

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ EN DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR :  
BENMIRA HASSAN

IMPLANTATION DE LA COMMANDE VECTORIELLE SPATIALE  
DANS UN CIRCUIT DÉDIÉ

Janvier 2003

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## ***Dédicacé***

*À mon père,*

*À ma sœur et mes frères,*

*À toute ma Famille.*

## *Remerciements*

Je voudrais ici exprimer ma gratitude à tous ceux qui ont contribué à la réalisation de ce travail.

- Mes remerciements vont à mon directeur de recherche Ahmed Cheriti qui m'aura encadré et soutenu tout au long de ce projet. Sa contribution technique a indéniablement constitué un très grand apport dans les résultats de cette recherche.
- J'adresse également mes sincères remerciements à mon codirecteur de recherche Kamal Sakkay qui nous aura apporté une très grande contribution technique.
- Je tiens à remercier également tous les professeurs de l'UQTR du département de Génie Électrique, et spécialement Messieurs Adam Skorek, Mohammed Benslim, Pierre Sicard, Mostapha Aboulhamid UQAM, Kodjo Agbossou.
- Avant de terminer, je voudrais dédier ce mémoire à l'esprit de ma mère que je lui souhaite la miséricorde divine, et une spéciale dédicace à mon père et à toute ma chère famille pour leur encouragement tout au long de mes études.
- Je suis également reconnaissant envers Belyamani Mostapha, Hmido Abderahime, Tabi El mekki, qu'ont toujours sus apporter l'aide dont j'avais besoin au où il le fallait.
- J'aimerais aussi exprimer ma sincère reconnaissance à Anour. Barraouk, Eljihad. Mohammed.
- Que tous ceux qui n'ont pas été cités ne sentent nullement oublier. ils auront tous contribué à leur manière à la réalisation de ce travail.

# *Table des matières*

## *Introduction*

## *Chapitre 1*

### **LES ONDULEURS MONOPHASÉS DE TENSION À MODULATION DE LARGEUR D'IMPULSIONS (MLI)**

1.1 Onduleur monophasé en demi- pont.....	9
1.2 Onduleur monophasé en pont.....	12
1.3 Commutation du courant dans un bras du convertisseur.....	15
1.3.1 Bras avec circuit R-C-D d'aide à la commutation.....	15
1.4 Pertes dans l'onduleur.....	31
1.4.1 Pertes de conduction dans l'onduleur par conduction.....	31
1.4.2 Pertes de commutation dans l'onduleur par commutation.....	31
1.4.2.1 Pertes dans l'onduleur avec circuit R-C-D d'aide à la commutation.....	32
1.4.2.2 Pertes dans l'onduleur avec circuit R-C d'aide à la commutation.....	33
1.5 Performance des convertisseurs.....	34
1.6 Conclusion.....	35

## *Chapitre 2*

### **COMMANDE DES ONDULEURS DE TENSION À MODULATION DE LARGEUR D'IMPULSIONS (MLI)**

2.1 Modulation MLI d'une seule impulsion.....	37
2.2 Modulation MLI de multiples impulsions.....	39
2.3 Modulation MLI sinusoïdale.....	40
2.4 Modulation MLI calculée.....	41
2.5 Motif de commande des IGBTs pour le convertisseur de ce travail.....	46
2.6 Conclusion.....	50

## *Chapitre 3*

### **COMMANDE MODULATION VECTORIELLE SPATIAL (SVM)**

3.1 Fonctionnement de la commande de modulation.....	52
--	----

3.1.1 L'onduleur vu coté machine.....	52
3.1.2 L'onduleur vu coté commande.....	53
3.2 Définition de la fonction modulation vectorielle.....	55
3.3 Analyse vectorielle d'une modulation vectorielle.....	56
3.4 Limitation du domaine des vecteurs tension réalisables.....	59
3.5 Différents modes de modulation vectorielle.....	60
3.5.1 Modulation vectorielle asynchrone.....	61
3.5.2 Modulation vectorielle synchrone.....	64
3.5.3 Relation entre ``moyenne`` et ``fondamental`` .....	65
3.5.4 Modulation vectorielle à angles calculés ( 1 angle ).....	67
3.6 Conclusion.....	68

## *Chapitre 4*

### ***SIMULATIONS ET RÉSULTATS D'IMPLANTATION EN TECHNOLOGIE***

#### ***VLSI***

4.1 Description des outils utilisés .....	71
4.1.1 Monet de Mentor Graphics .....	71
4.1.2 Description des outils de Synopsys .....	77
4-2 Simulation de la commande .....	80
4. 2. 1 La validation fonctionnelle .....	80
4. 2. 2 Système de commande SVM .....	80
4. 3 L'architecture proposée .....	88
4. 3.1 Description du processeur .....	88
4.3.1.1 Le bloc d'adressage:.....	89
4.3.1.2 Le bloc de donnée de la commande .....	90
4.3.1.3 Le bloc séquenceur programmable.....	93
4.4 Résultats de l'implantation globale.....	94

4.5 Synthèse optimisation.....	96
4.6 Conclusion.....	98
Conclusion générale.....	99
Bibliographie.....	102

## *Table des figures*

Figure 1.1 Onduleur monophasé en demi-pont.....	10
Figure 1.2 Onduleur monophasé de tension en pont.....	13
Figure 1.2 Signaux de fonctionnement.....	13
Figure 1.3 Bras de l'onduleur avec un circuit R-C-D d'aide à la commutation.....	16
Figure 1.4 Bras de l'onduleur qui montre les composants en conduction pendant la période $0 < t < t_f$ .....	17
Figure 1.5 Composant en conduction pendant la période $t_f < t < t_1$ .....	19
Figure 1.6 Composants du bras en conduction pendant la période $t_2 < t$ .....	22
Figure 1.7 Évolution des tensions $V_1$ , $v_2$ et $v_{D2}$ pendant l'ouverture d'un interrupteur avec un circuit R-C-D d'aide à la commutation.....	23
Figure 1.8 Évolution des courants $i_1$ , $i_2$ et $i_{D2}$ pendant l'ouverture d'un interrupteur avec un circuit R-C-D d'aide à la commutation.....	23
Figure 1.9 Circulation du courant dans le bras du convertisseur pendant la période $0 < t < t_f$ .....	24
Figure 1.10 Circulation du courant dans le bras du convertisseur pendant la période ( $t_r < t < t_1$ ) de l'ouverture d'un interrupteur.....	27
Figure 1.11 Circulation du courant dans le bras du convertisseur pendant la période $t_1 < t$ .....	29
Figure 1.12 Évolution des tensions $V_1$ , $V_2$ et $V_{D2}$ pendant l'ouverture d'un interrupteur avec circuit R-C d'aide à la commutation.....	30
Figure 1.13 Évolution des courants $I_1$ , $I_2$ , et $I_{D2}$ pendant l'ouverture d'un interrupteur avec circuit R-C d'aide à la commutation.....	30
Figure 2.1 Modulation MLI d'une seule impulsion.....	38
Figure 2.2 Modulation MLI sinusoïdale.....	40
Figure 2.3 Tension de sortie de l'onduleur en pont.....	42
Figure 2.4 Tension de sortie avec les harmoniques 3,5,7 éliminées.....	45
Figure 3.1 Onduleur de tension triphasée.....	53
Figure 3.2 Onduleur de tension triphasée dans l'état (000).....	54
Figure 3.3 Les 8 états de fonctionnement possibles.....	54
Figure 3.4 Vecteur tension instantanés.....	55
Figure 3.5 Domaine D des vecteurs moyens.....	56
Figure 3.6 SVM vectorielle réalisant une commande.....	57
Figure 3.7 Modulation vectorielle utilisée.....	58
Figure 3.8 Domaine du vecteur tension avec prise en compte des $T_{min}$ .....	60
Figure 3.9 Caractéristique d'un vecteur tension moyen.....	61
Figure 3.10 Caractéristique d'un vecteur tension moyen.....	66
Figure 4.1 Activités de conception matérielle/logicielle.....	70
Figure 4.2 Méthode de conception avec Monet.....	71
Figure 4.3 Synthèse avec Monet.....	72
Figure 4.4 Représentation des diagrammes de Kannt et d'états dans Monet.....	73
Figure 4.5 Étapes de conception avec les outils de Synopsys.....	75



Figure 4.6 L'interface du « Behavioral Compiler » de Synopsys.....	77
Figure 4.7 Seamless CVE pour la cosimulation matérielle/logicielle.....	78
Figure 4.8 Schéma fonctionnel.....	81
Figure 4.9 Schéma source triphasé.....	82
Figure 4.10 Transformation DQ et calcul de l'amplitude et la phase.....	83
Figure 4.11 Calcul de temps et les positions de vecteur de tension .....	84
Figure 4.12 Multiplexage des signaux de la commande .....	85
Figure 4.13 Résultats de simulation.....	86
Figure 4.14 Schéma synoptique de la commande numérique.....	87
Figure 4.15 Structure du processeur.....	88
Figure 4.16 Structure d'un compteur.....	89
Figure 4.17 Séquencement de compteur.....	90
Figure 4.18 Architecture matricielle.....	91
Figure 4.19 Bloc mémoire.....	92
Figure 4.20 Séquencement de la mémoire.....	92
Figure 4.21 la commande modulation vectorielle spatial.....	94
Figure 4.22 Les signaux de la commande modulation vectorielle spatial.....	95

## *Liste des abréviations*

<b>A</b>	Angle de déphasage, rad
<b>Ap</b>	Amplitude du signal porteur dans la modulation à largeur de impulsion
<b>ASIC</b>	(Application-Specific Integrated Circuit) Circuit Intégré spécialisé. En général, ils sont produits en petites quantités, car c'est du sur-mesures
<b>A</b>	Amplitude du signal de référence dans la modulation à largeur de impulsion
<b>a, a<sub>k</sub></b>	Angles de commutation, rad
<b>B</b>	Coefficient
<b>BJT</b>	Transistor bipolaire de jonction
<b>C</b>	Condensateur du circuit d'aide à la commutation
<b>Cc</b>	Condensateur du filtre de sortie
<b>CA</b>	Continu-alternatif
<b>cc</b>	Continu-continu
<b>CALC</b>	Circuit d'aide à la commutation des interrupteurs semi-conducteurs
<b>D</b>	Diode du circuit d'aide à la commutation
<b>Di</b>	Diode k de retour de l'onduleur
<b>L</b>	Largeur d'impulsion, rad
<b>Lm</b>	Largeur de l'impulsion m
<b>w</b>	Fréquence angulaire de la fondamentale de tension, rad/seconds
<b>wf</b>	Fréquence angulaire de résonance du filtre de sortie, rad/seconds
<b>i</b>	Constante égal à $R \cdot C$
<b>ROM</b>	Mémoire morte de lecture seulement
<b>FPGA</b>	Field programmable analogarray
<b>f</b>	Fréquence de la harmonique
<b>GTO</b>	Thyristor à gâchette contrôlé
<b>HD</b>	hardware description language
<b>Hz</b>	Hertz, unité de fréquence
<b>IGBT</b>	bipolaire à grille isolé
<b>IEEE</b>	Institute of Electrical and Electronic Engineers
<b>I<sub>ch</sub></b>	Courant à la charge, A

<b>I</b>	Courant nominal de l'onduleur, A
<b>iDk</b>	Courant instantané dans la diode de retour k
<b>i</b>	Courant instantané dans le condensateur du filtre de sortie, A
<b>i.h</b>	Courant instantané à la charge, A
<b>il</b>	Courant instantané à la charge, A
<b>io</b>	Courant à la sortie de l'onduleur, A
<b>isc</b>	Courant de court circuit, A
<b>iTk</b>	Courant instantané dans le transistor k, A
<b>k</b>	Constante j dans la solution d'un équation différentielle
<b>Lf</b>	Inductance du filtre de sortie, H
<b>M</b>	Indice de modulation
<b>MLI</b>	Modulation de largeurs d'impulsion
<b>MOSFET</b>	Transistor à effet de champ d'oxyde de métal
<b>mf</b>	Rapport de la modulation de fréquences
<b>P</b>	Puissance de l'onduleur, VA
<b>PTk</b>	Pertes de puissance dans le transistor k pendant son ouverture, W
<b>RTL</b>	Register Transfer Level
<b>SI</b>	Interrupteur k dans les circuits générateurs d'impulsions
<b>SVM</b>	Modulation vectorielle spatial
<b>SIMULINK</b>	Logiciel pour faire des simulations de systèmes en utilisant des fonctions en blocs. C'est un outil de simulation du logiciel MATLAB
<b>SDL</b>	(Specification and Description Language) voir Langage de Spécification et de Description
<b>TDH</b>	Taux de distorsion harmonique
<b>TDHT</b>	Taux de distorsion harmonique total
<b>Tk</b>	Interrupteur semi-conducteur k de l'onduleur
<b>t<sub>f</sub></b>	Temps de descente du courant dans le transistor pendant son ouverture, sec
<b>V<sub>E</sub></b>	Tension continue d'entrée à l'onduleur, volts
<b>V<sub>o</sub></b>	Tension de sortie de l'onduleur, volts
<b>V<sub>k</sub></b>	Tension dans les condensateurs k d'aide à la commutation, volts

# *Introduction*

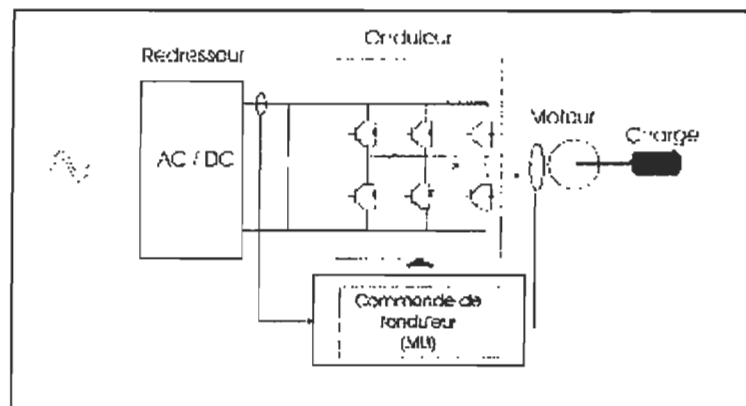
La progression impressionnante de la technologie en électronique a permis l'expansion des équipements électroniques partout, dans l'industrie, le champ militaire, le champ médical, le champ aérospatial, le divertissement, etc.. Nous vivons dans l'ère de l'électronique.

Un des domaines de l'électronique en développement est l'électronique de puissance qui traite et contrôle la circulation de l'énergie électrique ainsi que sa conversion pour fournir des tensions et des courants adéquats aux différents types de charges que l'on trouve dans différentes applications: sources d'énergie régulées, processus électrochimiques, compensateurs statiques d'énergie réactive, chauffages par induction, et la transmission d'énergie électrique en courant continu à haute puissance, etc..

Les différentes conversions sont: courant continu (c.c.) à courant alternatif (c.a.); courant alternatif (c.a.) à courant continu (c.c.); courant alternatif (c.a.) à courant alternatif (c.a.), et courant continu (c.c.) à courant continu (c.c.).

Dans ce travail nous sommes intéressés à la conversion de l'énergie électrique courant continu (c.c.) à énergie électrique courant alternatif (c.a.). Le système électronique de puissance qui fait cette conversion est dénommée l'onduleur. Les topologies les plus connues sont celles de l'onduleur à demi-pont et à pont complet. Chaque topologie possède des avantages et des désavantages.

Du point de vue du contrôle de l'onduleur, celui-ci peut être commandé en utilisant plusieurs techniques, la plus répandue est la technique de Modulation de Largeur d'impulsions (MLI), laquelle en anglais se connaît comme **Pulse Width Modulation (PWM)** [PaHo73]. MLI sert à calculer les instants de commutation des interrupteurs de l'onduleur qui alimente le moteur Figure suivant.



**Schéma Synoptique de la commande MLI avec l'onduleur qui alimente un moteur**

Deux des avantages les plus importants de cette technique sont:

- Permettre à l'onduleur statique de générer des ondes de sortie très proche de la forme idéale,

- Obtenir le contrôle linéaire de l'amplitude de la tension et le courant de sortie avec la commande des interrupteurs semi-conducteurs,

D'autre part, les désavantages suivants ont été constatés:

- L'amplitude du composant fondamental de l'onde de sortie est atténuée.
- Le stress sur les composants semi-conducteurs de commutation est plus élevé, ce qui diminue le temps d'opération - 'vie' du composant, ceci résultant de la fréquence plus élevée des commutations.

Les désavantages sont peu significatifs par rapport aux avantages, pourtant il faut en réduire l'impact, d'où l'utilisation de la technique de modulation de largeur d'impulsions.

Ainsi, plusieurs variations de cette technique ont été développées, à savoir: la technique MLI sinusoïdale originelle, la technique MLI sinusoïdale améliorée, la technique MLI avec l'injection de la troisième harmonique, etc.. Ces techniques déterminent en temps réel les angles de commutation des composants semi-conducteurs. Elles sont bien adaptées à des applications qui demandent des tensions et des courants variables, tels que les moteurs à fréquence ou tension variable.

Pour des applications qui demandent une tension et une fréquence fixes, telles que les sources d'alimentation sans interruption (en anglais, Uninterruptible Power Supplies, UPS), la technique MLI programmée ou modulation vectorielle spatiale (SVM) est très utilisée [PaHo73]. La caractéristique principale de cette technique est la détermination de façon préalable des angles de commutation des interrupteurs semi-conducteurs, en tenant compte de l'amplitude de la tension de sortie désirée et des harmoniques à éliminer. La modulation vectorielle spatiale permet de résoudre des problèmes de non-linéarité de la sortie du convertisseur ainsi que l'ondulation du couple de la machine à faible vitesse. L'approche classique pour réduire cette ondulation est d'augmenter la fréquence de commutation, or ceci augmente les pertes et pénalise donc le rendement. De plus, la MVS permet de produire la sortie du convertisseur un signal avec très peu

d'harmoniques tout en fonctionnant à faible fréquence. Par conséquent, les ondulations du couple sont réduites et le rendement n'est pas affecté [BoZi 88].

Durant les dernières années, la stratégie de commande MVS a été réalisée en utilisant des circuits analogiques ou des circuits numériques à microprocesseurs. Cependant, avec l'évolution des semi-conducteurs de puissance rapides fonctionnant à une fréquence de commutation élevée, la réalisation de la MVS n'est possible que par l'utilisation de composants rapides comme les micro-contrôleurs et les processeurs de signaux numériques. Actuellement, les circuits intégrés à applications spécifiques (ASIC) et les réseaux logiques programmables (FPGA) deviennent des éléments indispensables lors de la conception d'une commande MVS.

En effet, les ASIC et les FPGA :

- possèdent un temps d'exécution beaucoup plus rapide ;
- permettent une plus grande immunité au bruit généré par le convertisseur de puissance et une réduction du poids et de la taille du circuit de commande ;

Dans ce projet, il s'agit d'explorer une implantation sur circuit numérique spécialisé et programmable de type FPGA [SeLa89]. Le but du recours au FPGA est une réduction du poids et de la taille du circuit, une plus grande immunité au bruit, un meilleur temps de réponse et enfin de tirer profit de récents développements en micro-électronique dans le domaine de l'électronique industrielle. C'est dans cette perspective que s'inscrit ce projet de recherche.

## ***1 But du projet de recherche***

Le but de notre projet de recherche est de développer et de valider une méthodologie d'implantation de la commande de modulation vectorielle spatial. En fait, nous allons nous concentrer davantage sur le partitionnement logiciel/matériel à l'intérieur de la méthodologie. Nous devons déterminer en étudiant la commande modulation vectorielle spatial, son partitionnement optimal.

Pour y arriver, différents éléments doivent être étudiés. Tout d'abord, nous devons déterminer quelles sont les contraintes du système étudié. Dans notre cas, qui est l'implantation de la commande (SVM), les contraintes étudiées sont les **signaux de la commande**, la **validation et vérifications de la commande (simulation synthèse)**, les **temps d'exécution**, la **dissipation de puissance** et le **coût de l'implantation**. Ces contraintes doivent être évaluées à partir d'une description de haut niveau. Ces différentes métriques doivent être ensuite fusionnées en une seule valeur qui estimera la qualité de la commande. Un algorithme de partitionnement est également utilisé pour déterminer les différents blocs de la commande.

D'autres éléments sont également à considérer dans le développement de notre méthodologie. Nous voulions le plus possible utiliser des outils déjà existants, qui nous étaient en général fournis par notre département ou par d'autre professeur. Nous avons donc principalement utilisé les outils de Mentor Graphics et Xilinx pour les insérer dans notre méthodologie. Un autre critère que nous devons considérer est la possibilité d'utiliser cette méthodologie pour d'autres types d'applications. Donc, même si des métriques sont éliminées ou ajoutées, nous devons pouvoir continuer à utiliser notre méthodologie.

Finalement, l'aspect suivi pour implantation donnera au projet une allure industrielle. Le lecteur remarquera le long de ce projet que les démarches suivies ressemblent exactement à celles utilisés pour la conception des circuits numériques dans le domaine industriel. Cela veut dire qu'on trouvera toutes les démarches de modélisation ( Top-Down design) qui sont :

- 1- Définir les spécifications de l'architecture.
- 2- Répartition d'architecture.
- 3- Écriture dans un langage.
- 4- Vérification de la fonctionnalité des programmes.
- 5- Simulation, Synthèse et optimisation.
- 6- Vérification de la fonctionnalité après synthèse.



## ***2 Structure du mémoire***

Le présent document s'articule principalement autour de cinq grands chapitres.

Dans le premier, nous présentons une révision de la littérature qui traitant des onduleurs à modulation de largeur d'impulsions MLI, les topologies de l'onduleur en demi- pont et en pont complet, l'étude des évolutions de tension et du courant pendant l'ouverture des interrupteurs, les pertes dans l'onduleur et finalement nous précisons la nature de chaque élément, ses caractéristiques physiques et les paramètres de performance des onduleurs.

Dans le deuxième chapitre, nous exposons une brève analyse pour trois techniques de modulation : modulation MLI d'une seule impulsion, modulation ( MLI ) de multiples impulsions, modulation ( MLI ) sinusoïdale, et en détail la technique de modulation ( MLI ) précalculés. À qui nous permet de donner à la fin les avantages et les désavantages par rapport à la commande de modulation vectorielle spatial qu'on va présenter au troisième chapitre [BoZi 88].

Le troisième chapitre développe et traite des différents modes de commande de modulation vectorielle spatial utilisée dans le domaine de puissance. Nous présenterons tout d'abord la fonction modulation vectorielle puis nous détaillerons le mode de modulation vectorielle spatial avec leur limite de fonctionnement, nombre d'avantages et la qualité de la modulation vectorielle spatial.

Le quatrième chapitre sera consacré pour la présentation de la méthodologie d'implantation dans son ensemble. Puis nous irons plus en détails dans les parties qui nous intéressent. En particulier dans le développement de notre méthodologie pour le partitionnement matériel/logiciel. À partir de cette interprétation et des règles que nous exposons, le modèle est analysé systématiquement pour en déduire les implications sur l'intégration (analyse architecturale pour l'intégration matériel et estimation de paramètres

physiques). Ces analyses permettent ensuite au concepteur de baser ses décisions sur des données tangibles.

Les modèles et méthodes ayant été analysés, nous abordons finalement dans un cinquième chapitre avec un certain nombre de simulations. Celles-ci portent sur les blocs essentiels de la commande et de l'architecture proposée pour l'implantation. Les simulations ont été choisies à cause de leur implication et importance sur les performances de vitesse et la fonctionnalité des signaux de la commande.

# ***Chapitre 1***

## ***LES ONDULEURS MONOPHASÉS DE TENSION À MODULATION DE LARGEUR D'IMPULSIONS (MLI)***

L'onduleur est un des convertisseurs d'énergie électrique qui a comme fonction la conversion de l'énergie à courant continu (c.c.) en énergie à courant alternatif (c.a.). Il convertit une tension continue d'entrée en une tension alternative de sortie avec une amplitude et une fréquence convenable. Ces deux paramètres peuvent être fixes ou variables; si la tension d'entrée est variable et le gain de l'onduleur est constant, on obtient une tension variable à la sortie; cependant, si la tension d'entrée est constante et non commandée, la tension de sortie peut varier en changeant le gain de l'onduleur. Il y a plusieurs techniques pour obtenir cette variation, la technique de modulation de largeur d'impulsions (en anglais "pulse-width modulation", **PWM**)

est très répandue. Elle consiste à changer la largeur des impulsions de la tension de sortie avec des commandes appropriées d'ouverture et de fermeture des interrupteurs semi-conducteurs de l'onduleur [SeLa89].

L'idéal serait d'obtenir une onde sinusoïdale de la tension de sortie, ce qui n'est pas le cas dans la pratique. Ceci prouve la présence des harmoniques dans la tension. Le but des concepteurs des onduleurs est d'obtenir à la sortie un signal à taux de distortion harmonique faible.

La dénomination des convertisseurs diffère par rapport à leur source d'énergie, leur topologie, leur type de commande, etc.. Ainsi, par rapport à leur source nous avons:

- Le commutateur de courant, qui a une source de courant continu
- L'onduleur de tension, dont la source est une source de tension continue

Il y a plusieurs topologies des onduleurs. Dans ce rapport, nous allons faire une brève révision de deux des ces topologies, à savoir :

- Onduleur de tension monophasé en demi- pont
- Onduleur de tension monophasé en pont

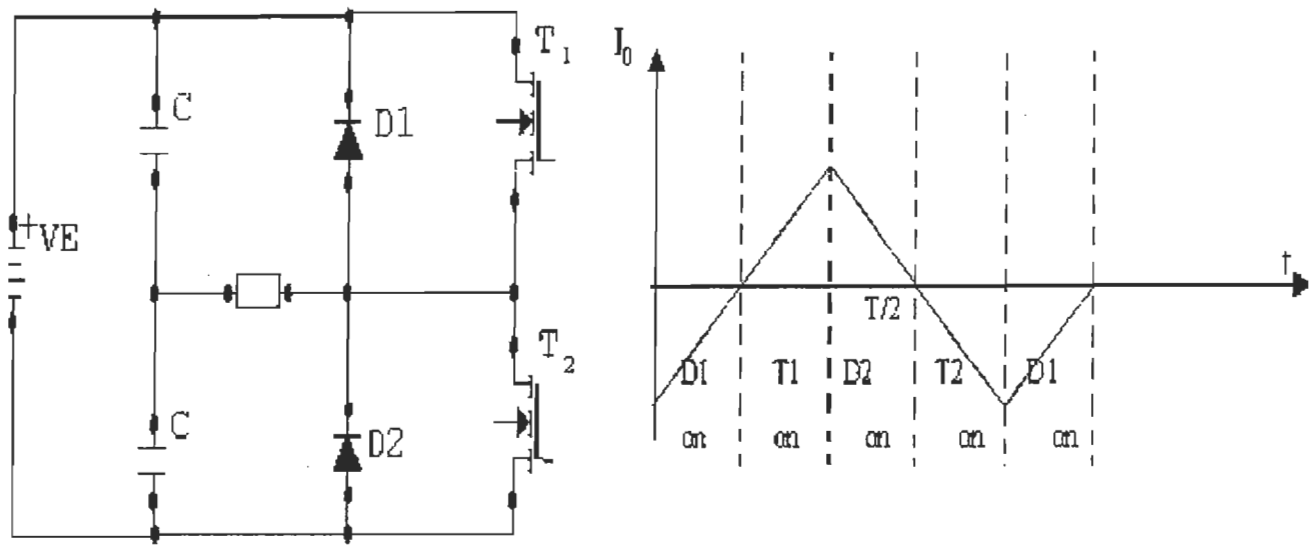
Par rapport à leur commande, nous pouvons mentionner les suivants:

- Onduleur de tension à modulation de largeur d'impulsions (MLI)
- Onduleur de tension à résonance

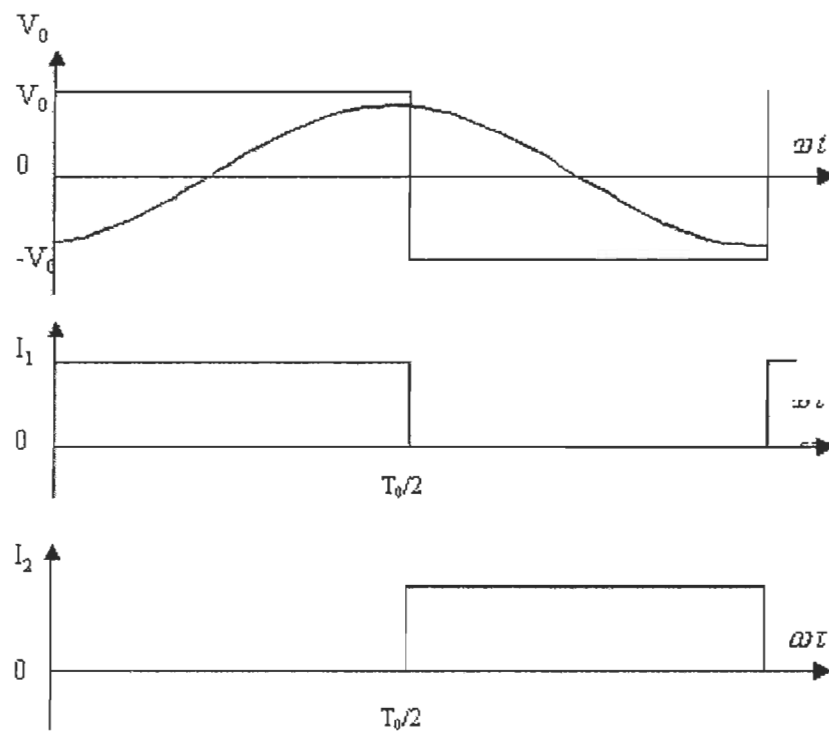
Notre intérêt portant sur les onduleurs à MLI, nous présenterons une révision de ces techniques dans ce travail, principalement sur la technique de modulation de largeur d'impulsions MLI programmée ou précalculée.

### **1-1 Onduleur monophasé en demi- pont**

La topologie de l'onduleur monophasé de tension en demi- pont est présentée à la figure[1.1].



(a)



(b)

**Figure 1.1 Onduleur monophasé en demi-pont**

Il utilise deux "interrupteurs" bidirectionnels en courant unidirectionnel en tension et une source de tension à point milieu. Cette tension s'obtient avec deux condensateurs  $C$ , de capacitance égale, branchés selon la figure [1.1], la tension aux bornes de chaque condensateur

est constante et égale à  $V_E/2$ . Les interrupteurs sont des composants électroniques de puissance commandables tels que le transistor bipolaire, le GTO, IGBT, etc.. Le principe d'opération de cet onduleur peut s'expliquer de la façon suivante [1,2]: l'interrupteur  $T_1$  se met en conduction pendant une demi- période  $T_0/2$ , tandis que  $T_2$  est à l'état bloqué; alors la tension instantanée aux bornes de la charge est égale à  $V_E/2$ . Mais, si l'on a le contraire,  $T_2$  en conduction et  $T_1$  bloqué aussi pendant une demi- période  $T_0/2$ , la tension instantanée à la charge sera maintenant égale à  $-V_E/2$ . La loi de la commande des interrupteurs doit être conçue de façon à ne pas avoir les deux interrupteurs en conduction simultanément. La figure [1.1(a)] montre la tension et le courant du composant fondamental pour une charge résistive-inductive, tandis que la figure [1.1(b)] montre l'évolution du courant de sortie et les temps de conduction des composants pour une charge purement inductive. Le courant de ce type de charge, ne peut pas changer ou s'annuler immédiatement avec un changement de la tension de sortie; si au temps  $T_0/2$  l'interrupteur  $T_1$  se bloque, le courant dans la charge ne s'annule pas immédiatement, il va continuer à circuler dans la diode  $D_2$  et la charge jusqu'à ce que le courant atteigne zéro.

L'effet sera similaire lorsque l'interrupteur  $T_2$  se bloque au temps  $T_0$ , pendant ce temps-là, le courant circule dans la partie supérieure de l'onduleur, c'est-à-dire, dans la diode  $D_1$ , la charge et le condensateur. Les diodes  $D_1$  et  $D_2$  s'appellent diodes de réaction ou de retour parce qu'elles conduisent l'énergie vers la source. Chaque interrupteur conduit pendant  $T_0/4$ ; la tension inverse à ses bornes est égale à  $V_E$ . Si l'on considère que la tension de sortie est égale à  $V_E/2$  pendant une demi-période, nous trouvons la valeur efficace de cette tension égale à:

$$V_0 = \sqrt{\frac{2}{T_0} \int_0^{T_0/2} \frac{V_E^2}{4} dt} \Rightarrow V_0 = \frac{V_E}{2} \quad (1.1)$$

Le développement en série de Fourier de la tension de sortie  $V_0$  est donné par (1.1) :

$$V_0 = \sum_{n=1,3,5,\dots}^{\infty} \frac{2V_E}{n\pi} \sin(n\omega t) \quad (1.2)$$

$$V_0 = 0 \quad \text{pour } n = 2,4,6,\dots$$

où  $\omega = 2\pi f_0$  rad/s est la fréquence de pulsation de la tension. Donc, l'expression de la composante fondamentale instantanée de cette tension serait:

$$V_{01} = \frac{2V_E}{\pi} \sin(\omega t)$$

d'où nous trouvons la valeur efficace égale à :

$$V_{01} = \left( \frac{1}{\pi} \int_0^\pi \frac{4V_E^2}{\pi^2} \sin^2(\omega t) d(\omega t) \right)^{1/2} = \frac{\sqrt{2}V_E}{\pi} \quad (1.3)$$

et le courant de sortie  $I_0$  pour une charge RL en séries de Fourier est donné par [1]:

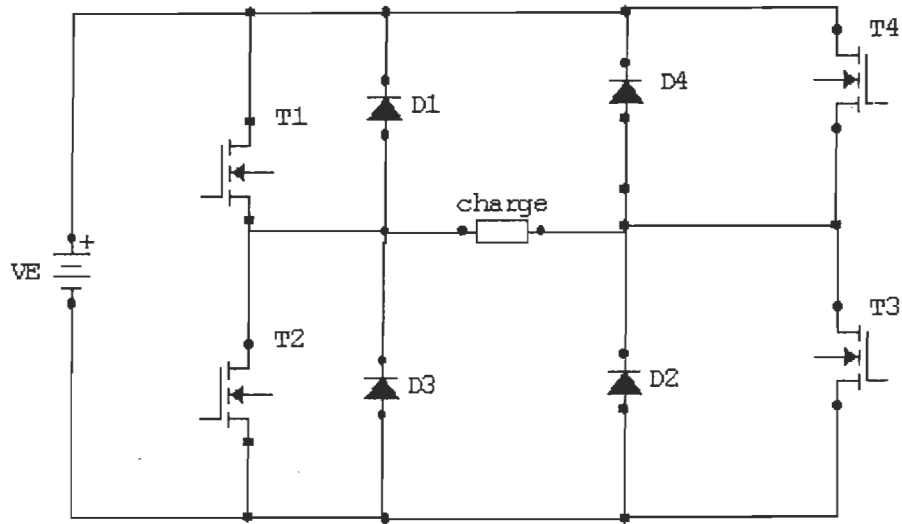
$$i_0 = \sum_{n=1,3,5,\dots}^{\infty} \frac{2V_E}{n\pi [R^2 + (nL\omega)^2]^{1/2}} \sin(n\omega t - \theta_n) \quad (1.4)$$

où  $\theta_n = \tan^{-1}(n\omega L / R)$

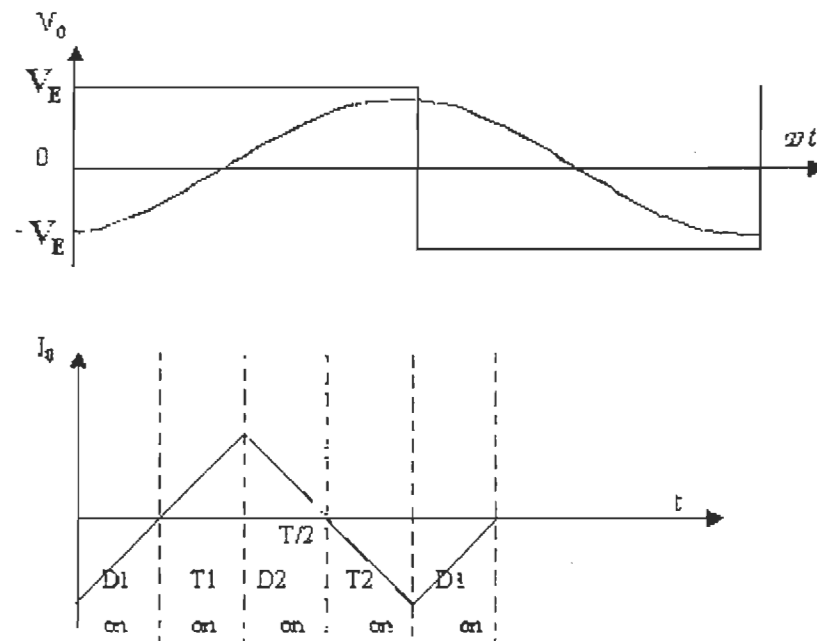
L'onduleur en demi-pont fournit à la sortie un signal de tension qui a deux états:  $+V_E/2$  et  $-V_E/2$ .

## 1.2 Onduleur monophasé en pont

La structure de base de l'onduleur de tension en pont (1. 4) se présente à la figure [1.2.a]. Il se compose de quatre interrupteurs semi-conducteurs avec des diodes en antiparallèle. La charge est branchée entre les deux bras de l'onduleur [Ch87].



**Figure 1.2.a. Onduleur monophasé de tension en pont**



**Figure 1.2.b. Signaux de fonctionnement**

Le principe de fonctionnement de cette onduleur est le suivant: quand les interrupteurs  $T_1$  et  $T_2$  sont fermés et  $T_3$  et  $T_4$  ouverts, la charge est connectée à la tension  $V_E$ , alors la tension de sortie  $V_o$  est égale à  $+V_E$  voir la figure [1.2.b]. Si les interrupteurs  $T_2$  et  $T_3$  sont fermés et les deux



autres ouverts, la tension  $V_o$  à la charge est maintenant égale à zéro,  $V_o = 0$ . Cette même valeur s'obtient si  $T_1$  et  $T_4$  sont fermés et  $T_3$  et  $T_2$  ouverts. De même, si les interrupteurs  $T_3$  et  $T_4$  sont fermés et  $T_1$  et  $T_2$  ouverts, la tension à la charge sera maintenant égale à  $V_o = -V_E$ . Ainsi, l'onduleur de tension en pont peut fournir trois niveaux de tension,  $+V_E$ , 0 et  $-V_E$ , ce qui n'est pas le cas pour l'onduleur monophasé en demi-pont, qui fournit deux niveaux,  $+V_E/2$  et  $-V_E/2$ . Cette caractéristique est un avantage de l'onduleur monophasé en pont, parce qu'elle permet de varier et régler la tension de sortie,  $V_o$ . Il faut aussi remarquer que la tension inverse maximum du blocage aux bornes des interrupteurs et des diodes de retour est la même pour les deux types d'onduleurs, si la tension  $V_E$  a la même valeur.

Cependant, les conditions sont égales pour les deux onduleurs, la puissance délivrée par l'onduleur en pont est quatre fois plus importante et son composant harmonique fondamental est deux fois plus élevé que celui de l'onduleur en demi-pont.

La tension efficace  $V_o$  de sortie pour l'onduleur, à un créneau par alternance, est égale à:

$$V_o = \sqrt{\frac{2}{T_o} \int_0^{T_o/2} V_E^2 dt} \quad \Rightarrow \quad V_o = V_E \quad (1.5)$$

et l'expression de cette tension en série de Fourier est [ 1 ]:

$$V_o = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_E}{n\pi} \sin(n\omega t) \quad (1.6)$$

$$= 0 \quad \text{pour } n = 2, 4, 6, \dots$$

où  $\omega = 2\pi f_0$  rad/s est la fréquence de pulsation de la tension. Donc, la valeur efficace de son composant fondamental serait égale à :

$$V_{o1} = \left( \frac{1}{\pi} \int_0^{\pi} \frac{(4V_E)^2}{\pi^2} \sin^2(\omega t) d(\omega t) \right)^{1/2} = \frac{2\sqrt{2}V_E}{\pi} \quad (1.7)$$

Le courant instantané de sortie  $I_o$  sur charge R-L, en série de Fourier, est donné par [1.1] :

$$i_o = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_E}{n\pi [R^2 + (nL\omega)^2]^{1/2}} \sin(n\omega t - \theta) \quad (1.8)$$

où  $\theta_n = \tan^{-1}(n\omega L / R)$

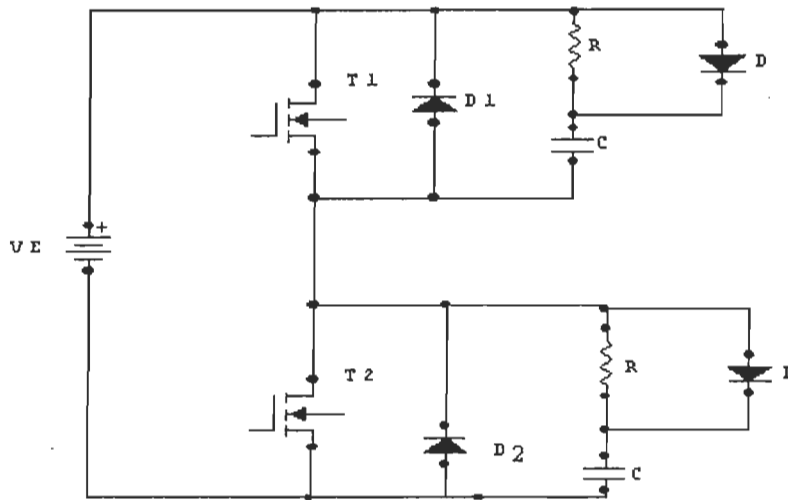
La forme d'onde du courant de sortie  $i_o$  et, les temps de conduction des interrupteurs et des diodes pour une charge inductive, sont présentés à la figure [1.2.b]. La forme d'onde est similaire à celle que l'on obtient avec l'onduleur en demi-pont, mais ce sont deux interrupteurs ou deux diodes qui conduisent en même temps, comme l'indique la figure. C'est important de noter que les deux interrupteurs d'un bras ne doivent pas conduire en même temps pour ne pas placer la source en court circuit et endommager les composants semi-conducteurs; alors la commande des interrupteurs doit prendre compte de cela [BoZi 88].

### 1.3 Commutation du courant dans un bras du convertisseur

L'objectif de cette partie du rapport est d'étudier l'évolution de la tension et du courant dans les composants d'un bras de l'onduleur pendant la commutation du courant d'un interrupteur, spécifiquement pendant son ouverture. Nous allons faire l'étude pour le bras avec un circuit **R-C-D** d'aide à la commutation.

#### **Bras avec circuit R-C-D d'aide à la commutation**

Dans la figure [1.3] ci-dessous se présente le schéma complet du bras de l'onduleur



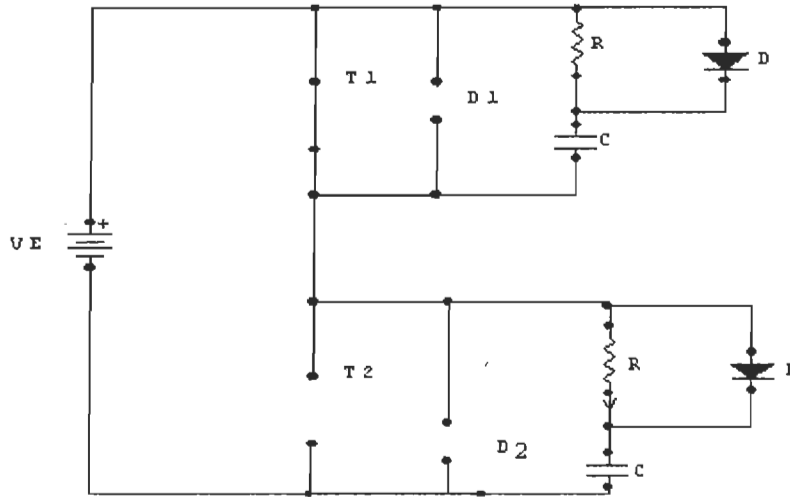
**Figure 1.3 Bras de l'onduleur avec un circuit R-C-D  
d'aide à la commutation**

Pour faire l'analyse nous allons considérer

- a) Le transistor  $T_1$  fermé et  $T_2$  ouvert
- b) La commutation de  $T_1$  assez rapide pour que pendant toute sa durée, le courant  $I_{ch}$  conserve la même valeur, alors ce courant sera appelé  $I_{ch}$ .
- c) La chute de tension de  $T_1$  en conduction négligeable par rapport à  $V_E$ .
- d) Trois périodes seront considérées :
  - 1) **Période**  $0 < t < t_f$ , où  $t_f$  est le temps de descente à zéro du courant circulant dans le transistor.
  - 2) **Période**  $t_f < t < t_1$ ; où  $t_1$  est le temps quand la diode  $D_2$  commence à conduire.
  - 3) **Période**  $t_1 < t$

**Période  $0 < t < t_f$**

Les composants qui sont en conduction pendant cette période sont  $T_1$  voir la figure [1.4].



**Figure 1.4 Bras de l'onduleur qui montre les composants en conduction pendant la période  $0 < t < t_f$**

Au moment que l'ouverture de l'interrupteur est commandée, les tensions aux bornes des condensateurs sont [SeLa89]:

$$V_{10} = 0 \quad V_{20} = V_E$$

Le courant  $i_{T1}$ , étant égal à  $I_{ch}$ , décroît linéairement après la commande d'ouverture, ce qui permet la circulation du courant à travers les condensateurs.

En utilisant les lois de Kirchoff, l'analyse du circuit résultant nous donne :

$$V_E = v_1 + v_2 + RC \frac{dv_2}{dt} \quad (1.9)$$

$$i_{T1} + i_1 = I_{ch} + i_2 \quad (1.10)$$

La descente du courant dans le transistor est considérée linéaire, alors

$$i_{T1} = I_{ch} \left(1 - t/t_f\right) \quad (1.11)$$

Les courants dans les condensateurs sont :

$$i_1 = C \frac{dv_1}{dt} \quad i_2 = C \frac{dv_2}{dt} \quad (1.12)$$

Remplaçant ces valeurs dans (1.10),

$$I_{ch} \left(1 - \frac{t}{t_f}\right) + C \frac{dv_1}{dt} = I_{ch} + C \frac{dv_2}{dt} \quad (1.13)$$

$$\text{de (1.9)} \quad \Rightarrow \quad v_1 = V_E - v_2 - RC \frac{dv_2}{dt}$$

$$\Rightarrow \quad \frac{v_1}{dt} = \frac{v_2}{dt} - RC \frac{dv_2}{dt}$$

Remplaçant cette dérivée dans l'équation (1.13) on arrive à l'équation différentielle suivante:

$$RC \frac{d^2 v_2}{dt^2} + 2 \frac{d^2 v_2}{dt^2} + I_{ch} \frac{t}{C t_f} = 0 \quad (1.14)$$

$$\frac{d^2 v_2}{dt^2} + \frac{2}{RC} \frac{dv_2}{dt} = -I_{ch} \frac{t}{RC^2 t_f}$$

$$\text{Posons} \quad A = \frac{2}{RC} = \frac{2}{\tau} \quad \text{et} \quad B = -I_{ch} \frac{1}{RC^2 t_f} = -I_{ch} \frac{1}{C \pi t_f} \quad (1.15)$$

$$\text{Où} \quad \tau = RC$$

$$\text{Aussi,} \quad v_2'' = \frac{d^2 v_2}{dt^2} \quad \text{et} \quad v_2' = \frac{dv_2}{dt}$$

$$\Rightarrow \quad v_2'' + A v_2' = B t \quad (1.16)$$

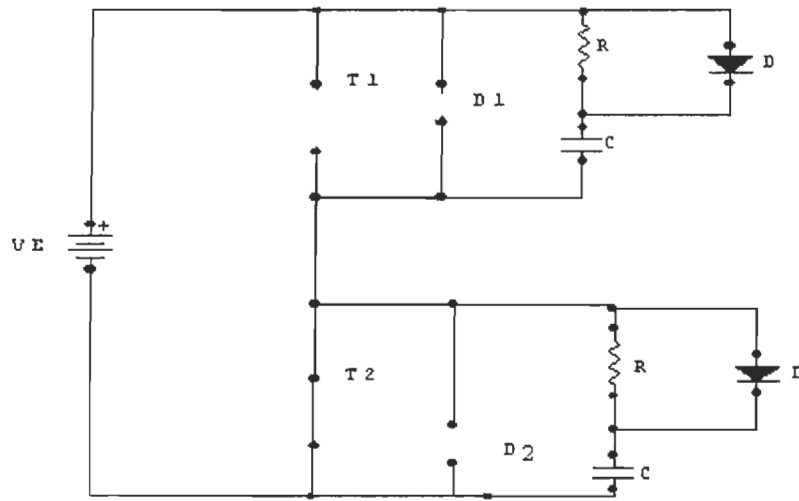
La solution totale de cette équation différentielle voir annexe

$$\Rightarrow v_1(t=t_f) = \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t_f}{\tau}} + \frac{2t_f}{\tau} \left( \frac{t_f}{\tau} + 1 \right) - 1 \right] \quad (1.17)$$

$$\text{et } \Rightarrow v_2(t=t_f) = V_E + \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t_f}{\tau}} - \frac{2t_f}{\tau} \left( \frac{t_f}{\tau} + 1 \right) - 1 \right] \quad (1.18)$$

### Période $t_f < t < t_1$

Les composants du bras qui sont en conduction pendant cette période sont montrés dans la figure [1.5]. Cette période finit quand la diode  $D_2$  commence à conduire, au temps  $t_1$ .



**Figure 1.5 Composant en conduction pendant la période  $t_f < t < t_1$**

L'analyse du circuit nous donne :

$$V_E = v_1 + v_2 + RC \frac{dv_2}{dt} \quad (1.19)$$

$$i_1 = i_2 + I_{ch}$$

$$\Rightarrow C \frac{dv_1}{dt} = C \frac{dv_2}{dt} + I_{ch} \quad (1.20)$$

De l'équation (1.19) nous trouvons que :

$$v_1 = V_E - v_2 - RC \frac{dv_2}{dt}$$

$$\Rightarrow \frac{dv_1}{dt} = -\frac{dv_2}{dt} - RC \frac{d^2 v_2}{dt^2}$$

remplaçant cette expression dans l'équation (1.20), nous arrivons à l'équation différentielle suivante :

$$\frac{d^2 v_2}{dt^2} + \frac{2}{RC} \frac{dv_2}{dt} + \frac{I_{ch}}{RC^2} = 0 \quad (1.21)$$

La solution de cette équation est trouvée par la méthode utilisée précédemment. Cette solution est donnée par les équations (1.22) et (1.23):

$$v_1 = V_E - V_{2f} - \frac{\tau}{2} \left[ \left( V_{2f}' + \frac{I_{ch}}{2C} \right) \left( 1 + e^{-\frac{2t}{\tau}} \right) - \frac{I_{ch}}{\tau C} (t + \tau) \right] \quad (1.22)$$

$$v_2 = V_{2f} - \frac{\tau}{2} \left[ \left( V_{2f}' + \frac{I_{ch}}{2C} \right) \left( 1 - e^{-\frac{2t}{\tau}} \right) - \frac{I_{ch}}{\tau C} t \right] \quad (1.23)$$

où  $V_{2f}$  = la valeur initiale de  $v_2$  pour cette période, soit à  $t = t_f$   
 $V_{2f}'$  = la valeur initiale de  $v_2'$  pour cette période, soit à  $t = t_f$   
 $V_E$  = tension d'entrée du convertisseur  
 $\tau = RC$

Les expressions pour les courants dans les condensateurs sont :

$$i_1 = C \left[ \left( V_{2f}' + \frac{I_{ch}}{2C} \right) \left( \frac{I_{ch}}{2} + e^{-\frac{2t}{\tau}} \right) \right] \quad (1.24)$$

$$i_2 = C \left[ \left( V_{2f}' + \frac{I_{ch}}{2C} \right) \left( \frac{I_{ch}}{2} - e^{-\frac{2t}{\tau}} \right) \right] \quad (1.25)$$

La tension aux bornes de la diode  $D_2$  est égale à :  $V_{D2} = V_1 - V_E$

$$\Rightarrow v_{D2} = -V_{2f} - \frac{\tau}{2} \left[ \left( V_{2f} + \frac{I_{ch}}{2C} \right) \left( 1 + e^{-\frac{2}{\tau}t} \right) - \frac{I_{ch}}{\tau C} (t + \tau) \right] \quad (1. 26)$$

Cette période se termine quand la diode  $D_2$  commence à conduire, c'est-à-dire, quand la tension  $V_{D2}$  devient positive, à  $t_1$ .

### Période $t_1 < t$

Le circuit qui montre les composants en conduction pendant cette période est présenté dans la figure [1.6]. Même que la diode  $D_2$  commence à conduire et que le condensateur d'aide à la commutation du  $T_1$  est déjà chargé,  $V_1 = V_E$ , le condensateur du circuit **R -C** de  $T_2$  n'est pas complètement déchargé, alors il y a encore un courant qui circule sur ce circuit. Donc, ceci nous emmène à écrire les équations des tensions et des courants suivants:

$$V_E = v_1 + v_2 + RC \frac{dv_2}{dt} \quad (1. 27)$$

$$\text{et} \quad i_{d2} = I_{ch} + C \frac{dv_2}{dt} \quad (1. 28)$$



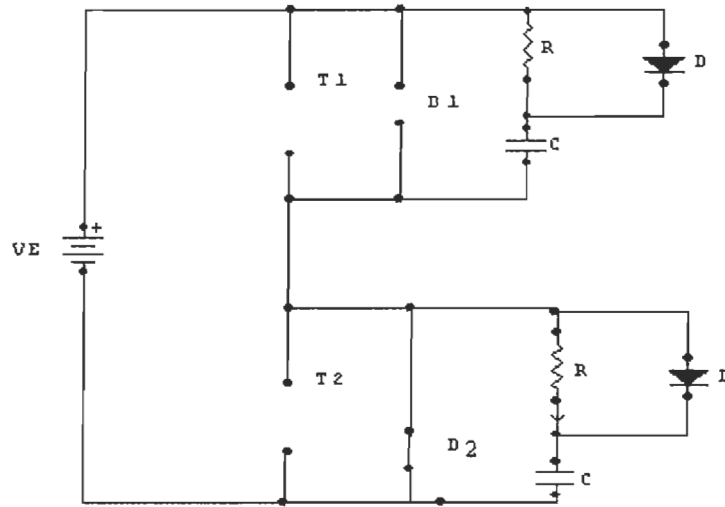


Figure 1.6 Composants du bras en conduction pendant la période  $t_2 < t$

$$v_1 = V_E \quad 0 = v_2 + RC \frac{dv_2}{dt}$$

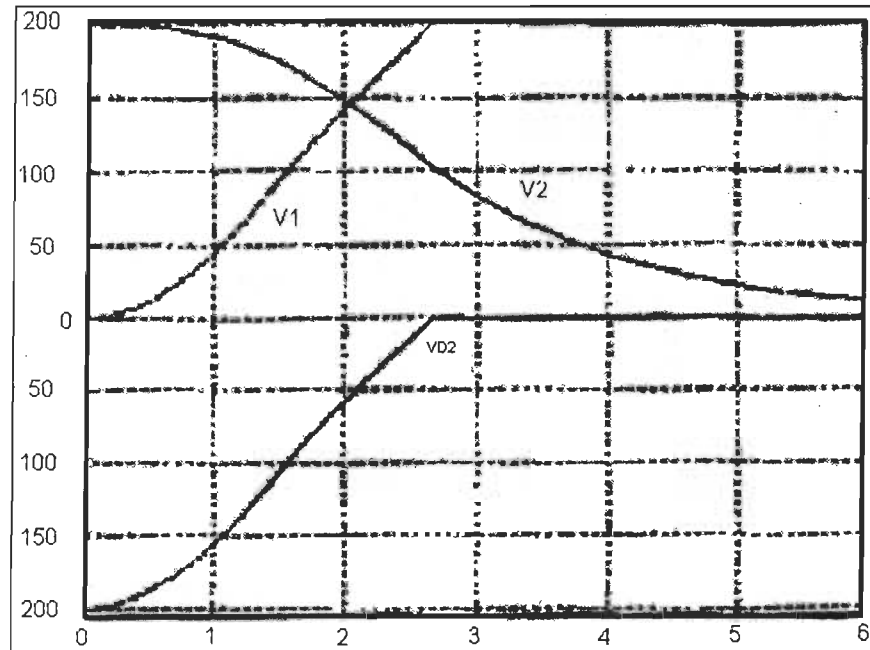
La solution de cette équation est :

$$v_2 = V_{20} e^{-t/RC} \quad \text{et} \quad i_2 = -\frac{V_{20}}{R} e^{-t/RC} \quad (1.29)$$

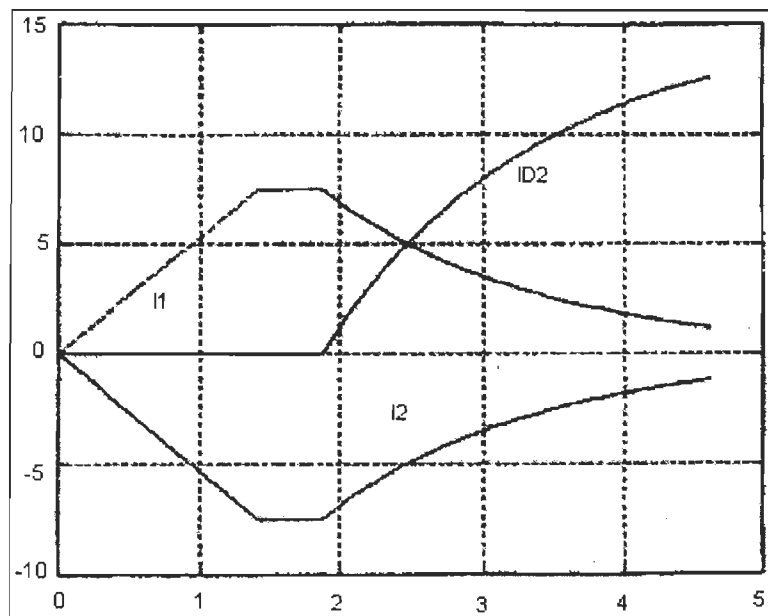
$$\Rightarrow \quad i_{D2} = I_{ch} - \frac{V_{20}}{R} e^{-t/RC} \quad (1.30)$$

où  $V_{20}$  = la tension initiale de  $V_2$  pour cette période

En utilisant les expressions obtenues ci-dessus, les évolutions des courants et des tensions dans les différents composants sont montrées dans les figures [1.7] et [1.8]. Les évolutions considérées sont pour le cas de la commutation du courant de  $T_1 \rightarrow D_2$ , c'est-à-dire, pendant l'ouverture de l'interrupteur  $T_1$ , (ou d'autre interrupteur) et avec un circuit **R-C-D** d'aide à la commutation.



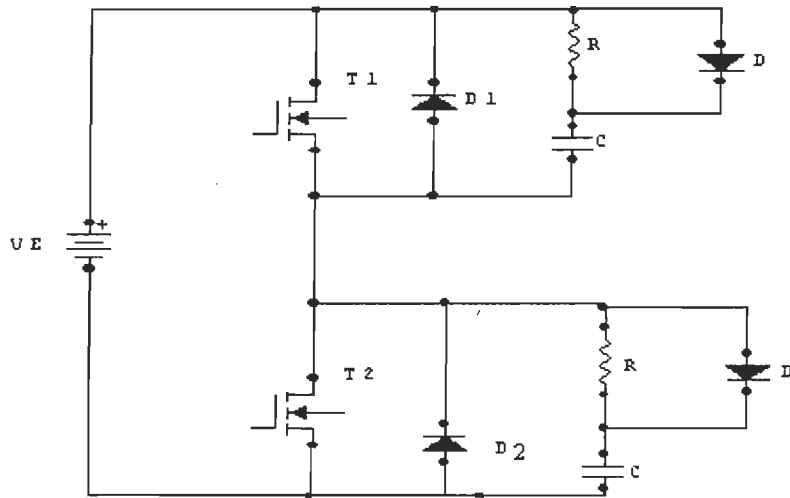
**Figure 1.7 Évolution des tensions  $V_1$ ,  $V_2$  et  $V_{D2}$  pendant l'ouverture d'un interrupteur avec un circuit R-C-D d'aide à la commutation**



**Figure 1.8 Évolution des courants  $I_1$ ,  $I_2$  et  $I_{D2}$  pendant l'ouverture d'un interrupteur avec un circuit R-C-D d'aide à la commutation.**

### Période $0 < t < t_f$

La circulation du courant pendant cette période est mise en évidence avec les équations établie à partir de la figure [1.9].



**Figure 1.9 Circulation du courant dans le bras du convertisseur pendant la période  $0 < t < t_f$ .**

Les valeurs initiales des condensateurs sont :  $V_{10} = 0$  et  $V_{20} = V_E$ . A  $t = 0$ , le transistor se bloque, mais le courant continue à circuler dans le transistor et à décroître d'une façon linéaire jusqu'à ce qu'il devient égale à zéro,  $t = t_f$ . Alors le courant  $I_{T1}$  a la forme suivante :

$$I_{T1} = I_{ch} \left( 1 - t/t_f \right) \quad (1.31)$$

L'équation différentielle des tensions est :

$$V_E = v_1 + RC \frac{dv_1}{dt} + v_2 + RC \frac{dv_2}{dt} \quad (1.32)$$

$$V_E = (v_1 + v_2) + RC \frac{d(v_1 + v_2)}{dt} \quad (1.33)$$

La solution de cette équation différentielle est de la forme :

$$v_1 + v_2 = V_E + Ae^{-t/RC} \quad (1.34)$$

La constante A est trouvée en utilisant les valeurs initiales  $V_{10}$  et  $V_{20}$ . Alors, dès que

$V_{10} + V_{20} = V_E$  à  $t = 0$ , cela signifie que  $A = 0$  ; donc la solution devient

$$V_1 + V_2 = V_E$$

Remplaçant cette expression dans l'équation (1.32) nous donne:

$$C \frac{dv_1}{dt} = -C \frac{dv_2}{dt} \quad (1.35)$$

Maintenant, en utilisant la loi de Kirchoff pour les courants, nous avons :

$$i_{T1} + C \frac{dv_1}{dt} = I_{ch} + C \frac{dv_2}{dt} \quad (1.36)$$

Cette relation peut s'écrire comme suit :

$$I_{ch}(1 - t/t_f) + C \frac{dv_1}{dt} = I_{ch} + C \frac{dv_1}{dt} \quad (1.37)$$

$$\Rightarrow \frac{dv_1}{dt} = \frac{I_{ch}t}{2Ct_f}$$

$$\Rightarrow v_1 = I_{ch} \frac{t^2}{4Ct_f} \quad et \quad v_2 = V_E - I_{ch} \frac{t^2}{4Ct_f} \quad (1.38)$$

Et les courants dans les condensateurs sont donnés par les expressions :

$$i_1 = I_{ch} \frac{t}{2t_f} \quad et \quad i_2 = -I_{ch} \frac{t}{2t_f} \quad (1.39)$$

$$\text{Or, } V_{T1} = V_1 + Ri_1 \quad et \quad V_{D2} = V_{T1} - V_E \quad (1.40)$$

$$\Rightarrow v_{T1} = v_1 + RC \frac{dv_1}{dt} = I_{ch} \frac{t^2}{4Ct_f} + RI_{ch} \frac{t}{2t_f} \quad (1.41)$$

$$\Rightarrow v_{T1} = \frac{RI_{ch}}{2t_f} \left( \frac{t^2}{2RC} + t \right) \quad (1.42)$$

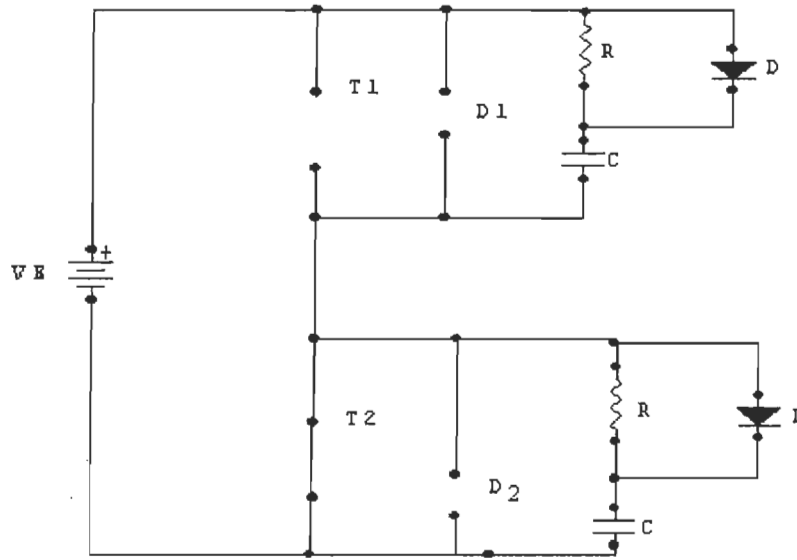
$$\text{et } v_{D2} = \frac{RI_{ch}}{2t_f} \left( \frac{t^2}{2RC} + t \right) - V_E \quad (1.43)$$

Les valeurs des tensions  $V_1$  et  $V_2$  à  $t = t_r$  deviennent les valeurs initiales de ces deux tensions pour la période

$$\Rightarrow v_1(t_f) = I_{ch} \frac{t_f}{4C} \quad \text{et} \quad v_2(t_f) = V_E - I_{ch} \frac{t_f}{4C}$$

### **Période ( $t_f < t < t_l$ )**

Pendant cette période le transistor  $T_1$  est bloqué et son courant est égal à zéro,  $i_{T1} = 0$  ; la tension  $v_{T1}$  dans ces bornes est inférieure à  $V_E$ , alors la tension  $V$  aux bornes de la diode  $D_2$  est négative, donc cette diode est aussi bloquée. La figure [1.10] ci-dessous montre le circuit et les composants qui sont en conduction pendant cette période.



**Figure 1.10 Circulation du courant dans le bras du convertisseur pendant la période  $(t_f < t < t_1)$  de l'ouverture d'un interrupteur.**

Les équations de tension et du courant pour ce circuit sont résolues de la même façon que pour celle de la première période; ainsi, l'équation des tensions est toujours la même, c'est-à-dire, égale à:

$$V_E = v_1 + RC \frac{dv_1}{dt} + v_2 + RC \frac{dv_2}{dt} \quad (1.44)$$

ce qui donne les même résultats, soit

$$v_1 + v_2 = V_E \quad ; \quad C \frac{dv_1}{dt} = -C \frac{dv_2}{dt} \quad (1.45)$$

L'équation des courants est cependant, dans ce cas, égale à:

$$C \frac{dv_1}{dt} = -C \frac{dv_2}{dt} + I_{ch} \quad (1.46)$$

$$C \frac{dv_1}{dt} = \frac{I_{ch}}{2} \quad (1.47)$$

$$v_1(t = t_1) = V_E - \frac{RI_{ch}}{2} \quad \text{et} \quad v_2(t = t_1) = \frac{RI_{ch}}{2} \quad (1.48)$$

Les tensions aux bornes de  $T_1$  et  $D_2$  sont encore données par l'équation (1.40), soit

$$v_{T1} = v_1 + Ri_1 \quad \text{et} \quad v_{D2} = v_{T1} - V_E$$

$$\Rightarrow \quad v_{T1} = \frac{I_{ch}}{2C} \left( t - \frac{t_f}{2} + RC \right) \quad (1.49)$$

$$v_{D2} = \frac{I_{ch}}{2C} \left( t - \frac{t_f}{2} + RC \right) - V_E \quad (1.50)$$

Cette période se termine quand la tension aux bornes de la diode  $D_2$  devient égale à zéro, soit pour  $t = t_1$ . Ce temps  $t_1$  peut se trouver de l'expression (1.50)

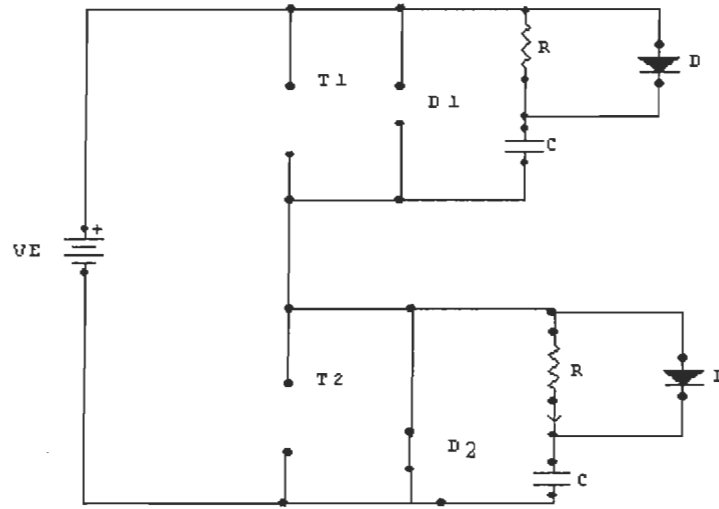
$$\text{Alors} \quad t_1 = \frac{2CV_E}{I_{ch}} + \frac{t_f}{2} - RC$$

Remplaçant cette expression de  $t_1$  dans les équations (1.48), on trouve les valeurs initiales de  $v_1$  et  $v_2$  pour la période suivante égales à :

$$v_1(t = t_1) = V_E - \frac{RI_{ch}}{2} \quad \text{et} \quad v_2(t = t_1) = \frac{RI_{ch}}{2} \quad (1.51)$$

### Période ( $t_1 < t$ )

Dans cette période la diode  $D_2$  conduit, mais encore un courant circule dans les circuits R-C.



**Figure 1.11** Circulation du courant dans le bras du convertisseur pendant la période  $t_1 < t$

Nous trouvons toujours les expressions suivantes :

$$V_1 + V_2 = V_E \quad ; \quad C \frac{dv_1}{dt} = -C \frac{dv_2}{dt}$$

En considérant que la diode  $D_2$  conduit, nous avons aussi :

$$V_E = V_1 + RC \frac{dv_1}{dt} \quad (1.52)$$

La solution de cette équation est donné par :

$$V_1 = V_E - \frac{RI_{ch}}{2} e^{(t_2-t)/RC} \quad (1.53)$$

$$\text{et } V_2 = \frac{RI_{ch}}{2} e^{(t_2-t)/RC} \quad (1.54)$$

finalement, pour le courant dans la diode  $D_2$  l'expression suivante a été trouvée:

$$i_{D2} = I_{ch} (1 - e^{(t_1-t)/RC}) \quad (1.55)$$

En utilisant les expressions obtenues, les évolutions des courants et des tensions dans les différents composants du bras du convertisseur sont montrés dans les figures [1.12] et [1.13]. Les évolutions sont pour le cas de la commutation du courant de  $(T_1 D_2)$ , c'est-à-dire, pendant l'ouverture de l'interrupteur  $T_1$ , (ou autre interrupteur) et avec un circuit **R-C** d'aide à la commutation [Ch87].



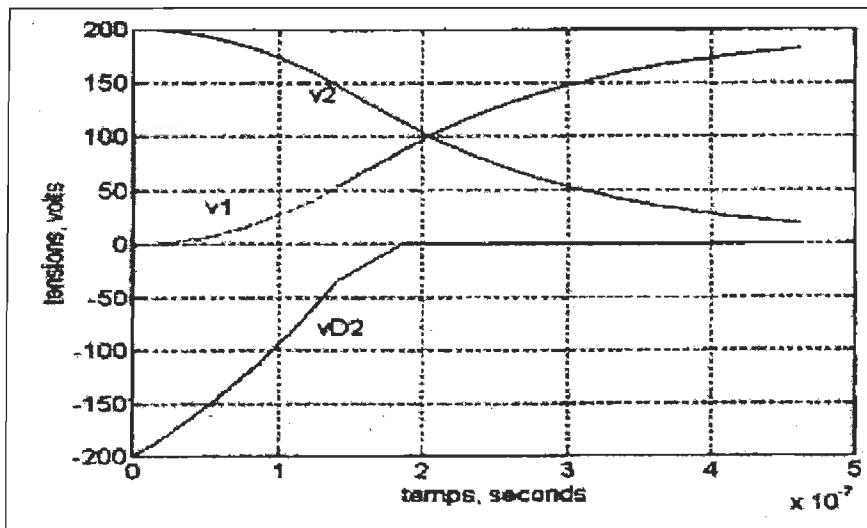


Figure 1.12 Évolution des tensions  $V_1$ ,  $V_2$  et  $V_{D2}$  pendant l'ouverture d'un interrupteur avec circuit R-C d'aide à la commutation.

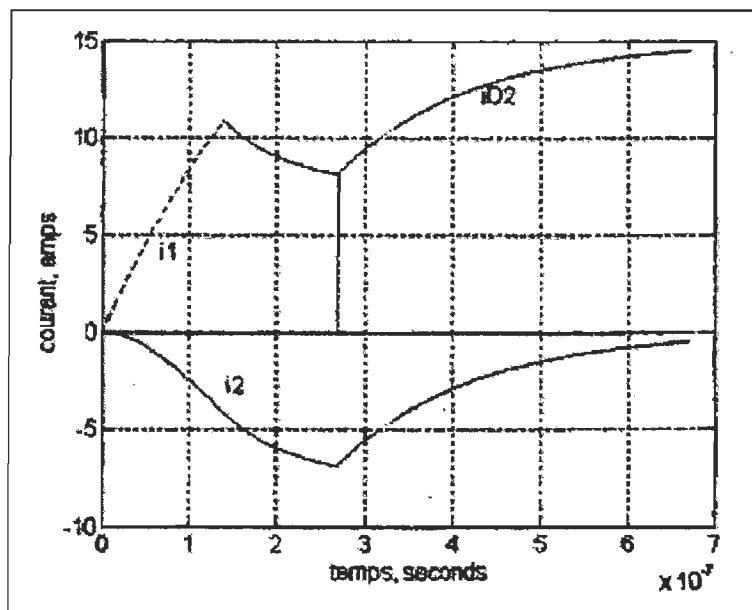


Figure 1.13 Évolution des courants  $I_1$ ,  $I_2$ , et  $I_{D2}$  pendant l'ouverture d'un interrupteur avec circuit R-C d'aide à la commutation.

## 1.4 Pertes dans l'onduleur

La dissipation de puissance dans l'onduleur est due à la conduction et à la commutation des transistors. Les pertes considérées dans notre onduleur sont:

- Les pertes par conduction
- Les pertes par commutation des transistors

En général, la dissipation de puissance est donnée par l'expression suivante:

$$P = \frac{1}{T} \int v i \, dt \quad (1.56)$$

### 1.4.1 Pertes de conduction dans l'onduleur par conduction

Les pertes par conduction sont celles dissipées dans les transistors et les diodes quand ils sont en conduction. Ces pertes peuvent se calculer avec l'expression suivante:

$$P_e = f V_{\text{moy}} I_{\text{moy}} \quad (1.57)$$

où

$V_{\text{moy}}$  = chute de tension moyenne

$I_{\text{moy}}$  = courant moyen

### 1.4.2 Pertes de commutation dans l'onduleur par commutation

Les pertes par commutation sont celles dissipées dans les interrupteurs et dans les circuits d'aide à la commutation. Dans ce travail nous avons considéré seulement les pertes dues à l'ouverture des interrupteurs. L'interrupteur utilisé est l'**IGBT**, sa rapidité de fermeture permet d'obtenir des pertes négligeables. Les expressions des pertes de puissance par l'ouverture des interrupteurs sont données ci-dessous pour les deux types de circuits d'aide à la commutation, le circuit **R-C-D** et le circuit **R-C**, dont l'évolution des courants et des tensions ont été étudiés à la section 1.3.

#### 1.4.2.1 Pertes dans l'onduleur avec circuit R-C-D d'aide à la commutation.

Dans cette étude nous avons constaté que pendant le blocage d'un transistor, le courant ne circule dans celui-ci qu'au cours de la première étape de la commutation du courant, c'est-à-dire, pour l'étape  $0 \rightarrow t_f$ , où  $t_f$  est le temps de descente du courant dans le transistor. Donc, c'est seulement à cette étape que la dissipation de puissance se produit dans le transistor [BoZi 88]. Ce courant est donné par:

$$i_{T1} = I_{ch} \left( 1 - \frac{t}{t_f} \right) \quad (1.58)$$

La tension aux bornes du transistor  $v_T$  est égale à la tension  $v_1$  aux bornes du condensateur C,  $v_T = v_1$ . Cette tension est donnée par l'expression (1.27), soit,

$$v_1 = \frac{I_{ch} R_\tau}{8t_f} \left[ e^{-\frac{2t}{\tau}} + \frac{2t}{\tau} \left( \frac{t}{\tau} + 1 \right) - 1 \right]$$

Ainsi, la dissipation de puissance dans le transistor à son ouverture est égale à:

$$\begin{aligned} P_{T1} &= \frac{1}{T} \int_0^{t_f} v_1 i_{T1} dt \\ &= \frac{1}{T} \int_0^{t_f} \frac{I_{ch} R_\tau}{8t_f} \left[ e^{-\frac{2t}{\tau}} + \frac{2t}{\tau} \left( \frac{t}{\tau} + 1 \right) - 1 \right] \left[ I_{ch} \left( 1 - \frac{t}{t_f} \right) \right] dt \end{aligned}$$

À l'aide du logiciel MAPL V, nous avons trouvé l'expression suivante pour cette intégrale:

$$P_{T1} = \frac{f R_\tau I_{ch}^2}{32t_f} \left[ \frac{\tau^2}{t_f} \left( e^{-\frac{2t_f}{\tau}} - 1 \right) + \frac{4t_f^2}{3\tau} \left( 1 + \frac{t_f}{2\tau} \right) + 2(\tau - t_f) \right] \quad (1.59)$$

Si nous considérons que dans chaque période T, il y a p nombre des impulsions, la puissance dissipée dans les transistor pendant une période T sera donc égale à  $P_{T1}$ .

Pour les circuits d'aide à la commutation **R-C-D**, il y a dissipation de puissance seulement dans la résistance du CALC du transistor  $T_2$  et pendant les trois périodes, donc,

$$P_R = fR \left[ \int_0^{t_1'} i_2^2 dt + \int_{t_1'}^{t_1} i_2^2 dt + \int_{t_1}^T i_2^2 dt \right]$$

Les expressions pour le courant  $i_2$  dans chaque étape sont données par les équations (1.30), (1.39), (1.41). Remplaçant ces expressions et faisant les intégrales,  $P_R$  est calculé. Il faut multiplier par le nombre  $p$  des impulsions par période  $T$  pour obtenir la puissance dissipée par période  $T$ .

#### 1.4.2.2 Pertes dans l'onduleur avec circuit R-C d'aide à la commutation

Les expressions pour les pertes de puissance dans cet onduleur ont été obtenues de façon similaire à celles de la section précédente. Pour les pertes dans le transistor l'expression est:

$$P_{T1} = fp \frac{R_{tr} I_{ch}^2}{12} \left( \frac{t_f}{4\tau} + 1 \right) \quad (1.60)$$

Tandis que pour les circuits CALC, la dissipation de puissance se produit seulement dans les deux résistances, donc,

$$P_R = 2fp R \frac{I_{ch}^2}{4} \left( 2 \frac{V_E}{I_{ch}} - \frac{t_f}{6} - \frac{\tau}{2} \right) \quad (1.61)$$

Où

- $t_f$  = temps de descente du transistor
- $f$  = fréquence de la signal de sortie
- $p$  = nombre des impulsions par période  $T$
- $\tau = R \cdot C$
- $V_E$  = tension d'entrée à l'onduleur
- $I_{ch}$  = courant dans le transistor à son ouverture

## 1.5 Performance des convertisseurs

La performance des convertisseurs s'évalue en considérant la grandeur du contenu harmonique dans son signal de sortie. Parmi les paramètres de performance se trouve [BoZi88]:

- Le facteur d'harmonique,  $FH_n$ . C'est la mesure de la contribution de la  $n^{eme}$  harmonique à la forme d'onde du signal de sortie [PaHo73].
- Le facteur de distorsion,  $FD$ . C'est la mesure de l'efficacité de l'onduleur pour réduire des harmoniques non-désirées sans spécifier les valeurs du filtre de la charge.
- Le taux de distorsion harmonique total,  $TDHT$ . C'est la distorsion du signal de sortie mesurée par rapport à son composant fondamental. Sa définition en forme d'équation est:

$$TDHT = \frac{1}{V} \left( \sum_{n=2,3,\dots}^{\infty} V_n^2 \right)^{\frac{1}{2}} \quad (1.62)$$

où  $TDHT$  = Taux de distorsion harmonique total.

$V_n$  = Valeur efficace de la  $n^{ieme}$  harmonique.

La performance de notre convertisseur sera mesurée en utilisant ce dernier paramètre.

## 1.6 Conclusion

Dans ce chapitre, nous avons présenté une révision de la littérature traitant les onduleurs à modulation de largeur d'impulsions MLI, les topologies de l'onduleur en demi- pont et en pont complet, l'étude des évolutions de tension et du courant pendant l'ouverture des interrupteurs, les pertes dans l'onduleur, et finalement les paramètres de performance des onduleurs. Ceci va nous permettre de développer en détail la commande MLI dans le deuxième chapitre.

# *Chapitre 2*

## ***COMMANDE DES ONDULEURS DE TENSION À MODULATION DE LARGEUR D'IMPULSIONS (MLI)***

Dans la pratique, beaucoup d'applications industrielles reçoivent leur alimentation d'une source de tension qui est stable et réglable. Cette tension est souvent obtenue avec des onduleurs qui éliminent les variations de la tension continue d'entrée, et maintiennent constante la relation tension/fréquence tout en réglant l'amplitude de la tension requise par la charge.

Plusieurs méthodes sont utilisées pour obtenir la tension désirée et l'une d'elle très efficace est appelée "**Modulation de Largeur d'Impulsions (MLI)**". En plus de régler l'amplitude, cette méthode contrôle le contenu harmonique de la tension de sortie de l'onduleur, en poussant les

harmoniques d'ordre faible vers des fréquences plus élevées. Cela permet que le filtrage de cette tension soit plus facile et peu coûteux, car la taille des composants du filtre est plus petite.

Il est important de mentionner que la technique MLI a des limitations par rapport à la fréquence d'opération des onduleurs, plus élevée est la fréquence, plus hautes sont les pertes dues à la commutation des interrupteurs semi-conducteurs; de même les stress dans ces dispositifs étant aussi élevés, leurs vies d'utilisation en sont diminuées. La fréquence d'opération des onduleurs à MLI est aussi limitée par la vitesse de commutation propre des interrupteurs semi-conducteurs.

Plusieurs techniques de contrôle à MLI ont été développées, les plus utilisées sont les suivantes:

- Modulation MLI d'une seule impulsion,
- Modulation MLI de multiples impulsions,
- Modulation MLI sinusoïdale,
- Modulation MLI précalculés.

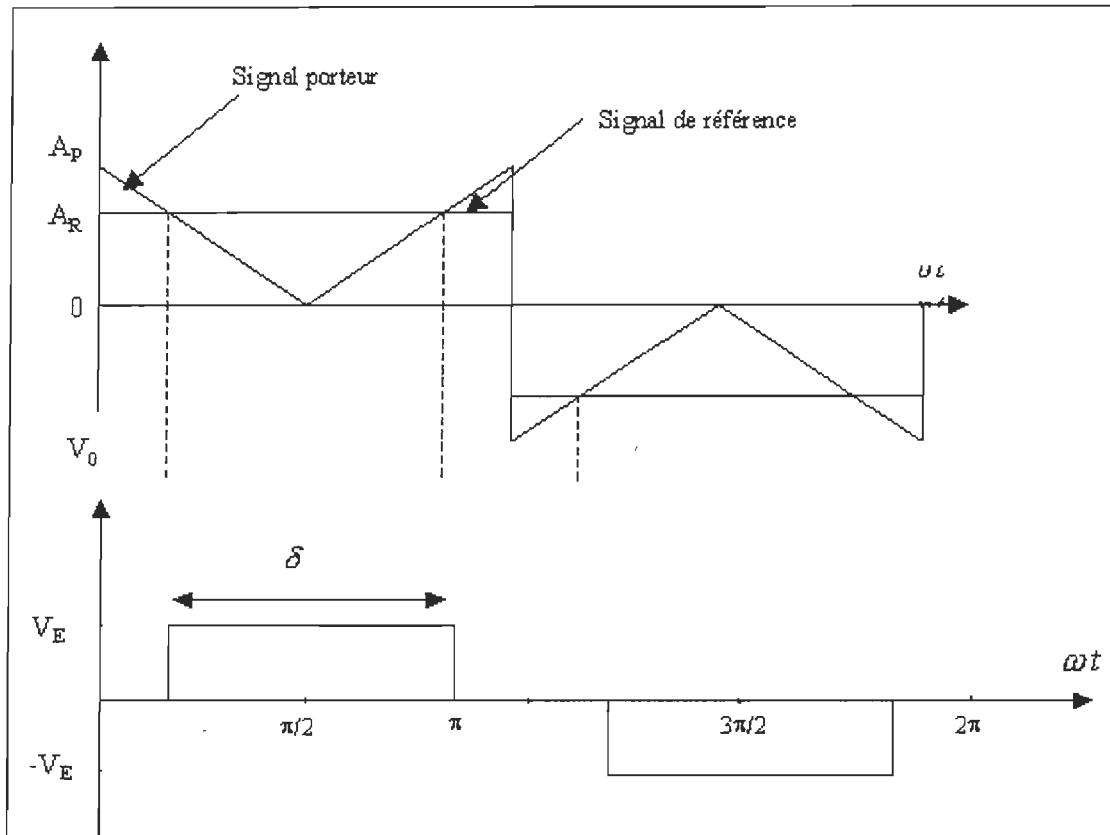
Dans ce chapitre, nous allons brièvement réviser les trois premières techniques et plus en détail la dernière. Dans la référence [BoZi 88] se présente une évaluation des techniques de MLI plus avancées, telles que: la technique MLI sinusoïdale modifiée, la technique MLI avec l'injection de la 5ème harmonique.

## **2.1 Modulation MLI d'une seule impulsion**

La technique de modulation MLI d'une seule impulsion contrôle la tension de sortie efficace en changeant la largeur de la seule impulsion par alternance que l'on trouve avec ce type de commande. Les signaux de commande des interrupteurs se génèrent en faisant la comparaison d'un signal triangulaire de référence, d'amplitude  $A_p$ , avec un signal porteur rectangulaire d'amplitude  $A_r$ . La figure [1.2] montre ces deux signaux et la forme de la tension de sortie pour un onduleur en pont. Les angles de commutation ou de commande sont définis par l'intersection des deux signaux; la largeur  $\delta$  de l'impulsion est définie par la différence entre ces angles et elle peut avoir une valeur de 0 à  $\pi$ , qui s'obtient en variant l'amplitude  $A_r$ , de zéro à  $A_p$ . La relation



$A_r/A_p$  est la variable de contrôle et s'appelle "index de modulation d'amplitude" de la tension de sortie.



**Figure 2.1 Modulation MLI d'une seule impulsion**

La tension efficace de sortie  $V_o$  pour un onduleur en pont avec une seule impulsion par alternance est donnée par:

$$V_o = \left[ \frac{2}{2\pi} \int_{(\pi-\delta)/2}^{(\pi+\delta)/2} V_E^2 d(\omega t) \right]^{\frac{1}{2}} = V_E \sqrt{\frac{\delta}{\pi}} \quad (2.1)$$

Alors, la tension de sortie  $V_o$  dépend de la largeur d'impulsion  $\delta$ .

Et l'expression pour la tension instantanée  $V_o$  en série de Fourier:

$$V_0 = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_E}{n\pi} \sin(n\omega t) \quad (2.2)$$

## 2.2 Modulation MLI de multiples impulsions

La technique précédente module la largeur d'impulsion pour varier la valeur efficace de la tension de sortie  $V_0$ . Lorsque, si l'on veut réduire le contenu harmonique de cette tension, il faut avoir plus d'une impulsion par demi-cycle, et pour en obtenir l'utilisation de la technique de modulation MLI de multiples impulsions est souvent requise. Les impulsions de commande des interrupteurs s'obtiennent d'une façon similaire à celle de la technique précédente, avec les intersections d'un signal de référence et d'un signal porteur. La fréquence du signal porteur est plus élevée que la fréquence du signal de référence, ce qui fait la différence principale de cette technique est qu'elle génère plus d'une impulsion à la sortie. La fréquence du signal de référence  $f_r$  détermine la fréquence de la tension de sortie tandis que la fréquence du signal porteur  $f_p$ , dénombre les impulsions par demi-cycle. L'expression suivante donne le nombre  $p$  des impulsions:

$$p = \frac{f_p}{2f_r} = \frac{m_f}{2} \quad (2.3)$$

$$\text{où } m_f = \frac{f_p}{f_r} \text{ est le rapport de modulation de fréquences.} \quad (2.4)$$

Cette technique est également nommée "modulation uniforme de largeur des impulsions (en anglais **UPWM**)" car toutes les impulsions ont la même largeur. La largeur  $S$  des impulsions peut se changer de  $0$  à  $n/p$  avec la variation de l'index de modulation  $M(=A_r/A_p)$ . L'expression de la tension efficace de sortie  $V_0$  pour cet onduleur est:

$$V_0 = \left[ \frac{2p}{2\pi} \int_{(\pi p - \delta)/2}^{(\pi p + \delta)/2} V_E^2 d(\omega t) \right]^{1/2} = V_E \sqrt{\frac{p\delta}{\pi}} \quad (2.5)$$

Le facteur de distorsion du signal de sortie est de beaucoup réduit par rapport à celui de l'onduleur d'une seule impulsion par alternance, mais les pertes de commutation croissent par le nombre plus élevé des commutations requises. Il faut remarquer que pour un nombre grand des impulsions  $p$ , les amplitudes des harmoniques de bas ordre sont beaucoup plus petites, mais pour quelques harmoniques de haute fréquence, leur amplitude augmente.

### 2.3 Modulation MLI sinusoïdale

Cette technique est similaire à la technique précédente, mais ici le signal de référence est un signal sinusoïdal. Ce changement donne comme résultat des impulsions de différentes largeurs, proportionnelles à l'amplitude de l'onde de référence. Les signaux de commande des interrupteurs sont toujours générés par l'intersection d'un signal porteur triangulaire de fréquence  $f_p$  avec un signal sinusoïdal de référence de fréquence  $f$ . La figure [2.2] montre la tension de sortie.

La fréquence  $f$ , du signal de référence détermine la fréquence du signal de sortie de l'onduleur, alors que l'amplitude maximum  $A$  contrôle l'index de modulation  $M$ , qui à son tour contrôle la tension efficace de sortie. La fréquence  $f_p$  du signal porteur détermine le nombre  $p$  des impulsions par demi-cycle.

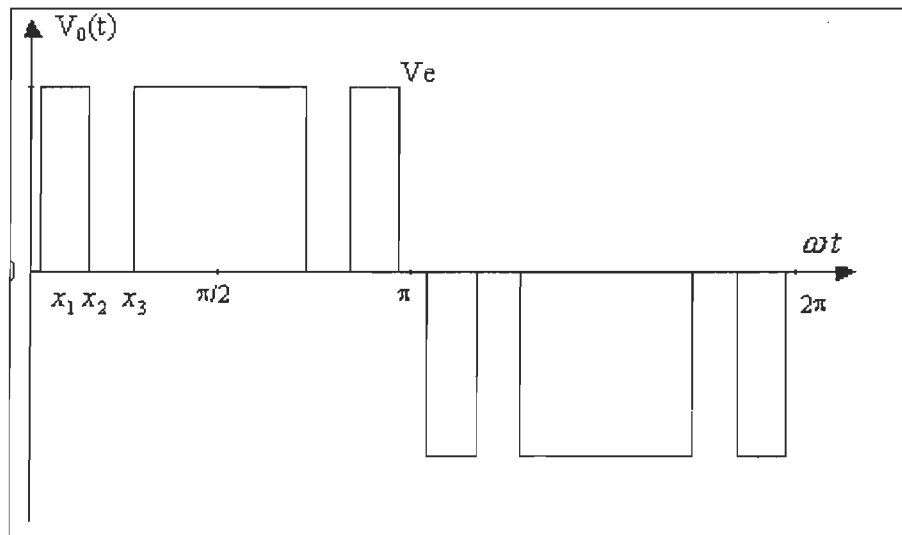


Figure 2.2 Modulation MLI sinusoïdale

La tension efficace  $V_o$  de sortie est donnée par:

$$V_o = V_E \left( \sum_{m=1}^p \frac{\delta_m}{\pi} \right)^{1/2} \quad (2.6)$$

où  $\delta_m = L$  *arg eur* du  $m^{\text{eme}}$  impulsion et  $p$  est le nombre des impulsions par alternance.

Ce résultat s'obtient de façon similaire à celui donné par l'équation (2.5).

La technique MLI sinusoïdale réduit de façon significative le facteur de distorsion de la tension de sortie, beaucoup plus que celui de la modulation MLI de multiples impulsions. Toutes les harmoniques d'ordre égal ou inférieur à  $2p-1$  sont éliminées par cette technique.

## 2.4 Modulation MLI calculée

Les techniques de modulation présentées précédemment sont des techniques qui déterminent les angles de commutation des interrupteurs en temps réel. Ces angles peuvent aussi se calculer de façon préalable et convenable, tenant compte du nombre des harmoniques de faible ordre que l'on désire éliminer ou pousser vers des fréquences plus élevées. Une fois les angles calculés, le motif de commande peut se générer préalablement et se mémoriser dans un **EPROM** [MhRa93] pour commander les interrupteurs de l'onduleur.

Le développement de la technique [PaHo73] tient compte de la périodicité et de la symétrie de la tension de sortie  $V_o$  en utilisant la théorie de la série de Fourier. En général, une fonction périodique et symétrique en demi-cycle peut s'exprimer en série de Fourier comme suit:

Soit  $f(\omega t) = -f(\omega t + \pi)$  une fonction symétrique en demi-cycle.

La série de Fourier de cette fonction serait:

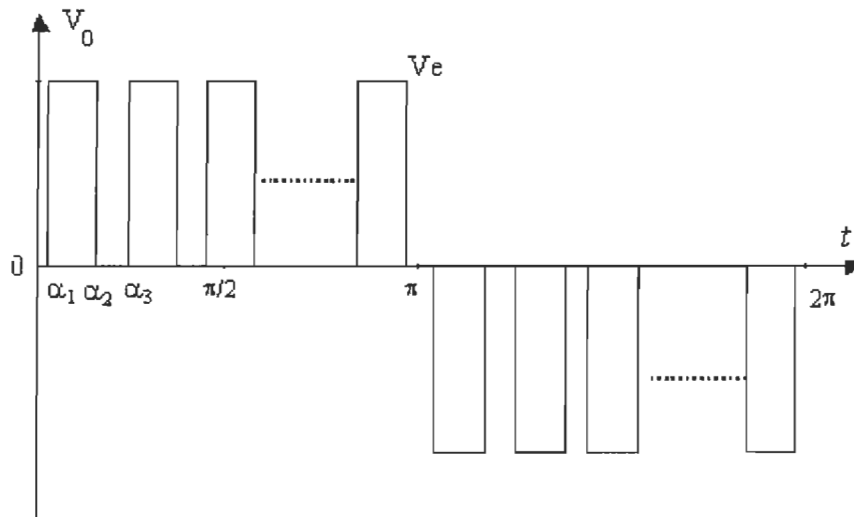
$$\Rightarrow f(\omega t) = \sum_{n=1}^{\infty} [a_n \sin(n\omega t) + b_n \cos(n\omega t)] \quad (2.7)$$

où  $a_n$  et  $b_n$  sont des constants données par:

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \sin(n\omega t) d(\omega t)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \cos(n\omega t) d(\omega t) \quad (2.8)$$

La figure [2.3] montre la forme d'onde de la tension de sortie  $V_o$  d'un onduleur en pont,  $p$  indique le nombre des impulsions par demi-cycle et  $\alpha_k$  les angles de commutations [Ch87].



**Figure 2.3 Tension de sortie de l'onduleur en pont**

Considérons que la forme d'onde est périodique et symétrique impair en quart d'onde, c'est-à-dire,

$$V_o(\omega t) = V_o(\pi - \omega t)$$

Pour une fonction de ce type, les coefficients  $a_n$  et  $b_n$  en série de Fourier sont égaux à:

$$a_n = \frac{4}{\pi} \int_0^{\pi/2} f(\omega t) \sin(n\omega t) d(\omega t) \quad (2.9)$$

et  $a_n = 0$  pour  $n = \text{pair}$ ,  
Tandis que  $b_n = 0$  pour tous  $n$ .

Donc, l'expression de  $V_0$  en série de Fourier serait:

$$V_0(\omega t) = \sum_{n=1}^{\infty} a_n \sin(n\omega t) \quad ; \quad n = 1, 3, 5, \dots \quad (2.10)$$

Maintenant nous allons obtenir une expression pour les coefficients  $a_n$ , premièrement pour un nombre impair des impulsions  $p$  par demi-cycle, en considérant que l'amplitude de la tension est l'unité, alors

$$a_n = \frac{4}{\pi} \left[ \int_{\alpha_1}^{\alpha_2} \sin(n\omega t) d(\omega t) + \int_{\alpha_3}^{\alpha_4} \sin(n\omega t) d(\omega t) + \dots + \int_{\alpha_p}^{\pi/2} \sin(n\omega t) d(\omega t) \right] \quad (2.11)$$

$$a_n = \frac{4}{\pi} \left[ \left( -\frac{\cos(n\omega t)}{n} \right) \Big|_{\alpha_1}^{\alpha_2} + \left( -\frac{\cos(n\omega t)}{n} \right) \Big|_{\alpha_3}^{\alpha_4} + \dots + \left( -\frac{\cos(n\omega t)}{n} \right) \Big|_{\alpha_p}^{\pi/2} \right] \quad (2.12)$$

sachant que  $\cos(n\pi/2) = 0$  pour  $n = \text{impair}$ , on arrive à l'expression de  $a_n$  suivante:

$$a_n = \frac{4}{n\pi} \sum_{k=1}^p (-1)^{k+1} \cos(n\alpha_k) \quad p \text{ et } n \text{ impairs} \quad (2.13)$$

Pour le cas d'un nombre pair des impulsions  $p$ , nous avons:

$$a_n = \frac{4}{\pi} \left[ \int_{\alpha_1}^{\alpha_2} \sin(n\omega t) d(\omega t) + \int_{\alpha_3}^{\alpha_4} \sin(n\omega t) + \dots + \int_{\alpha_p}^{\pi/2} \sin(n\omega t) d(\omega t) \right]$$

Ce qui donne l'expression suivante:

$$a_n = \frac{4}{n\pi} \sum_{k=1}^p (-1)^{k+1} \cos(n\alpha_k) \quad \text{p pair et n impair} \quad (2.14)$$

donc, pour tous p et n impair, les coefficients an sont donnés par,

$$a_n = \frac{4}{n\pi} \sum_{k=1}^p (-1)^{k+1} \cos(n\alpha_k) \quad \text{ou} \quad 0 < \alpha_1 < \alpha_2 < \dots < \alpha_p < \pi/2 \quad (2.15)$$

Pour éliminer les harmoniques désirées, l'expression précédente s'égale à zéro et le système des équations non linéaires résultantes est résolu pour obtenir les angles  $\alpha_1, \alpha_2, \dots, \alpha_p$  qui définent les commutations des interrupteurs, c'est-à-dire,

$$f_i(\alpha) = \frac{4}{n_i\pi} \sum_{k=1}^p (-1)^{k+1} \cos(n_i\alpha_k) = 0 \quad (2.16)$$

Où  $n_i$   $i = 1, 2, \dots, p$ , sont les harmoniques à être éliminées;  $0 < \alpha_1 < \alpha_2, \dots, < \alpha_p < \pi/2$

À titre d'exemple, si nous voulons éliminer les harmoniques 3, 5 et 7, nous devons résoudre le système des équations suivant:

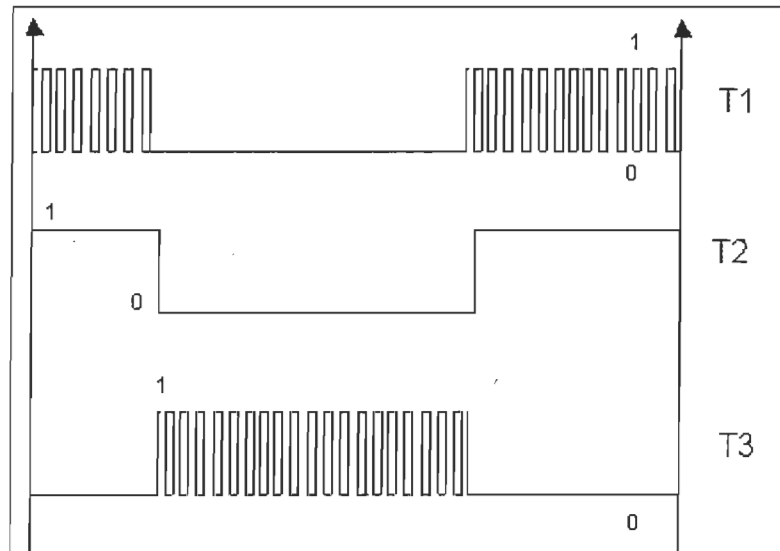
$$\cos 3\alpha_1 - \cos 3\alpha_2 + \cos 3\alpha_3 = 0 \quad (2.17)$$

$$\cos 5\alpha_1 - \cos 5\alpha_2 + \cos 5\alpha_3 = 0 \quad (2.18)$$

$$\cos 7\alpha_1 - \cos 7\alpha_2 + \cos 7\alpha_3 = 0 \quad (2.19)$$

que, en utilisant le logiciel **MABLV**, les angles suivants sont trouvés:

$$\alpha_1 = 22.7^\circ \quad \alpha_2 = 37.8^\circ \quad \alpha_3 = 46.8^\circ$$



**Figure 2.4 Tension de sortie avec les harmoniques 3, 5, 7 éliminées.**

C'est important de remarquer que le nombre des impulsions  $p$  par alternance est égal au nombre des harmoniques à éliminer. D'autre part, l'expression pour  $V_o$  est égale à:

$$V_o(\omega t) = \sum_{n=1}^{\infty} \frac{4}{n\pi} \sum_{k=1}^p (-1)^{k+1} \cos(n\alpha_k) \sin(n\omega t) \quad (2. 20)$$

$$n = 1, 3, 5, \dots$$



## 2.5 Motif de commande des IGBTs pour le convertisseur

La technique de modulation MLI programmée permet d'obtenir les angles de commutation en tenant compte d'un certain critère de performance, par exemple, la minimisation de la distorsion harmonique, la minimisation des pertes, etc.. Le critère dans notre cas est la minimisation de la distorsion harmonique [MoUn95]. Cette minimisation s'obtient en faisant le calcul des angles qui éliminent les harmoniques de rang faible dans la tension de sortie, permettant ainsi d'utiliser un filtre de sortie plus petit pour éliminer les harmoniques de fréquences plus élevées. De cette façon le taux de distorsion harmonique de l'onduleur est plus aisément réduit et moins dispendieux. Pour calculer les angles, il faut résoudre un système des équations transcendantes qui sont données par l'équation (2.16). Cette équation est incluse ci-dessous:

$$f_i(\alpha) = \frac{4}{n_i \pi} \sum_{k=1}^p (-1)^{k+1} \cos(n_i \alpha_k) = 0$$

où  $i = 1, 2, \dots, p$   $n_i$  = harmoniques à être éliminées  
 $p$  = nombre des impulsions par alternance  
 $\alpha_k$  = angles de commutation à déterminer

Les angles de commutation  $\alpha_k$  pour notre onduleur ont été obtenus en tenant compte des contraintes suivantes:

- 1) Symétrie par rapport à  $\pi/2$
- 2) Symétrie par rapport à  $\pi$
- 3)  $0 < \alpha_1 < \alpha_2 < \dots < \alpha_{p-1} < \pi/2$
- 4)  $(\alpha_k - \alpha_{k-1}) > 0.3^\circ$

La dernière contrainte nous assure le temps minimum en conduction ou en repos qu'imposent à l'interrupteur, soit l'IGBT dans notre cas, en considérant ses capacités physiques et sa performance. Elle limite le nombre des impulsions ou le nombre des harmoniques de faible ordre qui peuvent être éliminés dans la tension de sortie  $V_o$  de l'onduleur. Par les contraintes de symétrie, les harmoniques à éliminer sont les impairs 3, 5, 7, ....  $p$ , 'P' est le nombre des impulsions par alternance qui est défini par la validation de la contrainte N°. 4. Donc, le système des équations à résoudre est le suivant [Bo92 ] :

$$\cos 3\alpha_1 - \cos 3\alpha_2 + \cos 3\alpha_3 \dots \dots \dots + \cos 3\alpha_p = 0$$

$$\cos 5\alpha_1 - \cos 5\alpha_2 + \cos 5\alpha_3 \dots \dots \dots + \cos 5\alpha_p = 0$$

$$\cos 7\alpha_1 - \cos 7\alpha_2 + \cos 7\alpha_3 \dots \dots \dots + \cos 7\alpha_p = 0$$

.....

.....

.....

$$\cos(2p-1)\alpha_1 - \cos(2p-1)\alpha_2 + \cos(2p-1)\alpha_3 \dots \dots \dots + \cos(2p-1)\alpha_p = 0$$

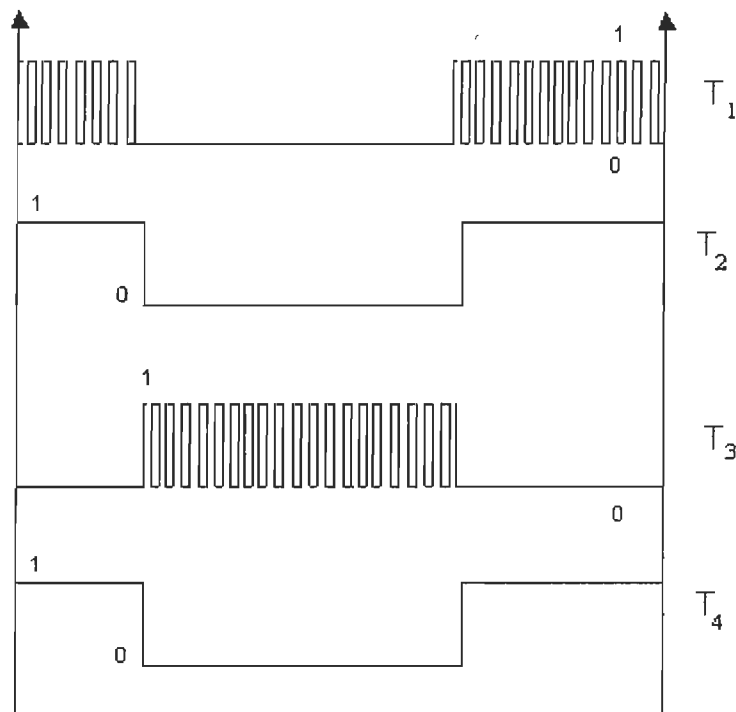
La résolution de ce système d'équations se fait seulement pour les angles de commutation  $\alpha_i$ , d'un quart ( $\pi/2$ ) de période; Par les contraintes de symétrie, il devient simple de trouver les autres à partir de ces angles. À l'aide du logiciel MATLAB, ce système d'équations a été résolu pour différents nombres des angles. Ci- dessous, se présentent les angles pour 25, 21 et les 15 commutations par quart de période:

**Table des angles de commutation (degrés)**

<u>25</u>	<u>21</u>	<u>15</u>
6.016	6.933	9.053
6.704	7.907	10.714
12.032	13.923	18.105
13.465	15.814	21.486
18.105	20.913	27.273
20.168	23.720	32.200
24.122	27.903	36.497
26.872	31.627	42.857
30.195	34.950	45.837
39.477	53.514	55.348
36.326	42.125	64.057
40.279	42.055	65.031
42.456	47.326	74.542
46.983	49.217	74.866
48.644	55.176	69.025
53.686	56.494	70.760
.	.	.
.	.	.
.	.	.

Pour le cas de 25 commutations, la différence des angles la plus petite est égale à:  $(80.271 - 80.214) = 0.057^\circ < 0.3^\circ$ ; pour les 21 commutations elle est égale à  $(78.667 - 78.495) = 0.172 < 0.3^\circ$ , tandis que pour celui des 15 commutations, la différence est égale à  $(74.542 - 74.866) = 0.344^\circ > 0.3^\circ$ . De tous les résultats des essais réalisés, nous avons trouvé que 15 est le nombre de commutations par quart de période le plus grand qui respecte la contrainte N°. 4, donc nous avons décidé d'avoir ce nombre de commutations pour les interrupteurs de notre onduleur.

Connaissant le nombre des commutations et leurs angles, les séquences des commutations des interrupteurs de l'onduleur ont été définies, c'est-à-dire, le motif de commande Soit à la figure[2.5]:



**Figure 2.5 Motif de commande des IGBTs**

Comme il est indiqué dans la figure [2.6], les interrupteurs  $T_1$  et  $T_3$  sont utilisés comme hacheurs, tandis que les interrupteurs  $T_2$  et  $T_4$  sont placés en conduction pendant la demi-période respective. De cette façon, les pertes de commutation sont minimisées le plus possible.

Comme nous l'avons mentionné dans la section 2.4, le motif de commande a ensuite été implanté dans une mémoire morte [MhRa93].

## **2.6 Conclusion**

L'utilisation de la commande à modulation de largeur d'impulsions (MLI) est très avantageuse pour des convertisseurs avec des charges définies fixes, par exemple, les sources de tension sans interruption. Dans ce chapitre, le principe et le développement de plusieurs techniques de la commande (MLI) pour les onduleurs de tension ont été présentées. Parmi ces techniques, la technique de modulation calculée ou programmée qui permettent de sélectionner les harmoniques que l'on désire éliminer par conséquent, ce chapitre a traité en détail les commandes MLI classiques, quant à la commande SVM, elle fera l'objet du chapitre suivant.

# *Chapitre 3*

## ***COMMANDE MODULATION VECTORIELLE SPATIAL (SVM)***

Ce troisième chapitre traite des différents modes de commande de Modulation vectorielle spatial utilisé dans le domaine de puissance. Nous présenterons tout d'abord la fonction modulation vectorielle puis nous détaillerons le mode de modulation vectorielle ainsi que les limites de fonctionnement, les avantages et la qualité d'un signal SVM qu'on évalue en considérant de nombreux critères :

- contenu harmonique des signaux générés
- fréquence de commutation des semi-conducteurs

- domaine des tensions réalisables (en particulier, tension maximale réalisable, fondamental maximal réalisable, effet des temps minimaux de conduction de semi-conducteurs)
- facilité d'implantation
- alimentation équilibrée
- Simulation

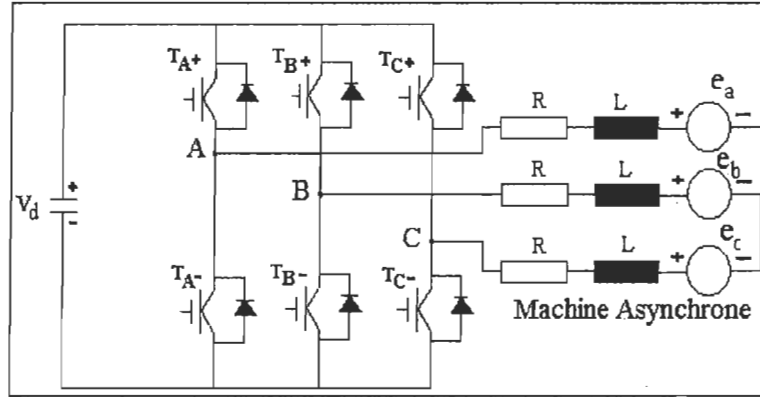
On remarquera la diversité de ces critères. Aussi ne cherchera-t-on pas à trouver une commande SVM « optimale » au sens mathématique du terme, c'est-à-dire optimisant un critère précis. On portera un jugement qualitatif sur l'efficacité d'une SVM vis-à-vis d'un ensemble de critères.

Le contenu harmonique des signaux générés est primordial. Du point de vue pratique, on peut avoir à s'intéresser à certaines fréquences particulières. En effet, dans le domaine de la traction ferroviaire, il existe des fréquences qu'il faut à tout prix éviter. Elles concernent en particulier la signalisation, qu'il ne faut pas perturber au niveau électromagnétique, et la fréquence de résonances mécaniques.

### **3.1 Fonctionnement de la commande de modulation**

#### **3.1.1 L'onduleur vu côté machine**

Avant tout, rappelons la notion de vecteur de tension. La tension obtenue et contrôlée aux bornes de la machine est fournie à partir de la commande des bras de l'onduleur triphasé.



**Figure 3.1 : Onduleur de tension triphasée**

La figure [3.1] représente un onduleur de tension triphasée. A tout système de grandeurs triphasés  $X_a(t)$ ,  $X_b(t)$ ,  $X_c(t)$  comme le courant, la tension le flux, on peut associer un vecteur tournant instantané définie par la relation suivant :

$$\vec{X}_s^p = K_T [x_a + x_b e^{j2\pi/3} + x_c e^{j4\pi/3}] \quad (3.1)$$

avec  $K_T = \sqrt{\frac{2}{3}}$  afin que cette transformation conserve la puissance.

Le vecteur  $\vec{V}_{sn}^p$  associé aux tensions simples délivrées aux bornes de la machine a donc pour expression :

$$\vec{V}_{sn}^p = \sqrt{\frac{2}{3}} (V_{an} + V_{bn} e^{j2\pi/3} + V_{cn} e^{j4\pi/3}) = V_{s\alpha} + jV_{s\beta} \quad (3.2)$$

### 3.1.2 L'onduleur vu côté commande

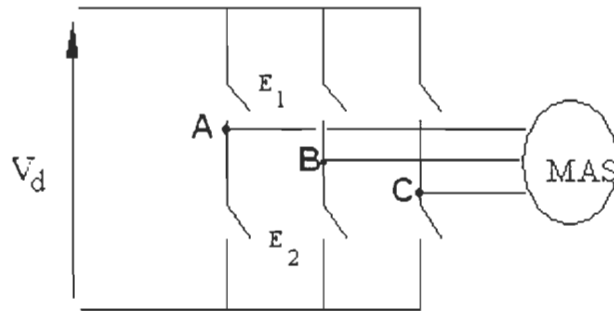
Le vecteur  $\vec{V}_{sn0}^p$  associé aux tensions simples contrôlées est de la forme :

$$\vec{V}_{sn0}^p = \sqrt{\frac{2}{3}} U_0 (S_a + S_b e^{j2\pi/3} + S_c e^{j4\pi/3}) \quad (3.3)$$



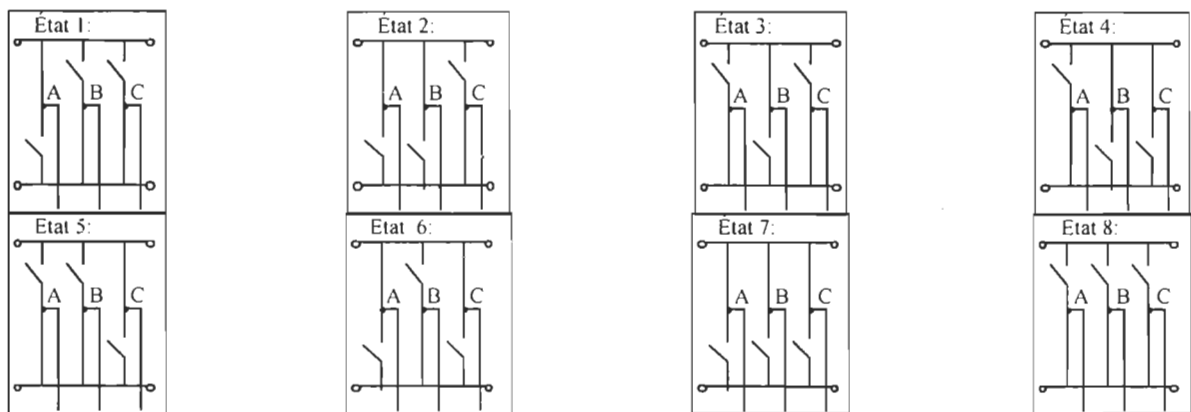
( $S_a, S_b, S_c$ ) représentent l'état des bras a, b, c (figure 3.2).  $S_a, S_b, S_c$  peuvent prendre les valeurs 1 ou 0. La valeur 1 correspond à l'interrupteur  $E_1$  du haut fermé et à l'interrupteur  $E_2$  du bas ouvert.

La valeur 0 correspond à l'interrupteur du haut ouvert et à celui du bas fermé



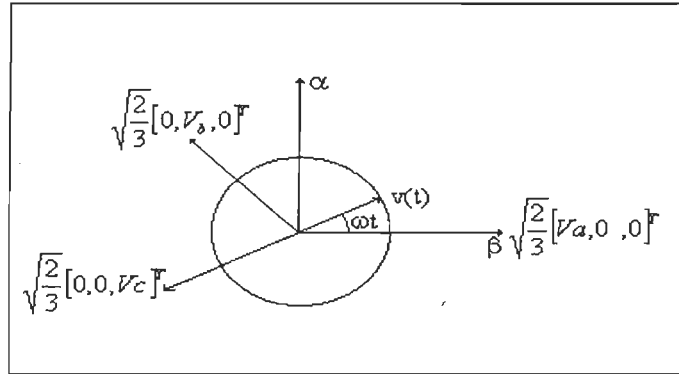
**Figure 3.2 Onduleur de tension triphasée dans l'état (000)**

Il existe 8 combinaisons possibles  $2^3$  pour les bras de l'onduleur figure [3.3]. On peut donc obtenir 8 vecteurs tension différent en fonction de la commande des bras  $S_1, S_2, S_3$ .



**Figure 3.3 Les 8 états de fonctionnement possibles**

La figure [3.4] représente ces différents vecteurs tension. Il existe deux façons d'obtenir un vecteur nul (commande (000) ou (111), les autres vecteurs ont la même amplitude  $U_0 \sqrt{\frac{2}{3}}$  et sont déphasés mutuellement de  $\pi/3$ .



**Figure 3.4 Vecteur tension instantanés**

### 3.2 Définition de la fonction modulation vectorielle

**Définition «moyenne » :**

On dit la fonction SVM (modulation vectorielle spatial ) comme suit :

« à partir d'un vecteur tension de référence appartenant au domaine D figure [3.4], la fonction SVM se charge de gérer les impulsions onduleur de façon à recréer en valeur moyenne sur une période un vecteur tension égale à celui de référence » la figure [3.5]. 5 représente le domaine D des vecteurs tensions moyennes que l'onduleur peut réaliser. Ce domaine est directement déduit des vecteurs instantanés de la figure [3.4]. Il s'agit d'un hexagone formé par les extrémités des vecteurs et dont la distance du centre à un sommet vaut  $U_0 \sqrt{\frac{2}{3}}$ .

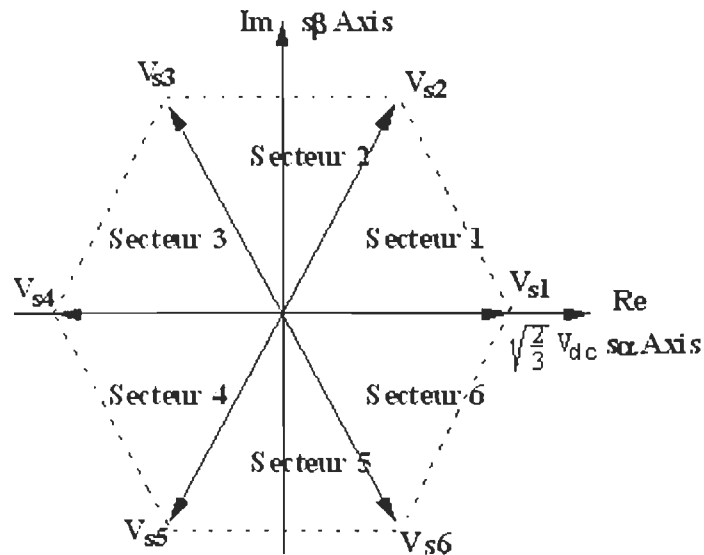


Figure 3.5 Domaine D des vecteurs moyens

#### Définition « fondamentale » :

A partir d'un vecteur tension référence la fonction SVM se charge de gérer les impulsions d'onduleur de façon à recréer en valeur fondamentale (modèle premier harmonique), sur une période, un vecteur tension égale à celui de référence. »

### 3.3 Analyse d'une Modulation vectorielle

Couramment, nous décrivons une SVM par les trois signaux de commande des bras de l'onduleur, mais on peut aussi la décrire de la façon vectorielle on peut voir notre fonction SVM de deux façons :

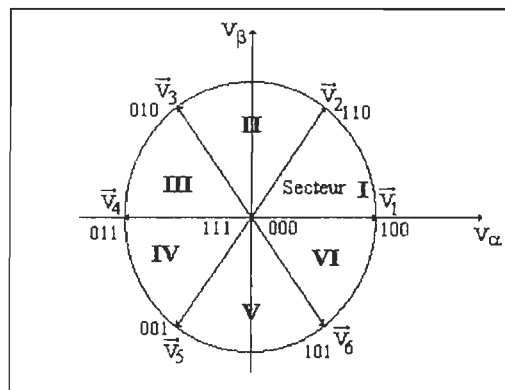
- Il s'agit de gérer les impulsions sur chaque bras de l'onduleur pour recréer en valeur moyenne ou en valeur fondamentale les trois phases neutre de référence.
- Il s'agit de gérer un trajet dans l'hexagone des vecteurs tensions réalisables par l'onduleur (voir figure [3.3]), de façon à recréer en valeur moyenne ou en valeur fondamentale le vecteur tension de référence. On peut alors parler de modulation dans l'espace vectoriel (space vector modulation) le vecteur tension moyen ou fondamentale

correspondant alors au barycentre des points balayés, pondérer par le temps passé sur chacun d'eux.

Ces deux descriptions expriment la même réalité physique, mais vu sous des angles différents.

La première correspond à une vision proche de l'onduleur : on retrouve les signaux de commande de l'onduleur. La seconde description correspond à une vision proche de la machine asynchrone.

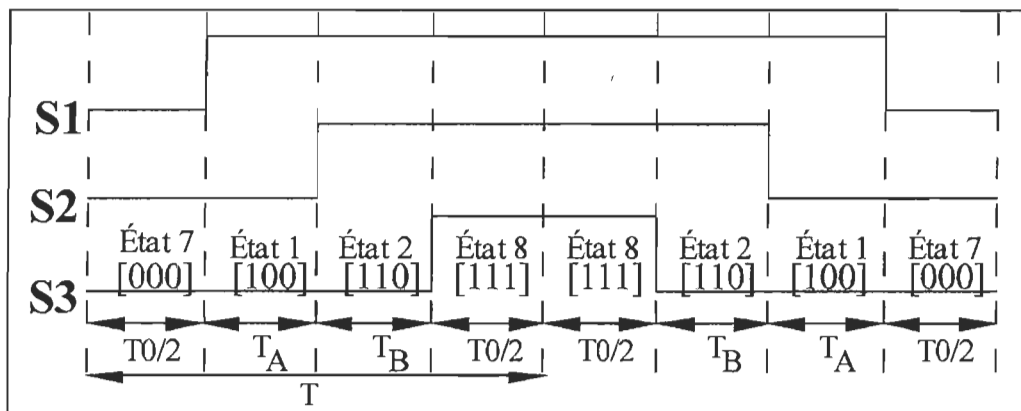
Sur le diagramme de la figure [3.5] les flèches représentent les vecteurs de l'onduleur utilisés pour fabriquer le vecteur tension de référence est obtenu à partir des vecteurs  $V_1$  et  $V_2$  et des vecteurs nuls. Cette SVM utilise des vecteurs tension voisins de celui de référence, donc proche du vecteur tension idéale.



**Figure 3. 6 SVM vectorielle réalisant une commande**

D'autre part, dans la mesure où la fréquence de découpe (fréquence de MLI) est très supérieure au fondamental des tensions en régime permanent, on peut considérer que l'on a presque le même vecteur tension de référence entre deux instants d'échantillonnage successifs. Il est alors assez naturel de chercher à obtenir des signaux les plus symétriques possibles pour réduire les harmoniques.

Grâce à la symétrie des signaux, lorsque la fréquence de la SVM est très supérieure au fondamentale des tensions, cette SVM permet d'avoir accès à la valeur moyenne des courants en échantillonnant ces mesures au milieu de la période (modèle moyen valable). En effet, la valeur instantanée du courant à la moitié d'une période est pratiquement égale à la valeur du courant moyen sur la période d'échantillonnage. Ceci permet de simplifier la mesure des courants : il n'y a aucun filtrage. L'autre avantage vient de l'absence d'un retour pour la commande ; les mesures étant obtenues avant la fin de la période. On dispose d'une demi-période pour faire le calcul du vecteur tension de référence suivant.



**Figure 3.7 Modulation vectorielle utilisée**

La modulation vectorielle retenue (car sa modulation est la plus intéressante pour notre système) figure [3.7] :

- réalise, en valeur moyenne sur une période le vecteur tension de référence
- ses signaux de commande sont symétriques par rapport à la demi-période d'échantillonnage à basse fréquence
- implantation relativement simple

### 3.4 Limitation du domaine des vecteurs tension réalisables

La SVM retenue permet d'attendre n'importe quel vecteur tension de référence appartenant à l'hexagone formé par les vecteurs tension onduleur. Autrement dit, cette SVM couvre le plus grand domaine possible. En particulier, elle peut atteindre le cercle limite inscrit dans l'hexagone si on veut des composantes moyennes à variations sinusoïdales figure [3.7]. Ce cercle limite correspond à une tension phase neutre :

$$\overline{V_s} = \sqrt{\frac{2}{3}} U_0 \cos \frac{\pi}{6} = \frac{U_0}{\sqrt{2}} \quad (3.4)$$

Il faut toutefois prendre en compte les temps minimaux de conduction ( $T_{\min}$ ) des semi-conducteurs. En Modulation vectorielle, la contrainte impose de rester sur le vecteur nul au moins un temps  $T_{\min}$ , et cela 2 fois par période. Des vecteurs de tensions réalisables sont donc un hexagone réduit que l'on déduit de base par une homothétie de facteur  $(1 - \frac{2T_{\min}}{T})$  avec  $T$  période de la MLI. Le cercle limité a alors comme rayon  $(1 - \frac{2T_{\min}}{T}) \frac{U_0}{\sqrt{2}}$ .

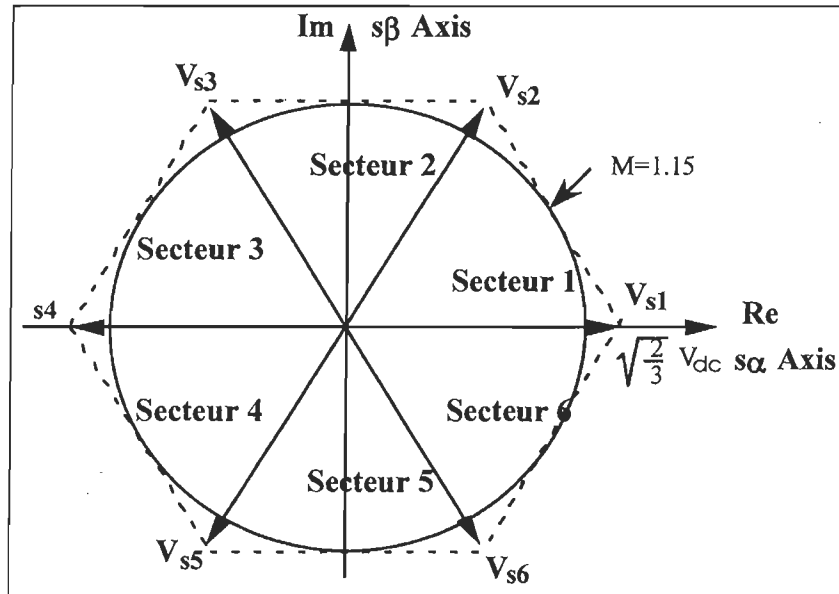


Figure 3. 8 Domaine du vecteur tension avec prise en compte des  $T_{min}$

### 3.5 Différents modes de modulation vectorielle

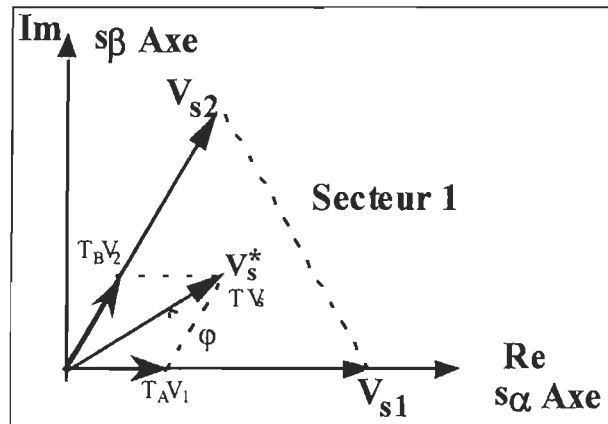
Après avoir introduit la fonction SVM, nous allons maintenant détailler les différents modes utilisés sur toute la plage de fonctionnement du moteur. À basse vitesse, c'est une modulation vectorielle asynchrone (fréquence de SVM asynchrone avec la fréquence des fondamentaux) introduite précédemment qui est utilisée. Elle est basée sur un modèle moyen où la valeur instantanée du courant à la moitié d'une période de SVM est pratiquement égale à la valeur du courant moyen sur la période d'échantillonnage. Lorsque le modèle moyen n'est plus vérifié, on enchaîne sur une modulation vectorielle synchrone (fréquence de modulation vectorielle synchrone avec la fréquence des fondamentaux). L'objectif de cette SVM est de définir un vecteur tension moyen sur la période de modulation vectorielle synchrone qui concorde avec le comportement fondamental de la machine. On réalise une trajectoire de l'extrémité des vecteurs tension circulaire. Le principe de cette SVM synchrone reste le même que la SVM asynchrone. Les signaux sont toujours symétriques, seul le séquençage des bras de l'onduleur est un peu différent. La SVM à angles calculés prend ensuite le relais lorsque les

temps minimums de conduction sont atteints en modulation vectorielle synchrone. L'objectif de la SVM à angles est de contrôler l'amplitude du fondamental de la tension et de rejeter les harmoniques si on utilise plusieurs angles.

Les signaux sont symétriques et les mesures sont toujours prises en milieu de la période afin de se rapprocher le plus possible de la modulation vectorielle.

### 3.5.1 Modulation vectorielle asynchrone

Dans tout ce qui suit, on appelle  $V_1, V_2 \dots V_6$  les vecteurs de l'onduleur figure [3.9].  $V_0$  et  $V_7$  sont les vecteurs nuls.



**Figure 3. 9 Caractéristique d'un vecteur tension moyen**

Nous avons vu qu'en SVM vectorielle asynchrone, un vecteur tension moyen  $\langle V_s \rangle$  fabrique à partir de 2 vecteurs de l'onduleur consécutifs  $V_i$  et  $V_{i+1}$  et de vecteurs nuls  $V_0$  et  $V_7$  figure [3.9]. La période d'échantillonnage est fixe et est égale à la SVM ( $T_{MLI}$ ). Les temps de conduction  $T_i$  et  $T_{i+1}$  correspondent aux temps d'application des vecteurs  $V_i$  et  $V_{i+1}$ , ( $T_{off}/2$ )



correspond aux temps de conduction des vecteurs nuls  $V_0$  et  $V_7$ . Le temps de conduction de  $V_0$  étant égale à celui de  $V_7$ .

Un vecteur tension moyen  $\langle V_s \rangle_{T_c}$  sur une demi - période  $T_c$  de SVM peut s'exprimer en fonction des temps et des vecteurs de l'onduleur qui sont appliqués :

$$\begin{aligned} \langle \bar{V}_s \rangle_{T_c} &= \frac{1}{T_c} \left\{ \int_0^{T_i} \bar{V}_i(t) dt + \int_0^{T_{i+1}} \bar{V}_{i+1}(t) dt + \int_0^{T_{off}/2} \bar{V}_0(t) dt + \int_0^{T_{off}/2} \bar{V}_7(t) dt \right\} \\ \langle \bar{V}_s \rangle_{T_c} &= \frac{T_i}{T_c} \bar{V}_i + \frac{T_{i+1}}{T_c} \bar{V}_{i+1} + \frac{T_{off}/2}{T_c} \bar{V}_0 + \frac{T_{off}/2}{T_c} \bar{V}_7 \end{aligned} \quad (3.5)$$

$$\text{et on doit avoir : } T_c = \frac{T_{MLI}}{2} = T_i + T_{i+1} + T_{off}$$

La figure [3.6] représente la séquence de vecteurs de l'onduleur qu'il faut appliquer pour obtenir un vecteur tension moyen dans chacune des 6 zones de l'hexagone. C'est le cas, en particulier à basse fréquence, ou le nombre de vecteurs moyens par tour est grand, puis ce nombre diminue lorsque la vitesse augmente.

L'objectif de la SVM est d'appliquer les vecteurs de l'onduleur pendant les temps de conduction qu'elle aura préalablement calculés.

Un vecteur de l'onduleur ainsi qu'un vecteur tension moyen peut se caractériser par leur norme et leur argument figure [3.9].

$$\begin{aligned} \bar{V}_i &= \frac{2}{3} U_0 e^{j(i-1)\pi/3} \\ \langle \bar{V}_s \rangle_{T_c} &= V_s e^{j\alpha_s} \end{aligned} \quad (3.6)$$

Du fait de la symétrie, pour simplifier les calculs nous raisonnons sur une demi - période d'échantillonnage  $\left( T_c = \frac{T_{MLI}}{2} \right)$  ; et exprimer le module de la tension, on utilisera la grandeur  $\beta$  en

$$\text{posant } \beta = \frac{V_s}{U_0/2}.$$

Le vecteur tension moyen sur un demi - période d'échantillonnage peut donc s'exprimer en fonction de  $\beta$ .

$$\langle \bar{V}_s \rangle_{T_c} = \beta \frac{U_0}{2} e^{j\alpha_s} \quad (3.7)$$

Avec  $0 \leq \beta \leq \frac{2}{\sqrt{3}}$  si on néglige le temps minimum de conduction.

On exprime alors le vecteur tension moyen en fonction des temps de conduction  $T_i$  et  $T_{i+1}$

$$\begin{aligned} \langle \bar{V}_s \rangle_{T_c} &= \beta \frac{U_0}{2} e^{j\alpha_s} = \frac{T_i}{T_c} \bar{V}_i + \frac{T_{i+1}}{T_c} \bar{V}_{i+1} \\ \langle \bar{V}_s \rangle_{T_c} &= \frac{4}{3} \frac{U_0}{2} \left[ \frac{T_i}{T_c} e^{j(i-1)\pi/3} + \frac{T_{i+1}}{T_c} e^{j(i\pi/3)} \right] \end{aligned} \quad (3.8)$$

$i$  correspond aux régions  $i = 1 \dots 6$ .

On décompose  $\bar{\beta}$  dans le référentiel lié au stator ( $\alpha, \beta$ ) :

$$\bar{\beta} = \beta_\alpha + j\beta_\beta = \beta e^{j\alpha_s} \quad (3.9)$$

En séparant partie réelle et imaginaire de (3. 8) et (3. 9) on calcule le rapport des rapport de temps  $T_i/T_c$  et  $T_{i+1}/T_c$

$$\begin{cases} \frac{T_{i+1}}{T_c} = \frac{\sqrt{3}}{2} \left[ -\beta_\alpha \sin(i-1)\frac{\pi}{3} + \beta_\beta \cos(i-1)\frac{\pi}{3} \right] \\ \frac{T_i}{T_c} = \frac{\sqrt{3}}{2} \left[ \beta_\alpha \sin i\frac{\pi}{3} - \beta_\beta \cos\left(\frac{\pi}{3}i\right) \right] \end{cases} \quad (3.10)$$

Quant au vecteur nul, il est appliqué pendant le temps  $\frac{T_{off}}{2} = \frac{1}{2}(T_c - T_i + T_{i+1})$

Les grandeurs  $\beta_\alpha$  et  $\beta_\beta$  correspondant au vecteur tension de référence.

La méthode que nous venons d'exposer permet de comprendre comment sont calculés les temps de conduction. Cependant, pour l'implantation effective dans le logiciel nous avons procédé d'une façon différente.

### 3.5.2 Modulation vectorielle synchrone :

La modulation vectorielle synchrone est très proche de la Modulation vectorielle asynchrone. Elle est basée sur le même principe avec comme grande différence une fréquence de découpe variable, synchrone avec la commande et la fréquence des fondamentaux des signaux, d'où le terme "synchrone".

En Modulation vectorielle, un vecteur tension moyen est fabriqué à partir d'un seul vecteur de l'onduleur et des vecteurs nuls. Le vecteur moyen résultant est donc proportionnel à un vecteur de l'onduleur figure [3.5] et seulement 6 vecteurs tension sont générés par tour.

Les signaux sont symétriques. Le vecteur moyen peut toujours s'écrire en fonction du temps  $T_i$  et du vecteur de l'onduleur  $V_i$  appliqué pendant ce temps –là.

$$\langle \overline{V_s} \rangle_{T_c} = \frac{T_i}{T_c} \overline{V_i} \quad (3.11)$$

Ici  $T_c$  est variable et s'exprime en régime établi en fonction de la commande  $T_s/6$ .

$$T_c = \frac{1}{2} \left( \frac{T_s}{6} \right) \quad (3.12)$$

avec 
$$T_c = \frac{T_s}{12} = T_i + T_{off}$$

La figure [3.6] montre la séquence des vecteurs de l'onduleur pour obtenir un vecteur tension moyen. Un vecteur de l'onduleur  $V_i$  est appliqué sur  $\frac{1}{2}$  période d'échantillonnage  $T_c$  pendant un temps  $T_i$  et les vecteurs nuls  $V_0$  et  $V_7$  sont appliqués pendant un temps identique égal à  $T_{off}/2$ .

On peut remarquer que dans ce cas là, deux bras de l'onduleur commutent en même temps au cours de la fréquence d'échantillonnage.

Pour calculer les temps  $T_i$  et  $T_{off}$  de conduction on procède comme précédemment en utilisant une transformation vectorielle utilisant les amplitudes :

$$\bar{V}_i = \frac{2}{3} U_0 e^{j(i-1)\pi/3} \quad i=1, \dots, 6 \quad (3.13)$$

Vecteur moyen :

$$\langle \bar{V}_s \rangle_{T_c} = V_s e^{j(i-1)\pi/3} \quad (\text{proportionnel à un vecteur de l'onduleur}) \quad (3.14)$$

$$\langle \bar{V}_s \rangle_{T_c} = \beta \frac{U_0}{2} e^{j(i-1)\pi/3} = \frac{T_i}{T_c} \bar{V}_i \quad (3.15)$$

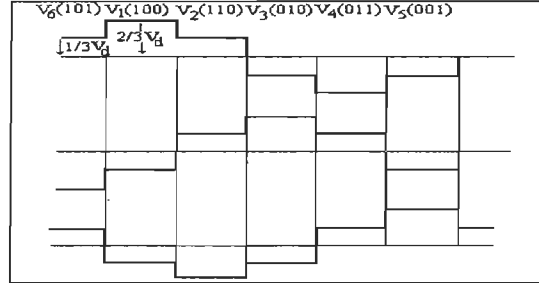
On obtient alors le temps de conduction  $T_i$ :

$$\frac{T_i}{T_c} = \frac{3}{4} \beta \quad \text{avec} \quad \beta = \sqrt{\beta_\alpha^2 + \beta_\beta^2} \quad (\alpha, \beta) \text{ référentiel} \quad (3.16)$$

Les vecteurs nuls sont appliqués pendant le temps :  $T_{off} = T_c - T_i$ .

### 3.5.3 Relation entre ``moyenne`` et ``fondamental`` :

Nous avons vu qu'en SVM vectorielle synchrone c'est le vecteur moyen qui va fixer le comportement fondamental de la machine. Établissons la relation entre moyenne et fondamental.



**Figure 3. 10 Allure de l'onde de tension**

En décomposant l'onde de tension figure [3.12] en série de Fourier, on définit l'amplitude du fondamental et des harmoniques.

$$(V_{jn0})_f = \frac{4}{\Pi} \frac{U_0}{2} \sin\left(\frac{5\Pi}{12}\right) \sin\left(\frac{5\Pi}{12}\right) \sin\left(\frac{\Pi}{6} \frac{T_i}{2T_c}\right) \quad (3.17)$$

Amplitude des harmoniques :  $k=2n+1$ ,  $\alpha = \frac{\Pi}{6} \left( \frac{1}{2} - \frac{T_i}{2T_c} \right)$

$$(V_{jn0})_k = \frac{8}{\Pi} \frac{1}{k} \frac{U_0}{2} \cos\left(\frac{3\Pi}{4}\right) \left[ \cos k \left( \alpha - \frac{3\Pi}{4} \right) + 2 \sin\left(k \frac{5\Pi}{12}\right) \sin k\alpha \right] \quad (3.18)$$

Le vecteur des fondamentaux au milieu des périodes de SVM a pour expression :

$$(\bar{V}_s)_f = \sqrt{\frac{2}{3}} \left[ V_{af}(t) + V_{bf}(t) e^{j\frac{2\Pi}{3}} + V_{cf}(t) e^{j\frac{4\Pi}{3}} \right]$$

$$\text{d'où} \quad (\bar{V}_s)_f = \frac{4}{\Pi} \sqrt{\frac{2}{3}} \sin\left(\frac{5\Pi}{12}\right) \sin\left(\frac{\Pi}{6} \frac{T_i}{2T_c}\right) \frac{U_0}{2} e^{j(i-1)\frac{\Pi}{3}} \quad (3.19)$$

Le vecteur de l'onduleur a pour expression :

$$\bar{V}_1 = \sqrt{\frac{2}{3}} U_0 e^{j(i-1)\frac{\Pi}{3}} \quad (3.20)$$

Donc à cet instant, le vecteur des fondamentaux est en phase avec un vecteur de l'onduleur.

En modulation vectorielle l'amplitude maximale du fondamental de la tension est obtenue pour :

$$T_i = T_c \quad \text{on a} \quad (V_{jn0})_{f \max} = \frac{4}{\Pi} \frac{U_0}{2} \quad (3.21)$$

En fait il faut tenir compte des temps minimums de conduction :

$$(V_{jn0})_{f \max} = \frac{4}{\Pi} \frac{U_0}{2} \sin\left(\frac{5\Pi}{12}\right) \sin\left(\frac{\Pi}{12} - \frac{\Pi}{12} \frac{T_{\min}}{T_c}\right) \quad (3.22)$$

### 3.5.4 Modulation vectorielle a angles calculés ( 1 angle )

La modulation vectorielle à angles calculés ``1 angle `` permet d'obtenir une amplitude du fondamental de la tension plus grande qu'en Modulation vectorielle et permet ainsi d'atteindre la pleine onde.

Cependant, elle ajoute des harmoniques. Le principe de cette SVM est de régler l'amplitude du fondamental de la tension à l'aide d'un angle calculé  $\theta_1$ . En utilisant un angle unique, seul le fondamental de la tension peut être contrôlé. Comme dans le cas de la SVM

synchrone, la fréquence de découpage de l'onduleur correspond à la fréquence de la commande. On aura donc une période de SVM  $T_s/6$  variable.

### **3.6 Conclusion**

Dans ce chapitre j'ai présenté la technique de modulation vectorielle spatial avec plus de détail, vu que celle-ci est la technique choisie pour commander notre onduleur triphasé. En utilisant cette technique les angles de commutation des interrupteurs ont été obtenus et le motif de commande de l'onduleur triphasé a été défini. Les systèmes numériques de l'information opèrent sur des nombres et font appel à un ordinateur. Les signaux doivent subir une conversion analogique-numérique (A-N) pour obtenir les résultats numériques avant de l'implanter puisque notre commande analogique, nous allons donc présenter dans le dernier chapitre les solutions de mise en œuvre de notre méthodologie de conversion et les différentes étapes d'intégration de notre commande en utilisant différents outils de modélisation, de simulation et de synthèse et à la fin les résultats obtenus.

# *Chapitre 4*

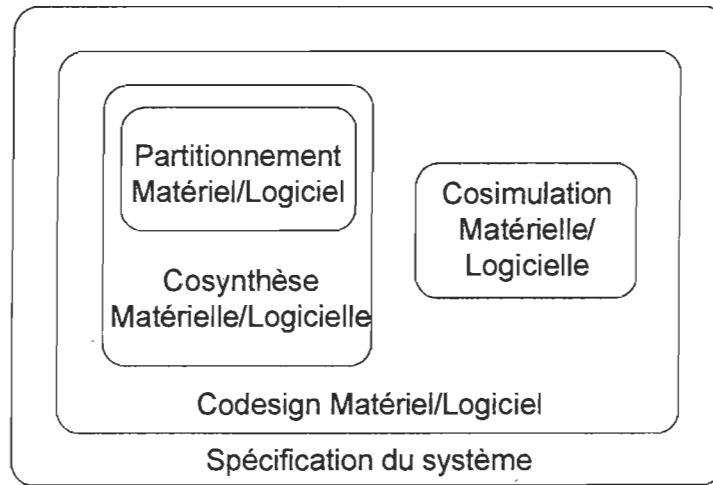
## ***SIMULATIONS ET RÉSULTATS D'IMPLANTATION EN TECHNOLOGIE VLSI***

### ***Introduction***

Nous allons pour débiter présenter d'abord une méthodologie d'implantation dans son ensemble. Puis nous allons traiter plus en détails les parties qui nous intéressent. En particulier dans le développement de notre méthodologie pour le partitionnement matériel/logiciel.

Chacune des activités de conception doit être faite en fonction du partitionnement logiciel et matériel. Que ce soit dans les spécifications du système, ou dans les simulations, tous les composants sont utilisés. Vous trouverez à la figure [4.1] les activités de conception logiciel/matériel [AdTh96].





**Figure 4.1 Activités de conception matérielle/logicielle**

Voici une brève description de ces différentes activités.

- **Spécification du système:** Une description du comportement ainsi que des contraintes (temps d'exécution, surface, dissipation de puissance, etc...) pour le système hétérogène au complet sont données. Cela peut être fait avec un programme dans un langage quelconque (C, C++, ADA, Java [YMS+98]), des ensembles d'instructions, des graphes de flots de données ou des langages spécifiques de description matérielle (VHDL, HardwareC, Lustre, Esterel, Verilog, Sillage) [GaRa94].

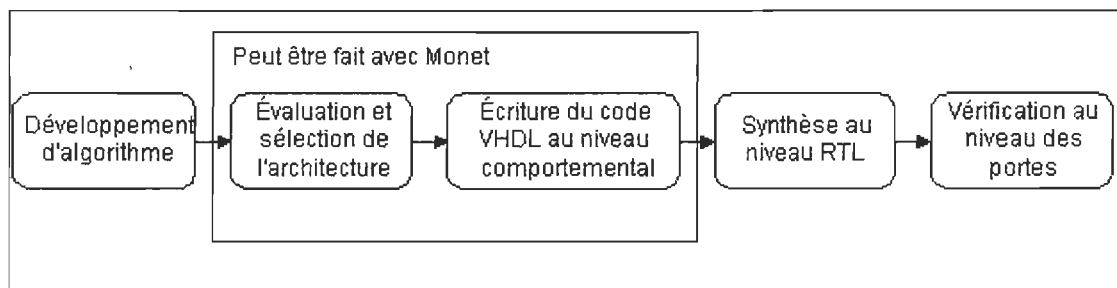
Les gens ont souvent tendance à choisir un langage de programmation couramment utilisé, car il a l'avantage d'être déjà connu [GuLi97]. Cela rend également possible la réutilisation de programmes déjà écrits. De plus, il est intéressant d'avoir une description qui peut être exécutable, car cela permet de vérifier si la fonctionnalité du système est bonne [DRG97].

## 4.1 Description des outils utilisés

Nous avons utilisé principalement trois outils commerciaux dans notre méthodologie. En voici une introduction, ainsi que leurs inconvénients et leurs avantages.

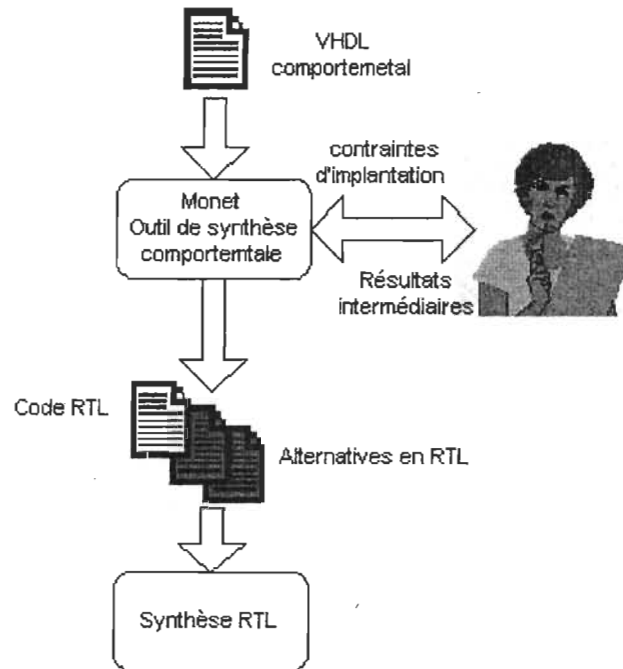
### 4.1.1 Monet de Mentor Graphics

Monet est un outil de synthèse comportementale. Il permet d'explorer rapidement différentes alternatives architecturales au niveau comportemental et génère automatiquement du code synthétisable au niveau RTL. Cela permet de réduire d'une façon importante le temps de conception et d'obtenir de bons résultats. En fait, Monet combine une exploration interactive et visuelle d'une description comportementale. À l'intérieur d'une méthodologie de conception pour le matériel comme celle de la figure [4.2], Monet serait utilisé pour valider et sélectionner une architecture, ainsi que pour la transformation du VHDL au niveau RTL.



**Figure 4.2 Méthode de conception avec Monet**

Monet interagit avec le concepteur pour obtenir les contraintes et certaines directions pour la synthèse comportementale. La figure [4.2] représente les interactions de Monet, ses entrées et ses sorties.



**Figure 4. 3 Synthèse avec Monet**

Les principaux avantages de Monet :

- Diminue le nombre d'implantation au niveau RTL. Le niveau comportemental permet d'explorer différentes architectures avant de choisir celle qui sera implantée au niveau RTL.
- Diminue la quantité de code à écrire en générant automatiquement du code RTL.
- Offre un interface simple et accessible permettant une meilleur compréhension des différentes étapes d'une synthèse au niveau comportementale.
- Permet une visualisation de l'ordonnancement en utilisant des diagrammes de Kantt, comme il est possible de le voir à la figure [4.4].
- Permet d'obtenir des diagrammes de transitions d'états et des diagrammes de bloc des chemins de données.
- Permet de donner un poids aux boucles, influençant ainsi le degré d'optimisation des boucles.
- Offre une plus grande liberté dans le déroulage des boucles, en permettant entre autre de fixer le nombre de déroulement des boucles.

- Permet de synthétiser des tableaux à plus d'une dimension. Les tableaux contenant plusieurs dimensions sont brisés en plusieurs tableaux.
- Permet de synthétiser des blocs plus volumineux que les autres outils de synthèse comportementale.
- Permet une plus grande flexibilité dans le code que les autres outils de synthèse.
- Offre une synthèse plus rapide que les autres outils de synthèse au niveau comportemental.

Les principaux désavantages de Monet :

- Ne peut estimer que le temps et la surface du matériel. Aucune estimation de la dissipation de puissance, qui est un élément qui devient de plus en plus important dans les systèmes.
- Ne réussit pas toujours à faire concorder une instruction VHDL et un opérateur provenant de ses bibliothèques. Il faut parfois modifier sans nécessairement de logique, le code VHDL comportemental pour réussir à synthétiser.

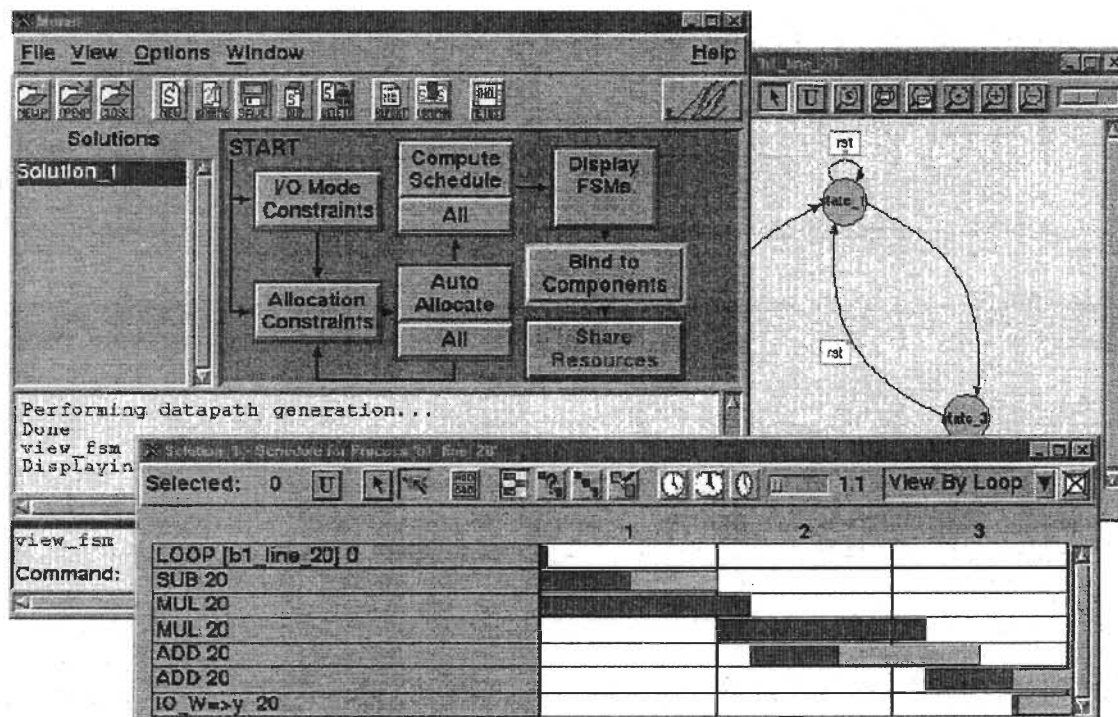


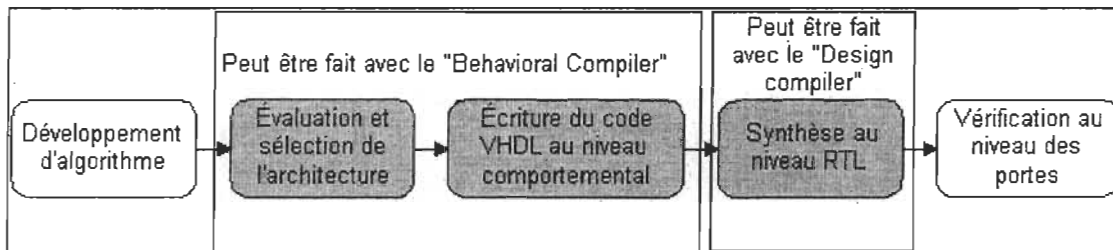
Figure 4. 4 Représentation des diagrammes de Kantt et d'états dans Monet

- Même s’il est plus rapide que les autres outils de synthèse, la synthèse d’un bloc contenant plusieurs boucles déroulées peut devenir fastidieuse. Par exemple la synthèse de deux boucles imbriquées de 16 itérations chacune prend plus d’une journée afin de passer à travers toutes les étapes de la synthèse comportementale.
- Permet uniquement de pipeliner les boucles et non les composants
- Les appels de fonction ont fait en ligne (online) uniquement. À chaque appel de fonction, le code de la fonction est répété à chaque fois, ce qui augmente la densité du code.
- Le code VHDL utilisé par Monet doit être compilé à l’extérieur de l’outil par une commande indépendante.
- Les rapports en format texte sont moins élaborés que dans Synopsys.
- L’utilisateur doit ajouter un reset pour sortir d’un pipeline. Cela ne se fait pas automatiquement.

#### **4.1.2 Description des outils de Synopsys**

Nous avons utilisé deux outils de Synopsys qui utilisent une interface commune. Ces outils sont le « Behavioral Compiler » pour la synthèse au niveau comportemental et le « Design Compiler » pour la synthèse au niveau RTL. La figure [4.5] montre les moments de la conception où ces outils sont utilisés. Si nous reprenons la figure [4.2] que nous avons vu précédemment avec Monet, il est possible de voir où se situent ces deux outils de synthèse.

Le « Behavioral Compiler » a des fonctionnalités similaires à Monet. Les deux ont des entrées et sorties similaires, mais elles sont présentées un peu différemment. Nous allons donc donner les avantages et les inconvénients de ces outils.



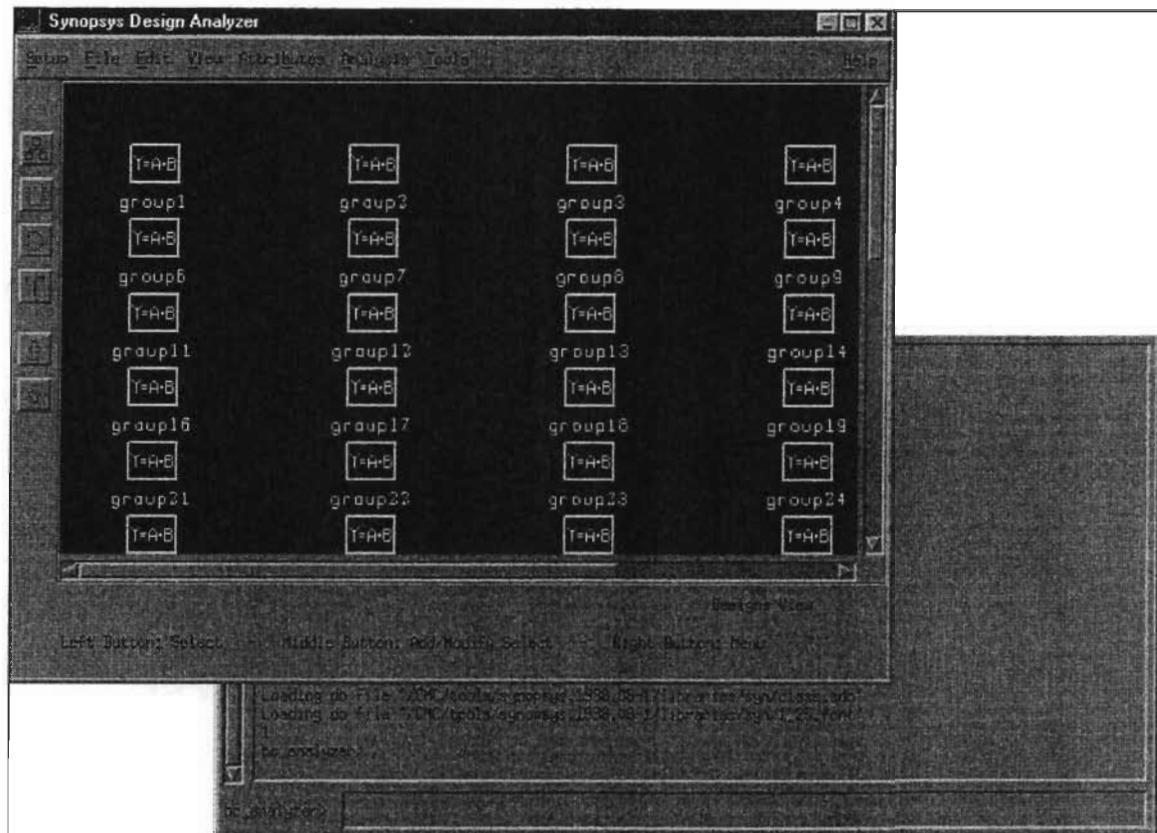
**Figure 4. 5 Étapes de conception avec les outils de Synopsys**

Les avantages des outils de Synopsys sont :

- Diminue le nombre d'implantation au niveau RTL. Le niveau comportemental permet d'explorer différentes architectures avant de choisir celle qui sera implantée au niveau RTL.
- Diminue la quantité de code à écrire en générant automatiquement du code RTL.
- Une fois le code VHDL au niveau RTL obtenu, il est possible de faire une synthèse et d'obtenir la puissance utilisée.
- Fait bien concorder les instructions VHDL et les opérations provenant des librairies. Il trouve toujours un composant pouvant être utilisé d'une façon pertinente.
- Permet de pipeliner des boucles et des composants.
- Les appels de fonction peuvent ne pas être en ligne (online), c'est-à-dire qu'un pragma est utilisé pour indiquer à l'outil que la fonction peut être partagée.
- Permet de pipeliner les boucles et les composants.
- Le code VHDL est compilé à l'intérieur de l'outil. Aucun besoin pour un outil indépendant de compilation.
- Les rapports en format texte sont plus élaborés que dans Monet.
- L'utilisateur n'a pas besoin d'ajouter un *reset* pour sortir d'un pipeline. Cela se fait automatiquement.

## Les inconvénients des outils de Synopsys

- Il n’offre pas une interface simple et accessible. Il est difficile de savoir dans quel ordre les commandes doivent être faites. La figure [4.6] montre un exemple de l’interface. Cela permet de voir qu’il est beaucoup moins amical que l’interface de Monet, vu à la figure [4.3].
- Les résultats ne sont pas offerts sous forme graphique, mais uniquement textuelle. (Selon les informations de leur site internet « [www.synopsys.com](http://www.synopsys.com) », la dernière version du « Behavioral Compiler » le permettrait).
- Synthèse comportementale plus lente que dans Monet [DeMi94].
- Il est incapable de synthétiser du code comportant plusieurs boucles. Par exemple, deux boucles imbriquées et déroulées comportant 16 itérations serait trop volumineux pour une synthèse dans le « Behavioral Compiler ».
- Les contraintes demandées pour l’écriture du code sont plus contraignantes que dans Monet. Par exemple les énoncés d’attente (Wait Statements) doivent être placés d’une manière plus rigoureuse, sinon l’outil est incapable de synthétiser.
- Il ne donne pas de poids aux boucles pour influencer leur degré d’optimisation lors de la synthèse.
- Il n’offre pas une aussi grande liberté que Monet pour le déroulement des boucles.
- Il ne permet pas de synthétiser des tableaux à plus d’une dimension.



**Figure 4. 6 L'interface du « Behavioral Compiler » de Synopsys**

### 4.1.3 Seamless de Mentor Graphics

Un des plus grands problèmes dans la conception de systèmes embarqués contenant du logiciel et du matériel est l'intégration de ces deux parties pour former un seul et même système. Dans les méthodologies de codesign matériel/logiciel traditionnel, le logiciel et le matériel sont développés séparément. Le logiciel ne peut être testé, tant que le matériel n'est pas validé et fonctionnel. Ce qui peut entraîner des erreurs complexes et longues à isoler et à réparer et cela principalement au niveau des interfaces. La cosimulation peut être utilisée à toutes les étapes de la conception, c'est-à-dire à différents niveaux d'abstractions [GaVa95]. Car, il permet de tester le logiciel sur un modèle matériel, même si celui-ci n'est qu'au niveau comportemental ou RTL.



Il est possible de faire de la cosimulation matérielle/logicielle en utilisant uniquement un simulateur matériel avec un modèle fonctionnel du processeur. Par contre, les vitesses de simulation sont très faibles et il est difficile de déterminer le logiciel. On pourrait également créer un modèle en C ou en C++ du processeur et le simuler sur le simulateur de jeux d'instructions, mais la précision de ces modèles est discutable [MaPe98].

Comme il est possible de le voir dans la figure [4.7], Seamless CVE est composé d'un simulateur de jeu d'instructions qui permettent d'exécuter et de déterminer la partie logicielle du système. Puis, un simulateur logique qui permet d'exécuter et de déterminer la partie matérielle du système. Ces deux simulateurs sont reliés et communiquent ensemble dans Seamless. Les mémoires utilisées sont également définies et configurées à l'intérieur de Seamless CVE, car elles doivent être accessibles autant à partir du simulateur logiciel, que du simulateur logique. En fait, à chaque fois que le processeur a besoin d'exécuter une instruction ou d'accéder à la mémoire, le simulateur de jeu d'instructions envoie une requête par l'intermédiaire de Seamless CVE vers le simulateur HDL. Le modèle d'interface de bus répond en exécutant les cycles de bus correspondants, l'accès à la mémoire s'effectue à travers le modèle matériel, puis les données ou instructions sont retournées au simulateur de jeu d'instructions.

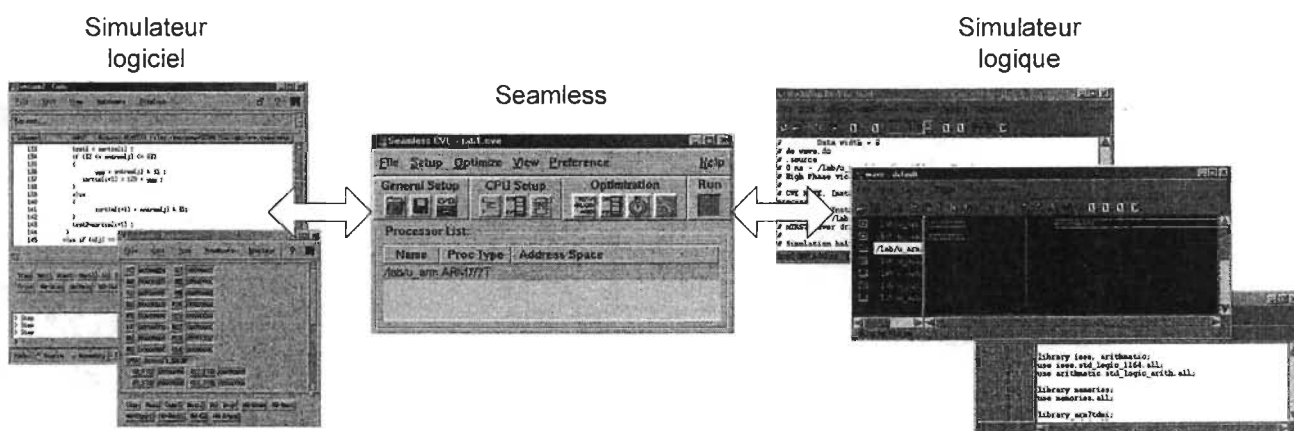


Figure 4-7 Seamless CVE pour la cosimulation matérielle/logicielle

Voici les principaux avantages que nous avons trouvés pour l'utilisation de l'outil Seamless CVE pour co-simuler le matériel et le logiciel à un haut niveau d'abstraction :

- Il réduit le temps de développement des systèmes embarqués.
- Il réduit le nombre d'itérations du prototype matériel.
- Il permet d'exécuter du vrai logiciel sur le design matériel avant que celui-ci soit implanté en silicium.
- Il accélère le déverminage des pilotes des périphériques et des diagnostics matériels.
- Il ne nécessite pas de modification du système du côté matériel ou logiciel.
- Des optimisations permettent d'atteindre de hautes performances pour la validation du système (temps de simulation).
- Il permet d'offrir un produit de qualité plus rapidement sur le marché.
- Il permet de simuler 10 000 fois plus rapidement qu'avec un simulateur logique.
- Il supporte plus de 35 processeurs et 10 simulateurs logiques.
- Utilisation des simulateurs de façon indépendante. Donc, en apprenant à utiliser Seamless, on apprend également une méthodologie pour simuler et déverminer à l'aide d'un simulateur de jeux d'instructions et d'un simulateur logique.

Voici les inconvénients de Seamless :

- Il possède un environnement complexe, utilisant deux simulateurs différents et une grande quantité de fenêtre.
- Difficulté de synchronisation entre les différents simulateurs. Il est difficile d'en arrêter un et de le repartir sans le désynchroniser avec l'autre.
- Il ne possède pas de point de contrôle ou de redémarrage. Après chaque modification, il faut revenir au point de départ et recharger le projet.
- Obligation d'utiliser les modèles de processeurs et de mémoire fournis par Seamless. Impossibilité d'en créer de nouveaux.
- Le changement de processeur demande également un changement de la logique (glue logic) entourant le processeur. Donc, l'interface doit être refaite pour chaque processeur utilisé.

## **4-2 Simulation de la commande**

Nous avons procédé à un certain nombre de simulations. Celles-ci portent sur les blocs essentiels de la commande et de l'architecture proposée pour l'implantation[BoZi 88]. Les simulations ont été choisies à cause de leur implication sur les performances de vitesse, précision binaires, dépassement de capacité.

### **4. 2. 1 La validation fonctionnelle**

La validation fonctionnelle doit nous apporter la preuve que le circuit respecte un certain nombre de propriétés critiques. Ces propriétés peuvent être spécifiques à un certain nombre de propriétés critiques de l'architecture qu'on doit implanter.

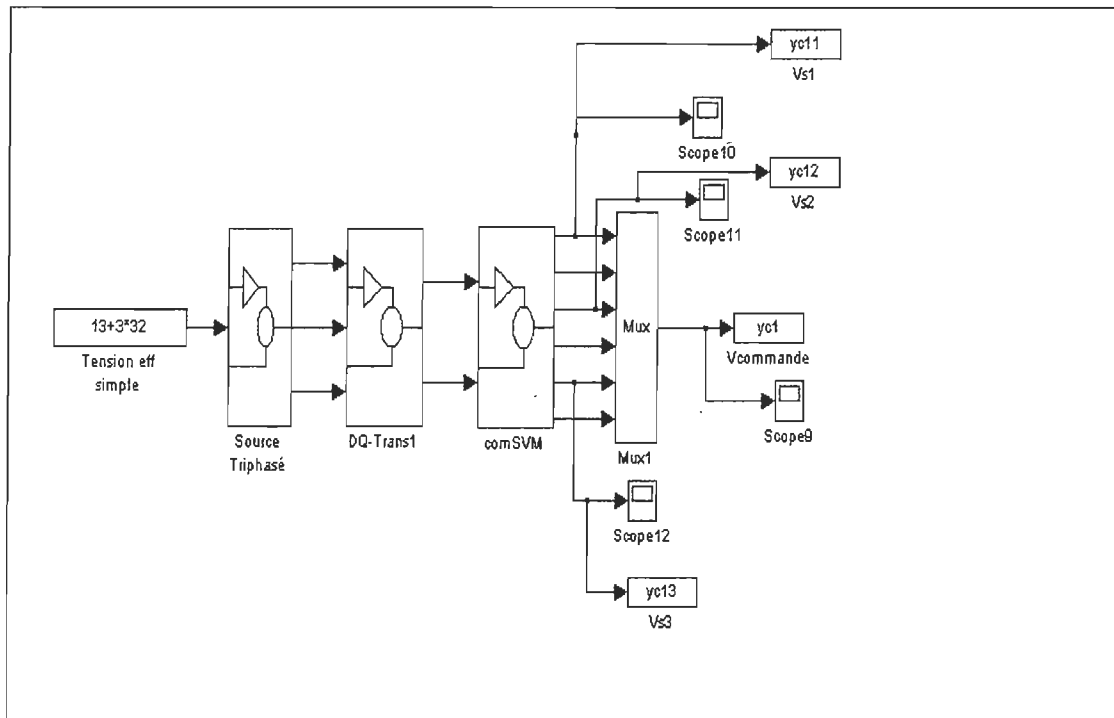
Remarquons que l'architecture proposée permet uniquement de dire si une propriété est vraie ou non pour un circuit donné. La difficulté réside dans le fait de définir des propriétés cohérentes.

Dans notre cas, nous voulons valider deux types de propriétés :

- Les propriétés logiques sur le fonctionnement d'architecture (simulation, synthèse, vérification de fonctionnement logique....).
- Les propriétés de l'implantation (temps, nœud, dépassement de capacité, précision binaire).

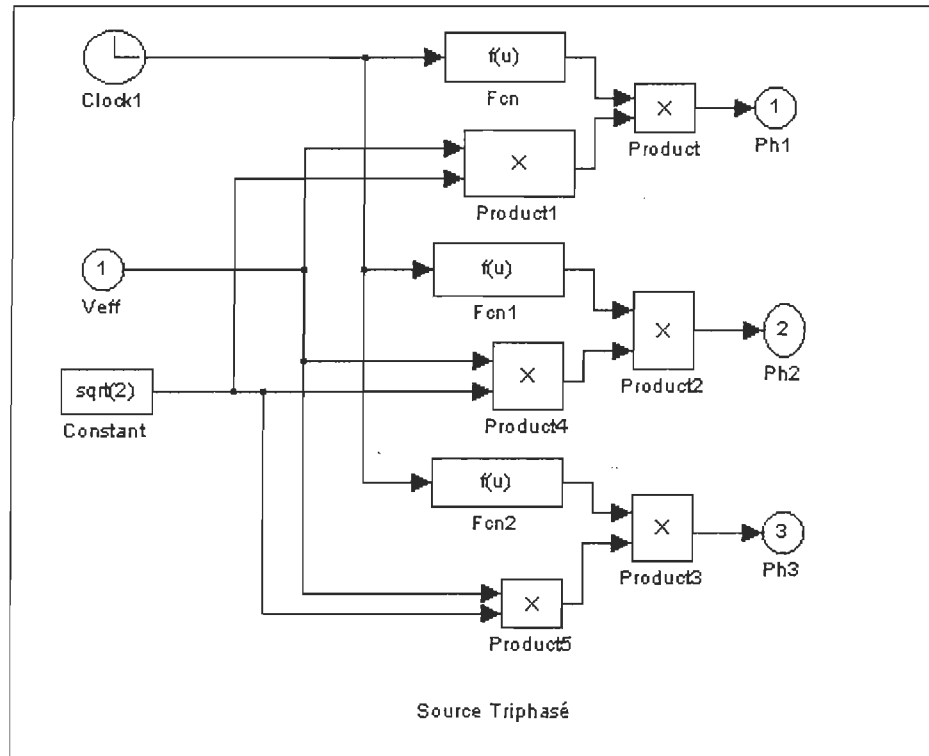
### **4. 2. 2 Système de commande SVM**

Le modèle de la commande est un modèle analogique, basé sur la résolution d'équations qu'on a vu précédemment (chapitre 3) et qui comporte plusieurs blocs fonctionnels qu'on a modélisé sous SIMULINK figure [4.8].



**Figure 4.8 Schéma fonctionnel**

Le schéma présente les blocs fonctionnels du système utilisé pour les simulations. Il est composé de cinq blocs : source de tension courant-continue  $V_E$ , source triphasé figure [4.9], transformé DQ figure [4.10], commande modulation vectorielle spatial et multiplexage figure[4.11].



**Figure 4.9 Schéma source triphasé**

- Le bloc de la transformation DQ effectue cette transformation et calcule l'amplitude et la phase du signal.

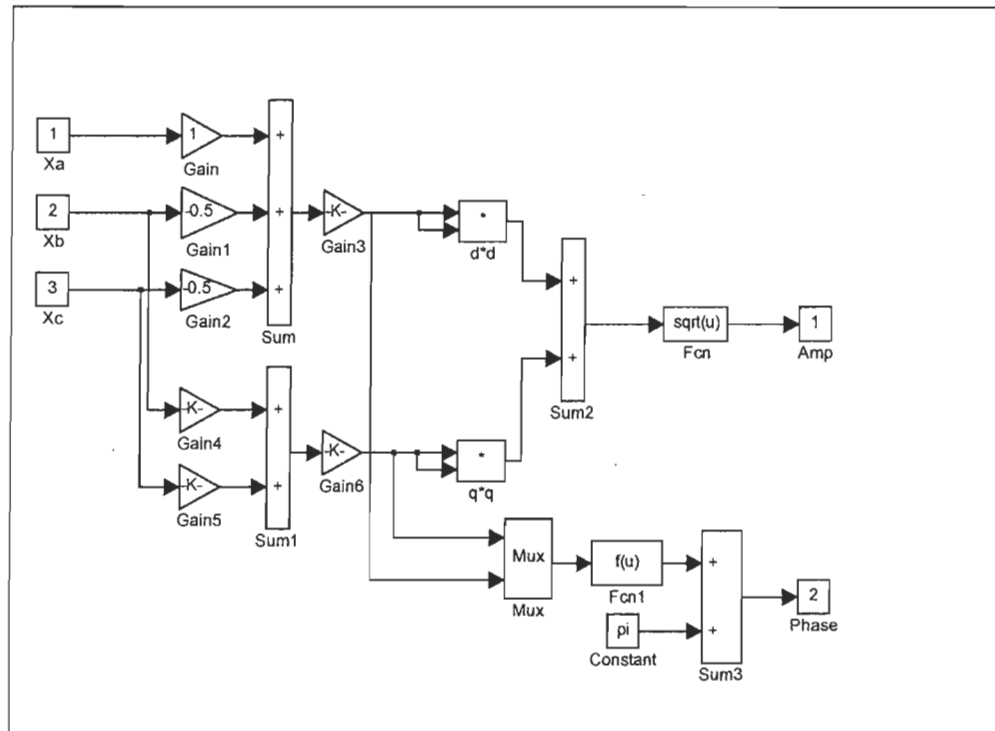
$$\begin{bmatrix} ed \\ eq \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} ea \\ eb \\ ec \end{bmatrix}$$

Le module du vecteur spatial est égal :  $\sqrt{eq^2 + ed^2}$

La phase est égale :  $\text{atan2}(u1, u2) + \pi$

La fonction MATLAB nous donne l'angle entre  $-\pi$  et  $+\pi$ .

La phase sera entre 0 et  $2\pi$ .



**Figure 4. 10 Transformation DQ et calcul de l'amplitude et la phase**

À la sortie on obtient l'amplitude et la phase qu'on va utiliser pour le calcul des temps et la position du vecteur [Zi80].

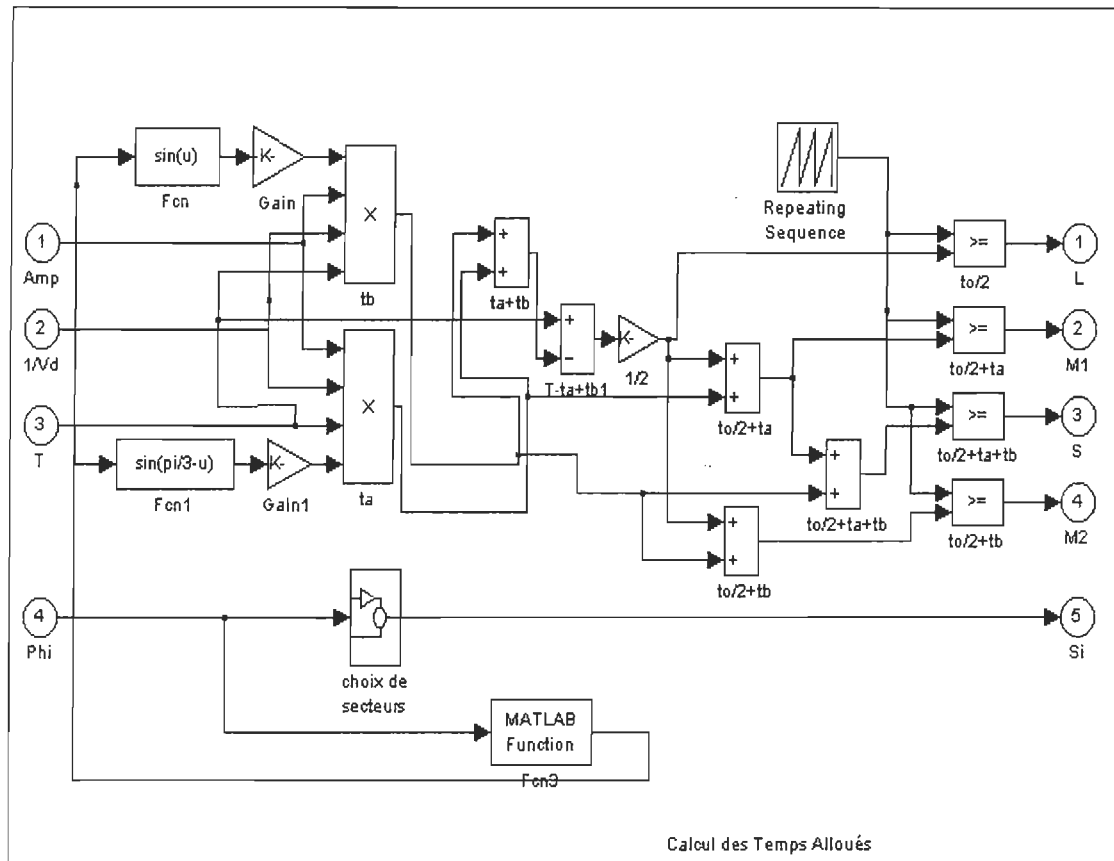
- Calcul des temps et la position du vecteur.

Pour le calcul des temps les paramètres d'entrées sont :

- 1) Amplitude
- 2) Phase

En faisant les calculs des temps  $t_a$  et  $t_b$ , on déduit les temps  $t_0/2$ ,  $t_0/2+t_a$ ,  $t_0/2+t_b$ ,  $t_0/2+t_a+t_b$ . En comparant ces temps à un signal en dents de scie on obtient les quatre signaux équivalents à ces temps figure [4.11].

Pour que l'angle reste toujours compris entre 0 et 60 degré, on utilise la fonction MATLAB  $\text{rem}(u, \pi/3)$ .



**Figure 4.11 Calcul de temps et les positions de vecteur de tension**

Pour préciser la position du vecteur

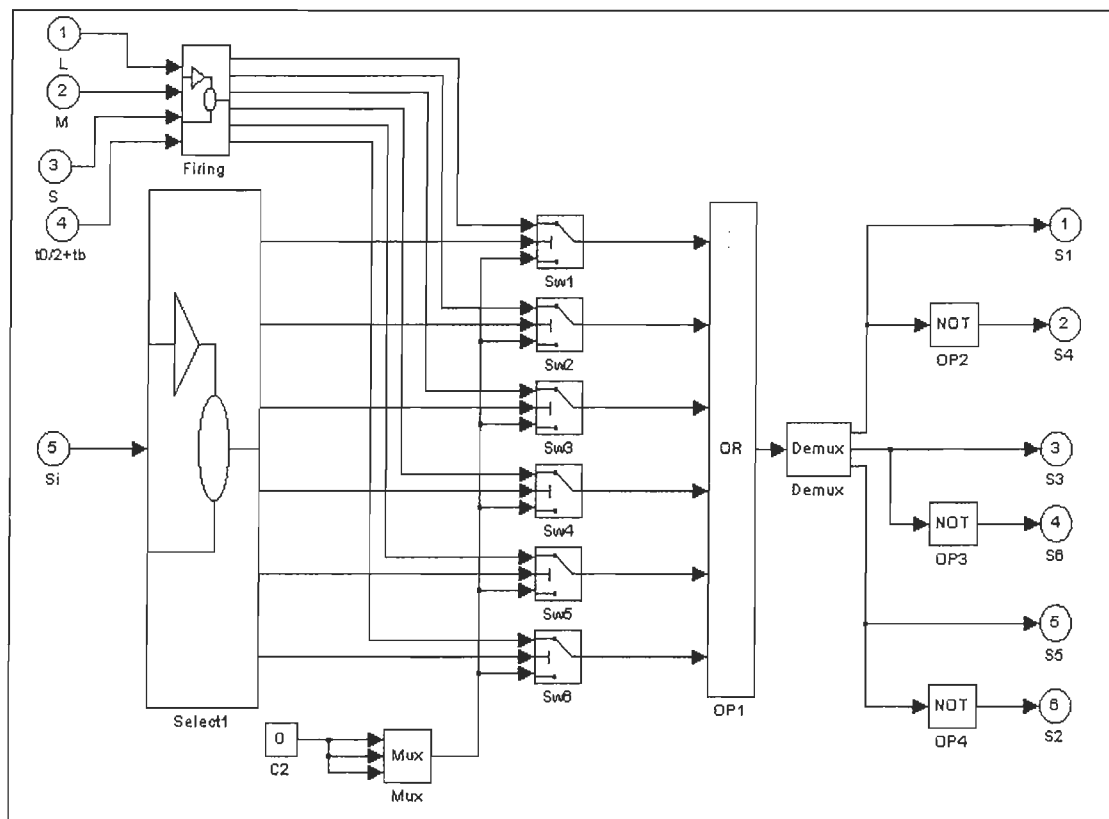
En utilisant la fonction  $\text{FIX}(u/\pi/3)$ , nous obtenons la position du vecteur sur le cercle chapitre 3 figure [3.8] (six position).

- Génération des six signaux de commande

Les paramètres d'entrées sont :

- Les quatre signaux équivalents aux temps figure [4.11].
- La position du vecteur.

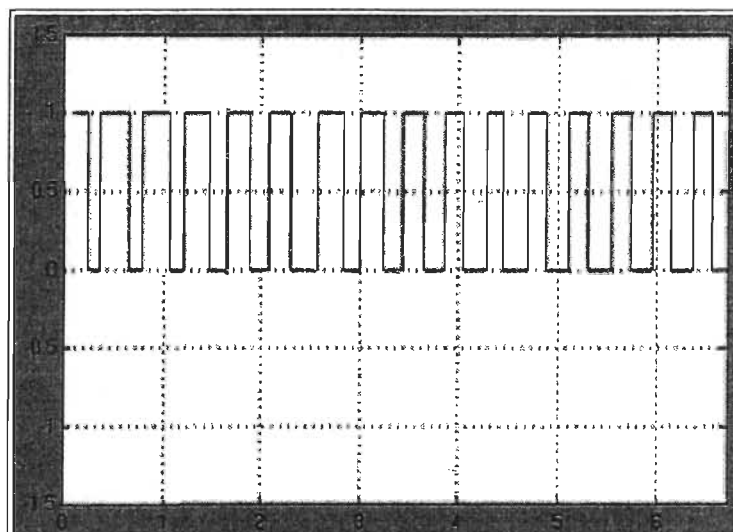
En formant les signaux de commandes pour chaque position, on peut sélectionner avec des multiplexeurs et des démultiplexeurs les 3 signaux de commande des trois transistors de haut et en inversant ces derniers on obtient la commande des trois transistors du bas.



**Figure 4.12 Multiplexage des signaux de la commande**

En tenant compte du motif de commande, la tension de sortie  $V_o$  a été générée pour une fréquence égale à 60 Hz. La figure [4.13] montre la simulation du motif de commande des interrupteurs semi-conducteurs de l'onduleur.





**Figure 4.13 Résultats de simulation**

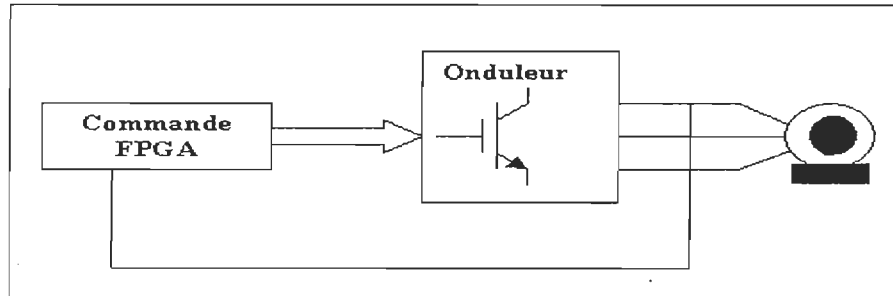
Pour implanter la commande avec des composantes élémentaires et peu coûteuse tels que les amplificateurs opérationnel sont sensibles au vieillissement et aux dérives de température. Quel que soit le soin avec lequel une implantation analogique a été conçue, elle présentera des caractéristiques indésirables avec le temps. Par ailleurs, entre autres inconvénients, on peut noter qu'une implémentation analogique est un circuit généralement câblé, rendant ainsi toute évolution difficile et quoiqu'il soit limité.

Avec les progrès de la micro-électronique, il est devenu possible de réaliser des implantations numériques à fréquence élevée dans un seul circuit. Le tableau [4.1] synthétise les principaux avantages et inconvénients des implantations analogiques et numériques [HsPe98].

**Tableau 4.1 Comparaison entre deux Implantation analogique et numérique**

	Implantation analogique	Implantation numérique
Avantages	Bande passante élevée Conception aisée	Solution évolutive Insensibilité à l'environnement Comportement précis Algorithmes sophistiqués
Inconvénients	Vieillessement des composants Dérive de température Conception câblé Conception figée Sensibilité à l'environnement	Approximation du réel qui crée des difficultés techniques Nécessité du circuit particulier pour chaque application différent

La figure [4.14] représente notre système numérique, le circuit de commande est un FPGA.



**Figure 4. 14 Schéma synoptique de la commande numérique**

Implantation : C'est dans cette étape que le système est partitionné et que ses différentes parties sont implantées. Puis le système est simulé pour en vérifier la fonctionnalité et le respect de ses contraintes. Les différentes parties de l'implantation sont expliquées plus en détail dans les points suivants :

- Partitionnement : Dans cette partie de la conception, une architecture est établie et utilisée pour déterminer les parties du comportement qui seront implantées en logiciel et celles implantées en matériel. Un partitionnement particulier est appelé un modèle et consiste à une des représentations possibles du système.

Afin d'utiliser une architecture spécifique, les différentes parties du comportement doivent être décomposées en blocs. Plusieurs facteurs peuvent influencer la façon dont le partitionnement entre le logiciel et le matériel sera fait [Axel97] :

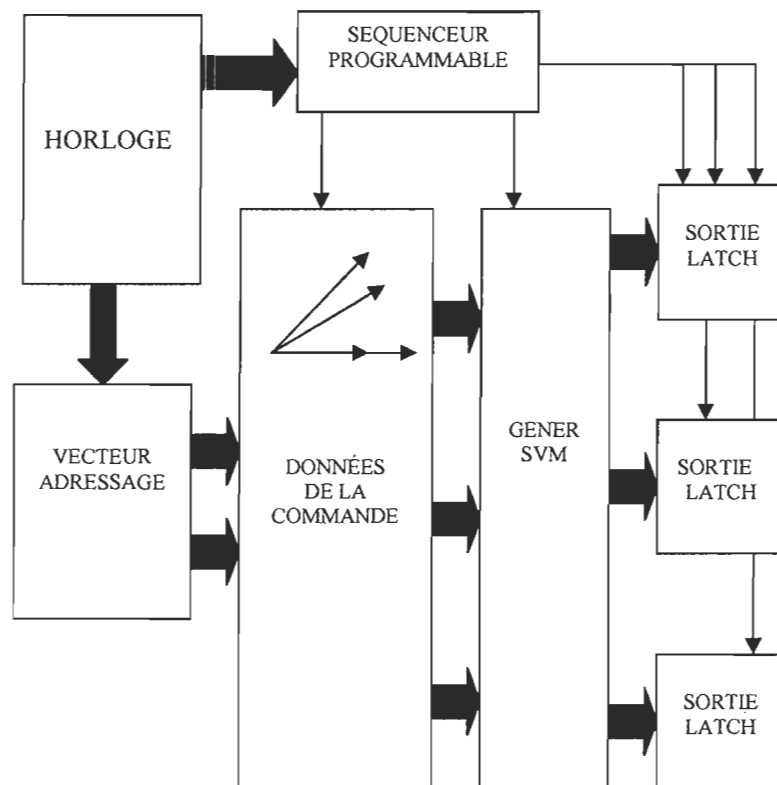
- Les performances globales du système.
- Le coût de l'implantation du système.
- La facilité à modifier les parties logicielles ou matérielles (avec par exemple des composants matériels interchangeables) du système.
- La nature des calculs, c'est-à-dire la façon dont le système traite les données.

- Le parallélisme entre les composants.
- Le temps de communications entre les composants.

### 4. 3 L'architecture proposée

#### 4. 3.1 Description du processeur

Selon l'analyse architecturale et le cahier de charges, nous proposons l'architecture présentée à la figure [4.15]



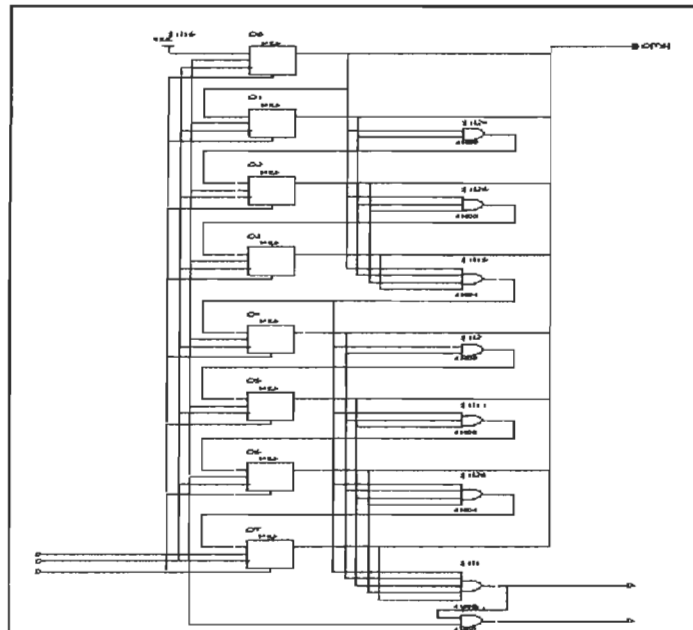
**Figure 4. 15 Structure du processeur**

Il est essentiel de s'assurer que la commande soit conforme avec le comportement fonctionnel exigé et qu'il rencontre toutes les contraintes de synchronisation et de fonctionnalité qui sont imposées au design VLSI. La description et la fonctionnalité des différents blocs de la commande SVM seront détaillées dans les sections suivantes.

#### 4.3.1.1 Le bloc vecteur d'adressage :

Après l'étude d'évolution des boucles de balayage des données stockées dans les mémoires ROM on a adopté des algorithmes particuliers, on peut le modifier en fonction du type d'adressage effectué dans la commande. Pour le test fonctionnel on a utilisé des compteurs synchrones dans le bloc d'adressage

Il y en a deux principaux types de compteurs, le premier joue le rôle d'adressage des mémoires ROM et le deuxième joue le rôle de sélection des mémoires ou le séquenceur programmable.



**Figure 4.16 Structure d'un compteur**

Les entrées des compteurs Clk sont l'horloge, ECLK active le comptage, l'entrée CLEAR est pour la remise à zéro du comptage, le signal TC indique la fin de comptage pour activation des tampons (buffer) et le BUS de comptage.

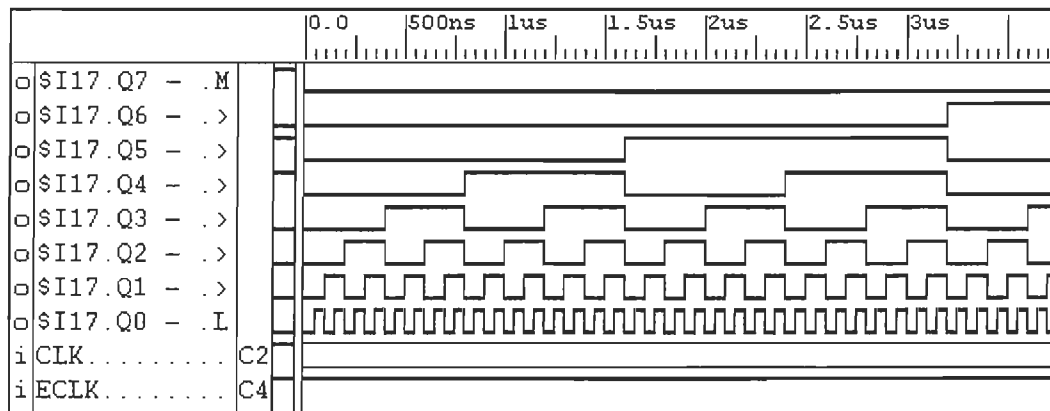
Vérification fonctionnelle : figure [4.16] représente les résultats obtenus durant la simulation et la synthèse [EHB93].

## Le test fonctionnel

Il faut répondre à trois questions :

- 1- Le système exécutera-t-il la fonctionnalité voulue ?
- 2- Existe-t-il un vecteur d'entrée qui puisse mettre le système en défaut ?
- 3- La précision voulue est-elle assurée pour n'importe quel vecteur de test ?

Pour répondre aux deux premières questions, nous avons recours à la vérification et à la validation fonctionnelle par bloc qu'on va voir dans ce chapitre. La Figure [4.17] représente les signaux obtenus lors de la vérification fonctionnelle [GuDe93].



**Figure 4.17 Séquencement de compteur**

CLK : horloge

ECLK : activé le comptage

Q0-Q7 : bus de sortie

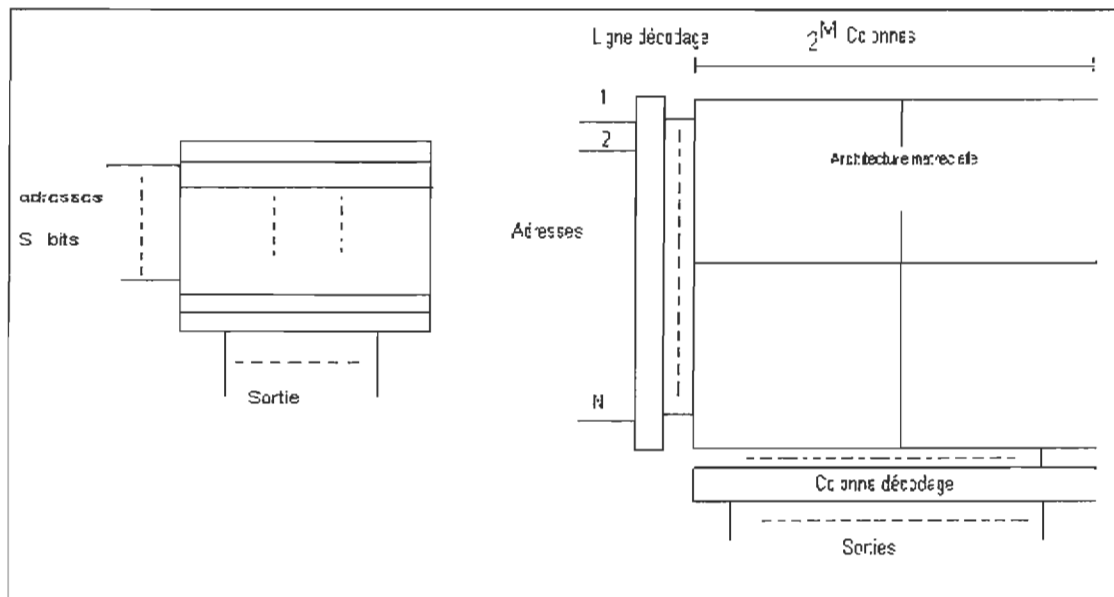
Le test de compteur est fait à partir d'une horloge de 60MHZ

### 4.3.1.2 Le bloc de données de la commande :

La taille du bloc de données est déterminé selon l'application et les contraintes du système. La grosseur d'un bloc peut aller d'un bloc de base qui est en fait la plus petite unité possible, comme une instruction jusqu'au système complet. Lorsque le partitionnement est à grains fins, c'est à dire qu'il y a un grand nombre de petits blocs, les résultats obtenus seront plus précis. Par contre, vu le nombre élevé de blocs à traiter et le nombre de combinaisons possibles entre le partitionnement logiciel et matériel, le temps de calcul sera beaucoup plus long. On doit donc trouver un juste milieu entre la taille des blocs et la vitesse du temps de calcul [HBKG98].

En général, nous allons appeler un partitionnement à grain fin, un partitionnement au niveau des blocs de base ou des instructions comme l'ont fait Ernst, Henkel et Brener dans Cosyma [HHE94], Gupta et De Micheli avec Vulcan [GuDe93], ainsi que Knudson et Madson avec Lycos [User99]. Tandis que Srivastona et Brodersen [SrBr91] et Athanos et Silverman [AtSi93] ont utilisé une approche à gros grains, ce qui signifie un partitionnement au niveau des tâches ou des processus [HeLi98].

On utilise dans ce bloc de données de la commande, des mémoires ROM (ROM 256 mots de 8 bit ) dans le but de paralléliser au maximum les lectures des données devant se rendre vers l'unité de la génération de la commande. Pour atteindre les performances suivant vitesse, capacité et consommation de puissance il est nécessaire d'utiliser des architectures matricielles des mémoires ROM dans notre implantation figure [4.18].



**Figure 4.18 Architecture matricielle**

Où

$$S = N + M$$

$$2^S = 2^N * 2^M$$

S : nombre de bits à entrer

N : nombre de lignes à l'entrée

M : nombre de lignes à la sortie

Pour fin de clarté nous avons dessiné et simulé seulement le chemin critique du bloc mémoire pour ne pas écrire les vecteurs de test de ce dernier au complet,

La figure [4.19] représente le schéma bloc une ROM qu'on a simulé et les résultats obtenus sont représentés à la figure [4.20].

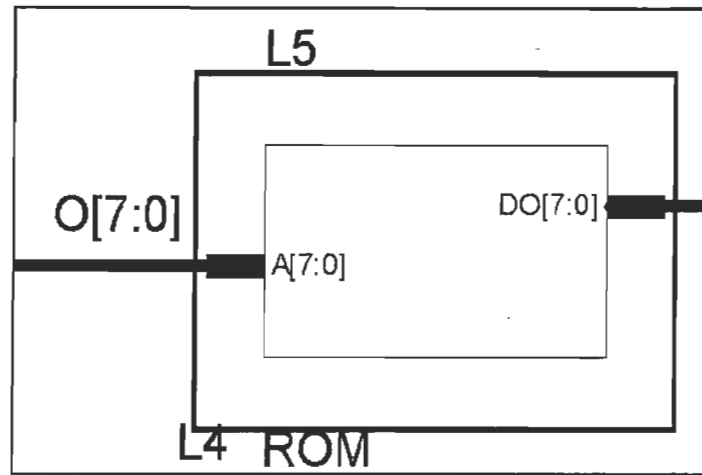


Figure 4.19 Bloc mémoire

O[7:0] : Bus d'adressage  
DO[7:0] : Bus de sortie

### Le test fonctionnel

Pour la simulation, nous adoptons un procédé à l'écriture des données de notre commande dans des cases mémoire (adresse) pour ensuite relire les mêmes données. La simulation de la mémoire fait apparaître le délai de la lecture qui est très élevé 45ns. Figure [4.20] représente les signaux obtenus lors de la vérification fonctionnelle.

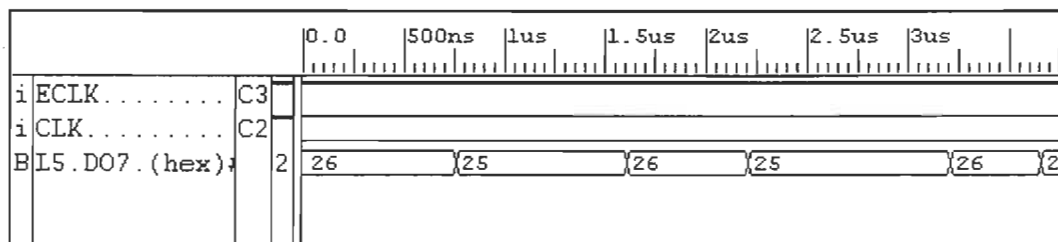


Figure 4.20 Séquencement de la mémoire

#### 4.3.1.3 Le bloc séquenceur programmable :

Le séquenceur a pour rôle d'assurer le fonctionnement synchronisé de l'ensemble selon un algorithme préalablement chargé. Vu le caractère flexible du processeur, on ne peut que choisir une architecture micro programmable.

L'unité de commande doit avoir les caractéristiques suivantes :

- pouvoir exécuter une boucle imbriquée
- pouvoir effectuer des sauts de retours de la lecture des données dans la mémoire
- avoir la possibilité d'accéder l'extérieur au processeur

La structure de bus :

Afin de réduire le coût de la mise en œuvre, il faut choisir la structure de bus minimum pour soutenir l'ensemble des transferts de données dans le processeur, et vers l'extérieur du processeur. Les flots de données vers le processeur concernent l'horloge ou les différents transferts internes entre les blocs. Le reste des blocs ce sont des multiplexeurs (génération de la commande SVM) et des Buffers (sotie).

Le test des multiplexeurs est fait à partir d'une horloge de 60 MHZ.

#### Testabilité

Nous avons donc opté pour une méthode de test par bloc qui permettrait d'évaluer partiellement l'état de notre architecture par des outils de simulation qui confirment l'atteinte de l'objectif principal de notre travail. Tableau [4.2].

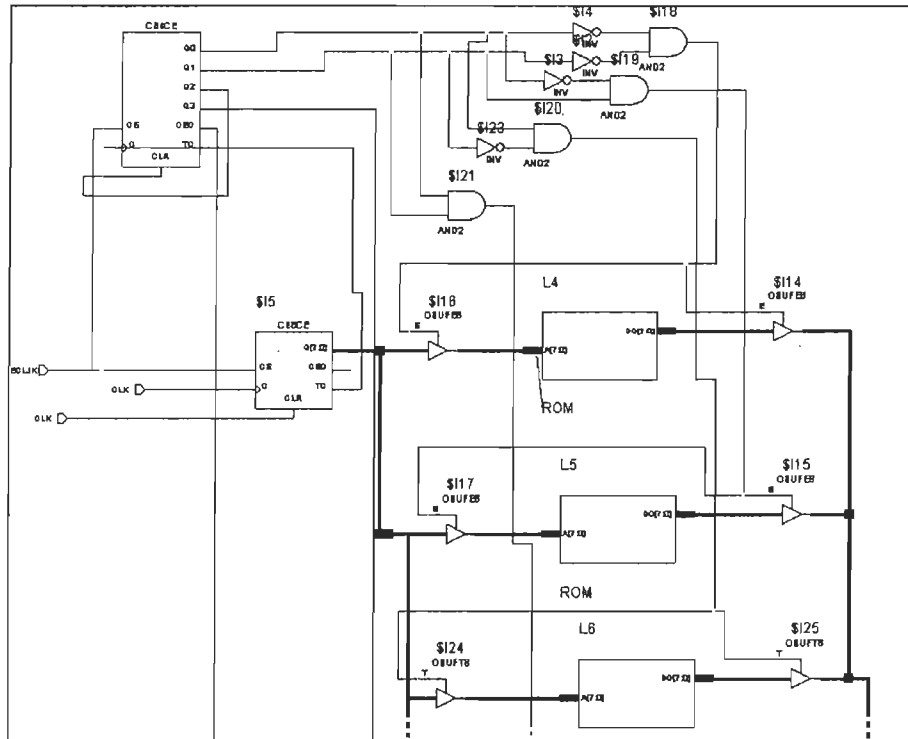
**Tableau 4.2 Résultats de simulation**

<b>Bloc</b>	<b>Description du test</b>	<b>Résultats</b>
Les compteurs d'adressage et de sélection	Reset, initialisation, test de fin de boucle	Fonctionnement à 60Mhz et indicateur de zéro.
Les blocs mémoires	Test du chemin pour relire les données stockées dans la mémoire (8 bits)	Délai 45ns pour obtenir les données stockées dan la mémoire (8bits)
Multiplexeurs	Test de fonctionnement	Fonctionnement à 60Mhz



## 4.4 Résultats de l'implantation globale

Après avoir simulé, testé et synthétiser chaque bloc indépendant des autre pour valider le fonctionnement des composantes utilisées dans les blocs et à la fin nous assemblons le tout pour le simuler, synthétiser et le tester figure [4.21]; les différents tests entrepris sont résumés au tableau [4.2].



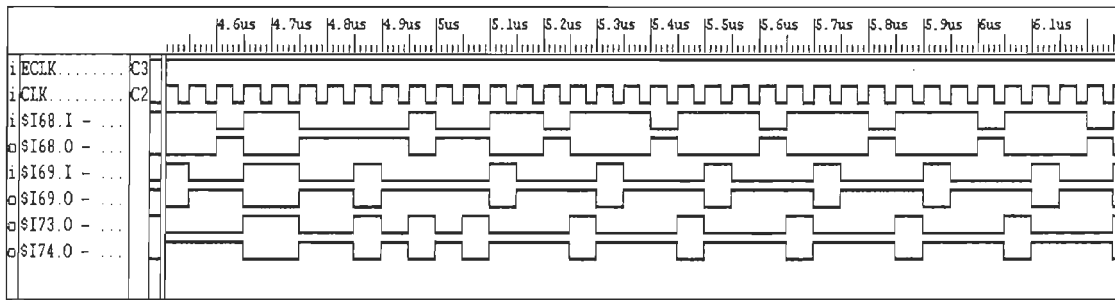
**Figure 4.21 La commande modulation vectorielle spatial**

Pour le test fonctionnel, on a obtenu les résultats apparaissant à la figure [4.18] qui répond au cahier de charge :

- fonctionnalité voulue
- atteindre un résultat semblable aux résultats obtenu par simulation, cela malgré une plus grande taille de données à stocker dans les mémoires
- précision assurée pour n'importe quel vecteur de test

- atteindre une vitesse de traitement des informations égale ou supérieure à celle de la simulation pour que la tension de sortie  $V_o$  soit générée pour une fréquence bien déterminée.

Entre autres, un paramètre à ne pas négliger était la flexibilité en terme de programmation. L'architecture devait accepter le chargement de programmes variés, tout en étant dédiée à la commande SVM tel que décrit au chapitre 3.



**Figure 4.22 Les signaux de la commande modulation vectorielle spatial**

**Tableau 4 .3 Signaux de commande**

	SA	SB	SC	<u>Signaux de sortie:</u>	b5	b4	b3	b4	b1	b0
État 1:	1	0	0		1	1	0	0	0	1
État 2:	1	1	0		1	0	0	0	1	1
État 3:	0	1	0		1	0	1	0	1	0
État 4:	0	1	1		0	0	1	1	1	0
État 5:	0	0	1		0	1	1	1	0	0
État 6:	1	0	1		0	1	0	1	0	1
État 7:	1	1	1		0	0	0	1	1	1
État 8:	0	0	0		1	1	1	0	0	0

On ajoute des inverseurs devant  $S_A$ ,  $S_B$ ,  $S_C$  pour obtenir les sorties de notre commande avec une latence de 45ns figure [4.22].

## 4. 5 Synthèse optimisation

Normalement, la réalisation de la commande SVM peut se faire avec plusieurs types de technologies de circuits tels que : les composantes normalisées ou programmables (microcontrôleur, DSP...), les cellules normalisées et les circuits dédiés. Généralement, les concepteurs de circuits intégrés cherchent à obtenir une performance optimale et à réduire le temps de conception, le temps de fabrication ainsi que le coût de production. Par ailleurs, Il faut signaler que les composantes programmables représentent une solution avantageuse qui respecte, pour un faible taux de production, les différents critères précités.

Après avoir simulé et fait le test fonctionnel de toutes les blocs et les sous blocs, il est nécessaire d'effectuer la synthèse de la description VHDL de la commande de modulation vectorielle spatial en un ensemble des portes logiques [VaGa95c].

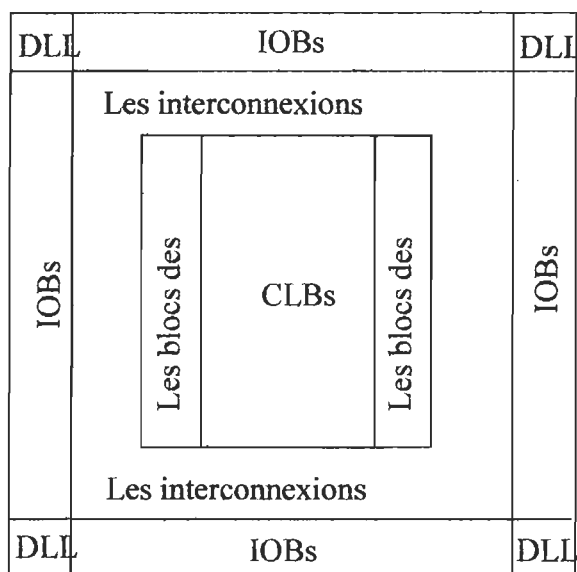
L'outil de synthèse est **Design Compiler Synopsys** et la librairie cible est la famille Xilinx (Vertex). Le résultat obtenu, c'est-à-dire du circuit final, c'est la génération des fichiers permettant la configuration des circuits programmables pour la réalisation pratique. Les performances du processeur complet sont évaluées en termes de temps et de surface. La vitesse globale du circuit sera dictée par le chemin le plus lent des blocs du processeur.

Le processeur de la commande SVM a été conçu pour être mis en œuvre avec des circuits FPGAs ( Field Programmable Gate Arrays) reprogrammables de la famille Xilinx (Vertex).

En outre, les circuits FPGAs de Xilinx sont organisés en matrice de cellules, chacune d'elles, peut être reliée par des connexions locales et globales programmables. Ces circuits contiennent essentiellement trois types de ressources : les cellules logiques CLB (Configurable Logic Block), les blocs d'entrées et de sorties IOB (Input Output Block) et les interconnexions. Les CLBs sont des cellules programmables pour réaliser les fonctions combinatoires et séquentielles de notre architecture à implanter. Chaque CLB contient un nombre des LUTs (Look Up Table) à 4 entrées ( 8 pour la famille Virtex2 de Xilinx) et un nombre de bascules (de même 8 pour la famille Virtex de Xilinx). Les IOBs, qui sont comme tampons aux entrées/sorties du FPGA, peuvent être programmés comme des entrées ou sorties avec ou sans bascules, et elles peuvent être également programmées comme des entrées / sorties à trois états avec ou sans

résistance de tirage (Pullup Resistor). Par ailleurs, les unités des interconnexions sont formées par des transistors appelés (Programmable Interconnect Points) qui assurent les interconnexions entre les différentes cellules du circuit programmable.

L'architecture interne de la famille Virtex, est illustrée dans la figure [4.23].



**Figure 4.23 Architecture interne d'un circuit de la famille Virtex**

Les Virtex sont établies avec une lithographie de 0,22 microns (avec une carte routière de 0,18 microns) et une technologie à cinq couches de métal.

Après La synthèse, le placement et le routage par les outils Synopsys et Seamless de Mentor Graphics en utilisant le circuit XC2V256FG250 de la famille Virtex de Xilinx du processeur de la commande on a obtenu une surface très grand avec un chemin critique de 45ns.

D'après les résultats obtenus, nous constatons que d'après le processeur de la commande, la surface est très grosse par rapport à ce qu'il a été prévu. Ceci est dû à la quantité énorme des composantes logiques constituant la commande SVM et la limitation d'organisation de l'architecture interne des FPGAs.

Pour le circuit XC2V256FG250, famille de Xilinx, il faut suivre des conditions pour implanter des architectures, des algorithmes et des applications pour définir des mémoires (ROM). Et pour cela nous avons utilisé dans le bloc de mémoire qu'on a définit dans la section 4.2.2, des mémoires ROM (ROM 256 mots de 8 bit) qui répond aux exigences d'implantations. Le circuit

choisi à une structure interne qu'exige moins de 256 mots par bloc avec le nombre de bits de chaque mot de mémoires soit égal avec les bits d'adressage. Ce qui lance un autre défi de recherche pour rendre la structure interne des circuits numériques très simple et donne plus de possibilité aux utilisateurs pour l'implantation de leurs application indépendamment des composantes programmable.

## **Conclusion**

Au niveau de ce dernier chapitre, nous avons proposé une architecture de la commande de modulation vectorielle spatial type adaptable avec les différents types d'applications. Cette architecture développée de la commande a été réalisée en langage VHDL, indépendant de la structure d'une certaine librairie d'intégration. À la suite, afin d'évaluer et de valider cette architecture, on a créé des environnements de test utilisés pour tester la fonctionnalité de chaque blocs indépendant des autres et à la fin un test globale. Également, les performances de cette commande ont été mises en œuvre par l'utilisation de l'outil synthèse **SYNOPSIS** de ces différents blocs sur une structure logique programmable d'un circuit numérique. D'après les résultats obtenus par la synthèse, la fréquence obtenue pour la commande vectorielle spatial était supérieur à celle des commandes existantes, ce qui était attendu et ce qui donne à notre application une grande flexibilité pour la modifier. Par contre au niveau de la surface on a trouvé trop de difficulté pour la diminuer cela est dû à la quantité énorme des composantes logiques constituant la commande à cause de la conversion analogique numérique qu'on a fait pour l'implantation dans le circuit numérique.

Avec une complexité croissante des systèmes à intégrer et des contraintes technologiques, temporelles et budgétaires toujours plus fortes, les approches classiques d'intégration ne suffisent plus, c'est le cas pour nous puisqu'on se retrouve avec 965 000 cellules comme surface de processeur de la commande SVM.

# *Conclusion générale*

Dans le Chapitre 1 de ce rapport, nous avons présenté un rappel théorique des onduleurs monophasés de tension; une étude de la commutation, position fermé-position ouvert, des interrupteurs semi-conducteurs de l'onduleur et une analyse des pertes de puissance dans l'onduleur. Le Chapitre 2 présente une révision des techniques de commande à modulation de largeur d'impulsions (MLI) avec un intérêt spécial à la technique MLI précalculée, qui est la technique utilisée pour commander notre onduleur.

Nous avons présenté dans le Chapitre 3 une étude de la commutation, position fermé-ouvert, des interrupteurs semi-conducteurs de l'onduleur triphasé commandé par la modulation vectorielle et la détermination des angles et des vecteurs de puissance de notre système. Et dans le dernier chapitre nous traitons l'implémentation de la commande SVM

dans un circuit numérique et on présente les outils matériels et logiciels qui permettent d'implanter une commande sur la base d'une application concrète ainsi que la méthodologie employée pour la mise en œuvre harmonieuse de ces différents outils.

Nous avons également clairement établi les limites de notre étude en précisant le sens que nous donnons au concept de la commande. En particulier, nous avons extrait les différents blocs fonctionnels et nous avons précisé la tâche de chacun d'eux. Nous exposons les problèmes vis à vis des besoins actuels et dans un deuxième temps, des méthodes plus avancées qui font l'objet de nombreux travaux et qui visent à apporter des solutions aux insuffisances des approches classiques.

L'objectif de notre travail était double : d'une part adopter une commande de modulation vectorielle spatial à implanté et valider une méthodologie d'implantation de la commande de modulation vectorielle spatial numériquement avec les différentes techniques et outils utilisés pour l'implantation.

Pour atteindre ces objectifs, nous avons procédé de la façon suivante:

- Nous avons analysé les techniques employées pour la mise en œuvre de l'architecture adoptée pour la commande.
- Nous avons ensuite étudié l'état des techniques et outils employés en microélectronique pour les simulation et synthèse.
- A partir des résultats précédents, nous avons alors proposé une méthode spécifique à la mise en oeuvre des algorithmes de notre commande.
- Finalement, nous avons adapté le programme VHDL aux contraintes de système qu'on a adopté.

Même si certains travaux portant sur l'intégration des algorithmes de commande des machines électriques ont présenté la nécessité d'aborder la conception autrement, leur approche ne permet pas de résoudre les problèmes évoqués. Ces travaux nous permettent néanmoins de mettre en évidence des spécificités de l'intégration de la commande dont il faudra tenir compte lors de l'application d'une méthodologie d'implantation. On gardera ainsi à l'esprit l'importance du choix de la fréquence de calcul du système, le problème de

l'arithmétique binaire à employer et du séquençement des opérations vis à vis de la précision binaire, et finalement, l'intérêt qu'il peut y avoir à modifier l'algorithme de départ pour tenir compte de la réalisation physique.

Après avoir formalisé les problèmes liés à l'intégration de la commande, et situé leur traitement dans la méthodologie, nous avons proposé des solutions pour les résoudre. Nous avons également défini des restrictions au champ d'application de la méthode afin de limiter la complexité des problèmes traités. Nous définissons en particulier l'architecture des systèmes de commande que nous utilisons.

Chaque type de système de commande a ses contraintes particulières. Il a donc des contraintes de temps très strictes à respecter. Par contre, une fois les temps d'exécution atteints, ce n'est pas avantageux de pousser encore plus loin les performances. Un autre critère à examiner est la dissipation de puissance dans notre cas en a aucun problème car en ce retrouve dans le cas normale de dissipation de puissance. Le troisième critère que nous devons examiner est la surface du matériel. Plus la surface matérielle est grande, plus le coût en sera élevé ce qui rendre notre application très coûteux. Donc, pour minimiser les coûts de fabrication, nous plaçons la plus grande quantité de comportement possible d'une application en logiciel. Car, la partie logicielle reste de la même grosseur quel que soit le nombre de blocs implantés en logiciel.



# *Bibliographie*

[Axel97] J. Axelsson. (1997). Analysis and Synthesis of Heterogeneous Real-Time systems, Linköping Studies in Science and Technology, Suède, 190 pages.

[AdTh96] Adams Jay K., Thomas Donald E. (1996). The Design of Mixed Hardware/Software Systems, DAC '96. Proceedings of the 33rd annual conference on Design automation conference, 515-520.

[Bo92] B. K. Bose. "Recent advances in power electronics." IEEE Transactions on Power electronics, Vol. 7, No. 1, January 1992 pp 2 - 16.

[BoZi 88] M. A. Boost, P.D. Ziogas, "State-of-the-art carrier PWM techniques: a critical evaluation". Transactions on Industry Applications, Vol. 24, No. 2, March/April, 1988, pp 271 - 280.

[Ch87] B. Chauveau. "Étude de la commande d'un onduleur alimentant une charge non-linéaire à partir d'une source impédante: optimisation de la distortion harmonique de sa tension de sortie". Thèse de Docteur de l'Institut National Polytechnique de Toulouse, France 1987.

[DRG97] Dasdan Ali, Ramanathan Dinesh, Gupta Rajesh K. (1998). Rate Derivation and Its Applications to Reactive, Real-Time Embedded Systems, DAC '98. Proceedings of the 35th annual conference on Design automation conference, 263-268.

[DeMi94] De Micheli Giovanni. (1994). Synthesis and Optimization of Digital Circuits, McGraw Hill, États-Unis, 579 pages.

[EHB93] Ernst R., Henkel J, Benner T. (1993). Hardware-Software Cosynthesis for Microcontrollers, IEEE Design & Test Computer, December 1993, 64-75.

[GuLi97] Gupta Rajesh K, Liao Stan Y, Using a Programming Language for Digital System Design, IEEE Design & Test of Computers, April-June 1997, 72-80.

[GaVa95] Daniel D. Gajski, Frank Vahid. (1995). Specification and design of embedded hardware/software systems, IEEE Design & Test of Computers, vol. 12, no. 1, 53.

[GaRa94] Gajski Daniel D, Ramachandran Loganath. (1994). Introduction to High-Level Synthesis, IEEE Design & Test of Computers, Vol. 11, No. 4, 44-54.

[GuDe93] Gupta R.K, De Micheli G. (1993). Hardware-Software Cosynthesis for Digital Systems, IEEE Design & Test of Computer, 29-41.

[HsPe98] Hsieh Cheng-Ta, Pedram Massoud. (1998). Microprocessor Power Estimation Using Profile-Driver Program Synthesis, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No 11, 1080-1089.

[HeLi98] Henkel Jorg, Li Yanbing. (1998). Energy-Conscious HW/SW-Partitioning of Embedded Systems : A Case Study on an MPEG-2 Encoder, CODES/CASHE '98. Proceedings of the sixth international workshop on Hardware/software codesign, 23-27.

[HHE94] Herrmann D, Henkel J., Ernst R. (1994). An Approach to the Adaptation of Estimated Cost Parameters in the Cosyma System, Proceedings of the 3rd International Workshop on Hardware/Software Codesign, 100-107.

[MaPe98] Macii Enrico, Pedram Massoud. (1998). High-Level Power Modeling, Estimation, and Optimization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 11.

[MoUn95] N. Mohan, T. M. Undeland, W. I. Robbins. Power electronics. Converters, applications, and design. 2ème édition. John Wiley & Sons, New York, 1995.

[MhRa93] M. H. Rashid, Power electronics. Circuits, devices, and applications. 2édition. Prentice-Hall, New Jersey, 1993.

[PaHo73] H. S. Patel, R. Hoft. "Generalized techniques of harmonic elimination and voltage control in thyristor inverters; Part I - "Harmonics elimination." Transactions on Industry Applications, Vol. IA-9, No. 3, May/June 1973, pp 310-317. Part II - "Voltage control techniques" IEEE Transactions on Industry Applications, Vol. IA-10, No. 5, Sept/Oct. 1974 pp 666-673.

[RuJa98] Russell Jeffry, Jacome Margarida. (1998). Software Power Estimation and Optimization for High Performance, 32-bit Embedded Processors, Proceedings of the 1998 IEEE International Conference on Computer Design, 328-333.

[SrBr91] Srivastava M.B, Brodersen R.W. (1991). Rapid-Prototyping of Hardware and Software in a Unified Framework, International conference on Computer-Aided Design, Los Alamitos, California, 152-155.

[SeLa89] G. Segulier, F. Labrique. Les convertisseurs de l'électronique de puissance. 4 La conversion continu-alternatif. Lavoisier, France 1989.

[Th98] K. Thorborg. Power electronics. Prentice-Hall International, Great Britain, 1998.

[TMW94] Tiwari Vivek, Malik Sharad, Wolfe Andrew. (1994). Power Analysis of Embedded Software : A First Step Toward Software Power Minimization, IEEE Transactions on Very Large Scale Integration Systems, Vol. 2, No. 4, 437-445.

[User99] [http://www.it.dtu.dk.lycos/users\\_guide](http://www.it.dtu.dk.lycos/users_guide)

[VaGa95c] Vahid Frank, Gajski Daniel D. (1995). Incremental Hardware Estimation During Hardware/Software Functional Partitioning, IEEE Transactions on Very Large Scale Integration Systems, Vol. 3, No. 3, 459-464.

[Zi80] P. D. Ziogas. "Optimum voltage and harmonic control PWM techniques for three-phase static UPS systems". IEEE Transactions on Industry Applications, Vol. IA-16, No. 4, July/August 1980, pp 542 - 580.

# **Annexe**

### Résolution d'une équation différentielle :

Remplaçant cette dérivée dans l'équation (1.13) chapitre 1 on arrive à l'équation différentielle suivante:

$$RC \frac{d^2 v_2}{dt^2} + 2 \frac{d^2 v_2}{dt} + I_{ch} \frac{t}{Ct_f} = 0 \quad (\text{A. 1})$$

$$\frac{d^2 v_2}{dt^2} + \frac{2}{RC} \frac{dv_2}{dt} = -I_{ch} \frac{t}{RC^2 t_f}$$

Posons  $A = \frac{2}{RC} = \frac{2}{\tau}$  et  $B = -I_{ch} \frac{1}{RC^2 t_f} = -I_{ch} \frac{1}{C\tau t_f}$  (A. 2)

Où  $\tau = RC$

Aussi,  $v_2'' = \frac{d^2 v_2}{dt^2}$  et  $v_2' = \frac{dv_2}{dt}$

$$\Rightarrow v_2'' + Av_2' = Bt \quad (\text{A. 3})$$

La solution totale de cette équation différentielle peut s'exprimer comme suit (chapitre 1 page 18):

$$v_2 = v_{2c} + v_{2p} \quad (\text{A. 4})$$

où  $V_{2c}$  = solution complémentaire,

$V_{2p}$  = solution particulière.

La solution complémentaire  $v_{2c}$ , est de la forme,

$$v_{2c} = k_1 + k_2 e^{-At} \quad (\text{A. 5})$$

où  $k_1$  et  $k_2$  sont des constantes trouvées à partir des tensions initiales des condensateurs,  $v_{10}$  et  $v_{20}$ .

La solution particulière  $v_{2p}$  est obtenue comme suit:

soit  $v_{2p} = g_1 + g_2 e^{-At}$  (A. 6)

où  $g_1$  et  $g_2$  sont des fonctions de  $t$ . Prenons la dérivée de  $v_{2p}$ ,

$$v'_{2p} = g'_1 + g'_2 e^{-At} - Ag_2 e^{-At}$$

posons  $g'_1 + g'_2 e^{-At} = 0$  (A. 7)

Prenons la dérivée seconde de cette équation,

$$v''_{2p} = -Ag'_2 e^{-At} + A^2 g_2 e^{-At} \quad (A. 8)$$

Remplaçant les expressions (A. 8) et (A.7) dans l'équation (A. 1) et simplifiant, nous arrivons à l'expression suivante:

$$-Ag'_2 e^{-At} = Bt \quad g'_2 = -\frac{Bte^{At}}{A} \quad (A. 9)$$

$$\Rightarrow g_2 = -\frac{B}{A} \int te^{At} dt$$

En utilisant les tables mathématiques [5] pour résoudre l'intégrale,  $g_2$  devient,

$$\Rightarrow g_2 = -\frac{Be^{At}}{A^3} (At - 1) \quad (A. 10)$$

De l'équation (1.16)  $g'_1 = -g'_2 e^{-At}$ , alors, remplaçant  $g'_2$ ,

$$g'_1 = -\frac{Bte^{At}}{A} e^{-At} = \frac{Bt}{A}$$

$$\Rightarrow g_1 = -\frac{B}{A} t^2 \quad (A. 11)$$

Remplaçant les expressions pour  $g_1$  et  $g_2$  dans l'équation (A. 6), nous trouvons que la solution particulière  $v_{2p}$  est égale à :

$$\Rightarrow v_{2p} = \frac{B}{A} \left[ t \left( \frac{t}{2} - \frac{1}{A} \right) + \frac{1}{A^2} \right]$$

La somme de cette expression plus la solution complémentaire  $V_{2c}$ , (voir équation (A. 5)), ce qui en résulte la solution totale de l'équation différentielle :

$$v_2 = k_1 + k_2 e^{-At} + \frac{B}{A} \left[ t \left( \frac{t}{2} - \frac{1}{A} \right) + \frac{1}{A^2} \right] \quad (\text{A. 12})$$

Les constantes  $K_1$  et  $K_2$  sont trouvées en utilisant les valeurs initiales pour  $V_2$ ,

$$\text{A } t=0 \quad \Rightarrow v_{20} = v \quad \text{et} \quad v'_{20} = 0$$

$$\Rightarrow \quad V_E = k_1 + k_2 + B/A^3 \quad \ominus \quad k_1 = V_E - k_2 - B/A^3$$

$$v'_2 = -Ak_2 e^{-At} + \frac{B}{A} \left( t - \frac{1}{A} \right) \Rightarrow 0 = -Ak_2 - \frac{B}{A^2}$$

$$k_2 = -B/A^3 \quad \text{et} \quad k_1 = V_E$$

Remplaçant  $K_1$  et  $K_2$  dans l'équation (A. 12), l'expression de  $V_2$  devient,

$$v_2 = V_E - B e^{-At} / A^3 + \frac{B}{A} \left[ t \left( \frac{t}{2} - \frac{1}{A} \right) + \frac{1}{A^2} \right]$$

$$v_2 = V_E - \frac{B}{A^3} \left[ e^{-At} - A^2 t \left( \frac{t}{2} - \frac{1}{A} \right) - 1 \right]$$

$$\text{Et} \quad v'_2 = \frac{B}{A^2} e^{-At} + \frac{B}{A} \left( t - \frac{1}{A} \right) = \frac{B}{A} \left[ \frac{e^{-At}}{A} + t - \frac{1}{A} \right]$$

$$v_1 = V_E - v_2 - RC \frac{dv_2}{dt}$$

Remplaçant les expressions pour  $V_2$  et  $V'_2$  et faisant des simplifications, nous arrivons à l'expression suivante:

$$v_1 = -\frac{B}{A^3} \left[ e^{-At} + A^2 t \left( \frac{t}{2} + \frac{1}{A} \right) - 1 \right]$$

$$\text{Et} \quad v_2 = V_E - \frac{B}{A^3} \left[ e^{-At} - A^2 t \left( \frac{t}{2} - \frac{1}{A} \right) - 1 \right]$$

Où  $A = \frac{2}{\tau}$   $B = \frac{I_{ch}}{t_f c}$  et  $\tau = RC$

Remplaçant A et B, les expressions de  $V_1$  et  $V_2$  deviennent:

$$v_1 = \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t}{\tau}} + \frac{2t}{\tau} \left( \frac{t}{\tau} + 1 \right) - 1 \right] \quad (\text{A. 13})$$

$$v_2 = V_E + \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t}{\tau}} - \frac{2t}{\tau} \left( \frac{t}{\tau} + 1 \right) - 1 \right] \quad (\text{A. 14})$$

Maintenant, nous trouvons les courants dans les condensateurs; ils sont donnés par les expressions (1.12):

$$i_1 = C \frac{dv_1}{dt} \quad i_2 = C \frac{dv_2}{dt}$$

alors,

$$i_1 = -\frac{I_{ch} R \tau}{4t_f} \left[ e^{-\frac{2t}{\tau}} - 2\frac{t}{\tau} - 1 \right] \quad (\text{A. 15})$$

$$i_2 = -\frac{I_{ch} R \tau}{4t_f} \left[ e^{-\frac{2t}{\tau}} + 2\frac{t}{\tau} + 1 \right]$$

Cette première étape se termine à  $t = t_f$ ; remplaçant ce temps  $t_f$  dans les équations (A. 13) et (A. 14), nous trouvons les valeurs initiales de  $V_1$  et  $V_2$  pour l'étape suivante.

$$\Rightarrow v_1(t = t_f) = \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t_f}{\tau}} + \frac{2t_f}{\tau} \left( \frac{t_f}{\tau} + 1 \right) - 1 \right] \quad (\text{A. 16})$$

$$\text{et } \Rightarrow v_2(t = t_f) = V_E + \frac{I_{ch} R \tau}{8t_f} \left[ e^{-\frac{2t_f}{\tau}} - \frac{2t_f}{\tau} \left( \frac{t_f}{\tau} + 1 \right) - 1 \right] \quad (\text{A. 17})$$

## Programme en VHDL

-----Compteurs d'adressages-----

```
-- LogiBLOX COUNTER Module "cont16"  
--  compteur LogiBLOX version D.19  
--  02 06:21:06 2001
```

```
LIBRARY IEEE;  
USE IEEE.std_logic_1164.ALL;
```

```
-- synopsys translate_off  
LIBRARY logiblox;  
USE logiblox.mvlutil.ALL;  
USE logiblox.mvlarith.ALL;  
USE logiblox.logiblox.ALL;  
-- synopsys translate_on
```

```
ENTITY cont16 IS  
  PORT(  
    CLK_EN: IN std_logic;  
    CLOCK: IN std_logic;  
    Q_OUT: OUT std_logic_vector(10 DOWNT0 0));  
END cont16;
```

```
ARCHITECTURE sim OF cont16 IS  
  SIGNAL START_PULSE: std_logic := '1';  
BEGIN  
  PROCESS  
    VARIABLE VD_IN: std_logic_vector(16 DOWNT0 0);  
    VARIABLE VLOAD: std_logic;  
    VARIABLE VUP_DN: std_logic;  
    VARIABLE VCLK_EN: std_logic;  
    VARIABLE VCLOCK: std_logic;  
    VARIABLE VASYNC_CTRL: std_logic;  
    VARIABLE VSYNC_CTRL: std_logic;  
    VARIABLE VQ_OUT: std_logic_vector(16 DOWNT0 0);  
    VARIABLE VTERM_CNT: std_logic;  
    VARIABLE lowest_count: std_logic_vector(16 DOWNT0 0) :=  
      ('0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0');  
    BEGIN  
      VLOAD := '0';  
      VUP_DN := '1';  
      VASYNC_CTRL := '0';  
      VSYNC_CTRL := '0';  
      VUP_DN := '0';  
      IF(  

```



```

        (CLOCK'EVENT AND stdbit2mvl(CLOCK)='1' AND
stdbit2mvl(CLOCK'LAST_VALUE)='0' AND stdbit2mvl(CLK_EN)='1')
        OR ( stdbit2mvl(START_PULSE)='1')
    ) THEN
        xb_counter(
            START_PULSE,
            VUP_DN,
            VASYNC_CTRL,
            ('0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','1'),
            VSYNC_CTRL,
            ('0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'),
            VLOAD,
            VD_IN,
            ('1','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'),
            VQ_OUT);
        Q_OUT <= VQ_OUT;
    ELSIF(
        (stdbit2mvl(CLOCK) = 'X')
        OR (CLOCK'EVENT AND stdbit2mvl(CLOCK)='1' AND
stdbit2mvl(CLOCK'LAST_VALUE)='0'
        AND stdbit2mvl(CLK_EN) = 'X' )
    ) THEN
        VQ_OUT := ('X','X','X','X','X','X','X','X','X','X','X','X','X','X','X','X','X');
        Q_OUT <= VQ_OUT;
    END IF;
    IF (START_PULSE='1') THEN
        START_PULSE <= '0' AFTER 1 ns;
    END IF;
    WAIT ON CLK_EN, CLOCK, START_PULSE;
END PROCESS;
END sim;
    ■ synopsys translate_on

```

```

-----
-- LogiBLOX COUNTER Module "cont16"
--  compteur LogiBLOX version D.19
--  02 06:21:06 2001

```

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;

```

```

-- synopsys translate_off
LIBRARY logiblox;
USE logiblox.mvlutil.ALL;
USE logiblox.mvlarith.ALL;
USE logiblox.logiblox.ALL;

```

```
-- synopsys translate_on
```

```
ENTITY cont16 IS
```

```
  PORT(
```

```
    CLK_EN: IN std_logic;
```

```
    CLOCK: IN std_logic;
```

```
    Q_OUT: OUT std_logic_vector(10 DOWNT0 0));
```

```
END cont16;
```

```
ARCHITECTURE sim OF cont16 IS
```

```
  SIGNAL START_PULSE: std_logic := '1';
```

```
BEGIN
```

```
  PROCESS
```

```
    VARIABLE VD_IN: std_logic_vector(10 DOWNT0 0);
```

```
    VARIABLE VLOAD: std_logic;
```

```
    VARIABLE VUP_DN: std_logic;
```

```
    VARIABLE VCLK_EN: std_logic;
```

```
    VARIABLE VCLOCK: std_logic;
```

```
    VARIABLE VASYNC_CTRL: std_logic;
```

```
    VARIABLE VSYNC_CTRL: std_logic;
```

```
    VARIABLE VQ_OUT: std_logic_vector(10 DOWNT0 0);
```

```
    VARIABLE VTERM_CNT: std_logic;
```

```
    VARIABLE lowest_count: std_logic_vector(10 DOWNT0 0) :=  
('0','0','0','0','0','0','0','0','0','0','0','0');
```

```
  BEGIN
```

```
    VLOAD := '0';
```

```
    VUP_DN := '1';
```

```
    VASYNC_CTRL := '0';
```

```
    VSYNC_CTRL := '0';
```

```
    VUP_DN := '0';
```

```
    IF(
```

```
      (CLOCK'EVENT AND stdbit2mvl(CLOCK)='1' AND  
stdbit2mvl(CLOCK'LAST_VALUE)='0' AND stdbit2mvl(CLK_EN)='1')
```

```
      OR ( stdbit2mvl(START_PULSE)='1')
```

```
    ) THEN
```

```
      xb_counter(
```

```
        START_PULSE,
```

```
        VUP_DN,
```

```
        VASYNC_CTRL,
```

```
        ('0','0','0','0','0','0','0','0','0','0','0','1'),
```

```
        VSYNC_CTRL,
```

```
        ('0','0','0','0','0','0','0','0','0','0','0','0'),
```

```
        VLOAD,
```

```
        VD_IN,
```

```
        ('1','0','0','0','0','0','0','0','0','0','0','0'),
```

```
        VQ_OUT);
```

```

Q_OUT <= VQ_OUT;
ELSIF(
  (stdbit2mvl(CLOCK) = 'X')
  OR (CLOCK'EVENT AND stdbit2mvl(CLOCK)='1' AND
stdbit2mvl(CLOCK'LAST_VALUE)='0'
  AND stdbit2mvl(CLK_EN) = 'X' )
) THEN
  VQ_OUT := ('X','X','X','X','X','X','X','X','X','X','X');
  Q_OUT <= VQ_OUT;
END IF;
IF (START_PULSE='1') THEN
  START_PULSE <= '0' AFTER 1 ns;
END IF;
WAIT ON CLK_EN, CLOCK, START_PULSE;
END PROCESS;
END sim;
■ synopsys translate_on

```

-----Mémoires ROM-----

```

--
-- "mmp1"
-- version D.19
-- 06 02:29:47 2001
-- Attributes
--  MODTYPE = ROM
--  BUS_WIDTH = 8
--  DEPTH = 256
--  MEMFILE = pre1
--  TRIM = FALSE
--  STYLE = MAX_SPEED
--  USE_RPM = FALSE

--; memfile pre2.mem for LogiBLOX symbol mmt
--; Created on Thursday, December 06, 2001 02:31:18

--; Header Section
--RADIX 10
--DEPTH 256
--WIDTH 8
--DEFAULT 32
--data 42; 42; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 37; 37; 37; 37; 37 ;
37 ; 37; 37; 37; 37; 37; 37; 37; 21; 42; 42; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38;
38; 38; 38; 38; 38; 38; 37; 37; 37; 37 ; 37; 37; 37; 21; 21; 42; 42; 42; 38; 38; 38; 38; 38; 38;
38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 37; 37; 37; 37; 37; 21; 21; 42; 42; 42; 38;
38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 37;
21; 21; 21; 42; 42; 42; 42; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38;
38; 38; 38; 38; 22; 22; 22; 22; 21; 21; 42; 42; 42; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38;
38; 38; 38; 38; 38; 38; 38; 38; 22; 22; 22; 22; 22; 22; 21; 21; 42; 42; 38; 38; 38; 38; 38;

```

```

38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 22; 22; 22; 22; 22; 22; 22; 22; 22; 22; 22; 21; 42;
42; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 38; 22; 22; 22; 22; 22; 22; 22;
22; 22; 22;
--255: 3
;32,64,128,16,32, 32,64,128,16,32,;

```

```

; end of LogiBLOX memfile

```

```

-----
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;

```

```

LIBRARY logiblox;
USE logiblox.mvlutil.ALL;
USE logiblox.mvlarith.ALL;
USE logiblox.logiblox.ALL;
-- synopsys translate_on

```

```

ENTITY mmp1 IS
  PORT(
    A: IN std_logic_vector(7 DOWNTO 0);
    DO: OUT std_logic_vector(7 DOWNTO 0));
END mmp1;

```

```

-- synopsys translate_off
ARCHITECTURE sim OF mmp1 IS
  SIGNAL START_PULSE: std_logic := '1';
  TYPE mem_data IS ARRAY (255 DOWNTO 0) OF std_logic_vector(7 DOWNTO 0);
BEGIN
  PROCESS
    VARIABLE VD: mem_data;
    VARIABLE first_time: BOOLEAN := TRUE;
  BEGIN
    IF (first_time) THEN
      VD(0) := ('0','0','1','0','0','0','0','0');
      VD(1) := ('0','1','0','0','0','0','0','0');
      VD(2) := ('1','0','0','0','0','0','0','0');
      VD(3) := ('0','0','0','1','0','0','0','0');
      VD(4) := ('0','0','1','0','0','0','0','0');
      VD(5) := ('0','0','1','0','0','0','0','0');
      VD(6) := ('0','1','0','0','0','0','0','0');
      VD(7) := ('1','0','0','0','0','0','0','0');
      VD(8) := ('0','0','0','1','0','0','0','0');
      VD(9) := ('1','0','1','1','0','0','0','0');
      VD(10) := ('0','0','1','0','0','0','0','0');
    END IF;
  END PROCESS

```

VD(11) := ('0','1','0','0','0','0','0','0');  
 VD(12) := ('1','0','0','0','0','0','0','0');  
 VD(13) := ('0','0','0','1','0','0','0','0');  
 VD(14) := ('0','0','1','0','0','0','0','0');  
 VD(15) := ('0','0','1','0','0','0','0','0');  
 VD(16) := ('0','1','0','0','0','0','0','0');  
 VD(17) := ('1','0','0','0','0','0','0','0');  
 VD(18) := ('0','0','0','1','0','0','0','0');  
 VD(19) := ('0','1','1','0','0','0','0','0');  
 VD(20) := ('0','1','0','1','0','0','0','0');  
 VD(21) := ('0','1','0','1','0','0','0','0');  
 VD(22) := ('0','1','0','1','0','0','0','0');  
 VD(23) := ('0','0','0','1','0','0','0','0');  
 VD(24) := ('0','1','0','1','0','0','0','0');  
 VD(25) := ('0','0','0','1','0','0','0','0');  
 VD(26) := ('0','1','0','1','0','0','0','0');  
 VD(27) := ('0','1','0','1','0','0','0','0');  
 VD(28) := ('0','1','0','1','0','0','0','0');  
 VD(29) := ('0','1','0','1','0','0','0','0');  
 VD(30) := ('0','0','0','1','0','0','0','0');  
 VD(31) := ('0','1','0','1','0','0','0','0');  
 VD(32) := ('0','1','0','1','0','0','0','0');  
 VD(33) := ('0','0','0','1','0','0','0','0');  
 VD(34) := ('0','1','0','1','0','0','0','0');  
 VD(35) := ('0','1','0','1','0','0','0','0');  
 VD(36) := ('0','0','0','1','0','0','0','0');  
 VD(37) := ('0','1','0','1','0','0','0','0');  
 VD(38) := ('0','1','0','1','0','0','0','0');  
 VD(39) := ('0','1','0','1','0','0','0','0');  
 VD(40) := ('0','1','0','1','0','0','0','0');  
 VD(41) := ('0','0','0','1','0','0','0','0');  
 VD(42) := ('0','1','0','1','0','0','0','0');  
 VD(43) := ('0','1','0','1','0','0','0','0');  
 VD(44) := ('0','1','0','1','0','0','0','0');  
 VD(45) := ('0','1','0','1','0','0','0','0');  
 VD(46) := ('0','0','0','1','0','0','0','0');  
 VD(47) := ('0','1','1','1','0','0','0','0');  
 VD(48) := ('0','1','0','1','0','0','0','0');  
 VD(49) := ('0','1','1','1','1','0','0','0');  
 VD(50) := ('0','0','0','1','0','0','0','0');  
 VD(51) := ('0','0','0','1','0','0','0','0');  
 VD(52) := ('0','0','0','1','1','0','0','0');  
 VD(53) := ('0','0','0','1','0','0','0','0');  
 VD(54) := ('0','0','0','1','1','0','0','0');  
 VD(55) := ('0','0','0','1','0','0','0','0');  
 VD(56) := ('0','0','0','1','1','0','0','0');  
 VD(57) := ('0','0','0','1','0','0','0','0');

[illegible]

[illegible]

[illegible]



VD(199) := ('0','0','0','1','0','0','1','1');  
 VD(200) := ('0','0','0','1','1','1','1','0');  
 VD(201) := ('0','0','0','1','0','1','1','0');  
 VD(202) := ('0','0','0','1','0','1','1','1');  
 VD(203) := ('0','0','0','1','0','1','1','1');  
 VD(204) := ('0','0','0','1','0','1','0','1');  
 VD(205) := ('0','0','0','1','0','1','1','0');  
 VD(206) := ('0','0','0','1','0','1','1','1');  
 VD(207) := ('0','0','0','1','0','1','1','1');  
 VD(208) := ('0','0','0','1','0','1','0','1');  
 VD(209) := ('0','0','0','1','0','1','0','1');  
 VD(210) := ('0','0','0','1','0','1','0','0');  
 VD(211) := ('0','0','0','1','0','1','0','0');  
 VD(212) := ('0','0','0','1','0','1','0','0');  
 VD(213) := ('0','0','0','1','0','1','0','0');  
 VD(214) := ('0','0','0','1','0','1','0','0');  
 VD(215) := ('0','0','0','1','0','1','0','0');  
 VD(216) := ('0','0','0','1','0','1','0','0');  
 VD(217) := ('0','0','0','1','0','1','0','0');  
 VD(218) := ('0','1','0','1','0','1','0','0');  
 VD(219) := ('0','1','1','1','1','1','0','1');  
 VD(220) := ('0','0','0','1','0','1','0','1');  
 VD(221) := ('0','0','0','1','0','1','0','0');  
 VD(222) := ('0','0','0','1','0','0','0','1');  
 VD(223) := ('0','0','0','1','0','0','0','0');  
 VD(224) := ('0','0','0','1','0','0','0','1');  
 VD(225) := ('0','0','0','1','0','1','0','0');  
 VD(226) := ('0','0','0','1','0','1','0','0');  
 VD(227) := ('0','0','0','1','0','1','0','0');  
 VD(228) := ('0','0','0','1','0','0','1','0');  
 VD(229) := ('0','0','0','1','0','0','1','0');  
 VD(230) := ('0','0','0','1','0','0','1','0');  
 VD(231) := ('0','0','0','1','0','0','1','0');  
 VD(232) := ('0','0','0','1','0','1','0','1');  
 VD(233) := ('0','0','0','1','0','1','1','0');  
 VD(234) := ('0','0','0','1','0','1','1','0');  
 VD(235) := ('0','0','0','1','0','1','1','0');  
 VD(236) := ('0','0','0','1','0','0','1','0');  
 VD(237) := ('0','0','0','1','0','0','1','0');  
 VD(238) := ('0','0','0','1','0','0','1','0');  
 VD(239) := ('0','0','0','1','0','0','1','0');  
 VD(240) := ('0','0','0','1','0','1','1','0');  
 VD(241) := ('0','0','0','1','0','0','1','0');  
 VD(242) := ('0','0','0','1','0','1','1','0');  
 VD(243) := ('0','0','0','1','0','1','1','0');  
 VD(244) := ('0','0','0','1','0','1','1','0');  
 VD(245) := ('0','0','0','1','0','1','0','0');



[illegible]