

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

6/94

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT *ELECTRONIQUE*

PROJET DE FIN D'ETUDES

SUJET

ÉTUDE D'UNE TECHNIQUE
NUMÉRIQUE DE GÉNÉRATION
DE SIGNAUX M.L.I.

Proposé par :
M^{rs}. C. LARBES
&
M^{rs}. S.M. AIT-CHEIKH

Etudié par :
M^{rs}. A. DEROUAZI
&
M^{rs}. M. KADRI

Dirigé par :
M^{rs}. C. LARBES
&
M^{rs}. S.M. AIT-CHEIKH.

PROMOTION 1993 - 1994

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية

MINISTRE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT *ELECTRONIQUE*

PROJET DE FIN D'ETUDES

SUJET

ÉTUDE D'UNE TECHNIQUE
NUMÉRIQUE DE GÉNÉRATION
DE SIGNAUX M.L.I.

Proposé par :
M^{rs}: C. LARBES
&
M^{rs}: S.M. AIT-CHEIKH

Etudié par :
M^{rs}: A. DEROUAZI
&
M^{rs}: M. KADRI

Dirigé par :
M^{rs}: C. LARBES
&
M^{rs}: S.M. AIT-CHEIKH.

PROMOTION 1993 - 1994

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الحمد لله، أما بعد:

نهدي هذا العمل المتواضع للوالدين
العزيزين الذين مكثنا من الوصول
لما نحن عليه، راجين من السولى
أن نوفي ولو جزئاً من واجبنا
نحوهما.

Remerciements.

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

À nos parents qui ont veillé et souffert pour notre éducation, nous leur adressons nos plus sincères remerciements.

Nous exprimons notre vive reconnaissance à M^r. LARBES, notre promoteur, pour son suivi et son aide documentaire.

Nous tenons à exprimer notre gratitude envers M^r. AIT. CHEIKH pour ses précieux conseils.

À nos professeurs qui nous ont encadré pendant notre graduation à l'école nationale polytechnique, nous leur exprimons notre respect le plus profond.

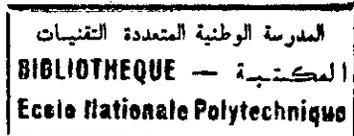
Nous n'oublierons pas de remercier tout le personnel de la bibliothèque, de son périodique et celui de l'annexe du centre de calcul.

Enfin, nous remercions toute personne ayant pu, de près ou de loin, contribuer avec un apport pour l'aboutissement de ce travail.

GLOSSAIRE :.

- N : nombre d'angles de commutations par quard de periode.
- M : indice de modulation dans la SHE.
- n_i : ordre de harmoniques.
- a_n ou h_n : n^{eme} harmoniques.
- Fmm: force magneto motrice.
- MCA: moteur à courant alternatif.
- LUT: look up tables.
- f_d : fréquence dominante.
- f_c : fréquence de commutation.
- U_d : tension d'alimentation.
- m : indice de modulation dans la SPWM.
- V : tension de sortie.
- i, I : courant de sortie.
- MEC: Motor equivalent circuit.
- p.u: par unité.

SOMMAIRE



INTRODUCTION

CHAPITRE 1 : GENERALITES

1.1 Introduction.....	2
1.2 Comparaison des differents types de moteurs.....	2
1.3 Commande de la vitesse.....	4
1.4 Convertisseurs de fréquence.....	4
1.4.1 Convertisseurs directs.....	4
1.4.2 Convertisseurs indirects.....	5
1.5 Effet des harmoniques.....	6
1.5.1 L'échauffement.....	6
1.5.2 Les couples pulsés.....	6
1.6 Conclusion.....	7

CHAPITRE 2 : LA MODULATION EN LARGEUR D'IMPULSION

2.1 Introduction.....	8
2.2 Contrôle de la tension et de la fréquence les onduleurs.....	8
2.3 Onduleur PWM.....	9
2.4 Classification des strategies PWM.....	10
2.4.1 PWM engendrée (sinusoidale), SPWM.....	12
2.4.2 PWM programmée, PPWM.....	13
2.4.2.1 PPWM avec élimination d'harmoniques.....	16
2.4.2.2 PPWM à contrôle optimale, EOC.....	16
a) Minimisation du taux d'harmoniques.....	18

b) Minimisation des ondulation.....	19
de couples	
c) Minimisation des pertes.....	19
2.5 Conclusion.....	20
 CHAPITRE 3 : METHODES NUMERIQUES POUR LA RESOLUTION	
DES EQUATIONS NON-LINEAIRES	
3.1 Introduction.....	21
3.2 Méthode de REGULA-FALSI.....	22
3.3 Principe de la méthode de NEWTON-RAPHSON.....	23
3.4 Critère d'arrêt des itérations.....	25
3.5 Condition de convergence.....	26
3.6 Ordre de convergence de la méthode de NEWTON-RAPHSON.....	27
3.7 extension aux systèmes d'équations non-linéaires.....	29
3.8 Méthode de NEWTON dite "amortie".....	31
3.9 Conclusion.....	32
 CHAPITRE 4 : LA PWM AVEC ELIMINATION SELECTIVES D'HARMONIQUES	
4.1 Intoduction.....	34
4.2 Expression des tensions.....	35
4.3 Elimination sélictives d'harmoniques pure.....	38
4.3.1 Organigramme.....	38
4.3.2 Résultats.....	40
4.3.3 Organigramme de génération des angles.....	44
4.4 Elimination sélective d'harmoniques avec asservissement du fondamental.....	46
4.4.1 Cas des systèmes d'ordre pair.....	46

4.4.2 Cas des systèmes d'ordre impair.....59

4.5 Conclusion.....69

CHAPITRE 5 : IMPLEMENTATION PAR MICROPROCESSEUR

5.1 Intoduction.....70

5.2 Implémentation OFF-LINE.....71

5.3 Implémentation ON-LINE.....75

5.4 Exemples de la mise en formes des signaux.....77

CONCLUSION

ANNEXES

BIBLIOGRAPHIE

INTRODUCTION

La variation de vitesse des moteurs électriques est, sans doute, une des plus importantes fonctions de base de

l'électronique de puissance; rendue indispensable par les besoins industriels actuels. Ceci est dû au souci d'obtenir la vitesse optimale par l'asservissement des moteurs de chaque partie d'un ensemble à chacune des phases d'un processus.

Ainsi, de nouveaux procédés et techniques apparaissent, dont l'objectif est de satisfaire les exigences du contrôle de vitesse et résoudre les problèmes d'ordre pratique.

Cependant, pour des raisons que nous citerons ultérieurement, on utilise plus fréquemment les variateurs de vitesse des moteurs asynchrones.

Parmi les divers convertisseurs, l'onduleur à modulation en largeur d'impulsion (PWM) présente d'excellentes performances. Selon la manière de déterminer les angles de commutations plusieurs stratégies découlent de cette technique.

L'objet de notre travail est l'étude d'une technique numérique de génération de signaux en largeur d'impulsion destinée à commander un onduleur de tension triphasé à deux niveaux alimentant un moteur asynchrone.

Le premier chapitre introduit des généralités. Le second présente le principe de la modulation en largeur d'impulsion ainsi que les différentes techniques existantes.

Une présentation de la méthode numérique utilisée fera l'objet du troisième chapitre, laquelle, dans le quatrième chapitre sera appliquée dans le cas de notre problème. Dans le cinquième chapitre nous donnerons un aperçu sur la commande de l'onduleur et nous présenterons la manière de l'implémentation. Finalement, nous résumeront le travail fait ainsi que les difficultés rencontrées dans une conclusion générale.

CHAPITRE 1

1.1 INTRODUCTION :

Un moteur électrique est destiné à fournir de l'énergie mécanique. Ainsi le choix du type de moteur et de ses caractéristiques dépendent essentiellement du domaine d'application.

Tout moteur comporte deux parties :

- Le rotor ou l'induit, qui est la partie mobile,
- le stator ou l'inducteur, qui est la partie fixe.

1.2 COMPARAISON DES DIFFERENTS TYPES DE MOTEUR :

Fondamentalement, deux grandes classes de moteurs peuvent être distinguées :

- Les moteurs à courant continu,
- Les moteurs à courant alternatif.

Le moteur à courant continu présente d'excellentes performances et une facilité relative dans sa commande.

Toutefois son collecteur lui limite son champs d'application tel que :

- tensions d'alimentation élevées,
- vitesses très élevées,
- très fortes puissances,
- restriction d'utilisation dans une atmosphère explosive.

Ces inconvénients ont conduit à opter, de plus en plus, pour l'utilisation du moteur à courant alternatif. Ce dernier comporte deux catégories :

- moteur synchrone,
- moteur asynchrone.

Le premier type, synchronisé à la fréquence du secteur, présente les avantages suivants :

- vitesse constante quelque soit la charge,
- peut supporter des chutes de tension relativement importantes,
- un rendement compris entre 93 et 97 % pour des puissances allant de 1 000 à 30 000 KW.

Mais d'un autre côté il présente les inconvénients suivants:

- la difficulté de démarrage,
- la nécessité de l'accrochage sur le réseau,
- le coût plus élevé.

Les inconvénients présentés par la machine synchrone et la machine à courant continu ont contribué à la prépondérance croissante des machines asynchrones avantagées par :

- une construction simple et robuste,
- un faible encombrement,
- un prix bas,
- une consommation d'énergie réduite,
- une maintenance restreinte.

Cependant, le moteur asynchrone présente des difficultés au niveau de la commande. Mais le développement de l'électronique de puissance durant ces dernières années, a apporté certaines solutions.

1.3 COMMANDE DE LA VITESSE :

La commande électronique du moteur asynchrone peut être réalisée de deux manières :

- Par variation de la tension du stator à fréquence constante,
- Par variation de la fréquence.

Actuellement le réglage s'effectue de plus en plus par variation de fréquence où le rapport tension/fréquence est maintenu constant. Ceci permet, notamment, d'obtenir des caractéristiques couples-vitesse semblables à celles des moteurs à courant continu.

1.4 CONVERTISSEURS DE FREQUENCE :

La commande électronique du moteur asynchrone fait appel à de nombreux types de convertisseurs. On peut y trouver, selon le type du moteur, la nature de la source d'alimentation et le mode de réglage, deux grandes classes :

- les convertisseurs directs,
- les convertisseurs indirects.

1.4.1 Convertisseurs Directs :

Les convertisseurs directs appelés "cycloconvertisseurs" transforment directement la tension du réseau en une tension ou un courant d'amplitude et de fréquence variables. Ils sont généralement à commutation naturelle, dont on distingue deux classes :

- Les cycloconvertisseurs commutés par le réseau,
- les cycloconvertisseurs commutés par la charge.

Le cycloconvertisseur ne peut fournir que des fréquences de sortie très basses ($f < 20$ Hz), il présente aussi un mauvais facteur de puissance, une commande très compliquée et une grande sensibilité aux perturbations du réseau. Tout cela rend l'emploi des cycloconvertisseurs limité au contrôle des moteurs lents.

1.4.2 Convertisseurs Indirects :

Il s'agit d'une double conversion passant par l'intermédiaire d'un étage à courant continu. Ce type de convertisseur réalise au moins trois fonctions :

- variation de la fréquence,
- réglage de la tension ou du courant,
- commutation du courant dans les bobinages du moteur commandé.

En outre le système de commande a également un rôle de protection, en ce sens qu'il limite l'intensité du courant à une valeur tolérable par la machine. Les convertisseurs indirects peuvent être classifiés comme suit :

- redresseur-commutateur de tension,
- redresseur-commutateur de courant,
- hacheur-commutateur de courant,
- hacheur-commutateur de tension,
- onduleur à couplage polygonal ou à ponts décalés,
- onduleur à modulation de largeur d'impulsion.

Pour un fonctionnement à flux constant, la solution ensemble hacheur-onduleur de tension est actuellement abandonnée à cause du fort taux d'harmoniques générés par l'alimentation en créneaux conduisant à des pertes d'énergie et un encombrement du système en augmentant le volume du filtre.

1.5 EFFET DES HARMONIQUES :

Dans la commande à vitesse contrôlée, les machines sont alimentées par des convertisseurs dont les signaux de sortie sont riches en harmoniques.

L'effet des harmoniques sur les moteurs apparait sous deux formes, l'échauffement et les couples pulsés [9].

1.5.1 L'échauffement :

Quand le signal est riche en harmoniques, la valeur efficace du courant augmente et les pertes dans le noyau et dans le cuivre augmentent aussi.

1.5.2 Les couples pulsés :

Un harmonique de couple pulsé est produit par l'interaction d'un harmonique de flux dans l'entrefer avec un harmonique de courant dans le rotor. L'importance du flux dans l'entrefer aux fréquences harmoniques est négligeable, et les couples dominants sont ceux dûs à l'interaction du flux fondamental dans l'entrefer avec les harmoniques de courant du rotor.

Ainsi, le $k^{\text{ème}}$ harmonique de courant I , réagit avec le flux fondamental ϕ_1 donnant une amplitude proportionnelle à :

$$T_{k\pm 1} = \phi_1 I_k \quad (1.1)$$

Le couple est d'ordre $(k+1)$ pour les séquences négatives du courant, et d'ordre $(k-1)$ pour les séquences positives.

On voit qu'une composante du couple d'ordre k est commune à deux harmoniques de courant (ou de tension) d'ordre $(k+1)$ et $(k-1)$.

Ces couples causent une instabilité de la vitesse de la machine. L'effet des composantes du couple de hautes fréquences peut être affaibli par l'inertie mécanique du système. La fluctuation de vitesse peut s'aggraver si les vitesses de travail sont réduites ou que le système possède une faible inertie.

La fréquence d'un couple pulsé peut être proche de la fréquence de résonance mécanique ce qui provoque de fortes vibrations d'axe, une usure des dents de l'engrenage et d'insatisfaisantes performances dans les systèmes asservis.

1.6 CONCLUSION :

Pour éviter ces problèmes et surtout celui du filtrage, les alimentations des machines asynchrones s'orientent de plus en plus vers la technique de modulation de largeur d'impulsion (PWM).

Son principe consiste à augmenter le nombre de commutations par période de la tension de sortie permettant ainsi le contrôle à la fois de l'amplitude et de la fréquence du fondamental et aussi réduire le taux d'harmoniques.

CHAPITRE 2

2.1 INTRODUCTION

Vu les progrès récents dans les semiconducteurs et par conséquent dans l'électronique de puissance, on s'oriente de plus en plus vers la commande en modulation de largeur d'impulsion [18]. Cette technique, à l'inverse de celle à un créneau par alternance, accroît le nombre de commutations par alternance, et cela a essentiellement deux avantages, d'un côté elle repousse les harmoniques de tension à des fréquences élevées facilitant ainsi le filtrage et d'un autre côté elle permet de varier l'amplitude du fondamental.

2.2 CONTROLE DE LA TENSION ET DE LA FREQUENCE DANS LES ONDULEURS :

La tension et la fréquence de sortie de l'onduleur doivent être contrôlées continuellement dans les applications de commande des machines asynchrones. Avec des alimentations en énergie régulée, la fréquence est fixée, par contre la tension doit être contrôlée à cause des variations de la charge et de la tension d'alimentation.

Le contrôle de la fréquence est direct. Si la fréquence désirée est fixée, la fréquence de référence est générée par un oscillateur à cristal. Pour des opérations à fréquence variable, on utilisera un oscillateur commandé par tension (VCO). La stabilité de la fréquence de sortie de l'onduleur est déterminée uniquement par celle de la référence et n'est pas affectée par les variations de la charge ou de la source.

En général, la tension de sortie de l'onduleur est contrôlée de deux manières :

- contrôle de la tension d'entrée ou modulation en amplitude d'impulsion (PAM),
- contrôle de la tension à l'intérieur de l'onduleur par la modulation en largeur d'impulsion (PWM).

2.3 ONDULEUR PWM :

les onduleurs à onde en étage (stepped wave inverter) ont des avantages et des limitations. Leur logique est simple et les pertes de commutations sont réduites. Une difficulté dans le pouvoir de commutation apparaît quand la tension est réduite. Le contrôle de vitesse devient un problème avec ce type d'onduleur car avec de faibles tensions les harmoniques de courant deviennent importantes causant un échauffement et des couples pulsés. De plus, l'introduction d'un filtre pass-bas au niveau de la tension d'entrée (dc link), peut déstabiliser le système aux faibles vitesses.

Ces problèmes peuvent être surmontés en utilisant la modulation en largeur d'impulsion. Les onduleurs PWM sont alimentés par un pont redresseur à diodes et un filtre LC pour les applications industrielles générales [9] (figure 2.1).

Les schémas des onduleurs restent les mêmes, c'est la commande qui est modifiée.

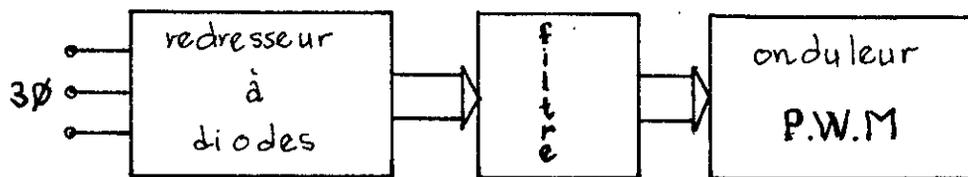


Figure 2.1 : Contrôle d'un moteur à induction à fréquence variable.

Le principe de la PWM consiste à former chaque alternance de la tension de sortie de plusieurs créneaux de largeurs convenables.

Le réglage de la fréquence est assuré électroniquement selon plusieurs techniques (figure 2.2).

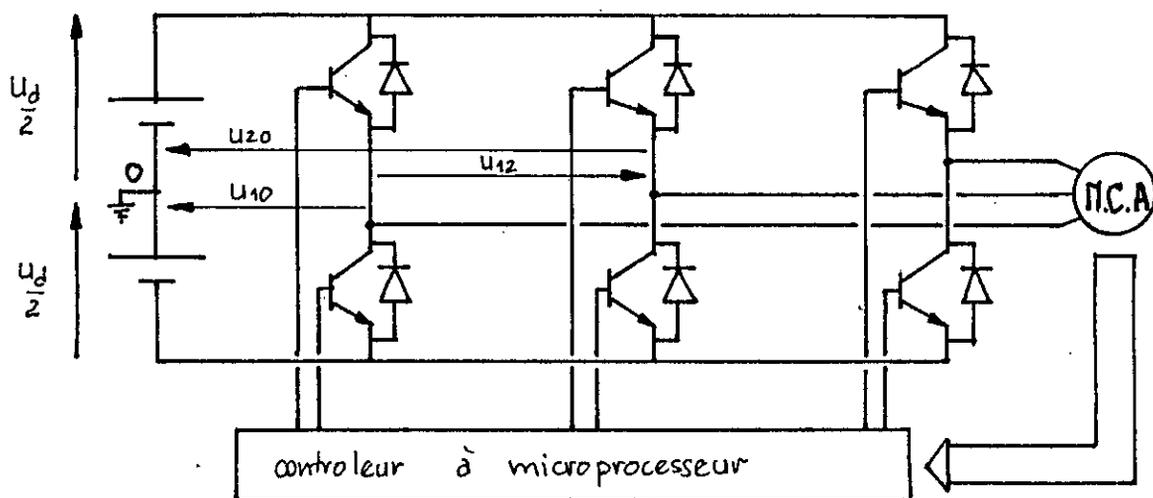


Figure 2.2 : Structure de base d'un onduleur commandé par μp

2.4 CLASSIFICATION DES STRATEGIES PWM :

On peut classer la PWM selon plusieurs critères, par exemple selon le mode de génération des angles (on-line, off-line) ou bien selon la technique utilisée, définie par la figure (2.3).

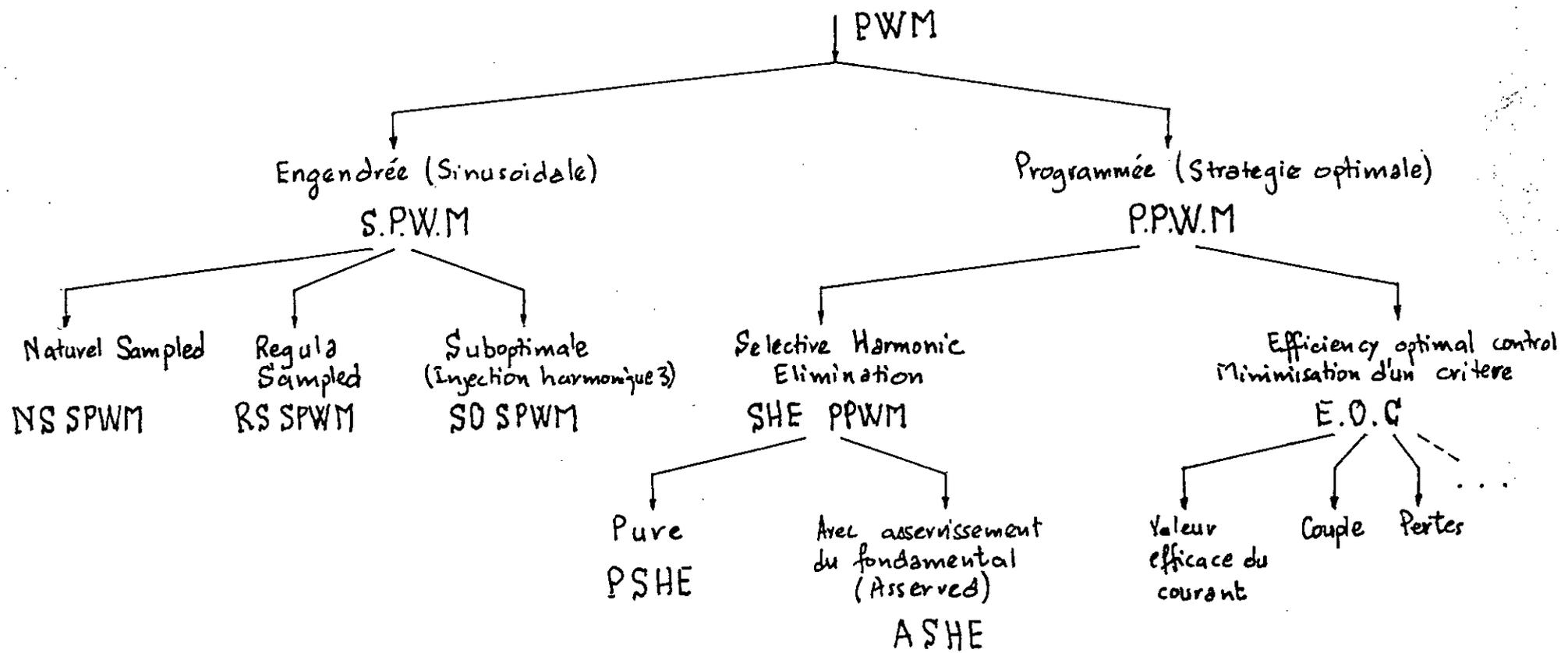


Figure 2.3 : Classification Des Strategies P.W.M

2.4.1 PWM Engendrée (sinusoidale), SPWM :

C'est la plus répandue, la figure (2.4) montre son principe. Elle est basée sur la comparaison d'une onde sinusoidale appelée modulante de fréquence fondamentale, et d'une autre généralement triangulaire appelée porteuse à fréquence plus élevée. Les intersections des deux signaux définissent les instants de commutations. La largeur des impulsions est sinusoidalement modulée, la commande de l'amplitude et de la fréquence fondamentale se fait par la variation de l'amplitude et de la fréquence de la modulante.

L'analyse de fourier donne :

$$v(t) = m \frac{U_d}{2} \sin(\omega_s + \phi) + \sum \text{harmoniques en fonctions de Bessel (2.1)}$$

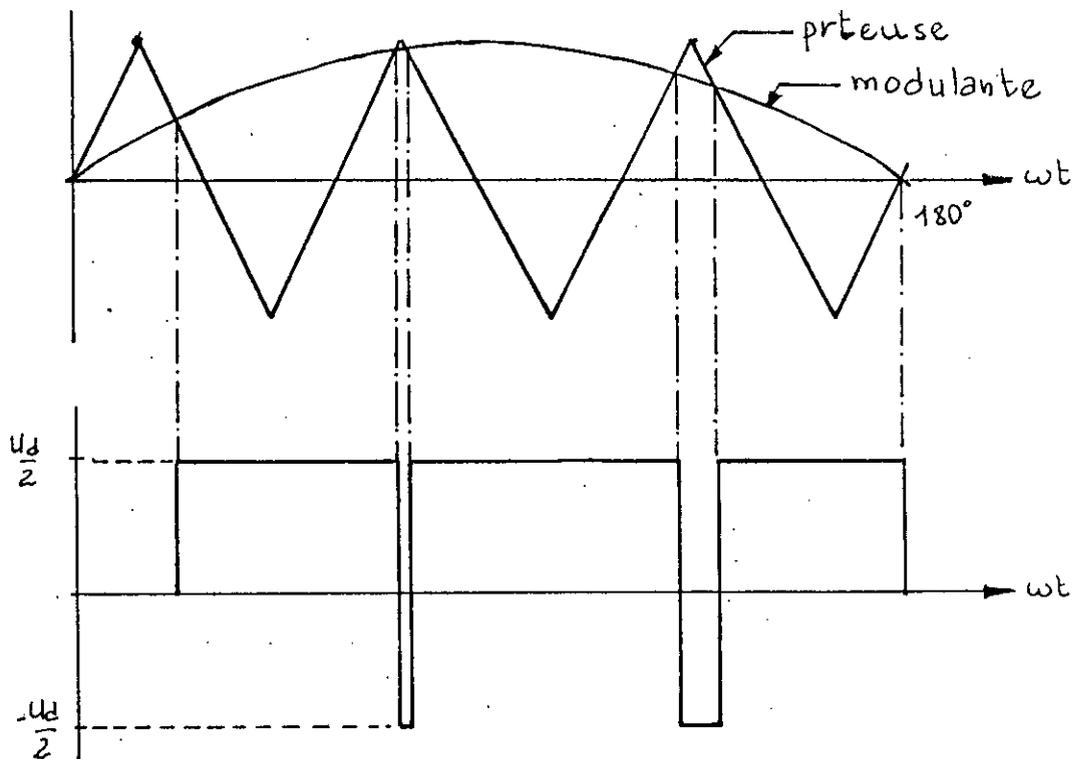


Figure 2.4 : Principe de la SPWM naturelle

Une autre variante est la SPWM régulièrement échantillonnée où la modulante (sinusoïde) est échantillonnée-bloquée avec une fréquence égale à celle de la porteuse (figure 2.5).

Cette méthode, qui peut être facilement implémentée sur microprocesseur, élimine les harmoniques des basses fréquences ainsi que les subharmoniques [9].

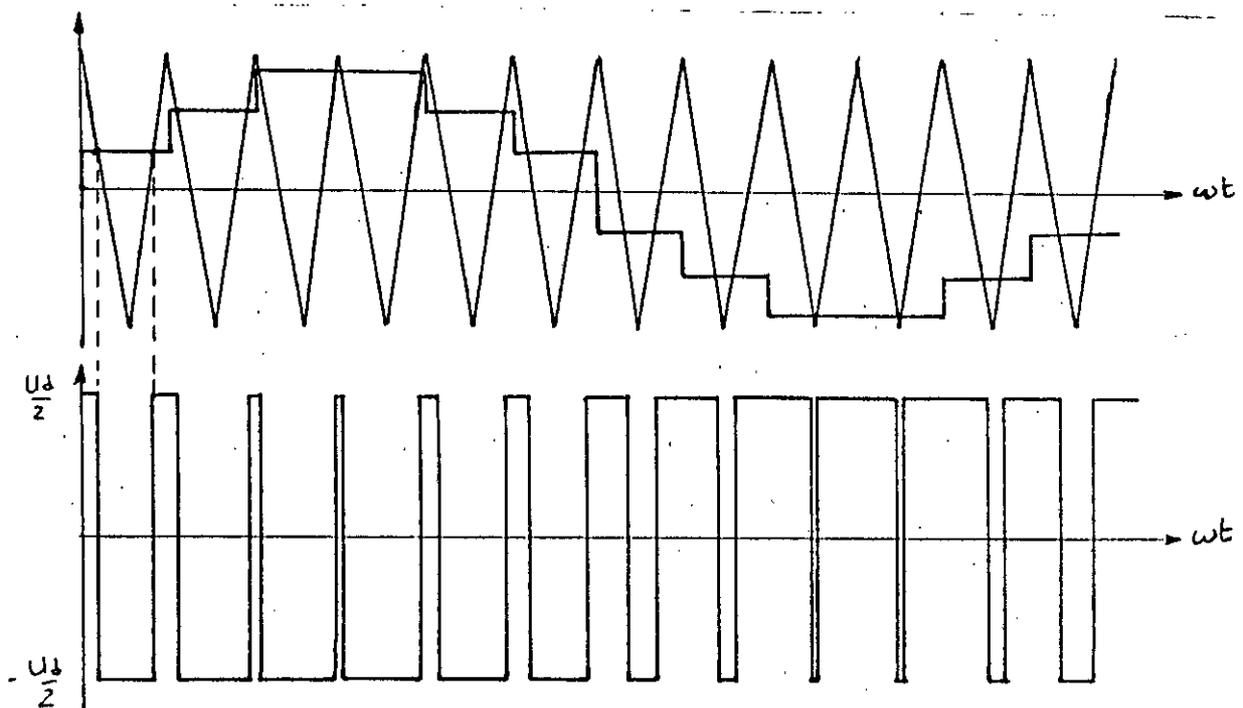


Figure 2.5 : SPWM régulière

2.4.2 PWM PROGRAMME , PPWM : [11]

Cette technique (appelée aussi numérique ou calculée) est par définition la plus efficace manière pour l'obtention des plus hautes

performances dans le contrôle des moteurs asynchrones. La raison est qu'avec cette technique on peut choisir l'optimisation de plusieurs objectifs comme l'élimination sélective des harmoniques (SHE), la minimisation des pertes dans les moteurs, la réduction des couples pulsés et la minimisation du bruit acoustique généré. En particulier l'élimination d'harmoniques a plusieurs avantages :

1°) La réduction approximativement de 50 % de la fréquence de commutation (switching frequency) comparée à la SPWM pour la même composante harmonique dominante (qui est le premier harmonique non éliminé).

2°) La génération de courants proches d'une sinusoïde avec des ondulations réduites. Ceci a pour conséquences des pertes harmoniques faibles, des ondulations de couple réduites et un bon rendement dans le moteur.

3°) La faible ondulation du courant d'entrée et la haute qualité du courant de sortie, amènent une réduction dans la taille du filtre d'entrée.

4°) La réduction de la fréquence de commutation contribue à la réduction des pertes de commutation dans les semiconducteurs.

5°) Le rapport entre la tension d'entrée du redresseur et la tension de sortie de l'onduleur est proche de l'unité (≈ 0.98).

Jusqu'à maintenant le problème majeur était la limitation de cette technique dans les basses fréquences. En effet le nombre d'angles de commutations nécessaires pour produire une bonne qualité du courant est trop grand pour le manipuler analytiquement (figure 2.6), en plus du large espace que demanderait son stockage pour différentes valeurs de l'indice de modulation.

On palie à ces problèmes en utilisant la SPWM pour les basses fréquences et la PPWM aux hautes fréquences.

En pratique, on maintient la composante harmonique dominante f_d dans une gamme de fréquence f_1 à f_u (1 à 1.5 KHz) respectivement. Ceci est souhaitable pour obtenir de bonnes performances. La variation de f_d en fonction de la vitesse du moteur (ou la fréquence f) est montrée dans la figure (2.6). On voit que le nombre N d'harmoniques éliminés, pour respecter la contrainte imposée sur f_d , varie avec la vitesse.

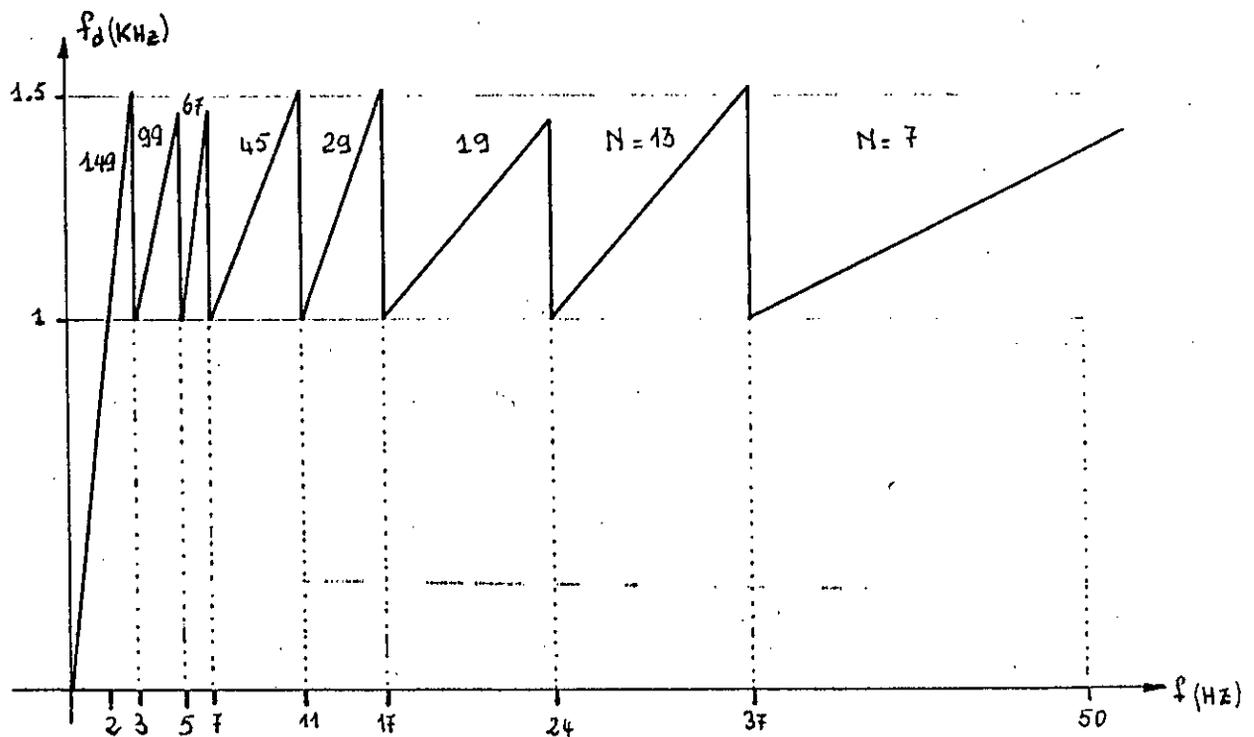


Figure 2.6 Variation de f_d en f^u de la fréquence

L'ordre de la composante dominante "D" peut être exprimé par :

$$D = 3N + 2 \quad (2.2)$$

et la fréquence correspondante par :

$$f_d = (3N + 2) f \quad (2.3)$$

Si par exemple on réduit la vitesse à 0.05 pu (i.e. 2.5 Hz), un simple calcul nous montrerait que N doit être égale à 110.

La fréquence de commutation f_c est donnée par :

$$f_c = (2(N+1) + 1) f \quad (2.4)$$

2.4.2.1 PPWM Avec Elimination D'harmoniques (SHE) [2]:

Elle consiste à choisir les N angles α_i ($i=1..N$) de telle façon à éliminer N harmoniques indésirables. Une autre approche suggère l'élimination de (N-1) harmoniques, la N^{ième} variable restante étant laissée pour asservir le fondamental à une valeur désirée. La première méthode est dite PSHE avec élimination sélective pure, La seconde est la ASHE avec élimination sélective d'harmoniques et asservissement du fondamental. Une étude plus approfondie lui sera consacrée au troisième chapitre.

2.4.2.2 PPWM A Contrôle Optimal (EOC): [13,14,15,16,18]

C'est le terme utilisé pour décrire les stratégies PWM développées spécialement pour optimiser (ou, minimiser) quelques critères de performances spécifiques, par exemple la minimisation du taux d'harmonique de courant, les pertes dans le moteur ou les

couples pulsés. Cette minimisation est basée sur des méthodes numériques.

La résolution d'un tel problème est plus difficile que la résolution d'un système d'équations non-linéaires du fait donné qu'on doit annuler la dérivée et donc on est amené à calculer la deuxième dérivée.

Différentes méthodes d'optimisation existent, dont beaucoup sont basées sur la méthode du gradient.

Le modèle exacte du moteur (MEC) constituant la charge de l'onduleur dépend de plusieurs paramètres (effet de peau, résistances du rotor et du stator..., figure 2.7). Ces paramètres dépendent aussi de la fréquence [17].

- U_k : tension stator
- R_1 : résistance du stator.
- R_2 : résistance rotor dépendante de la fréquence.
- R_{FE} : résistance représentant les pertes fer.
- L_{1s} : inductance du stator.
- L_{2s} : inductance du rotor
- L_M : inductance principale.
- S_k : glissement du $k^{ième}$ harmonique.

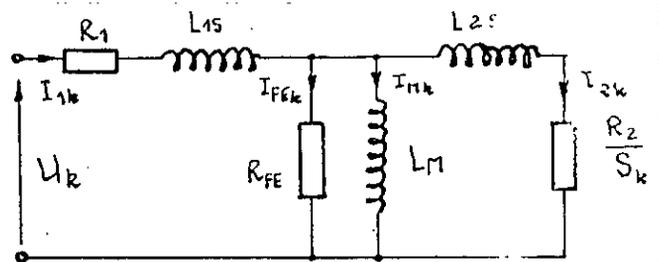


Figure 2.7 : Circuit Equivalent Moteur

a) Minimisation Du Taux D'harmoniques:

Un modèle simplifié d'un moteur est composé d'un circuit RL. Le taux d'harmoniques de courant est donné par :

$$THD = \frac{\sqrt{\sum_{n=5}^{\infty} |I_n|^2}}{|I_1|} \quad (2.5)$$

La composante du courant I_n de fréquence $n\omega$ est

$$I_n = \frac{V_n}{Z_n} \quad (2.6)$$

comme $Z_n = R + jnL\omega$

et $Ln\omega \gg R \Rightarrow Z_n \approx jnL\omega \quad (2.7)$

$$|I_n|^2 = \frac{|V_n|^2}{(nL\omega)^2}$$

il vient $THD = \frac{\sqrt{\sum_{n=5}^{\infty} \frac{|V_n|^2}{n^2}}}{|V_1|} \quad (2.8)$

En réalité on minimise le taux d'harmoniques de tensions pondérées par l'inverse du carré de leurs rangs.

On remarque une discontinuité des trajectoires des angles en fonction du fondamental [14,15].

b) Minimisation Des Ondulations De Couple [17] :

A partir du circuit de la figure (2.7) on tire les équations exactes qui régissent la marche du moteur, comprenant des termes électriques et d'autres mécaniques (couples). Ce sont des équations différentielles d'ordres cinq (5). Elles sont résolues par la méthode *de Range-Kutta*. Le couple est donné par :

$$T = (3/2) (pL_M) \sum_{k_f, k_b} (I_{skf} + I_{skb}) (I_{rkf} + I_{rkb}) \quad (2.9)$$

c) Minimisation Des Pertes :

Une minimisation rigoureuse doit considérer le modèle exacte du moteur (figure 2.7) incluant l'effet de peau dans le moteur (ainsi que sa dépendance en fréquence) et les pertes dans le fer.

L'expression des pertes causées par les harmoniques est :

$$S_k = 1 - \frac{1-S_1}{k} \quad \text{pour } k=7,13,19,\dots \quad (2.10)$$

(causants des champs magnetiques tournants dans le même sens que le fondamental) et :

$$S_k = 1 + \frac{1-S_1}{k} \quad \text{pour } k=5,11,17,\dots \quad (2.11)$$

(causant des champs tournants dans le sens inverse).

Avec le modèle simplifié (RL), l'expression des pertes devient :

$$P_H = \sum_{k=5,7,11,\dots} \frac{U_k^2 \cdot R}{R^2 + (k\omega_1 L)^2} \quad (2.12)$$

2.5 CONCLUSION :

Dans ce chapitre ont été présentés différents principes de techniques PWM. Leurs avantages et leurs inconvénients provenant essentiellement d'une forte présence d'harmoniques ont été aussi discutés.

Le but de chaque technique et finalement de produire une tension de sortie aussi sinusoidale que possible.

D'un point de vue pratique, des méthodes classiques comme la SPWM sont réalisées à partir de circuits analogiques relativement simples, dans d'autre, comme est le cas de la PPWM, l'implémentation des angles calculés nécessite une électronique plus complexe.

CHAPITRE 3

3.1 INTRODUCTION

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

Cette approche s'est vue épanouir avec l'apparition d'une nouvelle génération d'ordinateurs de plus en plus rapides et précis.

Fondamentalement deux questions se posent sur l'efficacité d'un algorithme :

- Converge-t-il ?;
- si oui, avec quelle rapidité ?.

D'innombrables méthodes ont été élaborées. Pour les équations non-linéaires, il existe des formules d'itérations à deux points [5], qui, pour évaluer le résultat d'une étape on a besoin du résultat des deux étapes précédentes. Lorsque la racine est localisée dans un intervalle, la convergence est assurée, la dichotomie en est un bon exemple. Dans les formules d'itérations à un seul point [5], on exploite seulement le résultat de la dernière itération.

Pour les systèmes d'équations non-linéaires, la résolution devient plus délicate. Ainsi les méthodes les plus rencontrées sont celles des substitutions successives [6], de Newton et d'optimisation [4]. Les deux premières sont une généralisation multidimensionnelle, la dernière connue aussi sous le nom de la méthode du gradient consiste à minimiser une norme du système.

L'algorithme de Newton-Raphson ([4],[5],[6]) est l'un des plus connus et des plus puissants. Basé sur un développement de Taylor au premier ordre, il converge rapidement vers la solution exacte si on a un bon estimé au départ. Cette question sera traitée plus en détail dans les prochaines sections.

3.2 METHODE DE REGULA-FALSI :

Elle est appelée aussi méthode de la corde. Soit x^* une racine de $f(x)$, comprise entre deux points x_g et x_d telle que :

$$f(x_d) \cdot f(x_g) < 0 \quad (3.1)$$

Le principe de la méthode consiste à prendre comme estimé x_k ($k=1, 2, \dots$), celui de l'intersection de la droite qui relie les points d'abscisses x_g et x_d avec l'axe des x .

Ainsi l'estimé x_k a pour expression :

$$x_k = \frac{y_d}{y_d - y_g} x_g - \frac{y_g}{y_d - y_g} x_d \quad (3.2)$$

Le point x_k dévise le segment $[x_g, x_d]$ en deux intervalles $[x_g, x_k]$ et $[x_k, x_d]$. L'étape suivante est de choisir celui des deux intervalles qui satisfait la condition de l'équation (3.1). Le choix d'un nouvel estimé se fait de la manière suivante :

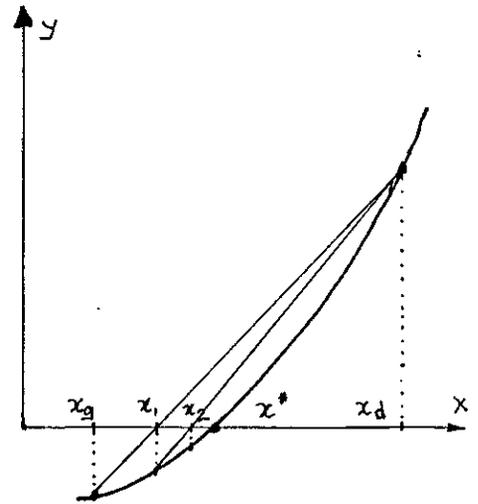


Figure 3.1 : Aspect géométrique de la méthode de Régula-Falsi.

$$p = f(x_d) \cdot f(x_k) \quad (3.3)$$

Si $p < 0$ alors $x_g = x_k$ $y_g = y_k$

Si $p > 0$ alors $x_d = x_k$ et $y_d = y_k$

L'avantage de cette méthode est que la convergence est assurée.

Variantes De La Méthode :

a) **Dichotomie:** elle divise le segment $[x_g, x_d]$ en deux parties égales.

b) Dans l'équation (3.3) on remplace par :

$$Si \quad p > 0 \quad \text{alors} \quad x_d = x_k, \quad y_d = y_k, \quad y_g = \alpha y_g \quad (3.4)$$

Si $\alpha = 1/2$ elle est dite méthode de Illinois. Dans la méthode de Pégasus, α prend la valeur $y_d / (y_d + y_k)$.

c) **Méthode de la sécante:** la méthode de Régula-Falsi nécessite deux points pour lesquels $f(x)$ a des signes opposés. Si on ne tient pas à épuiser l'effort dans la recherche de tels point, on peut utiliser les derniers deux points calculés, pour générer les nouveaux points de la séquence, par la formule:

$$x_{k+1} = \frac{y_k}{y_k - y_{k-1}} x_{k-1} - \frac{y_{k-1}}{y_{k-1} - y_k} x_k \quad (3.5)$$

3.3 PRINCIPE DE LA METHODE DE NEWTON-RAPHSON ⁽¹⁾ [4,5,6]:

Soit $f(x)$ une fonction continue et continuellement dérivable dans le voisinage de x^* , une racine de f , c.à.d :

$$f(x^*) = 0 \quad (3.6)$$

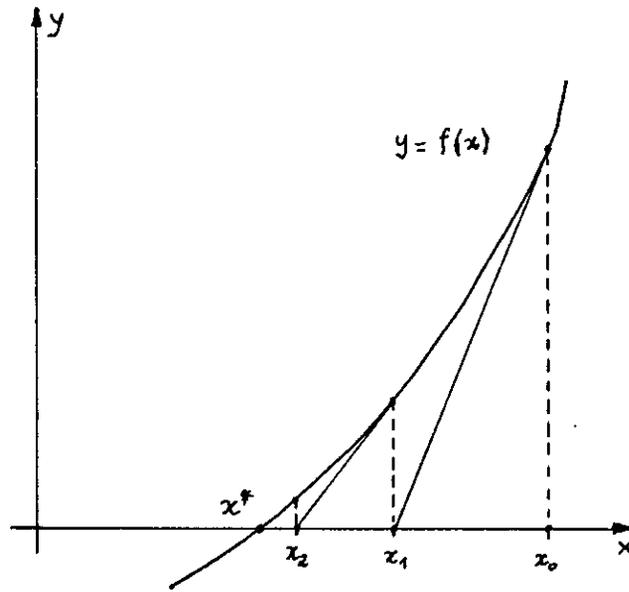


Figure 3.2 : Aspect géométrique de la méthode de Newton.

Le développement en série de Taylor autour d'un estimé x' s'écrit :

$$f(x^*) = f(x_n) + (x^* - x_n) f'(x_n) + \frac{(x^* - x_n)^2}{2!} f''(x_n) + \dots \quad (3.7)$$

Si x_n est proche de la solution exacte x alors on peut négliger le carré de l'erreur ϵ_n (où $\epsilon_n = x^* - x_n$) et tronquer la sommation au premier ordre, on obtient la relation approximative :

$$f(x_n) + (x^* - x_n) f'(x_n) = 0 \quad (3.8)$$

(1) Cette méthode est attribuée au mathématicien, physicien et astronome anglais Issac Newton (1642-1727). Toutefois, c'est Raphson qui publiait en 1690 la formule utilisée actuellement.

une approximation de l'erreur est donc :

$$\varepsilon_n = - \frac{f(x_n)}{f'(x_n)} \quad (3.9)$$

on peut donc considérer qu'un meilleur estimé de x^* sera :

$$x_{n+1} = x_n + \varepsilon_n \quad (3.10)$$

D'un point de vue géométrique, l'idée consiste à approximer $f(x)$ par sa tangente. A partir d'une condition initiale x_0 , on génère une séquence de points donnés par la formule :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (n=0, 1, \dots) \quad (3.11)$$

L'algorithme est donné par :

étape 1 : étant donnée une condition initiale x_0 .

étape 2 : poser $i = 0$.

étape 3 : calcul de $f(x_i)$ et $f'(x_i)$.

étape 4 : $x_i = x_i - f(x_i)/f'(x_i)$.

étape 5 : test d'arrêt : si oui aller à étape 7 sinon
aller à étape 5.

étape 6 : $i = i + 1$, aller à étape 4.

étape 7 : fin.

3.4 CRITERE D'ARRET DES ITERATIONS :

Il est nécessaire de choisir un critère de précision pour pouvoir arrêter le processus itératif. Le critère peut être basé soit sur l'erreur absolue tolérée ε_a , soit sur l'erreur relative

tolérée ϵ_r ; on peut en proposer quelques uns :

$$\begin{aligned} |x_n - x_{n-1}| &< \epsilon_a \\ |(x_n - x_{n-1}) / x_n| &< \epsilon_r \\ |f(x_n) - f(x_{n-1})| &< \epsilon_a \\ |(f(x_n) - f(x_{n-1})) / f(x_n)| &< \epsilon_r \end{aligned} \tag{3.12}$$

Dans notre cas le dernier critère a été choisit.

Dans tous les cas, il est sage de limiter le nombre d'itérations possible en arrêtant la simulation lorsque $n > n_{\max}$.

3.5 CONDITIONS DE CONVERGENCE :

La convergence est un réel problème dans la méthode de Newton, et des cas de divergence peuvent survenir (rejet à l'infini, attraction-répulsion ...[4]).

La condition suffisante de convergence est celle de Fourier :

$$\begin{cases} f''(x) \neq 0 \\ f(x_0) f''(x_0) > 0 \end{cases} \tag{3.13}$$

En pratique la convergence dépend du bon choix de la condition initiale dont nous n'avons a priori aucune information. Elle doit être suffisamment proche de la solution exacte. On verra dans le troisième chapitre que dans notre cas on peut la localiser avec une bonne approximation.

3.6 ORDRE DE CONVERGENCE DE LA METHODE DE NEWTON : [4,6]

Définition : Une méthode de résolution est dite d'ordre P si :

$$\text{si } \lim_{n \rightarrow \infty} \frac{\varepsilon_{n+1}}{(\varepsilon_n)^P} = \text{cte} \neq 0 \quad (3.14)$$

avec $\varepsilon_n = x^* - x_n$

L'ordre de convergence de la méthode choisie est quadratique, (c.à.d. P=2) ce qui lui confère une puissance et une supériorité à toutes les autres méthodes.

Preuve :

A partir de l'équation (3.11) on tire l'expression de l'erreur

$$\varepsilon_{n+1} = \varepsilon_n + \frac{f(x_n)}{f'(x_n)} \quad (3.15)$$

Réécrivons l'équation (3.15), il vient :

$$f(x^*) = f(x_n) + \varepsilon_n f'(x_n) + \frac{(\varepsilon_n)^2}{2} f''(\xi) = 0$$

$$\text{avec } \xi \in]x_n, x^*[\quad (3.16)$$

soit :

$$\frac{f(x_n)}{f'(x_n)} = -\varepsilon_n - \frac{(\varepsilon_n)^2}{2} \frac{f''(\xi)}{f'(x_n)}$$

d'où, par l'équation (3.15) :

$$\varepsilon_{n+1} = -\frac{f''(\xi)}{2f'(x_n)} \varepsilon_n^2 \quad (3.17)$$

D'après la définition de l'équation (3.14), si $f''(x)$ n'est pas

nulle, la méthode est d'ordre deux.

Remarque : Les résultats précédents ne sont plus valables si la solution est de multiplicité $r > 1$. On démontre dans ce cas que la convergence est linéaire.[5]

Preuve Dans Le Cas D'une Racine Double :

$$c.à.d \quad f(x) = (x-x^*)^2 g(x) \quad \text{avec} \quad g(x^*) \neq 0 \quad (3.18)$$

cherchons la dérivée :

$$f'(x) = (x-x^*) [2g(x) + (x-x^*) g'(x)] \quad (3.19)$$

remplaçons dans l'équation (3.17), on a :

$$\begin{aligned} \epsilon_{n+1} &= \frac{f''(\xi)}{2\epsilon_n [2g(x_n) + \epsilon_n g'(x_n)]} \epsilon_n^2 \\ \Rightarrow \quad \epsilon_{n+1} &= \frac{f''(\xi)}{2[2g(x_n) + \epsilon_n g'(x_n)]} \epsilon_n \quad c.g.f.d \end{aligned} \quad (3.20)$$

Ordres De Convergences De Quelques Méthodes :

Substitutions successives	1
Régula-Falsi	1
Illinois	1.442
Sécante	$(1+\sqrt{5})/2$
Newton-Raphson	2

3.7 EXTENSION AUX SYSTEMES D'EQUATIONS NON-LINEAIRES :

Un tel système se présente sous la forme :

$$\left\{ \begin{array}{l} f_1(x_1, x_2, \dots, x_N) = 0 \\ \dots \dots \dots \\ f_N(x_1, x_2, \dots, x_N) = 0 \end{array} \right. \quad (3.21)$$

Où les f_i sont des fonctions non-linéaires des variables $x_i (i=1, 2, \dots)$. On réécrit le système (3.21) dans la notation vectorielle :

$$\mathcal{F}(X) = \vec{0} \quad (3.22)$$

où $X = (x_1, x_2, \dots, x_N)^T$

Notons par $X^* = (x_1^*, x_2^*, \dots, x_N^*)$ la solution exacte du système.

Si chaque fonction est continue et continuellement dérivable, on procède à un développement en série de Taylor au voisinage d'un estimé X^k :

$$f_i(X^*) = f_i(X^k) + \sum_{j=1}^N \left. \frac{\partial f_i(X)}{\partial x_j} \right|_{X=X^k} (x_j^* - x_j^k) + \frac{1}{2!} \sum_{j=1}^N \sum_{r=1}^N (x_j^* - x_j^k) (x_r^* - x_r^k) \left. \frac{\partial^2 f_i(X)}{\partial x_j \partial x_r} \right|_{X=X^k} + \dots ; \quad (3.23)$$

$i=1, \dots, N$

Si X^k est assez proche de X^* , les termes de degrés supérieur à deux peuvent être négligés.

Comme $f_i(X^*) = 0$, l'équation (3.23) devient :

$$\sum_{j=1}^N \frac{\partial f_i(X)}{\partial X_j} \Big|_{X=X^k} (x_j^* - x_j^k) = f_i(X^k) \quad (3.24)$$

On définit alors la matrice carrée $N \times N$, d'éléments :

$$E_{ij}^k = \frac{\partial f_i(X)}{\partial X_j} \Big|_{X=X^k}, \quad i, j=1, \dots, N, \quad (3.25)$$

appelée JACOBIENNE.

L'idée est de remplacer X^* par X^{k+1} on a alors le système :

$$E^k \cdot \Delta X^k = F(X^k) \quad (3.26)$$

$$\text{avec } \Delta X^k = X^{k+1} - X^k$$

$$\text{et } F = (-f_1, \dots, -f_N)^t$$

L'équation (3.26) est un système linéaire qui peut être résolu par des méthodes directes ou itératives.

Ainsi le nouvel estimé est :

$$X^{k+1} = X^k + \Delta X^k \quad (3.27)$$

Conditions De Convergence : [4]

Il a été démontré que la suite des itérés de Newton converge sous les conditions suffisantes suivantes :

1. Si les fonctions $f_i(X)$ admettent des dérivée partielles premières continues dans un convexe contenant la solution X^* .

2. Si la jacobienne est non singulière dans un voisinage de X^* .

3. Si le vecteur initiale X^0 est suffisamment proche de X^* .

3.8 METHODE DE NEWTON DITE "Amortie" :

L'introduction d'un paramètre de pondération $0 < \lambda < 1$ tel que :

$$X^{k+1} = X^k + \lambda \Delta X^k \quad (3.28)$$

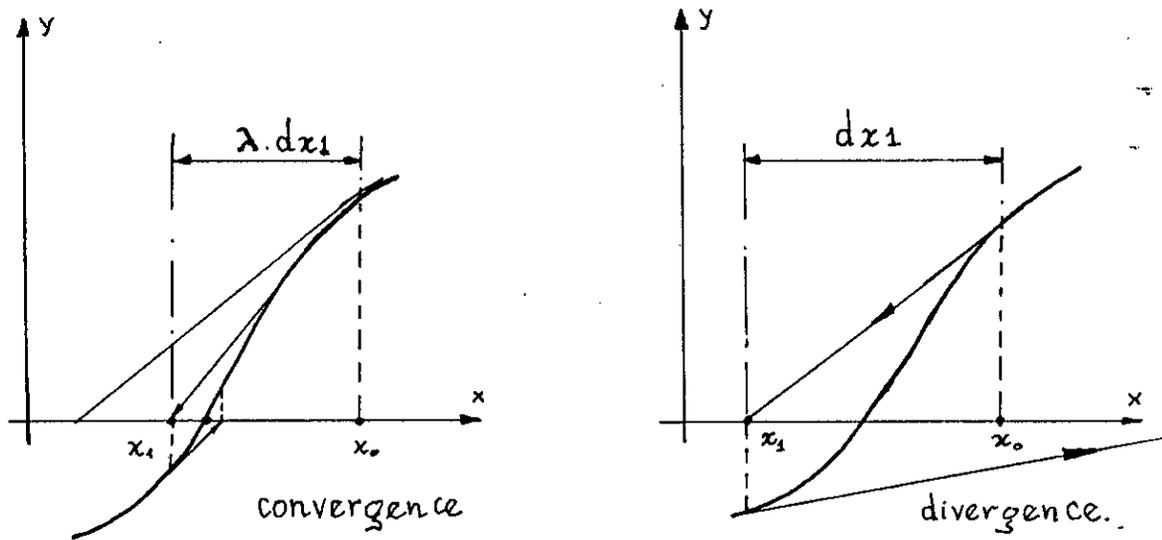


Figure 3.3 : Méthode de Newton amortie

Ceci a l'avantage d'accélérer la convergence et dans quelques cas d'éviter la divergence. On voit mieux l'effet de λ sur la figure (3.2).

Réglage Automatique De Lambda :

On choisit le paramètre λ de telle façon que la valeur d'une norme G de F à une itération soit plus petite que si on avait pas effectué sur λ un changement. Le paramètre λ est augmenté ou diminué selon le cas.

Algorithme :

données : $\lambda_0 < 1$; $r > 1$;

Etape 1 : poser $k=1$, $\lambda=\lambda_0$,

Etape 2 : si $G(\lambda/r) < G(\lambda)$ aller à étape 3 sinon aller à étape 5.

Etape 3 : $\lambda=\lambda/r$; $G(X^{k+1})=G(\lambda/r)$, aller à étape 5.

Etape 4 : calcul de $(X^k)=G(\lambda)$; augmenter λ jusqu'à ce que
 $G(\lambda) < G(X^k)$ ou que $\lambda=1$; $G(X^{k+1})=G(\lambda)$;

Etape 5 : $k=k+1$; test d'arrêt; si oui aller à étape 6 sinon
retour à étape 2.

Etape 6 : fin.

Notons par $G(\lambda) = G(X+\lambda\Delta X)$.

Une autre variante est la méthode de Newton dite "modifiée" qui consiste à ne calculer la jacobienne qu'à la première itération et l'utiliser pendant toutes les itérations suivantes afin de réduire le nombre de calculs. Toutefois, la rapidité de la convergence est ralentie par rapport à la méthode classique.

3.9 CONCLUSION :

A regard des ordres de convergence des différents algorithmes, le choix a porté sur la méthode de Newton-Raphson. A travers ce chapitre nous avons étudié en détail cette méthode pour la résolution des équations et systèmes d'équations non-linéaires.

Son principal avantage, par rapport aux autres méthodes, est sa convergence quadratique qui permet d'atteindre plus rapidement la

solution finale. L'algorithme risque de diverger lorsque l'estimé initial n'est pas assez proche de la solution. Un autre inconvénient est le nombre important de calculs de chaque itération dans le cas des systèmes d'équations non-linéaires. Nous avons aussi d'autres variantes de la méthode qui peuvent promouvoir la convergence.

CHAPITRE 4

4.1 INTRODUCTION :

Le présent chapitre est consacré au problème théorique de la PWM avec élimination sélective d'harmoniques (SHE). Notre étude s'inspire des travaux fait par Patel et Hoft [2]. Le choix de cette technique par rapport au contrôle optimal (EOC) peut se justifier par le fait que la solution est indépendante de la charge puisqu'on s'intéresse à la tension de charge et non au courant, et aussi du fait que la résolution est relativement simple (i.e., consomme moins de temps sur ordinateur, [16]).

Le but de la SHE est d'éliminer un nombre déterminé d'harmoniques indésirables. Cette technique nécessite une étude numérique avant la réalisation, ceci limite son application du fait que le calcul des angles exactes de commutations ne peut se faire en temps réel (on-line) par microprocesseur faute des performances de ce dernier, par contre on peut les stocker dans des mémoires (LUT).

Après développement des formules, nous appliquerons la méthode de Newton-Raphson traitée dans le troisième chapitre dans une première étape. Dans une deuxième étape, nous approximerons les résultats trouvés.

Enfin, beaucoup de figures sont illustrées, elles aideront à bien comprendre le cheminement suivi.

4.2 EXPRESSION DES TENSIONS :

En triphasé lorsque la charge est équilibrée (i.e somme des courants de chaque phase est nulle), les harmoniques multiples de trois sont inopérants [8]. Soit le cas de l'onduleur de la figure (4.1) :

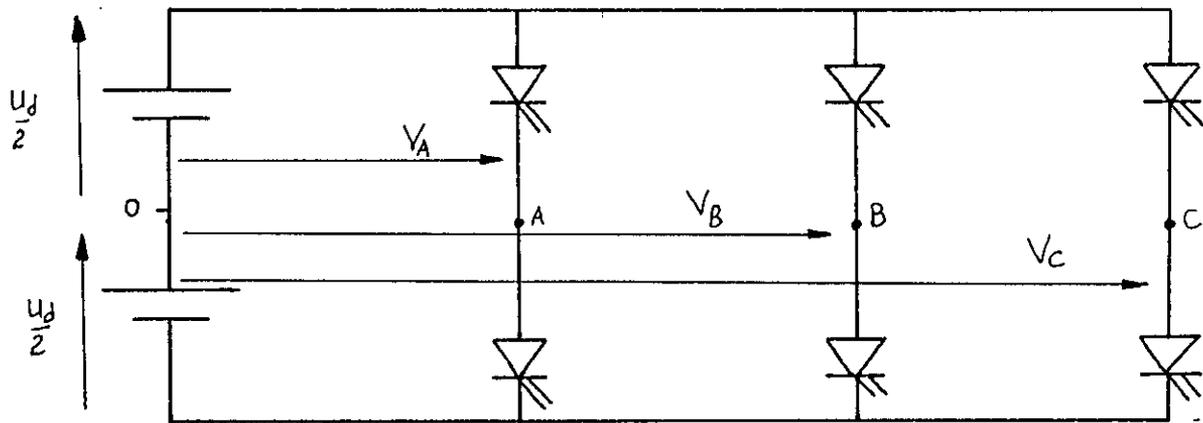


Figure 4.1 : Onduleur triphasé en demi-pont.

"o" point milieu fictif.

On tirera avantage d'un choix convenable des symétries dans l'onde résultante, figure (4.2).

Les tensions de phase sont déphasées de 120° i.e :

$$\begin{aligned} V_{bo}(\omega t) &= V_{ao}(\omega t - 2\pi/3) \\ V_{co}(\omega t) &= V_{ao}(\omega t + 2\pi/3) \end{aligned} \quad (4.1)$$

En plus nous prendrons une symétrie de glissement :

$$V(\omega t) = -V(\omega t + \pi) \quad (4.2)$$

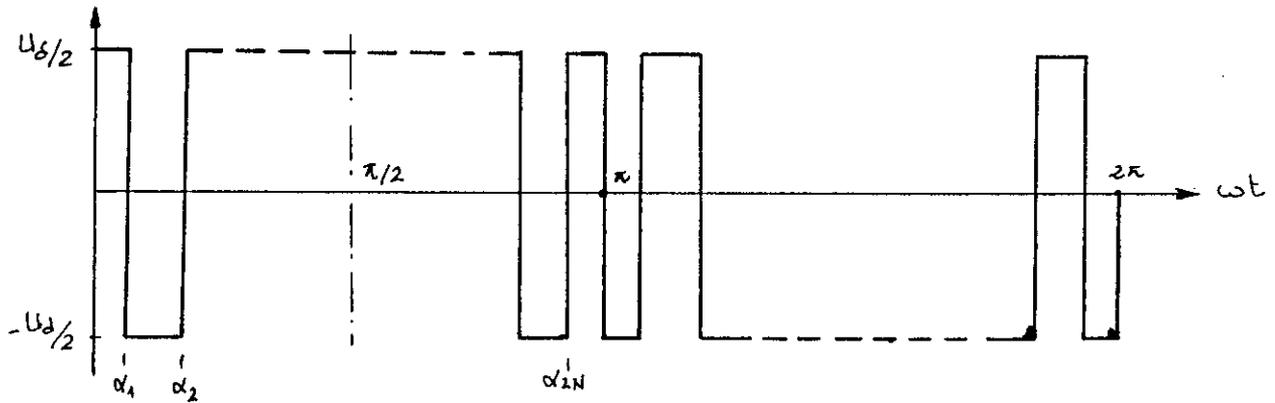


Figure 4.2 : Forme d'onde de sortie

Comme nous sommes en régime périodique nous aborderons le problème par une décomposition de la tension de sortie en série de Fourier [2] :

$$v(\omega t) = \sum_{n=1}^{\infty} [a_n \sin(n\omega t) + b_n \cos(n\omega t)] \frac{U_d}{2} \quad (4.3)$$

$$\text{avec} \quad a_n = \frac{1}{\pi} \int_0^{2\pi} v(\omega t) \sin(n\omega t) d(\omega t) \quad (4.4)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} v(\omega t) \cos(n\omega t) d(\omega t)$$

comme $\alpha_0 < \alpha_1 < \dots < \alpha_{2N+1}$ avec $\alpha_0 = 0$ et $\alpha_{2N+1} = \pi$, on obtient :

$$a_n = \frac{2}{\pi} \sum_{k=0}^{2N} (-1)^k \int_{\alpha_k}^{\alpha_{k+1}} \sin(n\omega t) d(\omega t) \quad (4.5)$$

Vu la symétrie par rapport à π , le développement ne contient pas d'harmoniques pairs. Si en plus nous choisissons une symétrie par rapport à $\pi/2$, i.e :

$$v(\omega t) = \dagger v(\pi - \omega t) \quad (4.6)$$

les termes en cosinus (b_n) sont nuls. Finalement :

En particulier pour le fondamental ($n = 1$)

M est appelé indice de modulation.

$$a_n = \frac{4}{n\pi} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(n\alpha_k) \right] \quad (4.7)$$

$$a_1 = M = \frac{4}{\pi} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(\alpha_k) \right] \quad (4.8)$$

$$-\frac{4}{\pi} \leq M \leq \frac{4}{\pi}$$

Dans la SHE pure, on égale alors à zero les N harmoniques qui viennent juste après le fondamental, tandis que dans la SHE asservie, on élimine (N-1) harmoniques indésirables et on donne une valeur au fondamental. Nous obtenons donc un système d'équations non-linéaires en fonctions trigonometriques qu'on résoud par une méthode numérique. La solution n'est pas évidente et devrait satisfaire à la condition $a_1 < a_2 < \dots < a_N$. Quant à la convergence, elle dépend surtout du choix des conditions initiales. Les harmoniques restant de rang assez élevé sont facilement filtrés.

Un avantage de la SHE est que pour une fréquence donnée, le rang du premier harmonique non éliminé est presque double de celui de la PWM engendrée (sinusoidale). Les inconvénients^{de la SHE} sont :

- La difficulté analytique de résolution du système d'équations non-linéaires et le temps de calcul nécessaire pour les systèmes de grand envergure.
- les harmoniques non éliminés peuvent présenter d'assez importantes amplitudes.

4.3 ELIMINATION SELECTIVE D'HARMONIQUE PURE PSHE [2] :

Dans ce chapitre les expressions des fonctions f_i sont déterminées et données par :

$$f_i = \frac{4\pi}{n_i} \left[1 + 2 \sum_{k=1}^N (-1)^k \cos(n_i \alpha_i) \right] = 0 \quad (4.9)$$

$$\Rightarrow 1 + 2 \sum_{k=1}^N (-1)^k \cos(n_i \alpha_i) = 0$$

quant à la jacobienne elle est donnée par :

$$E = \frac{\partial \mathcal{F}}{\partial X} = \begin{bmatrix} 2n_1 \sin(n_1 \alpha_1) & -2n_1 \sin(n_1 \alpha_2) & \dots & +2n_1 \sin(n_1 \alpha_N) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 2n_N \sin(n_N \alpha_1) & -2n_N \sin(n_N \alpha_2) & \dots & +2n_N \sin(n_N \alpha_N) \end{bmatrix} \quad (4.10)$$

La solution exacte devrait satisfaire à la condition :

$$0 < \alpha_1 < \alpha_2 < \dots < \alpha_N < 90^\circ \quad (4.11)$$

Pour résoudre ce système Nous appliquerons alors la méthode de Newton-Raphson amortie.

4.3.1 Organigramme :

Il est donné par la figure (4.3).

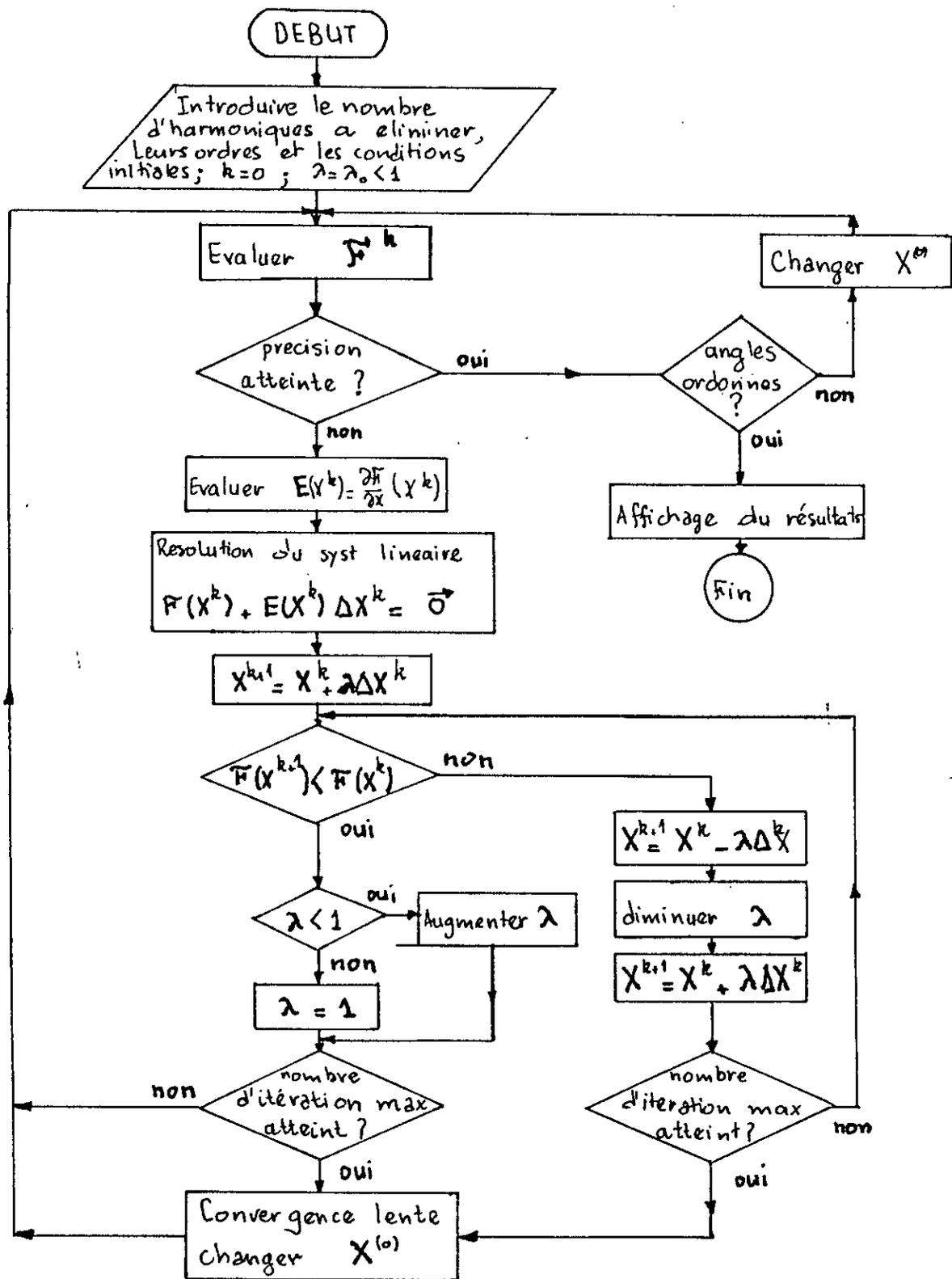


Figure 4.3 : Application de la méthode de Newton Amortie au système d'équations non-linéaires avec réglage automatique de λ .

4.3.2 Résultats :

L'élimination des harmoniques indésirables dans le cas de la PSHE a donné des résultats satisfaisant, où l'examen des spectres pour différent ordres montre que ces harmoniques ont été éliminés avec une précision de 10^{-20} ; reste que le premier harmonique non éliminé ~~est à~~ ^{possède} une amplitude assez importante qui est l'inconvénient de cette technique.

Ainsi les angles de commutations pour quelques ordres sont :

N=2 (5,7 éliminés)

$\alpha^1=16.247$, $\alpha_2=22.068$

N=5 (5,7,11,13,17 éliminés)

$\alpha_1=6.798$; $\alpha_2=17.302$; $\alpha_3=21.033$; $\alpha_4=34.670$; $\alpha_5=35.998$

N=8 (5,7,11,13,17,19,23,25 éliminés)

$\alpha_1=6.193$; $\alpha_2=10.456$; $\alpha_3=18.407$; $\alpha_4=21.057$; $\alpha_5=30.498$

$\alpha_6=31.864$; $\alpha_7=42.448$; $\alpha_8=42.914$

N=11 (5,7,11,13,17,19,23,25,31,35,37 éliminés)

$\alpha_1=4.051$; $\alpha_2=9.261$; $\alpha_3=12.280$; $\alpha_4=18.554$; $\alpha_5=20.640$

$\alpha_6=27.850$; $\alpha_7=29.14$; $\alpha_8=37.121$; $\alpha_9=37.781$; $\alpha_{10}=46.34$

$\alpha_{11}=46.564$

Le spectre d'harmoniques correspondant à chaque ordre cité est donné par la figure (4.4).

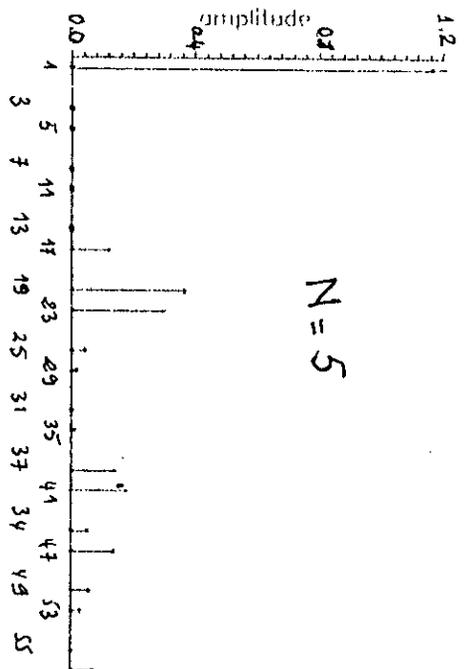
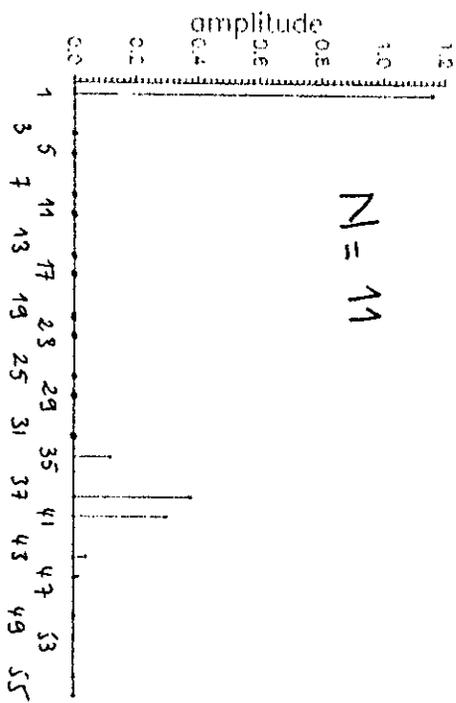
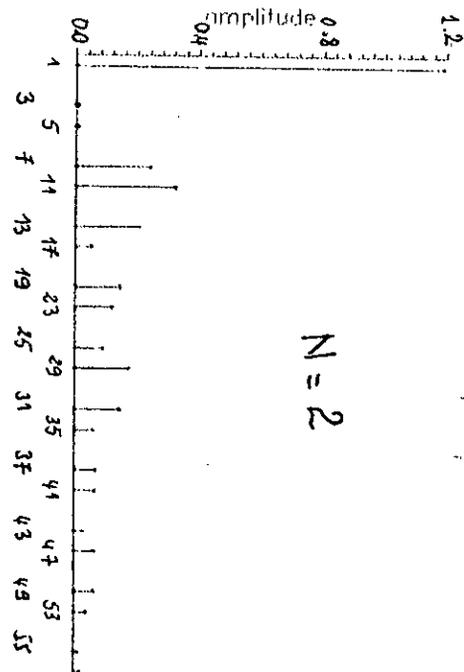
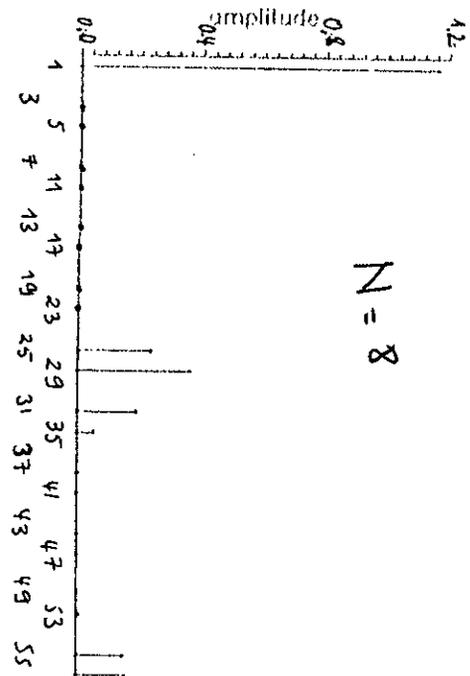


Figure 4.4 : Spectre des harmoniques éliminés dans le cas de la PSHE.

Constatation :

Les angles suivent une certaine loi en fonction de l'ordre du système, et cela en faisant la distinction entre les ordres pairs et impairs comme le montre la figure (4.5 a et b).

La trajectoire particulière que suit chaque angle nous a amenée à approximer les angles dans le but de déterminer une loi pour générer les conditions initiales; ainsi diminuer le temps de calcul des angles exactes.

Après plusieurs essais, il est apparu que l'approximation des angles par des fonctions puissances était la plus proche. L'expression des angles est :

$$\alpha_i = a_i N^{b_i} \quad (4.12)$$

Il fallait en plus faire une distinction selon la parité des indices "i" des angles. Ensuite nous avons tracé les a_i et b_i en fonction de i .

A leur tour, les trajectoires des coefficients a_i et b_i ont été approximées par des fonctions puissances et logarithmes.

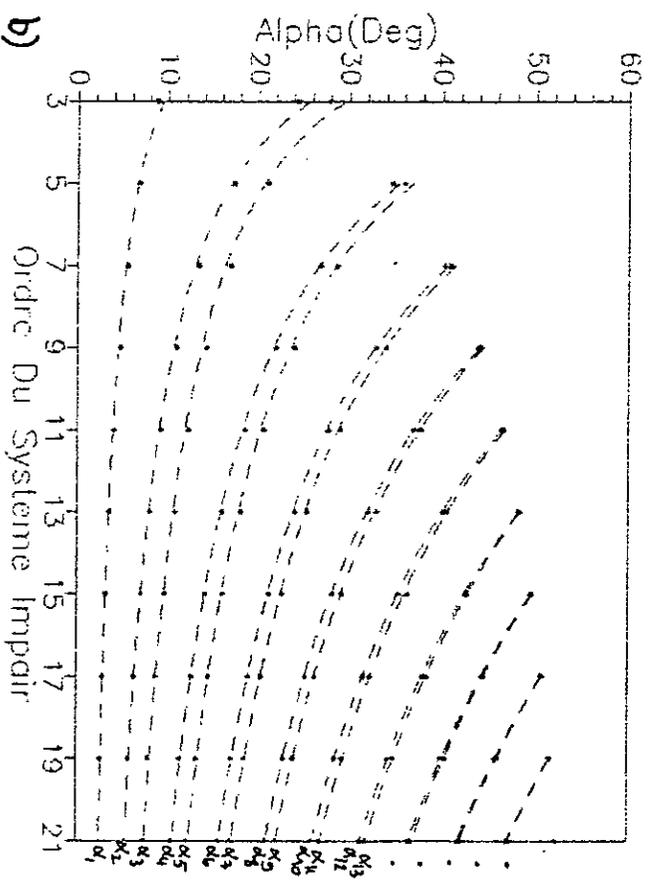
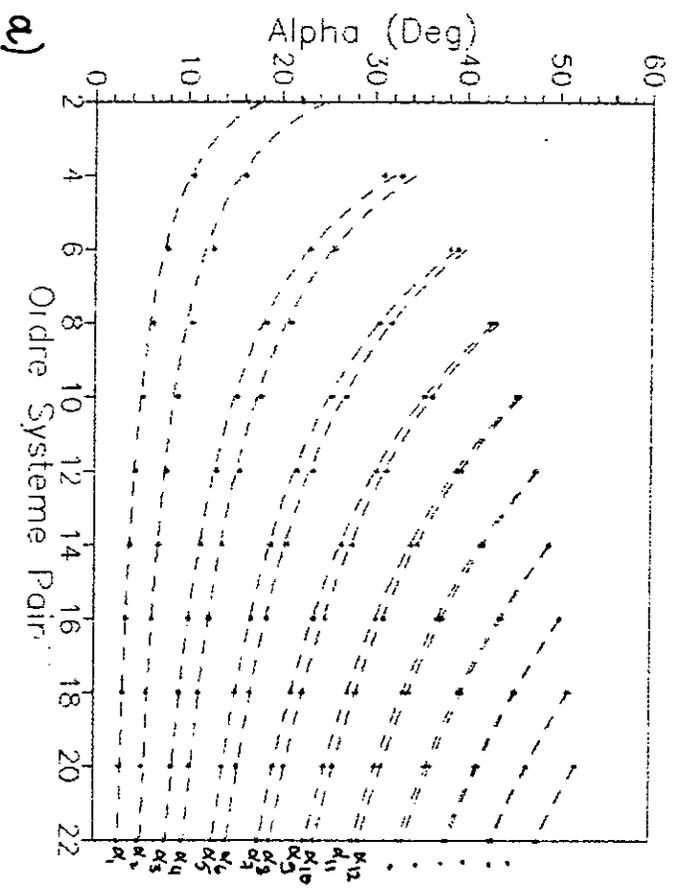


Figure 4.5 : Variation des angles α_i en fonction de l'ordre du système (a: pair ; b: impair) ; (PSHE).

Nous pouvons résumer le travail fait dans le schéma de la figure (4.6).

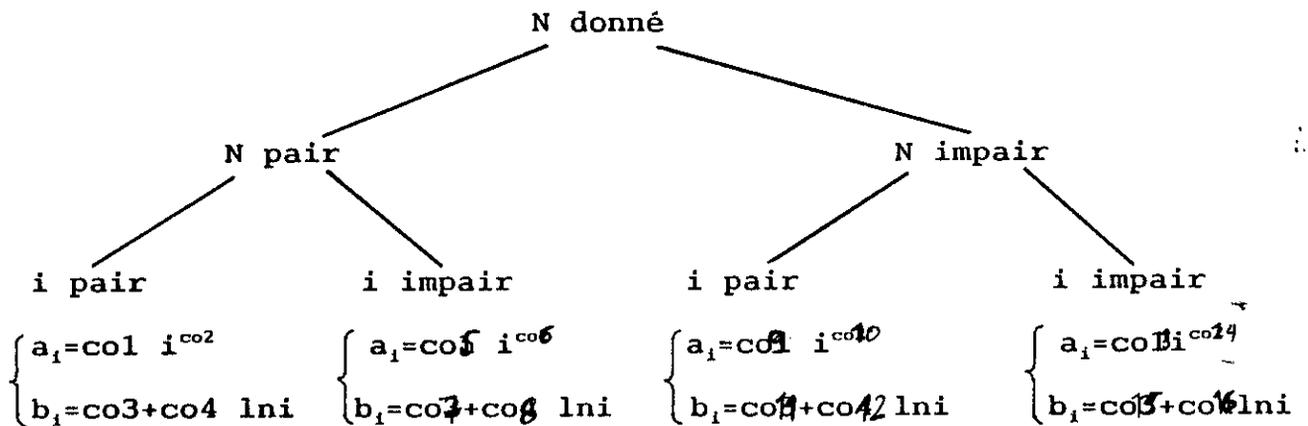


Figure 4.6 structure en arbre montrant la procédure suivie des approximations

Où les "co" sont des coefficients calculés.

co1 = 13.5024 ; co2 = 1.35367 ; co3 = 0.602196 ; co4 = 0.09845
 co5 = 30.4955 ; co6 = 1.10299 ; co7 = 0.9707 ; co8 = 0.0395882
 co9 = 28.5033 ; co10 = 1.13132 ; co11 = 0.773831 ; co12 = 0.0468122
 co13 = 18.0241 ; co14 = 1.23721 ; co15 = 0.644476 ; co16 = 0.081231.

Remarque : il est à signaler que la solution n'est pas unique. Nous pouvons citer comme exemple le cas de l'ordre quinze (15), où nous avons trouvé deux solutions différentes, mais une seule était choisie.

4.3.3 Organigramme De Génération Des Angles :

Il est donné sur la fig (4.7).

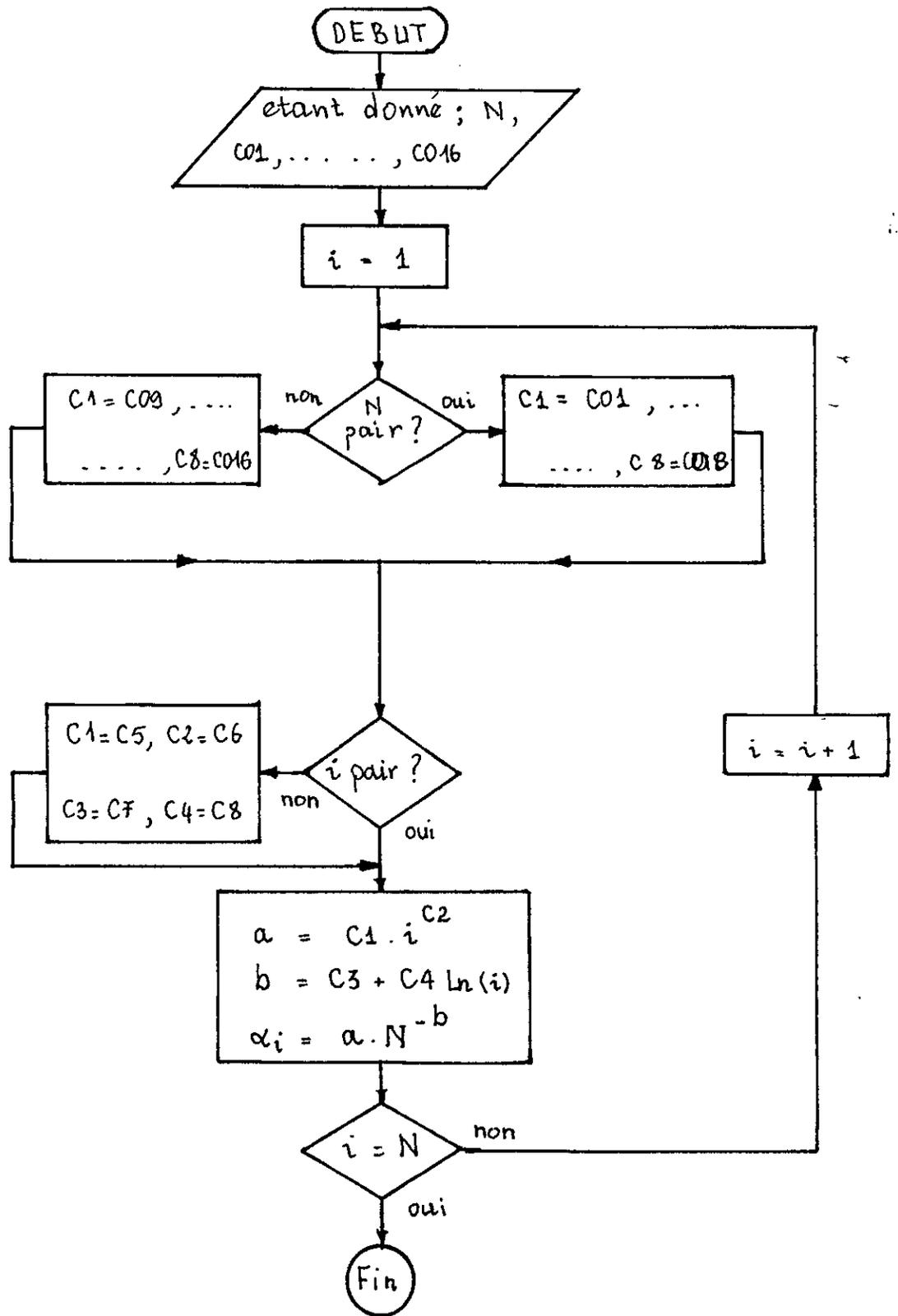


Figure 4.7 : Generation des angles dans le cas des la P.S.H.E.

4.4 ELIMINATION SELECTIVE D'HARMONIQUES AVEC ASSERVISSEMENT DU FONDAMENTAL :

Dans ce cas on élimine (N-1) harmoniques indésirables et on asservi le fondamental à une valeur. Le système non-linéaire devient

$$\begin{aligned} h_1 &= c^{te} & (4.13) \\ h_i &= 0 \quad , \quad i=2, \dots, N \end{aligned}$$

Pour retrouver le système (2.) il suffit de mettre :

$$\begin{aligned} f_1 &= h_1 - c^{te} = 0 & (4.14) \\ f_i &= h_i = 0 \quad , \quad i=2, \dots, N \end{aligned}$$

Ainsi pour chaque valeur du fondamental, on trouve une solution correspondante.

Remarque :

- 1°) L'ordre des harmoniques à éliminer est de la forme $6k + 1$, $k= 1, 2, \dots, N$. Chaque harmonique de tension (et de courant) d'ordre $6k-1$ et $6k+1$ cause un couple pulsé, on souhaite toujours les éliminer par paires. Pour cela N est souvent choisi impair [16].
- 2°) Il a été trouvé et prouvé [16] qu'il n'existe pas plus de deux solutions pour le système (4.13), désignées par solutions A et B.

4.4.1 Cas Des Systèmes D'ordre Pair :

Les angles exactes sont obtenus en résolvant le système (4.16) pour divers valeurs du fondamental et plusieurs ordres.

L'absence d'une information à priori sur le lieu exacte des angles nous a amené à choisir des conditions initiales arbitraires en les répartissant régulièrement dans l'intervalle $[0, 90^\circ]$. Pour minimiser la probabilité de divergence nous prendrons, au départ, une faible valeur de λ , puis progressivement nous augmentons λ . En suivant cette procédure, malgré quelques cas de divergence, nous avons aboutie à un résultat.

La figure (4.8) montre la variation des angles en fonction de l'indice de modulation pour les ordres $N=2, 4, 6, 8$.

Constatations :

- C1 - Lorsque le fondamental croît, les intervalles de conduction se retrécissent.
- C2 - Les angles suivent des trajectoires qui peuvent très bien être approximées à des droites (pentes constantes) sauf pour des valeurs élevées du fondamental ($M > 0.9$) où nous remarquons une diminution très rapide et non-linéaire de ces angles au voisinage du maximum du fondamental.
- C3 - Les droites correspondantes aux angles avec indices de même parités sont parallèles entre elles (mêmes pentes).
- C4 - Pour une amplitude du fondamental fixée (plus précisément la valeur zéro), les angles d'indices de mêmes parités sont séparés de :

$$\Delta = \frac{2 \times 60}{N} \quad \text{degrés} \quad (4.15)$$

- C5 - Comme conséquences de C1 et C2, il suffit de connaître α_1 et α_2 pour déduire $\alpha_3, \dots, \alpha_N$.

Le spectre pour chaque cas ($N=2, 4, 6, 8$) et pour quelques valeurs

du fondamental figure (4.9 - 4.12) montre que :

- 1 - La valeur du fondamental désirée est atteinte.
- 2 - Les harmoniques indésirables sont éliminés.
- 3 - Le premier harmonique non éliminé a une amplitude importante comme dans le cas d'élimination pure.

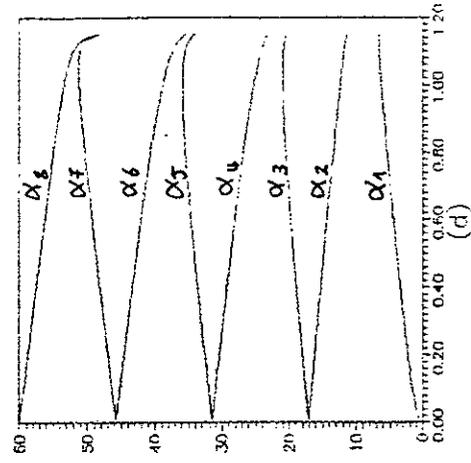
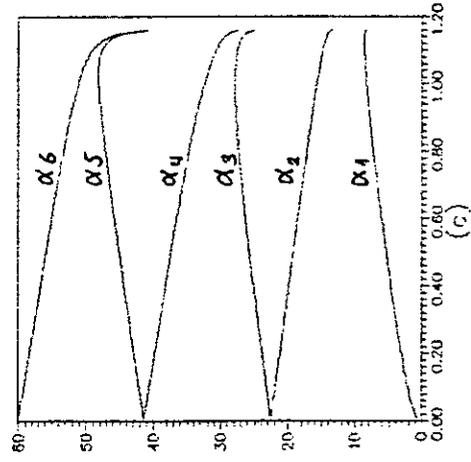
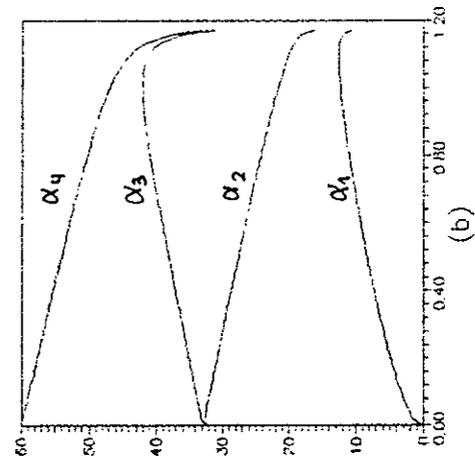
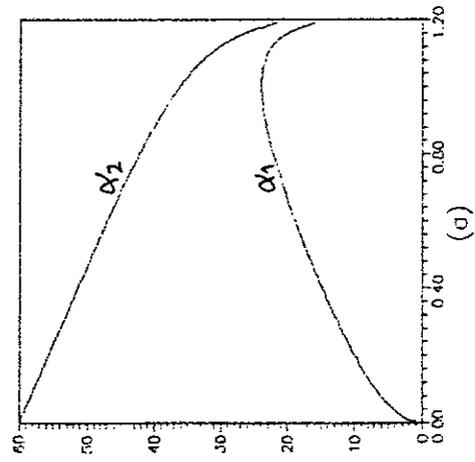


Figure 4.8 : Trajectoire des angles α_i en fonction de l'indice de modulation M . (ASHE); N pair :

a) $N=2$; b) $N=4$; c) $N=6$, d) $N=8$.

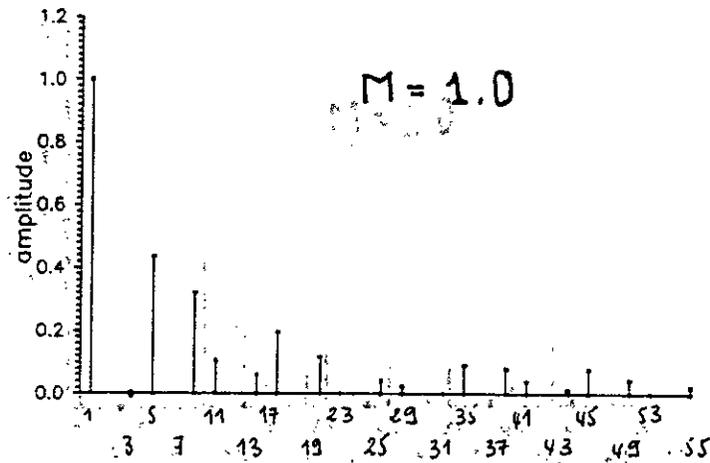
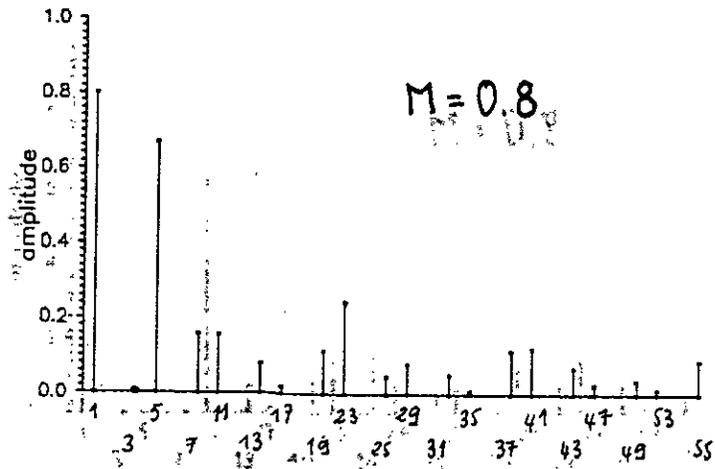
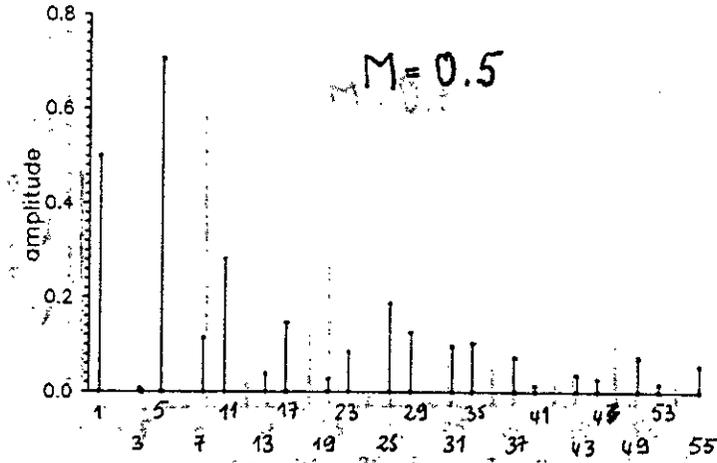
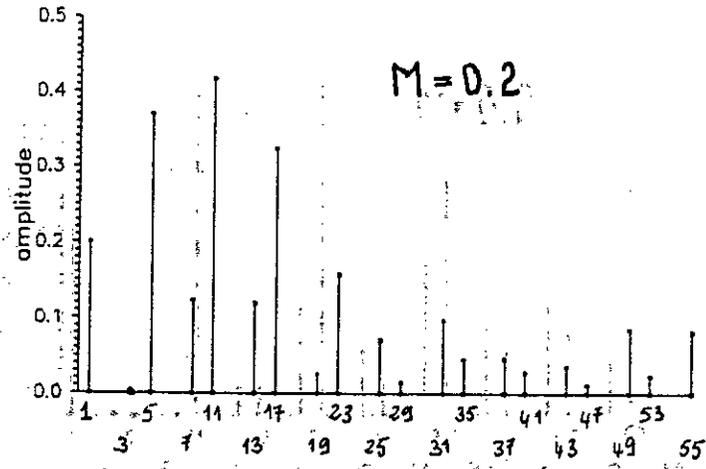


Figure 4.9 = Spectre des harmoniques éliminés pour l'ordre deux (2) (ASHE)

Propositions :

A partir des courbes donnant les angles en fonction du fondamental nous avons voulu, comme dans le cas d'élimination pure, aboutir à un algorithme qui nous permet de générer les conditions initiales assez proche de la solution exacte.

Comme, pour un ordre donné, les deux informations sur les pentes P1 (>0) et P2 (<0) de α_1 et α_2 respectivement, suffisent pour générer tous les angles, on se propose de tracer les variations de ces pentes en fonction de l'ordre N.

Dans le programme on a considéré les formules de P1 et P2 suivantes :

$$P1 = \frac{\alpha_1(M=0.9) - \alpha_1(M=0.2)}{0.9-0.2} \quad (4.16)$$

$$P2 = \frac{\alpha_2(M=0.9) - \alpha_2(M=0.2)}{0.9-0.2}$$

M étant l'indice de modulation.

L'organigramme de calcul de P1 et P2 est donné par la figure (4.13)

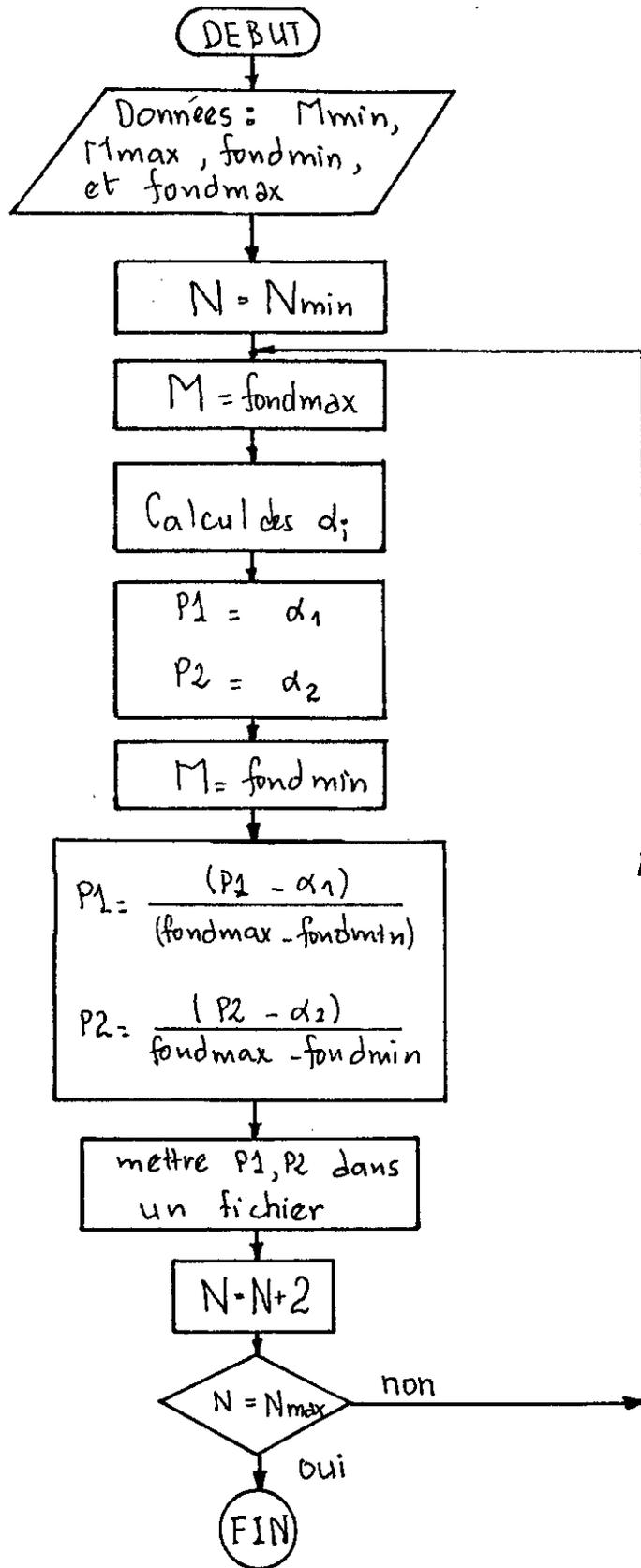


Figure 4.13 : Organigramme de calcul de P1 et P2

Résultats :

Les courbes résultantes sont données par la figure (4.14). L'étape suivante est d'approximer les trajectoires des pentes. Ainsi, la fonction puissance a été la plus adaptée.

Les expressions approximatives sont :

$$\begin{aligned} P1 &= co1 N^{-co2} & , N = 2,4,6,\dots & \quad (4.17) \\ P2 &= -co3 N^{-co4} & , N = 2,4,6,\dots & \end{aligned}$$

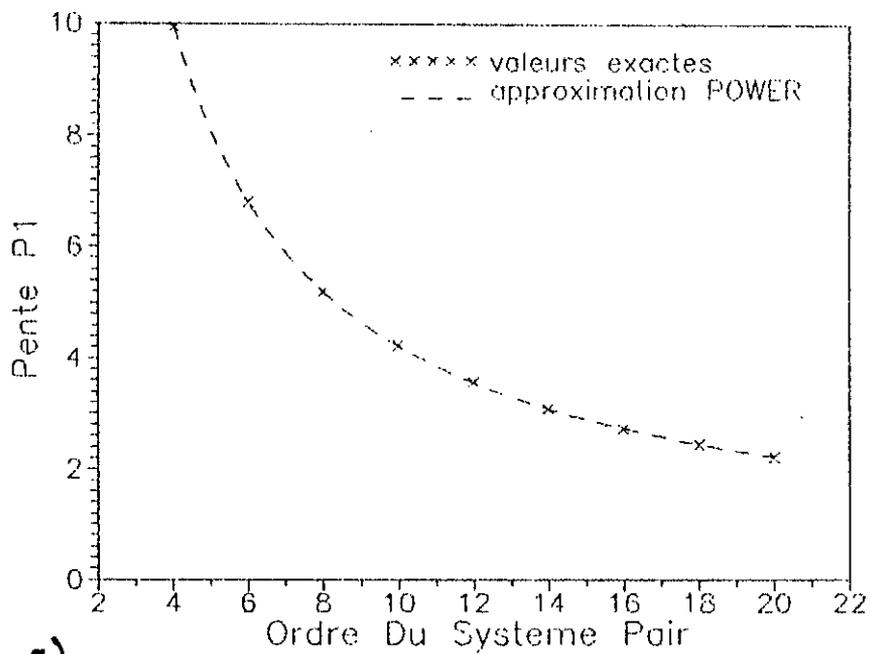
Mais si on commence de $N = 4$, l'approximation est encore plus proche. Et on a :

$$P2 = -co5 N^{-co6} \quad , N = 4,6,\dots \quad (4.18)$$

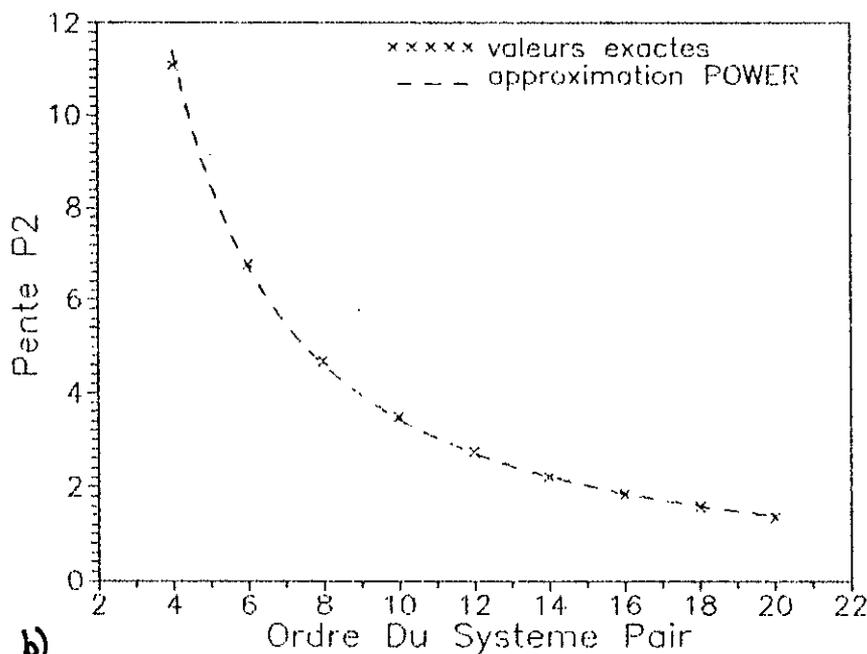
avec

$$\begin{aligned} co1 &= 36.8235 ; & co2 &= 0.94196 ; & co3 &= 59.6563 \\ co4 &= 1.24575 ; & co5 &= 69.8246 ; & co6 &= 1.30811. \end{aligned}$$

la figure (4.15) montre l'organigramme de génération des angles.



a)



b)

Figure 4.14 : Variation de P1 (a) et P2 (b) en fonction N (pair).

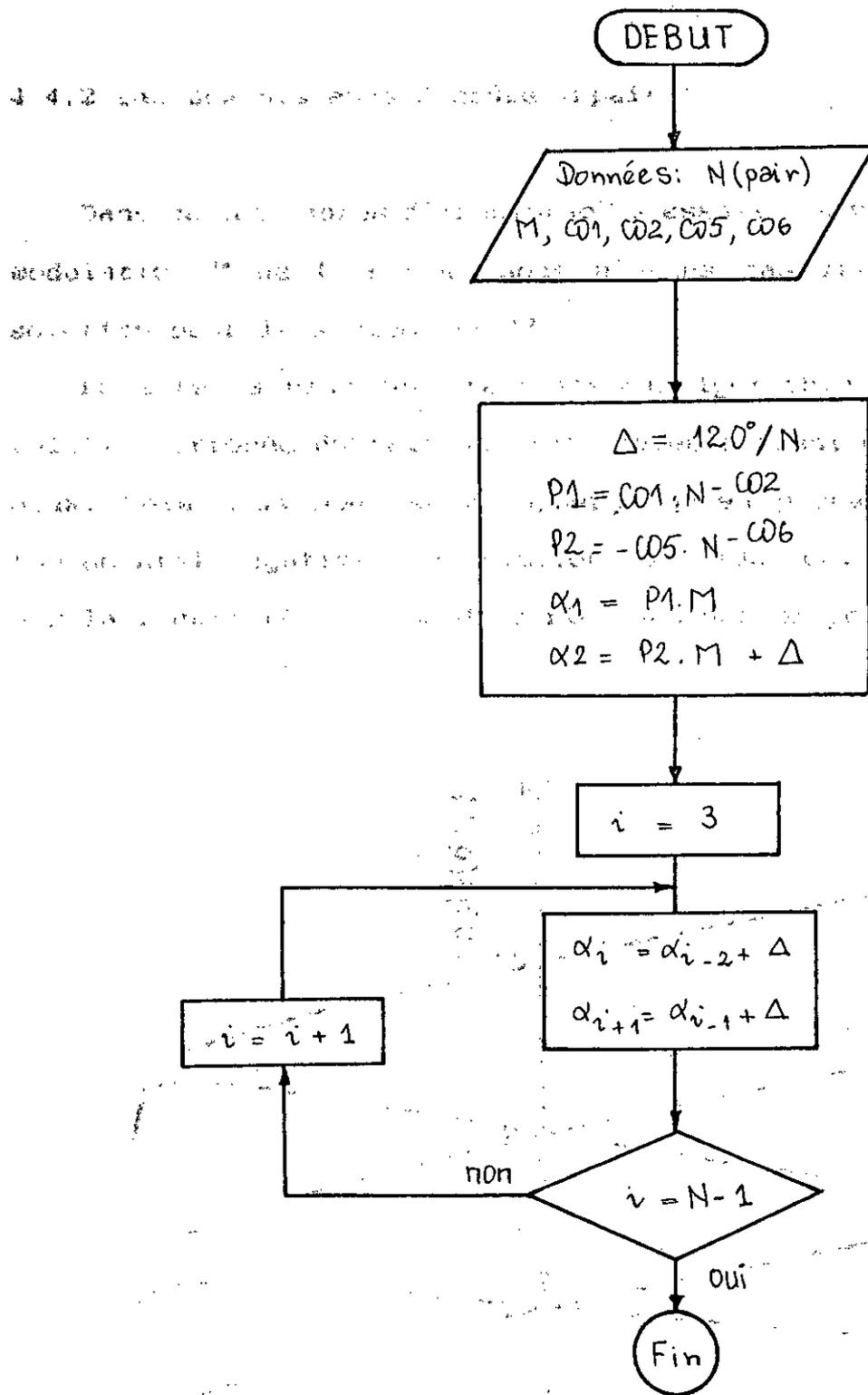


Figure 4.15 : Organigramme de generation des angles

pour des ordres pairs.

Chaque angle varie selon une droite. Deux angles successifs suivent des droites de pentes opposées. Le croisement se fait pour une valeur de M nulle, et lorsque ce dernier dépasse le zéro, chaque paires d'angles successifs se permutent. Quand le fondamental $V_1 = \pm 4U_d/2\pi$ on retombe sur l'onde carré pour laquelle:

$$\alpha_1 = \alpha_2 = \dots = \alpha_N = 0^\circ \text{ ou } 90^\circ \quad (4.19)$$

M positif ou négatif; c'est juste une différence mathématique. Pour la tension fondamentale, ceci equivaut à un décalage de phase de 180° , il n'a aucun effet sur la commande.

la figure (4.17) montre la trajectoire des angles en fonction de la valeur absolue de M pour différents ordres.

Les mêmes constatations peuvent se faire comme dans le cas des ordres pairs sauf que :

- 1 - Le signe des pentes s'inverse ($p_1 < 0$ et $p_2 > 0$).
- 2 - La séparation angulaire est donnée par :

$$\Delta = \frac{2 \times 60}{N+1} \quad (4.20)$$

Les observations concernant les spectres, dans le cas des ordres pairs, demeurent valables dans le cas des ordres impairs (voir les figures 4.18 - 4.21)

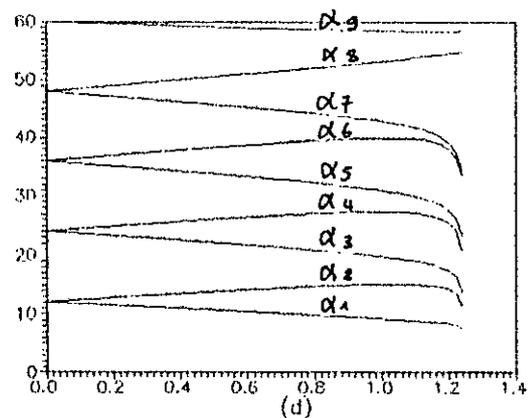
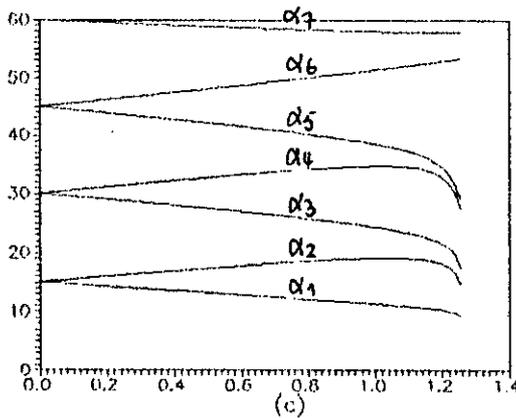
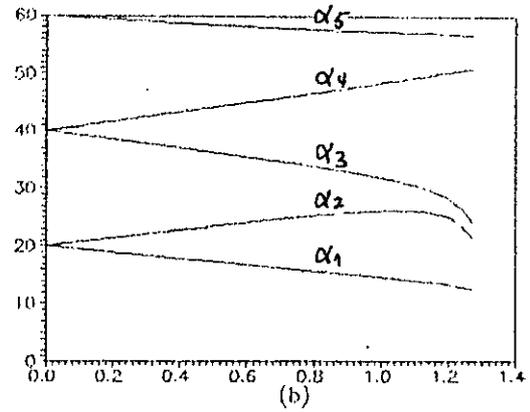
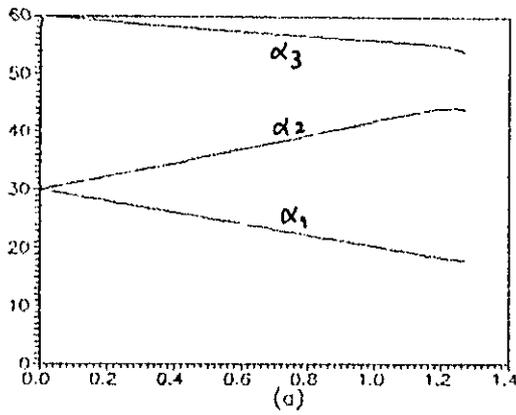


Figure 4.17 : Trajectoire des angles α_i en fonction de l'indice de modulation M . (ASHE); N impair.

a) $N=3$; b) $N=5$; c) $N=7$; d) $N=9$.

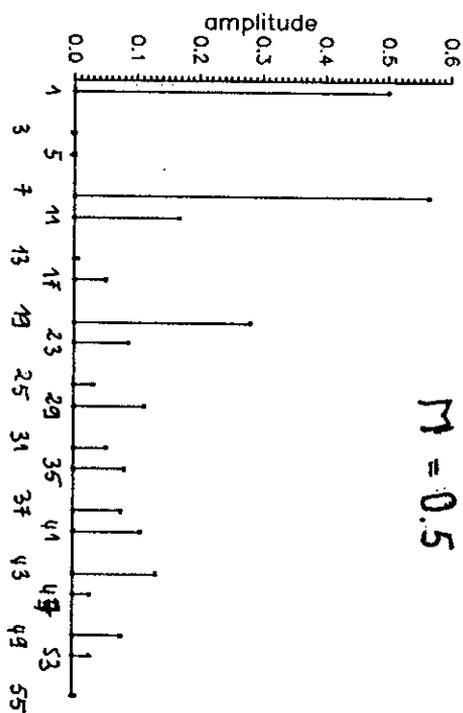
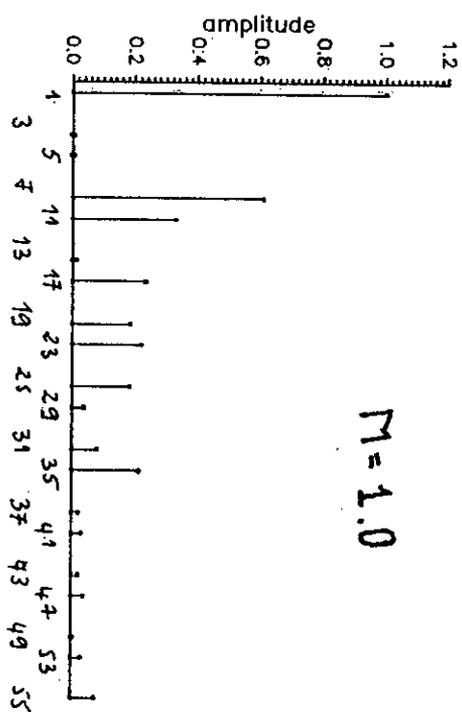
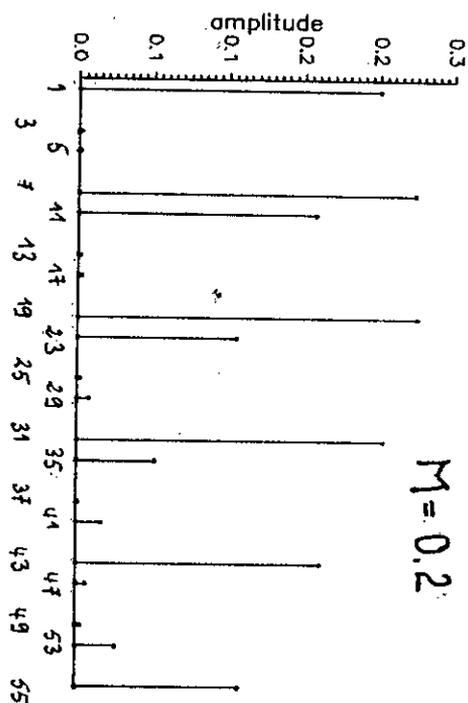
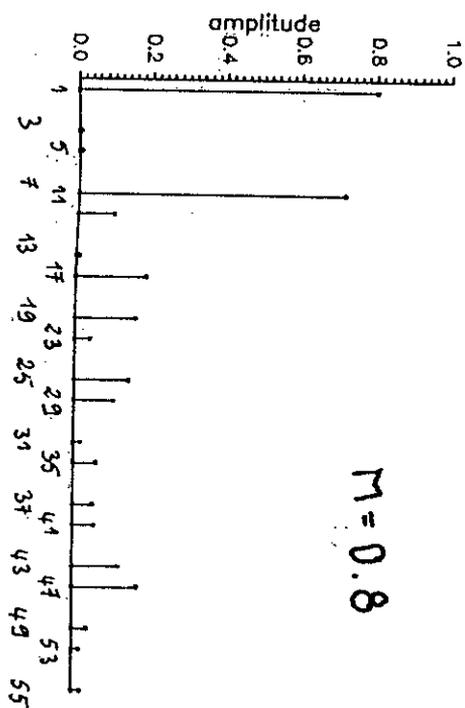


Figure 4.18 : Spectre des harmoniques éliminés pour l'ordre trois (3). (ASHE)

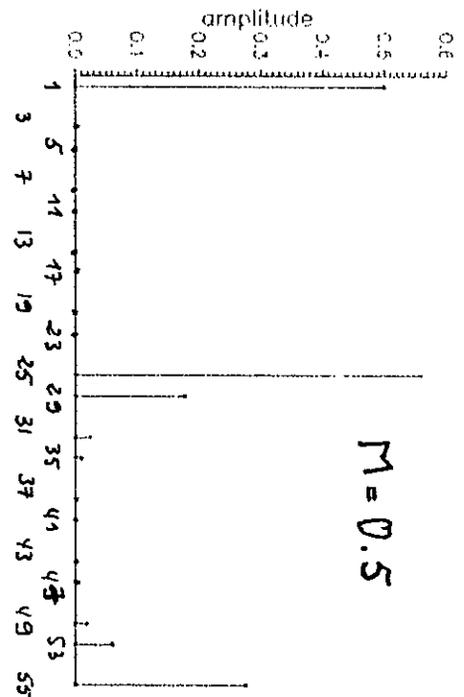
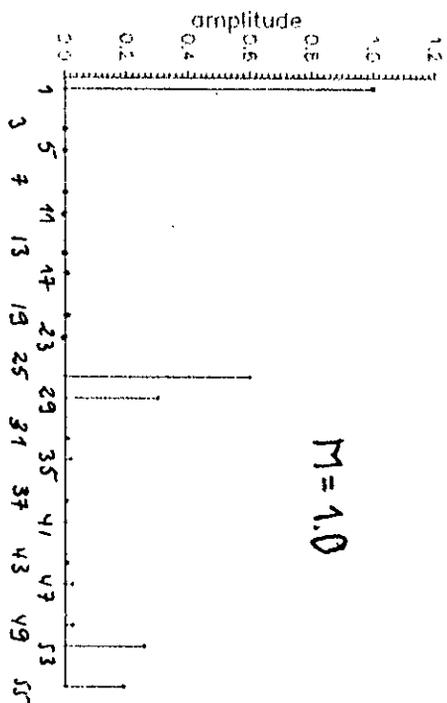
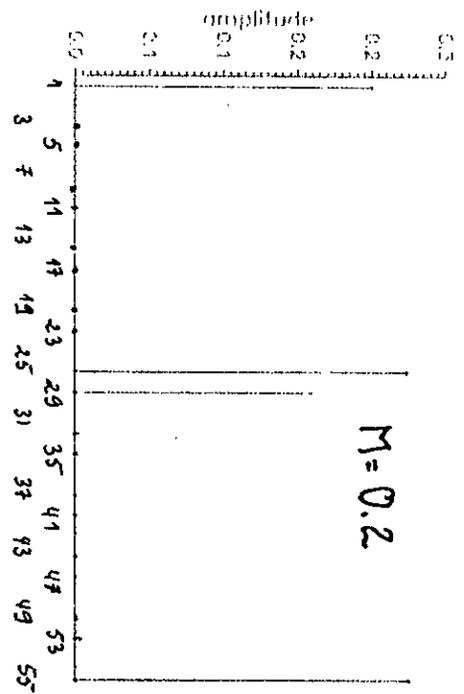
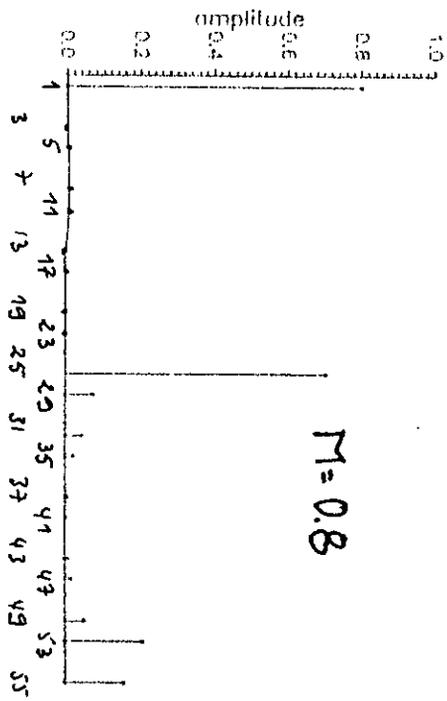


Figure 4.21 : Spectre des harmoniques éliminés pour l'ordre neuf (9). (ASHE).

La fig (4.22) montre les variations de P1 et P2 en fonction de l'ordre.

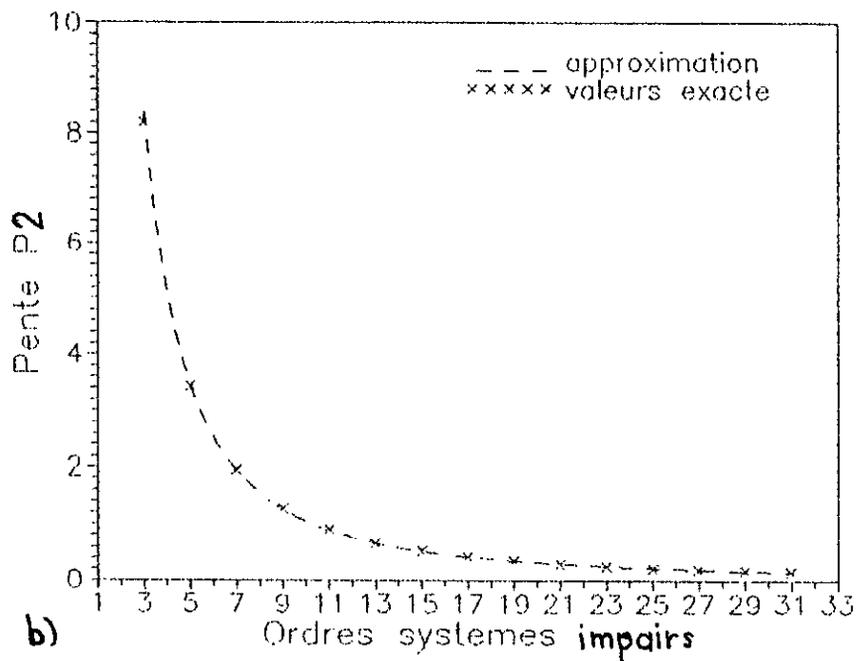
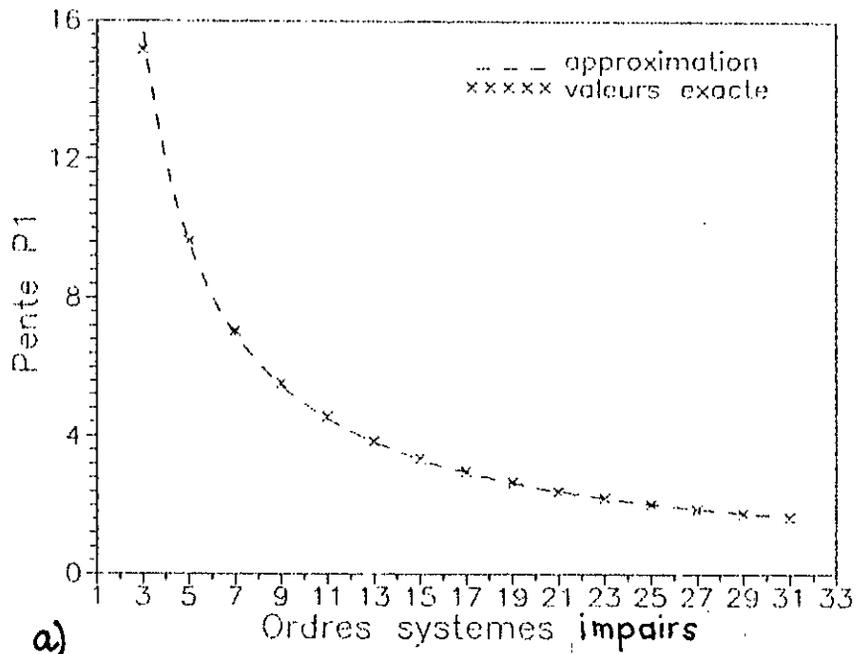


Figure 4.22 : Variation de P1 (a) et P2 (b) en fonction de N (impair).

Résultats :

Les mêmes organigrammes ont été suivis pour le calcul de P1 et P2 ainsi que les mêmes procédures d'approximation par des fonction puissance. Nous avons aboutit aux formules (figure 4.22) :

$$\begin{aligned} P1 &= -co1 N^{-co2} \\ P2 &= co3 N^{-co4} \end{aligned} \tag{4.21}$$

avec $co1 = 44.9129$; $co2 = 0.959911$; $co3 = 65.003$;
 $co4 = 1.73097$.

L'organigramme de génération des angles est donné par la figure
(4.23)

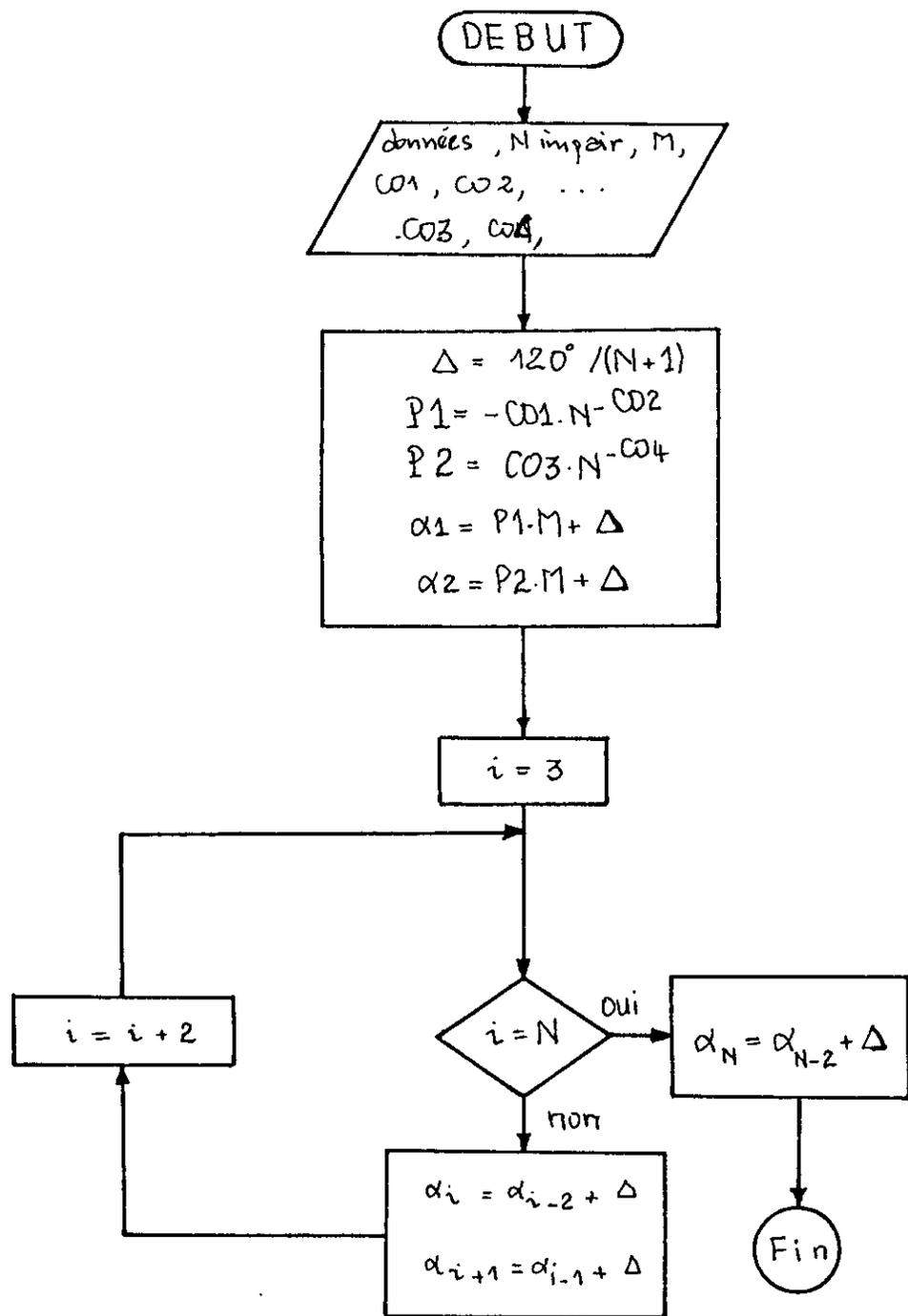


Figure 4.23 : Organigramme de generation des angles pour des ordres impairs.

4.5 CONCLUSION :

Les résultats précédents montrent que théoriquement il est possible d'atteindre une sortie presque sinusoidale. Le choix entre la PSHE et la ASHE dépend de l'application.

Les procédures d'approximation des résultats ont permis d'élaborer des algorithmes pouvant, pour un ordre et une amplitude du fondamental donnés, générer les angles α_i très proches de la solution exacte. Ces angles serviraient soit comme conditions initiales pour l'algorithme de Newton-Raphson, soit à être implémentés sur microprocesseur.

CHAPITRE 5

5.1 INTRODUCTION :

Dans ce chapitre, l'implémentation sur microprocesseur est présentée. On s'intéressera plus à la ASHE pour prendre en considération la variation du fondamental qui est proportionnelle à la variation de vitesse pour un fonctionnement à flux constant. Toutefois, ceci n'empêche pas d'utiliser la PSHE pour des systèmes non asservis.

Ces stratégies peuvent être implémentées offline ou online. La première méthode consiste à mémoriser dans des tables LUT (Look-Up Tables : LUT) les angles de commutations calculés auparavant comme expliqué au chapitre précédant, puis suivant la valeur de la fréquence présente à l'entrée, générer les angles correspondants. Dans la seconde méthode on génère en temps réel les angles de commutations calculés par des formules approximatives. Notons qu'on peut combiner les méthodes. Chaque cas sera étudié. L'idée et la manière d'implémentation sont expliquées et illustrées par synoptiques.

Enfin, dans la pratique la réalisation dépend de deux paramètres

- Le HARDWARE : dont on dispose, le CPU, le nombre de bits, l'horloge, le timer, les mémoires, ... etc.
- Le SOFTWARE : suivant la manière de développer l'idéesous formes de programmes et de l'adapter sur le matériel en exploitant au maximum ~~de~~ le microprocesseur.

5.2 IMPLEMENTATION OFF-LINE :

Ce type d'implémentation nécessite à priori des calculs numériques sur ordinateur pour déterminer les angles de commutations. Les résultats trouvés sont ensuite préprogrammés dans les mémoires EPROM du microprocesseur. Dans ce qui suit nous présentons une description de l'aspect de l'implémentation.

En respectant le schéma de la figure (5.1), l'axe des fréquences est divisé en plusieurs segments. Chaque segment $[f_i, f_{i+1}]$ correspond à un ordre N_i fixé. Lorsque N est fixé les angles ne changent que si le fondamental change.

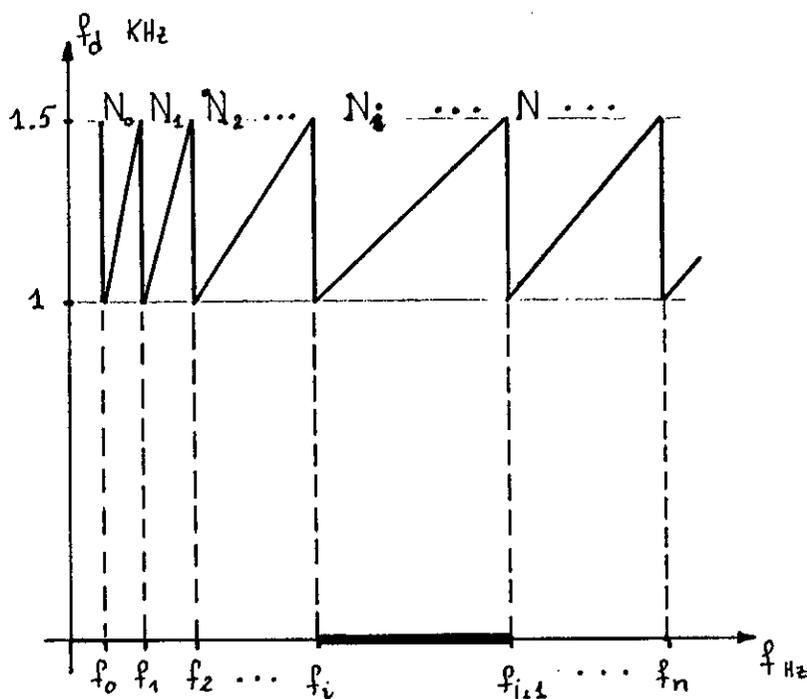


Figure 5.1 : Division de l'axe des fréquences en segments pour différents ordres.

Pour un fonctionnement avec un rapport V/f constant, à une fréquence $f \in [f_1, f_{1+1}]$ lui correspond une certaine valeur de l'indice de modulation et une solution d'angles unique. La fréquence est incrémentée d'un pas Δf .

L'idée est de mémoriser les angles pour chaque fréquence dans des tables (fig 5.2). Il est clair que la longueur des tables diffère d'un intervalle de fréquence à un autre puisque N change.

La valeur de la fréquence désirée est présentée à l'entrée. On doit lui adresser la mémoire contenant les angles correspondants. Il est astucieux alors, pour minimiser le temps de la recherche, de stocker les adresses de début des tables dans une mémoire.

En résumé, lorsque la fréquence désirée est imposée à l'entrée, elle aiguille la table des adresses, qui à son tour aiguille vers le début d'adresse de la table contenant les angles en degrés, fig(5.3).

L'aiguillage des adresses peut se faire par des multiplexeurs, mais il est plus logique et économique de confier cette tâche au microprocesseur afin de minimiser la circuiterie et utiliser à mieux la puissance les possibilités offertes par le

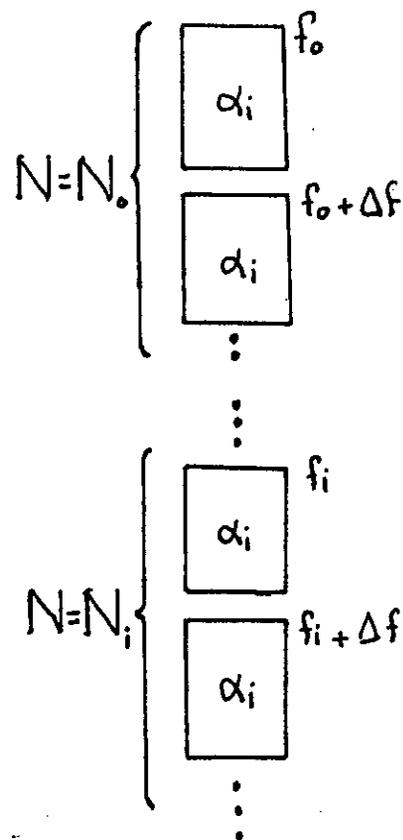


Figure 5.2 : mémorisation des angles dans des tables.

microprocesseur.

Ensuite c'est le CPU qui y introduira par programmes (Soft) les largeurs d'impulsions successivement dans le timer qui les convertit en intervalles de temps.

Il est possible de réduire le temps de calcul en exploitant les symétries à $\pi/2$ et à π . Le second quart d'onde ainsi que le demi cycle suivant peuvent être générés d'une manière similaire en complétant appropriément les états du signal de sortie.

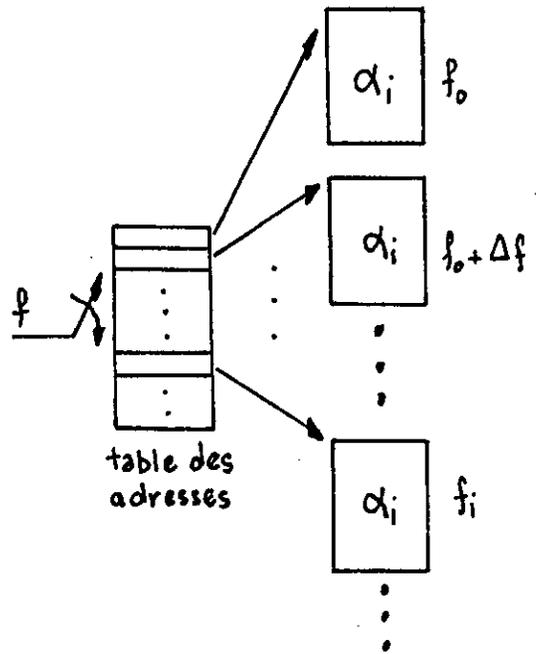


Figure 5.3: Aiguillage des tables de données par des tables d'adresses.

A une fréquence donnée f_i , une quantité angulaire de un degré (1°), lui correspond un temps t_i qui constituera un cycle d'horloge du timer pour cette fréquence.

Calcul De t_i :

t_i est donné par la formule :

$$t_i = \frac{1}{360^\circ \times f_i}$$

Le schéma synoptique de la configuration complète de l'implémentation est montré dans la fig (5.4).

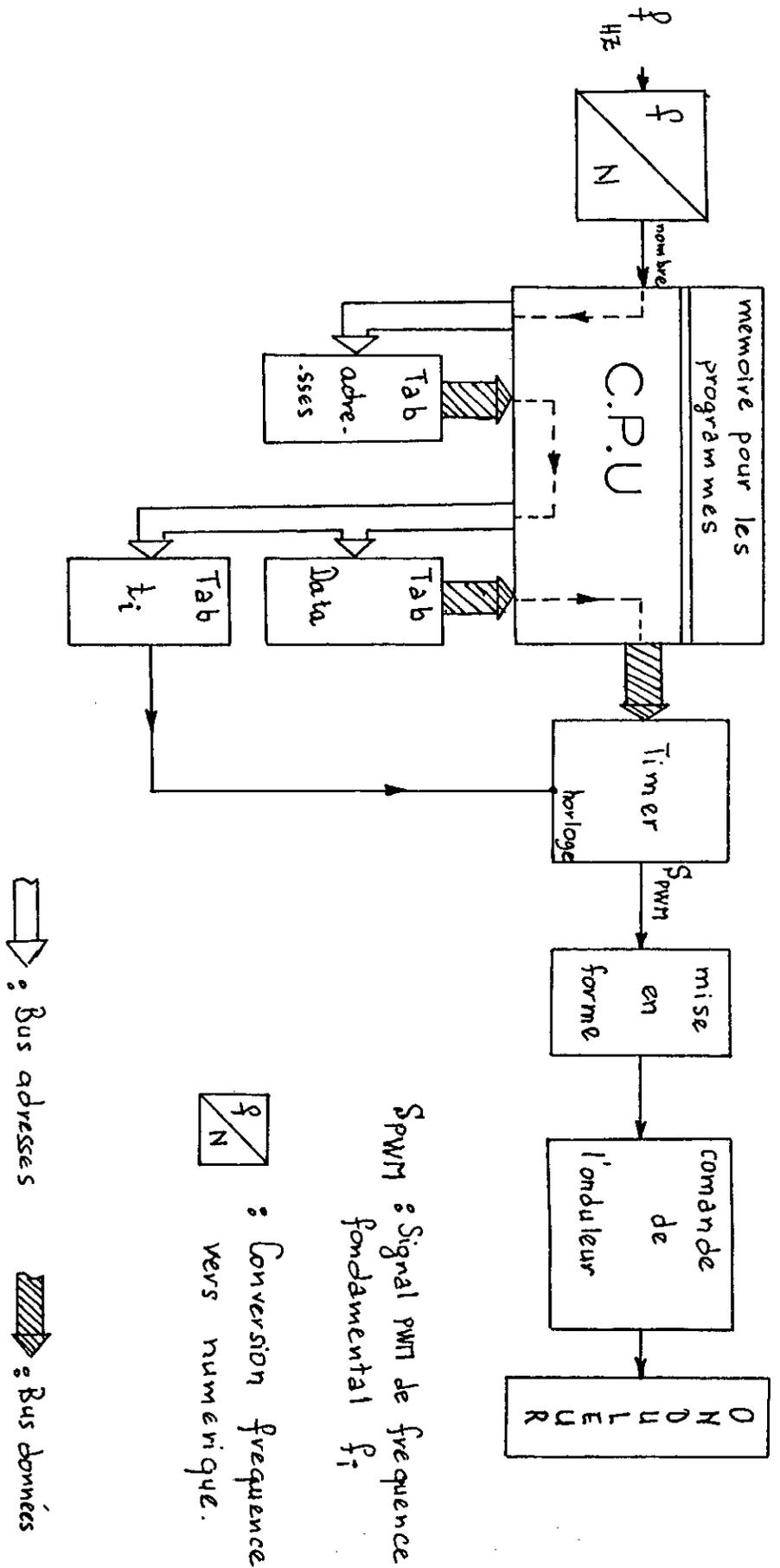


Figure 5.4 : Synoptique D'implémentation "Off-line" Des Angles Dans Des Mémoires.

5.3 IMPLEMENTATION ON-LINE :

Elle demande un effort et un temps de calcul plus considérable que dans le cas offline. En générale ceci est inacceptable dans quelques applications; particulièrement celles qui nécessitent des temps de réponse rapides (i.e. Application en temps reel). (annexe A).

Les procédures d'approximations trouvées sont exploitées. Aulieu de mémoriser les angles précalculés pour chaque ordre et pour chaque valeur de l'indice de modulation, il suffit de stocker les coefficients, les pentes et les séparations angulaires dans des mémoires ROM. On effectue alors les opérations arithmétiques des formules approximatives pour calculer tous les angles une seule fois. Ensuite on les stocke dans une mémoire RAM et on les charge à travers le microprocesseur dans le timer pour générer le signal PWM en temps réel de la même façon que dans le cas de l'implémentation offline.

Les opérations arithmétiques relativement complexes comme la multiplication et l'exponentiation peuvent être conçus soit en Hardware par des circuits spécialisés, soit par des programmes internes au microproceseur, mais qui consomment un temps de calcul pouvant être significatif. Dans une autre approche on envisage de stocker les valeurs de la fonction exponentielle dans des tables.

La figure (5.5) montre la configuration complète de l'implémentaion.

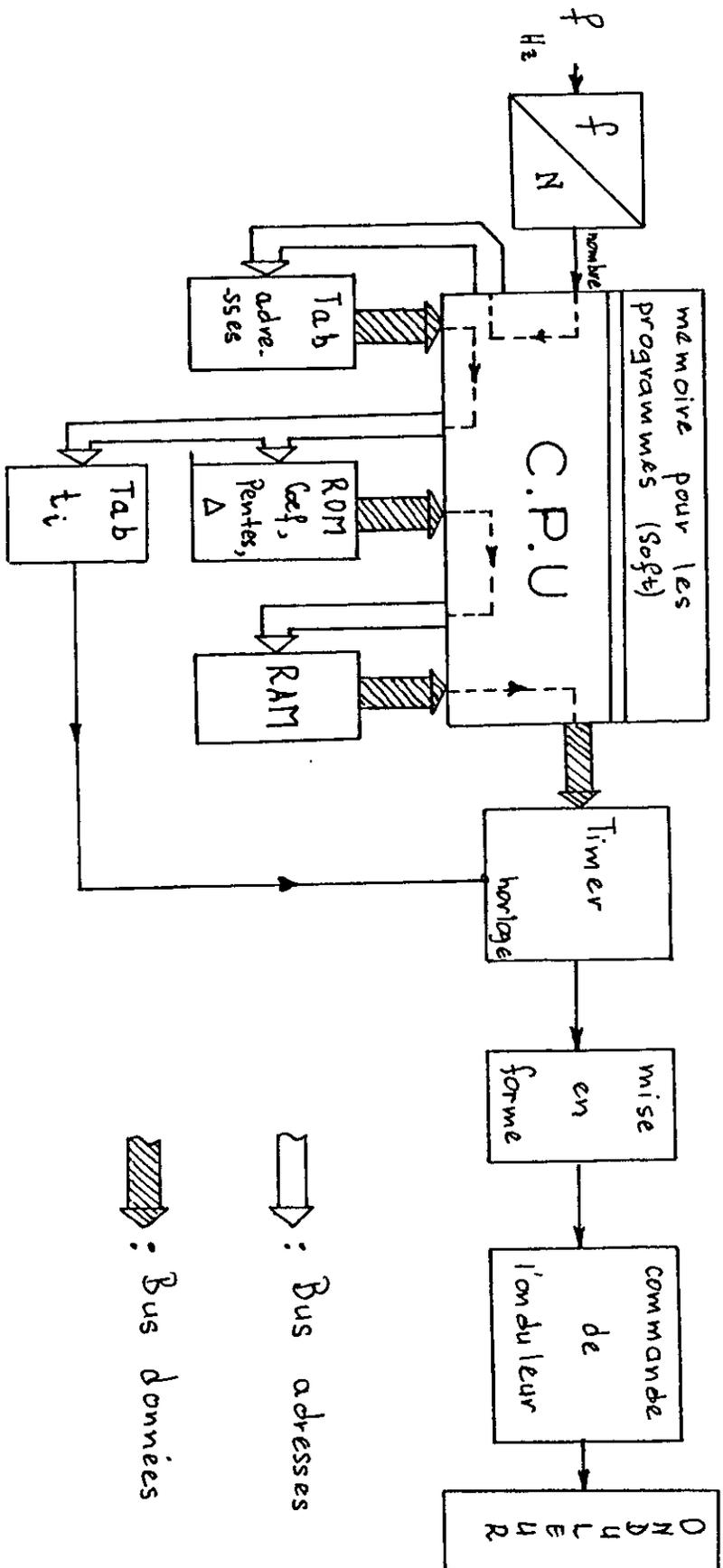


Figure 5.5 : Synoptique d'implémentation "On-line".

5.4 EXEMPLE DE LA MISE EN FORME DES SIGNAUX :

Soit le cas de la figure (5.6). Lorsque les thyristors Th_1 et Th_3 sont conducteurs, on a une tension positive en sortie, et lorsque c'est Th_2 et Th_4 qui sont conducteurs on a une tension négative.

L'amorçage de Th_1 et de Th_3 se fait aux instants des transitions montantes et celui de Th_2 et Th_4 aux instants des transitions descendantes, fig (5.6).

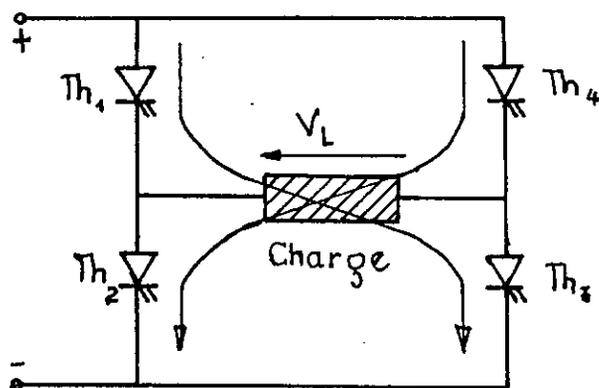


Figure 5.6 = Schéma simplifié d'un onduleur.

Après dérivation du signal carré PWM (S_{PWM}), on le sépare en deux signaux par des écréteurs de telle manière à ce que les impulsions positives commandent les thyristors Th_1 et Th_3 et que les impulsions négatives commandent les thyristors Th_2 et Th_4 . Les formes d'ondes des signaux sont montrées sur la figure (5.8). Le schéma de base est montré sur la figure (5.7).

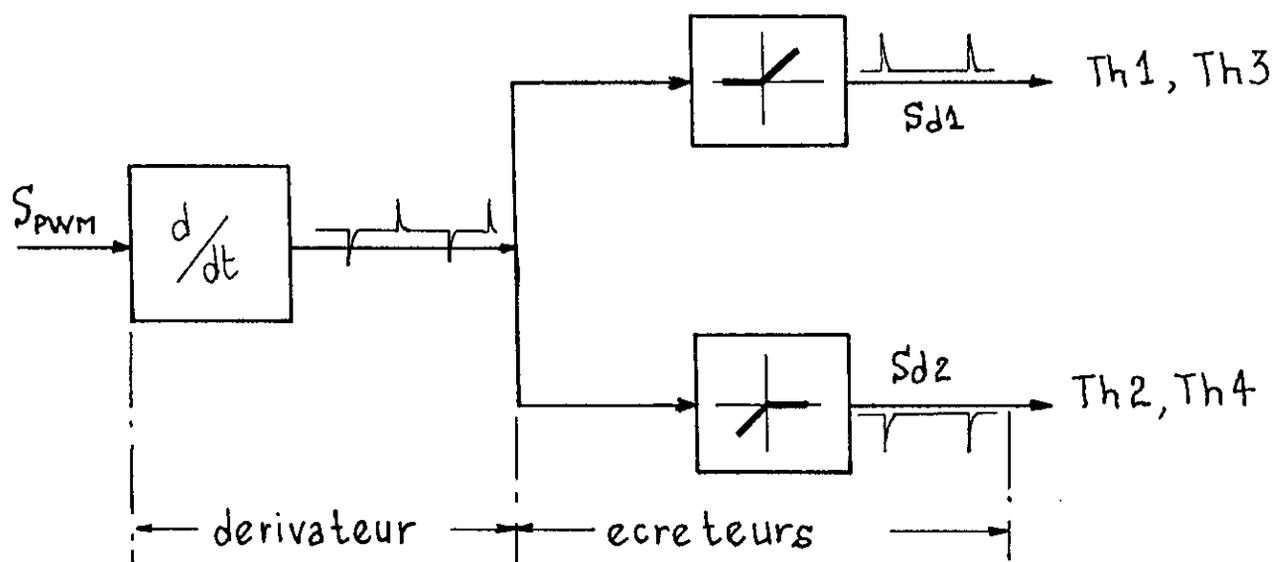


Figure 5.7 : Synoptique de la mise en forme des signaux de commande.

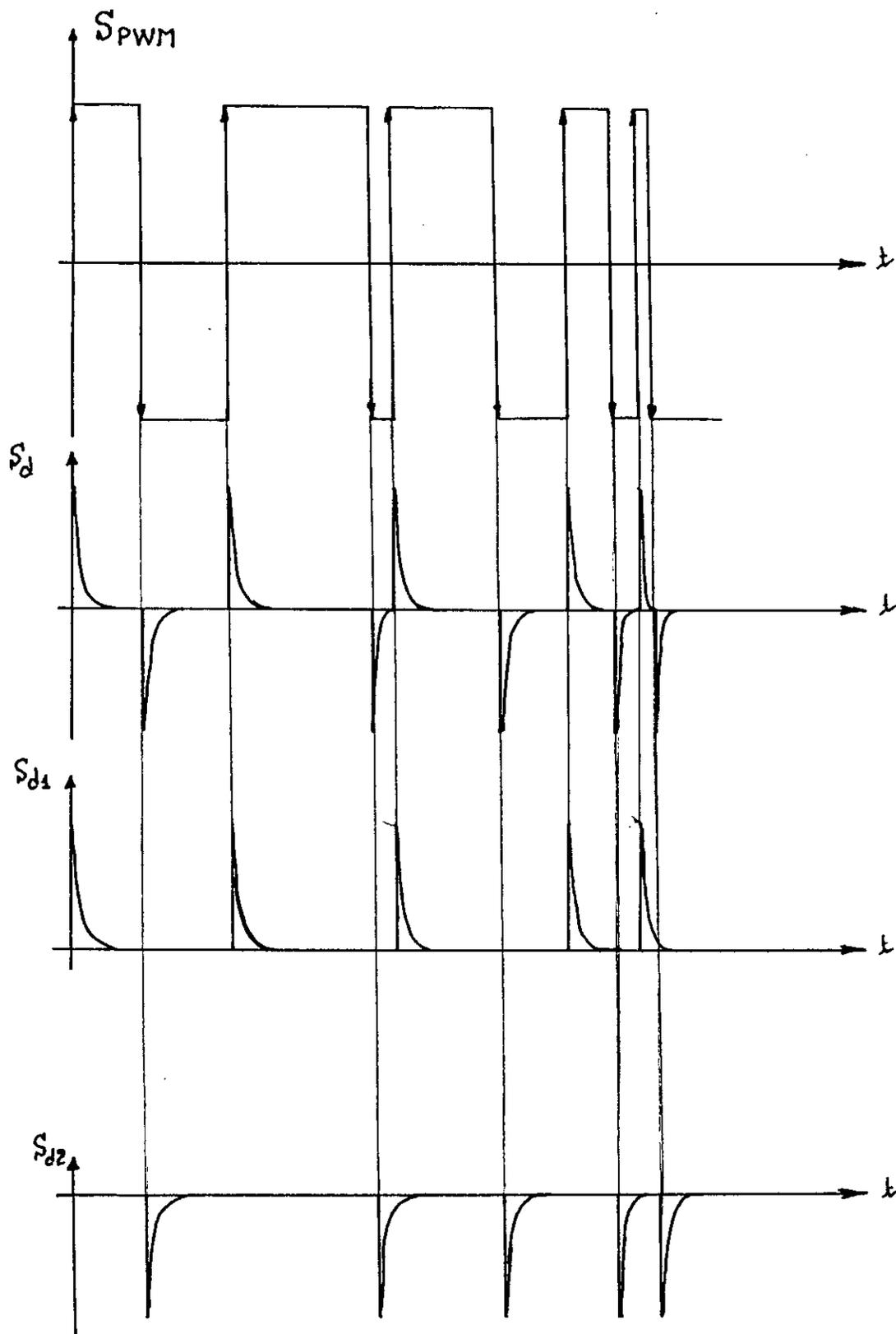


Figure 5.8 : Formes D'ondes Des Signaux De commande.

CONCLUSION

Une méthode numérique de résolution des équations non-linéaires associées à l'élimination d'harmoniques PWM en contrôle de puissance dans les deux cas de l'élimination pure et asservie a été traitée.

La difficulté majeure dans ce genre de problèmes est de trouver des conditions initiales assez proches de la solution pour assurer la convergence de l'algorithme.

Ce problème a été résolu en développant un algorithme qui permet de localiser la solution exacte et ainsi initialiser le processus itératif. Les résultats trouvés sont à un très haut degré de précision, où nous l'avons poussée jusqu'au maximum de capacité du micro-ordinateur utilisé.

Enfin, le problème du contrôle de tension dans les onduleurs PWM est résolu en éliminant constamment un nombre déterminé d'harmoniques. Si la valeur du fondamental n'influe guère il est préférable d'utiliser la PSHE, dans le cas contraire la ASHE est nécessaire. Par contre si le problème tend à optimiser certains critères, le contrôle optimale s'impose.

Ce travail peut être poursuivi par ^{une} implémentation sur microprocesseur.

ANNEXES

annexe A

QUELQUES ALGORITHMES IMPLEMENTES EN ON-LINE :

1 Algorithme de ENJETI [1] :

L'algorithme est basé sur la génération de deux groupes de droites avec des pentes positives et négatives en approximent le modèle de la solution exacte. L'algorithme est valide pour n'importe quel degré de liberté N (N impair et $N > 3$). Il est à noter que cet algorithme prend en considération la deuxième solution.

La pente positive m_p correspond aux angles ($\alpha_1, \alpha_2, \alpha_4, \dots$ etc)

La pente négative m_n correspond aux angles ($\alpha_3, \alpha_5, \alpha_7, \dots$ etc)

Ces pentes sont données par :

$$\begin{aligned} m_p &= 5.0391 e^{-0.07125N} \\ m_n &= -6.4384 e^{-0.05672N} \end{aligned} \quad (A.1)$$

Les angles approximés sont donnés par :

$$\begin{aligned} \alpha_k &= m \cdot M + C_k \\ \text{où } C_1 &= 0 \quad C_2 = 120 / (N+1) \\ C_k &= C_{k+1} = \frac{(k+1)60}{(N+1)} \\ C_{N-2} &= \frac{(N-1)60}{N+1} \quad 3 \leq k < N-2, k \text{ impair} \\ \text{et } C_{N-1} &= 60 \quad C_N = \frac{(N+3)60}{N+1} \end{aligned} \quad (A.2)$$

$$\text{ainsi } \alpha_k = m_p \cdot M + C_k \quad k=1, 2, 4 \text{ etc.} \quad (A.3)$$

$$\alpha_k = m_n \cdot M + C_k \quad k=3, 5, 7 \text{ etc.} \quad (A.4)$$

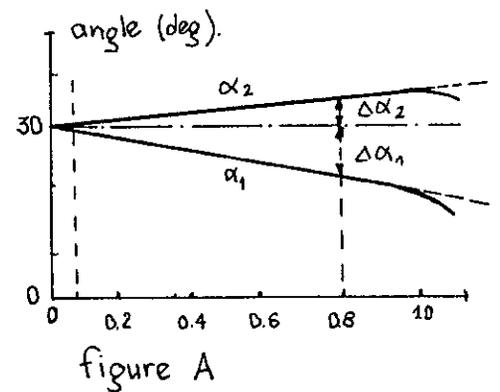
L'algorithme a les avantages suivants:

- L'algorithme est très utilisé pour générer les conditions initiales.
- L'algorithme est valide pour des ordres très élevés ($N > 101$).
- Il réduit considérablement le temps de calcul.

2 Algorithme de TAUFIQ [7] :

Comme pour l'algorithme précédent on prend avantage que les courbes donnant les angles en fonction du fondamental sont parallèles, ainsi le gradient de ces trajectoires

diminu quand le fondamental augmente. cette propriété est illustrée par la figure(A) où Δ_k dans la région linéaire est donnée par :



$$\Delta_k = C_1 \sin \left[\frac{\theta_1 (k-0.5)}{N} + \theta_2 \right] \quad , \quad \text{pour } k \text{ impair} \quad (A.5)$$

$$\Delta_k = C_2 \sin \left[\frac{k}{(N-1)} (\theta_3 - \phi_N) \right] \quad , \quad \text{pour } k \text{ pair} \quad (A.6)$$

avec

$$C_1 = 0.4025; \quad C_2 = 0.381; \quad \theta_1 = 59.184^\circ; \quad \theta_2 = 60.408^\circ; \quad \theta_3 = 58.558^\circ \quad (A.7)$$

$$\phi_N = \frac{1.135 \cdot 11}{N-2}$$

Les angles sont données par :

$$\alpha_k = \frac{60^\circ (k+1)}{N+1} - \left[\frac{2 \cdot 60}{N+1} \frac{\Delta_k \cdot M}{0.8} \right] ; \quad \text{pour } k \text{ impair} \quad (A.8)$$

$$\alpha_k = \frac{60^\circ \cdot k}{N+1} + \left[\frac{2 \cdot 60^\circ}{N+1} \frac{\Delta_k \cdot M}{0.8} \right] = ; \quad \text{pour } k \text{ pair.} \quad (A.9)$$

Quant la valeur du fondamental atteint la partie non-linéaire on introduit un facteur de correction sachant que l'erreur entre la valeur exacte et l'approchée varie non-linéairement. Ce facteur est donnée par :

$$\Delta D_k = \frac{(M-0.8)^2}{0.09} * \frac{13}{N} \sin\left[\frac{180^\circ k}{N+5}\right] ; \quad \text{pour } k \text{ impair} \quad (A.10)$$

$$\Delta D_k = \frac{(M-0.8)^2}{0.09} * \frac{14}{N} \sin\left[\frac{(k-1.5)180^\circ}{N}\right] ; \quad \text{pour } k \text{ pair} \quad (A.11)$$

Ce facteur est soustrait de l'angle approché pour des valeurs du fondamental supérieur à 0.8.

annexe B

```
PROGRAM RESOLUTION_SYSTEME_NON_LINEAIRE ;
  {CE PROGRAMME RESOUD UN SYSTEME D'EQUATIONS NON LINEAIRES
  AVEC LA METHODE DE NEWTON_RAPHSON ; LE SYSTEME LINEARISE
  RESULTANT
  EST RESOLU PAR GAUSS AVEC RECHERCHE DU PIVOT MAXIMALE ;
  LES CONDITIONS INITIALES SONT GENEREES SUIVANT PLUSIEURS ALGO-
  RITHMES .
  FAIT PAR DEROUAZI_ABDENOUR & KADRI_MOHAMED 5eme ANNEE
  ELECTRONIQUE : ENP .
  SUJET PROPOSE ET DIRIGES PAR : M) LARBES & AIT-CHEIKH.
  LE P.F.E EST INTITULE :
  ' ETUDE D'UNE TECHNIQUE NUMERIQUE POUR GENERATION '
  ' D'UN SIGNAL MLI ' }
```

```
uses crt , graph ;
label debut , fin ;
const mmax=50 ;
type
tab1=array[1..mmax ,1..mmax] of extended;
tab2=array[1..mmax] of extended;
tab3=array[1..mmax] of integer;
var
choix , asserv , cd ,repeter : char ;
nit,i,j,k,l,m,kmax,nitmax,ordrefin:integer ;
a : tab1 ;
n:tab3;
b,x,dx, hnouveau , hancien, deltax , c : tab2 ;
s ,y,err,precision , t , fond: extended;
testg , testh , testa : boolean ;
r,g1,g,lambda : extended ;
```

```
{-----}
{----- elle reste en attente jusqu'a la tape du caractere de -----}
{----- de commande exacte -----}
{-----}
```

```
PROCEDURE ATTENTE (encore :char ;enc1 , enc2 ,enc3: char);
BEGIN
repeat encore := upcase(readkey) until encore in [enc1,enc2,enc3] ;
END ;
```

```
{-----}
{----- resoud un syst d'eqt lineaire par la methode de gauss -----}
{-----}
```

```
PROCEDURE GAUSS( var ordre : integer ; var membre1 : tab1 ;
membre2 : tab2 ) ;
```

```
PROCEDURE ARRANGER ;
BEGIN(de la procedure arranger )
l:=k ;
repeat
if abs(membre1[i,k]) > abs(membre1[l,k]) then l:=i ;
i:=i+1
until i=(ordre+1) ;
i:=i-1;
if l<>k then
```

```

begin
  j:=k ;
  repeat
    s:=membre1[k,j] ;
    membre1[k,j]:=membre1[l,j] ;
    membre1[l,j]:=s;
    j:=j+1
  until j=ordre+1 ;
  j:=j-1 ;
  s:=membre2[k] ;
  membre2[k]:=membre2[l] ;
  membre2[l]:=s ;
end ;
i:=k+1 ;
END;    { de la procedure arranger }
BEGIN{ de la procedure gauss }
testg := true ;
k:=1 ;
repeat
i:=k+1 ;
arranger ;

  repeat
y:=membre1[i,k]/membre1[k,k] ;
membre1[i,k]:=0 ;
j:=k+1 ;
  repeat
membre1[i,j]:=membre1[i,j] -y*membre1[k,j] ;
j:=j+1
  until j=ordre+1 ;
j:=j-1 ;
membre2[i]:=membre2[i]-y*membre2[k] ;
i:=i+1 ;
until i=ordre+1 ;

  i:=i-1 ;
  k:=k+1 ;
until k=ordre ;
k:=k-1 ;
dx[ordre] := membre2[ordre]/membre1[ordre,ordre] ;
i:=ordre-1 ;
repeat
j:=i+1 ;
s:=0 ;
  repeat
s:=s+ membre1[i,j]*dx[j];
j:=j+1
  until j=ordre+1 ;
j:=j-1 ;
dx[i]:= (membre2[i]-s)/membre1[i,i] ;
i:=i-1 ;
until i=0 ;
i:=i+i ;
END ;

```

```

{-----}
{----- elle genere les ordres des harmoniques a eliminer : -----}
{-----impaire et non multiple de trois -----}
{-----}

```

```

PROCEDURE ORDHARM( ordre : integer);
BEGIN
  if asserv in ['A','P'] then case asserv of
    'P': n[1]:=5;
    'A': n[1]:=1
    end;{writeln('n[1]=' ,n[1]);}

  i:=2 ;
  while i<= ordre do
    begin
      n[i]:=n[i-1]+2;
      if n[i] mod 3 = 0 then n[i]:=n[i]+2;{writeln('n[' ,i,']=',n[i]);}
      i:=i+1
    end
  END;

```

```

{-----}
{----- elle test si la precision sur la nullete -----}
{----- des harmonique est atteinte -----}
{-----}

```

```

PROCEDURE CRITERE_H(var ordre : integer ; var eps : extended) ;
label 4 ;
BEGIN
  testh := true ;
  for j:=1 to ordre do
    begin
      err := abs(hnouveau[j] - hancien[j]) ;
      if err > eps then
        begin
          testh := false ; goto 4
        end
    end ;
  4:
  END ;

```

```

PROCEDURE NORME(ordre : integer);
BEGIN
  g:=0;
  for i:=2 to ordre do
    g:=g+sqr(hnouveau[i])
  END;

```

```

{-----}
{----- elle teste si la precision sur les angles est atteinte -----}
{-----}

```

```

PROCEDURE TESTANGLE(var ordre : integer ) ;
label 3 ;
BEGIN
  testa := true ;
  for j:=1 to (ordre-1) do
    begin
      if x[j] > x[j+1] then
        begin
          testa := false ; goto 3
        end
    end
  end

```

```

end ;
3:
END ;
PROCEDURE INITIALISATION( var ordre: integer ) ;
var v,u :integer ;
BEGIN
for v := 1 to ordre do
hancien[v] :=0
END ;

{-----}
{----- calcul l'amplitude des harmoniques -----}
{-----}

PROCEDURE HARMONIQUE( var ordre1:integer;ordre2 : integer ) ;
var arg:extended;
BEGIN
{ if asserv = 'A' then n[1] := 1 ;}
j:=0 ;
repeat
j:=j+1 ;
hnouveau[j]:=0 ;
for i:=1 to ordre1 do
begin
arg:=n[j]*x[ordre1+1-i];
arg:=frac(arg/(2*pi))*2*pi;
hnouveau[j]:=-2*cos(arg) - hnouveau[j] ;
end;
hnouveau[j]:=hnouveau[j]+1
until j=ordre2 ;
{ if asserv='A' then hnouveau[1]:= hnouveau[1] - fond}
END ;

{-----}
{----- calcul les elements de la jacobienne -----}
{-----}

PROCEDURE JACOBIEENNE( var ordre : integer);
var arg:extended;
BEGIN
for i:= 1 to ordre do
for j:= 1 to ordre do
begin
arg:=n[i]*x[j];
arg:=frac(arg/(2*pi))*2*pi;
a[i,j] := 2*n[i]*sin(arg);
if (j mod 2) =0 then a[i,j] := -a[i,j]
end
END ;

{-----}
{----- affiche les angles trouves est les harmonique crrespondent -----}
{-----}

PROCEDURE AFFICHAGE( var ordre : integer ; var xa: tab2 ) ;
var ch : char ;
BEGIN
{clrscr ;}
{gotoxy(20,2)} ; writeln(' LES ANGLES SONT :') ; writeln ;
for j:= 1 to ordre do

```

```

writeln('alpha['j,']=',(xa[j])*180/pi:4:3,' degree');
writeln (' si vous voulez les valeurs des harmoniques correspondants' );
writeln('pour continuer tapez < RETURN >  sinon < ESCAPE > ');
repeat cd := upcase(readkey) until cd in [#27,#13];
if cd=#13
then begin
write('jusqua quel ordre voulez vous afficher ?  ordre =');read(ordrefin);
ordharm(ordrefin);
harmonique(ordre,ordrefin) ;
writeln ;
  if asserv='A' then
    begin
      j:=1 ;
      writeln('h['n[1,']=',(hnouveau[1])*4/pi*n[1])
    end
  else j:=0 ;
repeat
j:=j+1 ;
writeln('h['n[j,']=',hnouveau[j]*4/(pi*n[j]))
until j=ordrefin ;
  end
END ;
PROCEDURE NOVELX(ordre: integer; lbd:real);
BEGIN
  for i:=1 to ordre do
    x[i]:=x[i]+lbd*dx[i]
END;

{-----}
{--- elle resoud le syst d'eqt non lin par la methode de newton_raphson---}
{-----}

PROCEDURE AUTONEWRAPH( var ordre_syst : integer ; var epsilon :exte-
nded;
                      var xn :tab2 );

BEGIN
  initialisation (ordre_syst );
  harmonique( ordre_syst,ordre_syst) ;
  if asserv='A' then hnouveau[1]:=hnouveau[1]-fond ;
  critere_h ( ordre_syst,epsilon); {testh := true ;}
  for j:= 1 to ordre_syst do hancien[j] := hnouveau[j] ;
  jacobienne( ordre_syst) ;
  for j:=1 to ordre_syst do hnouveau[j]:=-hnouveau[j] ;
  gauss (ordre_syst,a,hnouveau); {testg := true ;}
  for j:=1 to ordre_syst do hnouveau[j]:=-hnouveau[j] ;
  testa:=true;
  novelx(ordre_syst,lambda);
{
  if testg = true then
    if testa = true then
      if testh = true then
        begin
writeln('on est arrive a la solution demandee avec la precision demand^Be ')
;
writeln('apres ', nit,' operations et lambda =',lambda:6:5) ;
          if nit = nitmax then
            writeln('=nitmax ')
          end
        else

```

```
writeln(' la precision n',chr(39),'est pas atteinte essayer d',chr(39),'autres cd
initiales ')
    else
writeln(' les angles ne sont pas ordonnees ; essayer d',chr(39),'autres cd
initiales ')
    else
    begin
writeln('la resolution par gauss est impossible essayer d',chr(39),'autres cd
');
writeln ('initiales ')
    end;}
END ;
```

```
{-----}
{----- elle genere les cd initiales pour l'elimination pure -----}
{----- suivant l'algorithme qu'on a fait par des approximations -----}
{-----}
```

```
PROCEDURE CDPURE( var ordre : integer ) ;
var   co1,co2,co3,co4,co5,co6,co7,co8,
      co9,co10,co11,co12,co13,co14,co15,co16 : real ;
var arg , cof : real ;
BEGIN
  if ordre mod 2 = 0 then
    begin
      co1:=30.4955 ; co2:=1.10299 ; co3:=0.7907 ;
      co4:=0.0395882 ; co5:=13.5027 ;
      co6:=1.35367 ; co7:=0.602196 ; co8:=0.09845
    end
  else
    begin
      co1:=18.0241 ; co2:=1.23721 ; co4:=0.0811231 ;
      co3:=0.644476 ; co5:=28.533 ;
      co6:=1.13131 ; co7:=0.773831 ; co8:=0.0468122 ;
    end;
    i:=1 ;
    repeat
      cof := co1*exp(co2*ln(i)) ;
      arg := co4*ln(i)+co3 ;
      x[i]:=cof*exp(-arg*ln(ordre));
      x[i]:=x[i]*pi/180 ;
      if (ordre mod 2 =0) or (i<ordre) then
        begin
          cof:=co5*exp(co6*ln(i+1)) ;
          arg:=co8*ln(i+1)+co7 ;
          x[i+1]:=cof*exp(-arg*ln(ordre));
          x[i+1]:=x[i+1]*pi/180
        end ;
      i:=i+2
    until i> ordre
END ;
```

```
{-----}
{----- elle genere automatiquement les cd initiales -----}
```

```

{----- pour l'asservissement du fondamental -----}
{----- selon [1] (Enjeti) pour des ordres impairs -----}
{-----}

```

```

PROCEDURE CDENJETI(var ordre :integer ; var fondam:extended) ;
var mp,mn : extended ;
BEGIN
if ordre>=5 then
  begin
  if fondam=0 then fondam:=1E-10 ;
  mp:=5.0391*fondam*exp(-0.07125*ordre) ;
  mn:=-6.4384*fondam*exp(-0.05672*ordre) ;
  c[1]:=0 ; c[2]:=120/(1+ordre) ;
  k:=3 ;
  while k<(ordre-3) do
    begin
    c[k]:=(1+k)*60/(1+ordre) ;
    c[k+1]:=c[k] ;
    k:=k+2
    end ;
    c[ordre-2]:=60*(ordre-1)/(1+ordre) ;
    c[ordre-1]:=60 ;
    c[ordre]:=(3+ordre)*60/(1+ordre) ;
    x[1]:=mp+c[1] ;
    for k:=2 to ordre do
      begin
      if k mod 2=0 then x[k]:=mp+c[k]
      else x[k]:=c[k]+mn
      end ;
    for i:=1 to ordre do x[i]:=x[i]*pi/180
    end
  else
  begin
  writeln('donnez un ordre du systeme superieur a 5 ')
  end
  END ;

```

```

{-----}
{----- elle genere les cd initiales pour l'asservissement -----}
{----- du fondamental pour des ordres pairs -----}
{----- suivant un algorithme qu'on a fait -----}
{-----}

```

```

PROCEDURE CDSRVPAIRP1P2( var ordre : integer) ;
var p1 , p2 , delta :real ;
BEGIN
p1:=36.8235*exp(-0.94196*ln(ordre)) ;
p2:=-69.8246*exp(-1.30811*ln(ordre)) ;
delta:=(2*pi)/(3*ordre) ;
x[1]:=p1*fond ;
x[2]:=p2*fond+120/ordre ;
x[1]:=x[1]*pi/180 ;
x[2]:=x[2]*pi/180 ;
for i:=1 to (ordre-2) do
  x[i+2]:=x[i]+delta
END ;

```

```

{-----}
{----- elle genere les cd initiales pour l'asservissement -----}
{----- du fondamental pour des ordres impairs -----}
{----- suivant un algorithme qu'on a fait -----}
{-----}

```

```

PROCEDURE CDSRVIMPP1P2( var ordre : integer) ;
var p1 , p2 , delta :real ;
BEGIN
p1:=-44.9129*exp(-0.959911*ln(ordre)) ;
p2:= 65.003*exp(-1.73097*ln(ordre)) ;
delta:=(2*pi)/(3*(1+ordre)) ;
x[1]:=p1*abs(fond)+120/(1+ordre) ;
x[2]:=p2*abs(fond)+120/(1+ordre);
x[1]:=x[1]*pi/180 ;
x[2]:=x[2]*pi/180 ;
i:=3;
while i<> ordre do
  begin
    x[i]:=x[i-2]+delta; x[i+1]:=x[i-1]+delta; i:=i+2
  end;
x[ordre]:=x[ordre-2]+delta
END ;

```

```

{-----}
{----- genere les cd initiales selon [2] (Taufiq)-----}
{-----}

```

```

PROCEDURE CDTAUFIQ( var ordre :integer);
const c1=0.4025 ;th1=59.184 ;th2=60.408 ;c2=0.381 ;th3=58.558 ;
BEGIN
i:=1 ;
repeat
x[i]:=th1*(i-1)/ordre+th2 ;
x[i]:=x[i]*pi/180 ;
x[i]:=c1*sin(x[i]) ;
x[i]:=(60*(i+1))/(ordre+1)-(120*x[i]*fond)/(0.8*(ordre+1)) ;
x[i]:=x[i]*pi/180 ;
if i < ordre then
begin
x[i+1]:=1.135*11/(ordre-2) ;
x[i+1]:=(i*(th3-x[i+1]))/(ordre-1) ;
x[i+1]:=x[i+1]*pi/180 ;
x[i+1]:=c2*sin(x[i+1]) ;
x[i+1]:=(60*i)/(ordre+1)+(120*x[i+1]*fond)/(0.8*(ordre+1)) ;
x[i+1]:=x[i+1]*pi/180
end ;
i:=i+2 ;
until i>(ordre+1) ;
END;

```

```

{-----}
{----- elle trace le spectre des harmoniques -----}
{-----}

```

```

PROCEDURE SPECTRE ( ordspec:integer);
const inx:array [1..21] of string=('5','7','11','13','17','19','23','25','29',

```

```

'31','35','37','41','43','47','49','53','55','59','61','65');
iny:array [1..13] of string=('0.0-', '0.1-', '0.2-', '0.3-', '0.4-', '0.5-',
                           '0.6-', '0.7-', '0.8-', '0.9-', '1.0-', '1.1-', '1.2-');

var
xx,ii,separx:integer;
yy,arg:real;
quiter:char;
  grDriver : Integer;
  grMode   : Integer;
  ErrCode  : Integer;
BEGIN
  grDriver := Detect;
  InitGraph(grDriver,grMode,'c:\tp6\bgi');
  ErrCode := GraphResult;
  if errcode=grok then
  rectangle(65,50,615,350); {50 300}
  separx:=round(550/(ordspec+1));
  xx:=0;
  for ii:=1 to 5 do
    begin
    line(65,100+xx,615,100+xx);
    xx:=xx+50
    end;
  xx:=65;
  for ii:=1 to ordspec do
    begin
    line(xx,350,xx,355);
    xx:=xx+separx;
    end;
  outtextxy(60,360,'0');
  outtextxy(65+separx-3,360,'f');
  xx:=65+2*separx;
  if (ordspec <= 22) then
  begin
  for ii:=1 to ordspec do
  begin
  if length(inx[ii])=1 then outtextxy(xx-3,360,inx[ii])
    else outtextxy(xx-6,360,inx[ii]);
  xx:=xx+separx
  end
  end;
  settextstyle(0,0,1);
  xx:=347;
  for ii:=1 to 13 do
  begin
  outtextxy(35,xx,iny[ii]);
  xx:=xx-25;
  end;
  settextstyle(1,0,2);
  outtextxy(200,400,'spectre des harmoniques');
  settextstyle(0,0,1);
  outtextxy(65,370,'f:fondamentale');
  outtextxy(200,390,'ordre des harmoniques');
  outtextxy(30,30,'les harmopniques multiples de trois sont eliminees auto-
matiquement');
  settextstyle(0,1,1);
  outtextxy(15,200,'amplitude');
  {if asserv='A' then begin

```

```

        outtextxy(10,90,'amplitude en % par rapport');
        outtextxy(25,180,' au fondamental')
    end
else begin
    outtextxy(10,90,'amplitude en % par rapport');
    outtextxy(22,80,' au 1) harmonique non elimine')
end;}

setcolor(red);
xx:=65+separx; ii:=2 ;
if asserv='A' then
line(xx,350,xx,350-abs(round(abs(hnouveau[1]*4/pi)*(250))) )
    else
    begin
        yy:=0 ;
        for ii:=1 to m do
            begin
                arg:=x[m+1-ii];
                arg:=frac(arg/(2*pi))*2*pi;
                yy:=-2*cos(arg) - yy
            end;
            yy:=(yy+1);
            ii:=1 ;
        line(xx,350,xx,350-abs(round(abs(yy*4/pi)*(250))))
    end;
    xx:=xx+separx;
    repeat
        if asserv='A' then
            yy:=(abs(hnouveau[ii]*4/(pi*n[ii])*250))
            else
            yy:=(abs(hnouveau[ii]*4/(pi*n[ii])*250));
            yy:=round(yy);
            yy:=350-abs(yy);
            line(xx,350,xx,round(yy));
            xx:=xx+separx;
            ii:=ii+1;
        until ii > (1+ordspec) ;
        settextstyle(0,0,1);
        outtextxy(30,460,'pour quitter tapez < ESCAPE >');
        repeat cd:=upcase(readkey) until cd=#27 ;
    closegraph;
END ;

```

```

{-----}
{----- elle fait rentrer les differentes donnees du pb : -----}
{----- ordre ,assrvi ou non ... etc -----}
{-----}

```

```

PROCEDURE DATANDCALCUL( var dataordre : integer);
label 15 , retour, fin ;
var lambda0 :real; arret,ii,nbfois : integer;
BEGIN
if (fond*4/pi) < 0.15 then
    begin
        lambda := 0.01; r:=5; nbfois:=1
    end
else

```

```

begin
if ( (fond*4/pi) < 0.3) then lambda:=0.1 else lambda:=0.4; r:=2; nbfois:=1
end;
clrscr ;
writeln(' vous pouvez g^Bn^Brer des');writeln(' conditions initiales
tr^Bs');
writeln(' pr^Sches de la solutions '); writeln(' exacte avec
un ');
writeln(' Algorithme Personnel');
gotoxy(9,20);writeln(' < RETURN > pour oui... sinon < ESCAPE >
');
repeat cd:=upcase(readkey) until cd in [#13,#27];
clrscr; writeln; gotoxy(1,4);
if cd=#13 then
begin
case asserv of
'A' : begin
if dataordre mod 2 =0 then
cdsrvPAIRp1p2(dataordre)
else begin
if (fond > 0) then fond:=--fond;
cdsrvimpp1p2(dataordre)
end
end ;
'P' : begin
cdpure(dataordre)
end
end ;
for i:=1 to dataordre do writeln(' x['i,']=',x[i]*180/pi:5:3);
readln ;
goto 15
end ;
writeln(' donnez dans l'ordre croissant l'ordre des harmoniques a eliminer')
;
writeln (' — EN DEGREE — ') ;
for j:= 1 to dataordre do
begin
gotoxy(3,j+7) ;
write(' alpha['j,']=') ; readln(x[j]) ;
x[j] := (x[j] * pi)/180
end ;
15:
retour : clrscr ;
gotoxy(5,7) ;
write('nbre iterations max =') ; readln(nitmax) ;
gotoxy(5,9) ;
write('precision=') ; readln(precision) ;
nit:=1; arret:=0;
repeat
{***** Reglage automatique de lambda *****}
for ii:=1 to nbfois do autonewraph(dataor-
dre,precision,x) ;
novelx(dataordre,lambda/r);
harmonique(dataordre,dataordre);
norme(dataordre);
g1:=g;
novelx(dataordre,(r-1)*lambda/r);
harmonique(dataordre,dataordre);

```

```

if g1>=g then
  begin
    novelx(dataordre,-lambda);
    lambda:=(1+(r-1)*lambda)/r
  end
else
  begin
    if lambda <1 then lambda :=lambda/r;
    end;
    if lambda >=0.8 then
      begin
        lambda:=1;
        arret:=arret+1
      end;

    writeln('lambda = ', lambda:5:3);
    if (arret >=3) or (testh=true) then goto
fin;

{*****}
  nit:=nit+1
  until nit >= nitmax;
  fin : affichage(dataordre,x)
END ;

(***** debut du programme principal *****)
BEGIN
clrscr;
debut:
write('          Tapez A P ou <ESCAP>');
gotoxy(10,2+wherey);write(' votre choix est :') ;
repeat asserv:=upcase(readkey) until asserv in ['A','P',#27] ;
  case asserv of
    'A' : begin
      clrscr ;gotoxy(1,2);
      writeln(' donnez M l',chr(39),'ordre du systeme ');
      writeln(' avec le quel vous ne pouvez');writeln(' eliminer que (M-1)
harmonique ');
      gotoxy(5,6);write(' M=') ; readln(m) ;writeln;
      write('      donnez la valeur du fondamental =') ; readln(fond);
      n[1]:=1;
      fond := fond*pi/4 ;
      ORDHARM(m);
      datandcalcul(m) ;
      writeln;
      writeln('voulez vous visualiser le spectre ');
      write('tapez < RETURN > pour oui sinon < ESCAPE >') ;
      repeat cd:=upcase(readkey) until cd in [#13,#27];
      if cd=#13 then
        begin
          writeln;
          writeln('donnez le nombre d',chr(39),'harmoniques
a visualiser');
          write(' nombre =');read(ordrefin);
          ordharm(1+ordrefin);harmonique(m,1+ordrefin);
          spectre(ordrefin)

```

```

                                end
                                end ;
'P' : begin
  clrscr ;
  writeln(' donnez l',chr(39),'ordre du systeme qui est egal au ') ;
  writeln(' nombre d',chr(39),'harmoniques a eliminer ') ;
  gotoxy(5,7);write(' M=') ; readln(m) ;
  ORDHARM(m);
  datandcalcul(m) ;writeln;
  writeln('voulez vous visualiser le spectre ');
  write('tapez < RETURN > pour oui sinon < ESCAPE >') ;
  repeat cd:=upcase(readkey) until cd in [#13,#27];
  if cd=#13 then
    begin
      writeln;
      writeln('donnez le nombre d',chr(39),'harmoniques
a visualiser');
      write(' nombre =');read(ordrefin);
      ordharm(ordrefin);harmonique(m,ordrefin);
      spectre(ordrefin)
      end
    end ;
#27 : goto fin
end ;

writeln ;
write('voulez encore repetez < RETURN > sinon < ESCAPE > ');
repeat repeter:=upcase(readkey) until repeter in [#13,#27] ;
if repeter=#13 then
  begin
    clrscr ;
    goto debut
  end
  else halt(1) ;
fin : end.

```

BIBLIOGRAPHIE

- [1] P.ENJETI , J.F.LINDSAY , "Solving no linear equation of harmonic elimination PWM in power control", Electronics Letters June 1987 vol 23 N°12 pp 656-657.
- [2] H.PATEL , R.HOFT , " Generalised of harmonic elimination and voltage control in thyristor inverters : part I - harmonic elimination", IEEE Trans, 1973, vol IA-9, pp 310-317.
- [3] A.ZUCKERBERGER , A.ALEXANDEROVITZ , "Determination of commutation sequence with a view to eliminating harmonics in microprocesseur-controlled PWM voltage inverter", IEEE Trans an Ind Elec, vol IE-33 N°3, Aug 1986.
- [4] J.VIGNE , Algorithmes numériques, analyse et mise en oeuvre. tome 2 : équations et systèmes non-lineaires; ed TECHNIP, 1980.
- [5] A.RALSTON , A first course in numerical analysis ed MC Graw-Hill, 1985.
- [6] A.GOURDIN , Méthodes numériques appliquées ed Alger : OPU, 1991.
- [7] J.I.TAUFIQ "Novel algorithm for generating near optimal PWM waveforms for AC traction drives IEE Proc. Vol 133, pt.B, N°2, march 1986, pp 85-94.
- [8] G.SEGIER , Les convertisseurs de l'électronique de puissance. Vol 4 : La conversion continu-alternatif ed - Paris : tec et doc 1984.
- [9] B.K.BOSE , Power electronics and AC drives ed Prentice-Hill, 1986.
- [10] S.C.CHAPRA , Numerical methods for engineers MC Graw-hill, 1988.
- [11] P.ENJETI , "A new PWM speed control system for high performance AC motor drives". IEE Proc, Vol 133, pt.B, N°2, march 1986.
- [12] J.I.TAUFIQ "Novel algorithm for generating near optimal PWM waveforms for AC traction drives IEE Proc. Vol 133, pt.B, N°2, march 1986, pp 71-84.
- [13] H.PATEL , R.HOFT , " Generalised of harmonic elimination and voltage control in thyristor inverters : part II -Voltage control techniques", IEEE Trans, 1974, vol IA-10, pp 666-673.
- [14] G.BUGA , "Optimal PWM for feeding AC moteurs" , IEEE Trans. ind appl, vol IA-13, pp 38-44, Jan./Feb. 1977.

- [15] S.R.BOWES , "New PWM swithing strategy for microprocessor-controlled inverter drives" , IEE Proc.B, July 1986, pp 237-254.
- [16] F.ZACH , "Efficiency optimal control for AC drives with PWM inverter", IEEE Trans, Ind Appl. Jul.\Aug. 1985, pp 987-1000.
- [17] F.ZACH , "Dynamically optimal swithing patterns for PWM inverter drives (for minimizating of the torque and speed ripptes)", IEEE Trans, Ind Appl. Jul.\Aug. 1985, pp 975-968.
- [18] F.ZACH , "Comparaison of voltage and current specra and AC motor drive dynamics using various, especially optimized forms of pulse width modulation for inverters", motor-con april 1985.
- [19] S.R.BOWES , "Microprocessor implementation of new optimal PWM swithing strategies", IEE Proc.B, vol 135, sept 1988, pp 269-280.
- [20] J.MURPHY , M.EGAN , "A comparaison of PWM strategies for inverter-feed induction motors", IEEE Trans on ind.appl, vol IA-19, N°3, May./June 1983.
- [21] Z.RAMEL , T.ZENINA , "Etude de quelques techniques MLI" seminaire 1993.
- [22] AIT GOUGAM , "Etude des strategies MLI pour onduleur de tension". Thèse de magister 1992.