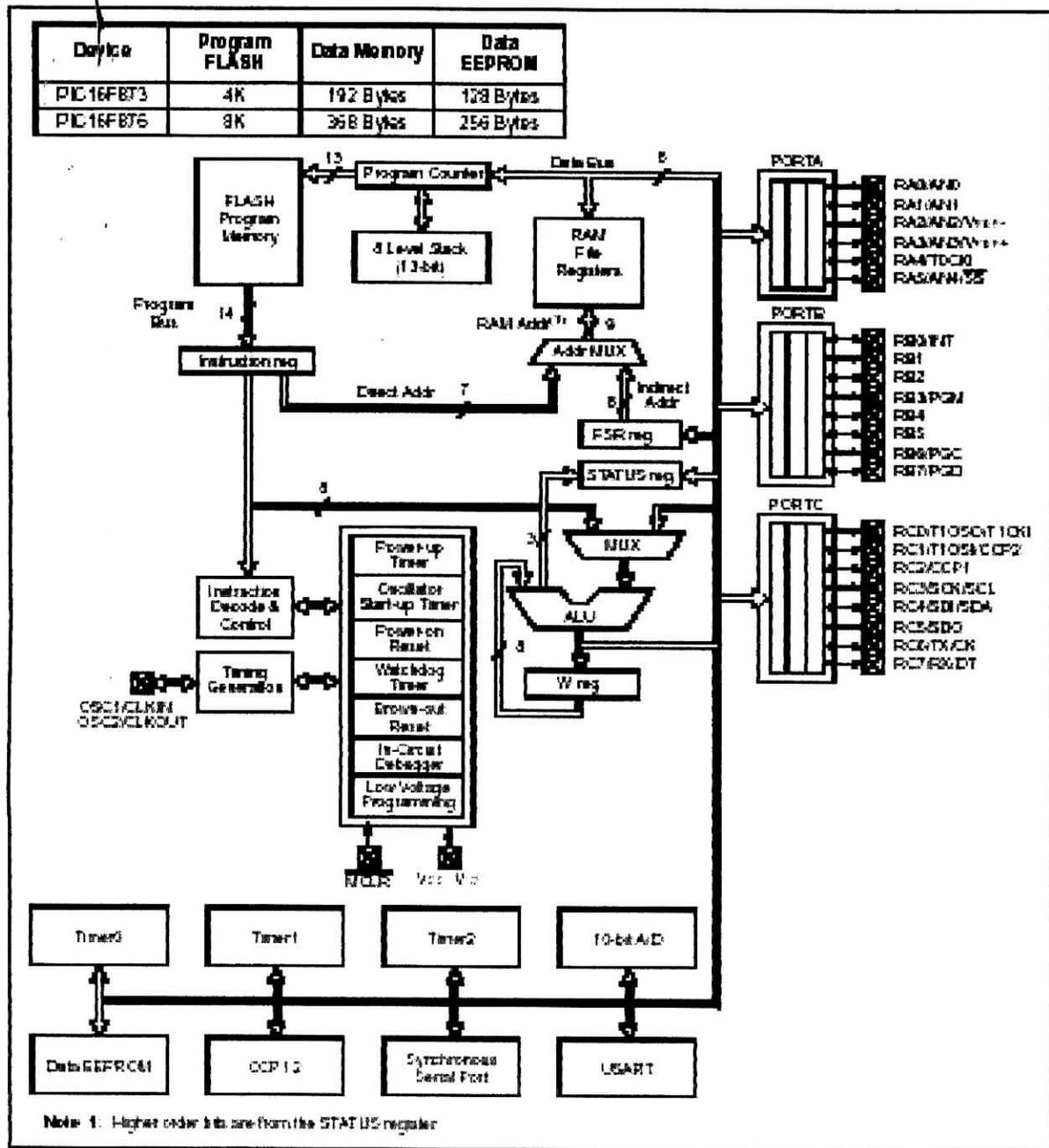


Figure 5.2 Structure interne du PIC16F876

2. Réalisation de la carte pour la génération du signal MLI

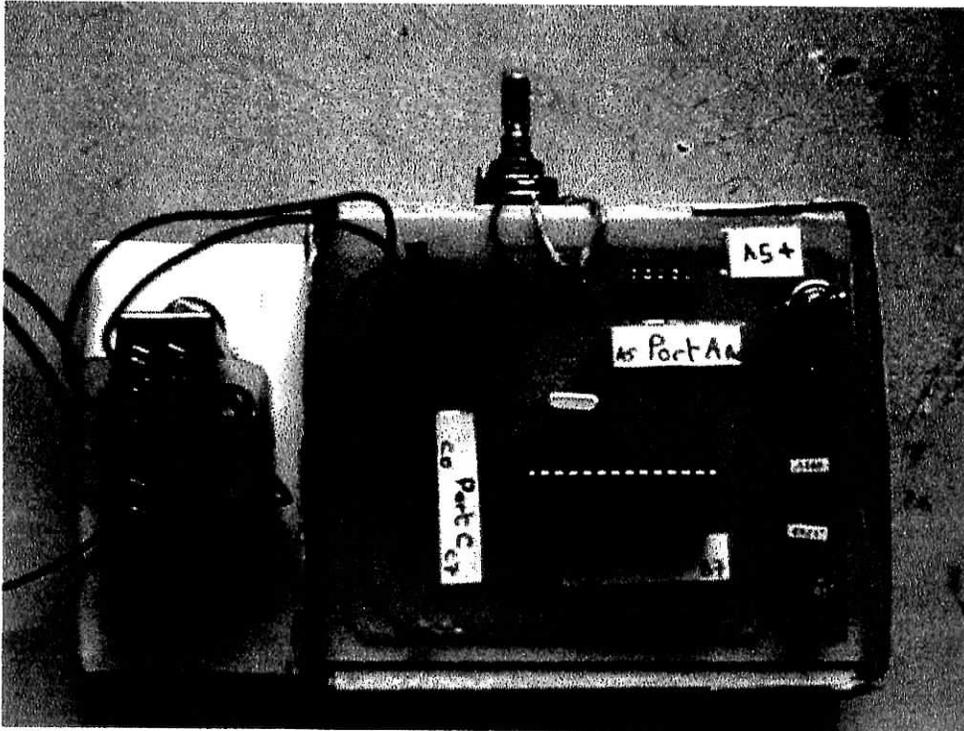


Figure5.3 Image de la réalisation

La réalisation de la carte électronique (figure ci-dessus) contient le module PIC 16F876 et une alimentation stabilisée nécessaire pour le fonctionnement du PIC.

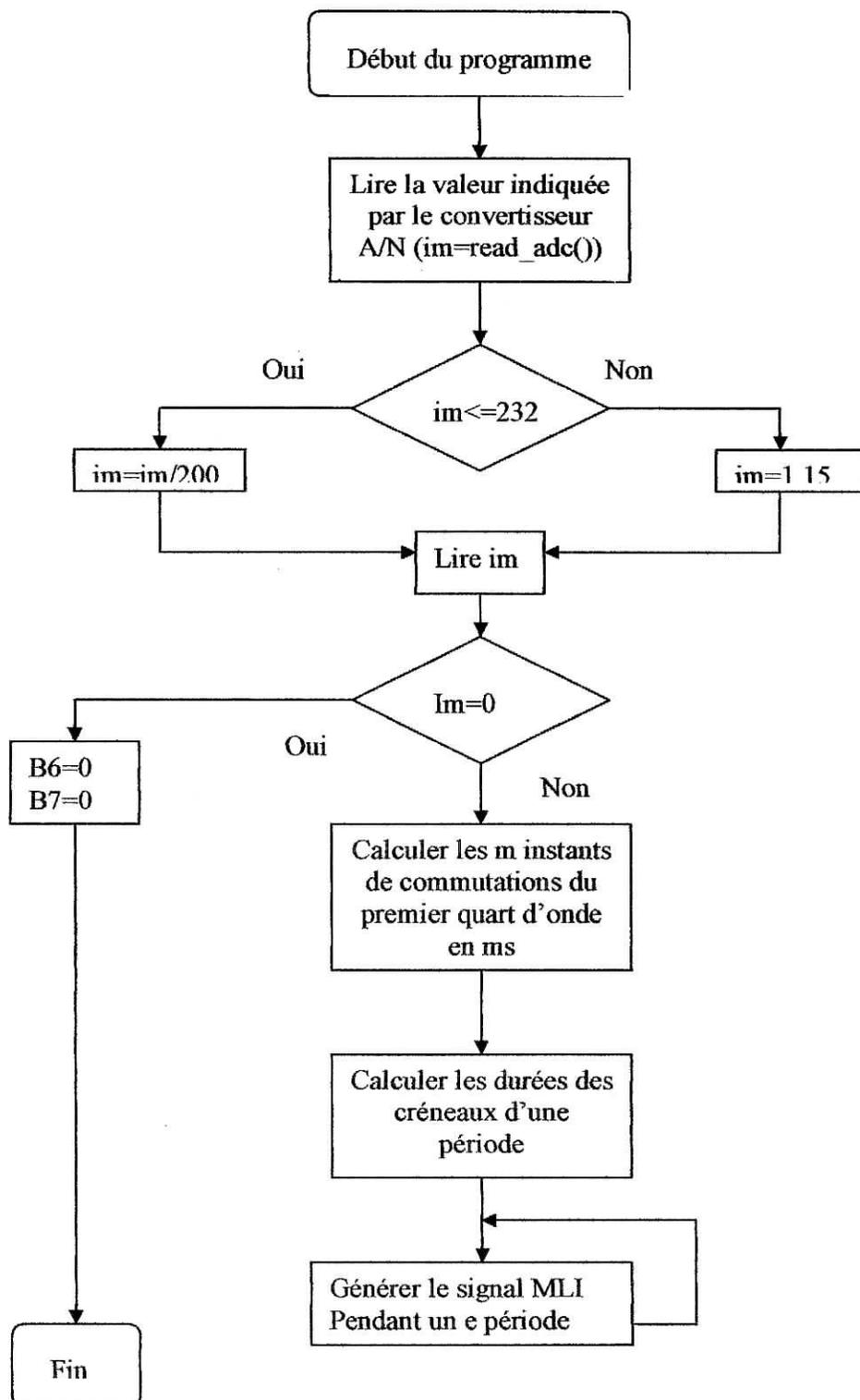
Le module PIC contient le PIC lui-même et un quartz de 20M, 2 condensateurs de 15pF utilisé pour le filtrage entre le quartz et la masse.

La broche A0 du portA est reliée à un potentiomètre qui nous permet de varier la tension d'entrée, c'est avec le potentiomètre qu'on va varier la vitesse du moteur. Les sorties des deux signaux MLI seront par les pins B6 et B7.

Un interrupteur est utilisé dans ce montage entre l'alimentation et le PIC.

Les ports sont tous tirés avec des fils vers des connecteurs pour nous permettre toute sorte d'extension (circuit de retard, afficheur LCD, boot loader, débogage,).

Cette réalisation se satisfait du minimum pour le fonctionnement du PIC et permet toute sorte d'extension future.

**Figure5.5 Organigramme du programme de génération du signal MLI**

La demande de la vitesse se fait par une tension variable à la sortie d'un potentiomètre R1 qui est monté en diviseur de tension, cette tension sera convertie par le convertisseur analogique/numérique qui est de 10 bits, mais dans notre application il sera forcé à 8 bits. Cette valeur nous permet de trouver la valeur de l'indice de modulation, ceci en incrémentant la valeur de im de 0 à 1.15 par pas de 0.01 qui correspond à une incrémentation de 2 dans le convertisseur. Après avoir calculer les valeurs des angles de commutations, puis une fois que le calcul des angles est terminer, le programme principal va les appeler pour générer les signaux MLI (PWM) qui vont commander la commutation des transistors de l'onduleur à travers les deux pins du portB (B6, B7).

Comme nous alimentons ce potentiomètre R1 à partir de Vdd et de Vss, la tension minimale présente sur le curseur sera de Vss, tandis que la tension maximale sera de Vdd. Nous devons donc choisir un mode de fonctionnement comportant 1 pin configuré en entrée analogique, et aucune tension de référence externe (1/x/0).

PCFG 3 à 0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	Vref-	Vref+	A/D:R
0000	A	A	A	A	A	Vss	Vdd	5.0.0
0001	A	Vref-	A	A	A	Vss	RA3	4.0.1
0010	A	A	A	A	A	Vss	Vdd	5.0.0
0011	A	Vref-	A	A	A	Vss	RA5	4.0.1
0100	D	A	D	A	A	Vss	Vdd	5.2.0
0101	D	Vref-	D	A	A	Vss	RA3	2.2.1
0110	D	D	D	D	D	-	-	0.5.0
0111	D	D	D	D	D	-	-	0.5.0
1000	A	Vref-	Vref-	A	A	RA2	RA3	3.0.2
1001	A	A	A	A	A	Vss	Vdd	5.0.0
1010	A	Vref-	A	A	A	Vss	RA3	4.0.1
1011	A	Vref-	Vref-	A	A	RA2	RA3	3.0.2
1100	A	Vref-	Vref-	A	A	RA2	RA3	3.0.2
1101	D	Vref-	Vref-	A	A	RA2	RA3	2.1.2
1110	D	D	D	D	A	Vss	Vdd	1.4.0
1111	D	Vref-	Vref-	D	A	RA2	RA3	1.2.2

Tableau5.1 Différents configurations du convertisseur A/N

Le tableau ci-dessus nous donne les possibilités de combinaison du convertisseur pour le 16F876, pour notre cas, la case avant dernière nous vérifie la configuration qu'on veut, c'est la case (1/4/0).

Une fois que la valeur de l'indice de modulation a été calculée, le programme va calculer les instants de commutations en utilisant les formules (4.16) et (4.17). Après avoir été converties en milli secondes en utilisant la formule (4.22) et en les multipliant par 1000, les formules (4.16) et (4.17) deviennent :

Pour k impaire :

$$T_k(im) \cong \left(\frac{60^\circ}{(m+1)} (k+1) + \left[A_0 - \frac{60^\circ}{(m+1)} \right] \times im - A_1 \times (im)^p \right) \times \frac{1000}{360 \times 50 \times im}$$

D'où la formule finale

$$T_k(im) \cong \left(\frac{60^\circ}{(m+1)} (k+1) + \left[A_0 - \frac{60^\circ}{(m+1)} \right] \times im - A_1 \times (im)^p \right) \times \frac{1}{18 \times im} \quad (5.1)$$

De même pour k pair, on trouve :

$$T_k(im) = \left(\frac{60^\circ}{(m+1)} \times k + A_0 \times im - A_1 \times im^p \right) \times \frac{1}{18 \times im} \quad (5.2)$$

On obtient un vecteur réel de dimension m avec T_k en milli-secondes.

Une fois les instants de commutations sont calculés, et pour des raisons d'optimisation de la RAM, on calcule un vecteur entier qui nous donne la largeur de chaque impulsion pour une période en milli-seconde avec une précision de 0.5 ms.

Une fois ce vecteur a été calculé, le programme exécute une boucle infinie pour générer le signal MLI utilisant le vecteur précédent. Uniquement l'interruption de fin de conversion peut interrompre cette boucle infinie. Elle permet de recalculer les angles de commutations pour la vitesse demandée.

4. Simulation du programme

The screenshot displays the MPLAB IDE interface. On the left, a window shows C code for a PIC16F876 simulation. The code includes a delay function and a main function that sets up variables and a table. On the right, a window titled 'Special Function Register' shows a list of SFRs with their names, hexadecimal values, decimal values, and binary representations.

```
#include <16F876.h>
#device *=16
#use delay(clock=2000000)
#fuses HS,NOWDT

float mlu (int p, float im);

void main() {
    int i,p;
    int angle[16],time[30];
    float imp,im,A0,A1;
    int t[7][3] = {
        {904, 453, 4000},
        {2600, 810, 6000},
        {511, 1002, 5000},
        {4364, 1054, 7000},
        {3514, 1100, 7000}
    };
}
```

SFR Name	Hex	Dec	Binary
w	06	6	000001
tmr0	40	75	010010
option_reg	C0	192	110000
pcl	E6	230	111001
pclath	00	0	000000
status	1C	28	000111
fsr	A2	162	101000
porta	00	0	000000
trisa	3F	63	001111
portb	80	128	100000
trisb	3F	63	001111
portc	00	0	000000
trisc	07	215	110101
intcon	04	4	000001
pir1	00	0	000000
pie1	00	0	000000
pir2	00	0	000000
pie2	00	0	000000
tmr1l	00	0	000000
tmr1h	00	0	000000
t1con	00	0	000000
tmr2	00	0	000000
pr2	00	0	000000
t2con	00	0	000000
sspbuif	00	0	000000
sspcon	00	0	000000
sspcon2	00	0	000000

Below the SFR window, a memory dump shows the execution of instructions:

```
1242 04D9 01F9      clrF  0x7
1243 04DA 01FA      clrF  0x7
1244 04DB 0000      nop
1245 04DC 3400      retlw 0x0
1246 04DD 30A2 delay_ movlw 0xA
1247 04DE 0004      movwf 0x4
```

Figure 5.6 Simulation du programme pour $m=7$ avec le MPLAB

La simulation du programme se fait par le MPLAB, outil de développement de MICROCHIP. On voit à gauche le programme écrit en langage C, à droite on voit le contenu des différents registres du microcontrôleur. L'exécution du programme pas à pas nous montre la variation de la valeur du portB, comme on le voit sur la figure, la valeur du portB est 80 en hexadécimale, preuve que la pin B7 est à 1.

5. Les résultats pratiques

Les figures si dessous nous donne les résultats obtenus. Elles nous montre le signal MLI calculé pour un indice de modulation $im=0.5$, ce qui donne une période du signal égale à 40 ms. On voit bien sur l'image une période complète du signal tracé pour $m=3$, l'échelle est de 2ms. Le nombre d'angles est égal à 3 pour un quart de période et 15 pour une période complète.

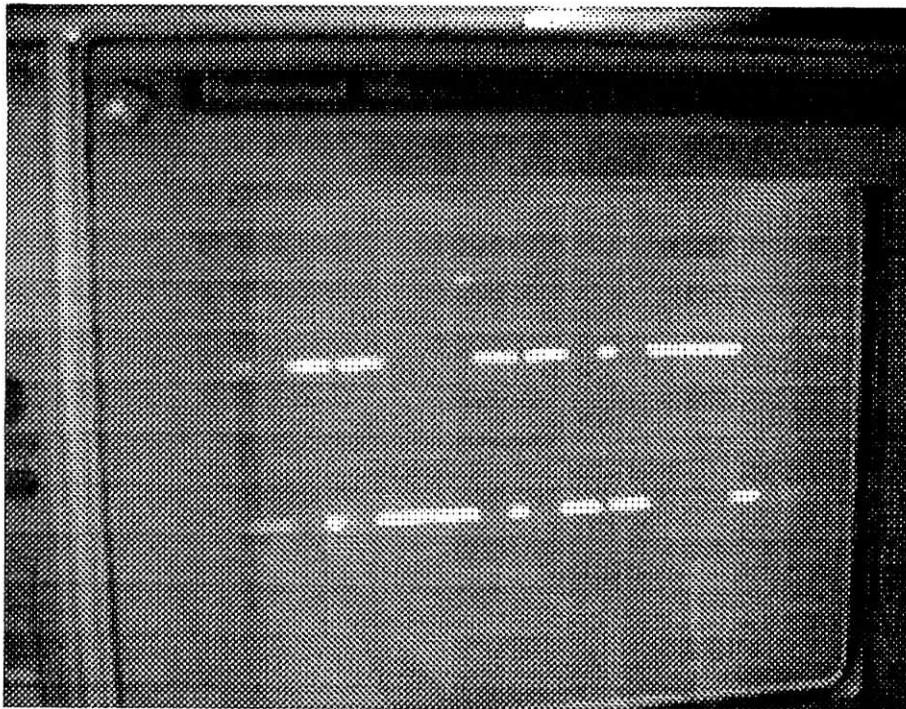
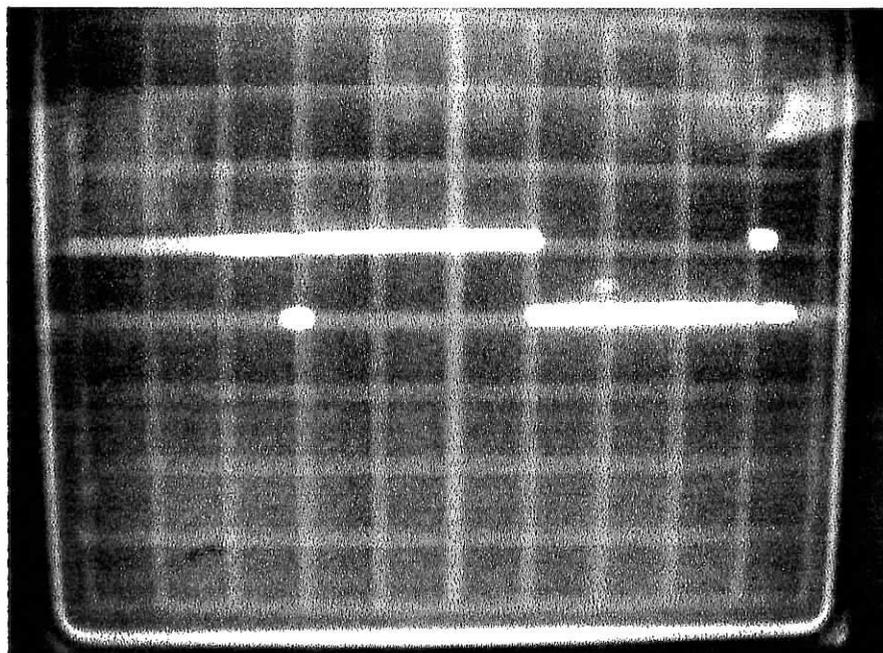


Figure5.7 Signal MLI obtenu dans la pratique pour $m=3$, $im=0.5$ (échelle de temps 4ms, échelle de tension 2V)



**Figure5.8 signal MLI visualiser sur oscilloscope
($m=3$ et $i_m=0.5$, échelle de temps 2ms, échelle de tension 2V)**



**Figure5.9 Signal MLI visualiser sur oscilloscope
($m=3$ et $i_m=0.1$, période 200ms, échelle temps 5ms, échelle de tension 5V)**

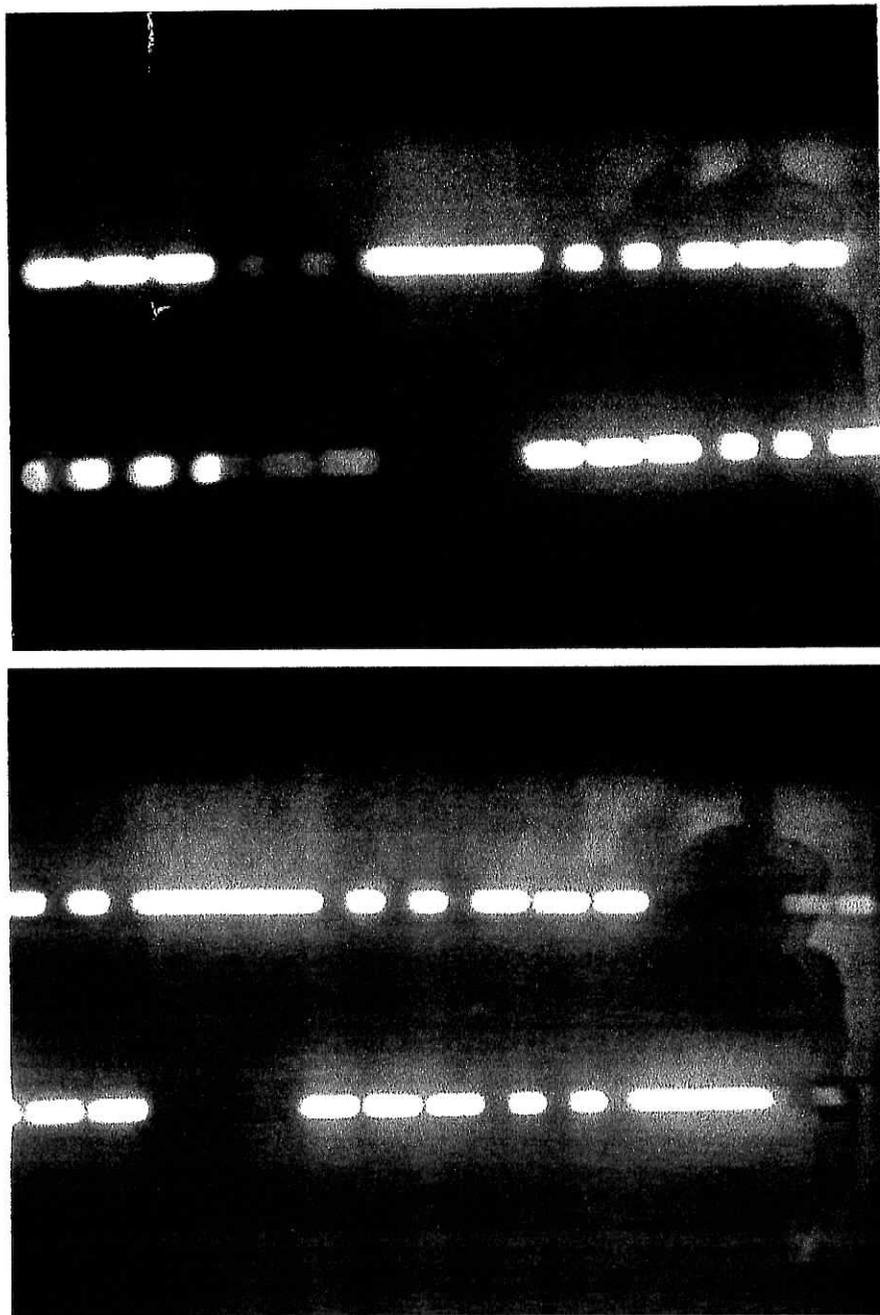


Figure5.10 MLI pour $m=5$ et $im=0.5$

Les figures ci-dessus ont été prises pour $m=5$ et $im=0.5$ pour différentes échelles. Les images ont été prises difficilement pour des raisons de stabilité du signal visualiser sur oscilloscope.

On voit ci-dessus pour différentes valeurs de m et de im le signal MLI généré, c'est les mêmes signaux obtenus avec la simulation MATLAB, même pour des indices de modulations faibles $im \leq 0.1$ et des indices de modulations grands $im \geq 1$.

6. Conclusion

La réalisation pratique d'un générateur d'une MLI calculée est possible avec l'algorithme proposé dans le chapitre 5. Cette algorithm nous simplifie le calcul des angles de commutations, et le rend possible à implémenter sur un microcontrôleur 16F876.

En a vue aussi que le signal sorti est d'une très bonne forme même pour les faibles indices de modulations que pour des grands indices de modulation.

Le signal est calculé pour $m=3$ et $m=5$. Pour $m=7$, la mémoire programme se sature et rend le calcul des angles impossibles.

La programmation d'une stratégie complète permettant de varier le nombre d'harmoniques à supprimer est impossible avec un seul PIC, car la mémoire programme ne suffit pas, alors il faut penser à un système à base de plusieurs microcontrôleurs qui exécute plusieurs tâches différentes (calcul des angles, génération du signal MLI, mémoire eeprom externe pour le sauvegarde des coefficients....).

La réalisation du circuit de retard peut être fait par un autre microcontrôleur 16F876 qui peut communiquer avec le microcontrôleur qui génère le signal MLI, il doit avoir comme données à stocker la fréquence du signal et le signal perdant une période.

Conclusion générale

Le but de ce travail est la réalisation d'une carte électronique à base d'un PIC16F876 destinée pour générer un signal MLI pour un onduleur triphasé alimentant un moteur asynchrone. Les angles de commutations sont calculés par un nouvel algorithme 'on-line' basé sur la méthode calculée.

En premier lieu, on a simulé l'algorithme avec le logiciel de simulation MATLAB, les signaux de sortie est une MLI, c'est les signaux qu'on doit obtenir dans la pratique.

Cette réalisation nous a permis de générer un signal MLI pour $m=3$ et $m=5$, donc on peut éliminer jusqu'à la 13^{ième} harmonique les résultats obtenus dans la simulation sont vérifiés, mais pour $m=7$ la mémoire programme se sature, donc on ne peut plus aller plus loin.

La réalisation complète d'un générateur de signal MLI pour un onduleur triphasé peut être faite par 2 microcontrôleurs, avec utilisation d'une mémoire externe pour sauvegarder les coefficients dont on a besoin pour calculer les angles de commutations, ou bien les sauvegarder dans la mémoire eeprom du PIC. Une stratégie prenant compte de la variation de m peut nous poser des problèmes au niveau de la mémoire programme du PIC, dans ce cas une solution DSP ou FPGA peut être envisagée.

BIBLIOGRAPHIE

- [1] KHIDER Moussa, Thèse magistère en électronique, option télécommunication 'Commande de vitesse en temps réel d'un moteur asynchrone triphasé', Avril 2002.
- [2] A.GOURDIN et M.BOUMAH RAT, 'méthodes numériques appliquées', OPU, seconde édition, 1991.
- [3] Guy SEGIER, 'L'électronique de puissance', édition DUNOD.
- [4] Guy SEGIER, et NITLET, 'L'électronique industrielle, édition DUNOD.
- [5] H.Foch, F. Forest et T. Meynard, 'onduleurs de tension. Structures. Principes. Applications', Techniques de l'ingénieur, vol. D3III, N°D3176, Novembre 1998.
- [6] Guy GRELLET et Guy CLERC ' Actionneurs électriques. Principes. Modèles. Commande', Edition Eyrolles, 1997.
- [7] Guy SEGIER et R. BAUSIERIE et F. LABRIQUE, ' es convertisseurs de l'électronique de puissance, Tome4 : a conversion continu-alternatif', Edition TECDOC, Paris, 2^{ième} Edition, 1996.
- [8] Christian Tavernier ' Mise en œuvre des PIC', Edition DUND, Paris 2001.
- [9] Christian Tavernier 'Application industriel des PIC', Edition DUND, Paris 2001.
- [10] BIGONOFF 'LA PROGRAMMATION DES PICS PAR BIGONOFF. LA GAMME MID-RANGE PAR L'ETUDE DES 16F87X (16F876-16F877)'.
[11] BIGONOFF 'LA PROGRAMMATION DES PICS PAR BIGONOFF. DEMARRER LES PICS AVEC LE PIC16F84'.
- [12] Pierre Cabanni 'Electronique digital'.
- [13] M. AUMIAUX, ' 'emploi des microprocesseurs', édition MASSN. 1982.

Annexe A

```

%*****MLI.m*****%
% Ce programme simule le nouvel algorithme decrit %
% dans le chapitre 4 %il calcule les angles de %
% commutations par leurs formules approximé et %
% génère le signal MLI correspondant pour %
% différents valeurs de m %
%*****%

%*** génération du signal MLI pour im=0.5 et m=3 ***%
m=3;
im=0.5;
e=0.01;
A0=[1.1659 9.0872 1.232];
A1=[1.3675 1.5766 2.286 ];
p=[4 10 8];
k=1;
a1=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a2=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=2;
a3=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;
a4=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=3;
a5=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a6=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
a=[ a1 a2 a3 a4 a5 a6 ];
ai=[ a6 a5 a4 a3 a2 a1 ];
ap=[ a 180-ai];
am=[ 0 ap 180-e 180+e 180+ap 360 ]/9;
figure;
s1=[ 1 1 -1 -1 ];
s2=[ s1 s1 s1 s1 s1 s1 s1 ];
plot(am,s2);
grid;
title('le signal MLI générer pour im=0.5 et m=3');
xlabel('angles de commutations');
ylabel('étant de sortie du bras onduleur');
pause

%*** génération du signal MLI pour im=0.5 et m=5 ***%
m=5;
im=0.5;
A0=[1.0514 4.4472 0.4619 7.1389 0.9653];
A1=[0.7118 1.3392 1.4311 0.7694 1.1071];
p=[4 6 6 11 9];
k=1;
a1=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a2=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=2;
a3=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;

```

```

a4=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=3;
a5=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a6=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=4;
a7=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;
a8=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=5;
a9=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a10=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
a=[ a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 ];
ai=[a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 ];
ap=[ a 180-ai ];
am=[ 0 ap 180-e 180+e 180+ap 360 ]/9;
s1=[ 1 1 -1 -1 ];
s2=[ s1 ];
plot(am,s2);
grid;
title('le signal MLI générer pour im=0.5 et m=5');
xlabel('angles de commutations');
ylabel('étant de sortie du bras onduleur');
pause

%*** génération du signal MLI pour im=0.5 et m=7 ***%

A0=[0.904 2.597 0.511 4.364 0.351 5.821 0.766];
A1=[0.453 0.810 1.002 1.053 1.106 0.403 0.585];
p=[4 6 5 7 7 12 10];
k=1;
a1=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a2=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=2;
a3=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;
a4=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=3;
a5=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a6=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=4;
a7=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;
a8=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=5;
a9=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a10=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
k=6;
a11=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)-e;
a12=60*k/(m+1)+A0(k)*im-A1(k)*im.^p(k)+e;
k=7;
a13=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)-e;
a14=60*(k+1)/(m+1)+[A0(k)-60/(m+1)]*im-A1(k)*im.^p(k)+e;
a=[ a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 ];
ai=[ a14 a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 ];

```

```
ap=[ a 180-ai ];
am=[ 0 ap 180-e 180+e 180+ap 360 ]/9;
s1=[ 1 -1 -1 ];
s2=[ s1 ];
figure;
plot(am,s2);
grid;
title('le signal MLI générer pour im=0.5 et m=7');
xlabel('angles de commutations sur une période');
ylabel('etant de sortie du bras onduleur');
```

Annexe B

```

////////////////////////////////////
// Ce programme genere le signal MLI pour m=3 //
// L'indice de modulation im est lue par //
// Le convertisseur analogique/numérique //
// La sortie du signal est dans B6 et B7 //
////////////////////////////////////

#include <16F876.h>
#define adc=8
#define delay(clock=20000000)
#define fuses HS,NOWDT
#define int_AD
AD_isr() {
reset_cpu();
}

float mlu (int p, float im);

void main() {
float angle[15],imp,im,A0,A1;
int time, i, p;
int t[3][3] = {
{1166, 1367, 4000},
{9087, 1576, 10000},
{1232, 2286, 8000}
};

setup_adc_ports(RA0_ANALOG);
setup_adc(ADC_CLOCK_DIV_2);
setup_spi(FALSE);
setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_ccp1(CCP_OFF);
setup_ccp2(CCP_OFF);
enable_interrupts(INT_AD);
enable_interrupts(global);
/*****
/* lecture de la vitesse demander */
*****/
im = Read_ADC();
if (im<=232)
im=im/200;
else im=1.15;
if (im == 0 )
{
output_low(PIN_B0);
output_low(PIN_B1);
}
}

```

```

else
{
/*****/
/*  calcule des angles de commutation en seconde m=3  */
/*****/

    angle[0]=0;
    p=t[1][2]/1000;
    A0=t[1][0]/1000;
    A1=t[1][1]/1000;
    imp=mlu(p,im);
    angle[2]=(30+A0*im-A1*imp)/(18*im);
    p=t[0][2]/1000;
    A0=t[0][0]/1000;
    A1=t[0][1]/1000;
    imp=mlu(p,im);
    angle[1]=(30+(A0-15)*im-A1*imp)/(18*im);
    p=t[2][2]/1000;
    A0=t[2][0]/1000;
    A1=t[2][1]/1000;
    imp=mlu(p,im);
    angle[3]=(60+(A0-15)*im-A1*imp)/(18*im);

    for (i=0 ; i<=3 ; i+=1)
    {
        angle[4+i]=20-angle[3-i];
    }

    for (i=8 ; i<=14 ; i+=1)
    {
        angle[i]=20+angle[i-7];
    }

/*****/
/*****/
/*                                generation du signal MLI
*/

/*****/
/*****/

do{
    for (i=0 ; i<=12 ; i+=2)
    {
        output_high(PIN_B6);
        output_low(PIN_B7);
        time=angle[i+1]-angle[i]+0.5;
        delay_ms(time);
        output_low(PIN_B6);
        output_high(PIN_B7);
    }
}

```

```
        time=angle[i+2]-angle[i+1]+0.5;
        delay_ms(time);
    }
}while (TRUE);
}
}
/*****
*****/
/*          calcule d'une puissance
*/
/*****
*****/

float mlu (int p, float im)

{   int i,t=1;

    for(i=1 ; i<=p ; i++)
        t=t*im;
    return(t);
}
```

Annexe C**Liste des composants utilisés**

U1 : microcontrôleur PIC16F876

C1, C4 : 2 condensateurs chimiques 47uF

C2, C3 : 2 condensateurs 100nF

C5, C6 : 15PF

U2 : pont redresseur

U3 : régulateur L7805

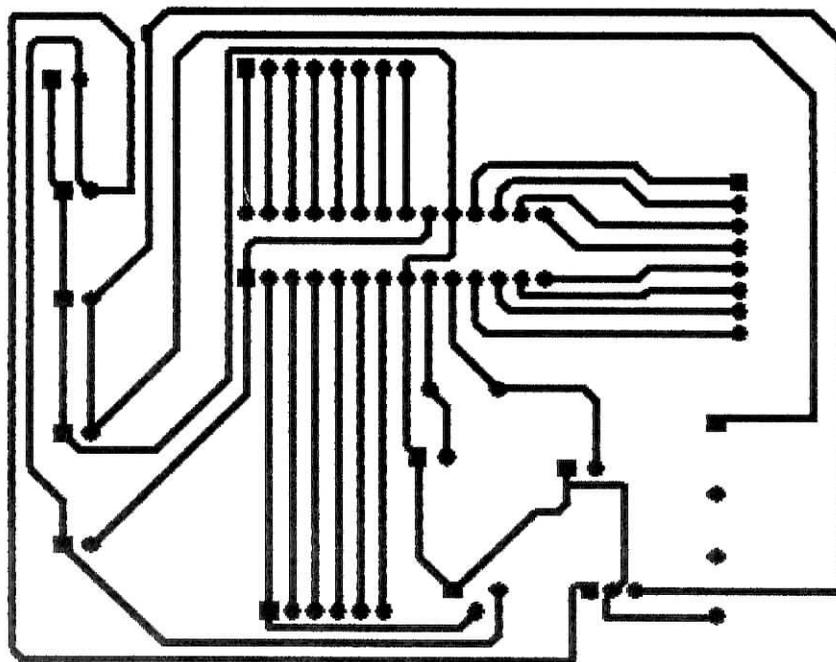
R1 : résistance variable 5k

Q : quartz 20MHz

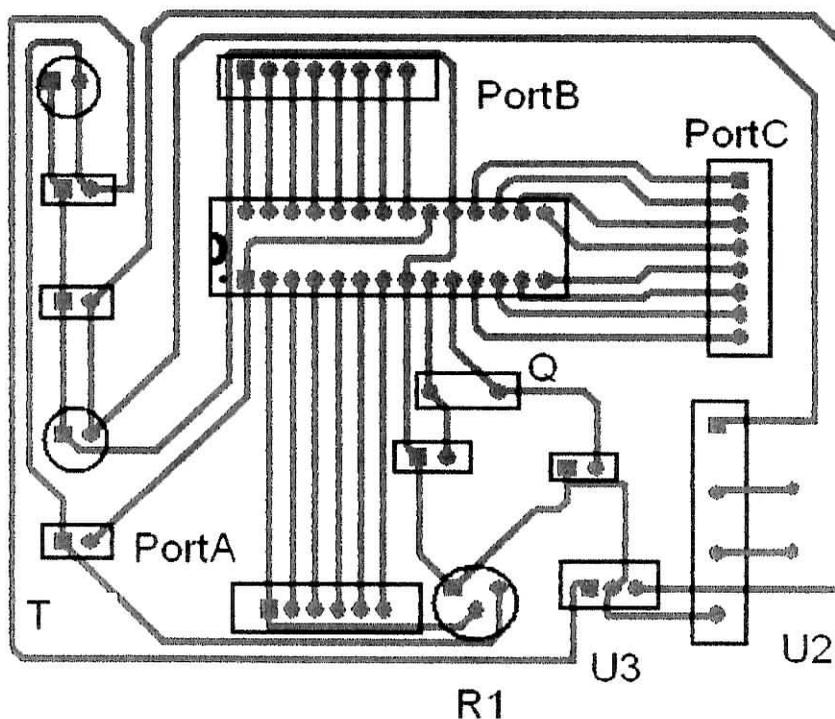
T : interrupteur

Annexe D

Tracé du circuit imprimé et implantation des composants



Tracé du circuit imprimé



Implantation des composants