

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Electronique
Laboratoire des Dispositifs de Communications
et de Conversions Photovoltaïques



Mémoire de Magister En Energies Renouvelables

Option : Photovoltaïque

Présenté par :

MANSOUR Samir

Ingénieur d'état en Électronique

Intitulé

Commande par DSP TMS 320F2812 d'un onduleur MLI destiné à la traction des véhicules électriques

Soutenu publiquement le 20 Juin 2012 devant le jury composé de :

Président :	HADDADI Mourad	Professeur	Ecole Nationale Polytechnique
Rapporteur :	LARBES Chérif	Professeur	Ecole Nationale Polytechnique
Examineurs :	AIT-CHEIKH Md Salah	Maitre de Conférences	Ecole Nationale Polytechnique
	MALEK Ali	Directeur de Recherche	Centre de Développement des Energies Renouvelables

ENP 2012

ملخص:

في منظور تطوير منصدة تجرية لمركبة كهربائية؛ نعرض مشروع انجاز مغير للسرعة. عملنا يرتكز حول دراسة و تصميم محول مستمر- متناوب ثلاثي الأطوار لأجل ضبط سرعة محرك لا متزامن ثلاثي الأطوار. بقصد تحسين أداء نظام التحكم و الاستجابة لمقتضيات الزمن الحقيقي؛ إستراتيجية و عملية توليد إشارات التحكم المعدلة عرضيا (MLI)؛ ستنفذ بواسطة البطاقة الالكترونية DSP TMS 320F2812. إعداد البرنامج المغروز في البطاقة (DSP) قد تم بلغة البرمجة C على البرنامج CCS IDE. بالتوازي مع الانجاز المادي و البرنامجي أنجزنا تطبيقية على البيئة MATLAB لأجل محاكاة و التحقق من فائدة تقنية التعديل الشعاعي (SVM) في التحكم في سرعة محرك لا متزامن ثلاثي الأطوار. في الأخير ففحص حسن اشتغال الجهاز المنجز قمنا بإجراء تجارب عليه.

كلمات مفتاحية: DSP TMS 320F2812؛ محول مستمر- متناوب ثلاثي الأطوار؛ محرك لا متزامن ثلاثي الأطوار؛ تغيير السرعة؛ مركبة كهربائية؛ تعديل الشعاعي لعرض النبضة.

Résumé :

En perspective de développement d'un banc d'essai pour véhicule électrique, on se propose le projet de réalisation d'un variateur de vitesse. Notre travail, consiste en étude et conception d'un onduleur triphasé pour l'asservissement de la vitesse d'un moteur asynchrone triphasé (MAS). En vue d'optimisation des performances du système de commande et répondre aux exigences temps réel, la stratégie de contrôle et la génération des signaux de commande MLI, seront réalisées à l'aide la carte DSP TMS 320F2812. L'élaboration du programme implanté sur le DSP est effectuée en langage C via l'outil logiciel dédié CCS IDE. En parallèle à la réalisation matérielle et l'implantation logicielle, une application sous l'environnement MATLAB est conçue, en vue de simuler et vérifier l'utilité de la technique de modulation vectorielle (SVM) dans la commande en vitesse d'un moteur MAS. A la fin, pour examiner le bon fonctionnement de la réalisation, des tests pratiques sont accomplis.

Mots Clés: DSP TMS 320F2812, Onduleur triphasé, Moteur asynchrone triphasé, Variation de vitesse, Véhicule électrique, Modulation vectorielle.

Abstract:

In prospect of developing a test bed for electrical vehicle, we propose the project for construction of a variable speed drive. Our work, consist of studding and conception of a three phase inverter to control the speed of an induction motor (ASM). In order to improve the performance of the controlling system and reply to the real time requests, the controlling strategy and the generation of PWM command signals, will be done with the DSP device TMS 320F2812. The program implemented on the DSP has been written on C language with the software tool dedicated CCS IDE. In parallel to the hardware construction and software implementation, an application under MATLAB environment is designed, in order to simulate and verify the utility of the space vector modulation (SVM) technique for controlling the speed of an ASM motor. In the last, to test the right operation of the constructed system, practical tests have been accomplished.

Key words: DSP TMS 320F2812, Three phase inverter, Induction motor, speed variation, Electrical vehicle, Space vector modulation.

Remerciements

Je tiens à remercier particulièrement mon promoteur, monsieur LARBES Cherif, professeur, pour son encadrement, sa gentillesse et sa patience avec mon caractère.

Mes sincères gratitude vont à monsieur le professeur HADDAD Mourad, qui nous a servi de sa belle expérience en électronique et qui était toujours présent au moment de besoin.

Je remercie également tous les membres de jury, d'avoir accordé un peu de leurs temps pour examiner et juger ce travail.

Mes remerciement vont aux enseignants, responsables et étudiants de l'école doctorale énergies renouvelables, ainsi que tous ceux qui m'ont de près ou de loin soutenu et encouragé.

Ma reconnaissance et ma gratitude vont à mes parents, pour m'avoir soutenu et permis d'en arriver là.

Samir

Dédicaces

*Je dédie ce travail à
ma mère et mon père à
mes sœurs et frères, à
celle qui m'est chère, à
la grande famille et tous
mes amis.*

Samir

Sommaire

<i>Introduction générale</i>	1
------------------------------------	---

Chapitre I

Introduction au transport électrique

1. Le véhicule électrique	2
1.1 Histoire du véhicule électrique	2
1.2 Les véhicules électriques en utilisation actuellement	4
1.2.1 Les véhicules électriques à batterie	4
1.2.2 Le véhicule hybride (moteur électrique plus moteur CI)	5
1.2.3 Les véhicules électriques à carburant propre.....	5
1.2.4 Les véhicules électriques à lignes d'alimentation	6
1.2.5 Les véhicules à alimentation solaire.....	7
1.2.6 Les véhicules électriques à Flywheels et Super-condensateurs	7
1.3 Architecture du véhicule électrique	9
2. Le véhicule Hybride électrique	12
2.1 Histoire du véhicule hybride électrique	12
2.2 Architectures des systèmes d'entraînement hybrides électriques.....	13
2.2.1 Véhicule hybride électrique série	13
2.2.2 Véhicule hybride électrique parallèle.....	14
2.2.3 Véhicule hybride électrique série-parallèle.....	14
2.3 Le choix entre série, parallèle, ou mixte.....	14
2.4 Les différentes phases de fonctionnements de l'hybride.....	14
2.5 Avantages et inconvénients des systèmes hybrides.....	15
2.6 Technologies pour un hybride réussi	16
3. Perspectives et avenir des véhicules électriques et hybrides.....	17

Chapitre II

Description du DSP TMS 320F2812

1. Généralités sur le DSP.....	19
1.1 Les calculateurs numériques.....	19
1.2 Chronologie brève d'évolution des DSP	21
1.3 Le marché des DSP.....	22
1.4 Le Portfolio des DSP de Texas Instruments	22
1.5 Nomenclature des DSP C2000 de TI	22
2. Description brève du DSP TMS 320F2812	24
2.1 Le Bus mémoire.....	25
2.2 Le bus périphérique.....	26
2.3 L'interface JTAG.....	26
2.4 L'interface externe XINTF	26
2.5 Les mémoires flash et OTP	26
2.6 Les SARAMs (M0 et M1).....	26
2.7 Les SARAMs (L0, L1 et H0).....	26
2.8 La Boot-ROM.....	27
2.9 Le Module de Sécurité.....	27
2.10 Le Bloc PIE (Peripheral Interrupt Expansion)	27
2.11 Les interruptions externes (XINT1, XINT2, XINT13, XNMI).....	27
2.12 Les Horloges des périphériques.....	27
2.13 L'oscillateur OSC et le bloc PLL	27
2.14 Le bloc LPM (Low-Power Modes)	28
2.15 Le bloc Watchdog.....	29

2.16 Les Périphériques	30
2.16.1 Les Timers 0/1/2 de la CPU	30
2.16.2 Les modules Event Manager EVA, EVB	31
2.16.3 Le module ADC	33
2.16.4 Le module eCAN.....	34
2.16.5 Le module McBSP.....	34
2.16.6 Le module SCI.....	34
2.16.7 Le module SPI	35
2.16.8 Les signaux GPIO multiplexés	35
3. Description de l'unité centrale de traitement	36
3.1 Les composants de la CPU	36
3.2 L'architecture de la CPU	37
3.3 Les registres de la CPU.....	38
4. Vue d'ensemble sur les interruptions.....	42
5. Le pipeline.....	43
5.1 Notion du pipeline	43
5.2 Le Pipelining des instructions.....	43
6. Le déroulement du programme.....	45
7. Les modes d'adressage.....	46
8. Les outils logiciels pour DSP	46

Chapitre III Moteur asynchrone et Modulation vectorielle

1. Généralités.....	49
1.1 Organisation du moteur asynchrone	49
1.2 Principe de fonctionnement du moteur asynchrone	51
1.3 Caractéristique du moteur asynchrone.....	52
2. Modélisation du moteur asynchrone.....	53
2.1 Le moteur asynchrone en régime quelconque	53
2.1.1 Transformation triphasé-diphasé $\alpha\beta$	54
2.1.2 La transformation de Park	56
2.1.3 Expressions des tensions	57
2.1.4 Expressions des flux.....	58
2.1.5 Expressions du couple électromagnétique.....	58
2.1.6 Choix du repère dq	59
2.2 Modèle du moteur asynchrone en régime permanent	60
3. Commande du moteur asynchrone	64
3.1 Commande scalaire	64
3.2 Commande vectorielle	65
4. La modulation vectorielle.....	68
4.1 Expressions des tensions phases dans le repère abc	69
4.2 Expression des tensions dans le repère $\alpha\beta$	71
4.3 Principe de la modulation vectorielle	72
4.4 Détermination des temps de modulation T_1	73
4.5 L'implémentation de la SVM	75

Chapitre IV Simulation d'une commande pour moteur asynchrone

1. Présentation brève de MATLAB.....	76
2. Réalisation de la commande à base de la modulation vectorielle.....	78
3. Présentation des blocs de la commande sous Simulink	78
3.1 Le modèle global du système de commande	78

3.2 Détermination de la phase θ	79
3.3 Détermination des composantes V_α, V_β	80
3.4 Détermination du secteur S	81
3.5 Calcul des variables X, Y et Z	82
3.6 Calcul de $t_1, t_2, t_{on}, t_{bon}$ et t_{con}	82
3.7 Calcul des valeurs $cmp1, cmp2$ et $cmp3$	84
3.8 Détermination des séquences de commutation	84
3.9 Génération des impulsions PWM	85
3.10 Le circuit de l'onduleur triphasé avec le moteur asynchrone	86
4. Résultats de la simulation	86
4.1 Signaux de commande	87
4.2 Les signaux de modulation de référence	87
4.3 La tension composée	88
4.4 Les courants des phases de stator	88
4.5 Simulation de la réponse en vitesse	89
4.6 Simulation lors de changement de vitesse	91
4.7 Simulation avec couple mécanique résistant	92
4.8 Simulation lors de l'inversion du sens de rotation	93
4.9 Simulation de la phase d'arrêt du moteur	94

Chapitre V Conception matérielle, implantation logicielle et résultats pratiques

1. Etude et réalisation de la carte électronique	95
1.1 Représentation synoptique du système	95
1.2 Description des différents blocs de la carte	96
1.2.1 La carte eZdsp TMS 320F2812	96
1.2.2 Mécanisme d'introduction de données	97
1.2.3 Le bloc d'alimentation	97
1.2.4 L'interface d'isolation galvanique	98
1.2.5 La partie puissance	99
1.2.5.1 Le circuit driver de transistors de puissance	100
1.2.5.2 Le bras de l'onduleur	103
1.2.5.3 Le circuit Snubber	104
1.2.6 Le processus commandé (MAS)	106
1.2.7 La carte complète	107
2. Implantation logicielle	108
2.1 Les modules utilisés dans l'implantation logicielle	108
2.1.1 Le module EV	108
2.1.2 Le module ADC	110
2.2 Les points principaux de l'implantation logicielle	110
2.3 L'organisation logicielle	111
2.4 L'organigramme global	112
3. Résultats pratiques	114
3.1 Les signaux de commande	114
3.2 Les signaux à la sortie du driver	114
3.3 Les tensions de phases	115
3.4 Les tensions composées	115
3.5 Les courants de phases	117

<i>Conclusion générale</i>	119
----------------------------	-----

Bibliographie

Annexes

Introduction générale

Le secteur du transport électrique constitue un élément clé dans le paysage global des énergies renouvelables et du développement durable, puisque l'énergie pour le ravitaillement et l'alimentation de ce type de transport est supposée d'être produite à partir des sources renouvelables et non-polluantes comme le vent, le soleil et l'hydroélectricité. [37]

Le concept du véhicule électrique VE ou hybride VHE remonte à plus de 100 ans. La première démonstration des véhicules électriques date de l'année 1830, et les véhicules électriques commerciaux sont devenus disponibles sur le marché vers la fin du 19^{ème} siècle. Cependant ces véhicules électriques n'ont pas connu le grand succès des véhicules à moteurs thermiques qui avaient une plus grande autonomie et plus faciles à ravitailler. Mais, avec les sensibilités environnementales grandissantes, les inquiétudes pour l'alimentation à long terme, les prix croissants des carburants fossiles et le développement technologique dans les différents domaines de l'ingénierie automobile, une grande motivation est apparue pour accélérer davantage ce segment de technologie verte. [22]

Actuellement différents modèles et concepts de véhicules électriques et hybrides sont présents sur le marché, grâce aux avancées enregistrées dans le domaine de commande et de contrôle des moteurs électriques, après l'apparition d'une nouvelle génération de calculateurs numériques, très performants pour réaliser ce type d'applications comme le DSP. Ce dernier étant une version optimisée des microprocesseurs, dont chacun de ses composants est absolument conçu pour accomplir les fonctions et tâches du traitement numérique du signal d'une manière plus efficace. Le DSP possède un matériel et des jeux d'instructions optimisés qui lui permettent de faire des traitements numériques, en temps réel, des signaux de l'environnement dans les différents domaines avec des vitesses d'exécution très élevées.

L'objectif de notre travail consiste en commande de la vitesse et du sens de rotation d'un moteur asynchrone à l'aide de la carte DSP TMS 320F2812 de Texas Instrument. Pour cela nous avons procédé à l'étude et la réalisation de la carte électronique constituée d'un onduleur triphasé et d'une interface d'isolation galvanique entre la carte DSP et la partie puissance. On a effectué une étude par simulation sous MATLAB qui nous a permis de mettre en évidence les performances de la technique de modulation vectorielle pour la commande d'un moteur asynchrone triphasé. En suite on a élaboré le programme à implanter sur le DSP en langage C à l'aide du logiciel CCS IDE. En dernière étape, nous avons accompli les essais et tests pratiques sur le moteur asynchrone, commandé par le système électronique réalisé.

Afin de mieux présenter notre travail, nous avons structuré le mémoire en cinq chapitres. Le premier est consacré à la présentation du transport électrique. Le deuxième chapitre décrit brièvement le DSP TMS 320F2812. Dans le troisième chapitre on a donné une description générale du moteur asynchrone et sa modélisation, puis on a introduit le principe de la technique de modulation MLI vectorielle. Dans le quatrième chapitre, est présentée une simulation sur le logiciel MATLAB/Simulink de la commande d'un moteur asynchrone. Le dernier chapitre est structuré en trois parties, dans la première est donnée une description de la carte électronique conçue et réalisée, dans la deuxième partie, est exposé l'aspect logiciel et l'algorithme global implanté sur le DSP, et la dernière partie est consacrée à la présentation des résultats pratiques. A la fin, une conclusion générale, clôt le mémoire.

Chapitre I

Introduction au transport électrique

Le secteur du transport électrique constitue un élément clé, dans le paysage global des énergies renouvelables et du développement durable, puisque l'énergie pour le ravitaillement et l'alimentation de ce type de transport est supposée d'être produite à partir des sources renouvelables et non-polluantes comme le vent, le soleil et l'hydroélectricité. [37]

Ce secteur n'est pas un phénomène récent, en effet, le concept du véhicule électrique (VE) ou hybride (VHE) remonte à plus de 100 ans. La première démonstration des véhicules électriques date de l'année 1830, et les véhicules électriques commerciaux sont devenus disponibles sur le marché vers la fin du 19^{ème} siècle. Durant cette période le véhicule électrique a bien réussi, prenant place à d'autres idées des différentes techniques dans le domaine du transport. Cependant les véhicules électriques n'ont pas connu le grand succès des véhicules à moteurs thermiques, qui avaient une plus grande autonomie et plus faciles à ravitailler. Mais avec les sensibilités environnementales grandissantes, les inquiétudes sur l'alimentation à long terme, les prix croissants des carburants fossiles et les développements technologiques dans les divers domaines de l'ingénierie automobile, une grande motivation est apparue pour accélérer davantage ce segment de technologie verte. [22]

Aujourd'hui, les soucis autour de l'environnement, particulièrement le bruit et les émissions de gaz polluants, peuvent tourner la balance en faveur des véhicules électriques grâce aux progrès technologiques enregistrés dans les systèmes de stockage de l'énergie, comme les batteries à émission nulle et les piles à combustibles. Actuellement différents types et concepts de véhicules hybrides électriques (VHE) et de véhicules électriques (VE), sont présents sur le marché. [37]

1. Le véhicule électrique (VE)

1.1 Histoire du véhicule électrique (VE) [22]

(a) Les premiers jours

Les premiers véhicules électriques sont apparus dans les années 1830, ils utilisaient des batteries non-rechargeables. Un demi-siècle s'est écoulé avant de développer des batteries efficaces pour être utilisées dans des véhicules électriques commerciaux. Vers la fin du 19^{ème} siècle, et avec une production massive des batteries rechargeables, les véhicules électriques deviennent assez largement utilisés.

Au début de 20^{ème} siècle, le véhicule électrique est devenu relativement fiable, et il apparaissait comme de grand concurrent au véhicule conventionnel à moteur thermique durant cette période.

(b) Le déclin relatif après l'an 1910

Durant cette période (après 1910), les véhicules conventionnels à moteur thermique, ont connu un grand succès vis-à-vis des véhicules électriques. Ce succès est dû au grand écart entre l'énergie spécifique du carburant pétrole et celle des batteries. En effet l'énergie spécifique de carburant pétrole varie, mais est autour de

9000 Wh/kg, tandis que l'énergie spécifique des batteries à plomb-acide était autour de 30 Wh/kg.

Pour surmonter le problème du stockage limité de l'énergie électrique par les batteries, d'autres solutions ont été envisagées, Comme le moyen de fournir l'énergie électrique via les lignes d'alimentation (tramway). Ce dernier était largement utilisé durant le 20^{ème} siècle, et permettait l'utilisation d'un transport calme et non-polluant dans les villes.

(c) Les véhicules électriques restaient populaires

Malgré le succès qu'ont connu les véhicules conventionnels, les véhicules électriques restaient populaires, grâce aux avantages qu'ils présentaient. En fait, les véhicules électriques sont calmes et n'émettent pas de gaz polluant, de plus ils possèdent la caractéristique de démarrage-arrêt du moteur pendant la circulation du véhicule tandis que le moteur thermique devient inefficace pour ce type de conduite.

(d) Les développements vers la fin du 20^{ème} siècle

Durant la partie suivante du 20^{ème} siècle, il y avait des changements qui ont pu faire du véhicule électrique une proposition plus attractive dans le transport quotidien. En premier lieu, il y a eu une augmentation dans les préoccupations autour de l'environnement (émissions de dioxyde de carbone et de fumées). Deuxièmement il y a eu des développements techniques dans le design du véhicule et des progrès technologiques dans les batteries rechargeables, les moteurs électriques et contrôleurs électroniques, plus l'apparition des piles à combustibles.



Figure 1.1 Un ancien automobile électrique et un cyclomoteur moderne.

1.2 Les véhicules électriques en utilisation actuellement

Les idées et concepts développés durant les 19^{ème} et 20^{ème} siècles, sont maintenant utilisés pour produire de nouvelles gammes de véhicules électriques, qui commencent à faire un impact salubre dans le champ de l'automobile. Actuellement, on distingue six types principaux de véhicules électriques.

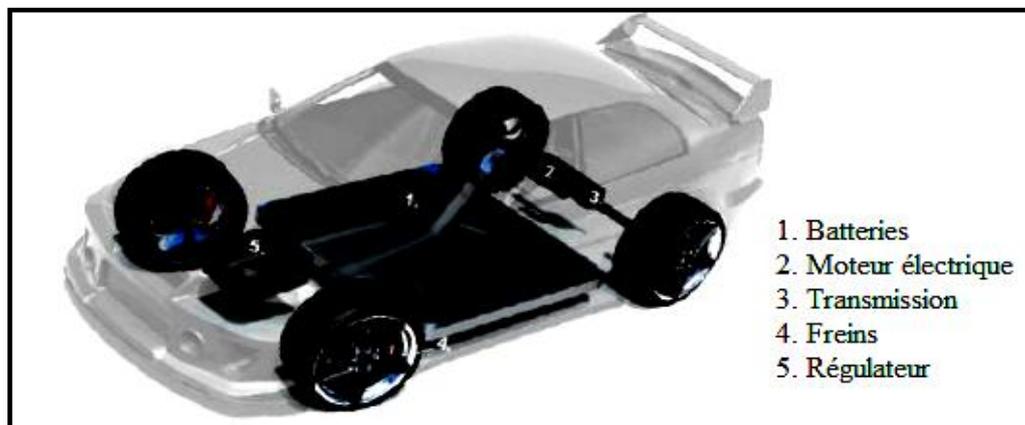
Dans la première place, on trouve le véhicule électrique classique à batterie. Le deuxième type, est le véhicule hybride électrique, qui combine un moteur électrique et un moteur à combustion interne (CI), qui est le type le plus courant dans les premières années. Troisièmement, les véhicules électriques qui utilisent de nouveaux carburants substitutifs comme source d'énergie, qui sont soit des piles à combustible ou des batteries air-métal (air-zinc) à émission nulle. Le quatrième type, les véhicules alimentés par les lignes d'électricité. En Cinquième place, viennent les véhicules électriques utilisant l'énergie reçue directement des radiations solaires. Le dernier type, les véhicules électriques qui utilisent d'autres moyens pour le stockage de l'énergie, comme le volant d'inertie (flywheel) et la super-capacité. [22]

1.2.1 Les véhicules électriques à batterie

Le véhicule électrique à batterie est caractérisé par une propulsion intégralement électrique (absence de moteur thermique). Une gamme de ce type de véhicules électriques sont, actuellement, disponibles sur le marché mais avec des portées et performances limitées. On peut distinguer deux sous-catégories : [10]

- *Les véhicules urbains* : d'une autonomie assez réduite (de 30 à 160 km), ils sont particulièrement destinés aux déplacements urbains sur de courtes distances. L'autonomie peut être augmentée par l'ajout d'un système de récupération d'énergie.
- *Les véhicules à plus grande autonomie* : ils disposent d'une autonomie d'au moins 160 km grâce à une batterie de plus grande dimension.

Le concept du véhicule électrique à batterie, consiste en batterie électrique pour le stockage d'énergie, un moteur électrique, d'un contrôleur et d'une transmission. [22]



1.2.2 Le véhicule hybride (moteur électrique plus moteur CI)

Un véhicule hybride est un véhicule pouvant fonctionner, avec deux ou plus de sources d'énergie différentes. Plusieurs combinaisons de ces sources d'énergies sont possibles. Les types les plus courants, sont les véhicules hybrides qui combinent un moteur à combustion interne (CI) avec un moteur électrique et un générateur. [22]

Il y'a deux arrangement de base pour les véhicules hybrides électriques, le hybride série et le hybride parallèle. Dans l'hybride série, le véhicule est propulsé par un ou plus de moteurs électriques alimentés soit par la batterie, ou par le générateur actionné par le moteur CI, ou par les deux (batterie et générateur). Dans le hybride parallèle le véhicule peut être soit actionné par le moteur CI en liaison directe avec les roues à travers un système de transmission mécanique, soit par un ou plus de moteurs électriques, ou par les deux à la fois (le moteur CI et le(s) moteur(s) électrique(s)). [22]

Dans les deux cas série et parallèle, la batterie peut être rechargée par la collaboration du moteur CI et le générateur pendant le déplacement. Les deux types série et parallèle, permettent aussi le freinage par récupération, où le moteur électrique fonctionne comme générateur pour ralentir le véhicule et recharger la batterie simultanément. [22]

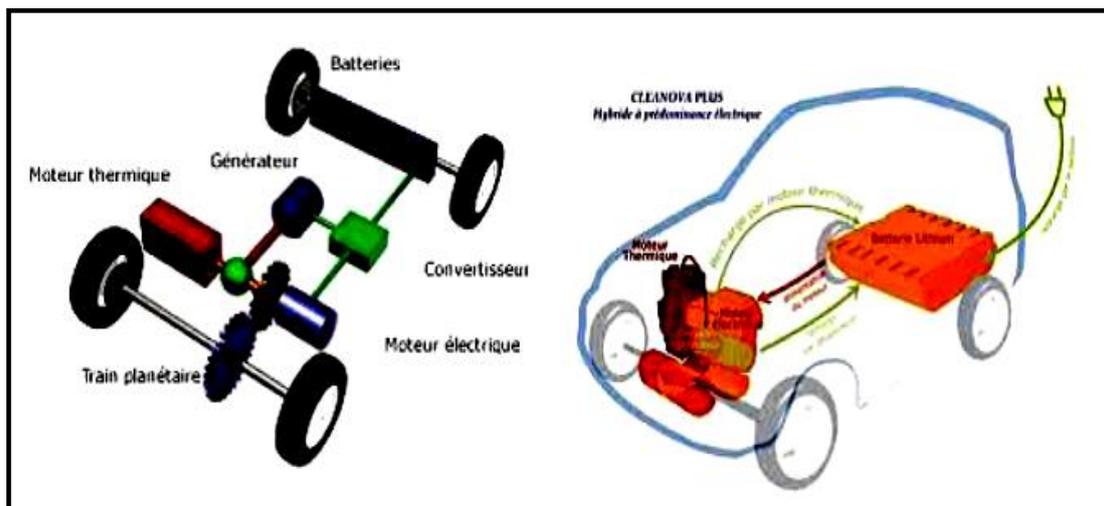


Figure 1.3 Modèles de véhicule hybride électrique avec moteur thermique (CI).

1.2.3 Les véhicules électriques à carburant propre

Le principe de base des véhicules électriques utilisant des carburants propres, est sensiblement le même que celui du véhicule électrique à batterie, mais cette fois-ci, l'énergie vient d'une source non-polluante, comme la pile à combustible et la batterie métal-air (zinc-air) à émission nulle de polluants. Des compagnies majeures (HONDA) dans le domaine de l'automobile ont développé des voitures à pile à combustible très avancées. Les véhicules de services publics comme les bus, présentent une application très prometteuse pour les piles à combustibles et peuvent aussi utiliser de carburants occupant moins de volume et de place comme l'hydrogène. [22]

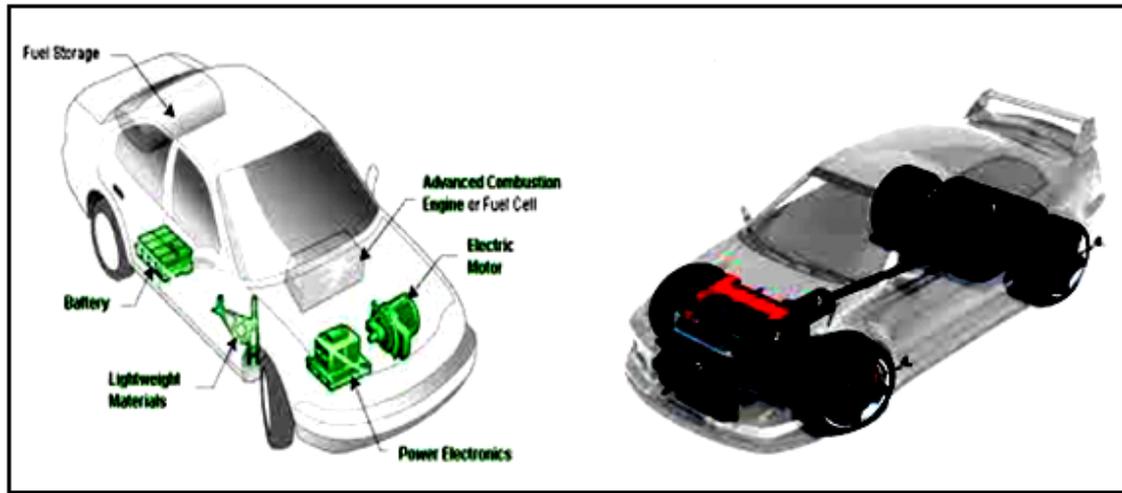


Figure 1.4 Modèles de véhicule utilisant des carburants non polluants.

1.2.4 Les véhicules électriques à lignes d'alimentation

Le tramway, à un certain temps, était largement utilisé dans les villes comme moyen de transport à émissions nulle de gaz polluants. L'énergie électrique est fournie par des lignes d'alimentation au-dessus et une petite batterie montée dans le tramway pour lui permettre une portée limitée sans l'utilisation des lignes d'alimentation. Les véhicules électriques alimentés par des lignes d'alimentation, peuvent faire un impact utile sur le transport moderne, même en maintenant les anciens designs de base utilisés durant les 18^{ème} et 19^{ème} siècles. [22]

Le véhicule connecté au réseau est un concept qui permet d'utiliser l'énergie stockée dans les véhicules électriques afin de soutenir le réseau électrique en période de pic de consommation ou en cas d'urgence (orage, coupure, ...). L'énergie stockée dans la batterie du véhicule pourrait également suppléer aux exigences électriques de l'habitation. Cette technologie nécessite, que le chargeur embarqué dans le véhicule ainsi que l'interface entre le véhicule et le réseau électrique soient bidirectionnels (réversibles), c'est-à-dire, l'énergie électrique doit circuler dans les deux sens (source vers charge et charge vers source). [30]

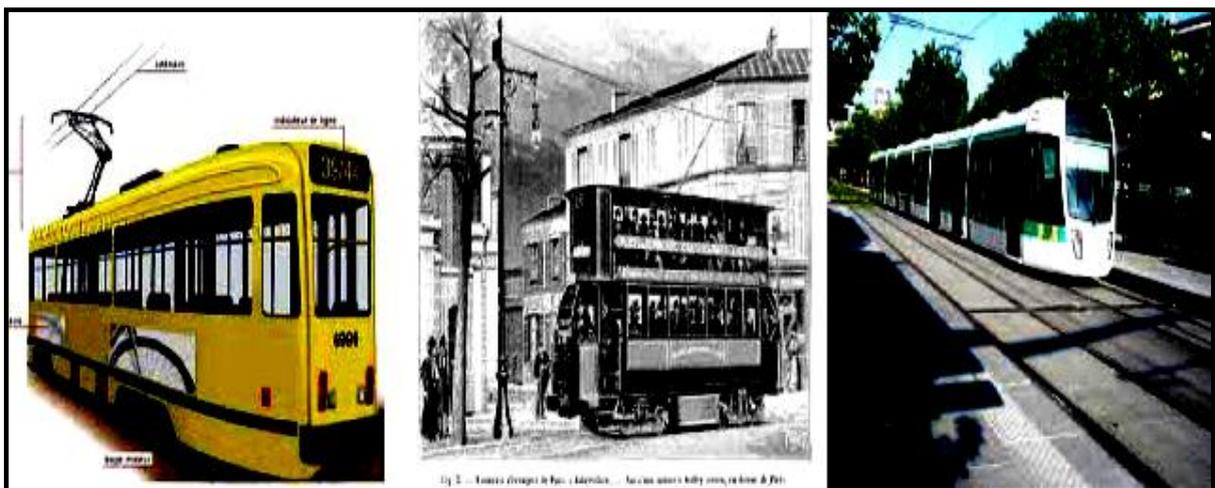


Figure 1.5 Des véhicules électriques utilisant des lignes d'alimentation (Tramway).

1.2.5 Les véhicules à alimentation solaire

Les véhicules à alimentation solaire ne peuvent pratiquement fonctionner que dans des zones hautement ensoleillées, et c'est improbable qu'une voiture de cette nature puisse être une proposition pratique comme voiture à utilisation quotidienne. Ce type de véhicule reste cher, cependant l'efficacité des cellules photovoltaïques s'améliore à tout moment et leur coût ne cesse de diminuer. Ainsi l'utilisation de cellules solaires, enveloppées sur la surface d'un véhicule pour garder les batteries chargées peut devenir faisable et il se peut qu'un jour prouve une proposition pratique pour l'utiliser dans des zones moins ensoleillées comme les villes et renforcer le transport urbain non-polluant. [22]



Figure 1.6 Plusieurs modèles de véhicules électriques équipés de panneaux solaires.

1.2.6 Les véhicules électriques à Flywheels et Super-condensateurs

Suite au développement du transport vert et non-polluant, il y a eu apparition d'autres appareils alternatifs pour le stockage d'énergie dans les véhicules électriques, tels que le super-condensateur et le flywheel (volant d'inertie). Ces deux équipements ont des puissances spécifiques élevées. Cependant, la quantité d'énergie qu'ils peuvent stocker est actuellement très restreinte. [22]

Les super-condensateurs possèdent des puissances spécifiques plus importantes que celles des batteries et des capacités plus grandes que celle des condensateurs classiques. Sa durée de vie est plus élevée que celle des batteries (environ 10 ans). Dans le domaine des transports, les applications envisagées pour les super-condensateurs regroupent les véhicules hybrides, les bus, les tramways, les métros, et les trains. Les super-condensateurs permettent d'améliorer le rendement énergétique des tramways et métros en permettant la récupération de l'énergie lors de la phase du freinage. [20]

Le schéma de la Structure d'un super-condensateur est donné sur la figure ci-après :

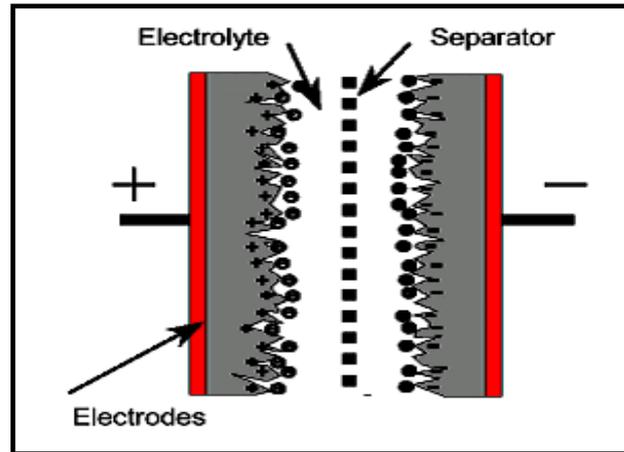


Figure 1.7 Structure d'un super-condensateur.

Par rapport aux super-condensateurs et aux accumulateurs électrochimiques également adaptés aux applications de transport électrique, la technologie de stockage d'énergie par volant d'inertie présente une marge de progression importante en terme de performances au niveau système. Grâce aux avantages qu'il offre, comme sa densité d'énergie, sa fiabilité, sa puissance massique et volumique (moment d'inertie) et sa durée de vie. Le système de stockage inertiel est constitué d'une machine électrique tournante (moteur/générateur) assurant la conversion réversible électrique mécanique de l'énergie, couplée à un convertisseur électronique de puissance assurant le contrôle de la machine et l'adaptation des grandeurs électriques. Une masse tournante couplée mécaniquement à la machine (moteur/générateur) assure l'appoint inertiel permettant le stockage énergétique requis de l'énergie cinétique du véhicule pour l'exploiter ensuite comme énergie électrique. [15]

Le schéma de principe du volant d'inertie (flywheel) est donné ci-dessous :

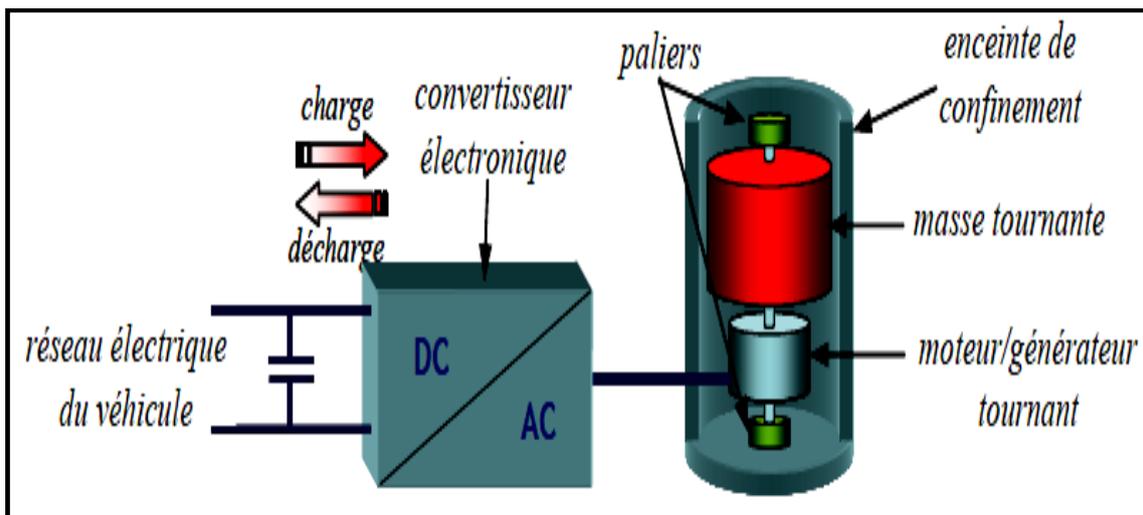


Figure 1.8 Schéma de principe du système de stockage inertiel.

1.3 Architecture du véhicule électrique

Auparavant, la configuration du véhicule électrique était principalement dérivée du concept du véhicule conventionnel à moteur thermique. Ceci en remplaçant, respectivement le moteur thermique et le réservoir de carburant par un moteur électrique et des batteries. Tandis que les autres éléments du véhicule (système de transmission, roues...) conservent leurs structures de base. Cependant ses inconvénients liés principalement à son poids, sa faible autonomie et la dégradation rapide de ses performances ont fait l'échec de ce type de véhicules électriques. [13]

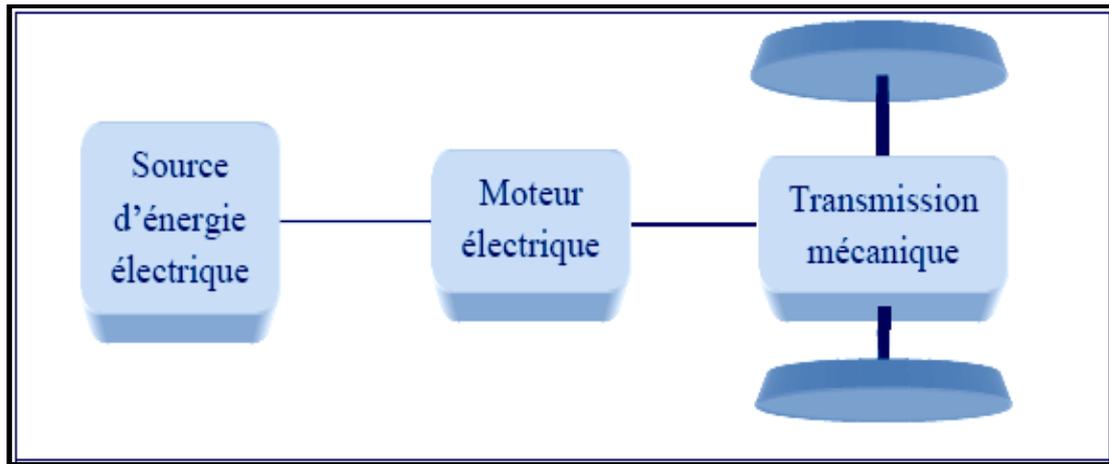


Figure 1.9 Système initial pour l'entraînement d'un véhicule électrique.

Le véhicule électrique moderne est conçu sur la base des designs originaux. Il consiste en trois sous-systèmes majeurs : un système de traction électrique, une source d'énergie, plus des accessoires.

- *Le sous-système de propulsion* : il est composé d'un contrôleur du véhicule, d'un convertisseur électronique de puissance, d'un moteur électrique, d'une transmission mécanique, et d'un volant.
- *Le sous-système de source d'énergie* : il comprend la source d'énergie, l'unité de management d'énergie, et l'unité de ravitaillement en énergie.
- *Le sous-système auxiliaire* : il consiste en unité de gestion et transfert de la puissance, plus une unité de contrôle de la température et une unité d'alimentation auxiliaire.

Le contrôleur du véhicule fournit des signaux de contrôle au convertisseur de puissance, qui permet de gérer le transfert de puissance entre le moteur électrique et la source d'énergie. L'unité management d'énergie coopère avec le contrôleur du véhicule pour contrôler le freinage par récupération et son énergie restituée. Elle collabore également avec l'unité de ravitaillement en énergie afin de contrôler l'utilisation de la source d'énergie. L'alimentation de puissance auxiliaire, fournit la puissance nécessaire avec des niveaux de tension divers pour alimenter tous les accessoires intégrés dans le véhicule électrique.

L'architecture d'un véhicule électrique moderne est illustrée par la figure 1.10.

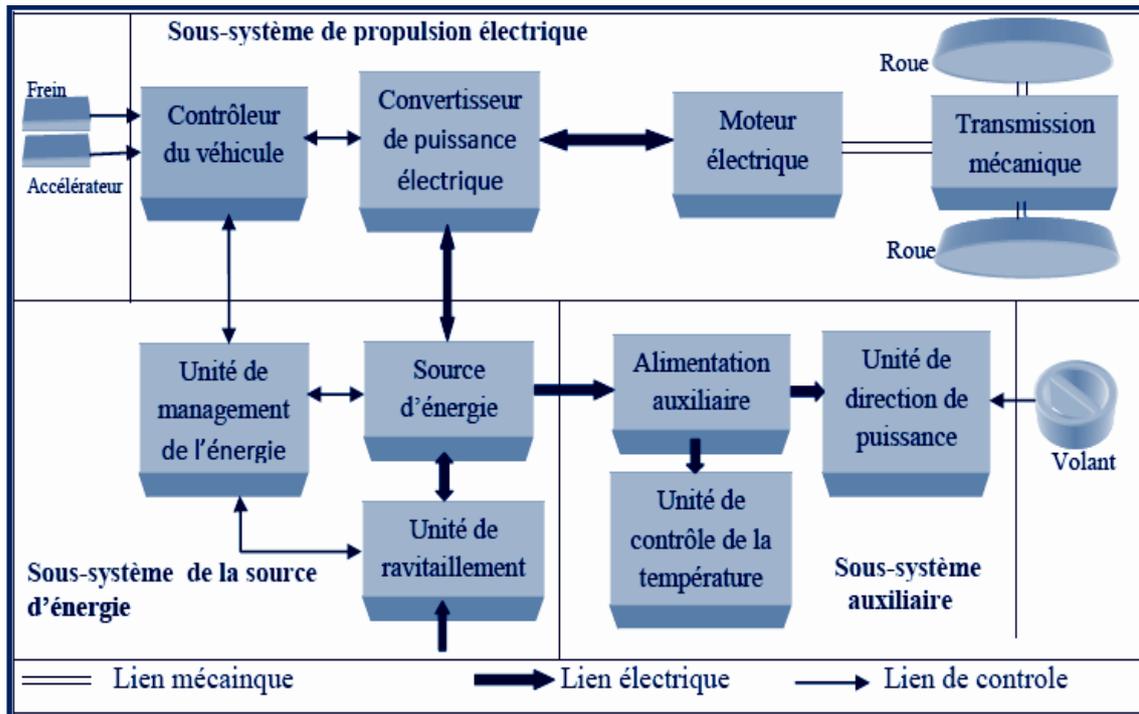


Figure 1.10 Illustration de l'architecture générale d'un véhicule électrique.

Il existe une variété de configurations de véhicules électriques qui sont possibles, en raison des variations des structures possibles pour le système de propulsion électrique, comme le montre la figure 1.11. [13]

- (a) Figure 1.11 (a) montre la configuration de la première variante, dans laquelle un système de propulsion électrique remplace le moteur thermique (CI) d'un véhicule conventionnel. Il est constitué d'un moteur électrique, un embrayage, une boîte de vitesses, et un différentiel. L'embrayage et la boîte de vitesses peuvent être remplacés par une transmission automatique. L'embrayage est utilisé pour faire la liaison entre le moteur et la boîte de vitesse. La boîte de vitesses fournit un ensemble de rapports de démultiplication pour modifier le profil vitesse-puissance (couple) en fonction de la caractéristique de la charge. Le différentiel est un dispositif mécanique (généralement un jeu d'engrenages planétaires), qui permet aux roues des deux côtés d'être entraînées à des vitesses différentes lorsque le véhicule roule sur une trajectoire courbe.
- (b) Avec un moteur électrique fonctionnant à une puissance constante sur une large gamme de vitesse, un système d'engrenage fixe peut remplacer la boîte de vitesse et réduire le besoin à l'embrayage. Cette configuration réduit non seulement la taille et le poids de la transmission mécanique, mais aussi simplifie le contrôle des systèmes d'entraînement puisque le réajustement manuel de la position de la vitesse n'est pas nécessaire.

- (c) Similaire au (b), le moteur électrique, l'engrenage fixe, et le différentiel peuvent être intégrés dans un seul assemblage.
- (d) Sur la figure 1.11 (d), le différentiel mécanique est remplacé par l'utilisation de deux moteurs électriques. Chacun d'eux entraîne la roue de chaque côté et fonctionnent à deux vitesses différentes quand le véhicule roule le long d'une trajectoire courbe.
- (e) Afin de simplifier davantage le système d'entraînement, un moteur de traction peut être placé à l'intérieur de chaque roue. Un jeu d'engrenages planétaire peut être utilisé pour le réajustement de la vitesse transmise aux roues et d'améliorer le couple moteur.
- (f) En abandonnant entièrement toute transmission mécanique entre le moteur électrique et la roue motrice, la sortie du rotor d'un moteur électrique peut être directement reliée à la roue motrice. Le contrôle de la vitesse du moteur électrique est équivalent au contrôle de la vitesse de la roue et donc la vitesse du véhicule. Cependant, cet agencement nécessite un moteur électrique avec un couple élevé au démarrage pour lancer et accélérer le véhicule.

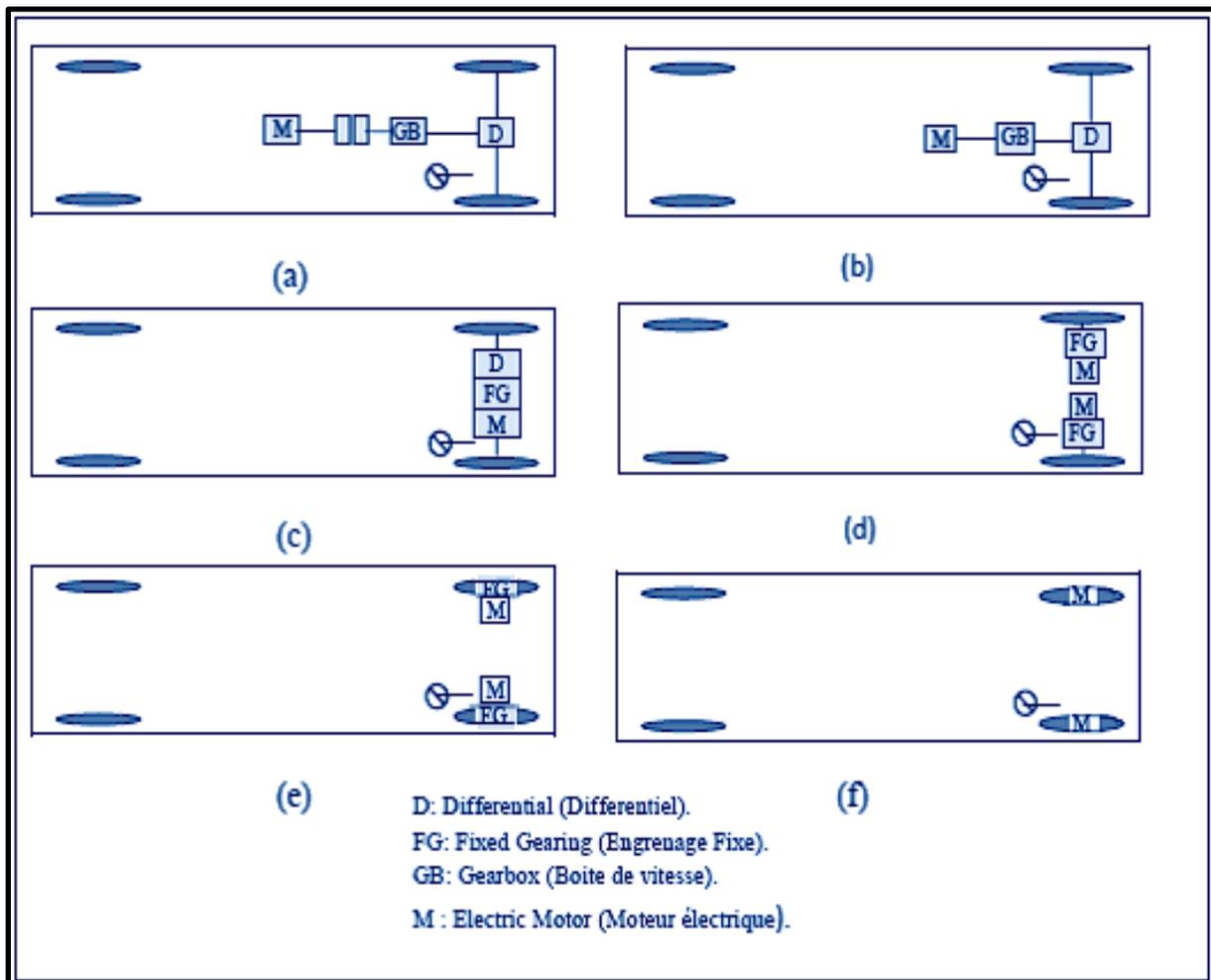


Figure 1.11 Les configurations possibles pour un véhicule électrique.

2. Le véhicule Hybride électrique (VHE)

2.1 Histoire du véhicule hybride électrique (VHE) [22]

En effet, le concept du véhicule hybride électrique est presque autant ancien que l'automobile même. Effectivement, avant l'apparition des véhicules électriques le concept était de développer des véhicules hybrides, dans lesquels un moteur thermique est utilisé conjointement avec un ou plusieurs moteurs électriques. L'objectif principal n'était pas de diminuer la consommation en carburant mais d'assister le moteur thermique pour fournir un niveau acceptable de performance, puisque à l'origine, l'ingénierie du moteur thermique était moins avancée que l'ingénierie du moteur électrique.

Les deux premiers véhicules hybrides rapportés, ont été exposés au salon de Paris en 1899. Le premier était un véhicule hybride parallèle équipé d'un petit moteur essence assisté par un moteur électrique alimenté avec des batteries acide de plomb. L'autre était un véhicule hybride série, qui est un tricycle, dont les deux roues arrière sont actionnées indépendamment par deux moteurs électriques. En 1902, un deuxième véhicule hybride série est apparu. Son design utilise des moteurs DC indépendants dans les roues avant. Une année après (1903), un autre véhicule hybride parallèle a été présenté au salon de Paris. Ce véhicule combine un moteur essence avec une machine électrique fonctionnant comme moteur ou générateur, pour assister le moteur ou charger les batteries. D'autres véhicules hybrides parallèles et séries, ont été conçus durant la période allant de 1899 jusqu'à 1914.

Cependant le problème de ces premiers designs était la difficulté de contrôler les machines électriques. L'électronique de puissance n'était disponible qu'en milieu de 1960 et les premiers moteurs électriques ont été contrôlés par des interrupteurs mécaniques et résistances. De l'autre côté son homologue, le moteur essence a fait d'énormes améliorations dans le terme de densité de puissance, les moteurs thermiques sont devenus petits et plus efficaces, et ils n'ont plus besoin d'être assistés par des moteurs électriques. Ainsi le coût additionnel d'avoir un moteur électrique et les risques associés aux batteries acides ont fait l'objet de la disparition des véhicules hybrides sur le marché après la 1^{ère} guerre mondiale.

En 1975, une nouvelle version de véhicule hybride parallèle a vu le jour. Son moteur CI est assisté par un moteur DC à excitation séparée et couplé à un système de transmission mécanique et des batteries de 12 V ont été utilisées pour le stockage de l'énergie électrique.

Malgré les deux crises de pétrole de 1973 et 1977 et en dépit des préoccupations environnementales, pas de véhicule hybride électrique sur le marché durant cette période. Les recherches ont été concentrées sur le développement des véhicules électriques, dont plusieurs prototypes ont été conçus durant les années 1980 et 1982.

Le concept du véhicule hybride électrique a connu un important impact dans les années 1990 quand c'est devenu clair que les véhicules électriques n'atteindront jamais l'objectif de l'économie d'énergie. L'effort le plus significatif dans le développement et commercialisation des véhicules hybrides électriques est réalisé par les firmes japonaises (Toyota et Honda) à partir de l'année 1997.

2.2 Architectures des systèmes d'entraînement hybrides électriques

Quelques années auparavant, les véhicules hybrides électriques (VHE) ont été classés en deux types de base ; série et parallèle. En 2000, quelques nouveaux designs de véhicules VHE introduits dans le champ du transport électrique ne pouvaient pas être classés dans ces deux types. Depuis cette année, les véhicules VHE sont classés en trois types : hybride série, hybride parallèle, hybride série-parallèle (léger ou complexe). [13]

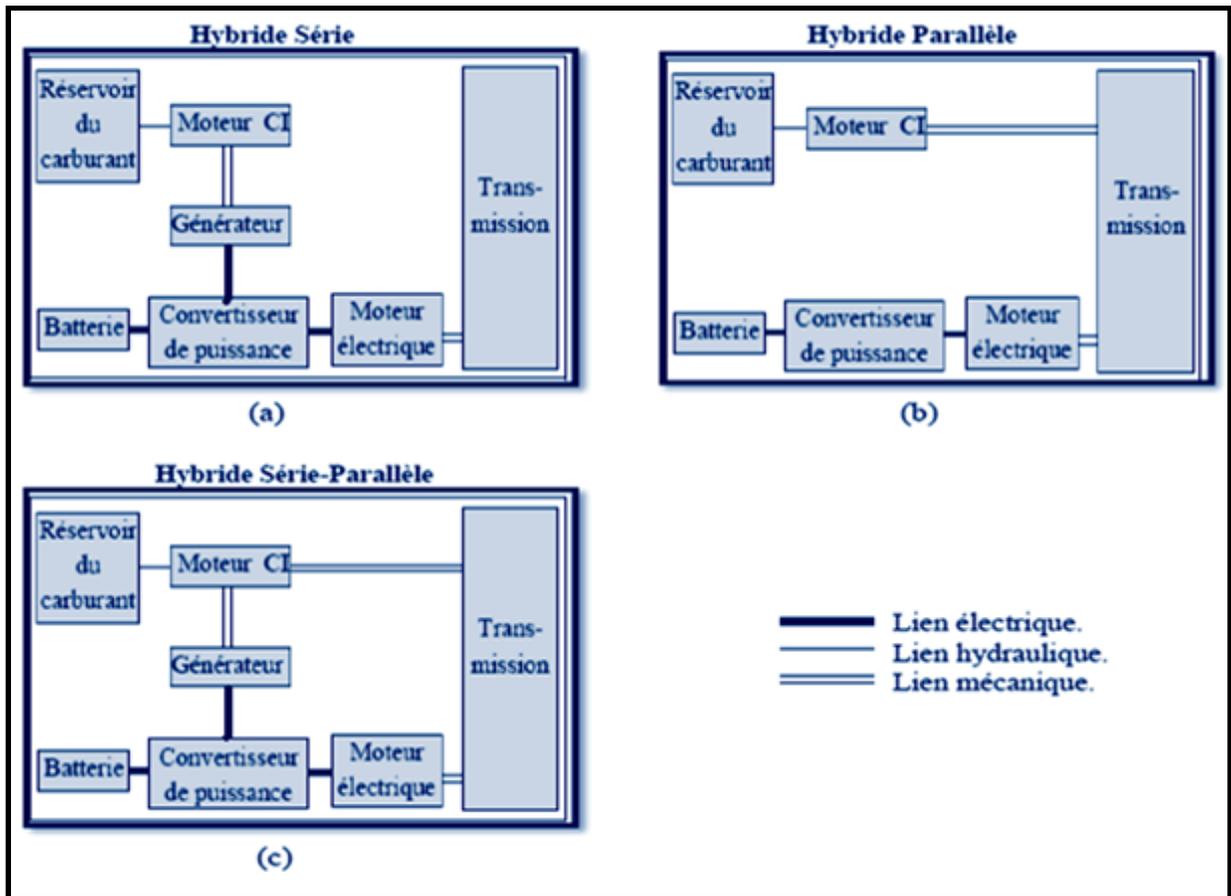


Figure 1.12 Classification des véhicules hybrides électriques.

2.2.4 Véhicule hybride électrique série

Le système hybride série permet au moteur électrique de diriger seul la rotation des roues en utilisant la puissance générée par le moteur thermique. Un hybride série, est composé d'un moteur électrique, d'un moteur thermique (CI), d'un générateur, d'une batterie et d'un convertisseur de puissance. Lors des faibles vitesses, le moteur thermique est utilisé par intermittence pour alimenter soit le moteur électrique, soit pour recharger la batterie. Le tout électrique, est utilisé en ville, sur route le moteur thermique assure la recharge de la batterie ainsi que l'alimentation du moteur électrique. Le nom série vient du fait que le moteur thermique, est directement lié en série au moteur électrique. [10]

2.2.2 Véhicule hybride électrique parallèle

Dans le cas hybride parallèle, les moteurs électrique et thermique assurent tous les deux la rotation des roues, par contre, c'est le moteur électrique qui recharge la batterie. Un hybride parallèle se compose d'un moteur thermique, un moteur électrique, une batterie, un convertisseur de puissance, et d'une transmission. Le moteur électrique, est alimenté cette fois par la batterie et non par le moteur thermique, il fonctionne également en tant que générateur pour recharger la batterie. L'inconvénient est qu'il ne peut produire de l'électricité que durant la conduite. Encore une fois le moteur électrique fonctionne lors des conduites en ville et le moteur thermique lors des plus fortes demandes en puissance soit sur la route. [10]

2.2.3 Véhicule hybride électrique série-parallèle

Le système série-parallèle est la combinaison du moteur électrique et thermique permettant la rotation des roues en rendant de l'électricité à la batterie par l'intermédiaire du générateur. Ce système a l'avantage de pouvoir sélectionner les moteurs soit électrique soit thermique ainsi qu'une recharge permanente des batteries. Il est composé d'un moteur électrique, d'un moteur thermique, un générateur, un répartiteur d'énergie, d'un module de commande l'alimentation (convertisseur de puissance). Un répartiteur, permet de faire fonctionner le moteur électrique ainsi que de produire de l'électricité pour recharger les batteries, ceci tout en essayant d'obtenir le meilleur rendement possible en fonction des conduites (ville, route...). [10]

2.3 Le choix entre série, parallèle, ou mixte

Les deux designs série seul ou parallèle seul, souvent, n'atteignent pas les objectifs et performances souhaités. Le choix du design hybride série, parallèle ou mixte dépend du type de conduite et de la fonction du véhicule, par exemple, taxi, camion ou bus.... L'utilité du design série ou parallèle est devenue moins significative. Les designs mixtes, plutôt que série et parallèle, offrent plus de flexibilité. L'autre facteur de sélection entre les différents designs est le niveau de la demande en puissance (basse, moyenne, haute). [18]

2.4 Les différentes phases de fonctionnements de l'hybride

Le but recherché est une consommation économique du carburant, tout en gardant les mêmes caractéristiques d'une cylindrée "normale". Pour cela les ingénieurs et constructeurs ont déterminé différentes phases (démarrage, vitesse de croisière, freinage = récupération d'énergie) pour le fonctionnement d'un système hybride. La technologie hybride permet un usage sélectif des moteurs thermique et électrique pour limiter la consommation en carburant, ainsi une baisse de la pollution atmosphérique. [10]

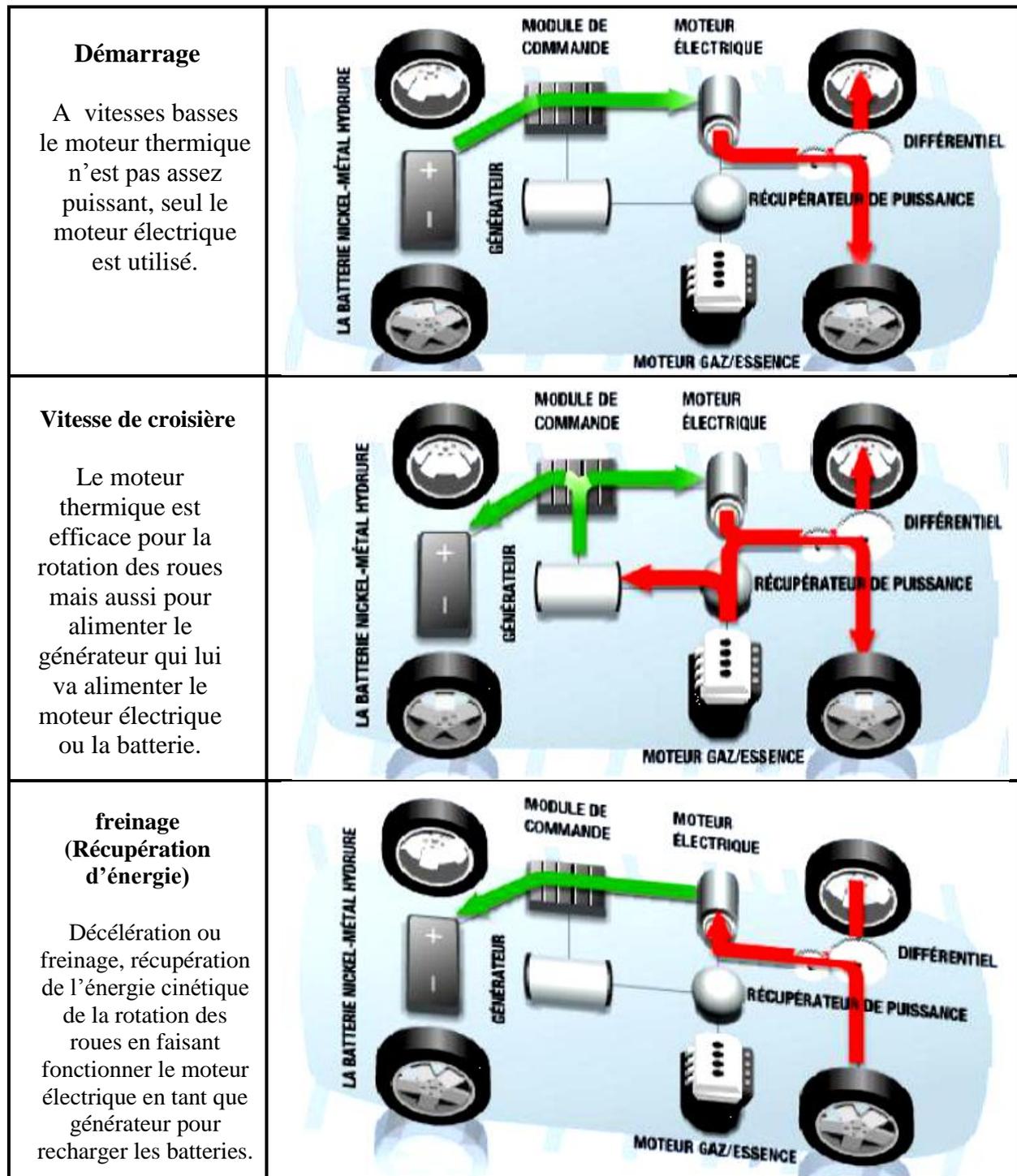


Figure 1.13 Le fonctionnement suivant les phases (démarrage, vitesse de croisière, freinage).

2.5 Avantages et inconvénients des systèmes hybrides

Nous trouvons de nombreux avantages pour les systèmes hybrides, mais certes quelques points restent tout de même à améliorer. [10]

A. Avantages :

- L'économie de carburant, ce qui nous mène aux faibles émissions polluantes.
- Une conduite agréable et performante, et bruit réduit (voitures quasi silencieuses).
- La souplesse au démarrage grâce à l'utilisation du moteur électrique.
- La conduite est agréable avec une accélération linéaire et puissante
- Consommation faible de carburant grâce à la combinaison intelligente des moteurs électrique et thermique.
- Les systèmes hybrides ont une très bonne autonomie. Les batteries se rechargent lors de l'utilisation du moteur thermique.
- Le système de freinage veille aussi à la recharge des batteries car une partie de l'énergie cinétique est transformée en énergie électrique.
- le crédit d'impôt pour l'achat d'un véhicule propre va augmenter, tandis que les propriétaires d'automobiles très polluantes paieront leur carte grise plus cher.

B. Inconvénients :

- Le prix d'un véhicule hybride reste bien élevé comparé aux véhicules "classiques".
- Hors ville, le moteur électrique est peu ou pas utilisé.
- L'utilisation de pièces supplémentaires, comme la batterie hybride, en font une voiture plus lourde.
- Enfin, le bruit quasi-silencieux, émis par un véhicule hybride, peut s'avérer dangereux pour les piétons qui n'entendraient pas le véhicule arriver.

2.6 Technologies pour un hybride réussi

Les véhicules hybrides sont d'une architecture complexe et plusieurs technologies sont impliquées pour assurer leur réussite. [18]

- **Biologie:** Développement de l'utilisation des biogaz non polluants.
- **Mécanique:** Développer des structures avec un coût bas et un poids léger.
- **Électromécaniques:** Permet d'améliorer l'efficacité des moteurs et générateurs.
- **Electrochimie:** Développer des batteries, ultra capacités, et piles à combustible.
- **Electroniques:** Ce domaine est intéressant pour améliorer l'efficacité des transistors de puissance et le fonctionnement en sécurité à des températures élevées. Améliorer les capacités des calculateurs et leurs vitesses de traitement.
- **Logiciel:** Développer des outils de simulations de véhicule hybrides. Développement d'algorithmes intelligents pour le contrôle des véhicules hybrides.

3. Perspectives et avenir des véhicules électriques et hybrides

La mutation du paysage automobile va se poursuivre sous l'impulsion de nombreux programmes gouvernementaux, et grâce à l'arrivée sur le marché d'une multitude de nouveaux véhicules complétant les gammes actuelles. Des initiatives rassemblant des producteurs d'énergie, des instances gouvernementales, le monde de la recherche, des usagers, des fabricants de batteries, les constructeurs, voient le jour un peu partout sur la planète. Les nombres, de quelques milliers à plusieurs millions d'unités, indiquent que nous assistons véritablement à un changement d'échelle du marché des véhicules électriques et hybrides. L'impact de certains programmes nationaux sur les capacités de production va permettre l'émergence de nouveaux acteurs sur le marché international. [4]

La Chine prévoit d'alimenter son gigantesque marché intérieur d'un pourcentage élevé de véhicules électriques et hybrides. Après avoir lancé en 2007 un vaste programme de R&D rassemblant des universités, des instituts de recherche et des industriels, « l'initiative 863 », le gouvernement chinois a impulsé en 2008 un programme de démonstration à grande échelle. Ce programme concerne des milliers de véhicules et repose sur la mise en place d'une infrastructure de charge destinée aux véhicules électriques dans les plus grandes villes chinoises. Pour donner suite à ce démonstrateur géant, des fonds importants sont mobilisés, afin de construire un vaste réseau de stations électriques à l'échelle du pays. [4]

Au Japon, le cabinet du premier ministre a affirmé qu'en 2020, la moitié des véhicules commercialisés seront propulsés par d'autres ressources que des carburants fossiles. Il encourage les utilisateurs de véhicules électriques par des aides substantielles et la conversion en électrique des grandes flottes d'entreprises. Ce sera le cas de la Poste japonaise et ses 21 000 véhicules. Le gouvernement supporte un programme d'installation de centaines de stations de charge et rassemble autour de son action tous les industriels concernés, producteurs d'énergie, constructeurs et fournisseurs de batteries. [4]

Les annonces de nouveaux véhicules effectuées par les industriels, nouveaux entrants ou majors constructeurs traditionnels, sont de plus en plus nombreuses et suivies d'effets. Pour le premier marché mondial, aux USA, les géants de l'automobile de Detroit, GM, Ford et Chrysler, préparent leur reconversion vers les électriques et hybrides. [4]

En France le scénario de développement retenu par les études est conforme aux objectifs du plan gouvernemental, soit 2 millions de véhicules électriques à l'horizon 2020. Il a été construit sur la base d'une pénétration progressive de ces véhicules dans le marché automobile, lente en début de période, puis, s'accéléralant à partir de 2015/2016 : des hypothèses de 7,5 % des ventes en 2016 et 30 % des ventes en 2020. [12]

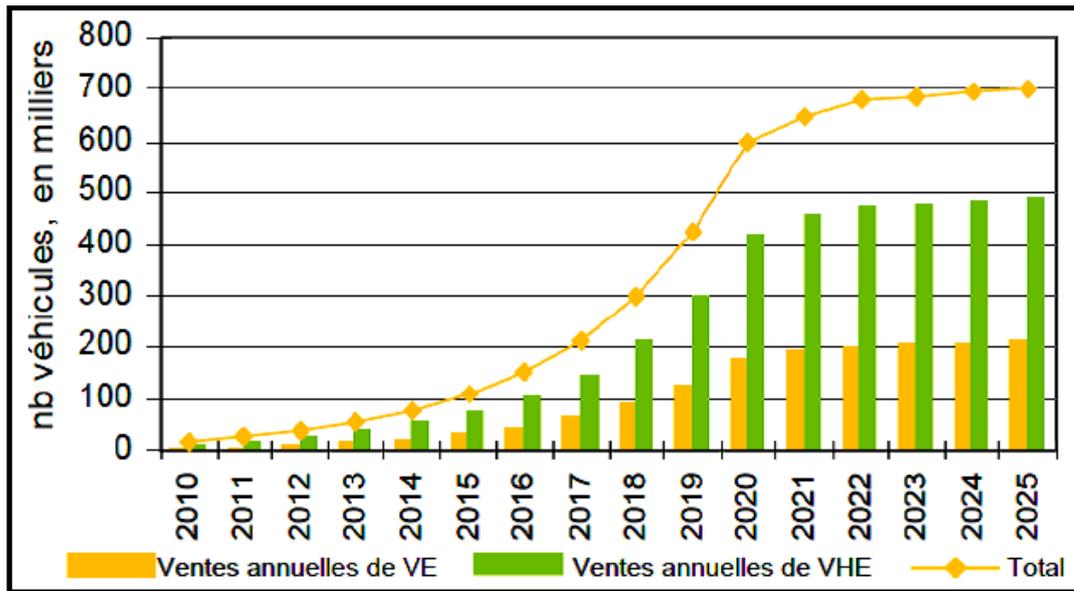


Figure 1.14 Scénario de développement du marché des véhicules électriques.

Conclusion

L'humanité devient de plus en plus concernée autour du dommage qu'elle cause à l'environnement, et les véhicules électriques ont été reçus pour jouer une partie dans le redressement de la balance. Il est donc important que l'impact environnemental des véhicules électriques soit complètement compris. Enfin de compte les véhicules électriques, peuvent être un bénéfice considérable et consistant, réduisant les émissions nuisibles.

Le futur des véhicules électriques, bien sur, reste à écrire. Cependant, le besoin aux véhicules qui minimisent les dommages sur l'environnement est urgent. Beaucoup de technologies pour produire de tels véhicules étaient développées. Leur coût élevé dans plusieurs cas, actuellement, chutera probablement avec l'augmentation de la demande, qui permettra une production en grande quantité. [22]

Pour commander et contrôler les moteurs dans les véhicules électriques l'utilisation des processeurs DSP (Digital Signal Processing) présente l'excellent choix, puisque les DSP offrent la combinaison d'une CPU très puissante en calculs mathématiques avec des modules et périphériques variés (PWM ADC, SPI...) pour la réalisation des tâches de contrôles et de communications.

L'utilisation des DSP C2000 de Texas Instruments tel que le TMS 320F2812 utilisé dans notre travail, qui traite l'information en temps réel, permet de concevoir des applications en un seul appareil DSP et d'implanter des boucles de contrôle de plusieurs grandeurs physiques avec des fréquences de traitement très élevées. Il permet de réaliser des stratégies de contrôle complexes et flexibles qui peuvent être gérées avec une grande efficacité, de plus le DSP a la capacité d'effectuer des diagnostics et détecter des anomalies et problèmes de fonctionnement. [37]

Chapitre III

Description du DSP TMS 320F2812

En effet le DSP est juste une forme spéciale de microprocesseur. Il comporte toutes les caractéristiques et composants de base, tels que la CPU, la mémoire, les jeux d'instruction, les bus, etc..... La principale différence est que chacun de ces composants est relativement conçu pour accomplir certaines fonctions et tâches plus efficacement. En général le DSP possède un matériel et des jeux d'instruction qui sont optimisés pour le traitement numérique en temps réel des signaux de l'environnement des différentes applications à des vitesses élevées. [28]

1. Généralités sur le DSP

1.1 Les calculateurs numériques

En informatique, pour spécifier un type donné d'appareil électronique qui fait du traitement numérique, sont utilisés des mots comme Microprocesseur (μP), Microordinateur, ou Microcontrôleur (μC). Dans le cas du traitement numérique du signal, les noms préférés sont le nom Digital Signal Processors (DSP), et le nom Digital Signal Controller (DSC). [35]

A. Microprocesseur (μP)

- **Le μP est l'élément central d'un système microordinateur multi puce :** Les μP sont basés sur une approche de traitement séquentielle : Lire la prochaine instruction du code machine depuis la mémoire programme, décoder l'instruction, lire les opérandes depuis la mémoire des données, exécuter l'instruction, écrire et sauver le résultat retourné.
- **Le μP est caractérisé par deux architectures de base :**
 - ✓ Architecture Von Neumann, caractérisée par :
 - Un espace mémoire partagé entre le programme et les données
 - Des bus mémoire partagés entre le programme et les données
 - ✓ Architecture Harvard, caractérisée par :
 - Deux espaces mémoire indépendants pour le code et les données
 - Deux systèmes de bus mémoire pour le code et les données
- **Un μP nécessite des appareils externes :** Pour l'utilisation d'un μP , on doit ajouter de la mémoire et des appareils (cartes électroniques) externes au μP .

B. Le Microordinateur

On a besoin d'ajouter des mémoires externes au μP pour stocker les instructions (code) et les données à traiter. On doit aussi utiliser quelques sortes de connexions vers le monde extérieur (périphériques), qui sont généralement, groupés en entrées/sorties numériques et entrées/sorties analogiques. Après l'ajout d'appareils externes au μP , on finit avec l'équipement d'un système ordinateur.

$$\text{Micro Ordinateur} = \text{Microprocesseur } (\mu P) + \text{Mémoires} + \text{Périphériques}$$

C. Le Microcontrôleur μC (un ordinateur sur une seule puce)

Au fur et à mesure des avancées technologiques, l'industrie de silicium cherche à construire tout ce qui est nécessaire pour un microordinateur sur une seule puce de silicium, ce qui conduit à la structure d'un microcontrôleur (μC). Aujourd'hui, des microcontrôleurs sont utilisés dans presque chaque produit industriel disponible sur le marché. Un produit sophistiqué comme un véhicule moderne est équipé de plus de 80 μC pour exécuter les nouvelles fonctions électroniques comme le frein ABS, le verrouillage central, le contrôle adaptif de la vitesse de croisière, les miroirs électriques et les ajustements des sièges, etc.... Les Microcontrôleurs sont disponibles sur 4, 8, 16, 32 ou même 64-bit, sachant que ces nombres indiquent les quantités des bits d'un opérande qui sont traités en parallèle. Les deux architectures Von Neumann et Harvard sont utilisés dans les μC . Le μC présente une solution puissante pour des applications de systèmes de contrôle embarqués, et est le pilier presque de tout les produits modernes, et il y a plus de 200 familles indépendantes de μC sur le marché.

D. Le DSP (Digital Signal Processor)

Un DSP est un appareil spécifique qui est conçu autour d'un noyau qui réalise des opérations mathématiques typiques pour traiter les données numériques mesurées par des capteurs de signaux. L'objectif du DSP est de traiter les données, le plus rapidement possible, pour générer la sortie de la nouvelle donnée en temps réel. Le DSP est similaire au μP , et il est caractérisé par :

- Unité(s) matérielle(s) supplémentaire (s) de multiplication (Multiplieur)
- Des systèmes de bus supplémentaires pour des accès parallèles
- Matériel supplémentaire tel que le registre à décalage (Shifter)

E. Le DSC (Digital Signal Controller)

Un DSC (Digital Signal Controller) est un nouveau type de microcontrôleur, où la fonction de traitement est menée par un DSP. Le DSC est un appareil sur une seule puce combinant la puissance de calcul d'un DSP et les périphériques intégrés d'un microcontrôleur. Pour des systèmes de contrôle temps réel avancés avec une quantité élevée de calculs mathématiques, un DSC est présenté comme le premier choix. Aujourd'hui, il y a peu de manufactures qui fabriquent des DSC. Vu les avantages des DSC dans plusieurs projets, des manufactures de silicium se sont lancées dans le développement de ce type de contrôleurs. Dans ce type de contrôleur on trouve la famille C2000 de Texas Instruments.

1.2 Chronologie brève d'évolution des DSP [28]

L'une des premières contributions apportées dans le domaine du traitement numérique du signal était l'utilisation des puces en silicium TTL (MSI : Medium Scale Integration). Plus de 100 de ces puces étaient utilisées pour former les sections de l'unité ALU et des multiplieurs autonomes. Mais ces anciens systèmes étaient larges, chers et chauds, ce qui a suscité de courir à la recherche d'autres solutions.

La première solution de DSP à une seule puce est apparue en 1982. C'était le TMS32010 de Texas Instruments. Pas long temps après, vient le uPD7720 de NEC. Ces processors avaient des performances qui atteignaient 5 MIPS et avaient des mémoires RAM très petites, ce qui a réduit la consommation et le cout des puces à un prix d'environ \$600.

Sur le marché les appareils DSP continuèrent à se développer, les constructeurs ont commencé d'ajouter plus de matériels intégrés; RAM, ROM, et EPROM internes. Des ports séries pour des transferts rapides des données ont été joints. D'autres améliorations architecturales pour cette deuxième génération ont inclus des timers, des mémoires à accès direct (DMA), des systèmes d'interruptions, et des convertisseurs analogiques numériques (ADC).

Quelques années plus tard, les premiers DSPs à virgule flottante caractérisé par leur simplicité de programmation sont apparus. Il y'avait le DSP32 de AT&T et le DSP TMS320C30 de Texas Instruments qui sont introduits sur le marché durant la même période, vers la fin des années 1980 (en 1988). A cause de la zone large en silicium nécessaire pour supporter l'architecture à virgule flottante, ces appareils coutaient plus que les processeurs traditionnels à virgule fixe.

En début de 1990, une structure DSP à traitement parallèle a commencé d'émerger. Les DSPs avec un support de communication très avancé, comme le TMS320C40 de TI est apparu. Des éléments de traitement multiples étaient conçus dans un seul circuit intégré comme dans le cas du DSP TMS320C80 de Texas Instrument.

Actuellement, plusieurs styles d'architecture avancés de DSP existent sur le marché et dans différents domaines (Télécommunication, automatique, véhicules électrique, etc...). Les progrès et améliorations ont inclus des unités fonctionnelles multiples, des architectures (VLIW : Very Long Instruction Word) et des unités fonctionnelles spéciales pour accomplir plusieurs taches simultanément et rapidement.

1.3 Le marché des DSP

Il y a peu de joueurs sur le marché dans le domaine de DSP et DSC, et parmi eux on trouve Texas Instruments, Lucent (Agere Systems), Analog Devices, Motorola. Texas Instruments est considéré comme le leader absolu dans le domaine du DSP sur le marché.[16]

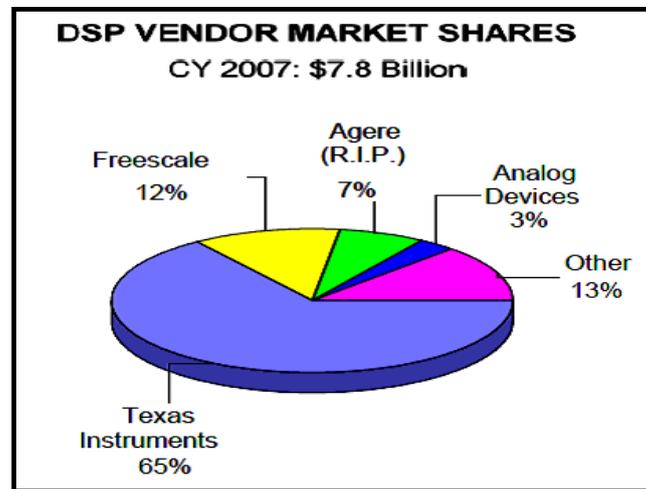


Figure 2.1 Répartition du marché du DSP en 2007.

1.4 Le Portfolio des DSP de Texas Instruments

Le portfolio DSP/DSC Texas instruments, est divisé en trois grandes familles, et qui sont : C2000, C5000 et C6000. [35]

- La branche C6000 est la plus puissante dans la capacité de calcul. Dans cette famille, il y a des appareils à virgule fixe et à virgule flottante. Les champs d'applications sont le traitement d'image, audio, serveur multimédia, communications sans fils.
- La famille C5000 est focalisée sur les systèmes portables avec une consommation de puissance très efficace par MIPS. Ses principaux domaines d'application sont la technologie de téléphone cellulaire, modems, internet, caméras numériques.
- Le groupe C2000 est dédié au contrôle numérique du signal, et est une très puissante solution pour des applications de contrôle temps réel (Contrôle de moteur, stockage d'énergie, systèmes de contrôle numérique).

1.5 Nomenclature des DSP C2000 de TI

Afin de développer facilement nos applications sur le DSP, la société Texas Instruments (TI) offre une gamme large d'outils de programmation et de développement pour la génération des DSP 320X28x, incluant des instruments pour évaluer la performance des processeurs et la génération de code. Des modules matériels et logiciels pour l'émulation et le débogage des programmes sont offerts. [39]

Pour désigner les différentes étapes dans le cycle de développement de l'appareil DSP, TI assigne des préfixes avant la partie chiffres de tous les DSP 320X28x. Chaque membre de la famille des DSP 320X28x possède un des trois préfixes : TMX, TMP, ou TMS. Pour la désignation des outils de développement, Texas Instruments recommande deux des trois préfixes possibles : TMDX et TMDS. [39]

Ces préfixes représentent les étapes évolutives du cycle de développement du produit DSP, à partir des types (TMX/TMDX) jusqu'à la production intégrale qualifiée des appareils/outils (TMS/TMDS).

- TMX : Appareil expérimental qui n'est pas nécessairement représentatif des spécifications électriques et techniques du produit (DSP) final.
- TMP : Produit final qui est conforme aux spécifications électriques de l'appareil DSP, mais qui n'a pas complété toutes les vérifications de qualité et de fiabilité.
- TMS : Appareil intégralement qualifié.
- TMDX : Support de développement qui n'a pas encore complété les tests internes de qualification au niveau de Texas Instruments.
- TMDS : Support de développement intégralement qualifié.

La nomenclature de l'appareil TI inclue aussi un suffixe avec le non de sa famille. Ce suffixe indique le type du boîtier (exemple : PBK) et la gamme de température (exemple: A).

La figure suivante fournit une légende pour la lecture complète du nom de l'appareil DSP pour tous les membres de la famille DSP 320x281x.

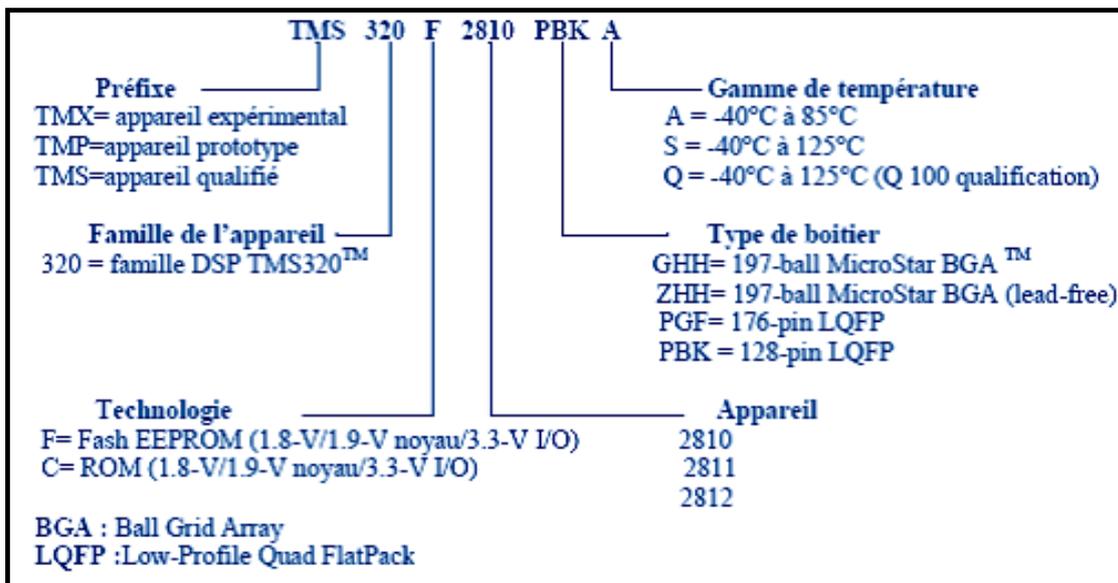


Figure 2.2 Nomenclature des DSP 320x281x.

Dans la suite de ce chapitre, on va s'intéresser particulièrement aux caractéristiques du DSP TMS 320F2812 de Texas Instrument.

2. Description brève du DSP TMS 320F2812 [39]

Le TMS320F2812 est un DSP autant efficace en tâches mathématiques qu'en tâches de contrôle de systèmes qui sont typiquement gérés par des microcontrôleurs. En plus, de la réponse rapide aux interruptions avec des sauvegardes automatiques des registres critiques, le DSP est capable d'entretenir plusieurs événements avec une latence minimale. Le TMS 320F2812 est un DSP à virgule fixe avec une vitesse de traitement de 150 MIPS, caractérisé par une architecture Harvard avec huit (8) phases pipeline, et des bus programme et data indépendants, ce qui lui permet de réaliser des accès parallèles aux mémoires programme et data simultanément, ce qui réduit énormément le temps d'exécution des programmes. [39]

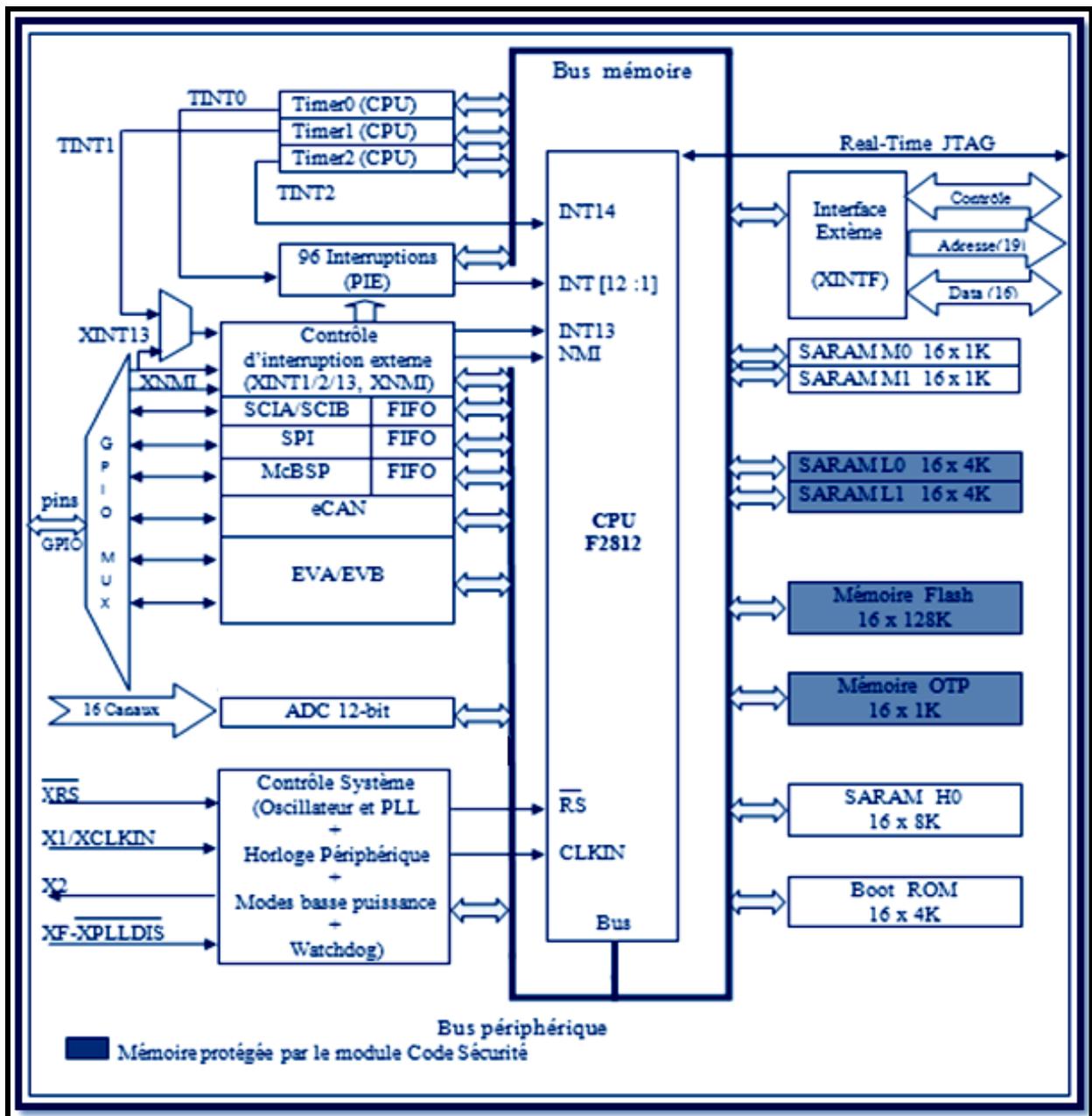


Figure 2.3 Diagramme du bloc fonctionnel du DSP TMS 320F2812.

Adresse de début du bloc	Mémoire interne (ON-chip)		Mémoire externe XINTF		Adresse de début du bloc		
	Espace Data	Espace programme	Espace Data	Espace programme			
0x00 0000	Vecteur-RAM M0 (32 x32) (Validé si VMAP=0)		Réservée				
0x00 0040	SARAM M0 (16 x 1k)						
0x00 0400	SARAM M0 (16 x 1k)						
0x00 0800	Peripheral Frame 0	Réservée					
0x00 0D00	Vecteur-RAM PIE (16 x256) (Validé si VMAP=1, et ENPIE=1)						
0x00 0E00	Réservée						
0x00 2000	Réservée		Zone0 XINTF (16 x8K, XZCS0AND1)	0x00 2000			
			Zone1 XINTF (16 x8K, XZCS0AND1) (protégée)	0x00 4000			
0x00 6000	Peripheral Frame 1 (Protégé)	Réservée	Réservée				
0x00 7000	Peripheral Frame 2 (Protégé)						
0x00 8000	SARAM L0 (16 x4k)						
0x00 9000	SARAM L1 (16 x4k)						
0x00 A000	Réservée					Zone1 XINTF (16 x0.5M, XZCS2)	0x08 0000
						Zone2 XINTF (16 x0.5M, XZCS6AND7)	0x10 0000
0x3D 7800	Mémoire OTP (16 X 1K)		Réservée		0x18 0000		
0x3D 7C00	Réservée (1K)						
0x3D 8000	Mémoire Flash (16 x128K)						
0x3F 7FF8	Mot de passe 128-bit						
0x3F 8000	SARAM H0 (16 x8K)						
0x3F A000	Réservée						
0x3F F000	Boot-ROM (16 x4k) (Validée si MP/MC=0)					Zone7 XINTF (16 x16K, XZCS6AND7) (Validée si MP/MC=1)	0x3F C000
0x3F FFC0	Vecteur-RMO BROM (32 x32) (Validé si VMAP=1, MP/MC=0, ENPIE=0)					Vecteur-RAM XINTF (32 x32) Validé si VMAP=1, MP/MC=1, ENPIE=1	

Figure 2.4 Organisation de la mémoire du DSP TMS 320F2812.

Le DSP TMS 320F2812 intègre les fonctionnalités suivantes : [39]

2.1 Le Bus mémoire

Comme plusieurs types de DSP, de multiples bus sont utilisés pour transférer les données entre les mémoires, les périphériques et la CPU. L'architecture du bus mémoire du DSP TMS 320F2812 comporte un bus pour la lecture du programme, un bus pour la lecture des données et un bus pour l'écriture des données. Le bus de lecture du programme consiste en 22 lignes d'adresses et 32 lignes de données. Les bus de lecture et écriture des données consistent en 32 lignes d'adresses et 32 lignes de données chacun. La largeur 32-bit du bus de données permet de réaliser des opérations 32-bit en un seul cycle.

2.2 Le bus périphérique

Le DSP 320F2812 intègre un bus périphérique standard pour l'interconnexion périphérique. Le pont de bus périphérique multiplexe les différents bus que constitue le Bus Mémoire en un seul bus composé de 16 lignes d'adresses et 16 ou 32 lignes de données, en plus des signaux de contrôle qui leur sont associés. Sur le DSP TMS 320F2812 on trouve deux versions de bus périphérique. Une version supporte uniquement des accès sur 16-bit (appelée: PF2). L'autre version supporte les deux accès 16-bit et 32-bit (appelée: PF1).

2.3 L'interface JTAG

Le DSP TMS 320F2812 intègre l'interface standard IEEE 1149.1 JTAG (Junction Transmit Action Group) qui est un outil très puissant qui supporte l'échange en temps réel des données entre le DSP et l'hôte (PC) durant la phase de débogage du projet. Avec ce mode temps réel, il est possible de visualiser et modifier les variables et les contenus de mémoires, registres, en temps réel lors de l'exécution du code sans aucune latence.

2.4 L'interface externe (XINTF)

L'interface externe XINTF est utilisée pour la connexion d'autres appareils externes comme des mémoires. Cette interface consiste en 19 lignes d'adresses, 16 lignes de données et 3 lignes de sélection de puce (chip-select). Les lignes chip-select sont organisées en cinq (5) zones externes, Zones 0, 1, 2, 6 et 7. Les zones 0 et 1 partagent un seul chip-select, les zones 6 et 7 partagent aussi un seul chip-select.

2.5 Les mémoires flash et OTP

Le DSP TMS 320F2812 contient (16 x 128K) de mémoire flash incorporée, séparée en quatre secteurs (16 x 8K), et six secteurs (16 x 16K). Il contient aussi une seule mémoire (OTP : One Time Programmable) de (16 x 1K). Les mémoires flash et OTP sont organisées en espace programme et données, utilisées pour exécuter le code ou stocker des données.

2.6 Les SARAMs (M0 et M1)

Le TMS 320F2812 contient deux blocs SARAM (Single-Access RAM), M0 et M1, de taille (16 x 1K) chacun. Les blocs M0 et M1, sont organisés en espace programme et données. Donc, l'utilisateur peut utiliser M0 et M1 pour exécuter un code programme ou pour mémoriser des données.

2.7 Les SARAMs (L0, L1 et H0)

Le TMS 320F2812 contient une mémoire additionnel SARAM (Single-Access RAM) de 16 x 16k, divisée en trois blocs L0 (8k), L1 (4k) et H0 (8k). Chaque bloc peut être accédé, indépendamment. Chaque bloc est organisé en espace programme et données.

2.8 La Boot-ROM

La Boot ROM est une mémoire programmée en usine (factory-programmed) de taille (16 x 4k). La Boot ROM contient des tables standards, comme les formes d'ondes Sin/Cos, à exploiter dans des algorithmes mathématiques.

2.9 Le Module de Sécurité

Le TMS 320F2812 comporte un niveau très élevé de sécurité qui permet de protéger le programme et empêcher les utilisateurs non autorisés d'examiner les contenus des mémoires via le port JTAG. Cette sécurité est obtenue grâce à un mot de passe sur 128 bit, que l'utilisateur programme en mémoire flash. Un seul module code sécurité (CSM) est utilisé pour protéger les mémoires (Flash/ROM/OTP) et les blocs SARAM (L0/L1).

2.10 Le Bloc PIE (Peripheral Interrupt Expansion)

Le bloc d'extension d'interruption périphérique PIE sert à multiplexer de nombreuses sources d'interruptions sous forme d'un ensemble plus petit d'interruptions. Le bloc PIE peut supporter jusqu'à 96 interruptions périphériques. Sur le TMS 320F2812, 45 des interruptions possibles sont utilisées par les périphériques. Les 96 interruptions sont regroupées sur 8 blocs et chaque groupe est introduit dans une des 12 lignes d'interruption (INT1-INT12) de la CPU.

2.11 Les interruptions externes (XINT1, XINT2, XINT13, XNMI)

Le TMS 320F2812 supporte trois interruptions externes mascables (XINT1, 2, 13). L'interruption XINT13 est combinée avec une autre interruption externe (XNMI) non-mascable. Le signal combiné résultant est XNMI-XINT13.

2.12 Les Horloges des périphériques

Chaque périphérique intègre une horloge individuelle qui peut être validée ou non, afin de réduire la consommation de puissance quand un périphérique n'est pas utilisé. En plus, les systèmes d'horloge de certains périphériques peuvent être adaptés et ajustés par rapport à l'horloge CPU. Ceci permet au cadencement des périphériques d'être découplé des vitesses élevées de l'horloge CPU.

2.13 L'oscillateur OSC et le bloc PLL

Le DSP TMS 320F2812 peut être cadencé par un oscillateur externe ou par un quartez attaché à l'oscillateur interne du DSP. Le TMS 320F2812 est équipé d'une PLL qui supporte dix (10) rapports horloge (pré-scalaires). Les rapports de la PLL peuvent être modifiés dans le programme, de plus le bloc PLL peut être placé en mode de déviation (bypass).

La figure ci-après montre l'oscillateur OSC et le bloc PLL du DSP TMS 320F2812 :

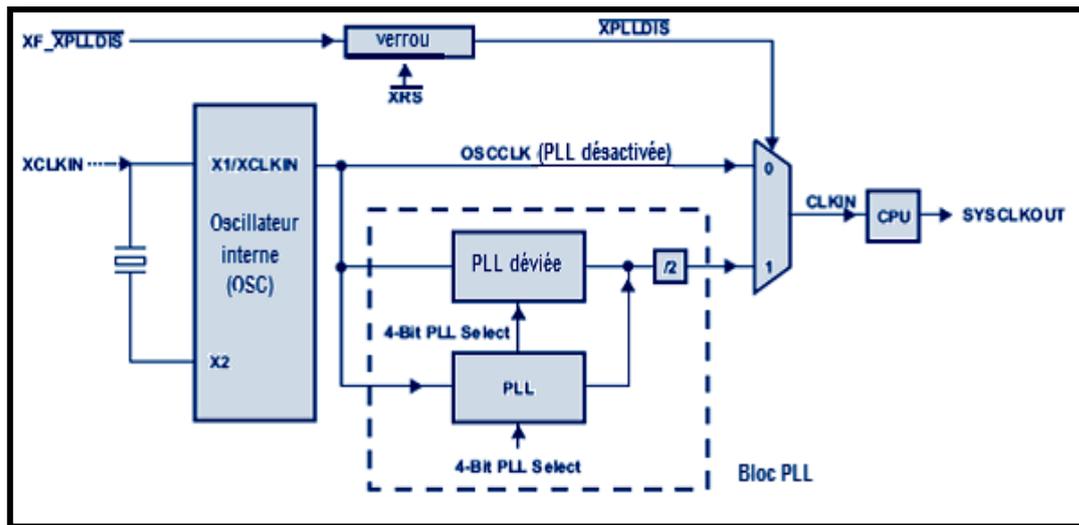


Figure 2.5 Les blocs de l'oscillateur OSC et la PLL.

Le circuit de l'oscillateur (OSC) permet à un quartz d'être attaché au DSP à travers les pins X1/XCLKIN et X2. Si un quartz n'est pas utilisé, un oscillateur externe peut être directement connecté à la pin X1/XCLKIN et la pin X2 reste non-connectée. Les bits [3 :0] du PLLCR fixent le rapport de cadencement (clocking ratio).

En mode PLL activé, si l'horloge d'entrée XCLKIN ou l'oscillateur d'horloge est supprimé (e), la PLL délivrera une horloge « mode-mou » (limp-mode). L'horloge limp-mode continuera de cadencer la CPU et les périphériques à une fréquence typique 1- 4 MHz.

2.14 Le bloc LPM (Low-Power Modes)

Le TMS 320F2812 dispose de trois modes basse-puissance, qui fonctionnent ainsi:

- a) **Le mode IDLE (inoccupé):** La CPU n'exécute aucune tâche durant ce mode tant que les bits (LPM) du registre LPMCR0 sont mis à 0. Le DSP sort de ce mode par la présence de n'importe quelle interruption activée ou un signal XNMI. Les signaux qui peuvent éveiller le DSP de ce mode sont : \overline{XRS} , \overline{WDINT} , XNMI, Debugger et toute interruption validée.
- b) **Le mode STANDBY:** L'utilisateur aura besoin de sélectionner quels signaux éveilleront le DSP. Les signaux qui peuvent éveiller le DSP de ce mode sont : \overline{XRS} , XINT1, XNMI, T1/2/3/4CTRIIP, C1/2/3/4/5/6TRIP, SCIRXDA, SCIRXDB, \overline{WDINT} et le Debugger.
- c) **Le mode HALT:** Les signaux qui peuvent sortir le DSP de ce mode HALT sont : \overline{XRS} , XNMI, Debugger.

Low Power Mode	Horloge CPU	Horloges Périphériques	Horloge Watchdog	PLL / OSC
Fonctionnement normal	on	on	on	on
IDLE	off	on	on	on
STANDBY	off	off	on	on
HALT	off	off	off	off

Tableau 2.1 Les modes basse puissance (Low Power Modes LPM).

Interruption de sortie / Low Power Mode	RESET	Interruption externe	Interruption périphérique validée
IDLE	oui	oui	oui
STANDBY	oui	oui	non
HALT	oui	non	non

Tableau 2.2 Sortir des modes basse puissance (Low Power Modes).

2.15 Le bloc Watchdog

Le bloc Watchdog du DSP TMS 320F2812, génère une impulsion en sortie (\overline{WDINT}), d'une largeur de 512 fois celle de l'horloge oscillateur (OSCCLK) à chaque fois que le compteur 8 bit du Watchdog atteint sa valeur maximale, ce qui conduit au reset du DSP. Pour éviter ceci, l'utilisateur doit désactiver le compteur ou bien écrire périodiquement la valeur 0x55 suivie de la valeur 0xAA dans le registre WDKEY (Watchdog Key Register), ce qui permettra de réinitialiser le compteur Watchdog et éviter son débordement.

La figure suivante montre les divers blocs fonctionnels dans le module watchdog.

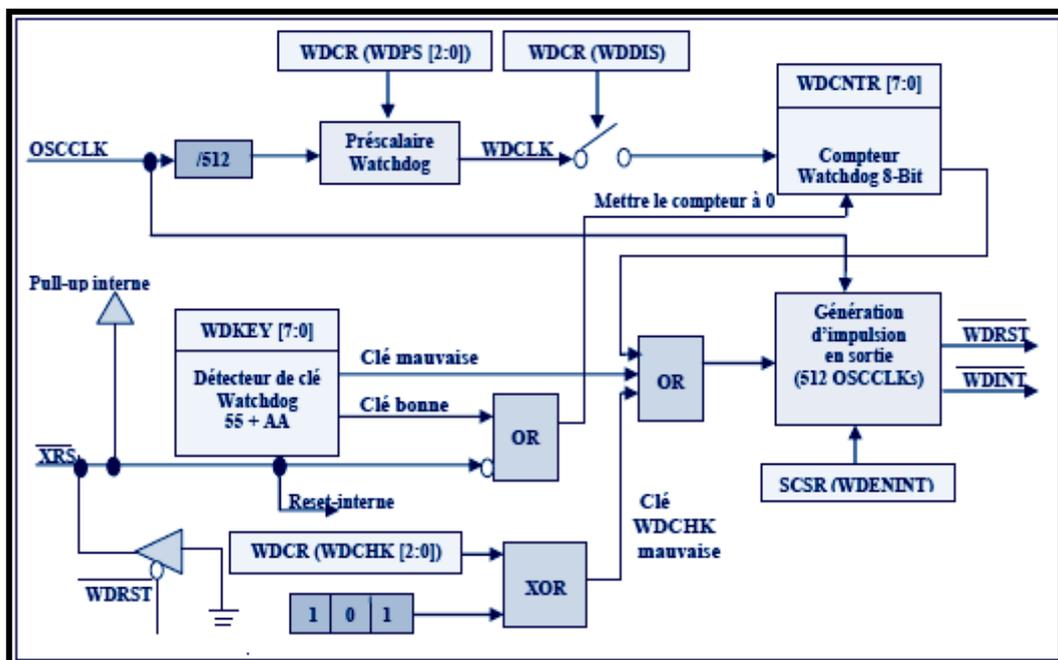


Figure 2.6 Schéma bloc du module Watchdog.

Le signal d'interruption \overline{WDINT} permet au watchdog d'être utilisé comme réveille pour sortir le DSP du mode IDLE/STANDBY.

2.16 Les Périphériques [39]

Le TMS 320F2812 comporte plusieurs périphériques intégrés, qui peuvent servir dans des applications de contrôle et de communication. Ces périphériques sont cités ci-dessous, et seront décrits avec plus de détails dans les sections qui suivront :

- Trois CPU-Timers 32-bit.
- Deux Modules EVA et EVB (Event-Manager).
- Un Module ADC (Analog-to-Digital Converter).
- Un Module eCAN (enhanced Controller Area Network).
- Un Module McBSP (Multichannel Buffered Serial Port).
- Deux Modules SCI-A et SCI-B (Serial Communications Interface).
- Un Module SPI (Serial Peripheral Interface).
- Des Entrée/Sortie numérique (GPIO).

2.16.1 Les Timers 0/1/2 de la CPU

Les Timers 0/1/2 de la CPU sont des timers 32-bit identiques avec des périodes prédéfinies et des préscales sur 16-bit. Ces timers possèdent un registre décompteur 32-bit, qui génère une interruption quand le compteur passe à la valeur zéro (0). Le compteur est décrémenté au rythme de l'horloge CPU divisée par la valeur du prescalaire. Quand le compteur atteint la valeur zéro (0), il est automatiquement rechargé par la valeur de la période.

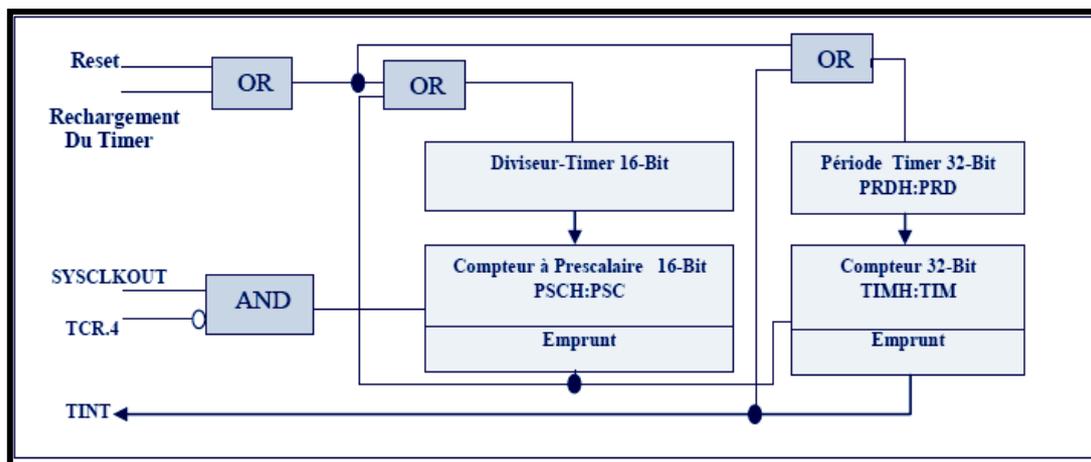


Figure 2.7 Schéma bloc des Timers de la CPU.

Le registre compteur 32-bit « TIMH:TIM » est chargé avec la valeur du registre période « PRDH:PRD ». Le registre compteur se décrémente et quand il passe à la valeur zéro (0), un signal d'interruption timer (TINT) sera généré.

2.16.2 Les modules Event Manager (EVA, EVB)

La figure ci-après montre le diagramme du bloc fonctionnel de l'Event Manager EVA dans le DSP TMS 320F2812. [39]

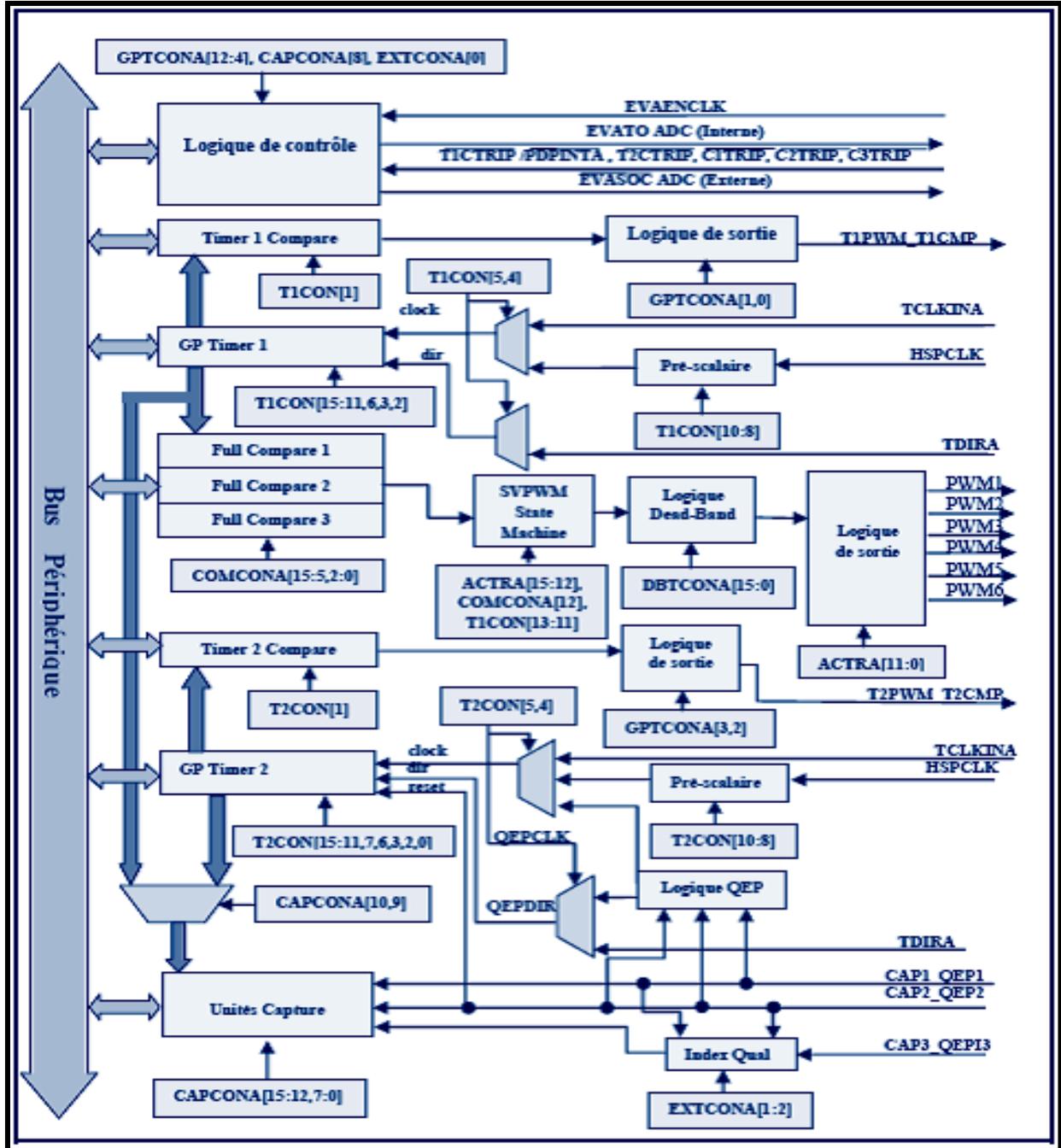


Figure 2.8 Diagramme du bloc fonctionnel de l'Event Manager A.

Les modules Event-Manager A et B incluent des timers GP à usage général, des unités full-compare / PWM, des unités capture (CAP) et un circuit QEP (Quadrature-Encoder Pulse).

A. Les timers GP (General-Purpose)

Chaque module Event-Manager (EVA ou EVB) comporte deux timers GP. Les timer1 et timer2 pour EVA, les timer3 et timer4 pour EVB), et chacun est caractérisé par :

- Un compteur 16-bit (TxCNT).
- Un registre compare 16-bit (TxCMPR).
- Un registre période 16-bit (TxPR).
- Un registre contrôle 16-bit (TxCON).

B. Les unités Full-Compare

Il y a trois (3) unités full-compare sur chaque Event-Manager du TMS 320F2812. Ces unités Full-compare utilisent le GP-timer1/2 comme base de temps et génèrent six sorties PWM, avec un circuit deadband (zone morte) programmable. L'état de chacun des six sorties est configurable indépendamment. Les registres compare des unités compare, permettent un changement programmable de la largeur de l'impulsion compare/PWM selon le besoin.

C. Le générateur deadband (zone morte)

Le générateur deadband peut être validé ou non individuellement pour chaque sortie de l'unité compare. Le générateur deadband produit deux sorties (avec ou sans temps mort) pour chaque signal de sortie de l'unité compare. Les états de sortie du générateur deadband sont configurables et modifiable selon le besoin.

D. Génération de l'onde PWM

Chaque Event-Manager peut générer jusqu'à huit (8) ondes PWM simultanément. Trois paires indépendantes (six sorties) par les trois unités Full-Compare avec des temps-morts programmables et deux PWMs indépendantes par chaque GP-timer1/2 compare. Les ondes PWM élaborées par le DSP sont caractérisées par les points suivants :

- Gamme large de temps-mort programmable pour les sorties PWMs de Full-compare.
- Modification de la fréquence porteuse de la PWM selon le besoin.
- Possibilité de génération d'ondes PWM asymétriques ou symétriques.

E. L'Unité Capture

L'unité capture assure une fonction de détection et d'enregistrement pour les différents événements ou transitions sur les pins CAPx. Les transitions détectées sur les pins d'entrée capture, CAPx (x = 1, 2, ou 3 pour EVA; et x = 4, 5, ou 6 pour EVB) sont comptées par le compteur du timer GP utilisé, puis leur valeur finale sera sauvegardée dans la pile FIFO. Le fonctionnement de l'unité capture est assuré par la collaboration de:

- Un registre de contrôle 16-bit, CAPCONx (R/W).
- Trois piles FIFO 16 bit à deux niveaux, une pour chaque unité capture.

- Trois pins Capture (CAP1/2/3 pour EVA, CAP4/5/6 pour EVB).
- Trois flags d'interruption mascable, un pour chaque unité capture.
- Des timer1/2 (pour EVA) ou 3/4 (pour EVB) comme base de temps.

2.16.3 Le module ADC

Le module ADC du TMS 320F2812 consiste en un noyau ADC 12-bit avec un circuit échantillonneur-bloqueur intégré. Ce module possède seize (16) canaux, configurable comme deux modules huit (8) canaux indépendants ou cascades pour former un seul module seize (16) canaux. Sur chaque canal, une fois la conversion est complétée, la valeur numérique du signal présent sur le canal sélectionné est sauvegardée dans son registre RESULT (Résultat) approprié. Le module ADC du DSP TMS 320F2812 est caractérisé par :

- Un noyau ADC 12-bit avec deux échantillonneurs-bloqueurs intégrés.
- Une tension sur les entrées analogiques allant de 0.0 V à 3.0 V.
- Vitesse de conversion élevée : 80 ns pour une horloge ADC de 25 MHz.
- Seize (16) canaux multiplexés avec des pins GPIO.
- Seize (16) registres résultats, pour sauver les valeurs retournées par la conversion.
- La valeur numérique de la tension analogique est tirée par :

Valeur numérique = 0	Si l'entrée ≤ 0
Valeur numérique = $4095 \times \frac{\text{tension analogique} - \text{ADCLO}}{3}$	Si 0 < l'entrée < 3V
Valeur numérique = 4095	Si l'entrée ≥ 3V

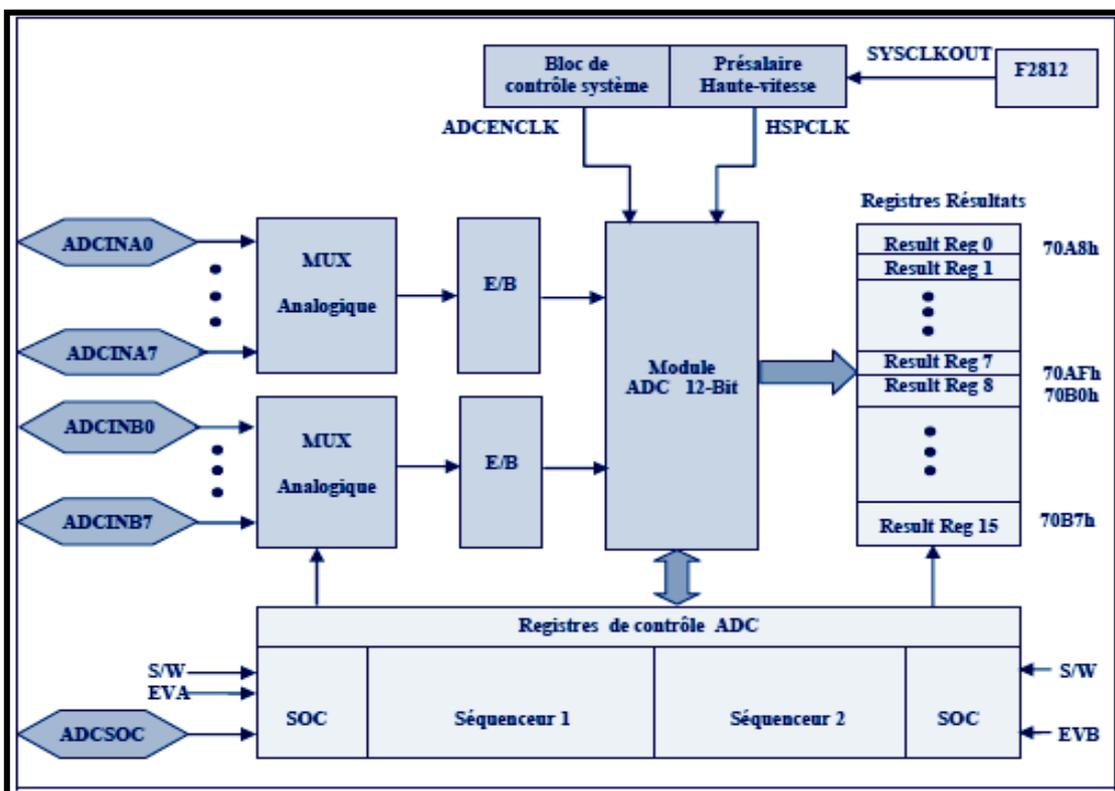


Figure 2.9 Diagramme du bloc fonctionnel du module ADC.

2.16.4 Le module eCAN

Le module eCAN (Enhanced Controller Area Network) est la version développée du périphérique CAN. Certaines de ses caractéristiques sont données ci-dessous :

- Complètement compatible avec le Protocol CAN, version 2.0B.
- Supporte une vitesse de transmission de données supérieure à 1 Mbps.
- 32 mail-boxes (boîtes aux lettres).
- Supporte le fonctionnement en mode basse-puissance.

2.16.5 Le module McBSP

Le module de communication série McBSP (Multichannel Buffered Serial Port) possède les caractéristiques suivantes:

- Assure une communication Full-duplex.
- Une large sélection des tailles des données (8-, 12-, 16-, 20-, 24-, ou 32-bits).
- Transfert de données sur 8-bit avec LSB ou MSB comme premier bit.
- Horloge interne configurable.
- Fonctionne avec des appareils SPI-compatible.

2.16.6 Le module SCI

Le TMS 320F2812 comporte deux modules SCI (Serial Communication Interface) SCI-A et SCI-B. Ils assurent des communications numériques entre la CPU et d'autres appareils et périphériques. Ils peuvent être exploités indépendamment ou simultanément en mode full-duplex. Les caractéristiques de chaque module SCI sont citées ci-dessous:

- Deux pins, une (SCITXD) pour la transmission et l'autre (SCIRXD) pour la réception.
- Fonctionnement en Half- duplex ou full-duplex.
- La trame de la donnée est formée d'un bit de start, d'une donnée de 1 à 8 bits, d'un bit de parité optionnel et d'un ou deux bits de stop.
- Vitesse Max = $75 \text{ MHz} / 16 = 4.688 \times 10^6 \text{ b/s}$.
- Format de codage NRZ (Non Retour à Zéro).
- Circuit logique pour la détection automatique du débit de transfert.
- Débit de transfert en Baud programmable, qui est donné par :

$$\begin{aligned} \text{Débit (Baud)} &= \frac{\text{LSPCLK}}{(\text{BRR} + 1) * 8} && \text{Si SCIBRR} \neq 0 \text{ (Baude Rate Register)} \\ \text{Débit (Baud)} &= \frac{\text{LSPCLK}}{16} && \text{Si SCIBRR} = 0 \end{aligned}$$

2.16.7 Le module SPI

Le TMS 320F2812 intègre un module SPI (Serial Peripheral Interface) qui permet de transmettre ou recevoir des données en série avec un taux de transfert de bits programmable. Généralement, le module SPI est utilisé pour des communications entre le DSP et des périphériques externes ou un autre processeur. Certaines caractéristiques du module SPI sont :

- Deux modes de fonctionnement : maître et esclave.
- La longueur de la donnée est sur 16 bits.
- Des transmissions et réceptions simultanées.
- Débit en Baud donné par:

$$\text{Débit (Baud)} = \frac{\text{LSPCLK}}{(\text{SPIBRR} + 1)} \quad \text{Si SPIBRR} = 3, \dots, 127.$$

$$\text{Débit (Baud)} = \frac{\text{LSPCLK}}{4} \quad \text{Si SPIBRR} = 0, 1, 2.$$

2.16.8 Les signaux GPIO multiplexés

La plupart des signaux des périphériques sont multiplexés avec des signaux GPIO (Entrée/Sortie à usage général). Ce multiplexage permet l'utilisation d'une pin comme pin GPIO si la fonction périphérique n'est pas utilisée. Les pins partagées sur le DSP peuvent être individuellement configurées via les registres GPxMUX pour fonctionner comme des Entrées/Sorties numériques ou utilisées par les divers périphériques EV, ADC, etc.....

Si les registres sont sélectionnés pour fonctionner en mode « Entrée/Sortie numérique », des registres GPxDIR sont utilisés pour définir la direction (Entrée ou Sortie) des pins.

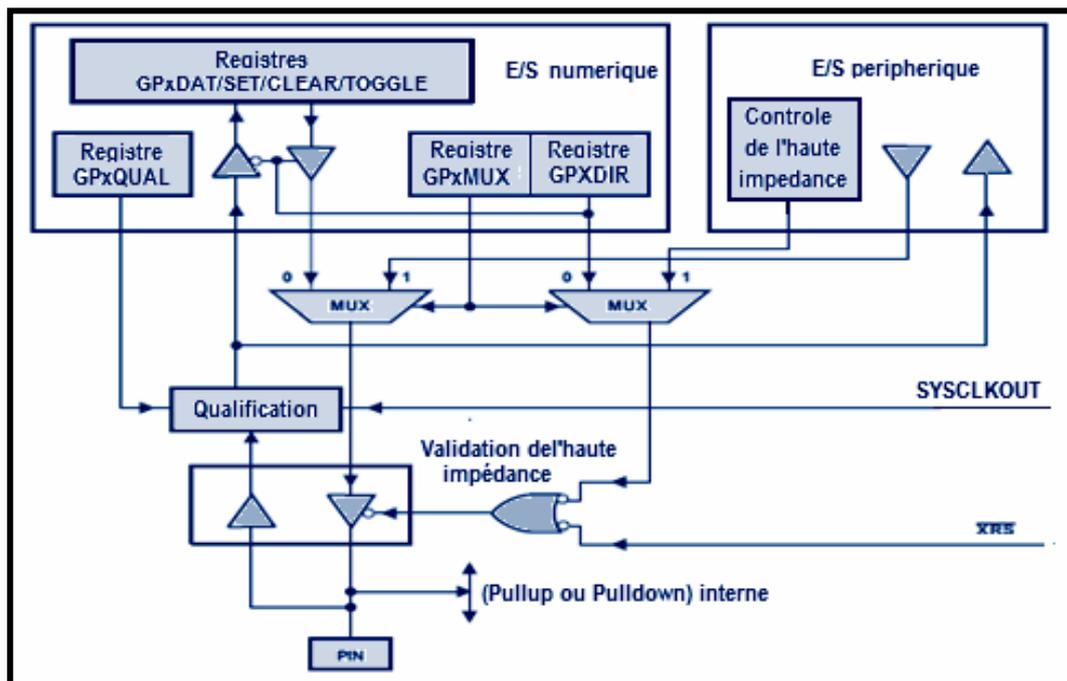


Figure 2.10 Multiplexage des pins GPIO/Périphérique.

3. Description de l'unité centrale de traitement (CPU) [38]

Le TMS 320F2812 est un DSP à virgule fixe, avec une architecture Harvard modifiée, ce qui lui permet d'effectuer la recherche d'instructions et data en parallèle, et permet de lire des instructions et data, et d'écrire les données dans la mémoire en même temps. La CPU assure le contrôle du déroulement du programme et le traitement des instructions. Elle effectue des opérations arithmétiques, logiques, et de décalage de bits.

3.1 Les composants de la CPU

La CPU du DSP TMS 320F2812 comporte: [38]

- Une unité (CPU) pour générer les adresses pour les mémoires data et programme, décoder et exécuter les instructions, effectuer des opérations arithmétiques, logiques et de décalage, et contrôler les transferts de données entre les registres de la CPU et les mémoires data et programme.
- Des signaux pour s'interfacer avec des mémoires et des périphériques, plus des signaux pour cadencer (clocking) et contrôler la CPU et la logique d'émulation, et des signaux d'interruptions.
- Une logique d'émulation pour surveiller et contrôler les différentes parties et fonctionnalités du DSP.

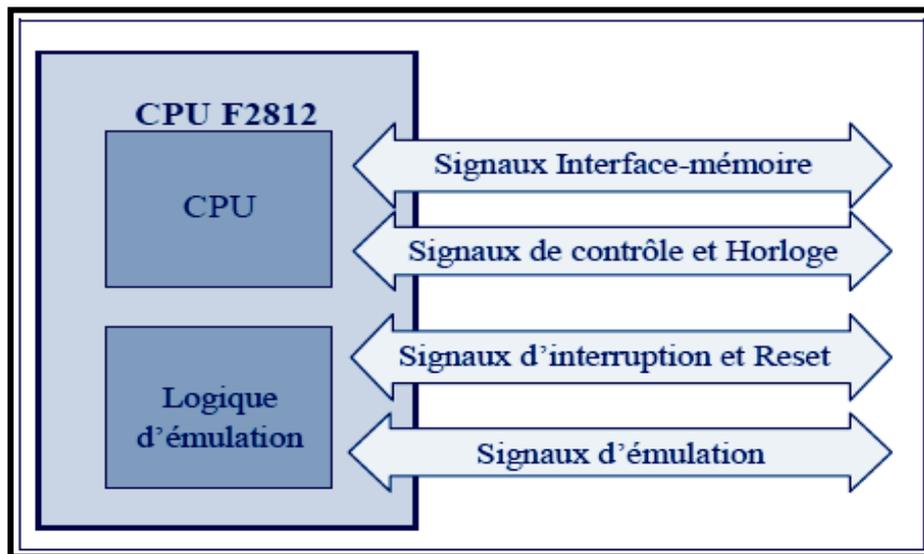


Figure 2.11 Diagramme conceptuel haut-niveau de la CPU.

3.2 L'architecture de la CPU

Le TMS 320F2812 comporte une CPU, une logique d'émulation et des signaux pour l'interfaçage avec la mémoire et les périphériques, plus trois bus d'adresses et trois bus data.

La figure 2.12 montre les blocs majeurs et les différents chemins des données. Le bus opérande fournit les valeurs pour les différentes opérations du multiplieur, du registre à décalage (Shifter), et de l'unité arithmétique et logique ALU, tandis que le bus résultat est chargé de porter les résultats vers les registres et la mémoire. Les autres bus (DRAB, DWAB, PRDB, PAB, DRDB, DWDB) illustrés sur la figure sont les bus interface-mémoire qui se trouvent à l'extérieur de la CPU. [38]

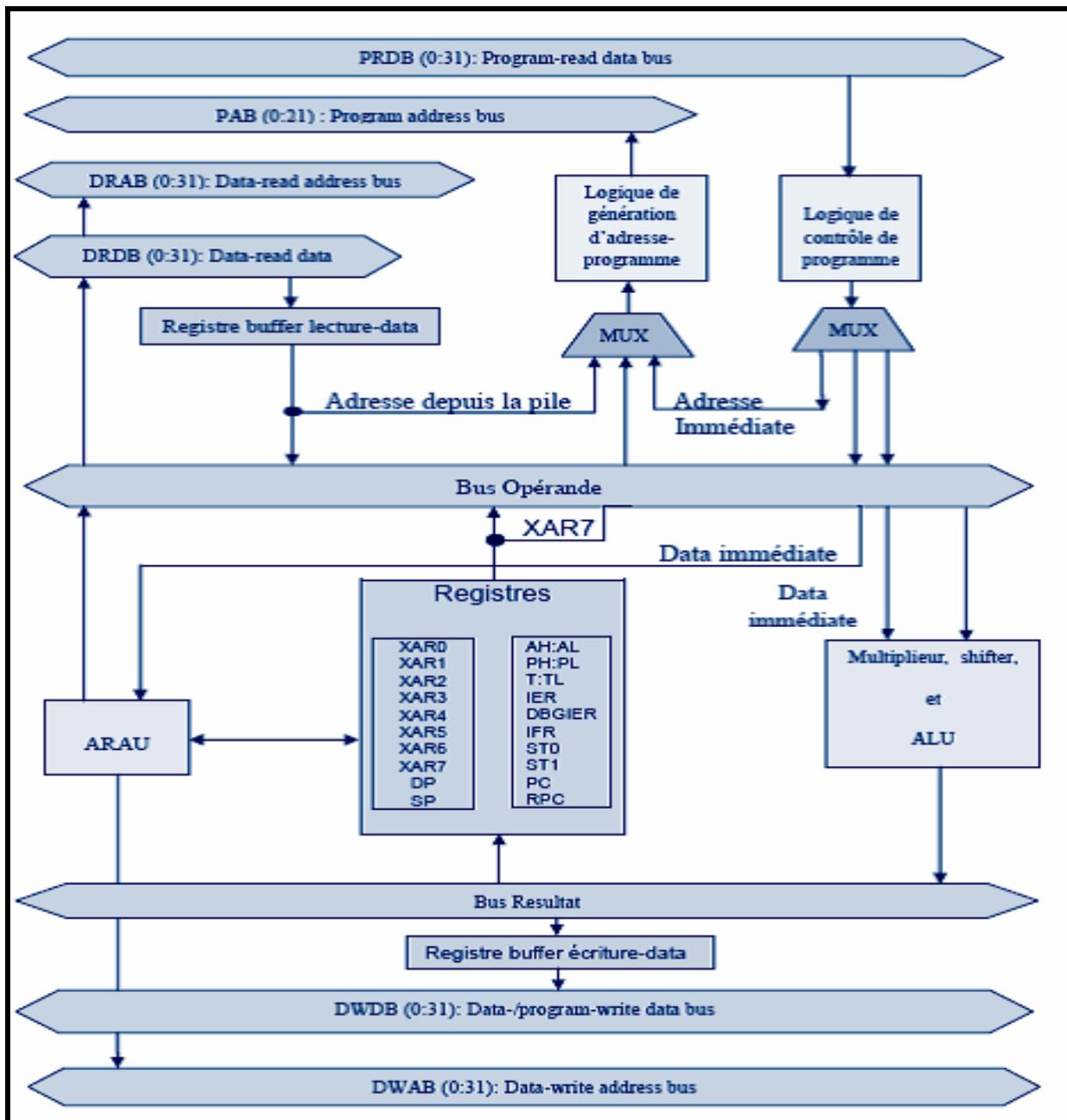


Figure 2.12 Diagramme conceptuel du bloc de la CPU.

La CPU comporte les blocs principaux suivant : [38]

- ❖ **Une logique de contrôle de programme et data:** Cette logique est chargée d'aligner et préparer une queue d'instructions issues de la mémoire programme.
- ❖ **Address Register Arithmetic Unit (ARAU) :** L'unité ARAU génère les adresses des données à rechercher à partir de la mémoire data. Pour la lecture des données, l'unité ARAU place l'adresse sur le bus DRAB, tandis que pour l'écriture des données elle utilise le bus d'adresse DWAB. L'unité ARAU permet aussi l'incrémentation ou la décrémentation du pointeur de pile (SP) et des registres auxiliaires (XAR0,..., XAR7).
- ❖ **Unité arithmétique et logique (ALU):** L'unité ALU effectue le traitement des opérations arithmétiques et des fonctions logiques. Pour faire ces calculs, l'ALU reçoit les données à partir des registres, de la mémoire data, ou de la logique de contrôle du programme. L'unité ALU sauvegarde les résultats dans un registre ou dans un emplacement de la mémoire data.
- ❖ **Multiplieur à virgule fixe:** Il accomplit des opérations de multiplication 32 x 32 bits en utilisant le codage complément-2 avec un résultat sur 64 bits.
- ❖ **Générateur d'adresse :** Il génère les adresses des mémoires programme et data.

3.3 Les registres de la CPU

La figure ci-dessous illustre les différents registres que comporte et utilise la CPU du DSP TMS 320F2812 : [38]

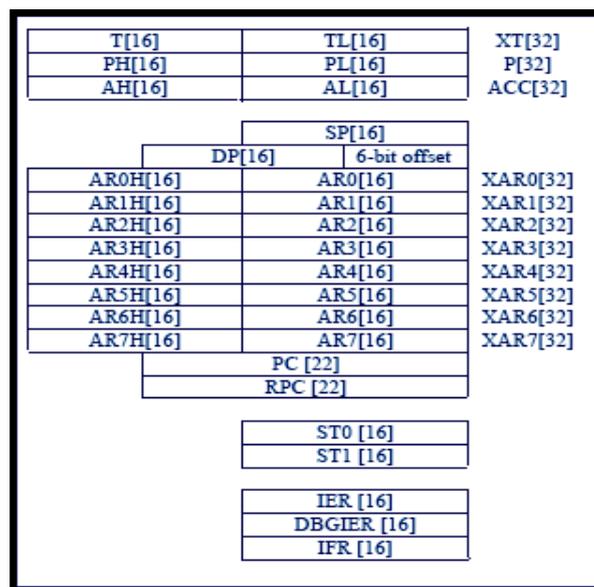


Figure 2.13 Les registres de la CPU du TMS 320F2812.

A. L'Accumulateur (ACC, AH, AL)

L'accumulateur ACC est le principal registre de travail dans la CPU. L'accumulateur ACC supporte des opérations sur 32-bit en un seul-cycle, telles que l'addition, la soustraction, et la comparaison. Il peut aussi recevoir temporairement les 32 bit des résultats de l'opération multiplication 32x32 bits. L'accumulateur ACC peut être traité comme deux registres 16-bit indépendants AH (High 16 bit) et AL (Low 16 bit) qui peuvent être accédés indépendamment.

B. Le Registre Multiplicande (XT)

Le registre multiplicande (XT) est principalement utilisé pour sauvegarder des entiers signés sur 32 bit, avant d'effectuer l'opération de multiplication 32 x 32 bits. La portion basse sur 16 bits du registre XT se reporte au registre TL, et la portion haute sur 16 bits se reporte au registre T qui est principalement utilisé pour sauvegarder une valeur sur 16 bits avant l'opération de multiplication 16 x 16 bits. Le registre T peut être aussi utilisé pour spécifier la valeur de décalage dans certaines instructions.

C. Le Registre Produit (P, PH, PL)

Le registre produit (P) est typiquement utilisé pour maintenir le résultat 32 bit issu d'une multiplication. Il peut être aussi chargé directement par une valeur à partir d'un emplacement 16 ou 32 bit de la mémoire data, par une constante 16 bit, par l'accumulateur ACC ou par un autre registre de la CPU. Le registre P peut être traité comme un registre 32 bit ou comme deux registres 16 bit indépendants PH (High 16 bits) et PL (Low 16 bits). Les registres P, PH, ou PL, sont aussi utilisés dans certaines instructions de décalage de bits.

D. Le Pointeur de Pile (SP)

Le pointeur de pile SP permet l'utilisation d'une pile logicielle (Software Stack) dans la mémoire data. Le pointeur SP possède 16 bit, donc peut adresser uniquement les 64 k mots de la partie basse de la mémoire data. Quand le SP est utilisé, les six bits supérieurs des 22 bit de l'adresse sont forcés à zéro (0).

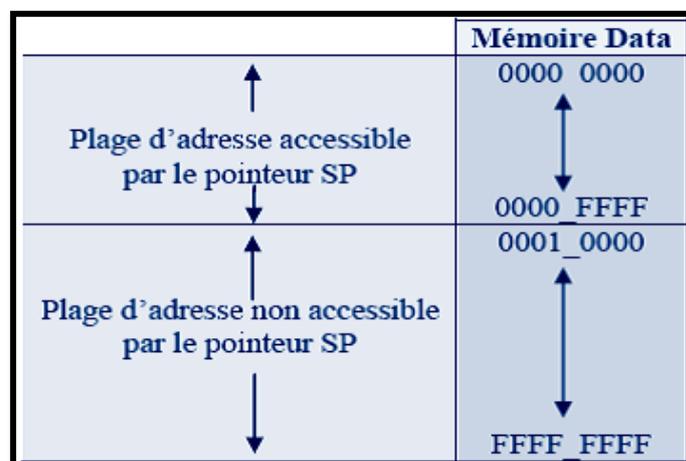


Figure 2.14 Les adresses à portée du pointeur de pile.

Le fonctionnement de la pile est caractérisé par les points suivants:

- Le pointeur monte du bas vers le haut de la mémoire data.
- Le SP pointe toujours à cote de l'emplacement vide dans la pile.
- Lors d'un reset, le SP est initialisé, et pointe à l'adresse 0000 0400h.
- Lors de la sauvegarde de valeurs 32 bits dans la pile, les 16 bits moins significatifs sont sauvegardés en premier, puis les 16 bits les plus significatifs sont sauvegardés à la prochaine adresse supérieure.

E. Le pointeur de Page Data (DP)

Dans ce mode d'adressage direct, la mémoire data est adressée en blocs de 64 mots chacun, appelés pages data. Les 4 M mots de la mémoire data consistent en 65 536 pages étiquetées de 0 à 65 535, comme c'est montré sur la figure 2.15. Dans le mode d'adressage direct, le pointeur DP sur 16 bit maintient le numéro de la page data actuelle.

Page Data	Offset	Mémoire Data
00 0000 0000 0000 00	00 0000	Page 0: 0000_0000-0000_003F
.	.	
00 0000 0000 0000 00	11 1111	
00 0000 0000 0000 01	00 0000	Page 1: 0000_0040-0000_007F
.	.	
00 0000 0000 0000 01	11 1111	
00 0000 0000 0000 10	00 0000	Page 2: 0000_0080-0000_00BF
.	.	
00 0000 0000 0000 10	11 1111	
.	.	.
.	.	.
.	.	.
.	.	.
11 1111 1111 1111 11	00 0000	Page 65_535: 003F_FFC0-003F_FFFF
.	.	
11 1111 1111 1111 11	11 1111	

Figure 2.15 Les pages de la mémoire Data.

F. Les Registres Auxiliaires (XAR0-XAR7)

La CPU du DSP comporte huit (8) registres 32-bits (XAR0-XAR7), qui peuvent être utilisés soit comme des pointeurs aux emplacements de la mémoire ou comme des registres à usage général. Les 16 bit LSB du registre auxiliaire nommé AR0-AR7, peuvent être utilisés comme un registre à usage général dans des boucles de contrôle ou dans des comparaisons de valeurs sur 16 bits. Cependant la partie MSB du registre auxiliaire nommé AR0H-AR7H, est accessible uniquement comme une partie de XAR0-XAR7 et n'est pas accessible individuellement.

G. Le Compteur de Programme (PC)

Quand le pipeline est plein, le compteur programme PC sur 22 bits pointe toujours à l'instruction en cours de traitement qui a atteint la deuxième phase de décodage (Decode2) dans le pipeline.

H. Le Compteur de Retour du Programme (RPC)

Le registre RPC est exploité uniquement, lors de l'utilisation des instructions d'appel ou de retour d'un sous-programme, LCR (Long call) et LRETR (Long return).

Quand un appel d'une fonction ou d'un sous-programme est effectué en utilisant l'instruction LCR, l'adresse du retour est sauvegardée dans le registre RPC et la valeur ancienne dans RPC est déplacée vers la pile (en deux opérations 16-bit). Quand un retour d'une fonction ou d'un sous-programme est réalisé à l'aide de l'instruction LRETR, l'adresse de retour est lue à partir du registre RPC et la valeur dans la pile est transférée vers le registre RPC (en deux opérations 16-bit).

I. Les registres d'état (ST0 et ST1)

Le TMS 320F2812 possède deux registres d'état ST0 et ST1, qui contiennent plusieurs bits flags et bits de contrôle. Ces registres peuvent être sauvegardés dans la mémoire data ou chargés à partir de celle-ci. Les bits d'états sont organisés, selon la phase de modification de leurs valeurs dans le pipeline. Les bits du registre ST0 sont modifiés dans la phase d'exécution (E), et les bits du registre ST1 sont modifiés dans la 2^{ème} phase de décodage (D2).

J. Les Registres Contrôle-Interruption (IFR, IER, DBGIER)

La CPU du DSP TMS 320F2812 comporte trois registres destinés au contrôle et la manipulation d'interruptions, qui sont IFR (Interrupt Flag Register), IER (Interrupt Enable Register) et DBGIER (Debug Interrupt Enable Register).

Le registre IFR contient des bits flag, des interruptions mascables. On valide ou non une interruption mascable par le bit correspondant dans le registre IER. Le registre DBGIER indique les interruptions temps-critique (time-critical) qui seront gérées quand le DSP est en mode d'émulation temps réel.

4. Vue d'ensemble sur les interruptions [38]

Les interruptions sont des signaux d'origines matérielles ou logicielles, qui forcent la CPU de suspendre le traitement de sa séquence-programme actuelle, et passer à l'exécution d'une routine d'interruption ISR (sous-programme d'interruption).

Sur le TMS 320F2812, les interruptions peuvent être déclenchées par le programme (les instructions: INTR, OR IFR, ou TRAP) ou par le matériel (une pin, un périphérique externe, ou par la logique interne de contrôle des périphériques). Si plusieurs interruptions matérielles sont déclenchées au même moment, le DSP prend en charge ces interruptions selon leurs priorités.

Le DSP TMS 320F2812 comporte un module PIE (Peripheral Interrupt Expansion) qui multiplexe plusieurs signaux d'interruption, issues des divers périphériques, en un seul signal d'interruption CPU. Le module PIE assure un contrôle complémentaire avant qu'une interruption atteigne la CPU.

La figure suivante montre comment diverses sources d'interruption sont multiplexées dans le DSP 320F2812. [38]

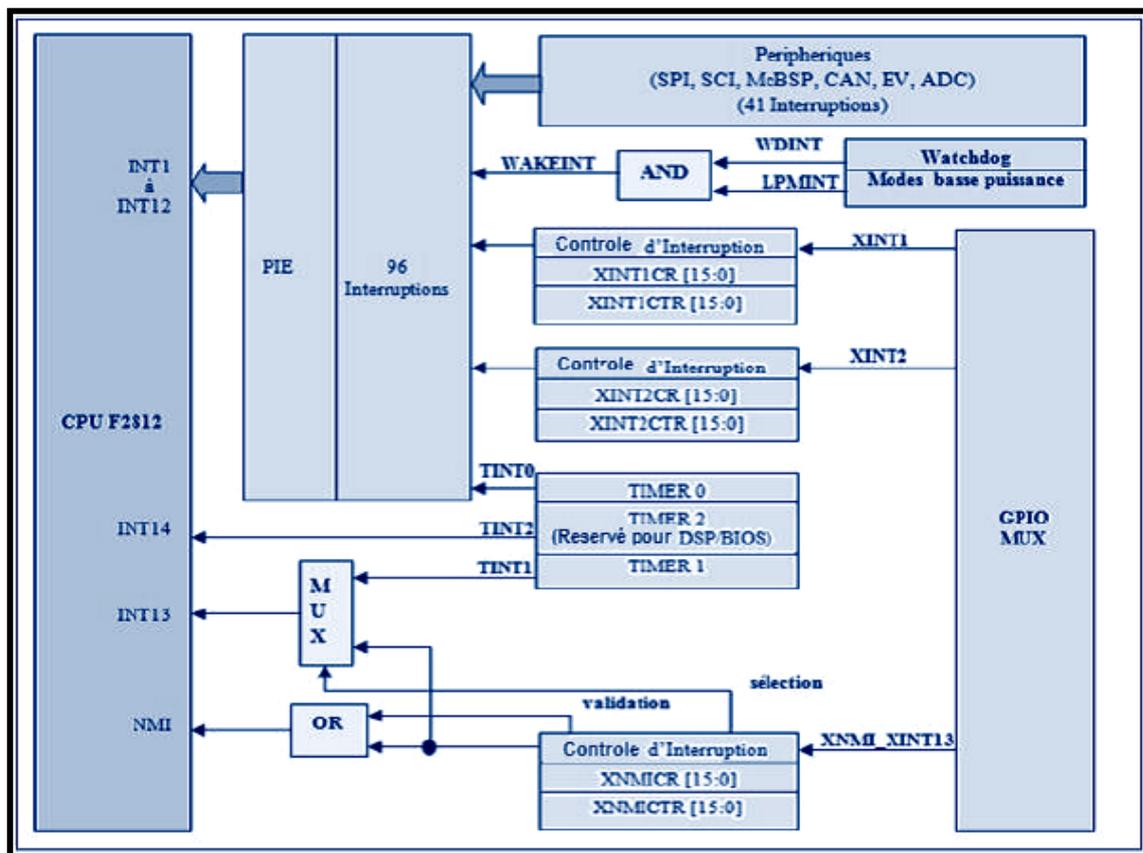


Figure 2.16 Diverses sources d'interruptions sont multiplexées dans le DSP 320F2812.

Le TMS 320F2812 manipule les interruptions dans quatre phases principales : [38]

- 1) **Recevoir la requête d'interruption:** La suspension de la séquence-programme actuelle doit être requise par une interruption logicielle ou matérielle.
- 2) **Approuver l'interruption:** Le DSP doit approuver la requête d'interruption. Si l'interruption est mascable, certaines conditions doivent être vérifiées pour les approuver, tandis que pour les interruptions matérielles non-masquables et les interruptions logicielles, l'approbation est immédiate.
- 3) **Se préparer pour la routine ISR et sauvegarder les valeurs des registres:**
Les principales tâches effectuées dans cette phase sont :
 - Une exécution complète de l'instruction actuelle et rejet du pipeline de toute instruction qui n'a pas encore atteint la phase decode2 dans le pipeline.
 - Enregistrer automatiquement la plupart du contexte actuel du programme en sauvegardant les registres suivants dans la pile: ST0, T, AL, AH, PL, PH, AR0, AR1, DP, ST1, DBGSTAT, PC, et IER.
 - Aller chercher le vecteur d'interruption et le charger dans le compteur PC.
- 4) **Exécuter la routine de service interruption:** Le DSP se branche à l'adresse (vecteur) sauvegardée dans un emplacement vecteur prédéterminé et exécute la routine de service interruption ISR.

5. Le pipeline

5.1 Notion du pipeline

A l'inverse de l'architecture Von Neumann où les mémoires programme et data sont situées dans la même zone, les processeurs caractérisés par l'architecture Harvard ont des mémoires programme et données séparées. Donc l'accès à chacune des deux mémoires se fait via des bus distincts. Cette organisation permet de transférer une instruction et des données simultanément, ce qui améliore les performances et les délais de traitement. Cette réorganisation est connue sous le terme « pipeline », qui offre la possibilité du déroulement simultané et parallèle des différentes étapes de traitement des instructions par le processeur.

5.2 Le Pipelining des instructions

Lors de l'exécution d'un programme, la CPU du DSP TMS 320F2812, accomplit les tâches et fonctions de base citées ci-dessous : [38]

- Aller chercher les instructions dans la mémoire-programme.
- Décoder les instructions.
- Lire les valeurs data depuis la mémoire ou les registres de la CPU.
- Exécuter les instructions.
- Ecrire les résultats dans la mémoire ou dans les registres CPU.

Pour plus d'efficacité, le DSP TMS 320F2812 accomplit ces opérations sur huit (8) phases indépendantes. A tout moment, il peut y avoir jusqu'à huit (8) instructions à traiter, chacune dans une phase de traitement distincte. On donne ci-dessous les huit (8) phases pipeline, dans leur ordre de traitement : [38]

- 1) **Fetch 1(F1):** Dans cette première phase de recherche d'instructions, la CPU achemine une adresse de la mémoire-programme sur le bus d'adresse programme 22-bit, PAB (21:0)
- 2) **Fetch 2(F2) :** Dans cette deuxième phase de recherche d'instructions, la CPU lit l'instruction à partir de la mémoire-programme à l'aide du bus PRDB (Program-Read Data Bus) 32-bit, et charge l'instruction dans une queue d'instructions.
- 3) **Decode1(D1):** Dans cette première phase de décodage d'instructions, le hardware de décodage identifie les limites de l'instruction dans la queue d'instructions et détermine la taille (32-bit ou 16-bit) de la prochaine instruction à exécuter.
- 4) **Decode2(D2) :** Dans cette deuxième phase de décodage d'instructions, le hardware de décodage requête une instruction à partir de la queue d'instruction. L'instruction requise est chargée dans le registre d'instruction, où le décodage sera complété. Dans cette phase pipeline, les tâches suivantes sont accomplies :
 - Si la donnée sera lue à partir de la mémoire, la CPU génère l'adresse source.
 - Si la donnée sera écrite dans la mémoire, la CPU génère l'adresse destination.
 - L'unité ARAU (Address Register Arithmetic Unit), effectue toute modification requise sur le pointeur de pile (SP) ou au registre auxiliaire (XARn) et pointeur de registre auxiliaire (ARPn)
 - Si une discontinuité (ex : branchement) dans le déroulement du programme, est requise, elle sera prise en compte.
- 5) **Read1(R1) :** Si la donnée sera lue à partir de la mémoire, le hardware de la lecture (R1) mène l'adresse sur le bus d'adresse approprié DRAB (0:31).
- 6) **Read2 (R2):** Si la donnée est adressée dans la phase (R1), le hardware (R2) va chercher cette donnée à l'aide du bus data approprié DRDB (0:31).
- 7) **Execute (E):** Dans la phase d'exécution (E), la CPU effectue toutes les opérations arithmétiques et logiques, de multiplication, et décalage. L'opération d'écriture du résultat dans un registre de la CPU est réalisée à la fin de la phase d'exécution (E).
- 8) **Write (W) :** Si une valeur est transférée ou un résultat doit être écrit dans la mémoire, son écriture s'effectuera dans cette phase d'écriture (w).

Cycle	F1	F2	D1	D2	R1	R2	E	W
1	I1							
2	I2	I1						
3	I3	I2	I1					
4	I4	I3	I2	I1				
5	I5	I4	I3	I2	I1			
6	I6	I5	I4	I3	I2	I1		
7	I7	I6	I5	I4	I3	I2	I1	
8	I8	I7	I6	I5	I4	I3	I2	I1
9		I8	I7	I6	I5	I4	I3	I2
10			I8	I7	I6	I5	I4	I3
11				I8	I7	I6	I5	I4
12					I8	I7	I6	I5
13						I8	I7	I6
14							I8	I7
15								I8

Figure 2.17 Diagramme simplifié de l'activité du pipeline.

6. Le déroulement du programme [38]

Généralement, le déroulement du programme est séquentiel, c'est-à-dire, que la CPU exécute les instructions issues des adresses consécutives de la mémoire-programme. Parfois, une discontinuité (interruption, branchement, appel, retour, répétition d'instruction) est requise, où, le programme doit se brancher à une adresse non successive, ensuite la CPU exécute comme auparavant les instructions séquentiellement à partir de cette nouvelle adresse.

Le déroulement propre et net du programme, est assuré par la collaboration de la logique de contrôle du programme et de la logique de génération d'adresse-programme. Le déroulement propre et soigné du programme nécessite également un bon déroulement au niveau du traitement de l'instruction. Ceci est réalisé par le mécanisme pipeline qui recherche et prépare une queue d'instructions (32-bit ou 16-bit), qui seront maintenues dans le pipeline pour leur exécution parallèle par la suite.

Chaque instruction du programme passe par huit (8) phases indépendantes qui forment une instruction pipeline. A tout moment, jusqu'à huit (8) instructions peuvent être actives, et chacune dans sa propre phase de traitement.

Ce mécanisme, utilise trois compteurs d'adresses-programme, PC (Program Counter), IC (Instruction Counter), et FC (Fetch Counter). Le compteur PC pointe toujours à la deuxième phase de décodage dans le pipeline, tandis que le compteur IC pointe à la prochaine instruction à traiter. Quand le compteur PC pointe à une instruction 16-bit, le compteur IC= (PC+1), et quand il pointe à une instruction 32-bits, le compteur IC= (PC+2). La valeur dans le compteur FC est l'adresse où la prochaine recherche d'instruction sera réalisée.

7. Les modes d'adressage

La CPU du DSP TMS 320F2812 supporte les modes d'adressage suivants : [38]

A. Le mode d'adressage direct (DP) : Dans ce mode, le registre DP (16-bit) se comporte comme un pointeur de page de la mémoire data. Ce mode d'adressage est utile pour accéder aux adresses des structures de données dans la mémoire data.

Ex : `MOVW DP,#VarA` ; Charge DP avec la valeur de la page contenant la variable VarA.

B. Le mode d'adressage pile (SP) : Dans ce mode, le pointeur de pile SP 16-bit est utilisé pour accéder aux données dans la pile (software stack). Le pointeur SP peut être post-incrémenté ou pré-décrémenté lors de l'écriture (Pushing) ou de la lecture (Popping) des données dans la pile, respectivement.

Ex: `MOV *SP++, AL` ; Transférer le contenu 16-bit du registre AL vers le top de la pile.

C. Le mode d'adressage indirect (XARn) : Dans ce mode, les registres 32-bit XARn se comportent comme des pointeurs génériques de la mémoire data. L'instruction peut indiquer une post-incrémentation, pré/post-décrémentation, ou une valeur indexée.

Ex: `MOVL XAR1,#Tab1` ; Charger XAR1 avec l'adresse de début du Tableau1.

D. Le mode d'adressage registre : Dans ce mode, un autre registre peut jouer le rôle de l'opérande source ou destination d'une instruction. Ceci permet d'effectuer des opérations registre-vers-registre.

Ex: `MOVL XAR6,@ACC` ; Charger XAR6 avec le contenu du ACC.
`ADDL ACC,@ACC` ; $ACC = ACC + ACC$

8. Les outils logiciels pour DSP

A. Code Composer Studio IDE

La programmation du DSP TMS 320F2812 peut se faire à l'aide du logiciel Code Composer Studio (CCS IDE) en langage assembleur et/ou en langage C/C++. Code Composer Studio (CCS IDE) est un environnement de développement intégré (IDE) souple et facile à utiliser, qui incorpore une variété d'outils logiciels pour la génération du code exécutable, comme, le compilateur-C, l'assembleur, et le linker. Il permet une facilité dans l'organisation du projet et une émulation et débogage du programme en temps réel, Visualisation des contenus des registres CPU et périphériques, tracer et visualiser des graphes en temps réel.

La figure ci-après illustre l'environnement CCS IDE pour le développement de projets, et tous les outils nécessaires pour construire une application à base des DSP de TI. [40]

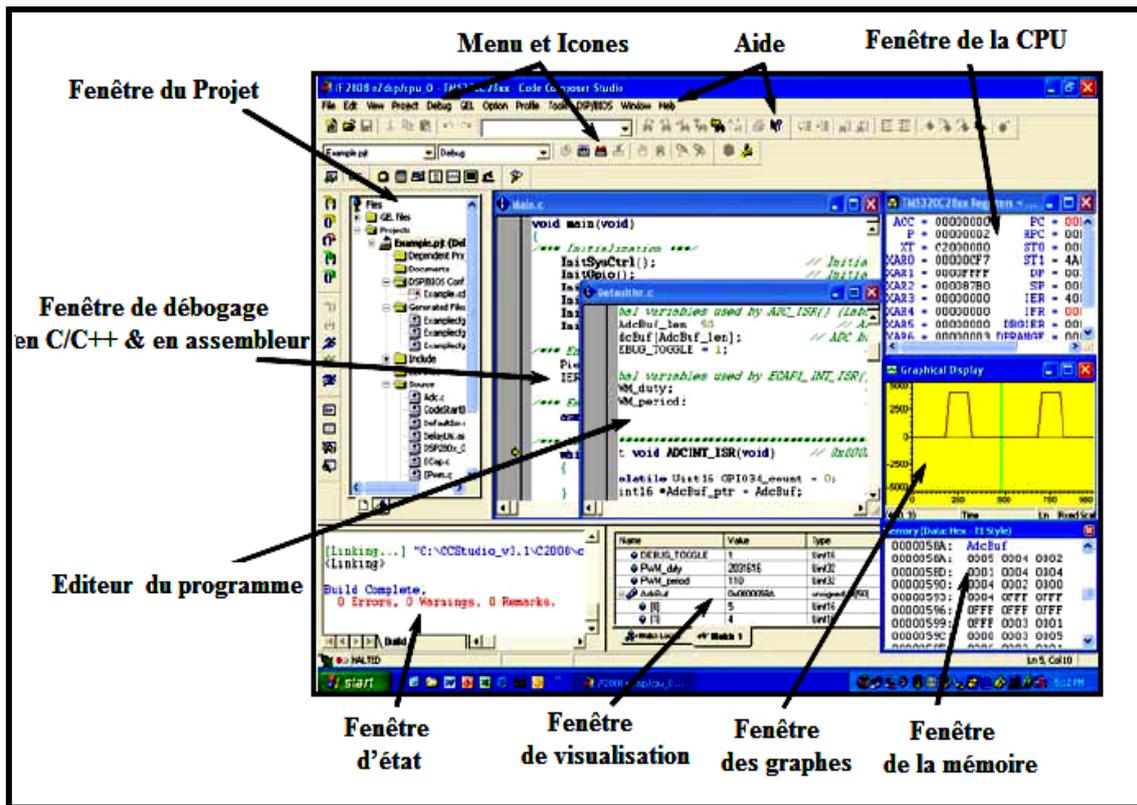


Figure 2.18 Environnement de développement Code Composer Studio.

La programmation avec CCS IDE utilise une structure de projets. Avec CCS on peut créer un projet (Nom.pjt) qui peut sauvegarder toutes les informations nécessaires pour construire (build) le fichier exécutable (Nom.out). Le projet (Nom.pjt) peut comporter, des fichiers source (C et/ou Assembleur), des fichiers d'entête (.h), des fichiers d'organisation de la mémoire de la cible (.cmd), et des options pour la construction du programme. Après compilation et construction du projet, un fichier d'extension (.out) sera généré. Ce fichier .out contient le code exécutable à charger dans la mémoire du DSP.

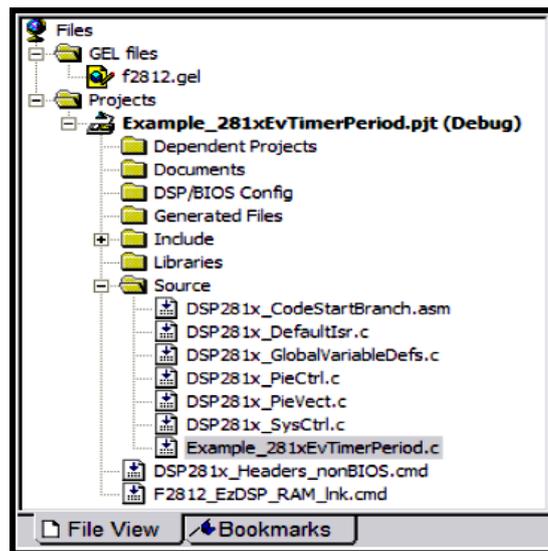


Figure 2.19 Différents fichier constituant un projet.

B. MATLAB/ Simulink

La collaboration des deux logiciels MATLAB/Simulink de MathWorks et Code Composer Studio IDE de Texas Instruments, nous permet d'effectuer la génération automatique du code programme, faire des prototypes et développer des systèmes embarqués. Avec, ce procédé, on peut développer et valider des designs de contrôle et des algorithmes DSP. Les caractéristiques de cette méthode sont : [41]

- Générer le code C en format de projet dans Code Composer Studio IDE
- Automatiser le test et l'exécution des modèles Simulink
- Permet l'évaluation en temps réel des designs du système sur la carte eZdsp
- Permet d'accéder aux périphériques internes du DSP

C. VisSim

VisSim est un logiciel graphique, pour la conception et la simulation d'application, et le développement de systèmes embarqués. L'interface visuelle de schémas fonctionnels offre une méthode simple pour construire, modifier et valider les modèles des systèmes les plus complexes. [17]. Les caractéristique de VisSim sont : [41]

- Visualisation en temps réel des variables lors de l'exécution du code par le DSP
- Branchement avec Code Composer Studio IDE pour la création automatique du projet
- Génération automatique du code C pour les périphériques: ADC, SCI, SPI, I2C, CAN, ePWM, GPIO des DSP F28xx.

VisSim supporte les DSP de Texas Instruments: F2804, F2812, F2808, F280x, Delfino (F2833x), Piccolo (F2802x/F2803x), MSP430, F281x, LF240x, F243, C5510, C6713. [17]

D. LabVIEW et Visual Basic

Pour la programmation des applications DSP et la génération du code exécutable, Il est aussi possible d'utiliser les deux produits, soit LabVIEW de National Instruments ou Visual Basic de Microsoft. [32]

Conclusion

Ainsi, on a donné un aperçu global et bref sur le monde des DSP, et on a repris plus particulièrement les éléments et fonctionnalités de base du DSP TMS 320F2812 de TI, afin de mieux exploiter ses caractéristiques et ressources internes dans la réalisation pratique de notre système qui consiste en variation de la vitesse d'un moteur asynchrone.

Chapitre III

Moteur asynchrone et Modulation vectorielle

Ce chapitre est réparti en deux grandes parties, la première est consacrée à la description et modélisation du moteur asynchrone, qui est un élément largement utilisé dans l'industrie grâce à sa simplicité de construction, sa facilité de mise en œuvre, son faible encombrement, et son bon rendement qui en font un matériel très fiable et qui demande peu d'entretien. Dans la deuxième partie, sera donnée une présentation globale de la technique de modulation vectorielle, qui est une nouvelle méthode utilisée pour la génération de signaux modulés en largeur (MLI) dans des applications de systèmes triphasés.

1. Généralités

1.1 Organisation du moteur asynchrone

Le moteur asynchrone est constitué des éléments suivants : [24] [1]

- a) **Le stator (partie fixe):** Il est constitué d'enroulements bobinés répartis dans les encoches du circuit magnétique statorique qui est constitué d'un empilage de fines tôles métalliques pour diminuer l'effet de la circulation des courants de Foucault.
 - b) **Le rotor (partie tournante):** On distingue deux types de rotors, le rotor bobiné et le rotor à cage. Le moteur à cage, est beaucoup plus aisé à construire que le moteur à rotor bobiné, par conséquent d'un prix de revient inférieur et caractérisé par une robustesse intrinsèquement plus grande. Son inconvénient majeur est au démarrage, il a de mauvaises performances (courant élevé et faible couple).
- **Rotor bobiné :** Le rotor comporte des enroulements bobinés à l'intérieur d'un circuit magnétique constitué de disques en tôle empilés sur l'arbre de la machine. Les extrémités des enroulements rotoriques sont sorties et reliées à des bagues montées sur l'arbre, sur lesquelles frottent des balais en carbone.
 - **Rotor à cage :** Le circuit du rotor est constitué de barres conductrices régulièrement réparties entre deux couronnes métalliques formant les extrémités, le tout rappelant la forme d'une cage d'écureuil. Cette cage est insérée à l'intérieur d'un circuit magnétique analogue à celui du moteur à rotor bobiné.

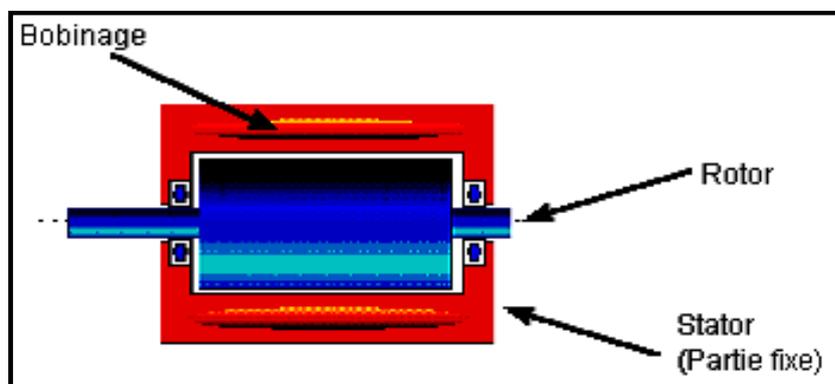


Figure 3.1 Les grandes parties constituantes du moteur asynchrone.

- c) **Les organes mécaniques:** Ils permettent le maintien des différentes parties de la machine asynchrone. Le stator autoporteur reçoit de chaque côté un flasque sur lequel le rotor sera positionné grâce à des roulements à billes ou à rouleaux. Un ventilateur est placé en bout d'arbre sur le rotor pour le refroidissement de la machine. [24]

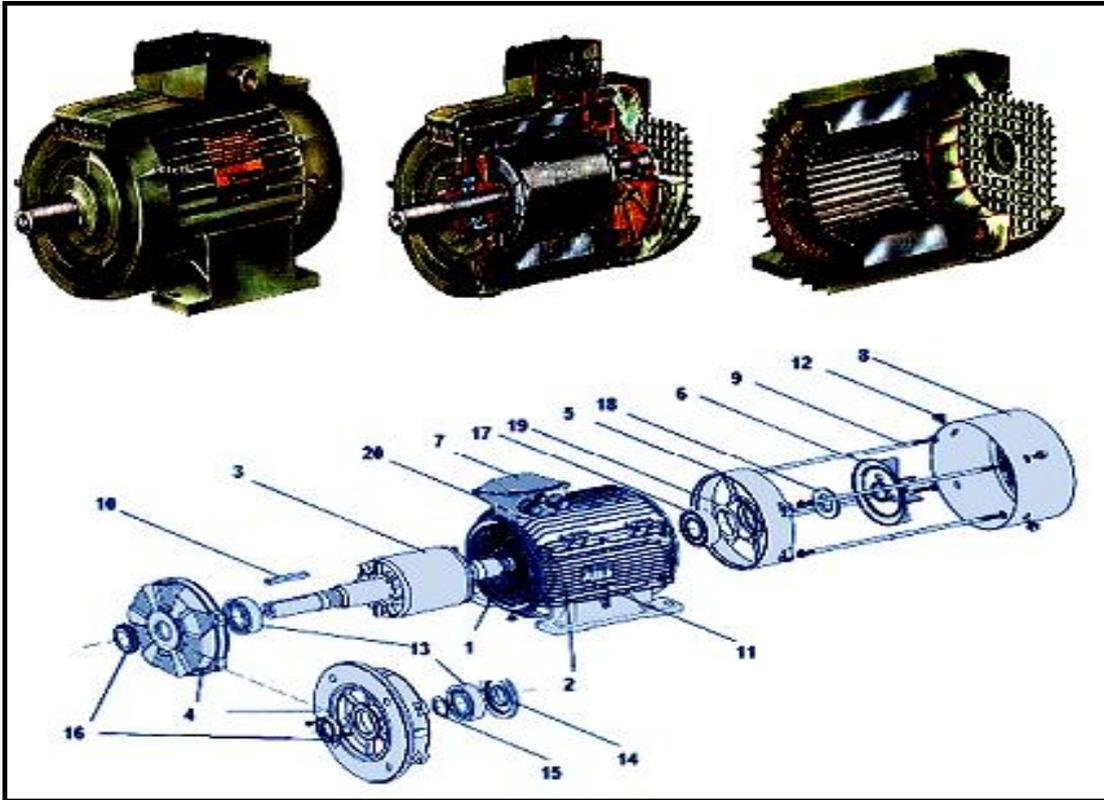


Figure 3.2 Vue en coupe de la machine asynchrone.

- | | |
|---------------------------------|--|
| 1 : Stator bobiné | 11 : Plaque signalétique |
| 2 : Carter | 12 : Vis de fixation du capot |
| 3 : Rotor | 13 : Roulement côté accouplement |
| 4 : Flasque côté accouplement | 14 : Chapeau intérieur côté accouplement |
| 5 : Flasque arrière | 15 : Circlips de roulement côté accouplement |
| 6 : Ventilateur | 16 : Joint côté accouplement |
| 7 : Couvercle de boîte à bornes | 17 : Roulement arrière |
| 8 : Capot de ventilation | 18 : Joint arrière |
| 9 : Tiges de montage | 19 : Rondelle de précharge |
| 10 : Clavette | 20 : Corps de boîte à bornes |

1.2 Principe de fonctionnement du moteur asynchrone [1] [8]

Le principe des moteurs à courant alternatif réside dans la création d'un champ magnétique tournant dans le bobinage du stator par l'utilisation de tensions alternatives. En fait, la circulation d'un courant dans le bobinage crée un champ magnétique \vec{B} dans l'axe de chaque bobine, dont la direction et l'intensité sont fonction du courant I qui la parcourt. De plus si le courant I est alternatif et périodique, le champ magnétique varie en sens et en direction à la même fréquence que celle du courant I . [1]

Dans le cas du moteur asynchrone triphasé, les trois bobines sont disposées dans le stator à 120° les unes des autres, trois champs magnétiques déphasés sont ainsi créés. Le champ magnétique résultant de la somme des trois champs est un champ tournant dont la fréquence de rotation est fonction de la fréquence du courant et du nombre de paire de pôles.

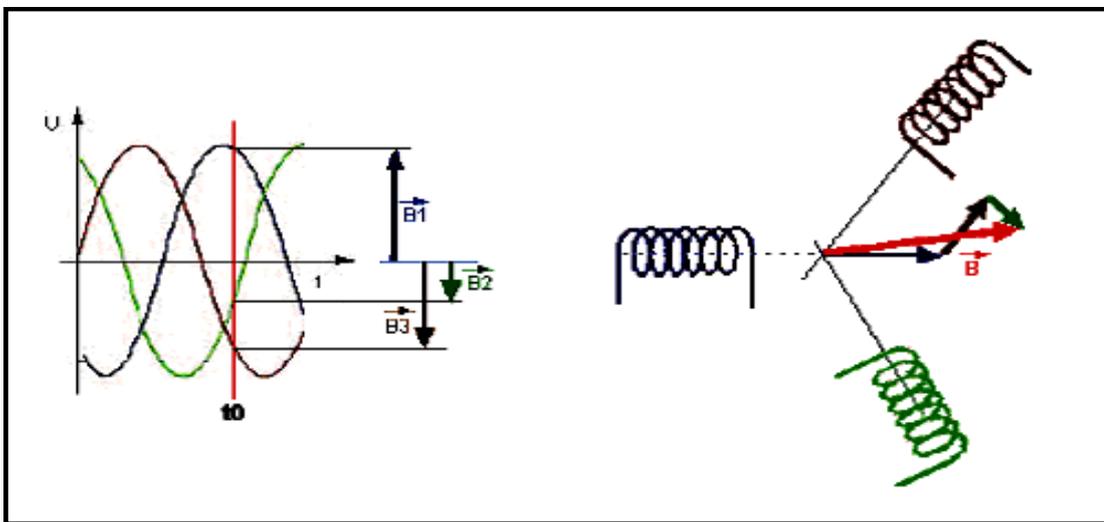


Figure 3.3 Principe du fonctionnement du moteur asynchrone MAS.

Supposons que les trois (3) enroulements statoriques du moteur asynchrone, sont alimentés par un système de tensions équilibrées de fréquence ω_s . Il va en résulter la création d'un *champ magnétique glissant* dans l'entrefer de la machine, dont la vitesse de glissement (synchronisme) par rapport au stator est: $\Omega_s = \frac{\omega_s}{p}$, où p désigne le nombre de paires de pôles du champ. [8]

Les trois enroulements statoriques créent donc un champ magnétique tournant, sa fréquence de rotation $p f_s$ est nommée *fréquence de synchronisme*.

En régime permanent le rotor tourne à une vitesse angulaire constante Ω et le champ à la vitesse de glissement (synchronisme) Ω_s , avec $\Omega < \Omega_s$. Le rotor perçoit donc un champ glissant à la vitesse relative : $\Omega_r = \Omega_s - \Omega$.

La pulsation de ce champ glissant est :

$$\omega_r = p \Omega_r = p (\Omega_s - \Omega) = p \Omega_s - p \Omega = \omega_s - p \Omega$$

$$\text{Ce qui donne : } \omega_s = \omega_r + p\Omega \quad (3.1)$$

En désignant par f la fréquence de rotation du rotor et f_s celle du champ glissant, la fréquence rotorique sera :

$$f_r = p (f_s - f) \quad (3.2)$$

Il en résulte la création dans les bobinages rotoriques une force électromotrice FEM induite de fréquence f_r . La valeur efficace de cette FEM induite est donnée par : [8]

$$E_r = \frac{\pi}{\sqrt{2}} K_{br} N_r \varphi_p f_r \quad (3.3)$$

Avec: K_{br} : Facteur de bobinage d'un enroulement rotorique.

N_r : Nombre de brins de chaque enroulement rotorique.

φ_p : Le flux sous un pôle du champ glissant.

Cette FEM induite court-circuitée sur l'enroulement rotorique va donner naissance à un courant induit. L'interaction entre ce courant et le champ glissant va donner naissance à des forces de Laplace s'exerçant sur les brins du rotor dont le moment par rapport à l'axe de rotation constituera le *couple électromagnétique* de la machine.

On remarque que ce couple n'existe que si la FEM est non nulle c'est-à-dire si $f_r \neq 0$, il en résulte que le rotor ne doit pas tourner au synchronisme avec le champ glissant pour qu'il y ait couple électromagnétique. Les deux fréquences de rotation ne peuvent pas donc être synchrones d'où le nom de *moteur asynchrone*.

1.3 Caractéristique du moteur asynchrone

Le couple électromagnétique développé par le moteur varie avec la vitesse de rotation. Le réglage de la vitesse du moteur asynchrone est donc obtenu en agissant sur le couple qu'elle produit par variation du nombre de paires de pôles, de la tension d'alimentation du stator, du glissement ou de la fréquence d'alimentation. Le rotor du moteur tourne à la vitesse angulaire pour laquelle le couple électromagnétique du moteur et le couple résistant de la charge sont identiques. [1]

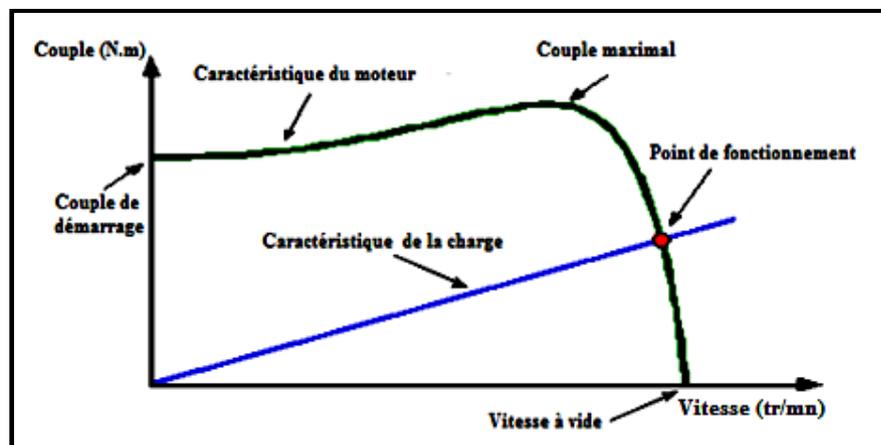


Figure 3.4 Caractéristique Couple-Vitesse du moteur asynchrone.

2. Modélisation du moteur asynchrone

2.1 Le moteur asynchrone en régime quelconque [6]

Les enroulements des trois phases statoriques et des trois phases rotoriques placés dans l'espace peuvent être représentés comme indiquée en figure 3.5. Où les phases rotoriques sont court-circuitées sur elles mêmes.

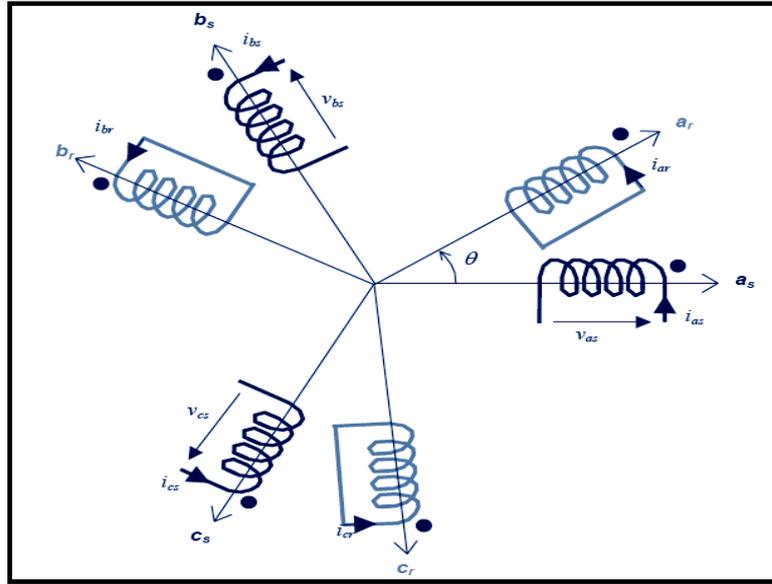


Figure 3.5 Représentation des enroulements statoriques et rotoriques.

La variable θ est l'angle électrique entre l'axe de la phase (a) statorique et la phase (a) rotorique. Lorsque la machine tourne, l'angle θ est une fonction du temps et sa dérivée Ω (la pulsation mécanique) est constante en régime permanent.

La loi des mailles permet d'écrire :

$$v = Ri + \frac{d\varphi}{dt} \quad (3.4)$$

Pour les trois phases statoriques on écrit :

$$\begin{pmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{pmatrix} = R_s \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \varphi_{as} \\ \varphi_{bs} \\ \varphi_{cs} \end{pmatrix} \quad (3.5)$$

Ou bien sous la forme matricielle condensée :

$$(v_{abcs}) = R_s (i_{abcs}) + \frac{d}{dt} (\varphi_{abcs}) \quad (3.6)$$

Le rotor étant en court-circuit, ses tensions sont nulles:

$$(v_{abcr}) = R_r (i_{abcr}) + \frac{d}{dt} (\varphi_{abcr}) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.7)$$

Avec : R_s la valeur des résistances statoriques

R_r la valeur des résistances rotoriques

Chaque flux comporte une interaction avec les courants de toutes les phases y compris la sienne (notion de flux / inductance propre). Par exemple pour la phase (a):

$$\varphi_{as} = l_s i_{as} + m_s i_{bs} + m_s i_{cs} + m_1 i_{ar} + m_3 i_{br} + m_2 i_{cr}$$

Pour toutes les phases du stator et du rotor, on a la forme matricielle:

$$\begin{pmatrix} \varphi_{as} \\ \varphi_{bs} \\ \varphi_{cs} \\ \varphi_{ar} \\ \varphi_{br} \\ \varphi_{cr} \end{pmatrix} = \begin{pmatrix} l_s & m_s & m_s & m_1 & m_3 & m_2 \\ m_s & l_s & m_s & m_2 & m_1 & m_3 \\ m_s & m_s & l_s & m_3 & m_2 & m_1 \\ \hline m_1 & m_2 & m_3 & l_r & m_r & m_r \\ m_3 & m_1 & m_2 & m_r & l_r & m_r \\ m_2 & m_3 & m_1 & m_r & m_r & l_r \end{pmatrix} \begin{pmatrix} i_{as} \\ i_{bs} \\ i_{cs} \\ i_{ar} \\ i_{br} \\ i_{cr} \end{pmatrix} \quad (3.8)$$

Avec :

l_s : L'inductance propre d'une phase statorique.

l_r : L'inductance propre d'une phase rotorique.

m_s : L'inductance mutuelle entre deux phases statoriques.

m_r : L'inductance mutuelle entre deux phases rotoriques.

$$m_1 = m_{sr} \cos(\theta)$$

$$m_2 = m_{sr} \cos\left(\theta - \frac{2\pi}{3}\right)$$

$$m_3 = m_{sr} \cos\left(\theta + \frac{2\pi}{3}\right)$$

m_{sr} : Le maximum de la mutuelle entre une phase statorique et une phase rotorique.

2.1.1 Transformation triphasé-diphasé $\alpha\beta$ [9] [6]

La transformation $\alpha\beta$ représente un outil mathématique très utile pour l'analyse et la modélisation des systèmes électriques triphasés. Le but de cette transformation, est de passer d'un système triphasé abc vers un système diphasé $\alpha\beta$. [9]

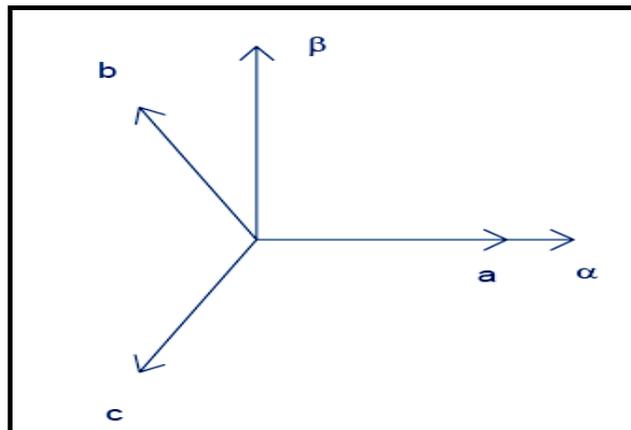


Figure 3.6 Les repères triphasé abc et diphasé $\alpha\beta$.

Il existe principalement deux transformations triphasé-diphasé : la transformation de Clarke et la transformation de Concordia : [6]

<i>Transformation de Concordia</i>	<i>Transformation de Clarke</i>
<i>Passage du système triphasé abc vers le système diphasé αβ</i>	
$\begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} \xrightarrow{T_{23}} \begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} : (x_{\alpha\beta}) = T_{23}(x_{abc})$ <p>Avec : $T_{23} = \sqrt{\frac{2}{3}} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$</p>	$\begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} \xrightarrow{T'_{23}} \begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} : (x_{\alpha\beta}) = T'_{23}(x_{abc})$ <p>Avec : $T'_{23} = \frac{2}{3} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$</p>
<i>Passage du système diphasé αβ vers le système triphasé abc</i>	
$\begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} \xrightarrow{T_{32}} \begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} : (x_{abc}) = T_{32}(x_{\alpha\beta})$ <p>Avec : $T_{32} = \sqrt{\frac{2}{3}} \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$</p>	$\begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} \xrightarrow{T'_{32}} \begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} : (x_{abc}) = T'_{32}(x_{\alpha\beta})$ <p>Avec : $T'_{32} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$</p>

Tableau 3.1 Les transformations de Concordia et Clarke.

L'application de la transformation de Concordia aux équations de la machine permet de réduire le système de trois équations en un système de deux équations :

$$\begin{aligned} (v_{\alpha\beta s}) &= T_{23}(v_{abcs}) = T_{23}(R_s(i_{abcs}) + \frac{d}{dt}(\varphi_{abcs})) = R_s T_{23}(i_{abcs}) + \frac{d}{dt} T_{23}(\varphi_{abcs}) \\ (v_{\alpha\beta s}) &= R_s(i_{\alpha\beta s}) + \frac{d}{dt}(\varphi_{\alpha\beta s}) \end{aligned} \quad (3.9)$$

De même pour le rotor, on écrit :

$$(v_{\alpha\beta r}) = R_r(i_{\alpha\beta r}) + \frac{d}{dt}(\varphi_{\alpha\beta r}) \quad (3.10)$$

Ainsi que pour l'écriture des flux en fonction des courants :

$$\begin{pmatrix} \varphi_{\alpha s} \\ \varphi_{\beta s} \\ \varphi_{\alpha r} \\ \varphi_{\beta r} \end{pmatrix} = \begin{pmatrix} L_s & 0 & M \cdot P(\theta) \\ 0 & L_s & \\ M \cdot P(-\theta) & L_r & 0 \\ 0 & L_r & \end{pmatrix} \begin{pmatrix} i_{\alpha s} \\ i_{\beta s} \\ i_{\alpha r} \\ i_{\beta r} \end{pmatrix} \quad (3.11)$$

L_s , L_r et M : Sont dites des inductances cycliques, avec :

$$\begin{aligned} L_s &= l_s - m_s \\ L_r &= l_r - m_r \\ M &= \frac{3}{2} m_{sr} \end{aligned}$$

$P(\theta)$: est La matrice de rotation, avec :

$$P(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Ainsi, on dispose d'une modélisation pour chaque partie du moteur asynchrone (stator et rotor) dans deux repères séparés. Les grandeurs statoriques sont exprimées dans le repère $\alpha\beta s$ du stator et les grandeurs rotoriques sont exprimées dans le repère $\alpha\beta r$ du rotor.

Mais, pour la modélisation du moteur asynchrone, on a besoin d'exprimer toutes les équations dans le même repère (repère commun). Pour réaliser ceci, on procède à transformer les grandeurs statoriques et les grandeurs rotoriques vers un repère commun dit dq , et ceci à l'aide de deux transformations (qui sont des rotations dans le plan). Ces deux transformations (rotations) avec la transformation $\alpha\beta$ constituent la transformation dite de Park qu'on verra dans la section suivante.

2.1.2 La transformation de Park [9] [6]

L'idée derrière la transformation de Park est assez simple. En effet, au lieu de mettre le système triphasé dans un référentiel à deux axes fixes, cette transformation met le système dans un repère à deux axes et tournant. Ceci signifie, pratiquement, le passage d'une transformation statique à une transformation dynamique dont les coefficients sont variables.

On peut se référer à la figure 3.7 pour obtenir une idée sur la transformation de Park. La transformation définit un nouveau groupe d'axes de référence, appelés d et q , qui tournent autour du repère statique $\alpha\beta$ à une vitesse angulaire ω constante. [9]

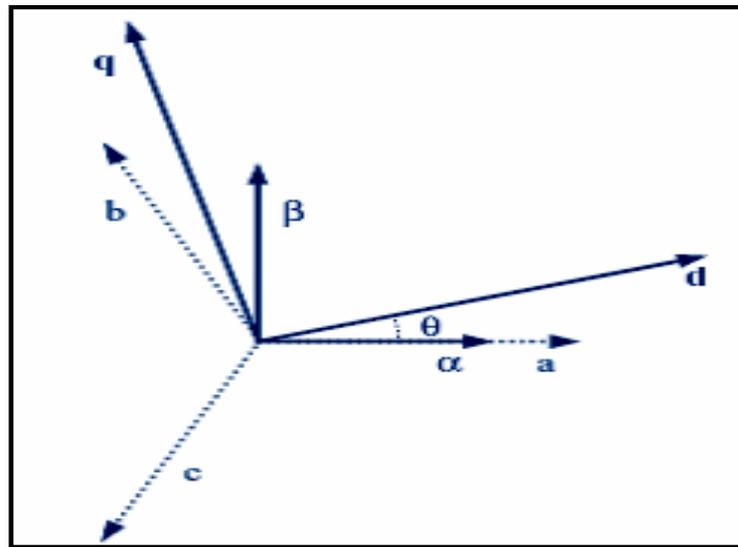


Figure 3.7 Le repère de la transformation de Park.

La transformation de Park est constituée d'une transformation triphasé-diphasé suivie d'une rotation. Elle permet de passer du repère abc vers le repère $\alpha\beta$ puis vers le repère dq . le repère $\alpha\beta$ est toujours fixe par rapport au repère abc figure 3.6, par contre le repère dq est mobile. Il forme avec le repère fixe $\alpha\beta$ un angle qui est appelé l'angle de la transformation de Park ou angle de Park.

Si l'on note par θ_s l'angle de transformation de Park statorique et par θ_r l'angle de transformation de Park rotorique, on peut simplifier les équations et le modèle final en liant les angles θ_s et θ_r par la relation :

$$\theta_s = \theta + \theta_r \quad (3.12)$$

Pour les grandeurs statoriques et rotoriques, on écrit :

$$(x_{\alpha\beta s}) = P(\theta_s)(x_{dq s}) \quad (3.13)$$

$$(x_{\alpha\beta r}) = P(\theta_r)(x_{dq r}) \quad (3.14)$$

$$\text{Avec : } P(\theta_s) = \begin{pmatrix} \cos \theta_s & -\sin \theta_s \\ \sin \theta_s & \cos \theta_s \end{pmatrix} \text{ et } P(\theta_r) = \begin{pmatrix} \cos \theta_r & -\sin \theta_r \\ \sin \theta_r & \cos \theta_r \end{pmatrix}$$

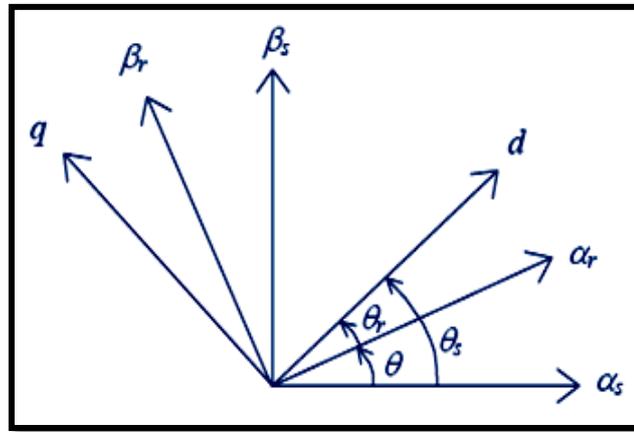


Figure 3.8 Principe de la transformation de Park.

2.1.3 Expressions des tensions

Les équations des tensions, dans le repère de Park deviennent :

$$\text{Pour le stator : } \begin{pmatrix} v_{ds} \\ v_{qs} \end{pmatrix} = R_s \begin{pmatrix} i_{ds} \\ i_{qs} \end{pmatrix} + \dot{\theta}_s P\left(\frac{\pi}{2}\right) \begin{pmatrix} \varphi_{ds} \\ \varphi_{qs} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \varphi_{ds} \\ \varphi_{qs} \end{pmatrix}$$

$$\text{Pour le rotor : } \begin{pmatrix} v_{dr} \\ v_{qr} \end{pmatrix} = R_r \begin{pmatrix} i_{dr} \\ i_{qr} \end{pmatrix} + \dot{\theta}_r P\left(\frac{\pi}{2}\right) \begin{pmatrix} \varphi_{dr} \\ \varphi_{qr} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \varphi_{dr} \\ \varphi_{qr} \end{pmatrix}$$

$$\text{Ou bien : } \begin{cases} v_{ds} = R_s i_{ds} - \dot{\theta}_s \varphi_{qs} + \frac{d\varphi_{ds}}{dt} \\ v_{qs} = R_s i_{qs} + \dot{\theta}_s \varphi_{ds} + \frac{d\varphi_{qs}}{dt} \\ v_{dr} = R_r i_{dr} - \dot{\theta}_s \varphi_{qs} + \frac{d\varphi_{dr}}{dt} = 0 \\ v_{qr} = R_r i_{qr} + \dot{\theta}_r \varphi_{qr} + \frac{d\varphi_{qr}}{dt} = 0 \end{cases} \quad (3.15)$$

Avec : $\dot{\theta}_s$ et $\dot{\theta}_r$ sont les dérivées des angles des transformations de Park des grandeurs statoriques et rotoriques respectivement.

2.1.4 Expressions des flux

Les flux statoriques et rotoriques peuvent se mettre sous la forme matricielle:

$$\begin{pmatrix} \varphi_{ds} \\ \varphi_{qs} \\ \varphi_{dr} \\ \varphi_{qr} \end{pmatrix} = \begin{pmatrix} L_s & 0 & M & 0 \\ 0 & L_s & 0 & M \\ M & 0 & L_r & 0 \\ 0 & M & 0 & L_r \end{pmatrix} \begin{pmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{pmatrix}$$

Ou bien sous la forme d'un système d'équations:

$$\begin{cases} \varphi_{ds} = L_s i_{ds} + M i_{dr} \\ \varphi_{qs} = L_s i_{qs} + M i_{qr} \\ \varphi_{dr} = M i_{ds} + L_r i_{dr} \\ \varphi_{qr} = M i_{qs} + L_r i_{qr} \end{cases} \quad (3.16)$$

On constate, dans la nouvelle écriture de la matrice des inductances qu'elle ne dépend plus de l'angle électrique θ entre le stator et le rotor.

2.1.5 Expressions du couple électromagnétique

a) La puissance instantanée

La puissance instantanée de la machine s'écrit ainsi: [6] [8]

$$\begin{aligned} P_E &= v_{ds} i_{ds} + v_{qs} i_{qs} \\ P_E &= \left(R_s i_{ds} - \omega_s \varphi_{qs} + \frac{d\varphi_{ds}}{dt} \right) i_{ds} + \left(R_s i_{qs} + \omega_s \varphi_{ds} + \frac{d\varphi_{qs}}{dt} \right) i_{qs} \\ P_E &= (R_s i_{ds}^2 + R_s i_{qs}^2) + \left(\frac{d\varphi_{ds}}{dt} i_{ds} + \frac{d\varphi_{qs}}{dt} i_{qs} \right) + \omega_s (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) \end{aligned} \quad (3.17)$$

- Le premier terme $\{R_s i_{ds}^2 + R_s i_{qs}^2\}$ représente les pertes joules.
- Le second terme $\{\frac{d\varphi_{ds}}{dt} i_{ds} + \frac{d\varphi_{qs}}{dt} i_{qs}\}$ correspond à de la puissance électromagnétique stockée dans le champ tournant.
- Le dernier terme $\{\omega_s (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds})\}$ représente la puissance électrique transformée en puissance mécanique, sachant que les pertes fer ont été négligées.

b) Le couple instantané

La puissance mécanique peut se mettre sous la forme :

$$\begin{aligned} P_M &= C_E * \Omega_E = \omega_s (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) \\ C_E &= \frac{\omega_s}{\Omega_E} (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) \end{aligned}$$

D'où l'expression du couple :

$$C_E = p (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) \quad (3.18)$$

Avec : p le nombre de paires de pôles.

c) Autres expressions du couple

Il est possible d'obtenir d'autres expressions pour le couple instantané en utilisant les expressions des flux statoriques :

$$\begin{aligned} C_E &= p \left((L_s i_{ds} + M i_{dr}) i_{qs} - (L_s i_{qs} + M i_{qr}) i_{ds} \right) \\ C_E &= pM (i_{dr} i_{qs} - i_{qr} i_{ds}) \end{aligned} \quad (3.19)$$

Ou bien encore en faisant appel aux expressions des flux rotoriques :

$$\begin{aligned} \varphi_{dr} &= M i_{ds} + L_r i_{dr} \Rightarrow i_{dr} = \frac{\varphi_{dr}}{L_r} + \frac{M}{L_r} i_{ds} \\ \varphi_{qr} &= M i_{qs} + L_r i_{qr} \Rightarrow i_{qr} = \frac{\varphi_{qr}}{L_r} + \frac{M}{L_r} i_{qs} \end{aligned}$$

Ce qui donne :

$$C_E = p \frac{M}{L_r} (\varphi_{dr} i_{qs} - \varphi_{qr} i_{ds}) \quad (3.20)$$

Dans les expressions du couple électromagnétique citées ci-dessus, on constate que ce couple résulte de l'interaction d'un terme de flux et d'un terme de courant. C'est là où réside la difficulté du contrôle de la machine asynchrone c'est-à-dire pouvoir contrôler le flux et le courant indépendamment l'un de l'autre. [8]

2.1.6 Choix du repère dq [6]

Pour la sélection du repère dq , on a trois choix intéressants. On peut fixer le repère dq au stator, au rotor ou au champ tournant.

❖ Le cas du référentiel lié au champ tournant

Choisissons de fixer le repère dq au champ tournant. En effet l'avantage d'utiliser ce référentiel, est d'avoir des grandeurs constantes en régime permanent comme le montre l'exemple ci-dessous : [6]

Supposons que le stator du moteur asynchrone est alimenté par le système de tensions triphasé équilibré suivant :

$$\begin{cases} v_{as} = V_s \cos(\omega_s t) \\ v_{bs} = V_s \cos\left(\omega_s t - \frac{2\pi}{3}\right) \\ v_{cs} = V_s \cos\left(\omega_s t - \frac{4\pi}{3}\right) \end{cases} \quad (3.21)$$

Notons par $\omega_s = \dot{\theta}_s$ la pulsation du champ statorique, par $\omega_r = \dot{\theta}_r$ la pulsation du champ rotorique et par $\omega = \omega_s - \omega_r = \dot{\theta} = p \Omega$ la pulsation mécanique du rotor.

Le repère de Park dq étant lié au champ tournant, on peut écrire : $\theta_s = \omega_s t$.

On a la relation suivante:

$$\begin{aligned} (x_{\alpha\beta s}) &= P(\theta_s)(x_{dq s}) \Rightarrow (x_{dq s}) = P^{-1}(\theta_s)(x_{\alpha\beta s}) \\ &\Rightarrow (x_{dq s}) = (P^{-1}(\theta_s) T_{23})(x_{abc}) \end{aligned}$$

Sachant que :

$$(P^{-1}(\theta_s) T_{23}) = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos \theta_s & \sin \theta_s \\ -\sin \theta_s & \cos \theta_s \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$$

$$(P^{-1}(\theta_s) T_{23}) = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos \theta_s & \cos(\theta_s - \frac{2\pi}{3}) & \cos(\theta_s - \frac{4\pi}{3}) \\ -\sin \theta_s & -\sin(\theta_s - \frac{2\pi}{3}) & -\sin(\theta_s - \frac{4\pi}{3}) \end{pmatrix}$$

Après calculs, et remplacement dans les expressions des tensions statoriques on aboutit à :

$$\begin{cases} v_{ds} = V_s \cos(\omega_s t - \theta_s) \\ v_{qs} = V_s \sin(\omega_s t - \theta_s) \end{cases} \quad (3.22)$$

Mais comme, $\theta_s = \omega_s t$, on obtiendra :

$$\begin{cases} v_{ds} = V_s \\ v_{qs} = 0 \end{cases} \quad (3.23)$$

Dans ce dernier résultat on constate que les composantes de Park v_{ds} et v_{qs} sont des constantes en régime sinusoïdal forcé lorsque l'on prend pour θ_s l'angle de synchronisme $\omega_s t$. Il est alors plus aisé d'en faire la régulation.

2.2 Modèle du moteur asynchrone en régime permanent [6]

Afin de modéliser la machine asynchrone par un schéma électrique équivalent, on introduit la notation des nombres complexes $\bar{X} = x_d + jx_q$ qui permettra d'alléger les calculs.

On écrit, alors :

$$\bar{V}_s = v_{ds} + jv_{qs} = (R_s i_{ds} - \omega_s \varphi_{qs} + \frac{d\varphi_{ds}}{dt}) + j(R_s i_{qs} + \omega_s \varphi_{ds} + \frac{d\varphi_{qs}}{dt})$$

$$\bar{V}_s = R_s (i_{ds} + j i_{qs}) - \omega_s (-\varphi_{ds} + j\varphi_{qs}) + \frac{d}{dt} (\varphi_{ds} + j\varphi_{qs})$$

$$\bar{V}_s = R_s (i_{ds} + j i_{qs}) + \frac{d}{dt} (\varphi_{ds} + j\varphi_{qs}) + j \omega_s (\varphi_{ds} + j\varphi_{qs})$$

$$\text{D'où : } \bar{V}_s = R_s \bar{I}_s + \frac{d}{dt} \bar{\varphi}_s + j \omega_s \bar{\varphi}_s$$

Mais comme, le flux est constant en régime permanent, on aura :

$$\bar{V}_s = R_s \bar{I}_s + j \omega_s \bar{\varphi}_s \quad (3.24)$$

Les mêmes calculs, pour le rotor donneront :

$$\bar{V}_r = R_r \bar{I}_r + j \omega_r \bar{\varphi}_r = 0 \quad (3.25)$$

En introduisant le terme du glissement g qui est défini comme suit :

$$g = (\omega_s - \omega) / \omega_s \quad \text{ou} \quad \omega_r = g \omega_s.$$

Et en remplaçant les termes des flux par :

$$\begin{cases} \bar{\varphi}_s = L_s \bar{I}_s + M \bar{I}_r \\ \bar{\varphi}_r = M \bar{I}_s + L_r \bar{I}_r \end{cases}$$

On aura:

$$\begin{cases} \bar{V}_s = R_s \bar{I}_s + jL_s \omega_s \bar{I}_s + jM \omega_s \bar{I}_r \\ \bar{V}_r = \frac{R_r}{g} \bar{I}_r + jL_r \omega_s \bar{I}_r + jM \omega_s \bar{I}_s = 0 \end{cases} \quad (3.26)$$

Ainsi on obtient, à partir de ces deux relations, le schéma électrique équivalent par phase du moteur asynchrone en régime permanent, illustré sur la figure 3.9.

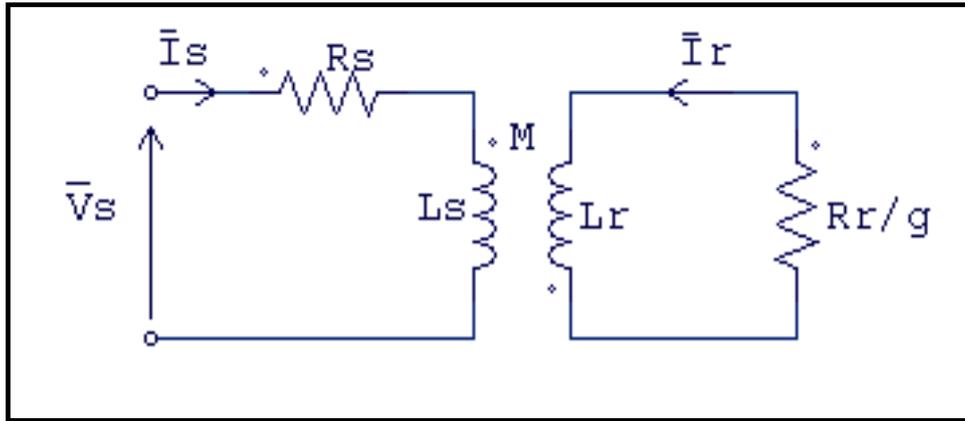


Figure 3.9 Schéma électrique par phase en régime permanent.

Il est possible de modéliser le moteur asynchrone autrement (figure 3.10) en ramenant tout le circuit au stator. Pour ce faire on pose :

$$\begin{aligned} N_r &= L_r \sigma = L_r - \frac{M^2}{L_s} \\ N_r' &= N_r \left(\frac{L_s}{M} \right)^2 \\ R_r' &= R_r \left(\frac{L_s}{M} \right)^2 \\ \bar{I}_r' &= \bar{I}_r \left(\frac{M}{L_s} \right) \end{aligned}$$

A partir des équations du schéma de la figure 3.9, on peut écrire :

Pour le stator :

$$\bar{V}_s = R_s \bar{I}_s + jL_s \omega_s \bar{I}_s + jL_s \omega_s \bar{I}_r' = \bar{V}_s = R_s \bar{I}_s + jL_s \omega_s (\bar{I}_s + \bar{I}_r')$$

Pour le rotor :

$$\begin{aligned} \bar{V}_r &= \left(\frac{R_r}{g} \bar{I}_r + jL_r \omega_s \bar{I}_r + jM \omega_s \bar{I}_s \right) \left(\frac{L_s}{M} \right) = 0 \\ 0 &= \frac{R_r}{g} \frac{L_s}{M} \bar{I}_r + jL_r \omega_s \frac{L_s}{M} \bar{I}_r + jL_s \omega_s (\bar{I}_s + \bar{I}_r' - \bar{I}_r') \end{aligned}$$

$$\begin{aligned}
 0 &= \frac{R_r}{g} \left(\frac{L_s}{M}\right)^2 \frac{M}{L_s} \bar{I}_r + jL_r \omega_s \left(\frac{L_s}{M}\right)^2 \frac{M}{L_s} \bar{I}_r - jL_s \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \\
 0 &= \frac{R_r}{g} \left(\frac{L_s}{M}\right)^2 \frac{M}{L_s} \bar{I}_r + jL_r \omega_s \left(\frac{L_s}{M}\right)^2 \frac{M}{L_s} \bar{I}_r - jL_s \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \\
 0 &= \frac{R'_r}{g} \bar{I}'_r + jL_r \omega_s \frac{N'_r}{N_r} \bar{I}'_r - jL_s \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \\
 0 &= \frac{R'_r}{g} \bar{I}'_r + \left(\frac{L_r}{N_r} - \frac{L_s}{N_r}\right) jN'_r \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \\
 0 &= \frac{R'_r}{g} \bar{I}'_r + jN'_r \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r)
 \end{aligned}$$

Avec :

$$\left(\frac{L_r}{N_r} - \frac{L_s}{N'_r}\right) = \frac{L_r}{N_r} - \frac{M^2}{N_r L_s} = \frac{L_r}{N_r} - \frac{M^2}{\frac{N_r}{L_r} L_r L_s} = \frac{1}{\sigma} - \frac{M^2}{\sigma L_r L_s} = \frac{1}{\sigma} - \frac{1}{\sigma} (1 - \sigma) = 1$$

$$\sigma = 1 - \frac{M^2}{L_r L_s} \text{ (Le coefficient de dispersion de Blondel)}$$

Le coefficient de dispersion de Blondel permet d'estimer les fuites de flux du moteur. Typiquement, on a : $\sigma \leq 10\%$. [8]

Ainsi on obtient les équations du schéma de la figure 3.10:

$$\begin{cases} \bar{V}_s = R_s \bar{I}_s + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \\ 0 = \frac{R'_r}{g} \bar{I}'_r + jN'_r \omega_s \bar{I}'_r + jL_s \omega_s (\bar{I}_s + \bar{I}'_r) \end{cases} \quad (3.27)$$

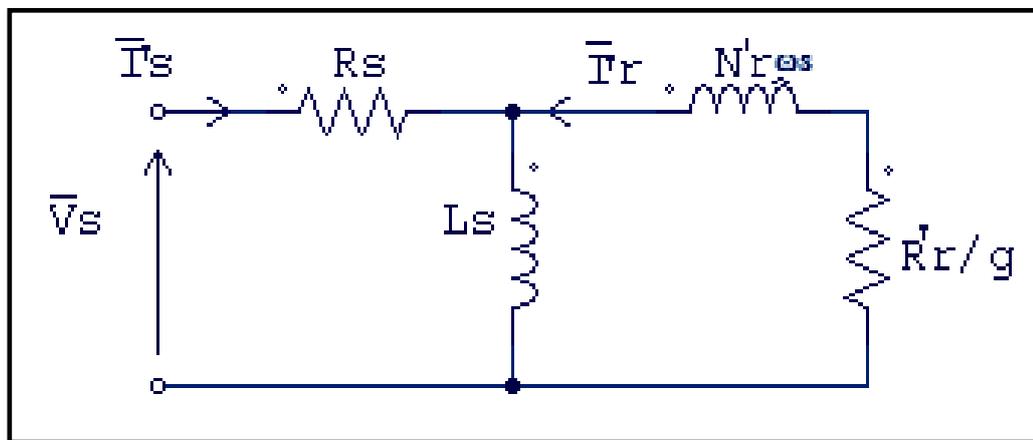


Figure 3.10 Schéma électrique par phase ramené au stator en régime permanent.

En régime permanent le couple électromagnétique peut s'écrire, ainsi :

$$C_e = \frac{P_m}{\Omega} = 3 \frac{p}{\omega_s} \frac{R_r'}{g} I_r'^2$$

Avec :

$$\overline{I_r'} = \frac{\overline{V_s} - R_s \overline{I_s}}{\frac{R_r'}{g} + j N_r' \omega_s}$$

Si on néglige la résistance statorique R_s , on aura une écriture plus simple du couple :

$$C_e = 3 \frac{p}{\omega_s} V_s^2 \frac{\frac{R_r'}{g}}{\left(\frac{R_r'}{g}\right)^2 + (N_r' \omega_s)^2} \quad (3.28)$$

Le couple est maximum à : $\left(\frac{R_r'}{g}\right)^2 = (N_r' \omega_s)^2$

C'est-à-dire quand : $g = g_m = \frac{R_r'}{N_r' \omega_s}$

On obtiendra alors : $C_{max} = \frac{3}{2} \frac{p}{N_r'} \left(\frac{V_s}{\omega_s}\right)^2$ (3.29)

Pour un glissement faible le couple varie linéairement en fonction de celui-ci suivant l'expression:

$$C_e = 3 \frac{p}{\omega_s} V_s^2 \frac{g}{R_r'} \quad (3.30)$$

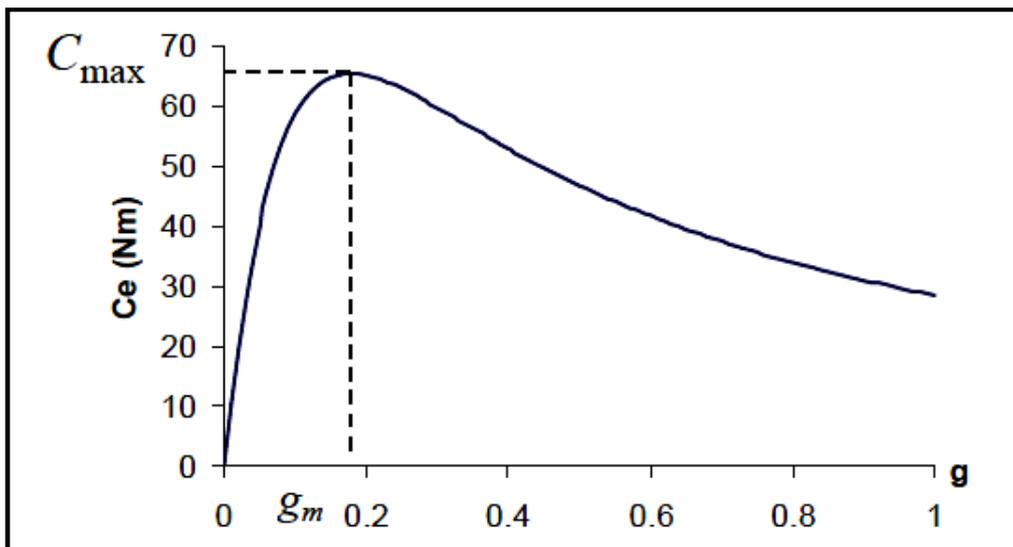


Figure 3.11 Caractéristique du couple électromagnétique en fonction du glissement.

3. Commande du moteur asynchrone

Dans la technologie de la commande des moteurs asynchrones, on rencontre deux types de commandes, la commande scalaire et la commande vectorielle. [23] [6]

	Commande scalaire	Commande vectorielle
Les spécificités de la commande	<ul style="list-style-type: none"> • Contrôle des grandeurs en amplitude. • Contrôle de flux en U/f constant. • Entraînement en boucle ouverte ou fermée. 	<ul style="list-style-type: none"> • Contrôle des grandeurs en amplitude et en phase. • Régulation indirecte du couple. • Entraînement en boucle fermée.
Les avantages	<ul style="list-style-type: none"> ✓ Faible encombrement. ✓ Robustesse. ✓ Coût réduit et maintenance réduite. ✓ Simplicité du fait de l'absence de capteurs. ✓ Simple à implanter 	<ul style="list-style-type: none"> ✓ Faible encombrement. ✓ Robustesse. ✓ Maintenance réduite. ✓ Temps de réponse courts en régulation de couple. ✓ Excellente précision en régulation de vitesse et de position. ✓ Couple très élevé à vitesse nulle.
Les inconvénients	<ul style="list-style-type: none"> ○ Les paramètres du moteur ne sont pas pris en compte. ○ Pas de contrôle, ni de régulation de couple. ○ Temps de réponse plus longs. 	<ul style="list-style-type: none"> ○ Utilisation obligatoire de capteurs. ○ Technique coûteuse. ○ Exige des calculateurs puissants comme le DSP.

Tableau 3.2 Caractéristiques de la commande scalaire et de la commande vectorielle.

3.1 Commande scalaire [6]

Plusieurs commandes scalaires existent, selon que l'on agit sur le courant ou sur la tension. Elles dépendent en principe de la configuration du convertisseur utilisé (onduleur de tension ou de courant). Dans le cas de l'onduleur de tension, qui est maintenant le plus exploité en petite et moyenne puissance, on trouve le contrôle V/f .

❖ Contrôle en V/f constant du moteur asynchrone

Son principe est de maintenir le rapport $V/f = \text{constant}$, ce qui signifie, garder le flux constant. Le contrôle du couple se fait par l'action sur le glissement.

D'après le modèle établi en régime permanent, le couple maximum s'écrit :

$$C_{max} = \frac{3}{2} \frac{p}{N_r} \left(\frac{V_s}{\omega_s} \right)^2$$

On voit bien que le couple est directement proportionnel au carré du rapport de la tension sur la fréquence statorique. Donc on peut maintenir le couple maximale C_{max} constant en maintenant ce rapport constant.

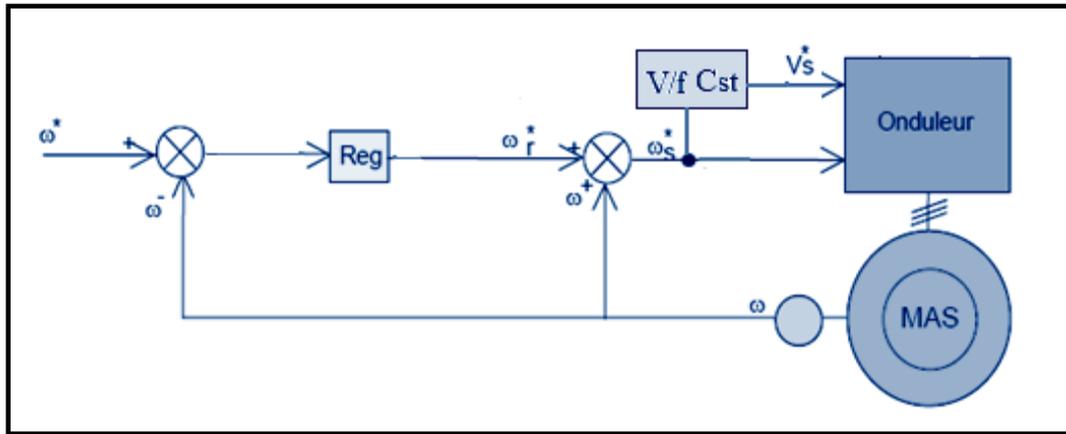


Figure 3.12 Contrôle scalaire en tension et fréquence du MAS.

Le schéma de commande ci-dessus (figure 3.12) présente la manière de réguler la vitesse du moteur asynchrone en reconstituant la pulsation statorique à partir de la vitesse et de la pulsation rotorique. Cette dernière, est issue du régulateur de vitesse. Si le moteur est chargée, la vitesse a tendance à baisser, le régulateur va fournir donc plus de couple afin d'assurer cet équilibre. La pulsation statorique est donc modifiée pour garder cet équilibre. La tension est calculée de manière à garantir le mode de contrôle en V/f constant.

3.2 Commande vectorielle [6]

Pour le contrôle de la machine asynchrone plusieurs stratégies sont envisageables. Parmi elles on rencontre la commande « Indirecte par Orientation du Flux Rotorique (IRFO) ». L'objectif de cette commande est d'orienter le flux pour qu'il n'ait qu'une seule composante (sur l'axe (d) par exemple). [6]

❖ Le principe de la commande vectorielle (IRFO).

On a vu que le couple électromagnétique C_e de la machine asynchrone peut s'écrire dans le repère dq en fonction des courants et des flux comme suit:

$$C_e = P \frac{M}{L_r} (\varphi_{dr} i_{qs} - \varphi_{qr} i_{ds})$$

Dans cette expression, on s'aperçoit que si l'on élimine le deuxième produit ($\varphi_{qr} i_{ds}$), alors le couple ressemblerait fort à celui d'un moteur MCC où le découplage entre le courant et le flux est naturel. Il suffit, pour ce faire, d'orienter le repère dq de manière à annuler la

composante de flux en quadrature φ_{qr} . C'est-à-dire, de choisir convenablement l'angle de rotation de Park de sorte que le flux rotorique soit entièrement porté sur l'axe direct (d) et donc avoir $\varphi_{qr} = 0$. Ainsi on aura : $\varphi_r = \varphi_{dr}$. (Figure 3.13).

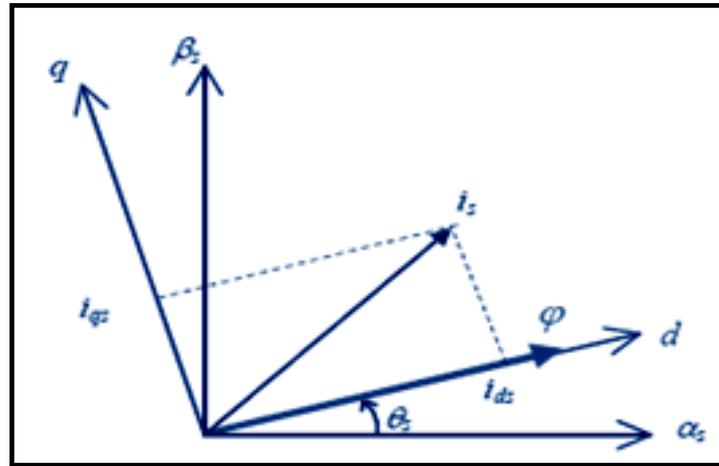


Figure 3.13 Principe du contrôle vectoriel.

Le couple s'écrit alors, ainsi:

$$C_e = P \frac{M}{L_r} \varphi_r i_{qs} \tag{3.31}$$

La stratégie de contrôle consistera donc à contrôler de façon indépendante le terme de flux φ_r et le terme de courant i_{qs} pour imposer le couple. La figure 3.14 représente le schéma de contrôle vectoriel de la machine asynchrone avec régulation de vitesse et régulation des deux courants i_{ds} et i_{qs} . Ces deux courants sont régulés par deux boucles de courants dont les sorties sont les tensions de références v_{ds}^* et v_{qs}^* dans le repère dq .

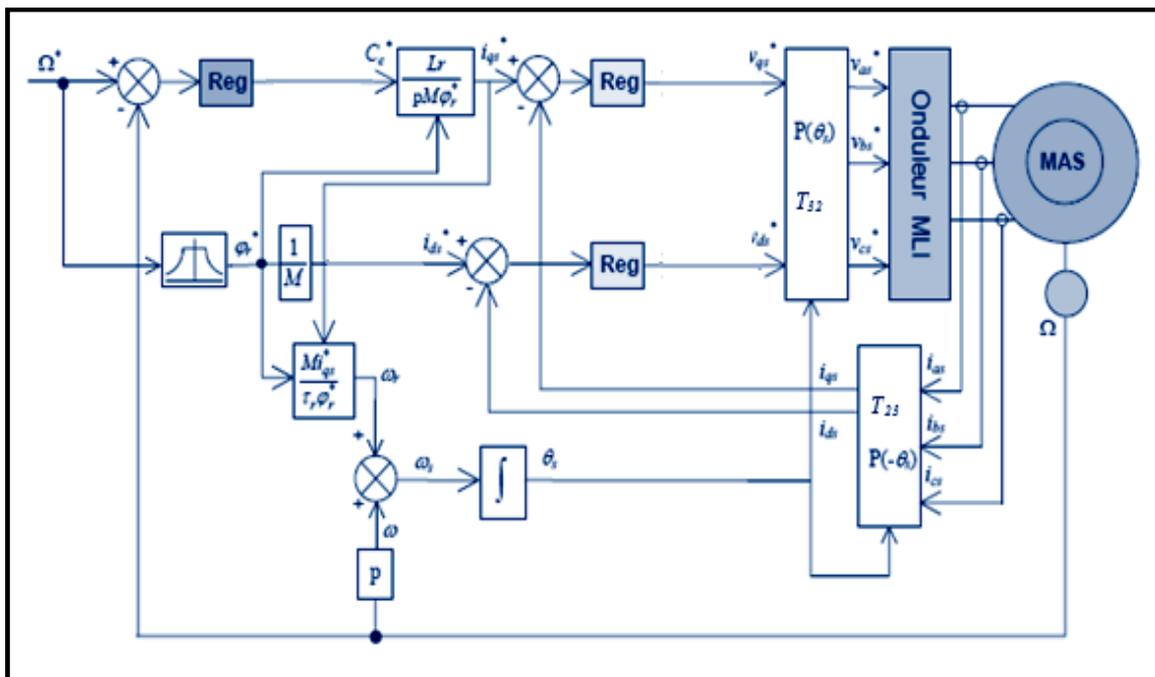


Figure 3.14 Schéma de régulation de la vitesse du MAS en IRFO.

Dans ce schéma (Figure 3.14), le flux est régulé en agissant sur la composante i_{ds} du courant statorique et on régule le couple en agissant sur la composante i_{qs} . Le régulateur du courant i_{ds} s'occupe de maintenir le courant i_{ds} constant et égal à la référence i_{ds}^* (i_{ds}^* : i_{ds} référence).

- a) **Le régulateur de vitesse** : prend en entrée la vitesse de référence et la vitesse mesurée, ensuite il agit sur le couple pour réguler la vitesse.
- b) **Le régulateur de courant i_{qs}** : Il prend en entrée le courant de référence i_{qs}^* et sa mesure et il agit sur la tension de référence v_{qs}^* pour ajuster le courant i_{qs} .
- c) **Le régulateur de courant i_{ds}** : Il prend en entrée le courant de référence i_{ds}^* et sa mesure. Il agit sur la tension de référence v_{ds}^* .

En fait, réguler ce courant à une valeur constante, permet de garantir un flux rotorique constant, puisque on a :

$$\begin{aligned}\varphi_{dr} &= M i_{ds} + L_r i_{dr} = \varphi_r \\ R_r i_{dr} - \omega_r \varphi_{qr} + \frac{d\varphi_{dr}}{dt} &= 0 \\ \varphi_{qr} &= 0\end{aligned}$$

Ce qui donne: $\varphi_r = M i_{ds} + \frac{L_r}{R_r} \frac{d\varphi_r}{dt}$

On voit alors, qu'en régime permanent :

$$\varphi_r = M i_{ds} \quad (3.32)$$

Donc pour garantir un flux rotorique constant, il suffit de réguler le courant i_{ds} à une valeur constante.

d) Le calcul de l'angle θ_s

A partir, des équations ci-dessous :

$$\begin{aligned}\varphi_{qr} &= M i_{qs} + L_r i_{qr} = 0 \\ v_{qr} &= R_r i_{qr} + \omega_r \varphi_{dr} = 0\end{aligned}$$

On déduit l'expression de la pulsation de glissement :

$$\omega_r = \frac{M i_{qs}^*}{\tau_r \varphi_r^*} \quad (3.33)$$

Ainsi le calcul de l'angle statorique θ_s , est obtenue par :

$$\theta_s = \int \omega_s dt = \int (P\Omega + \frac{M i_{qs}^*}{\tau_r \varphi_r^*}) dt \quad (3.34)$$

4. La modulation vectorielle [19]

L'objectif de la commande à MLI (Modulation de Largeur d'Impulsion) est de diminuer les harmoniques présents dans les courants générés par l'onduleur. On peut distinguer deux groupes de commande MLI : La MLI intersective et La MLI vectorielle.

Le principe de la MLI intersective est de comparer un signal triangulaire appelé porteuse qui détermine la période de découpage à une modulante dont le fondamental est à la fréquence du réseau. Le résultat de cette comparaison est le signal MLI qui est utilisé comme fonction de commutation pour commander les interrupteurs de l'onduleur. Cette méthode permet d'obtenir, de façon simple, les temps de conduction de chaque transistor. L'intérêt de cette méthode est d'obtenir, durant une période de découpage, une tension en sortie de l'onduleur directement proportionnelle à la valeur de la modulante utilisée. Plusieurs types de MLI intersectives sont utilisés. On peut citer par exemple, la MLI intersective sinus avec porteuse triangulaire, la MLI avec injection d'harmonique de rang 3, la MLI avec une porteuse aléatoire...

La MLI vectorielle (en anglais : Space Vector Modulation) diffère des autres méthodes de modulation par le fait que les signaux de commandes sont élaborés en tenant compte de l'état des trois bras de l'onduleur en même temps (contrôle du vecteur d'état de l'onduleur). L'objectif du contrôle vectoriel est de générer des tensions aussi proches que possible d'une tension référence que l'on fixe. Si ces références sont sinusoïdales triphasées équilibrées, alors le vecteur de référence tournera à une vitesse constante et son module suivra une trajectoire circulaire.

Dans le tableau suivant, on donne une comparaison des modulations MLI les plus utilisées :

Type de MLI	Amplitude maximale du fondamental de la tension en sortie de l'onduleur
MLI en pleine onde	$\frac{4}{\pi} * \frac{V}{2} \approx 0.637 VDC$
MLI vectorielle	$(\frac{2}{\pi^2} + 1) * \frac{V}{2} \approx 0.601 VDC$
MLI sinusoïdale avec harmonique 3	$(\frac{1}{6} + 1) * \frac{V}{2} \approx 0.583 VDC$
MLI sinusoïdale	$\frac{V}{2} \approx 0.5 VDC$

Tableau 3.3 : Comparaison des MLI en fonction de l'amplitude de la tension en sortie de l'onduleur.

La MLI en pleine onde donne l'amplitude de tension en sortie de l'onduleur maximale pour une tension du bus continu donnée. Cependant le contenu en harmoniques des courants est trop important pour les applications réseau. Ainsi le meilleur compromis est la MLI vectorielle, mais qui doit être réalisée obligatoirement sur un calculateur comme le DSP pour le calcul numérique des temps de conduction des interrupteurs électroniques.

Ainsi dans notre travail on opte pour la technique de modulation vectorielle (en anglais : Space vector modulation (SVM)), qui est une méthode fréquemment utilisée pour générer une PWM dans des applications de systèmes triphasés comme le moteur asynchrone. La suite de ce chapitre sera consacrée à la description de cette technique SVM.

En fait Cette technique de modulation utilise une configuration spéciale et particulière pour réaliser les commutations des interrupteurs électroniques de l'onduleur triphasé, afin de générer des courants pseudos sinusoïdaux dans les phases statoriques du moteur. Cette technique est basée sur la interprétation des projections du vecteur tension référence dans le repère $\alpha\beta$ par des durées (temps) d'ouverture ou de fermeture (on/off) pour chaque interrupteur constituant l'onduleur triphasé. [14]

4.1 Expressions des tensions phases dans le repère abc [14]

La méthode utilisée pour la génération d'un champ magnétique tournant consiste en utilisation de trois sources de tension indépendantes déphasées de 120° l'un à l'autre.

Dans le cas de l'onduleur triphasé, les sources de tension sinusoïdales sont remplacées par la tension continue (redressée) DC et les six (6) transistors de puissance agissent comme des interrupteurs on/off. Le but est de produire des courants pseudos sinusoïdaux dans les enroulements statoriques du moteur pour générer le champ magnétique tournant.

Les principes de base de la modulation vectorielle SVM peuvent être expliqués en se référant à l'onduleur de tension triphasé sur la figure 3.15. [14]

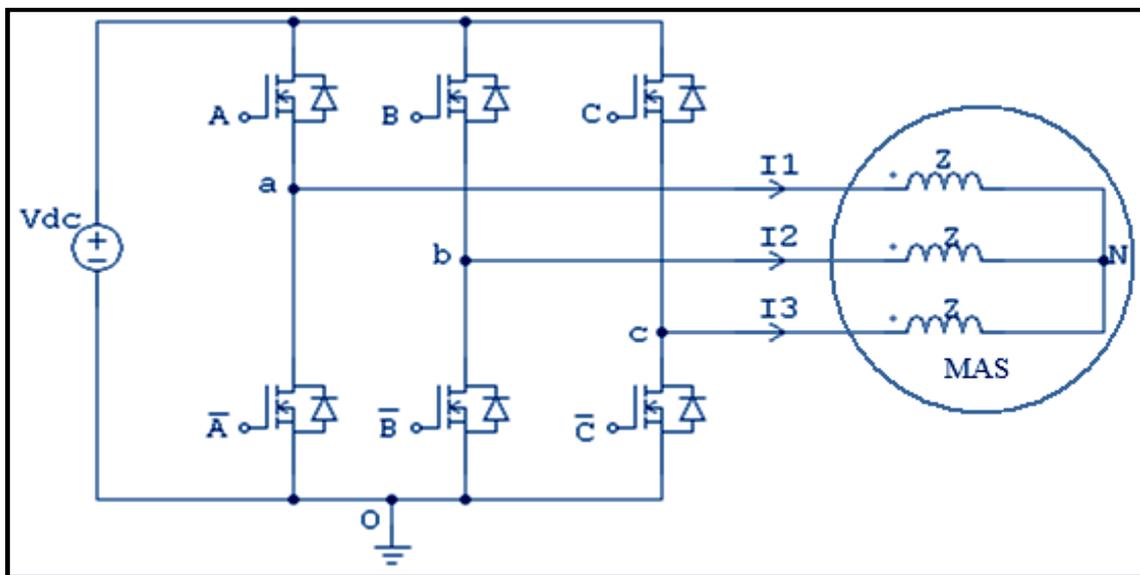


Figure 3.15 Alimentation du MAS à partir d'un onduleur triphasé.

Sur la figure 3.15, les transistors de puissance sont commandés par les signaux (A, B, C) et leurs valeurs complémentaires (\bar{A} , \bar{B} , \bar{C}).

Sachant que notre système est équilibré on peut écrire les équations suivantes :

$$\begin{cases} V_{on} = V_{oa} + Z * I_1 \\ V_{on} = V_{ob} + Z * I_2 \\ V_{on} = V_{oc} + Z * I_3 \end{cases} \quad \text{Avec: } (I_1 + I_2 + I_3) = 0$$

A partir des équations ci-dessus on aboutit :

$$V_{on} = \frac{1}{3} (V_{oa} + V_{ob} + V_{oc}) \quad (3.35)$$

Pour la phase (a), la tension phase au neutre peut être calculée comme suit :

$$\begin{aligned} V_{an} &= V_{on} - V_{oa} = \frac{1}{3} (V_{oa} + V_{ob} + V_{oc}) - V_{oa} \\ V_{an} &= \frac{1}{3} (2 V_{ao} - V_{bo} - V_{co}) \end{aligned}$$

Les mêmes calculs conduisent aux expressions suivantes :

$$\begin{cases} V_{an} = \frac{1}{3} (2 V_{ao} - V_{bo} - V_{co}) \\ V_{bn} = \frac{1}{3} (2 V_{bo} - V_{ao} - V_{co}) \\ V_{cn} = \frac{1}{3} (2 V_{co} - V_{ao} - V_{bo}) \end{cases} \quad (3.36)$$

Avec la configuration précédente de l'onduleur (figure 3.15), il y a huit (8) combinaisons des états (ON/OFF) des interrupteurs qui sont possibles, selon les états des signaux de commande (A, B, C), voir (Tableau 3.4):

A	B	C	V _{ao}	V _{bo}	V _{co}	V _{an}	V _{bn}	V _{cn}
0	0	0	0	0	0	0	0	0
0	0	1	0	0	V _{dc}	-1/3 V _{dc}	-1/3 V _{dc}	2/3 V _{dc}
0	1	0	0	V _{dc}	0	-1/3 V _{dc}	2/3 V _{dc}	-1/3 V _{dc}
0	1	1	0	V _{dc}	V _{dc}	-2/3 V _{dc}	1/3 V _{dc}	1/3 V _{dc}
1	0	0	V _{dc}	0	0	2/3 V _{dc}	-1/3 V _{dc}	-1/3 V _{dc}
1	0	1	V _{dc}	0	V _{dc}	1/3 V _{dc}	-2/3 V _{dc}	1/3 V _{dc}
1	1	0	V _{dc}	V _{dc}	0	1/3 V _{dc}	1/3 V _{dc}	-2/3 V _{dc}
1	1	1	V _{dc}	V _{dc}	V _{dc}	0	0	0

Tableau 3.4 Les tensions de sortie (V_{ao}, V_{bo}, V_{co}) et (V_{an}, V_{bn}, V_{cn}).

4.2 Expressions des tensions dans le repère $\alpha\beta$

La transformation $\alpha\beta$ donnée par l'expression ci-dessous, permet de transformer le vecteur tension triphasé dans le repère abc en un vecteur tension diphasé dans le repère $\alpha\beta$.

$$\begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{pmatrix} \begin{pmatrix} Van \\ Vbn \\ Vcn \end{pmatrix} \quad (3.37)$$

Puisque uniquement huit combinaisons sont possibles pour les états des interrupteurs, les tensions V_α et V_β ne peuvent prendre aussi que des valeurs finies dans le repère $\alpha\beta$ selon les états des signaux de commande (A, B, C), comme le montre le tableau suivant :

A	B	C	V_α	V_β	
0	0	0	0	0	\vec{V}_0
0	0	1	$-\sqrt{6}/6V_{dc}$	$-\sqrt{2}/2V_{dc}$	\vec{V}_1
0	1	0	$-\sqrt{6}/6V_{dc}$	$\sqrt{2}/2V_{dc}$	\vec{V}_2
0	1	1	$-\sqrt{2/3}V_{dc}$	0	\vec{V}_3
1	0	0	$\sqrt{2/3}V_{dc}$	0	\vec{V}_4
1	0	1	$\sqrt{6}/6V_{dc}$	$-\sqrt{2}/2V_{dc}$	\vec{V}_5
1	1	0	$\sqrt{6}/6V_{dc}$	$\sqrt{2}/2V_{dc}$	\vec{V}_6
1	1	1	0	0	\vec{V}_7

Tableau 3.5 Les tensions V_α et V_β dans le repère $\alpha\beta$.

Les huit vecteurs tension définis par la combinaison des états des interrupteurs, sont représentés sur la figure 3.16.

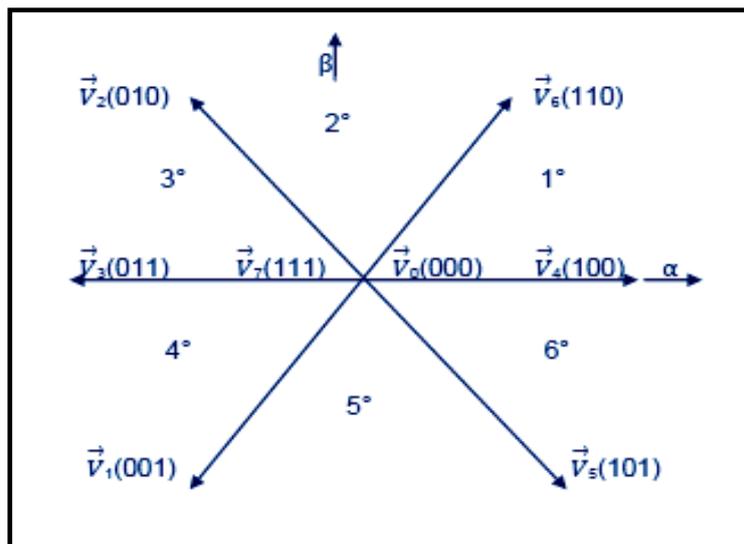


Figure 3.16 Diagramme des vecteurs tensions dans le repère $\alpha\beta$.

4.3 Principe de la modulation vectorielle

Le principe de la modulation SVM, consiste à obtenir le vecteur tension désiré, représenté dans le repère $\alpha\beta$, à partir de la superposition des vecteurs tensions possibles en sortie de l'onduleur, pour qu'en moyenne, à la fin de chaque période de modulation une tension égale à celle désirée soit générée. [9]

La procédure peut être expliquée en se référant à la figure 3.17. Le vecteur tension désiré, $\vec{V}_{\alpha\beta}$ est projeté sur les deux vecteurs adjacents qui délimitent un secteur de 60° (\vec{V}_{100} et \vec{V}_{110} , dans le cas de la figure 3.17). [9]

Les projections, \vec{V}_1 et \vec{V}_2 du vecteur $\vec{V}_{\alpha\beta}$, déterminent les fractions $\frac{T_i}{T}$ de la période de modulation T qui sera occupée par chaque vecteur (\vec{V}_{100} et \vec{V}_{110}), selon les relations :

$$\frac{T_1}{T} = \frac{|\vec{V}_1|}{|\vec{V}_{100}|} \quad \text{et} \quad \frac{T_2}{T} = \frac{|\vec{V}_2|}{|\vec{V}_{110}|} \quad (3.38)$$

Pour que la période de modulation soit complètement occupée par les vecteurs on applique le vecteur tension nul (\vec{V}_{000} ou \vec{V}_{111}) durant une fraction $\frac{T_3}{T}$ de la période de modulation, ce qui doit satisfaire la condition suivante :

$$T_1 + T_2 + T_3 = T \quad (3.39)$$

En suivant cette procédure, la tension de sortie moyenne \vec{V}_o sera donnée par :

$$\vec{V}_o = \frac{T_1}{T} \vec{V}_{100} + \frac{T_2}{T} \vec{V}_{110} + \frac{T_3}{T} \vec{V}_{111} = \vec{V}_1 + \vec{V}_2 = \vec{V}_{\alpha\beta} \quad (3.40)$$

Le locus des vecteurs tournants d'amplitudes constantes qui peuvent être générés avec l'onduleur sans distorsion, est représenté par le *cercle inscrit* dans l'hexagone (figure 3.17). En fait, tout vecteur se trouvant à l'intérieur du cercle génère un triplet (T_1, T_2, T_3) valide. Par contre, un vecteur qui se trouve partiellement à l'extérieur du cercle conduit à une somme des T_1, T_2, T_3 supérieure à la période. Cette situation est appelée saturation de l'onduleur et généralement cause une distorsion dans la tension en sortie de l'onduleur. [9]

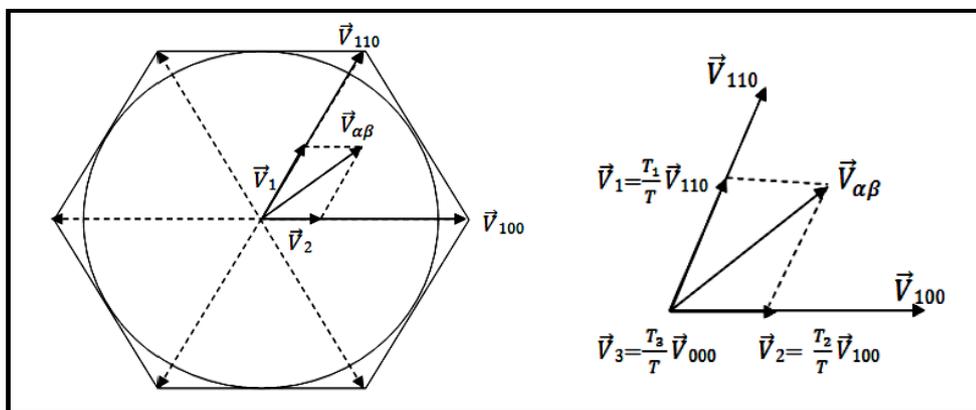


Figure 3.17 Diagramme du principe de la modulation SVM.

4.4 Détermination des temps de modulation T_i [14]

Les temps T_i sont déterminés par la projection dans la plan $\alpha\beta$ du vecteur référence \vec{V}_{ref} sur les deux vecteurs adjacents qui délimitent le secteur où se trouve \vec{V}_{ref} .

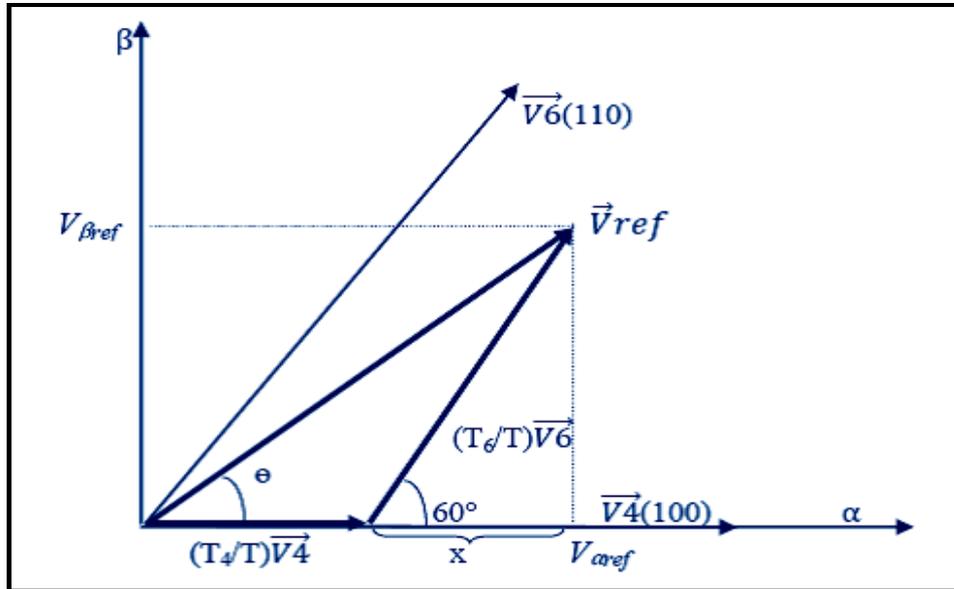


Figure 3.18 Diagramme de projection du vecteur de la tension référence.

Sur la figure 3.18, le vecteur tension de référence \vec{V}_{ref} se trouve dans le premier secteur et l'application temporelle de chaque vecteur adjacent est donnée par :

$$\begin{cases} T = T4 + T6 + T0 \\ \vec{V}_{ref} = \frac{T4}{T} \vec{V4} + \frac{T6}{T} \vec{V6} + \frac{T0}{T} \vec{V0} \end{cases} \quad (3.41)$$

La détermination des durées (temps) $T4$ et $T6$, est possible par de simples projections:

$$\begin{cases} V\beta_{ref} = \frac{T6}{T} \|\vec{V6}\| \cos(30^\circ) \\ V\alpha_{ref} = \frac{T4}{T} \|\vec{V4}\| + X \\ X = \frac{V\beta_{ref}}{\text{tg}(60^\circ)} \end{cases} \quad (3.42)$$

Or à partir du tableau 3.4, on a : $\|\vec{V4}\| = \sqrt{2/3}$ et $\|\vec{V6}\| = \sqrt{2/3}$

Ainsi, les quantités des temps allouées à chaque vecteur adjacent seront données par:

$$\begin{cases} T4 = \left(\frac{\sqrt{6}}{6}\right) \left(\frac{T}{V_{dc}}\right) (3V\alpha_{ref} - \sqrt{3} V\beta_{ref}) \\ T6 = \sqrt{2} \left(\frac{T}{V_{dc}}\right) (V\beta_{ref}) \end{cases} \quad (3.43)$$

Ou bien sous la forme matricielle :

$$\text{Pour le secteur 1 : } \begin{pmatrix} T4 \\ T6 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} \sqrt{3}/2 & -\sqrt{2}/2 \\ 0 & \sqrt{2} \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

Des calculs similaires donneront :

$$\text{Pour le secteur 2 : } \begin{pmatrix} T6 \\ T2 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} -\sqrt{3}/2 & \sqrt{2}/2 \\ \sqrt{3}/2 & \sqrt{2}/2 \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

$$\text{Pour le secteur 3 : } \begin{pmatrix} T2 \\ T3 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} 0 & \sqrt{2} \\ \sqrt{3}/2 & -\sqrt{2}/2 \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

$$\text{Pour le secteur 4 : } \begin{pmatrix} T3 \\ T1 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} 0 & -\sqrt{2} \\ -\sqrt{3}/2 & \sqrt{2}/2 \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

$$\text{Pour le secteur 5 : } \begin{pmatrix} T1 \\ T5 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} -\sqrt{3}/2 & -\sqrt{2}/2 \\ \sqrt{3}/2 & -\sqrt{2}/2 \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

$$\text{Pour le secteur 6 : } \begin{pmatrix} T5 \\ T4 \end{pmatrix} = \begin{pmatrix} T \\ V_{dc} \end{pmatrix} \begin{pmatrix} \sqrt{3}/2 & \sqrt{2}/2 \\ 0 & -\sqrt{2} \end{pmatrix} \begin{pmatrix} V_{\alpha ref} \\ V_{\beta ref} \end{pmatrix}$$

A des fins de programmation et d'implantation sur ordinateur, les temps de modulation $t1$ et $t2$ pour chaque secteur peuvent être associés aux variables X , Y et Z définies comme suit :

$$\begin{cases} X = \sqrt{2} \left(\frac{T}{V_{dc}} \right) (V_{\beta ref}) \\ Y = \left(\frac{T}{V_{dc}} \right) \left(\frac{\sqrt{2}}{2} V_{\beta ref} + \sqrt{\frac{3}{2}} V_{\alpha ref} \right) \\ Z = \left(\frac{T}{V_{dc}} \right) \left(\frac{\sqrt{2}}{2} V_{\beta ref} - \sqrt{\frac{3}{2}} V_{\alpha ref} \right) \end{cases} \quad (3.44)$$

Dans l'exemple précédent pour le secteur 1, $t1=T4 = -Z$ et $t2=T6 = X$. Pour les autres secteurs on résume les valeurs de $t1$ et $t2$ dans le tableau suivant :

Secteur	t1	t2
1	-Z	X
2	Z	Y
3	X	-Y
4	-X	Z
5	-Y	-Z
6	Y	-X

Tableau 3.6 Les durées $t1$ et $t2$ pour tous les secteurs.

4.5 L'implémentation de la SVM

La technique de modulation SVM, nécessite beaucoup de calculs et l'implémentation de son algorithme sur les processeurs devient possible et simple grâce aux développements enregistrés dans le domaine du contrôle numérique avec l'apparition des microcontrôleurs et DSP. Les étapes d'exécution de la SVM sont données ci-dessous : [14]

- 1) La première étape est de déterminer les composantes $V_{\alpha ref}$ et $V_{\beta ref}$, puis le secteur.
- 2) La deuxième étape est de calculer les durées t_1 et t_2 , suivant le tableau 3.6.
- 3) La troisième étape est de calculer les instants des commutations, comme suit :

$$\begin{cases} t_{aon} = (T - t_1 - t_2) / 2 \\ t_{bon} = t_{aon} + t_1 \\ t_{con} = t_{bon} + t_2 \end{cases} \quad (3.45)$$

- 4) La dernière étape est d'assigner, suivant la valeur du secteur S, le rapport cyclique (t_{xon}) adéquat à la phase du moteur appropriée (autrement dit au registre CMPRx appropriée dans le cas du DSP). Le tableau ci-après représente cette détermination :

Secteur	1	2	3	4	5	6
CMPR1	t_{aon}	t_{bon}	t_{con}	t_{con}	t_{bon}	t_{aon}
CMPR2	t_{bon}	t_{aon}	t_{aon}	t_{bon}	t_{con}	t_{con}
CMPR3	t_{con}	t_{con}	t_{bon}	t_{aon}	t_{aon}	t_{bon}

Tableau 3.7 Assignement de l'instant de commutation au registre CMPRx adéquat.

La figure 3.19 illustre les instants des commutations et les séquences PWM, dans le cas d'un vecteur tension \vec{V}_{ref} se trouvant dans le premier secteur ($S=1$).

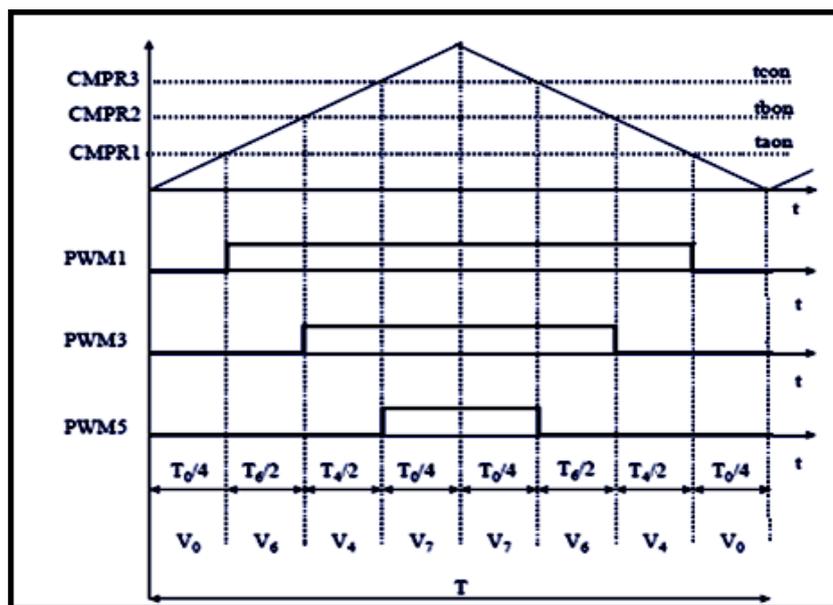


Figure 3.19 Les séquences PWM dans le cas du secteur 1.

Chapitre IV

Simulation d'une commande pour moteur asynchrone

Dans ce chapitre sera présentée une simulation sur le logiciel MATLAB/ SIMULINK, de la commande en vitesse d'un moteur asynchrone alimenté par un onduleur triphasé, en utilisant la technique de modulation vectorielle (SVM) pour la détermination des signaux de commande (MLI) appropriés pour chaque interrupteur de l'onduleur. Cette simulation nous permettra de mettre en évidence les performances de la technique SVM pour la commande en vitesse d'un moteur asynchrone triphasé.

1. Présentation brève de MATLAB

MATLAB dont le nom provient de MATrix LABoratory est un langage de calcul scientifique basé sur le type de variable matricielle. Outre ses grandes capacités de calcul numérique appliqué aux différents domaines scientifiques tels que le traitement de signal l'automatique...etc. MATLAB intègre des fonctionnalités graphiques de grande qualité en 2D ou 3D. [27]

MATLAB fut conçu initialement (au milieu des années 1980) pour manipuler aisément des matrices à l'aide de fonctions préprogrammées (additions, multiplications, inversions, décompositions, déterminants...), en s'affranchissant des contraintes des langages de programmation classique (déclarations de variables, phase d'édition-compilation-exécution). Cette orientation de calcul matriciel a depuis évolué vers un outil pouvant être vu comme une super-calculatrice graphique et regroupant dans la version de base la quasi-majorité des problèmes numériques. [26]

L'outil additionnel Simulink dans le logiciel MATLAB permet la programmation graphique et possède des bibliothèques de blocs permettant la simulation de systèmes dynamiques linéaires ou non linéaires. Ces systèmes peuvent être analogiques, discrets ou hybrides. Les paramètres régissant le fonctionnement de ces systèmes peuvent être modifiés en cours de simulation, et on peut observer leurs effets immédiatement. Simulink possède une interface graphique pour visualiser les résultats sous forme de graphiques ou de valeurs numériques en cours de simulation. Simulink est bâti autour d'une bibliothèque de fonctions classées par catégories (systèmes discrets/continus, blocs linéaires ou non linéaires...etc.) [27]

MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques. Il fait partie des langages interprétés (non compilés) c'est-à-dire que les instructions sont décodées et exécutées ligne par ligne. MATLAB intègre deux modes de fonctionnement: [25]

- ❖ **Mode interactif** : MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur dans la fenêtre de commande.
- ❖ **Mode exécutif** : MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).

Le logiciel MATLAB comporte les fenêtres suivantes [25] :

- ❖ **Fenêtre Commande** : Dans cette fenêtre, l'utilisateur donne les instructions et MATLAB retourne les résultats.
- ❖ **Fenêtres Graphique** : MATLAB trace les graphiques dans ces fenêtres.
- ❖ **Fichiers M** : Ce sont des programmes en langage MATLAB (écrits par l'utilisateur).
- ❖ **Toolboxes** : Ce sont des collections de fichiers M développés pour des domaines d'application spécifiques (Signal Processing Toolbox, Control System Toolbox, Robust Control Toolbox, Neural Network Toolbox, Fuzzy Logic Toolbox, ... etc.)
- ❖ **Simulink** : C'est l'extension graphique de MATLAB permettant de travailler avec des diagrammes en blocs. Il permet de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler leurs fonctionnements.
- ❖ **Blocksets** : Ce sont des collections de blocs Simulink développés pour des domaines d'application spécifiques (DSP Blockset, Power System Blockset, etc.).

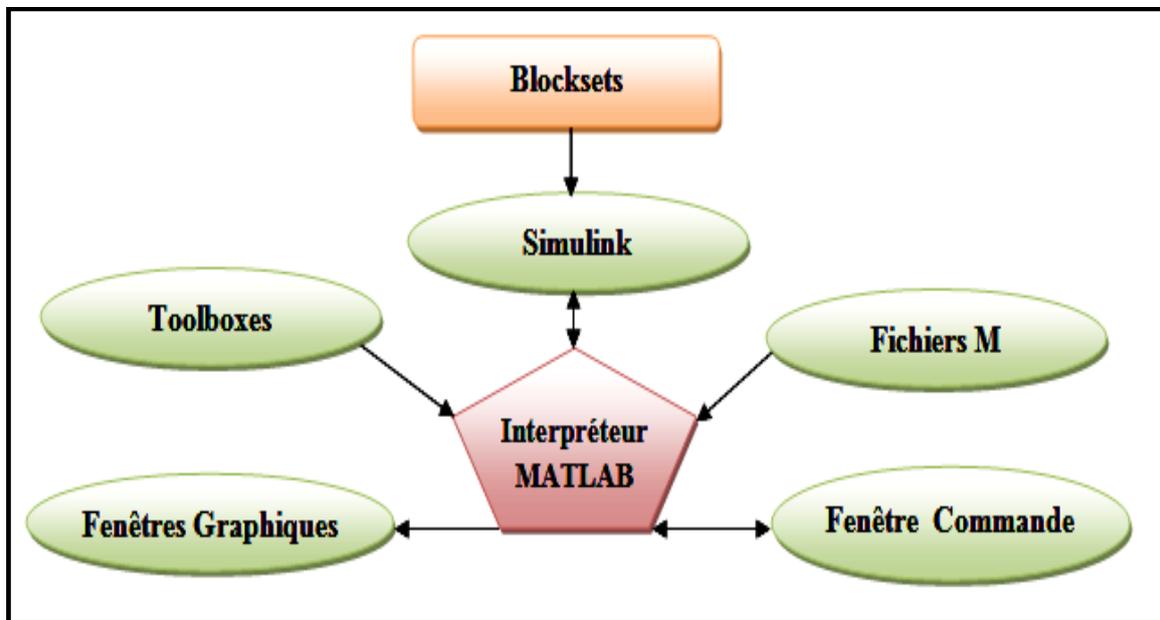


Figure 4.1 L'environnement MATLAB

2. Réalisation de la commande à base de la modulation vectorielle

Le développement de la modulation de largeur d'impulsion (MLI) a apporté une plus grande souplesse dans le contrôle des systèmes triphasés comme le moteur asynchrone triphasé. Parmi les diverses techniques de modulation MLI, on trouve la technique de modulation vectorielle SVM (Space Vector Modulation) qui est la plus utilisée ces dernières années dans la commande des systèmes triphasés. Le principe de cette méthode consiste en détermination des intervalles de temps (durée de modulation) qui doivent être allouées à chaque composante du vecteur de tension référence durant chaque période de modulation.

❖ Les étapes de réalisation des blocs de la commande

Le modèle global de la commande en vitesse d'un moteur asynchrone sous MATLAB/Simulink est constitué de plusieurs blocs, et chacun exécute une fonction spécifique dans le système de commande. Les fonctions réalisées par les blocs du système de commande sont:

- 1) Détermination de la phase *thêta* du vecteur tension référence \vec{V}_{ref} .
- 2) Détermination des composantes V_{α}, V_{β} de la tension de référence.
- 3) Détermination du secteur S du vecteur \vec{V}_{ref} .
- 4) Calcul des variables intermédiaires X, Y et Z .
- 5) Calcul des durées t_1 et t_2 et des temps de commutation t_{aon}, t_{bon} et t_{con} .
- 6) Détermination des séquences de commutations.
- 7) Génération des séries d'impulsions PWM1...PWM6.

3. Présentation des blocs de la commande sous Simulink

Toutes les étapes impliquées dans le fonctionnement du système de commande en vitesse d'un moteur asynchrone en utilisant la technique de modulation vectorielle SVM, seront représentées par des modèles de diagrammes en blocs sous MATLAB/Simulink, afin de le simuler et de tester l'intérêt et l'efficacité de cette technique SVM.

3.1 Le modèle global du système de commande

Ce modèle nous permet de simuler la commande en vitesse d'un moteur asynchrone en boucle ouverte alimenté par un onduleur MLI, et en utilisant la technique SVM.

Dans ce modèle les données d'entrée sont la fréquence des courants statoriques qui permettent d'agir sur la vitesse de rotation du moteur, le sens de rotation et le couple mécanique (la charge). Les sorties peuvent être les grandeurs électriques et mécaniques (les tensions des phases ou composées, les courants des phases, le couple électromagnétique, la vitesse du rotor...etc.) et leurs allures en fonction du temps sont facilement visualisées à l'aide de l'oscilloscope (Scope) de Simulink. Le modèle global de la commande du moteur asynchrone sous Simulink est donné sur la figure 4.2 ci-après:

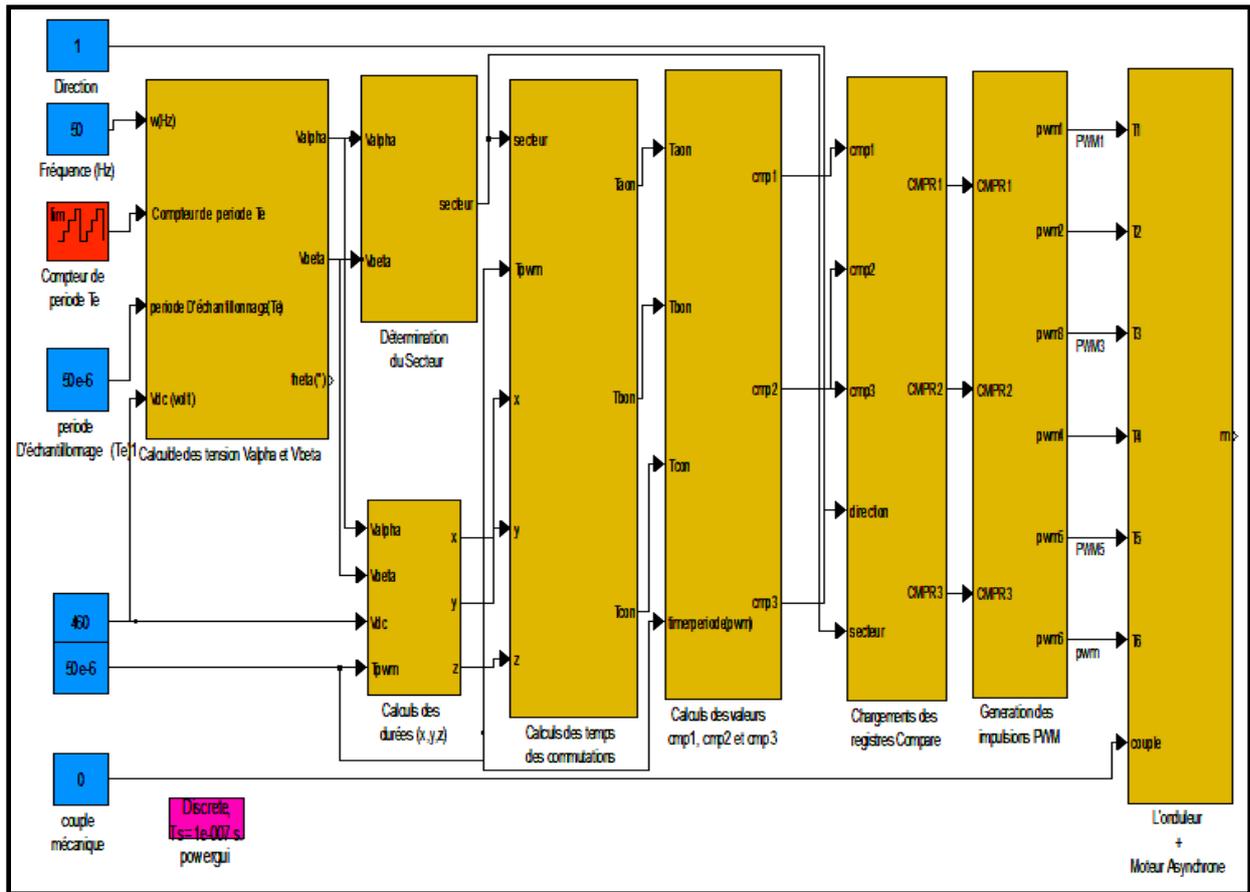


Figure 4.2 Le modèle Simulink de la commande en vitesse du moteur asynchrone.

3.2 Détermination de la phase θ

La phase θ du vecteur \vec{V}_{ref} est évaluée à l'aide du modèle sur la figure ci-dessous, à chaque période de la base de temps (période d'échantillonnage T_e) de 50 μs . La phase instantanée θ est directement liée à la fréquence de référence (f_{ref}) par la relation :

$$\theta(i + 1) = \omega_{ref} * T_e + \theta(i) = 2\pi * f_{ref} * T_e + \theta(i) \quad (4.1)$$

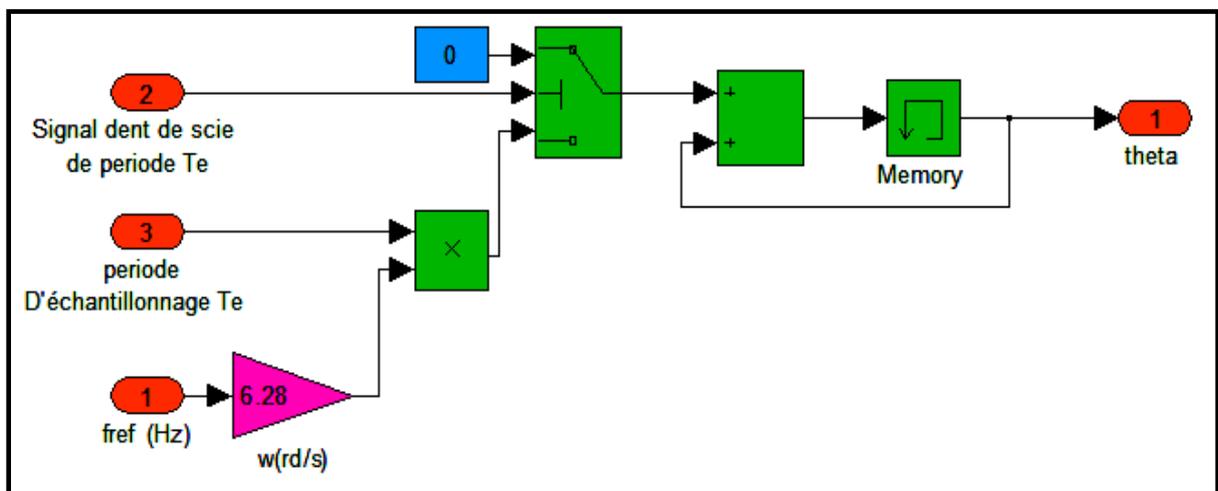


Figure 4.3 Le modèle Simulink qui détermine la phase θ .

3.3 Détermination des composantes $V\alpha, V\beta$

Après avoir obtenu la phase instantanée θ du vecteur tension référence \vec{V}_{ref} , on utilise sa valeur pour déterminer les composantes $V\alpha$ et $V\beta$ de ce vecteur.

Le modèle sur la figure 4.4 permet de déterminer les composantes $V\alpha, V\beta$ du vecteur \vec{V}_{ref} à partir de l'amplitude U_{ref} du vecteur \vec{V}_{ref} et sa phase θ suivant les relations :

$$\begin{cases} V\alpha = U_{ref} * \cos(\theta) \\ V\beta = U_{ref} * \sin(\theta) \end{cases} \quad (4.2)$$

L'amplitude U_{ref} peut être évaluée :

- Soit comme constante, tel que : $0 < U_{ref} \leq V_{dc}$.
Pour éviter des distorsions au niveau des tensions en sortie de l'onduleur on prend : [9]

$$0 < U_{ref} \leq \frac{\sqrt{2}}{2} V_{dc} \quad (V_{dc} : \text{La tension du bus continu de l'onduleur.})$$

- Soit en utilisant la commande : $\frac{V}{f} = \text{Const}$, et ceci en ajoutant un bloc qui réalise la fonction : $U_{ref} = \text{Const} * f_{ref}$

Afin d'éviter la distorsion des tensions en sortie de l'onduleur on doit vérifier la condition :

$$0 < \text{Const} \leq \frac{\sqrt{2} V_{dc}}{2 f_{max}} \quad (\text{Fréquence nominale du moteur : } f_{max} = 50 \text{ Hz})$$

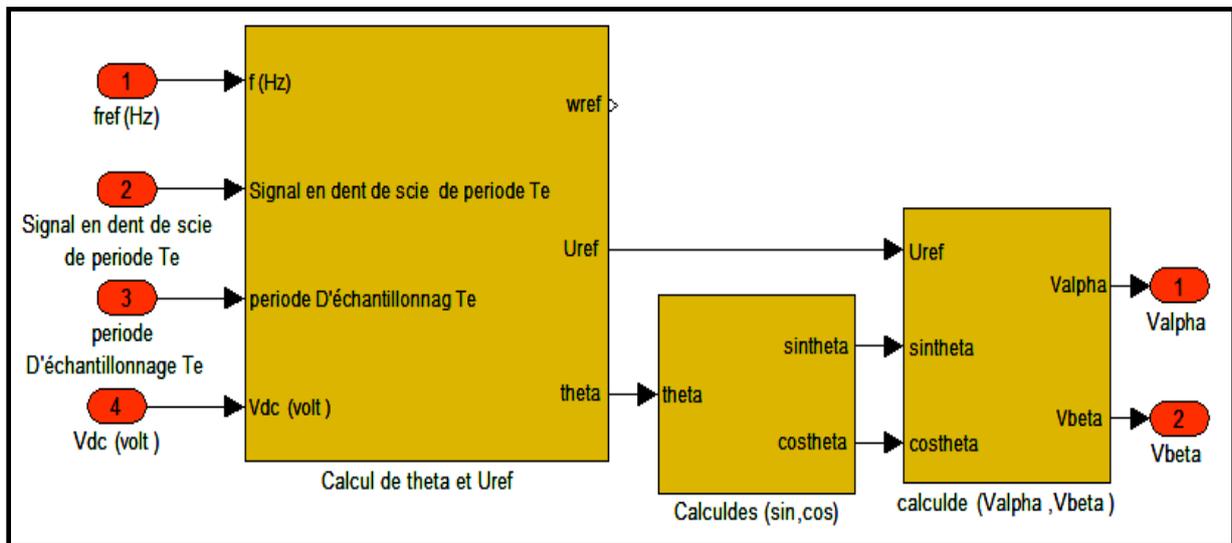


Figure 4.4 Le bloc Simulink qui calcule les composantes $V\alpha$ et $V\beta$.

3.4 Détermination du secteur S

Le secteur S du vecteur référence \vec{V}_{ref} peut être déterminé directement par la connaissance de la phase $theta$ du vecteur \vec{V}_{ref} en comparant et en situant la phase $theta$ du vecteur \vec{V}_{ref} par rapport aux angles principaux délimitant chaque secteur de (60°), ou bien par la méthode basée sur la connaissance des amplitudes et signes des composantes V_α, V_β du vecteur \vec{V}_{ref} .

Cette deuxième méthode est synthétisée dans le tableau 4.1 qui a une forme d'une table logique dont les variables sont : $A0, A1$ et $A2$. [29]

- $A2$: est le signe de V_α , ($A2$ égale à 0 si V_α est négatif sinon $A2$ égale à 1).
- $A1$: est le signe de V_β , ($A1$ égale à 0 si V_β est négatif sinon $A1$ égale à 1).
- $A0$: est déterminé à partir de la valeur absolue de la tangente de l'angle $theta$, ($A0$ égale à 1 si la valeur absolue de (V_β / V_α) est supérieure ou égale à $\tan 60^\circ$ (1.732) sinon $A0$ égale à 0).

A2	A1	A0	Secteur
0	0	0	4
0	0	1	5
0	1	0	3
0	1	1	2
1	0	0	6
1	0	1	5
1	1	0	1
1	1	1	2

Tableau 4.1 Détermination du secteur S.

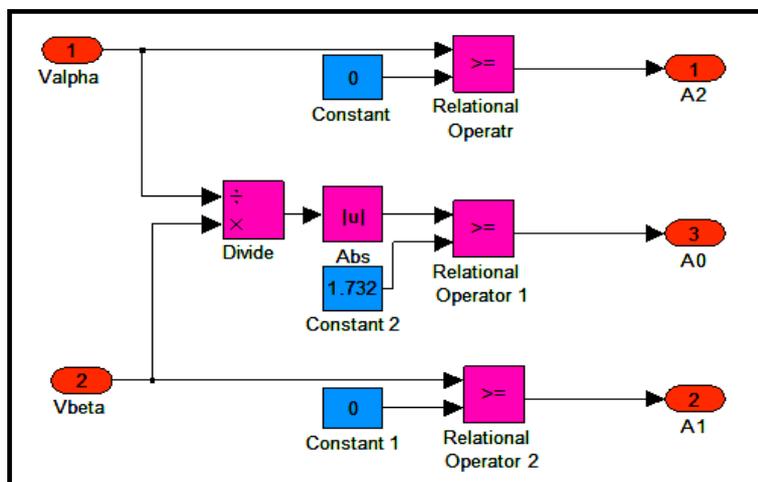


Figure 4.5 Le bloc Simulink qui calcule les variables $A0, A1$ et $A2$.

3.5 Calcul des variables X, Y et Z

En fait le calcul des variables X, Y et Z n'est qu'une détermination antérieure des grandeurs temporelles $t1$ et $t2$. Les variables X, Y et Z sont calculées suivant les équations :

$$\begin{cases} X = \sqrt{2} \left(\frac{T}{V_{dc}} \right) (V\beta_{ref}) \\ Y = \left(\frac{T}{V_{dc}} \right) \left(\frac{\sqrt{2}}{2} V\beta_{ref} + \sqrt{\frac{3}{2}} V\alpha_{ref} \right) \\ Z = \left(\frac{T}{V_{dc}} \right) \left(\frac{\sqrt{2}}{2} V\beta_{ref} - \sqrt{\frac{3}{2}} V\alpha_{ref} \right) \end{cases} \quad (4.3)$$

Le modèle Simulink effectuant ces calculs est donné sur la figure suivante :

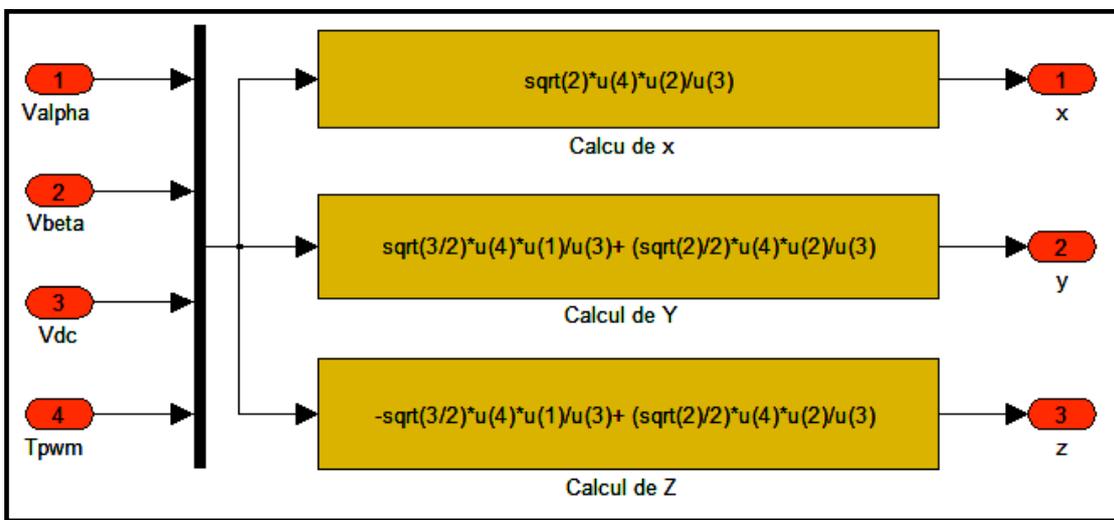


Figure 4.6 Le modèle Simulink qui détermine les variables X, Y et Z.

3.6 Calcul de $t1$, $t2$, $taon$, $tbon$ et $tcon$

a) Calcul de $t1$ et $t2$

Dans cette phase de calcul on cherche à déterminer les temps (durées) $t1$ et $t2$ suivant la valeur du secteur S occupé par le vecteur \vec{V}_{ref} et les valeurs des variables X, Y et Z. Le calcul des temps $t1$ et $t2$ est résumé dans le tableau suivant :

Secteur	1	2	3	4	5	6
$t1$	$-Z$	Z	X	$-X$	$-Y$	Y
$t2$	X	Y	$-Y$	Z	$-Z$	$-X$

Tableau 4.2 Détermination des durées $t1$ et $t2$.

b) Calcul de $taon$, $tbon$ et $tcon$

Les grandeurs $taon$, $tbon$ et $tcon$ qui déterminent les instants de commutation sont calculées suivant les équations ci-dessous :

$$\begin{cases} taon = (T - t1 - t2) / 2 \\ tbon = taon + t1 \\ tcon = tbon + t2 \end{cases} \quad (4.4)$$

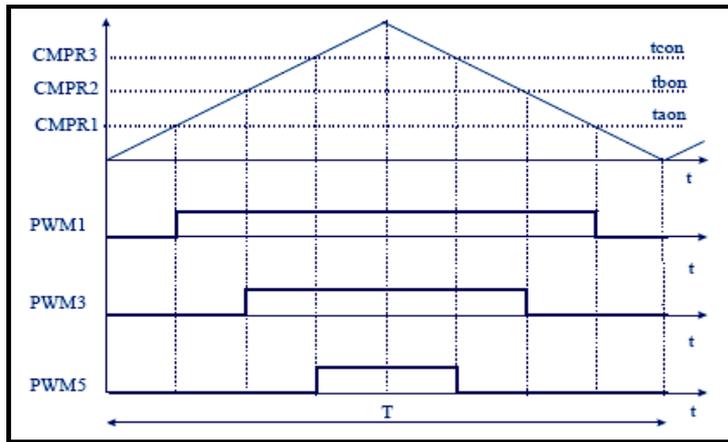


Figure 4.7 Illustration de la signification des temps $taon$, $tbon$ et $tcon$.

Le calcul des durées $t1$ et $t2$, puis des instants de commutation $taon$, $tbon$ et $tcon$ est effectué sous Simulink par le modèle sur la figure suivante :

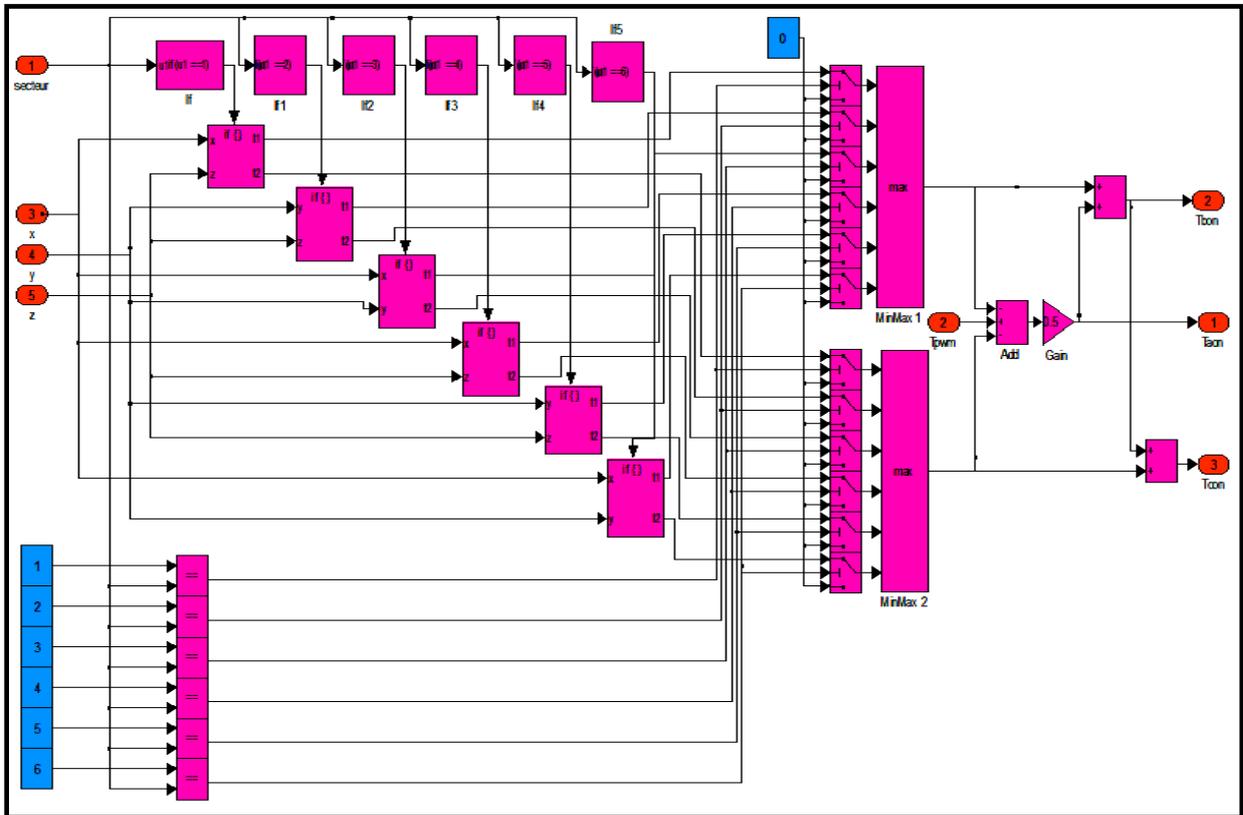


Figure 4.8 Le modèle Simulink qui détermine $t1$ et $t2$ puis $taon$, $tbon$ et $tcon$.

3.7 Calcul des valeurs *cmp1*, *cmp2* et *cmp3*

Les valeurs à calculer *cmp1*, *cmp2*, *cmp3* représentent les quantités à charger dans les registres compare CMPR1, CMPR2 et CMPR3.

La détermination des quantités *cmp1*, *cmp2* et *cmp3* est réalisée par le modèle sur la figure suivante :

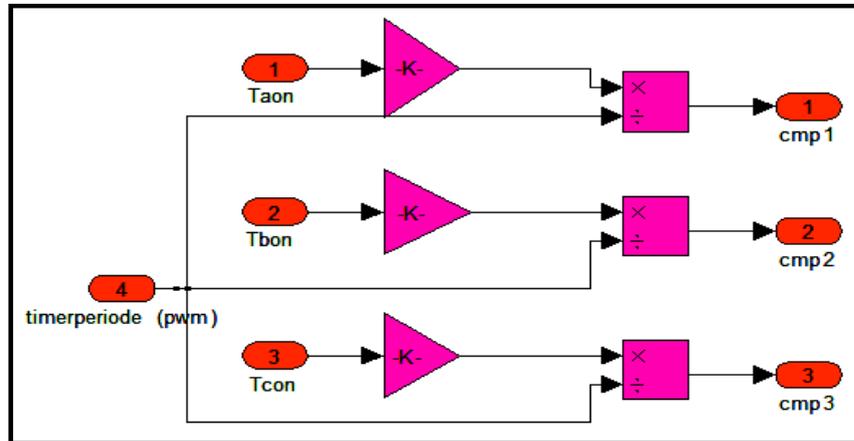


Figure 4.9 Le modèle qui calcule les valeurs *cmp1*, *cmp2* et *cmp3*.

3.8 Détermination des séquences de commutation

Dans cette étape, on procède à l'affectation de chaque valeur *cmp1*, *cmp2* et *cmp3* aux registres CMPRx appropriés suivant la valeur du secteur *S* occupé par le vecteur \vec{V}_{ref} .

Les deux tableaux ci-dessous résument les séquences de commutation pour les deux sens de rotation direct et inverse.

Secteur	1	2	3	4	5	6
CMPR1	<i>cmp1</i>	<i>cmp2</i>	<i>cmp3</i>	<i>cmp3</i>	<i>cmp2</i>	<i>cmp1</i>
CMPR2	<i>cmp2</i>	<i>cmp1</i>	<i>cmp1</i>	<i>cmp2</i>	<i>cmp3</i>	<i>cmp3</i>
CMPR3	<i>cmp3</i>	<i>cmp3</i>	<i>cmp2</i>	<i>cmp1</i>	<i>cmp1</i>	<i>cmp2</i>

Tableau 4.3 Séquence de commutation dans le sens de rotation direct.

Secteur	1	2	3	4	5	6
CMPR1	<i>cmp2</i>	<i>cmp1</i>	<i>cmp1</i>	<i>cmp2</i>	<i>cmp3</i>	<i>cmp3</i>
CMPR2	<i>cmp1</i>	<i>cmp2</i>	<i>cmp3</i>	<i>cmp3</i>	<i>cmp2</i>	<i>cmp1</i>
CMPR3	<i>cmp3</i>	<i>cmp3</i>	<i>cmp2</i>	<i>cmp1</i>	<i>cmp1</i>	<i>cmp2</i>

Tableau 4.4 Séquence de commutation dans le sens de rotation inverse.

Le modèle Simulink du bloc qui réalise les séquences de commutation est donné par la figure ci-après :

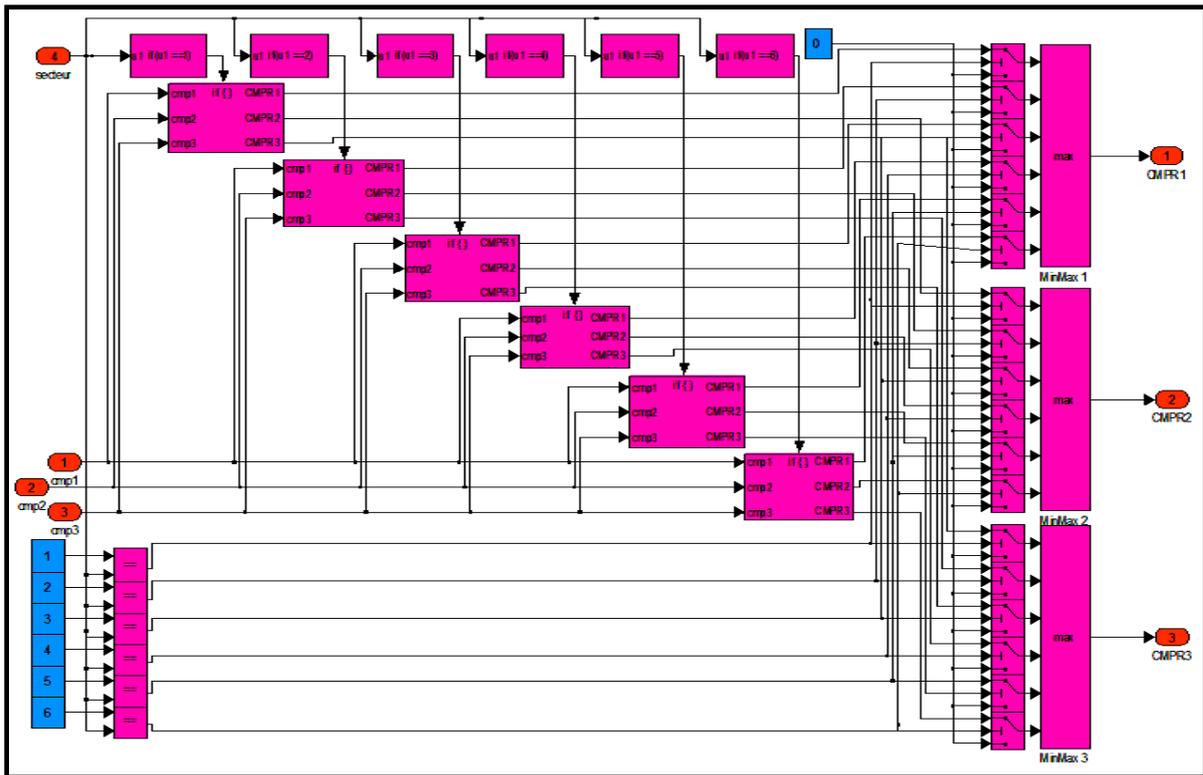


Figure 4.10 Le modèle Simulink qui détermine les séquences de commutation.

3.9 Génération des impulsions PWM

Ce bloc génère les séries d'impulsions PWM1...PWM6 qui serviront comme signaux de commande pour les MOSFET de l'onduleur MLI qui alimente le moteur asynchrone.

Le bloc de génération des impulsions PWM sous Simulink est représenté ci-dessous :

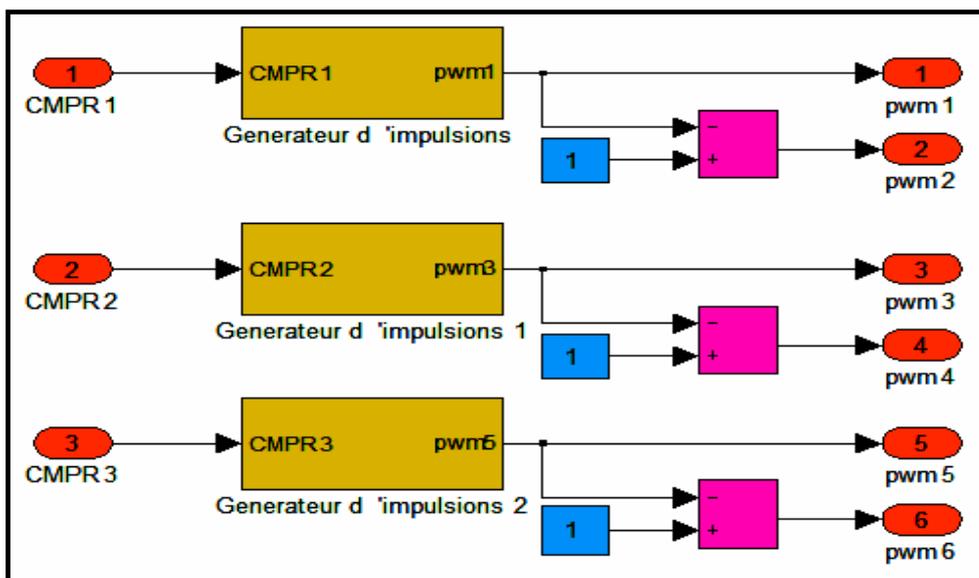


Figure 4.11 Le modèle Simulink qui génère les impulsions PWM.

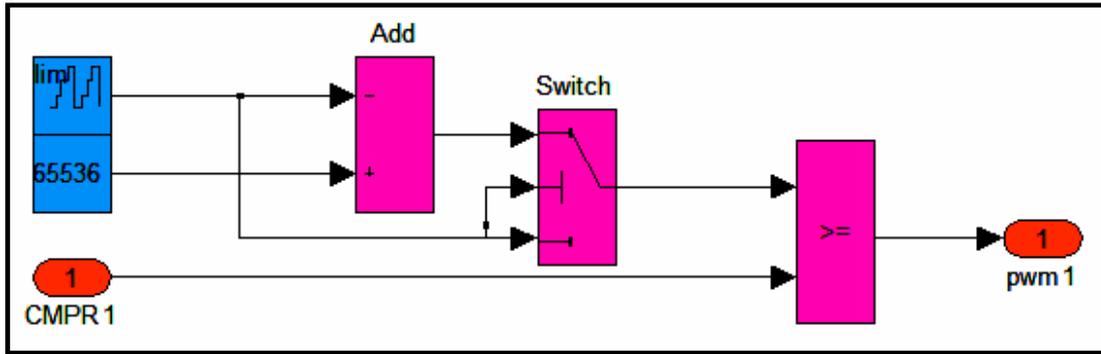


Figure 4.12 La structure interne du bloc générateur d'impulsions.

3.10 Le circuit de l'onduleur triphasé avec le moteur asynchrone

La figure suivante illustre le modèle Simulink de la partie puissance constituée de l'onduleur triphasé en MOSFET et du moteur asynchrone.

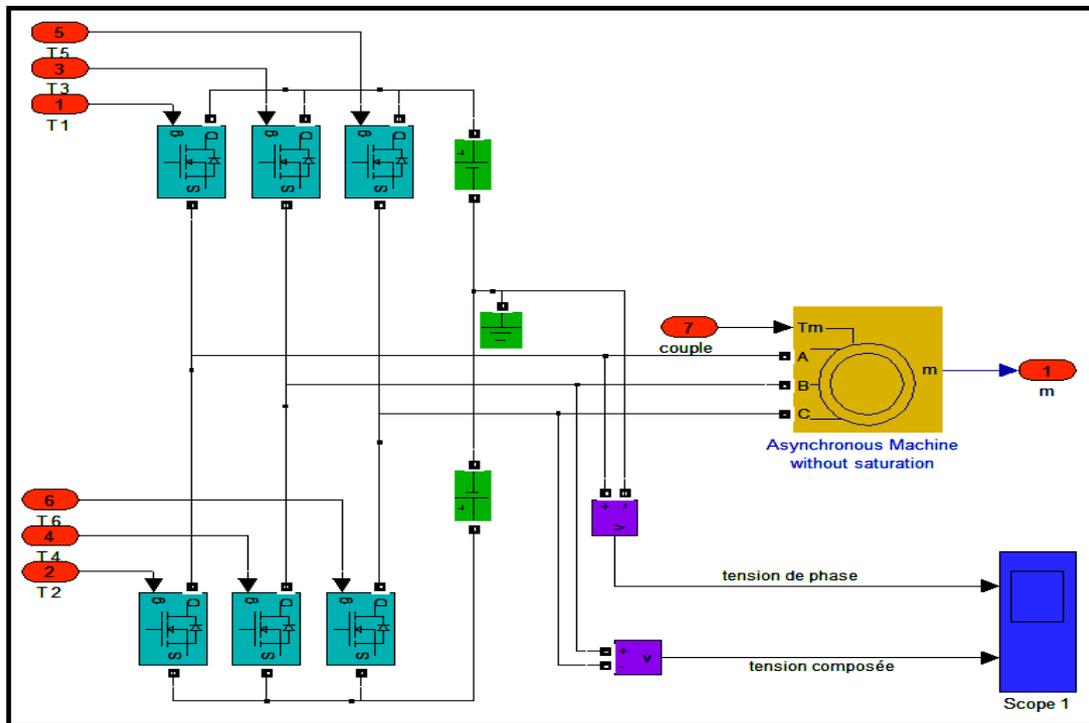


Figure 4.13 Le modèle Simulink de l'onduleur et le moteur.

4. Résultats de la simulation

Dans la phase de simulation, il est simulé le comportement du moteur asynchrone, lors d'une variation de la référence de fréquence et lors de changement du sens de rotation, et il est simulé le comportement du moteur soumis à un couple mécanique résistant (charge).

Remarque : Dans les figures qui suivent, les valeurs sur les axes des abscisses (axes des temps) sont données en seconde (s).

4.1 Signaux de commande

La figure ci-après (Figure 4.14) illustre la forme des impulsions modulées en largeur des signaux de commande PWM1 et PWM2 qui prennent des états logiques complémentaires.

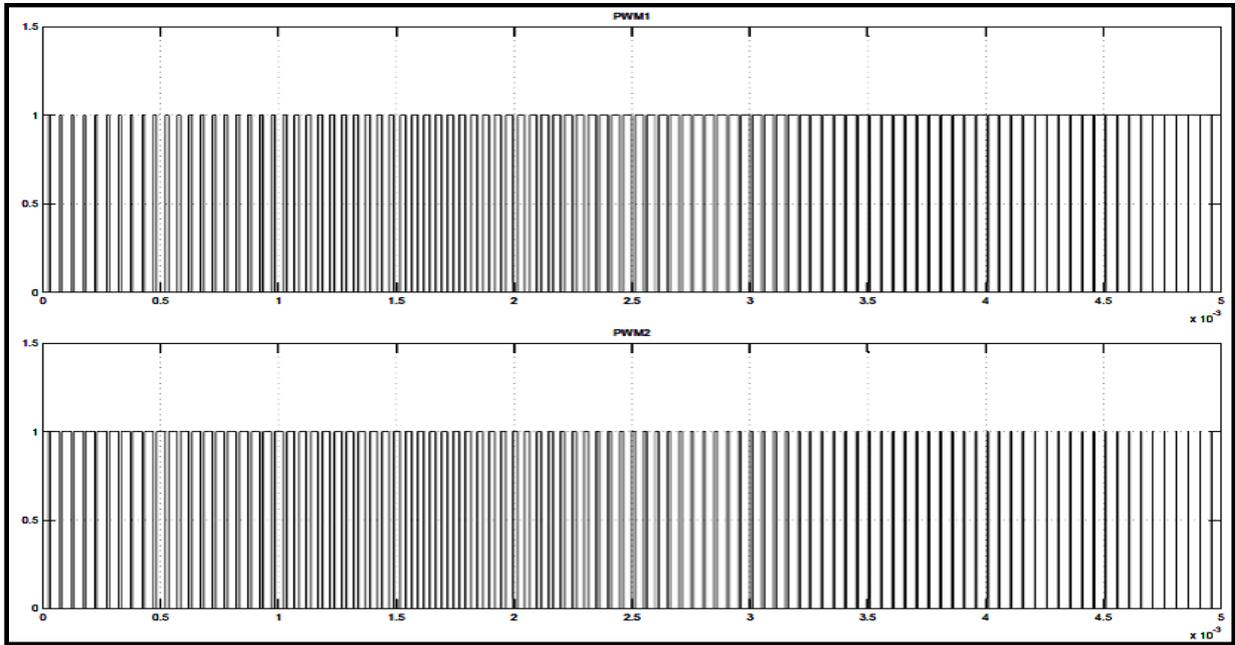


Figure 4.14 Allures des signaux de commande PWM1 et PWM2.

4.2 Les signaux de modulation de référence

La figure suivante illustre l'évolution des valeurs dans les registres CMPR1, CMPR2 et CMPR3, qui représentent, en effet, l'évolution dans le temps des largeurs (rapport cyclique) des impulsions des signaux de commande MLI. Ces signaux peuvent être interprétés comme des références pour les signaux en sortie de l'onduleur.

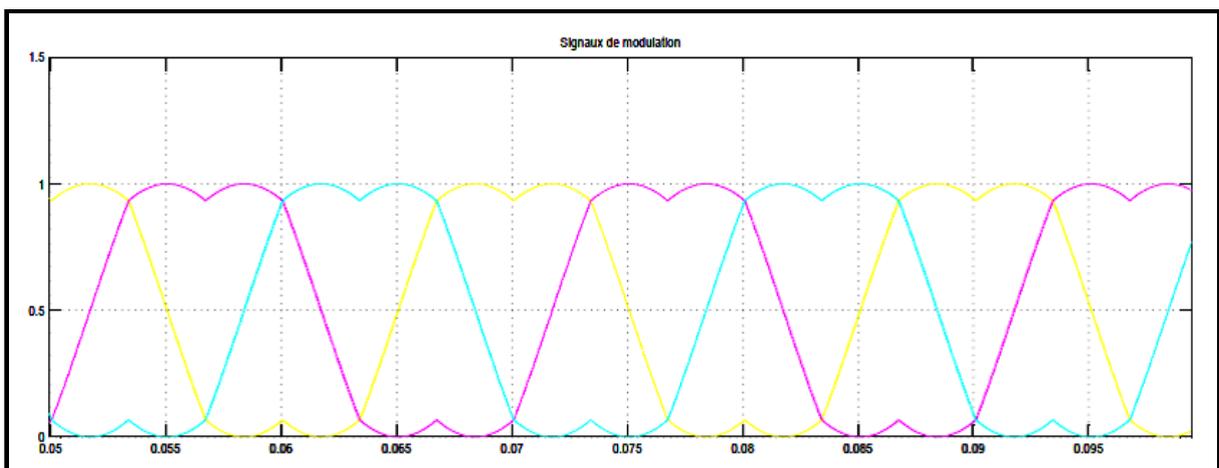


Figure 4.15 Les signaux de modulation pour $U_{ref} = \frac{\sqrt{2}}{2} V_{dc}$ et $f_{ref} = 50$ Hz.

4.3 La tension composée

La figure suivante illustre la forme de la tension composée prélevée entre deux phases du moteur, à la fréquence de référence 50 Hz.

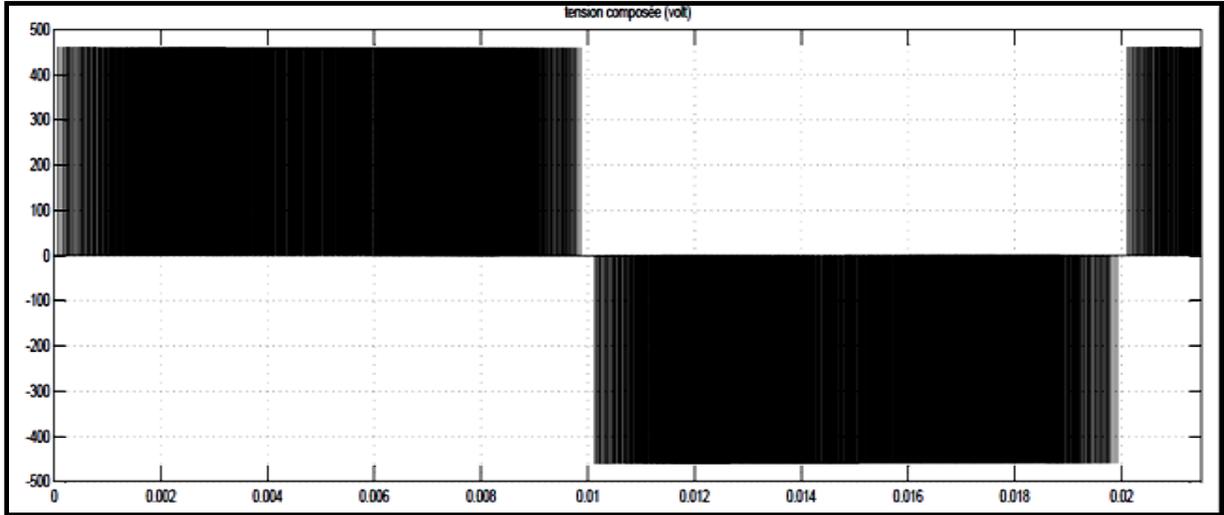


Figure 4.16 Allure de la tension composée à la fréquence 50 Hz.

4.4 Les courants des phases de stator

La simulation du système de commande pour la fréquence de référence 50 Hz, nous a permis d'obtenir l'allure illustrée sur la figure 4.17 pour le courant statourique I_{s_a} de la phase (a) du moteur.

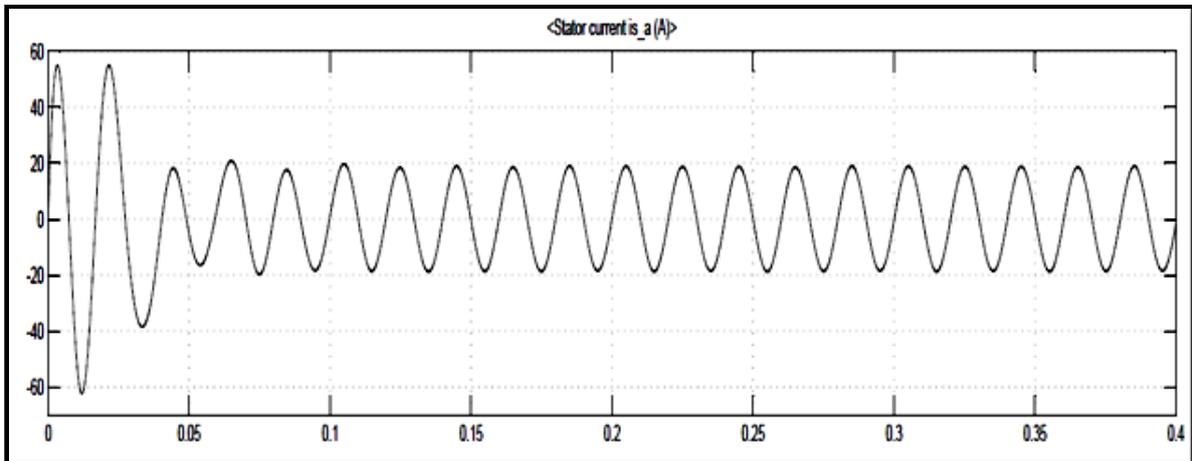


Figure 4.17 Allure du courant dans la phase du stator pour la fréquence 50 Hz.

On constate la forme sinusoïdale du courant dont l'amplitude au démarrage atteint 50 A (en régime transitoire) puis s'atténue et se stabilise à 20 A (en régime permanent). De plus on vérifie que la fréquence de ce courant de phase est égale exactement à la fréquence de référence imposée (50 Hz), ce qui reflète la précision du modèle Simulink réalisé et l'importance de la technique SVM pour la commande d'un moteur asynchrone.

La visualisation sur l'oscilloscope des courants statoriques I_{s_a} , I_{s_b} et I_{s_c} dans les trois phases du stator, à la fréquence de référence 50 Hz nous a offert les allures sur la figure 4.18.

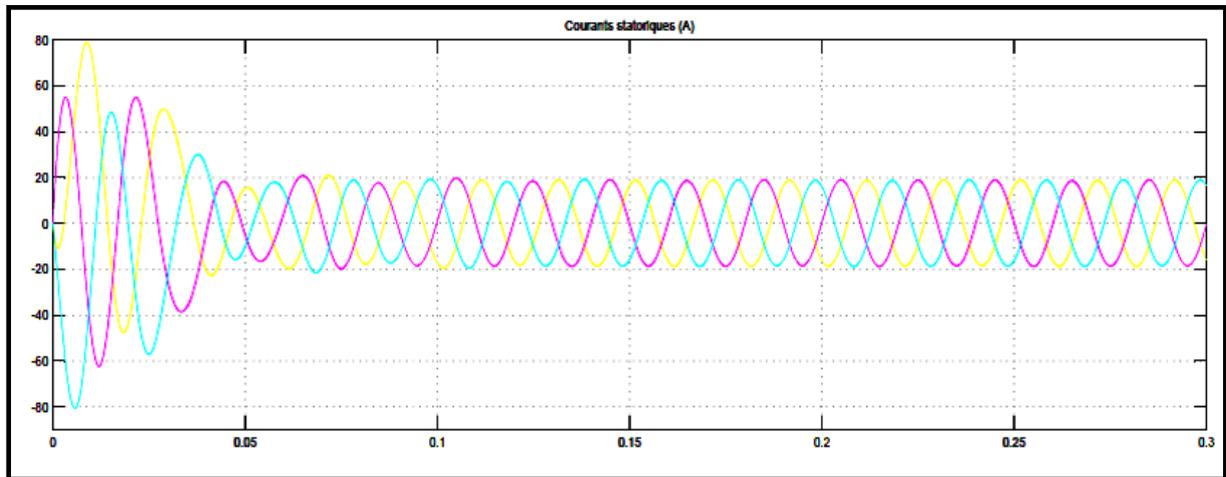


Figure 4.18 Allures des trois courants statoriques I_{s_a} , I_{s_b} et I_{s_c} à la fréquence 50 Hz.

La figure suivante illustre les allures des courants en régime permanent :

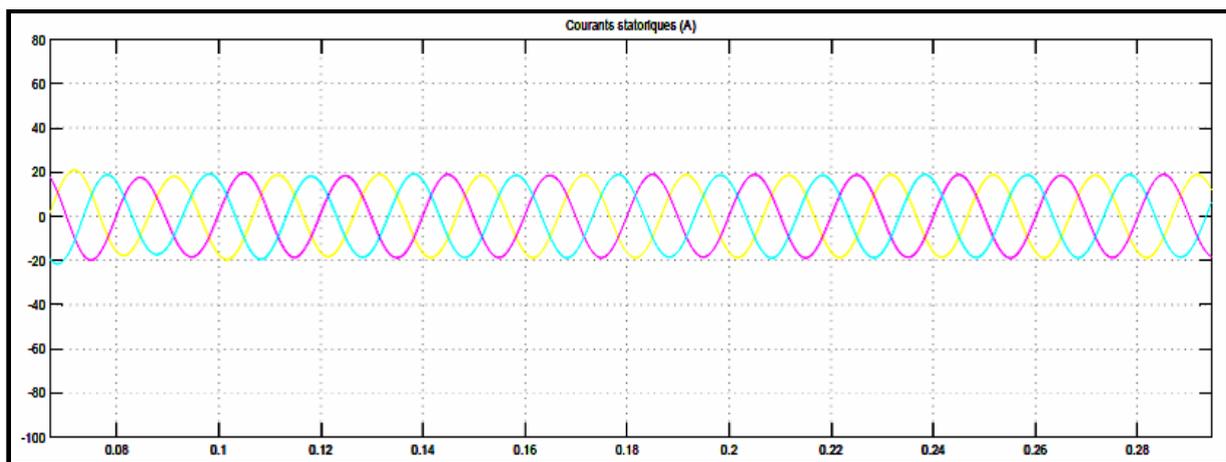


Figure 4.19 Le déphasage entre deux courants statoriques I_{s_a} , I_{s_b} et I_{s_c} est de 120° .

On constate qu'en régime permanent les amplitudes des courants de phase se stabilisent à la valeur 20A et leurs fréquences suivent exactement la fréquence 50 Hz imposée. De plus, on remarque clairement que les courants des trois phases sont déphasés de 120° l'un de l'autre. Ainsi avec ces trois caractéristiques (mêmes fréquence et amplitude et déphasage 120°), on vérifie que le stator est alimenté avec un système triphasé équilibré.

4.5 Simulation de la réponse en vitesse

❖ Consigne 50 Hz

La figure 4.20 illustre l'allure du courant à la fréquence de référence 50 Hz, et l'allure d'évolution de la vitesse du rotor à vide (pas de couple mécanique résistant).

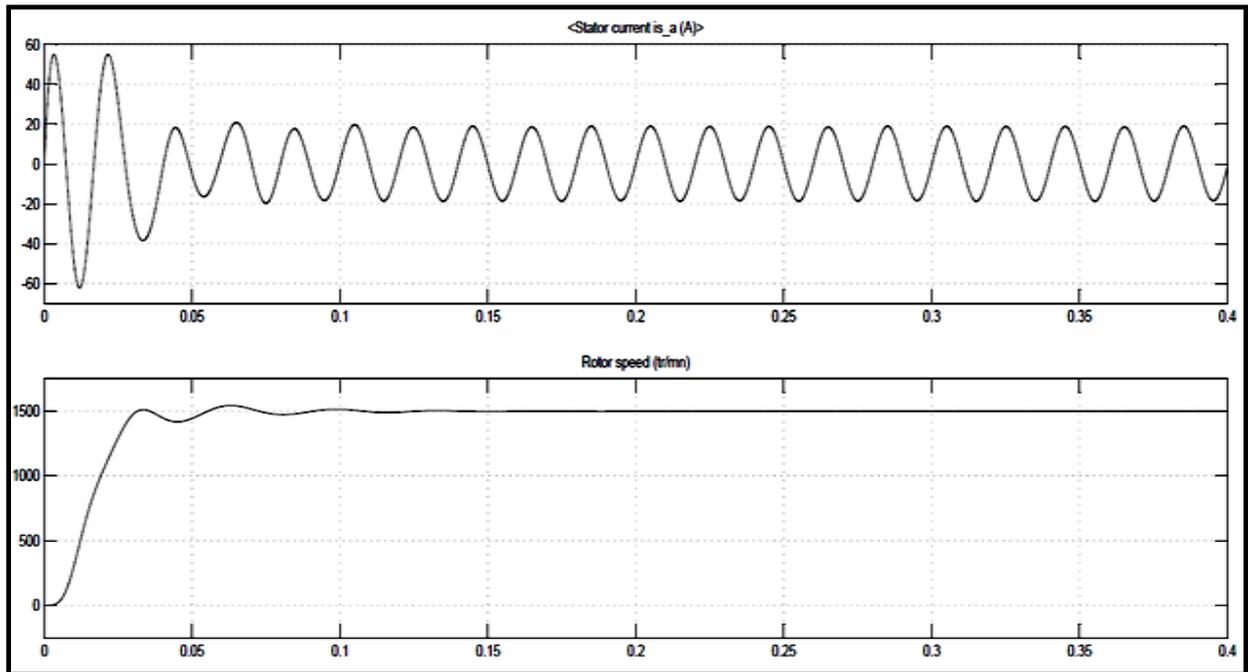


Figure 4.20 Allures du courant et de la vitesse du rotor à la fréquence de référence 50 Hz.

On remarque qu'après le passage de la phase de démarrage (régime transitoire), la vitesse du rotor se stabilise en régime permanent à la vitesse de synchronisme 1500 tr/mn (pas de glissement) puisque le moteur tourne à vide.

Note : $1500 \text{ tr/mn} = (50 \text{ Hz}) (60 \text{ s}) / P$ avec : ($P=2$: nombre de paires de pôles).

❖ Consigne 25 Hz

La figure suivante illustre l'allure du courant pour une fréquence de référence 25 Hz, et l'allure d'évolution de la vitesse du rotor à vide.

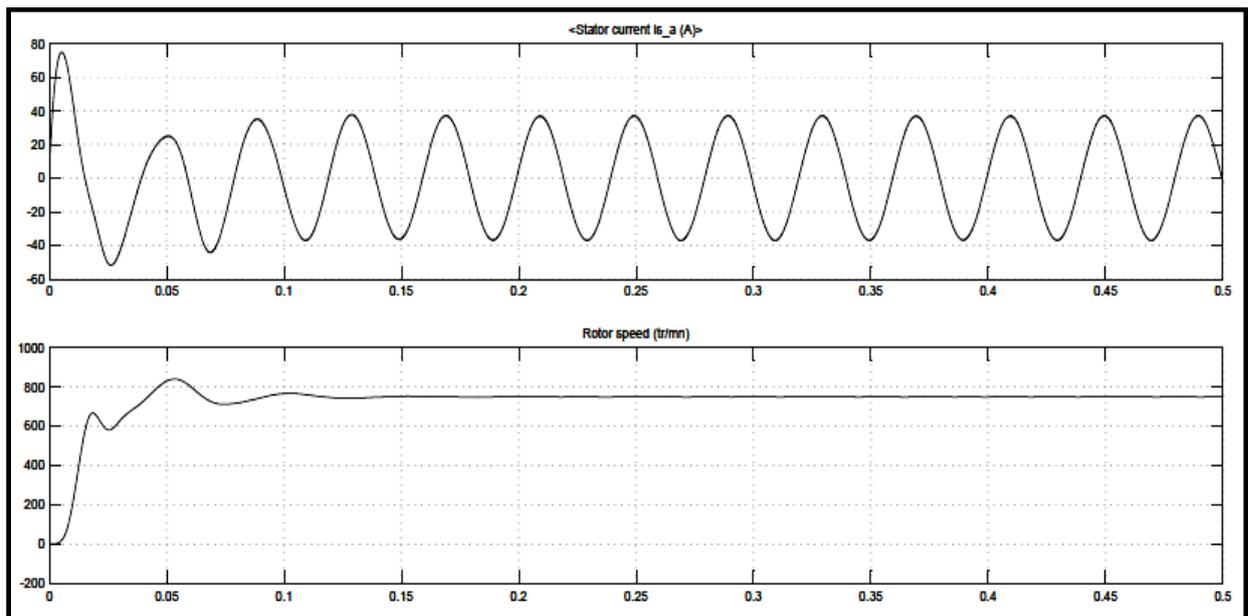


Figure 4.21 Allures du courant et de la vitesse du rotor pour la fréquence 25 Hz.

4.6 Simulation lors de changement de vitesse

L'application d'une fréquence de référence 25 Hz suivie d'une autre référence 50 Hz nous a permis de simuler le comportement du moteur et de voir l'évolution du courant de phase lors de la modification de la fréquence de référence pour changer la vitesse du moteur.

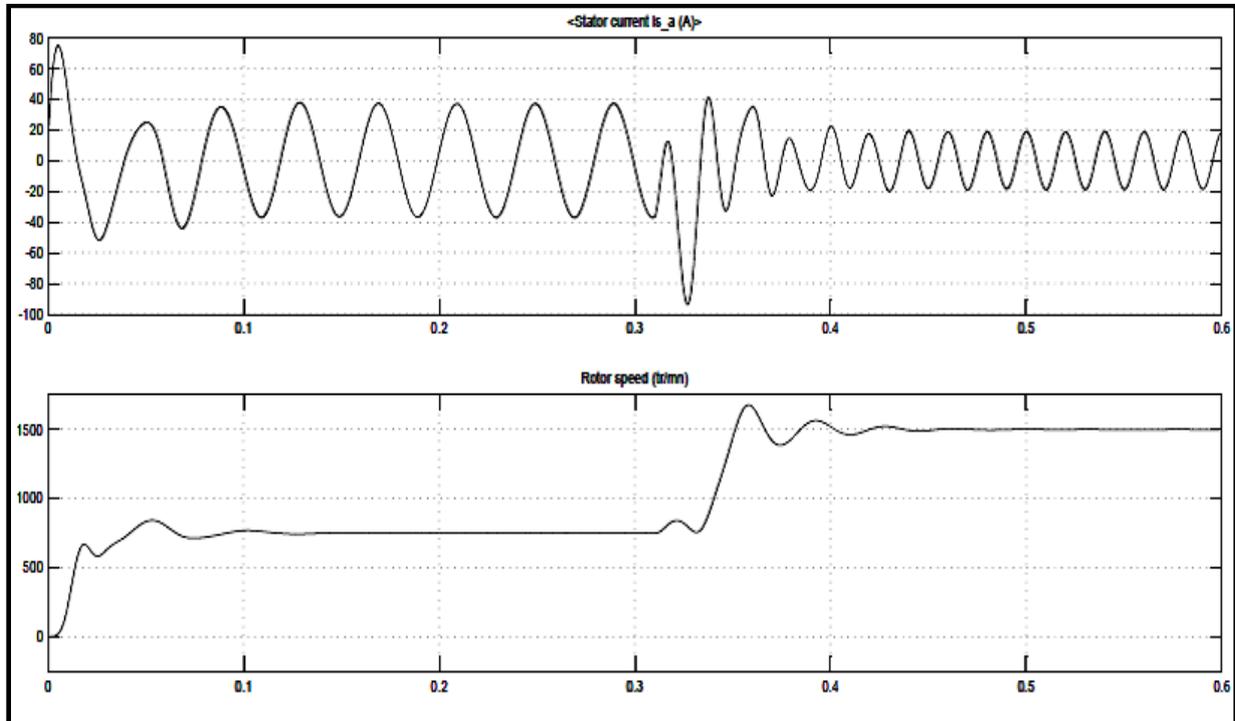


Figure 4.22 Allures du courant et de la vitesse lors de changement de consigne de 25 à 50 Hz

On remarque la présence des pics de courant passagers (de courte durée) dans la phase de démarrage et lors de la phase de changement de vitesse.

4.7 Simulation avec couple mécanique résistant

L'application d'une consigne de fréquence 50 Hz, et d'un couple mécanique résistant de 10 N.m, nous a offert les allures sur la figure 4.23 pour le courant de phase, la vitesse du rotor et pour le couple électromagnétique développé par le moteur.

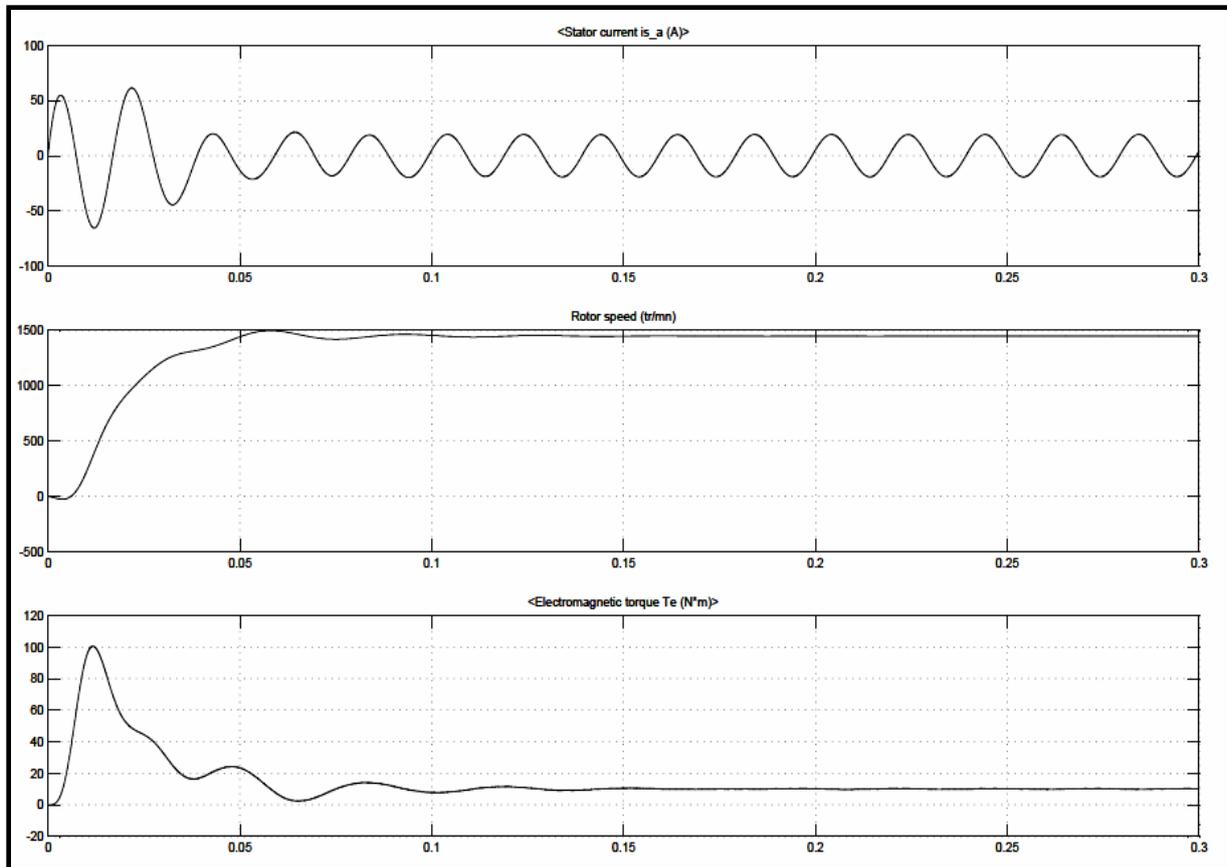


Figure 4.23 Allures du courant, de la vitesse et du couple à 50 Hz et une charge de 10 N.m.

En régime transitoire, on remarque des amplitudes élevées pour le courant de phase du stator et des oscillations amorties pour le couple électromagnétique lors de la phase de démarrage. En régime permanent on note que le couple électromagnétique se stabilise à la valeur du couple résistant appliqué 10 N.m.

Cette fois-ci le rotor tourne à une vitesse légèrement inférieure à la vitesse de synchronisme 1500 tr/mn à cause de la présence du glissement dû au couple mécanique 10 N.m appliqué.

4.8 Simulation lors de l'inversion du sens de rotation

Dans cette partie, on a simulé le comportement du moteur lors de l'inversion du sens de rotation et en maintenant la fréquence de référence à la valeur fixe 50 Hz et sans charge. La figure 4.24 (ci-après) illustre les allures de l'évolution du courant statorique, de la vitesse du rotor et du couple électromagnétique.

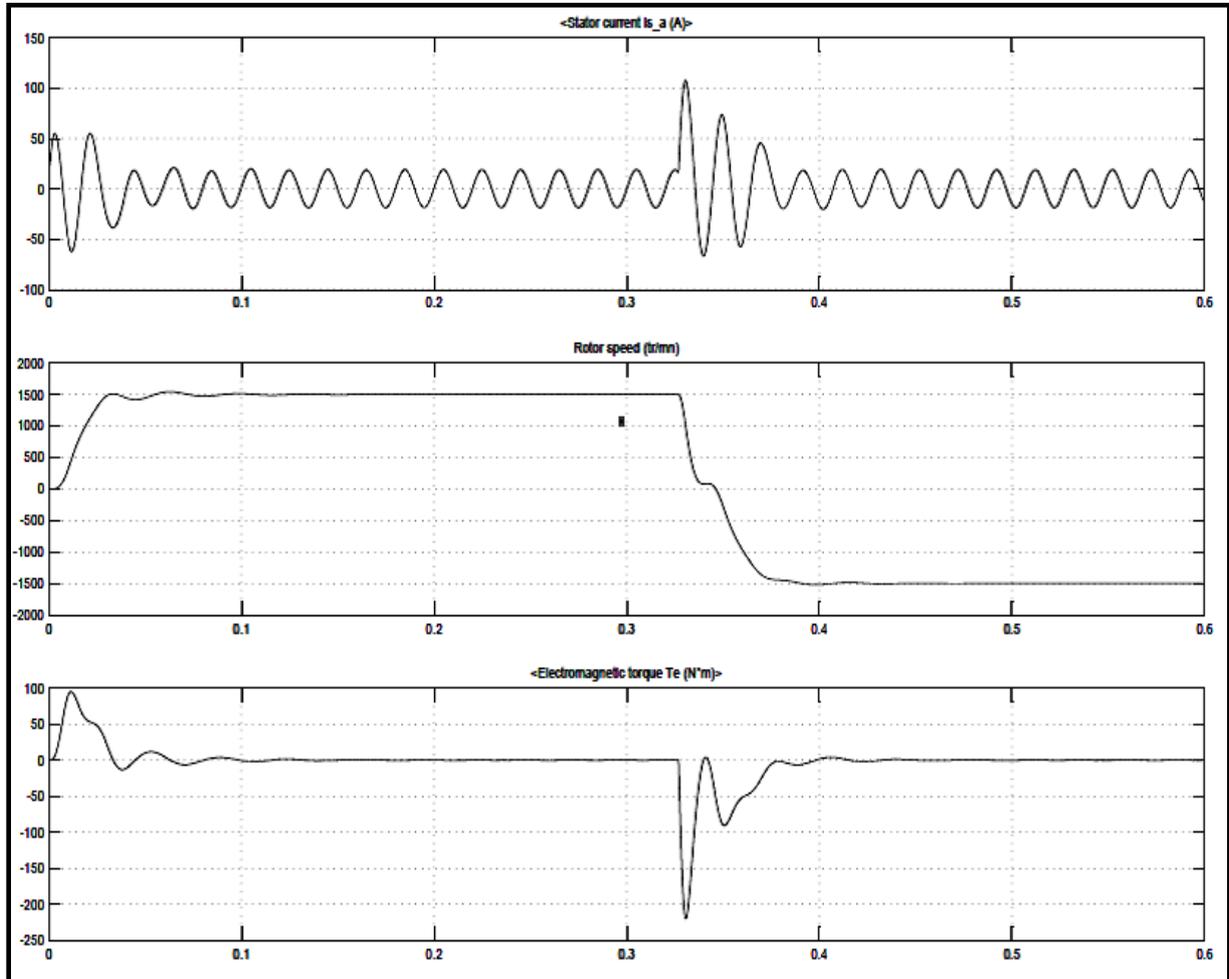


Figure 4.24 Allures du courant, de la vitesse et du couple lors changement du sens de rotation.

Lors de la phase de changement du sens de rotation, on constate des amplitudes élevées qui atteignent 100 A (deux fois sa valeur lors du démarrage 50 A) pour le courant et la présence d'oscillations amorties pour le couple électromagnétique développé durant cette phase transitoire. En régime permanent, on note que le couple électromagnétique développé est nul, puisque il n'y a pas de couple mécanique résistant imposé.

4.9 Simulation de la phase d'arrêt du moteur

Dans cette partie de simulation on a essayé de simuler le comportement du moteur lors de la phase d'arrêt. Pour cette simulation on a appliqué une fréquence de référence 50 Hz et un couple mécanique résistant de 10 N.m. La figure ci-après illustre les allures de l'évolution du courant statorique, de la vitesse du rotor et du couple électromagnétique.

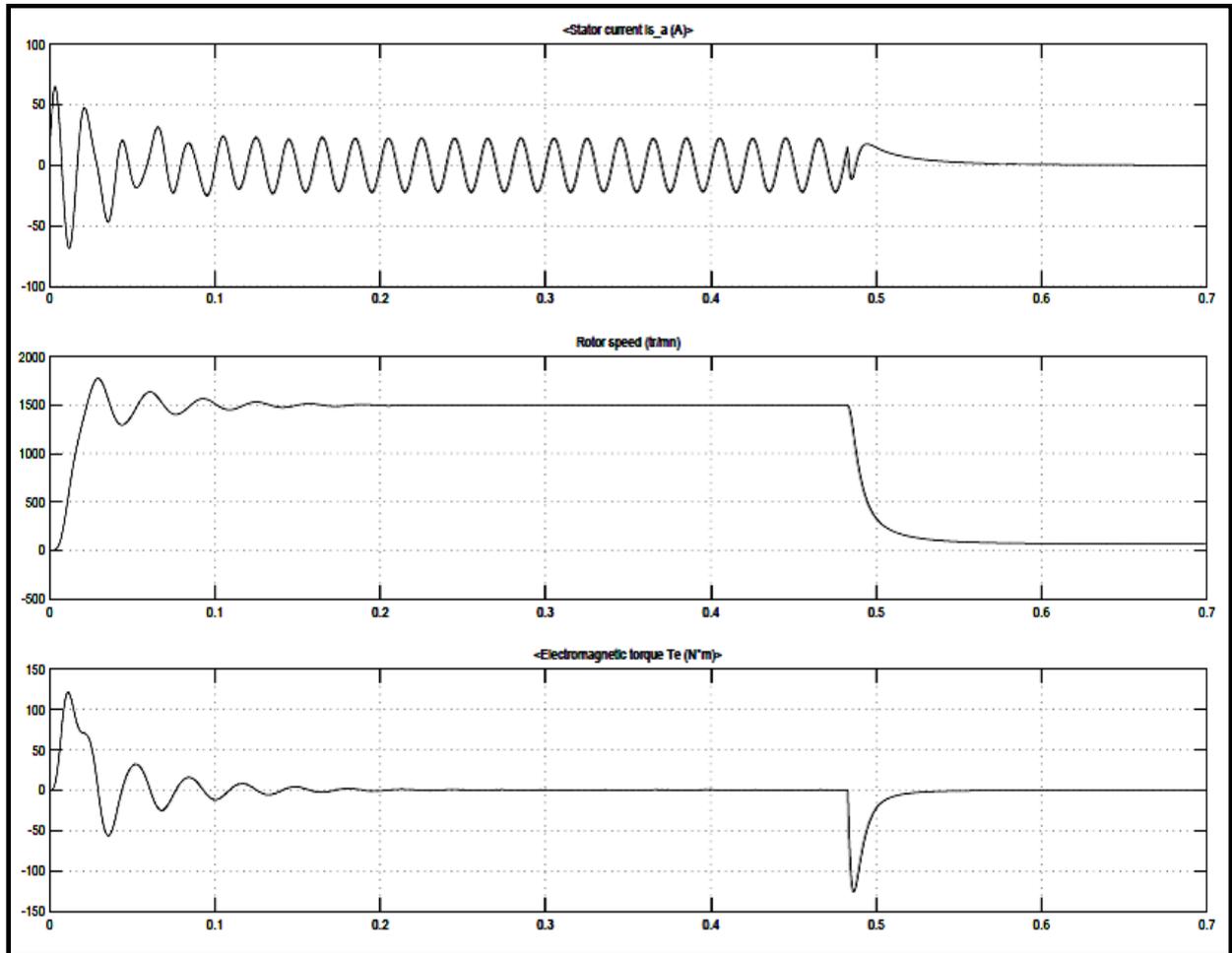


Figure 4.25 Allures du courant, de la vitesse et du couple électromagnétique lors d'un arrêt.

Dans cette simulation, on a remarqué qu'à l'inverse de la phase de démarrage, dans la phase d'arrêt du moteur ne se présentent pas les pics de courant et les oscillations du couple électromagnétique constatés dans la phase de démarrage. Ainsi, l'arrêt s'effectue en douceur et sans que le moteur soit affecté par les risques engendrés par les pics de courant élevés.

Conclusion

La grande flexibilité de la simulation des systèmes de contrôle et de puissance sous MATLAB, nous a permis de concevoir une commande de vitesse en boucle ouverte d'un moteur asynchrone en utilisant la technique de modulation vectorielle SVM. Cette technique de modulation présente l'avantage d'être d'une implémentation très simple sur calculateur. La simulation nous a permis de voir et comprendre le comportement du moteur asynchrone et de tester les performances de notre commande avant sans implantation en pratique sur le DSP.

Chapitre V

Conception matérielle, implantation logicielle et résultats pratiques

Ce chapitre structuré en trois parties, résume le travail personnel accompli en pratique le long du projet. Dans la première partie on va donner une description concernant la carte électronique adoptée comme solution pour commander la vitesse d'un moteur asynchrone. Dans la deuxième partie, sera dévoilé l'aspect logiciel de la réalisation en exposant l'algorithme global de la commande et son implantation sur la carte eZdsp TMS 320F2812. La dernière partie sera consacrée aux essais effectués sur le moteur asynchrone commandé par le système électronique réalisé, et aux résultats pratiques obtenus en illustrant les graphes des grandeurs électriques prélevés à l'aide de l'oscilloscope et qui sont suivis de commentaires.

1. Etude et réalisation de la carte électronique

1.1 Représentation synoptique du système

Le système global de la réalisation est constitué de la partie commande (DSP) et de la partie puissance (onduleur triphasé + moteur asynchrone), qui sont isolées électriquement par l'interface optoélectronique. La figure 5.1 illustre le schéma synoptique de la réalisation :

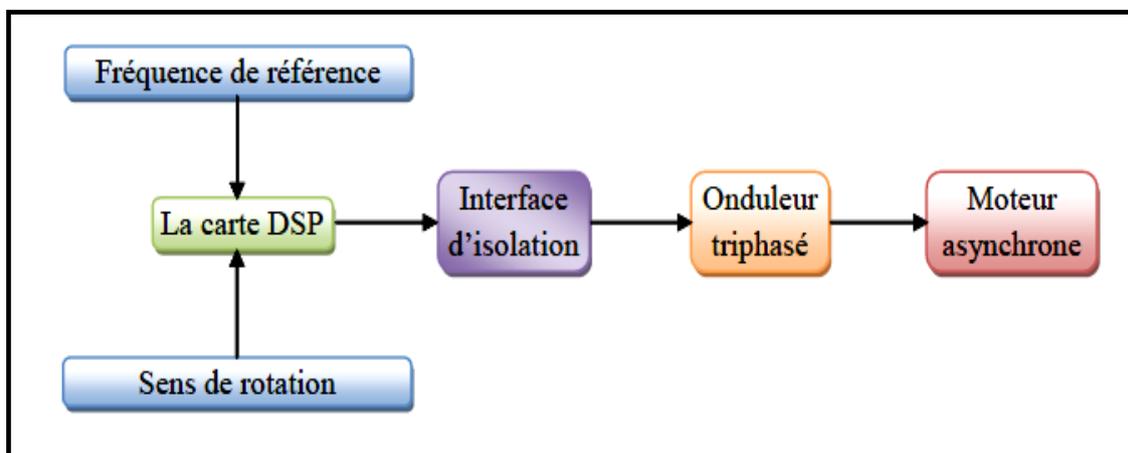


Figure 5.1 Schéma synoptique de la carte.

Le système complet comporte les blocs suivant :

- La carte eZdsp TMS 320F2812 avec son alimentation 5V DC.
- Le mécanisme d'introduction des données constitué, d'un potentiomètre pour introduire la fréquence de référence, et d'un interrupteur On/Off pour le choix du sens de rotation.
- L'interface d'isolation galvanique pour protéger la carte eZdsp TMS 320F2812.
- Le convertisseur de puissance représenté par l'onduleur de tension triphasé.
- Le système commandé qui est un simple moteur asynchrone (MAS).

La carte eZdsp TMS 320F2812 représente le noyau de notre système, puisqu'elle constitue la partie fonctionnelle intelligente qui s'occupe de l'analyse et du traitement des données reçues de l'environnement extérieur (fréquence de référence et sens de rotation), et puis de générer les signaux PWMs convenables pour la commande du processus (MAS).

L'interface d'isolation galvanique composée essentiellement d'opto-coupleurs est primordiale afin d'assurer la protection la carte DSP à faibles signaux, de la partie constituée de l'onduleur et du moteur qui manipulent des signaux de puissance.

1.2 Description des différents blocs de la carte

1.2.1 La carte eZdsp TMS 320F2812

Le eZdsp TMS 320F2812 est une carte sur circuit imprimé multicouche alimentée sous une tension DC de 5 Volt. En général l'unité demande un courant de 500 mA, mais si d'autres cartes d'extension sont connectées à la carte eZdsp une alimentation d'ampérage supérieur peut paraître nécessaire. La carte eZdsp TMS 320F2812 est caractérisée par : [33].

- Fonctionnement sous 5-Volts.
- Une horloge quartz de 30 MHz.
- Un processeur DSP TMS 320 F2812.
- Une vitesse de traitement de 150 MIPS.
- Connecteur d'émulation IEEE 1149.1 JTAG.
- Deux connecteurs d'extensions (analogique, I/O).
- Une mémoire flash de 128K mots (On-chip).
- Une mémoire SARAM de 64K mots (Off-chip).
- Un bloc 16 x 8K de mémoire SARAM (On-chip).
- Deux blocs 16 x 1K de mémoire SARAM (On-chip).
- Deux blocs 16 x 4K de mémoire SARAM (Single Access RAM) (On-chip).

La figure 5.2 illustre la carte eZdsp TMS 320F2812. [36]

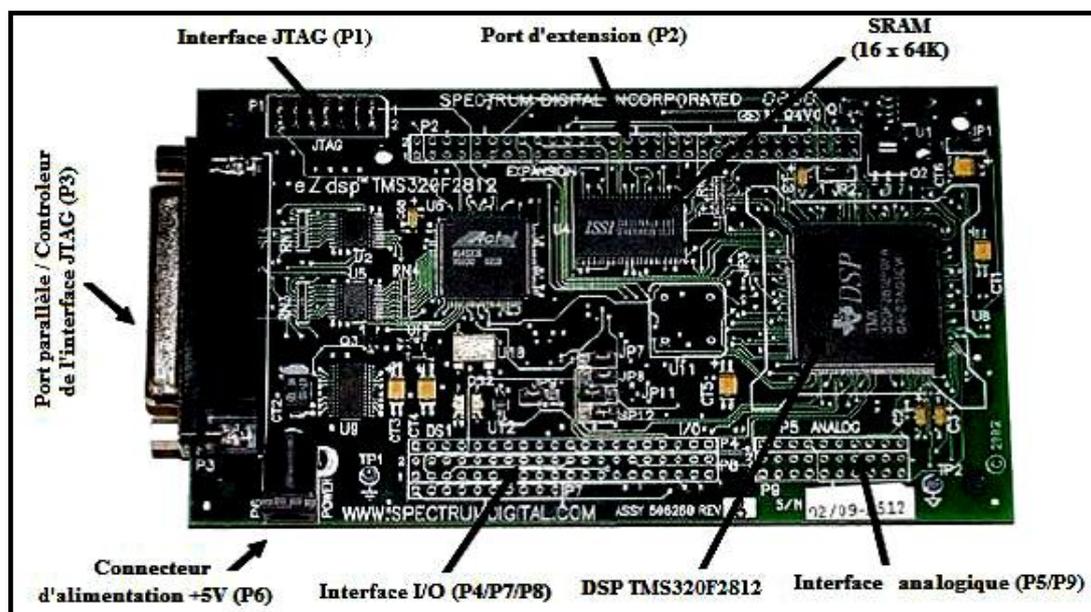


Figure 5.2 La carte eZdsp TMS 320 F2812 de Spectrum Digital.

1.2.2 Mécanisme d'introduction de données

Le mécanisme d'introduction des données est constitué d'un potentiomètre pour introduire la fréquence de référence et d'un interrupteur pour le choix du sens de rotation. En effet la valeur numérique de la fréquence de référence est introduite sous forme d'un signal analogique qui est la tension sur la sortie du potentiomètre connectée au canal ADCIN0 du convertisseur ADC. Ce dernier, permet de convertir le signal analogique (tension sur la sortie du potentiomètre) en valeur numérique (fréquence de référence), afin de la traiter par le DSP.

La sélection du sens de rotation est réalisée à l'aide de l'interrupteur connecté à la pin GPIO B0, à l'ouverture de l'interrupteur Figure 5.3 la tension sur la pin B0 est 3V, ce qui traduit l'un des sens de rotation, lors de sa fermeture la pin B0 est mise à la masse GND (0V), ce qui traduit l'autre sens de rotation.

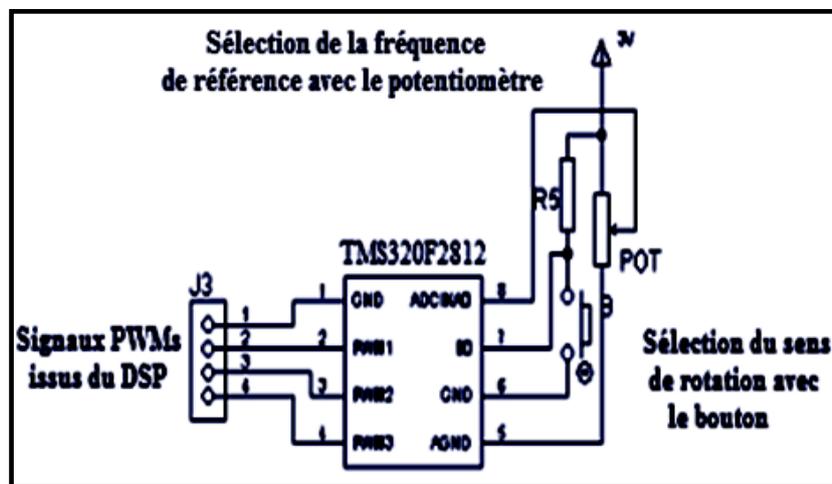


Figure 5.3 Mécanisme de sélection de la fréquence et du sens de rotation.

1.2.3 Le bloc d'alimentation

Le circuit d'alimentation nécessaire pour le fonctionnement des circuits et composants électroniques (Opto-coupleurs et driver) est dérivé directement de la tension du bus continu VDC qui alimente l'onduleur. Le circuit est illustré par la figure 4.5.

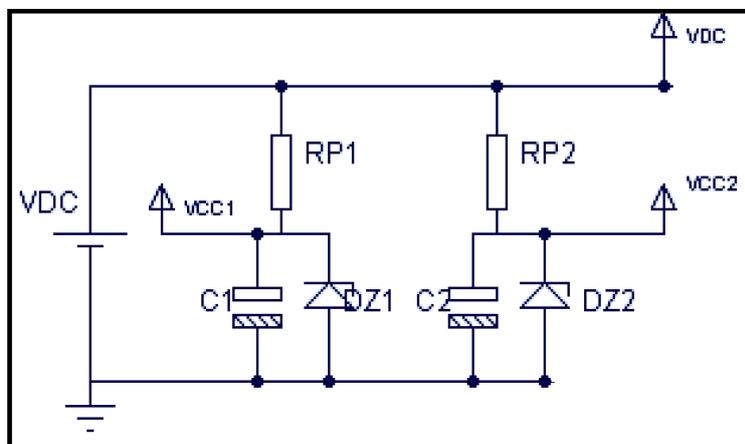


Figure 5.4 Le circuit d'alimentation.

La tension d'alimentation des opto-coupleurs est VCC1 (12V) et celle des drivers est VCC2 (16V), et sont produites à l'aide de deux circuits de stabilisation de tension. Ces circuits sont composés essentiellement de diodes Zener DZ1 et DZ2, connectées en série à des résistances de puissance RP1 et RP2 pour assurer la chute de la tension VDC.

Le circuit d'alimentation est réalisé avec les composants suivants :

- DZ1 (12V) et DZ2 (16V) : Diodes Zener pour la stabilisation de la tension.
- RP1 et RP2 : Résistances de puissance pour assurer la chute de la tension.
- C1 et C2 : Condensateurs de filtrage.

1.2.4 L'interface d'isolation galvanique

L'interface d'isolation est constituée principalement de trois circuits opto-coupleurs alimentés sous la tension VCC1 (12V). Les opto-coupleurs assurent la transmission sous forme de lumière des signaux électriques issus de la carte DSP vers la partie puissance sans qu'il y ait de contact physique entre ces deux parties. La figure 5.5 illustre le schéma électronique de l'interface d'isolation galvanique.

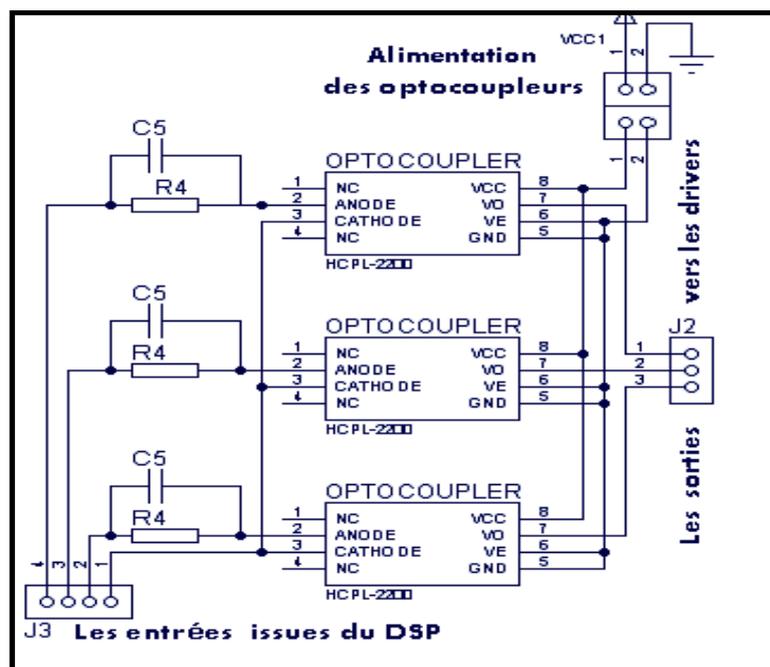


Figure 5.5 Le circuit de l'interface d'isolation galvanique.

Dans le circuit de l'interface galvanique :

- Les résistances R4 servent à limiter les courants dans les photodiodes et assurer leur sécurité.
- Les capacités C5 d'ordre de dizaines de Pico Farad (120 pF) sont optionnelles, leur rôle est d'améliorer le temps de propagation du signal en réduisant les valeurs des temps de montée t_r (rise time) et du temps de descente t_f (fall time). [5]

La figure 5.6 illustre le schéma fonctionnel de l'optocoupleur [5] :

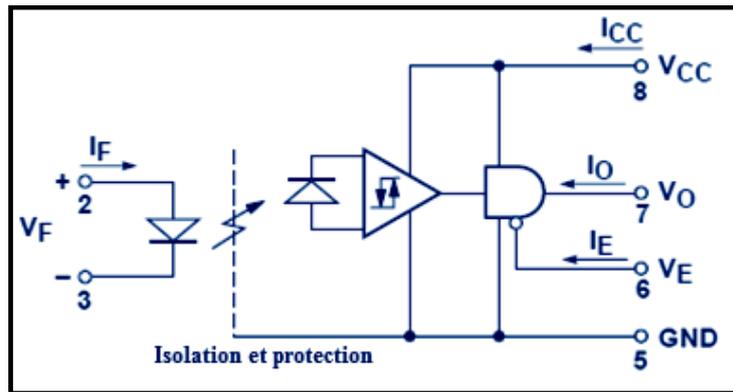


Figure 5.6 Schéma fonctionnel de l'opto-coupleur HCPL 2200.

1.2.5 La partie puissance

La partie puissance est constituée de l'onduleur de tension triphasé et du processus commandé qui est le moteur asynchrone. La partie puissance est illustrée par la figure 5.7

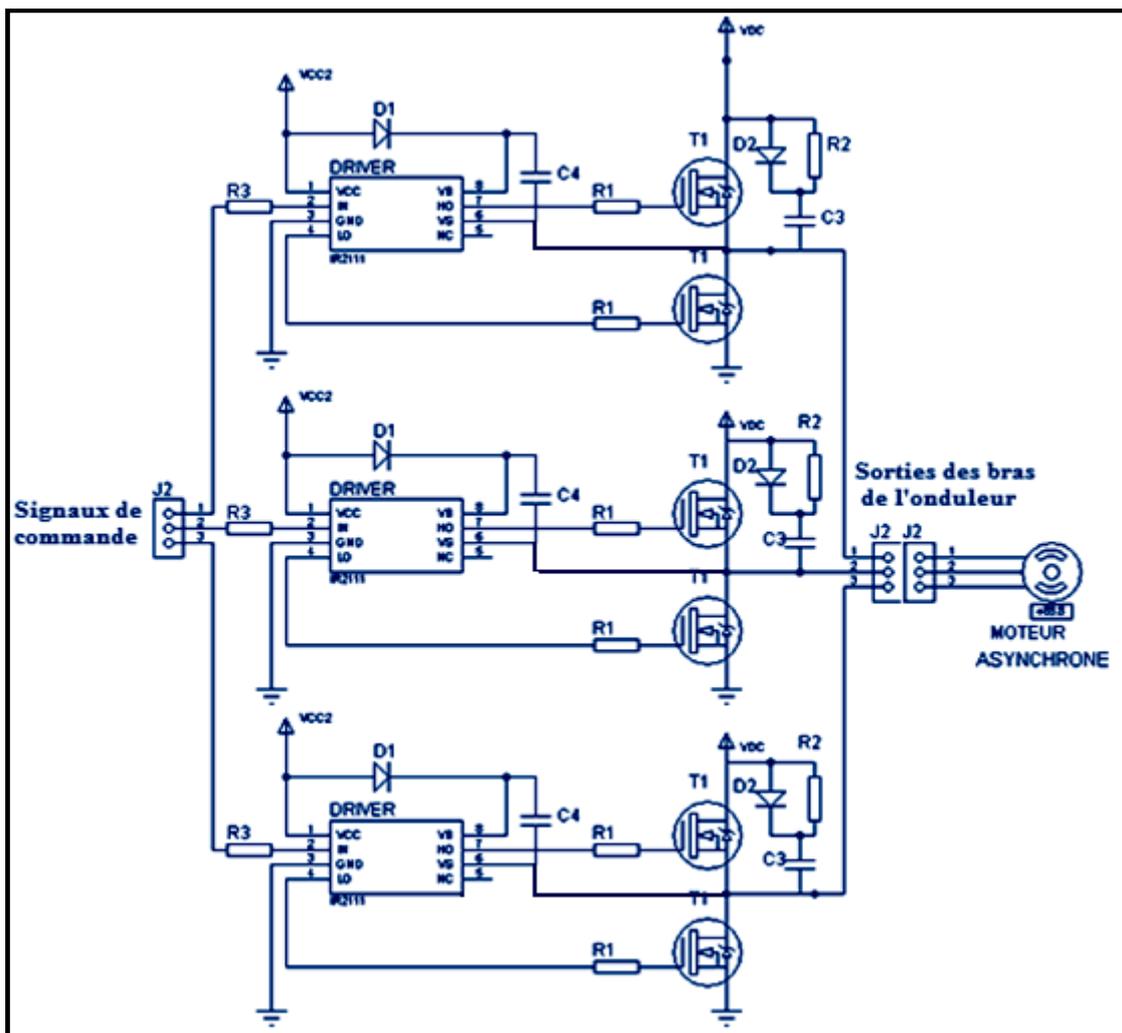


Figure 5.7 Le bloc de la partie puissance.

Sur la figure 5.7, l'onduleur triphasé est formé de trois demi-ponts en parallèle à base de transistors de puissance MOSFET. Chaque demi-pont est précédé d'un driver qui sert à produire une masse flottante afin de pouvoir amorcer le transistor high-side. Ce dernier, comme on le note sur la figure 5.7 est shunté par un circuit RDC qui est un simple snubber servant à limiter ou éviter les pics de tension dangereux causés par les circuits inductifs parasites, lors des commutations (fermeture-ouverture) rapides des transistors MOSFET. Chaque bloc de la partie puissance est présenté, dans la suite, avec plus de détails :

1.2.5.1 Le circuit driver de transistors de puissance

La circuitrie du montage du driver est présentée sur la figure suivante :

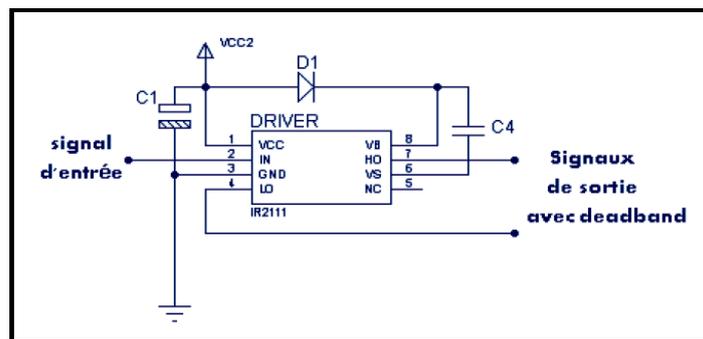


Figure 5.8 Le montage typique du driver.

Dans le montage de la figure 5.8, le rôle du condensateur C1 est d'assurer le filtrage de l'alimentation VCC, La tension VB est obtenue à partir de la tension VCC, par la charge du condensateur Boot-strap C4 à travers la diode D1, quand le transistor low-side conduit.

a) Fonctionnement du driver

Le driver est un circuit intégré qui permet le pilotage en haute fréquence de deux transistors (MOSFET ou IGBT), sous des tensions pouvant atteindre des centaines de volts, en garantissant l'isolation électrique de chaque étage du driver [2]. La figure 5.9 montre le diagramme fonctionnel du driver [21].

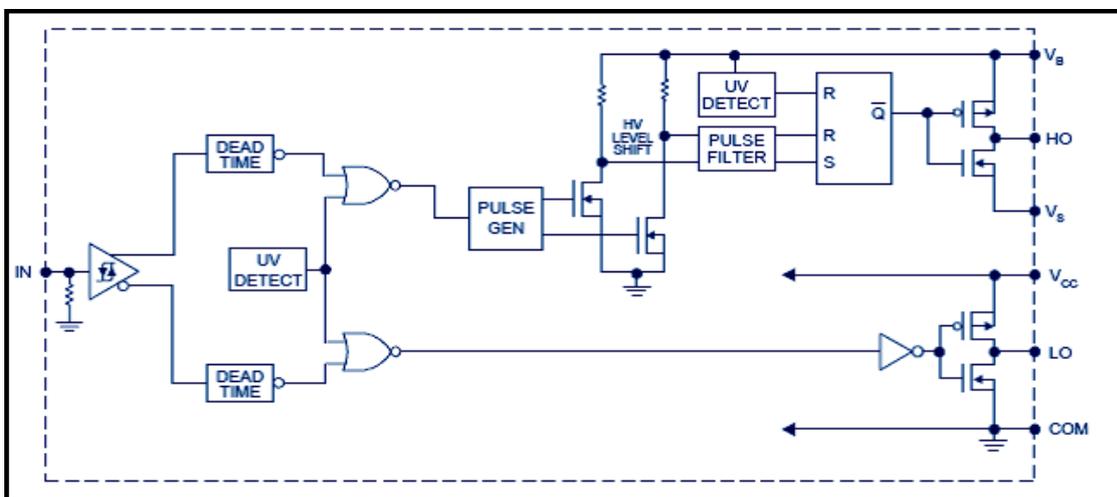


Figure 5.9 Le diagramme fonctionnel du driver.

b) Fonctionnement de l'entrée logique

L'entrée logique IN est compatible CMOS et TTL. Les niveaux de basculement sont liés à la valeur de la tension d'alimentation VCC. Par exemple, pour $V_{CC} = 15V$, le niveau haut de l'entrée IN est compris entre 9,5V et 15V, et le niveau bas est compris entre 0V et 6V.

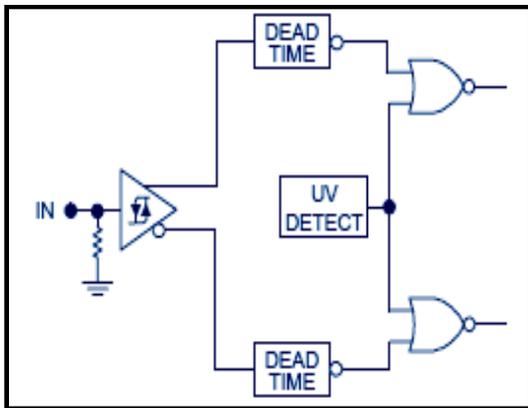


Figure 5.10 (a) Etage d'entrée.

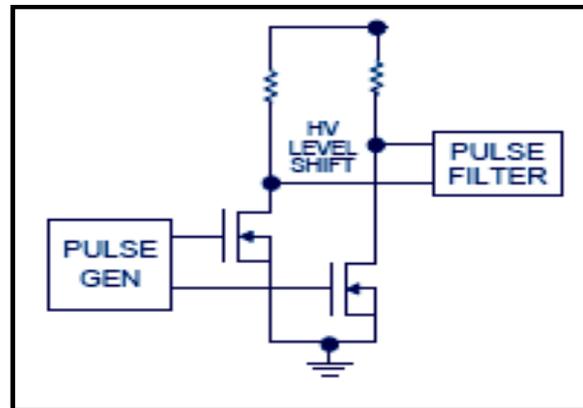


Figure 5.10 (b) Étage de décalage de niveau haut.

Les impulsions qui arrivent sur l'entrée IN vont reproduire deux séries d'impulsions en sortie avec un temps mort (Dead-time) intégré, d'une valeur de quelques centaines de nano secondes (650 ns pour la valeur typique).

c) Les étages de sortie

Les figures ci-dessous illustrent les diagrammes fonctionnels des étages de sorties :

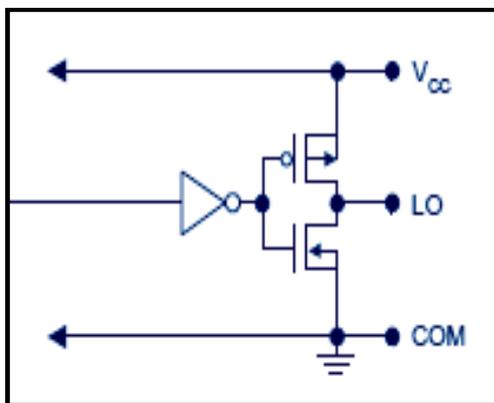


Figure 5.11 (a) L'étage de sortie bas.

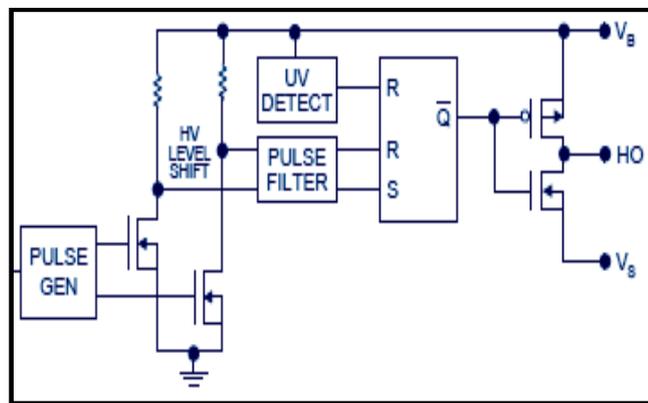


Figure 5.11 (b) L'étage de sortie haut.

L'étage bas du driver dispose d'une sortie LO pouvant délivrer un courant 200 mA par l'intermédiaire du montage push-pull formé des deux transistors complémentaires. Ainsi que pour l'étage haut, qui dispose d'une sortie HO qui délivre un courant de 420 mA.

Pour l'étage bas lorsque la tension VCC est insuffisante, le transistor High-side devient bloqué et l'autre est saturé. Donc la sortie sera mise à la masse COM et ne dépend pas de l'état de l'entrée. Dans le cas contraire (tension VCC suffisante) l'état de la sortie LO sera fixé par l'état logique du signal présent sur l'inverseur.

Idem pour l'étage haut lorsque la tension V_B est insuffisante, le transistor High-side est bloqué et l'autre est saturé donc la sortie est à la tension V_S et ne dépend pas de l'état de la sortie \bar{Q} de la bascule RS. Dans le cas contraire (tension V_B suffisante), l'état logique de la sortie HO est déterminé par le bloc Pulse Filter.

d) Technique de Boot-strap [7]

La technique de Boot-strap permet de réaliser une alimentation flottante pour le circuit de commande du transistor high side, à partir de l'alimentation V_{CC} du circuit de commande du transistor Low-side. [7]

Le principe de la technique Boot-Strap illustrée par le montage sur la figure 5.12 consiste à charger la capacité Boot-strap C_B sous la tension V_{CC} à travers la diode D_B au moment de conduction (fermeture) du transistor Low-side.

Lorsque le transistor High-side passe de l'état bloqué (ouvert) à l'état saturé (fermé), le point milieu du bras de l'onduleur passe de 0 à U_{DC} , l'armature positive de la capacité C_B se trouve alors portée au potentiel $U_{DC} + V_{CC}$. Ainsi on aura le blocage de la diode D_B puisqu'elle est polarisée en inverse dans ce cas. Cette diode (D_B), doit donc être capable de supporter la haute tension U_{DC} du bus continu.

La charge portée par C_B sert d'alimentation pour le circuit de commande du transistor High-side, pendant la durée de conduction de ce dernier.

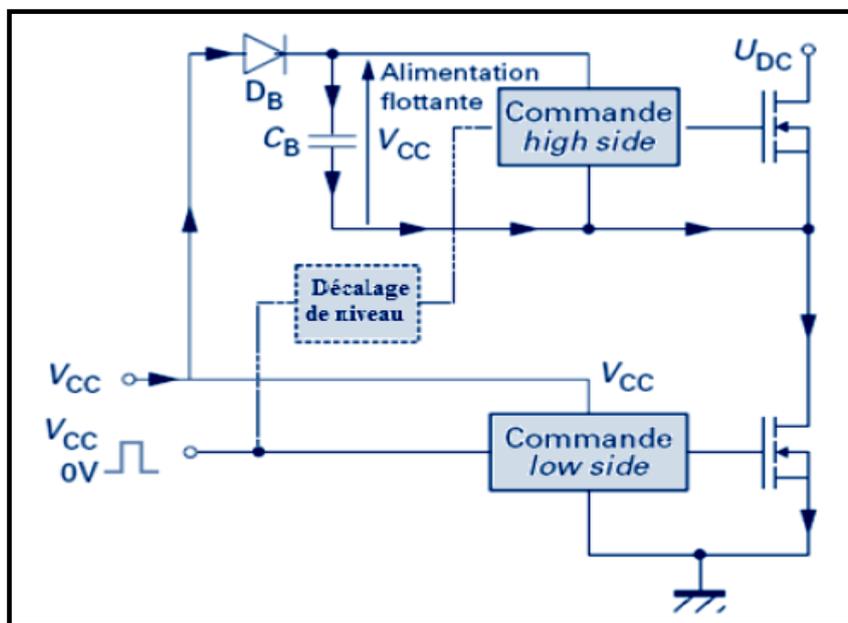


Figure 5.12 Principe de la technique Boot-strap.

❖ Choix du condensateur de Boot-strap C_B

La durée de conduction du transistor Low-side doit être assez suffisante pour que le condensateur Boot-strap C_B puisse se charger totalement à la tension d'alimentation V_{CC} . À l'inverse, la durée de conduction du transistor High-side ne doit pas être très longue pour éviter la décharge complète du condensateur Boot-strap C_B , compte tenu de la consommation du circuit de commande, donc la fréquence de découpage ne doit pas être trop basse.

Donc le dimensionnement du condensateur Boot-strap C_B prend donc en considération la fréquence de découpage (commutation) comme paramètre principal, de plus la valeur de C_B dépend de la tension d'alimentation V_{CC} et de la tension nécessaire sur la grille pour mettre en conduction le transistor High-side. [7]

❖ Choix de la diode de Boot-strap D_B

La tension inverse aux bornes de la diode D_B au moment du blocage est égale à la tension U_{DC} du bus continu alimentant l'onduleur, donc lors de dimensionnement on doit tenir compte de cette contrainte (la tension inverse maximale supportée par la diode), de plus la diode D_B doit être de type rapide.

1.2.5.2 Le bras de l'onduleur

Chaque bras de l'onduleur, est réalisé par assemblage de deux transistors de puissance MOSFET qui travaillent en commutation (saturé ou bloqué) et en complémentarité (l'un bloqué l'autre saturé), et alimenté sous une tension continue VDC comme le montre le montage de la figure 5.13. Dans ce montage le drain du transistor High-side est connecté au potentiel VDC du bus continu et la source du transistor Low-side est à la masse. Le point milieu formé par les interconnexions entre la source du transistor High-side et le drain du transistor Low-side constitue la sortie du bras de l'onduleur.

La sortie du bras ne peut prendre que deux états, la sortie est à la masse (0V) si le transistor Low-side est saturé (le transistor High-side étant nécessairement bloqué), et la sortie est la tension VDC quand le transistor High-side est saturé (le transistor Low-side étant nécessairement bloqué).

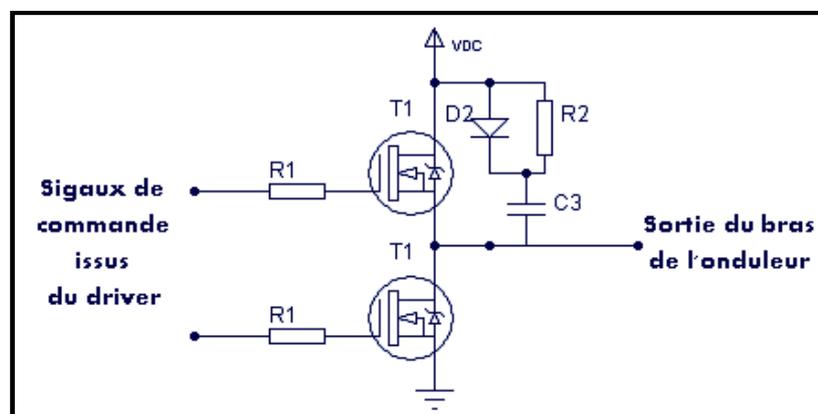


Figure 5.13 Le circuit du bras de l'onduleur.

Pour éviter la conduction (saturation) simultanée des deux transistors, ce qui devrait court-circuiter certainement la tension d'alimentation VDC, on doit garantir un certain temps de retard entre les instants de fermeture d'un transistor et d'ouverture de l'autre. Ce temps de retard, appelé communément temps mort (Dead-time) est assuré par le circuit du driver et qui est d'environ 650 ns.

Le Dead-time (temps mort), est défini comme étant le temps entre l'ouverture du transistor Low-side et la fermeture du transistor High-side, ou le temps entre l'ouverture du transistor High-side et la fermeture du transistor Low-side. Cette définition peut être rappelée par le digramme sur la figure 5.14.

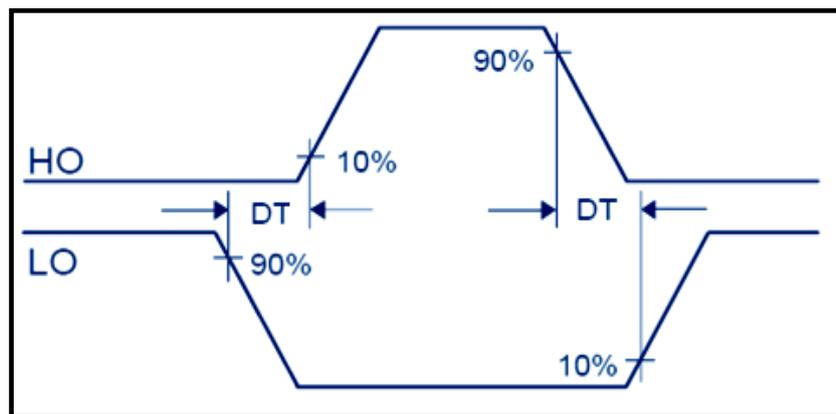


Figure 5.14 Définitions du temps mort (Deadtime).

1.2.5.3 Le circuit Snubber

a) Conception du Snubber RDC

Les circuits Snubbers sont des circuits qui sont placés à travers les interrupteurs qui travaillent en commutation pour la protection et améliorer les performances du système complet. Il y'a plusieurs types de Snubbers, mais les deux les plus courants sont le réseau RC (Résistance-Capacité) et le circuit RDC (Résistance-Diode-Capacité) utilisés durant le temps d'ouverture de l'interrupteur. Les Snubbers RDC (Résistance Diode Capacité) sont de simples circuits d'absorption de l'excès d'énergie dû à la commutation rapide des interrupteurs, ils sont utilisés pour réduire ou éliminer les pics de tension causés par les circuits inductifs lors de l'ouverture d'un interrupteur mécanique ou électronique. Les circuits Snubbers peuvent assurer les fonctions suivantes : [11] [31]

- Réduire les pertes totales dues à la commutation.
- Réduire ou éliminer les pics de tension ou de courant.
- Limiter les variations du courant di/dt ou de la tension dv/dt .
- Réduire les émissions électromagnétiques (EMI) qui perturbent le voisinage.
- Transférer la dissipation de la puissance de l'interrupteur vers une résistance afin de réduire la température de la jonction du transistor sensible à la chaleur.

Le montage du snubber RDC est illustré par la figure suivante :

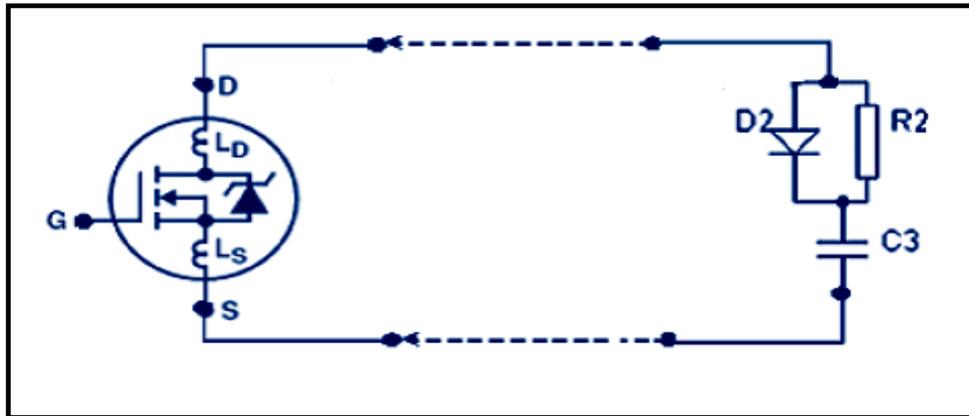


Figure 5.15 Le MOSFET avec ses inductances parasites, shuntés par le Snubber RDC.

Le snubber est actif uniquement durant les transitions de commutation. A l'ouverture (*turn_off*), le courant est dévié (détourné) à travers la diode vers la capacité du snubber jusqu'à ce que la capacité soit chargée à la valeur du pic de tension maximal. A la fermeture (*turn_on*) du transistor, la capacité du snubber sera déchargée à travers la résistance du snubber et le transistor. La capacité doit être complètement déchargée pour bien contrôler et maîtriser la variation dv/dt dans le transistor. [44]

b) Détermination des valeurs des composants du Snubber (RDC)

❖ La Capacité C

Le condensateur C doit assurer une capacité de stockage d'énergie supérieure à l'énergie dans le circuit inductif : [11]

$$E_c > E_l \Rightarrow \frac{1}{2} C V_c^2 > \frac{1}{2} L I^2$$

Ce qui donnera :

$$C > L \frac{I^2}{V_c^2} \quad (5.1)$$

- I : Le courant maximum dans le circuit à la fermeture de l'interrupteur.
- L : Inductance du circuit (fils + inductances parasites des MOSFET) estimée à quelques nH.

❖ La résistance R

La constante du temps RC du Snubber, produite par l'association de la capacité C et la résistance R , doit être plus petite que le temps de fermeture de l'interrupteur ton le plus petit, pour que la capacité du snubber puisse complètement se décharger à travers la résistance R . Une valeur typique pour la constante de temps RC est de prendre le dixième du temps de conduction ton du transistor : [44]

$$RC = \frac{1}{10} ton$$

Ce qui donnera :

$$R = \frac{1}{10} \frac{ton}{C} \quad (5.2)$$

Toute l'énergie stockée dans la capacité sera dissipée dans la résistance à chaque cycle de décharge. La puissance dissipée est donnée par :

$$P = \frac{1}{2} C V^2 f = \frac{1}{2} C V_{dc}^2 f \quad (5.3)$$

- P : La puissance dissipée dans la résistance du Snubber.
- f : Le nombre de cycle de décharge par seconde (la fréquence de commutation).

❖ La Diode D

La diode dans le Snubber RDC doit supporter le pic de tension qui apparaît sur la capacité C du snubber. En général le courant moyen dans la diode est relativement petit, mais les pics de courant sont substantiels. Le pic du courant doit être la base pour la sélection de la diode. Le temps de recouvrement inverse (tr_i) de la diode peut affecter aussi le rôle du Snubber et donc on doit choisir des diodes rapides ou ultra-rapides. [31]

Le temps de recouvrement direct (tr_d) peut constituer une préoccupation, parce que la chute de tension initiale à travers la diode, dans le sens direct, peut être beaucoup plus élevée pour une durée de plusieurs centaines de nano secondes que sa valeur à l'état statique. [31]

1.2.6 Le processus commandé (MAS)

Le processus commandé est un moteur asynchrone triphasé de puissance moyenne de 370 Watt (figure 5.16). Il est caractérisé par une construction fermée à carcasse ventilée, et le rotor est en forme de cage d'écureuil fait en aluminium. Il peut être assimilé à une charge inductive triphasée équilibrée avec neutre isolé et peut être câblé en étoile ou en triangle. Le moteur peut tourner à des vitesses différentes et dans les deux sens de rotation et reçoit son alimentation à partir de l'onduleur MLI qui délivre un système de tensions triphasées. La nature inductive du circuit du stator qui se comporte comme un filtre permet d'avoir des courants pseudos-sinusoïdaux dans les enroulements statoriques du moteur. La vitesse de synchronisme du moteur est directement liée à la fréquence de ces courants, donc pour varier la vitesse de rotation du moteur il suffit d'agir sur la fréquence du réseau d'alimentation.

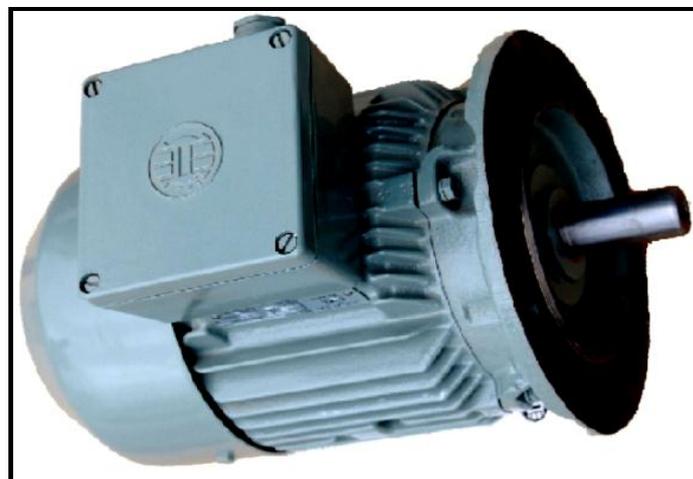


Figure 5.16 Le bloc du moteur asynchrone triphasé.

1.2.7 La carte complète

L'assemblage des différents blocs fonctionnels discutés dans les sections précédentes, nous amène au schéma électronique de la carte complète illustré par la figure 5.17.

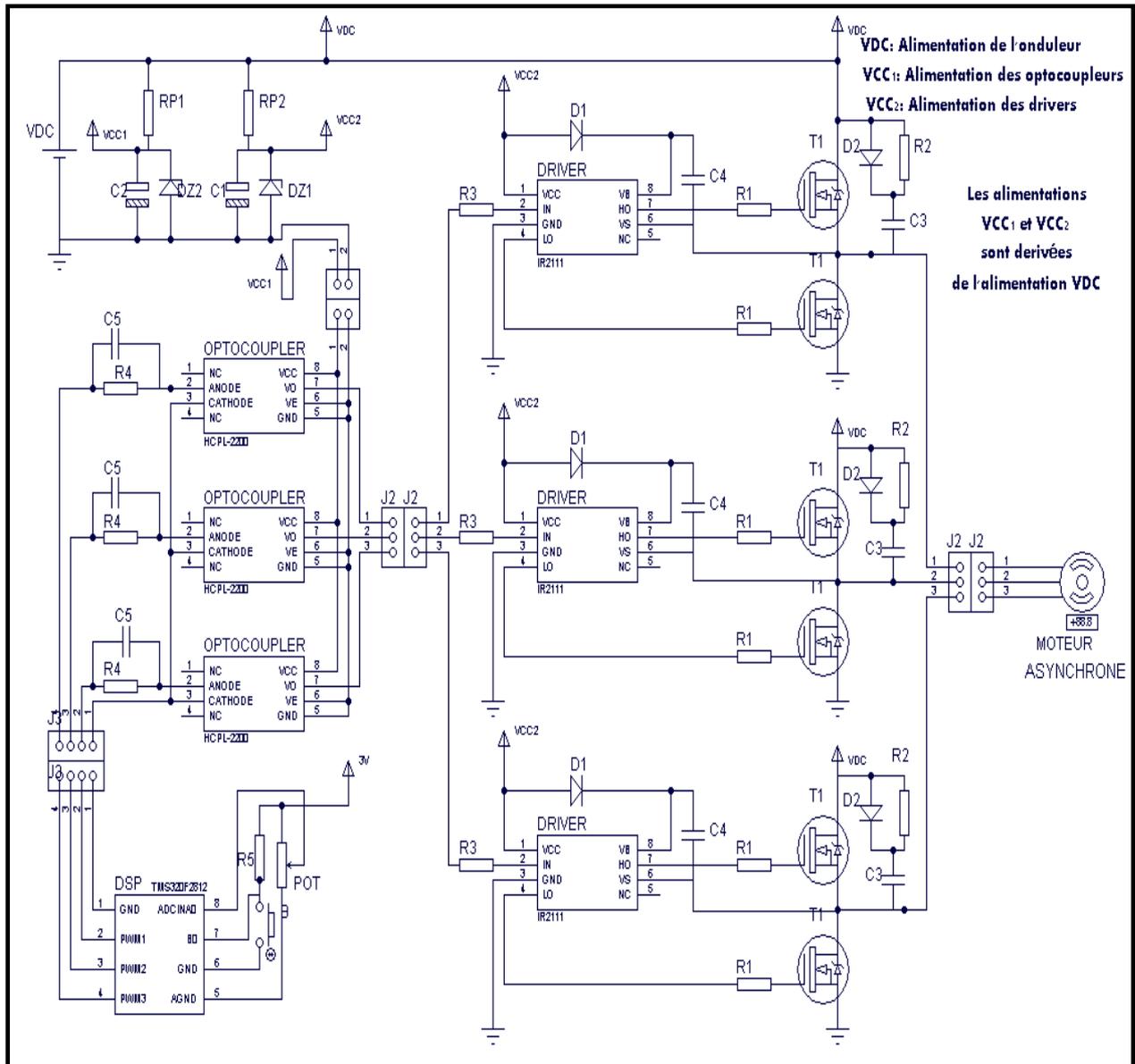


Figure 5.17 Le circuit complet de la carte électronique réalisée.

2. Implantation logicielle

Le DSP TMS 320F2812 comporte toutes les caractéristiques (rapidité de traitement, capacité de la mémoire, périphériques intégrés divers,...etc.) nécessaires pour la commande des différents convertisseurs de puissances et permettre une implémentation aisée pour divers algorithmes de contrôle de moteurs électriques tel que le moteur asynchrone.

Le DSP TMS 320F2812, comporte seize (16) sorties PWM, réparties en deux sorties TxPWM / TxCMP pour le module EVA et deux autres pour le module EVB, les 12 sorties PWM restantes sont les six PWM1-6 du module EVA et les six PWM7-12 du module EVB, et peuvent être programmées avec des temps morts (Dead-band). [43]

En effet, un signal PWM est juste une séquence d'impulsions avec variation de leurs largeurs. Les impulsions sont propagées sur un nombre de périodes de longueurs fixes. La période fixe est appelée la période (porteuse) PWM. Les largeurs des impulsions PWM sont modifiées d'une impulsion à l'autre, et sont déterminées suivant l'algorithme de la technique de modulation implanté dans le calculateur utilisé (DSP, Microcontrôleurs, etc.....). [34]

Le choix de la fréquence de l'onde PWM dépend de la constante de temps électrique L/R du moteur. En effet, si la fréquence est trop basse un bruit audible peut être entendu du moteur. Usuellement, les fréquences PWM sont dans la gamme de 20 KHz. [14]

Dans un système de contrôle d'un moteur (MAS) alimenté par un convertisseur de puissance (Onduleur triphasé), les signaux PWM sont utilisés pour la commande des instants d'ouvertures et de fermetures des interrupteurs de puissance qui permettent de contrôler les formes des courants et la puissance délivrée aux enroulements du moteur. Ainsi, avec le contrôle des fréquences des courants phases et la maîtrise de la quantité d'énergie délivrée aux enroulements du moteur, on assure le contrôle de la vitesse et du couple du moteur.

2.1 Les modules utilisés dans l'implantation logicielle

2.1.1 Le module EV

Les modules Event-manager (EV) fournissent une large gamme de caractéristiques et fonctions qui sont particulièrement utiles dans le développement des applications de contrôle de moteurs, de robotiques et d'automobiles. Le module EV comporte des timers, des unités full-compare/PWM, des unités capture, et des circuits (QEP : quadrature-encoder pulse). [43]

a) Génération du signal PWM avec EVA

Pour générer le signal PWM, un timer (GP Timer1) approprié est utilisé pour la désignation de la période du signal. La période du Timer1 est fixée par le registre T1PR. Un registre compare (CMPRx) est utilisé pour maintenir les valeurs des largeurs des impulsions.

La valeur du registre compare (CMPRx) est constamment comparée avec la valeur du compteur du timer. Quand les deux valeurs se croisent, une transition se produit sur la sortie associée. Quand un deuxième croisement est réalisé entre les deux valeurs, ou quand la fin de la période timer est atteinte, une autre transition se produit sur la sortie associée.

Ainsi, une impulsion est générée sur la sortie associée, dont la durée *ton* ou *toff* est liée et proportionnelle à la valeur contenue dans le registre compare. La répétition de ce processus avec l'utilisation de valeurs de modulation différentes à chaque période dans les registres compare (CMPRx) permet d'obtenir un signal MLI en sortie.

Un temps-mort (Dead-band) est souvent inséré entre l'ouverture d'un interrupteur et la fermeture de l'autre interrupteur du même bras du convertisseur de puissance (onduleur). Ce retard permet l'ouverture d'un interrupteur avant la fermeture de l'autre interrupteur, ainsi on évite des courts-circuits sur l'alimentation. Le temps de retard (Dead-band) requis est spécifié par les caractéristiques d'ouverture et fermeture des transistors de puissance MOSFET utilisés. Dans notre application le Dead-band est d'environ 600 ns et généré à l'aide du driver IR2111 et non pas par le générateur Dead-band intégré dans DSP.

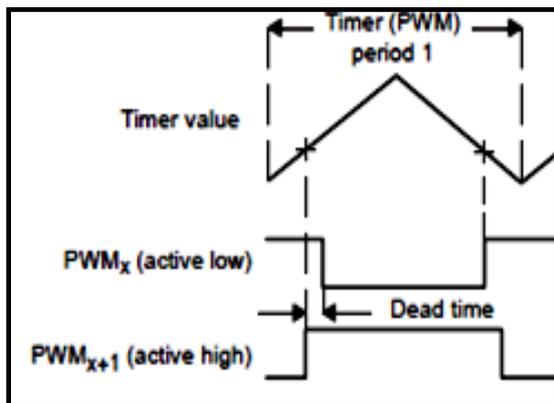


Figure 5.18 (a) Génération d'onde symétrique.

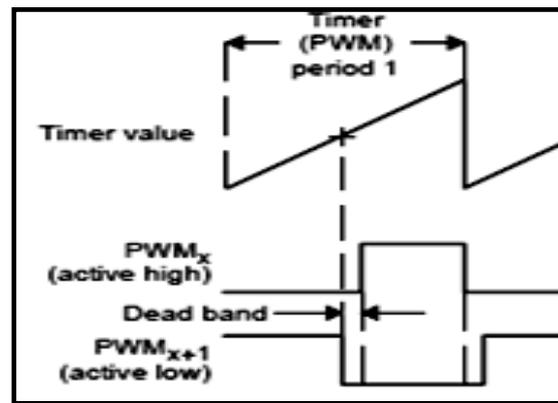


Figure 5.18 (b) Génération d'onde asymétrique.

b) Configuration des registres pour la génération d'une PWM avec EVA

Pour parvenir à générer un signal PWM, il faut, configurer en premier lieu les pins de sortie des signaux car la plupart des pins sont multiplexées et exercent une double fonction. Le processus de configuration des registres de l'Event Manager EVA pour la génération PWM comporte les étapes suivantes :

- Configurer le registre période T1PR.
- Configurer et charger le registre ACTRA.
- Configurer et charger le registre de contrôle COMCONA.
- Initialiser les registres CMPR1, CMPR2 et CMPR3 utilisés.
- Configurer et charger le registre de contrôle T1CON.
- Configurer et charger le registre DBTCONA, si le Dead-band est nécessaire.
- Recharger les registres CMPR1, CMPR2 et CMPR3 avec les nouvelles valeurs.

2.1.2 Le module ADC

Le module ADC du TMS 320F2812 est un convertisseur analogique numérique 12-bit. Il possède seize (16) canaux, configurables comme deux modules huit (8) canaux. Les deux modules 8-canaux peuvent être cascades pour former un seul module 16-canaux. [42]

Sur chaque pin ADC, une fois la conversion complétée, la valeur sur le canal ADCIN_x utilisé est sauvée dans le registre ADCRESULT approprié. La valeur numérique de la tension analogique est tirée par :

- Valeur numérique = 0 Si l'entrée ≤ 0
- Valeur numérique = $4095 \times \frac{\text{tension analogique} - \text{ADCL0}}{3}$ Si $0 < \text{l'entrée} < 3V$
- Valeur numérique = 4095 Si l'entrée ≥ 3V

2.2 Les points principaux de l'implantation logicielle

Afin de commander la vitesse du moteur asynchrone triphasé, on utilise les unités Full/ Compare du module EVA du DSP TMS 320F2812 pour produire les signaux PWMs appropriés. Les largeurs modulées des signaux PWM sont calculées en utilisant un algorithme de contrôle basé sur la technique de modulation vectorielle SVM.

Pour l'implantation de cet algorithme basé sur la technique de modulation SVM, on représente dans l'algorithme le vecteur tension référence tournant \vec{V}_{ref} par sa fréquence de rotation et son amplitude.

Les caractéristiques principales de cette implantation logicielle sont; l'intégration de la fréquence de rotation du vecteur tournant \vec{V}_{ref} afin d'obtenir sa phase θ , le calcul des composantes $V_{\alpha ref}$ et $V_{\beta ref}$ à partir de l'amplitude de \vec{V}_{ref} et de la phase θ , détermination du secteur actuel du vecteur \vec{V}_{ref} , calcul des quantités $T1, T2$ et $T0$ suivant les formules propres à chaque secteur, et la détermination de la séquence de commutation appropriée pour chaque secteur. Les trois (3) registres full-compare (CMPR1, CMPR2 et CMPR3) sont mis à jours à chaque appel de la routine d'interruption PWM et sont rechargés avec les nouvelles valeurs calculées t_{aon}, t_{bon} et t_{con} .

❖ Les principales étapes impliquées dans cette implantation sont:

- 1) Lecture de la fréquence de référence f_{ref} et du sens de rotation.
- 2) Intégration de la fréquence pour obtenir la phase θ du vecteur référence.
- 3) Calculs de $\sin(\theta), \cos(\theta)$ et des composantes $V_{\alpha ref}$ et $V_{\beta ref}$.
- 4) Détermination du secteur S dans lequel se trouve le vecteur \vec{V}_{ref} .
- 5) Utiliser le secteur S pour le calcul des quantités $T1, T2$ et $T0$.
- 6) Utiliser le secteur S pour déterminer la séquence de commutation convenable.
- 7) Charger les registres compare CMPR1, CMPR2, CMPR3 avec les valeurs appropriées

2.3 L'organisation logicielle

Dans la partie logicielle, le GP Timer1 est utilisé comme base de temps pour la génération des sorties PWM par les unités full compare avec une fréquence de 20 KHz. Le GP Timer1 est utilisé aussi comme base de temps pour appeler à chaque période de 50 us le sous programme d'interruption. Le canal ADCINA0 est utilisé pour introduire la fréquence de référence. La pin B0 du port B est utilisée pour introduire le sens de rotation. Le programme général de la commande de vitesse du moteur asynchrone est constitué de trois parties :

a) Initialisation du programme

Dans cette partie d'initialisation, on configure le système de contrôle, on configure les pins GPIO pour la sélection de la fonction de chaque pin (la pin B0 est utilisée pour l'introduction du sens de rotation du moteur), puis on initialise la table vecteur du module PIE et des interruptions CPU, ensuite on configure et on initialise le module EVA utilisé pour la génération des signaux (PWM1...PWM6) par les unités full compare, et on configure et on initialise le module ADC utilisé pour l'introduction de la fréquence de référence à partir du canal ADCIN0, dans cette partie aussi on initialise la période (50 us) du GP Timer1 et on valide l'interruption T1PINT qui sera utilisée comme base de temps pour le déclenchement du programme d'interruption.

b) Programme principal

Le programme principal est bouclé sur lui-même, ce qui permet d'introduire et de renouveler la fréquence de référence et le sens de rotation le moment désiré, et c'est dans cette partie que s'effectue presque la plupart des opérations et calculs :

- Lecture de la fréquence de référence f_{ref} et du sens de rotation.
- Calcul des composantes V_{aref} et $V_{\beta ref}$ du vecteur \vec{V}_{ref} .
- Détermination du secteur S du vecteur \vec{V}_{ref} .
- Détermination de la séquence de commutation.
- Calcul des quantités $T1, T2$ et $T0$.

c) Programme d'interruption

La routine d'interruption traite une petite partie du programme global, mais qui est très sensible. La routine d'interruption est appelée régulièrement à chaque nouvelle période PWM.

La période de l'interruption (ou d'échantillonnage) est de 50 us (20 KHz), elle est établie par la mise du T1PER à la valeur 1875 (PWMPRD=1875). Ce timer est placé en mode up-down count et génère une interruption périodique à chaque événement Under_flow du compteur Timer1. La période du Timer1 est calculée comme suit :

$$\text{Periode_PWM} = 2 * \text{T1PER} * \text{HSPCLK} * \text{TPS0_2} \quad (5.4)$$

Dans notre application on a pris $T1PER = 1875$ et $HSPCLK = 75$ MHz

Ce qui donne : $Period = 50 \text{ us}$ soit (20 KHz)

Sachant que: $HSPCLK = SYSCLKOUT / (2 * HISPCP) = 150 / (2 * 1) = 75$ MHz

- **SYSCLKOUT** : L'horloge système de fréquence maximale 150 MHz, cadencant la CPU et les périphériques.
- **HSPCLK (High speed clock)**: Dérivée de l'horloge système, alimente uniquement les périphériques, et ajustable grâce au pré-scalaire HISPCP.
- **HISPCP (High-speed Clock Pre-scaler)** : Diviseur d'horloge sur trois bits, il peut prendre sept valeurs possibles {1, 2, 3, 4, 5, 6,7}.
- Les bits TPS0-2 donnent la valeur du diviseur de fréquence pour le TIMER Counter, {1, 2, 4, 8, 16, 32, 64, 128}. Ces bits sont présents dans le registre T1CON.

A chaque période de la base de temps (50 us) une interruption est générée par le GP Timer1, ce qui permet de déclencher l'exécution de la routine de service interruption afin de calculer la nouvelle valeur pour la phase θ du vecteur \vec{V}_{ref} par l'intégration de la vitesse angulaire. Cette routine effectuée aussi à chaque nouvelle période de 50 us (20 KHz) le chargement des registres compare (CMPR1, CMPR2 et CMPR3) par les valeurs des quantités t_{aon} , t_{bon} et t_{con} déjà calculées dans le programme principal.

2.4 Organigramme global

L'organigramme sur la figure 5.19 illustre l'organisation de l'implantation logicielle, en présentant les liaisons entre les trois parties (Initialisation, Programme principal, Programme d'interruption). La figure 5.20 illustre l'organigramme global avec plus de détails.

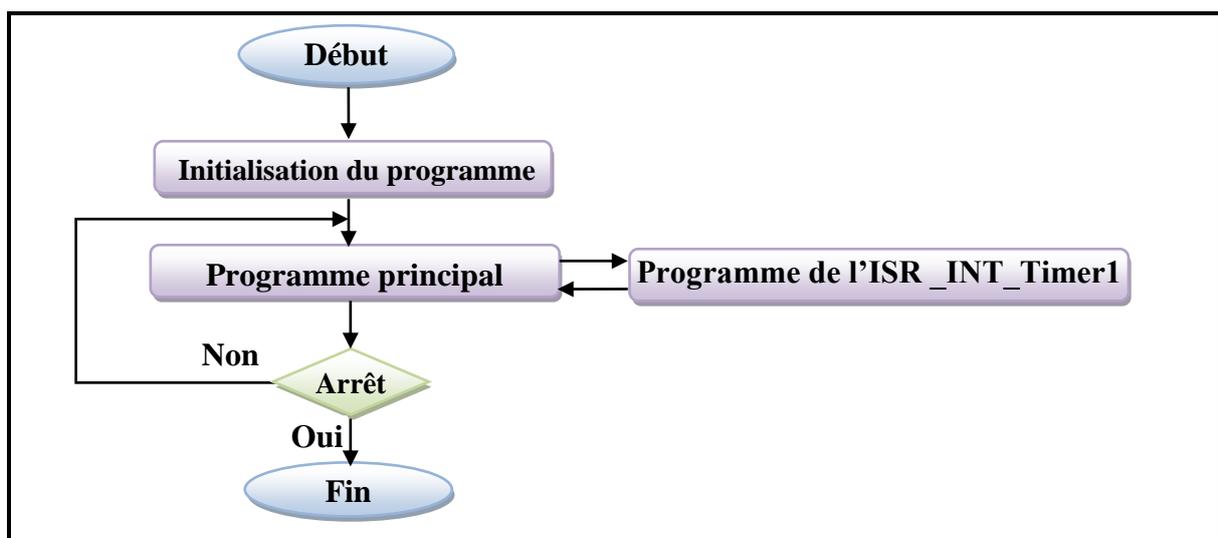


Figure 5.19 Organigramme global de l'implantation logicielle.

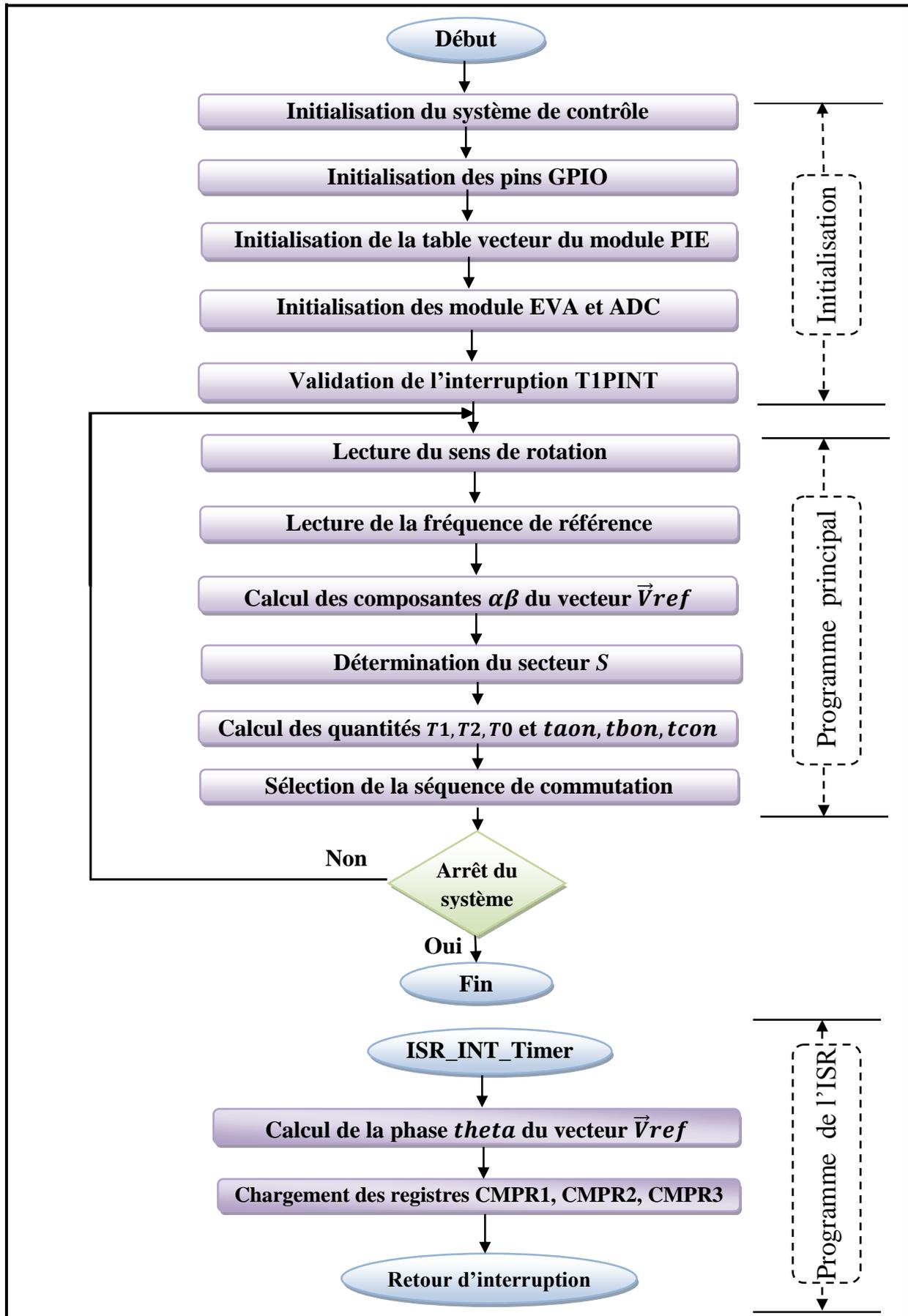


Figure 5.20 Organigramme global de l'implantation logicielle avec plus de détails.

3. Résultats pratiques

Afin de mettre en évidence le bon fonctionnement du système réalisé, on a effectué des essais expérimentaux sur le moteur asynchrone avec différentes références de fréquence et dans les deux sens de rotation. Pour visualiser et prélever les grandeurs électriques on a utilisé l'oscilloscope numérique (GDS-840C) connecté au PC via l'USB.

3.1 Les signaux de commande

La figure 5.21 montre la forme du signal de commande issu de la pin PWM1 du DSP.

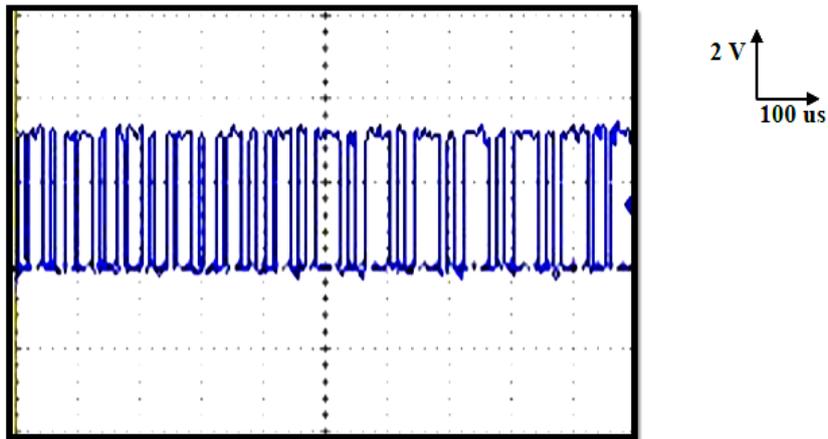


Figure 5.21 La forme du signal de commande PWM1.

Le signal prend l'allure d'une séquence d'impulsions d'amplitude de 3.3V et dont les largeurs prennent des valeurs distinctes sur chaque période de 50 us (20kHz).

3.2 Les signaux à la sortie du driver

Pour mettre en vérité la notion du temps mort (Dead-band), on a visualisé les allures des deux signaux de sortie LO et HO du driver. On constate effectivement, la présence d'un temps mort d'environ 600 ns entre les deux signaux de sortie LO (rouge) et HO (bleu).

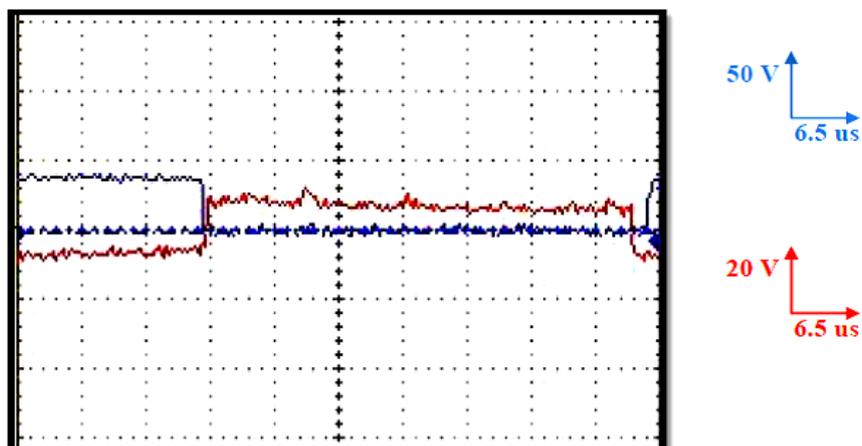


Figure 5.22 Illustration du temps mort entre les signaux de sortie LO et HO du driver IR2111.

3.3 Les tensions des phases

L'allure de la tension phase est présentée sur la figure suivante :

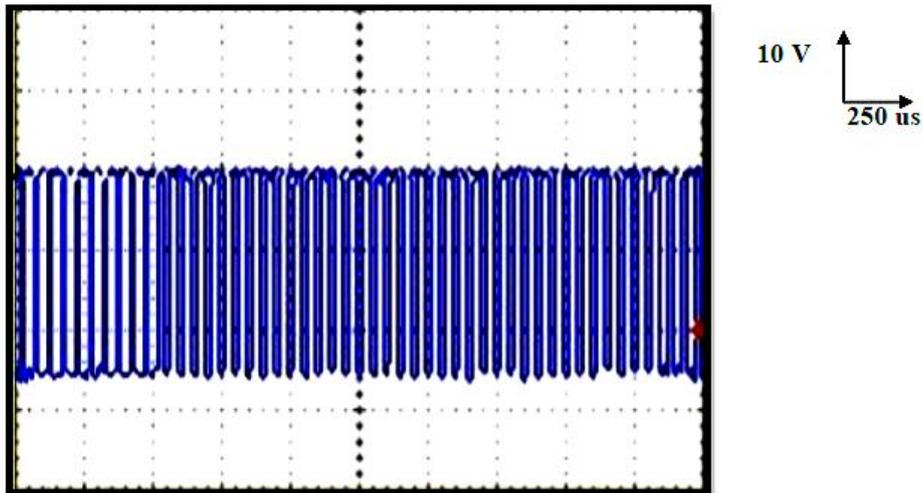


Figure 5.23 L'allure de la tension de phase (a) de l'onduleur.

On remarque sur la figure que la tension de phase suit la même allure que le signal de commande, c'est-à-dire une séquence de signaux rectangulaires avec des largeurs différentes et d'une période (50 us) fixe imposée par le signal de commande, et d'amplitude de 25 V imposée par la tension de bus continu alimentant l'onduleur triphasé.

3.4 Les tensions composées

Dans cette étape de tests expérimentaux, on a visualisé sur l'oscilloscope les tensions entre deux bras de l'onduleur pour deux références de fréquence 20 et 40 Hz.

❖ Fréquence de référence 20 Hz

La figure suivante illustre l'allure de la tension composée (entre deux phases) pour une fréquence de référence 20 Hz.

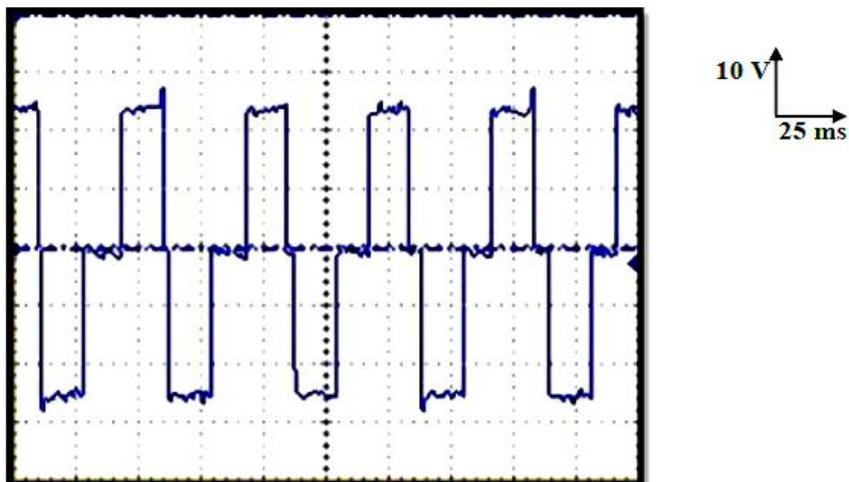


Figure 5.24 L'allure de la tension composée à la fréquence de référence 20 Hz.

❖ Fréquence de référence 40 Hz

La figure 5.25 illustre l'allure de la tension composée (entre deux phases) pour une fréquence de référence 40 Hz.

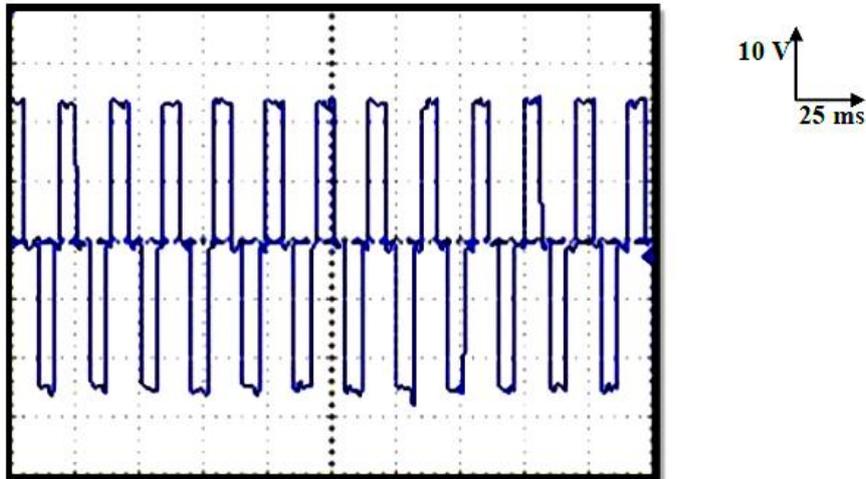


Figure 5.25 L'allure de la tension composée à la fréquence de référence 40 Hz.

Sur les deux figures précédentes on constate que les allures des tensions composées prennent la forme de signaux périodiques alternatifs dont les amplitudes basculent entre + 25 V, 0 V et -25 V (trois niveaux), et leurs fréquences sont fondamentalement établies et fixées par les fréquences de références imposées.

Sur la figure suivante est présentée la tension composée sur un court intervalle de temps où est augmenté le nombre d'échantillons prélevés à chaque seconde par l'oscilloscope.

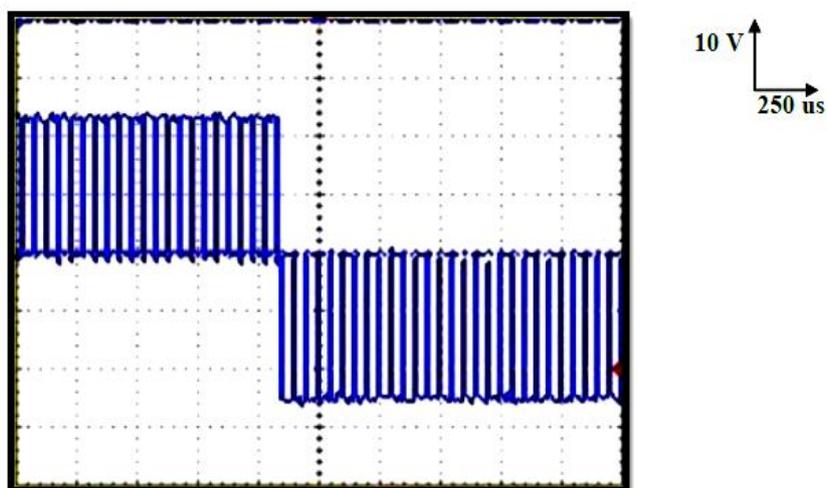


Figure 5.26 La tension composée présentée par l'oscilloscope sur un court intervalle du temps.

3.5 Les courants des phases

Dans cette partie des essais pratiques, on a utilisé l'oscilloscope pour la visualisation et le prélèvement des allures des courants dans les phases statoriques du moteur pour les fréquences de référence 5, 8 et 10 Hz.

Afin de visualiser ces courants des phases statoriques du moteur, on a câblé une résistance shunt de 300 mΩ avec chaque phase statorique, ensuite on a prélevé la forme de la tension entre les bornes de la résistance shunt, qui traduit exactement l'allure du courant dans la phase du moteur.

❖ Fréquence de référence 5 Hz

Sur la figure suivante, est présentée l'allure du courant dans la phase du stator pour la fréquence de référence 5 Hz.

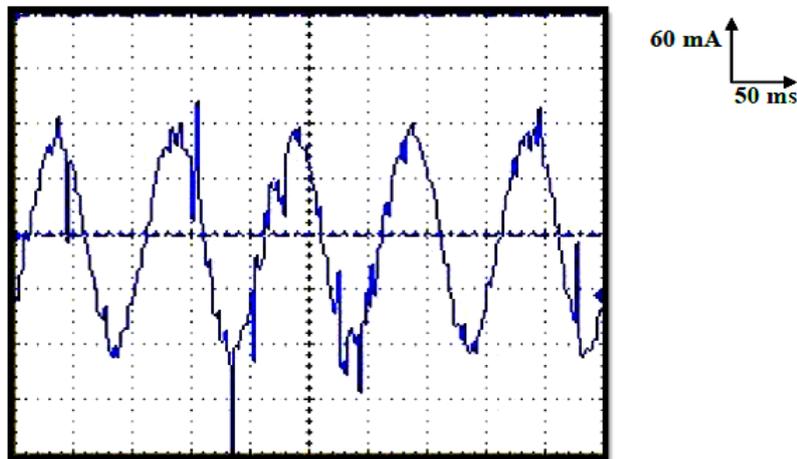


Figure 5.27 Allure du courant de phase à la fréquence de référence 5 Hz.

❖ Fréquence de référence 8 Hz

Sur la figure suivante, est montrée l'allure du courant dans la phase du stator pour la fréquence de référence 8 Hz.

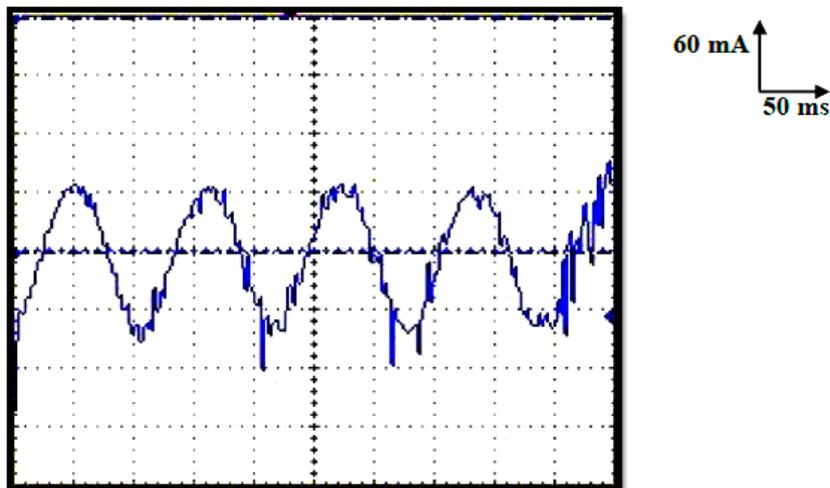


Figure 5.28 Allure du courant de phase à la fréquence de référence 8 Hz.

❖ Fréquence de référence 10 Hz

La figure suivante, illustre l'allure du courant dans la phase statorique du moteur pour la fréquence de référence 10 Hz.

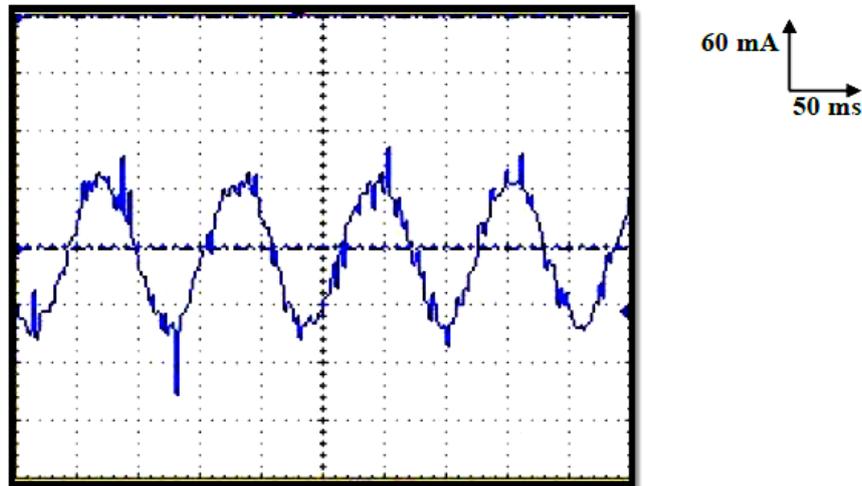


Figure 5.29 Allure du courant de phase à la fréquence de référence 10 Hz.

D'après les figures précédentes, illustrant les allures des courants dans les phases statoriques du moteur, on constate que les formes de ces courants se rapprochent fortement de la forme sinusoïdale (l'effet des harmoniques de courant est très faible), et que la fréquence du courant de phase du stator, est absolument la même que la fréquence de référence imposée.

Ainsi on déduit, l'intérêt et l'efficacité de l'algorithme de commande utilisé, qui est basé sur la technique de modulation vectorielle (SVM).

Conclusion

En ce chapitre, on a exposé les aspects matériel et logiciel du système adopté comme solution pour commander en boucle ouverte la vitesse d'un moteur asynchrone triphasé. Ceci en commençant, par la description de chaque bloc fonctionnel constituant la carte électronique réalisée qui est composée de l'onduleur triphasé et de l'interface d'isolation galvanique, puis on est passé par la présentation de l'implantation logicielle sur le DSP TMS 320F2812 de la stratégie de commande basée sur l'algorithme de la modulation MLI vectorielle (SVPWM), et par la suite on a exposé les différents tests expérimentaux effectués et résultats pratiques obtenus. A l'issue de ces tests et résultats, on déduit le bon fonctionnement du système réalisé et on peut le qualifier d'acceptable et de satisfaisant comme début dans le domaine de traction électrique. Néanmoins, il est toujours possible d'améliorer les performances du système réalisé, soit par l'optimisation de l'aspect logiciel en étendant le contrôle scalaire de la vitesse en boucle ouverte au contrôle vectoriel du couple et de la position en boucle fermée, soit par l'amélioration de l'aspect matériel par l'exploitation des sources d'énergies renouvelables dans le ravitaillement en énergie du système en adjoignant un système d'alimentation à base de panneaux solaires, plus l'intégration d'un système de management de l'énergie et un système de freinage par récupération pour une exploitation plus efficace de l'énergie.

Conclusion générale

Au cours de ce travail, nous avons réalisé un variateur de vitesse à base du contrôleur DSP TMS 320F2812 de Texas Instruments, en vue d'application dans un système de traction pour véhicule électrique. Pour cela nous avons commencé par l'étude des caractéristiques des processeurs DSP et en particulier le TMS 320F2812 qui fait l'objet de notre travail. Afin de mieux exploiter les ressources et les fonctionnalités du DSP, dans la conception de la carte de commande, nous avons étudié son architecture interne et sa programmation en langage C/C++ via l'environnement CCS IDE. Dans la partie réalisation électronique, nous avons conçu un onduleur triphasé et une interface d'isolation galvanique entre la carte eZdsp TMS 320F2812 et la partie puissance. En parallèle avec ce travail, nous avons effectué une simulation sous MATLAB/Simulink de la commande de la vitesse d'un moteur asynchrone en utilisant le principe de la technique de modulation MLI vectorielle (SVPWM), afin d'apercevoir la réponse et le comportement du moteur asynchrone vis-à-vis de cette technique SVPWM.

Dans la partie rédaction, on a organisé le mémoire en cinq chapitres. Le premier chapitre a été consacré à la présentation du transport électrique et les différentes technologies des véhicules électriques. Le deuxième chapitre a été dédié à la description du DSP TMS 320F2812. Le chapitre suivant expose le principe de fonctionnement du moteur asynchrone et sa modélisation, et dans ce même chapitre on a donné un aperçu sur la technique de modulation MLI vectorielle. Puis vient le quatrième chapitre, où nous avons fait et présenté une simulation sur le logiciel MATLAB de la commande en vitesse d'un moteur asynchrone. Dans le dernier chapitre nous avons décrit les blocs fonctionnels de la carte électronique réalisée, il a été exposé l'aspect logiciel et l'algorithme global implanté sur le DSP, puis on a terminé par la présentation des résultats pratiques obtenus qui sont commentés et interprétés.

L'élaboration de ce travail, nous a permis d'améliorer nos acquis dans le domaine de conception de commandes numériques à base de microcontrôleurs, et de s'introduire dans le monde des processeurs DSP et de s'initier à leur programmation. Ce travail nous a permis d'apercevoir le principe et les notions de base sur la technique de modulation vectorielle et de voir son utilité et efficacité dans la commande des systèmes triphasés. Le long de ce projet, on a eu l'occasion de travailler avec pas mal d'outils logiciels, l'environnement (CCS IDE) dans la programmation des DSP, l'outil MATLAB/Simulink dans la simulation des systèmes de contrôle et de puissance, le logiciel Proteus (ISIS/ARES) utilisé pour la simulation de circuits électroniques numériques et analogiques et pour la conception de circuits imprimés, ainsi que les logiciels PSpice et PSIM utilisés dans la simulation des circuits électroniques et électroniques de puissance.

Comme début dans le domaine de conception de systèmes de traction électrique à base de DSP, ce travail peut être qualifié d'acceptable et de satisfaisant, et nous espérons qu'il sera suivi d'extensions et d'améliorations. Les extensions et les améliorations peuvent toucher, soit l'aspect logiciel en étendant le contrôle scalaire de la vitesse en boucle ouverte au contrôle vectoriel du couple et de la position en boucle fermée, soit l'aspect matériel par l'exploitation des sources renouvelables dans le ravitaillement en énergie du système en adjoignant un bloc d'alimentation à base de panneaux solaires et/ou éoliennes, plus l'intégration d'un système de management de l'énergie et un système de freinage par récupération pour une exploitation plus efficace de l'énergie. Ou bien par l'intégration de tout autre mécanisme nécessaire afin de parvenir à un système de traction exploitable dans un véhicule électrique. Ce dernier étant incontournable dans le futur transport, et la nécessité à ce type de véhicule devient urgente afin de minimiser les dommages causés par l'humanité et subis par l'environnement innocent qui est indispensable pour la survie des générations futures.

Bibliographie

Bibliographie

- [1] Académie Caen. « *Le moteur asynchrone triphasé* ». Cours, science de l'ingénieur en S. (Document consulté en octobre 2010).
Adresse URL : <http://www.discip.ac-caen.fr/sti/stibacs/pdf/moteur-as.pdf>
- [2] P. ABATI. « *Le Driver de MOSFET et d' IGBT IR2113* ». Cours. Septembre 2001.
Adresse URL : <http://stielec.ac-aix-marseille.fr/cours/abati/ir2113.htm>
- [3] ALTER-MOTIVE. « *Véhicules hybrides et électriques* ». Les cahiers des carburants et motorisations alternatifs. février 2011.
Adresse URL : http://www.ddrhonealpesraee.org/dump/Cahier4O_59.pdf
- [4] ARVAL. « *Tout savoir sur les véhicules électriques* ». Brochure. Arval.2010.
Adresse URL : http://www.arval.be/docs/Arval_Brochure_FR_finaal.pdf
- [5] Avago Technologies.«*HCPL-2200, HCPL-2219 Low Input Current Logic Gate Optocouplers* ». Datasheet, AV01-0557EN. July 2007. Site web: www.avagotech.com
- [6] L. BAGHLI. « *Modélisation et commande de la machine asynchrone* ». Notes de cours. IUFM de Lorraine-UHP. 2005. Adresse URL :
http://www.clubeea.org/documents/mediatheque/Cours_modele_cmde_MAS.pdf
- [7] D. BENDIB. « *Etude et réalisation d'une commande MLI on-line sur circuit FPGA* ». Mémoire de magister. ENP, Alger. Juin 2009.
- [8] P. BRUNET. « *Introduction à la commande vectorielle des machines asynchrones* ». Cours. (Document consulté en décembre 2010).
Adresse URL : <http://www.iufmrese.cict.fr/catalogue/MAS/Mas.PDF>
- [9] S. BUSO et P. MATTAVELLI. «*Digital Control in Power Electronics*». First Edition by Morgan & Claypool. United States of America. Copyright © 2006.
- [10] R. CARTON et C. STRASSER. « *Les technologies hybrides* ». Rapport de TIPE. EPMI (Ecole supérieure d'Electricité, de Production et des Méthodes Industrielles) 2006.
Adresse URL : <http://homepages.laas.fr/echanthe/doc/TIPE techno hybrides.pdf>
- [11] D. CORNELL.« *Snubber Capacitors* ». Application Guide. (document consulté en juin 2011).
Adresse URL: <http://www.cde.com/catalogs/igbtAPPguide.pdf>
- [12] S. DEPOORTER et P. ASSIMON « *Les véhicules électriques en perspective Analyse coûts-avantages et demande potentielle*». Collection "Études et documents" du Service de l'Économie, de l'Évaluation et de l'Intégration du Développement Durable (SEEIDD) du Commissariat Général au Développement Durable (CGDD). Mai 2011. Adresse URL :
http://www.inrets.fr/fileadmin/recherche/transversal/pfi/PFI_VE/pdf/CGDD_VE_perspective.pdf
- [13] M. EHSANI, Y. GAO, S. E. GAY et A. EMADI. « *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles Fundamentals, Theory, and Design* ». CRC Press © 2005. Adresse URL:
<http://dualibra.com/wp-content/uploads/2011/06/Modern-Electric-Hybrid- Electric-and-Fuel-Cell-Vehicles.pdf>
- [14] S. ERWAN. « *Implementation of a Speed Field Oriented Control of 3-phase PMSM Motor using TMS320F240* ». Application Report, SPRA588. Texas Instruments Incorporated. September 1999.
Adresse URL: <http://www.ti.com/lit/an/spra588/spra588.pdf>
- [15] C. ESPANET , D. DEPERNET et F. DUBAS « *SEVIT : Stockage d'Énergie par un système de Volant d'Inertie optimisé pour des applications de Transport ferroviaire*VITFER : Volant d'Inertie pour le Transport FERroviaire». Fiche_projet_SEVIT_VITFER 2009.
Adresse URL : http://www.femto-t.fr/f/d/fiche_projet_SEVIT_VITFER.pdf

- [16] Forward Concepts. « *Forward concepts' wireless/dsp market bulletin* », Bulletin. February, 2008.
Adresse URL : http://www.fwdconcepts.com/DSPBulletin_2408.pdf
- [17] FranceCOL Technology. « *Gamme commerciale logiciels* » brochure publicitaire. (Document consulté en 2012). Adresse URL : <http://www.francecol.com/img/upload/pdf/DC50110fr.pdf>
- [18] A. FUHS. « *Hybrid vehicles and the future of personal transportation* ». CRC Press Taylor & Francis Group. Copyright © 2009.
- [19] C. GOMBERT « *Simulation temps-réel des dispositifs d'Electronique de Puissance dédiés aux Réseaux d'Energie Electrique* ». Thèse de Doctorat. Laboratoire d'Electrotechnique de Grenoble dans le cadre de l'Ecole Doctorale 'Electronique, Electrotechnique, Automatique et Traitement du signal'. le 29 Septembre 2005.
Adresse URL : http://hal.inria.fr/docs/00/17/10/82/PDF/These_Gombert_2005.pdf
- [20] H. GUALOUS, R. GALLAY ET A. BERTHON. « *Utilisation des supercondensateurs pour le stockage de l'énergie embarquée: applications transport* ». Article. CH-1728 Rossens, Suisse. (Document consulté en janvier 2012).
Adresse URL : <http://www.garmanage.com/atelier/root/public/Contacting/biblio.cache/REE-final.pdf>
- [21] International Rectifier. « *Half-Bridge Driver* ». DataSheet I R2111(S) & (PbF). 4/12/2004.
Adresse URL : <http://www.irf.com/product-info/datasheets/data/ir2111.pdf>
- [22] J. LARMINIE ET J. LOWRY. « *Electric Vehicle Technology Explained* ». John Wiley & Sons Ltd, England. Copyright © 2003.
Adresse URL : [www.ufsm.br/righi/projetos/Veículos elétricos baterias.pdf](http://www.ufsm.br/righi/projetos/Veículos_elétricos_baterias.pdf)
- [23] P. LE BRUN. « *Variation de vitesse des machines asynchrones* ». Cours. (Document consulté en janvier 2011). Adresse URL : http://www.larmand.fr/fichiers/Ancien_site/enseigne/ressources/techno/bourse%20cours/COURS/machine%20asynchrone%20variation%20vitesse.pdf
- [24] P. LE BRUN. « *Machine asynchrone : Technologie, choix et alimentation des machines asynchrones* ». Cours. (Document consulté en janvier 2011).
Adresse URL : <http://educyclopedia.karadimov.info/library/mas.pdf>
- [25] H. LE-HUY. « *Introduction à MATLAB et Simulink* ». Cours. Département de génie électrique et de génie informatique Université Laval Québec, CANADA. 1998.
Adresse URL : <http://w3.gel.ulaval.ca/~lehuy/intromatlab/intromat.pdf>
- [26] O. LOUISNARD. « *Initiation à MATLAB* ». Rapport d'activité. septembre 2005.
Adresse URL : <http://www.mines-albi.fr/telechargements/rapport-d-activite/2010-2011/rapport-d-activite-2010/downloadFile/attachedFile/RA2010-2011.pdf?nocache=1309262292.78>
- [27] M. MOKHTARI « *MATLAB 5.2&5.3 et SIMULINK 2&3 pour étudiants et ingénieurs* ». Springer. 2000.
- [28] R. OSHANA. « *DSP Software Development Technique for Embedded and Real-Time Systems* ». Elsevier Inc. 2006.
- [29] S.Rafa. H.Zeroug. L.Hocine. K.Boudjit. « *Simulation sur Matlab/Simulink et implémentation sur DSP/FPGA de la commande vectorielle de la machine synchrone à aimants permanents (PMSM) alimenté onduleur de tension à Modulation vectorielle (SVM)* » Laboratoire des systèmes électriques industriels. Faculté d'Electronique et d'Informatique. USTHB, Alger, Algérie. (Document consulté en mars 2011). Adresse URL : <http://www.docstoc.com/docs/22674283/Simulation-sur-MatlabSimulink-et-impl%C3%AF%C2%BF%C2%BDmentation-sur-DSPFPG-A-de>

- [30] C. RICAUD et P. VOLLET. « *Le système de raccordement sur la borne de recharge, élément clé pour les véhicules Electriques* ». Schneider Electric. 2010.
Adresse URL : http://www.inrets.fr/fileadmin/recherche/transversal/pfi/PFI_VE/pdf/Vehicules-electriques-et-infrastructures-de-recharge.pdf
- [31] R. SEVERNS. « *Design of snubbers for power circuits* ». Document technique. (Document consulté en juin 2011). Adresse URL: <http://www.cde.com/tech/design.pdf>
- [32] Sheldon Instruments. « *Easy Real-Time DSP Programming with LabVIEW and Visual Basic* ». Brochure publicitaire. Copyright 2007. Adresse URL : <http://www.sheldoninst.com/downloads/easyrealtimedspprogrammingwithlabviewandvisual.pdf>
- [33] Spectrum Digital. « *Operation of the eZdsp F2812* ». Technical reference. (Document consulté en mars 2011). Adresse URL: http://www.roinos.com/board/ezdsp/2812_ezdsp_r.pdf
- [34] N. TADRIST. « *Implémentation par DSP d'une commande en vitesse d'une machine à courant continu sans collecteur (BDCM)* ». Mémoire de magister. Systèmes électro- énergétiques, faculté d'électronique et informatique. USTHB. 2009.
- [35] Texas Instruments, European Customer Training Center, and University of Applied Sciences Zwickau (FH). « *TMS320F2812 digital signal processor* ». Implementation Tutorial. (Document consulté en juin 2011). Adresse URL : <http://cegt201.bradley.edu/projects/proj2011/pkohonen/Resources/28857722-F2812-DSP-Full-Tutorial.pdf>
- [36] Texas Instruments. « *C28x DSP Design Workshop* ». Student Guide, Revision 5.1. May 2004.
Adresse URL: <http://www.ebookpp.com/tm/tms320c28x-pdf.html>
- [37] Texas Instruments. « *System Solutions for Hybrid and Electric Vehicles* ». Technical Support, Literature Number: szza058. 2010. Adresse URL: www.ti.com/hev
- [38] Texas Instruments. « *TMS320C28x CPU and Instruction Set* ». Reference Guide. Literature Number: SPRU430E. August 2001 – Revised January 2009.
Adresse URL: <http://www.ti.com/lit/ug/spru430e/spru430e.pdf>
- [39] Texas Instruments. « *TMS320F2812* » Datasheet, Literature Number: SPRS174R. April 2001, Revised May 2010. Adresse URL: <http://cegt201.bradley.edu/projects/proj2011/pkohonen/Resources/tms320f2812%5B1%5D.pdf>
- [40] Texas Instruments. « *C28x Digital Power Supply Workshop* ». Workshop Guide and Lab Manual. Revision 1.1 May 2008.
Adresse URL : http://focus.ti.com/en/download/mcu/C28x_DPS_Workshop.pdf
- [41] Texas Instruments. « *C2000 Real-Time Microcontrollers* ». Technical Support SPRB176L Copyright 2011. Adresse URL : <http://www.ti.com/lit/sg/sprb176l/sprb176l.pdf>
- [42] Texas Instrument. « *TMS320x281x DSP Analog-to-Digital Converter (ADC)* ». Reference Guide, Literature Number: SPRU060D. June 2002 – Revised July 2005.
Adresse URL: <http://www.ti.com/lit/ug/spru060d/spru060d.pdf>
- [43] Texas Instruments. « *TMS320x281x DSPEvent Manager (EV)* ». Reference Guide. Literature Number: SPRU065E. November 2004 – Revised June 2007.
Adresse URL: <http://www.ti.com/lit/ug/spru065e/spru065e.pdf>
- [44] P. C. Todd. « *Snubber Circuits: Theory, Design and Application* ». Paper. Texas Instruments Incorporated. 2001. Adresse URL : <http://space.dianyuan.com/blog/u/50/1170381240.pdf>

Annexes

1. Plaque signalétique du moteur

Tension nominale : 220/380 V Δ/Y 1.9/1.1 A

Puissance nominale : 0.37 kW

Vitesse nominale : 2765 tr/mn

Facteur de puissance : $\cos\varphi=0.79$

Fréquence nominale : 50 Hz.

2. Nomenclature des composants

❖ Résistances :

R1= 10 Ω	(0.25 Watt)	(N ^{bre} : 6)
R2= 4.7 Ω	(2 Watt)	(N ^{bre} : 3)
R3= 470 Ω	(0.5 Watt)	(N ^{bre} : 3)
R4= 470 Ω	(0.5 Watt)	(N ^{bre} : 3)
R5= 1 k Ω	(0.25 Watt)	(N ^{bre} : 1)
RP1= 1 k Ω	(2 Watt)	(N ^{bre} : 1)
RP2= 1.5 k Ω	(2 Watt)	(N ^{bre} : 1)
POT= 1 k Ω	(1 Watt)	(N ^{bre} : 1)

❖ Capacités :

C1= 100 uF	(25 V)	(N ^{bre} : 1)
C2= 100 uF	(25 V)	(N ^{bre} : 1)
C3= 33 nF	(100 V)	(N ^{bre} : 3)
C4= 0.1 uF	(50 V)	(N ^{bre} : 3)
C5= 100 pF	(10 V)	(N ^{bre} : 3)

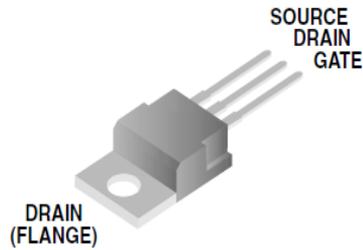
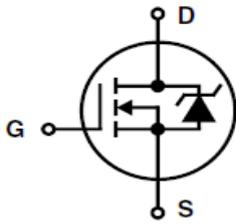
❖ Diodes :

D1: IN4004	(N ^{bre} : 3)
D2: IN4148	(N ^{bre} : 3)
DZ1 : BZX C12 (12 V)	(N ^{bre} : 1)
DZ2 : BZV 85C (16 V)	(N ^{bre} : 1)

❖ **Transistors :**

T1 : IRF 840

(N^{bre} : 6)



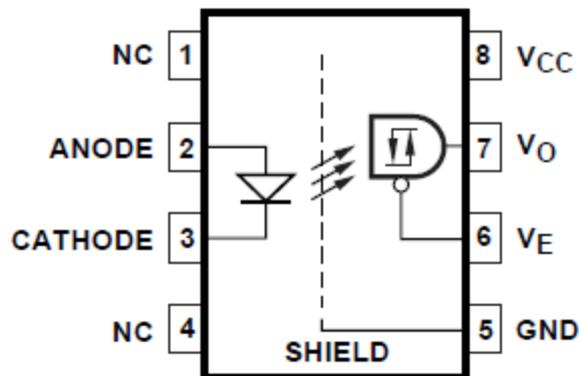
$I = 8A$
 $V = 500V$
 $R_{DS(ON)} = 0.850 \Omega$

❖ **Circuits intégrés :**

L'optocoupleur :

CI1 : HCPL 2200

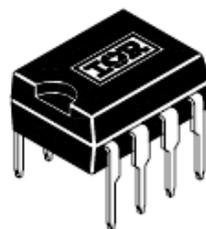
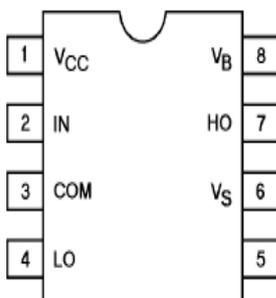
(N^{bre} : 3)



Le driver :

CI2 : IR 2111

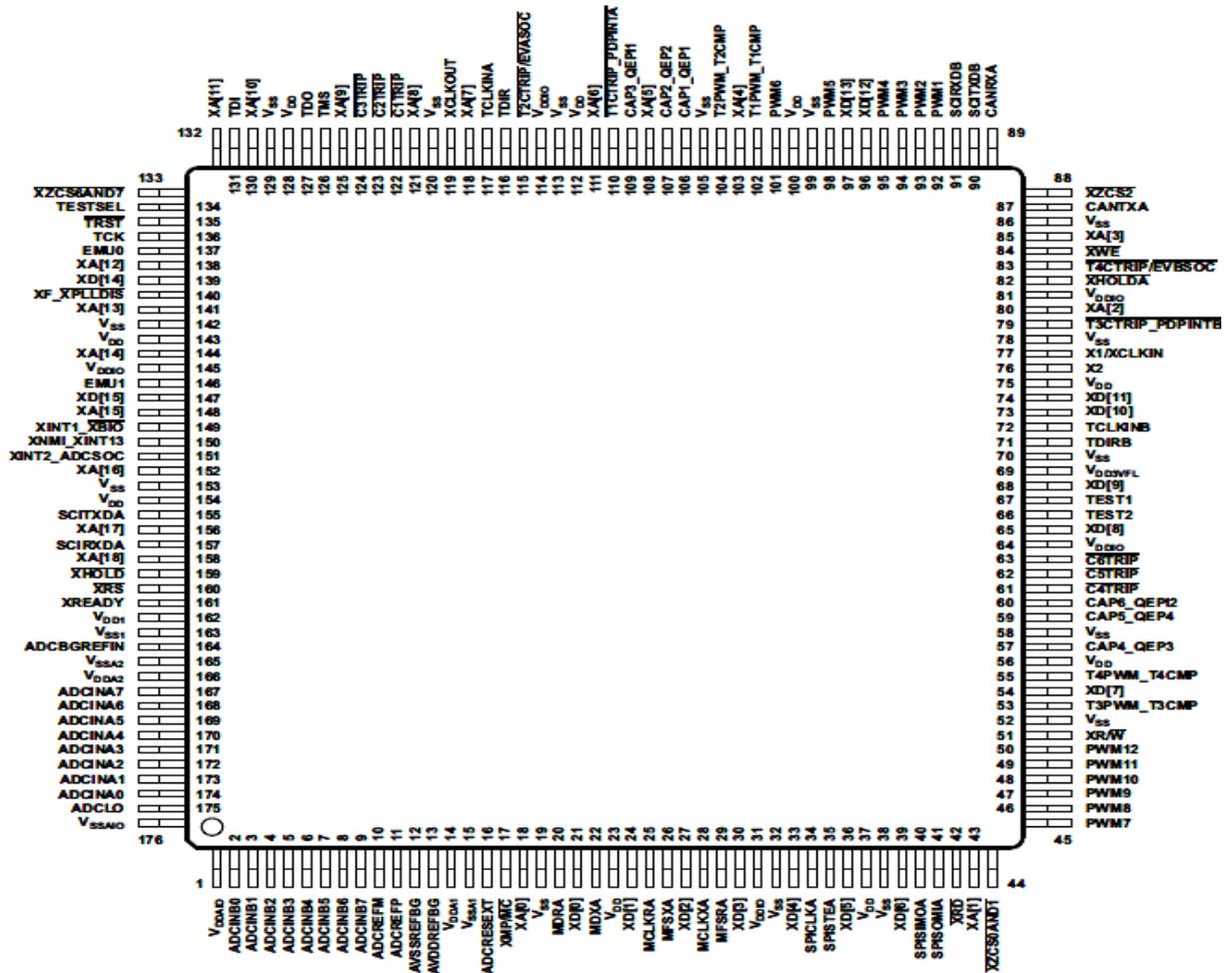
(N^{bre} : 3)



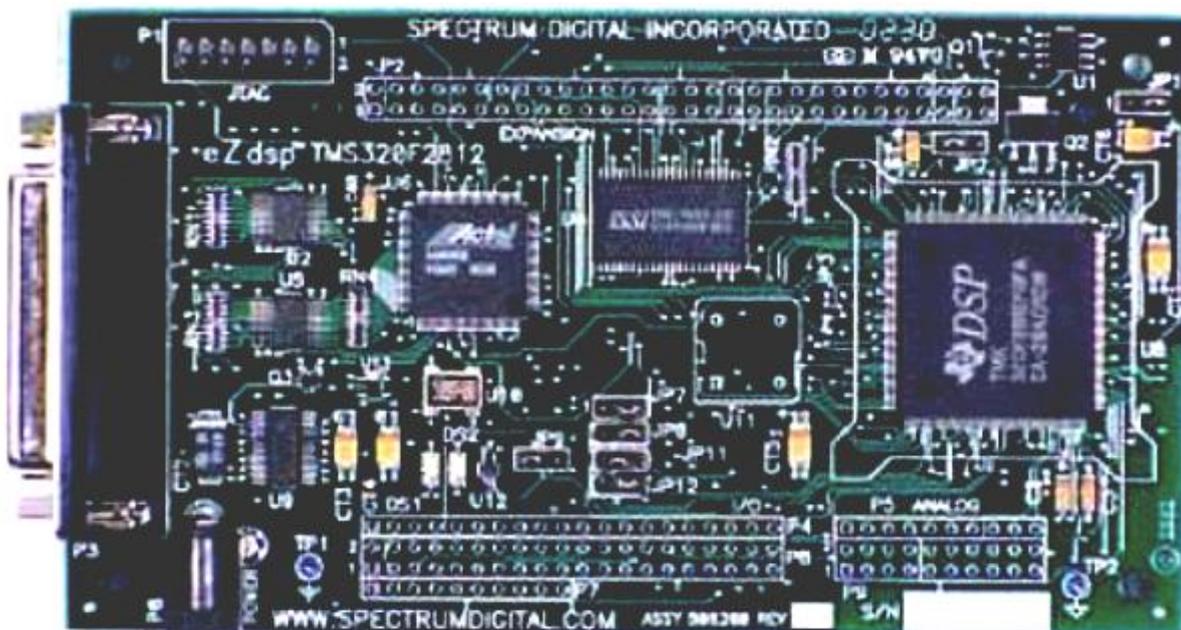
8-Lead PDIP

Voffset	600V max	200
IO+/-	mA / 420 mA	
VOUT	10 - 20V	
ton/off (typ.)	750 & 150 ns	
Deadtime (typ.)	650 ns	

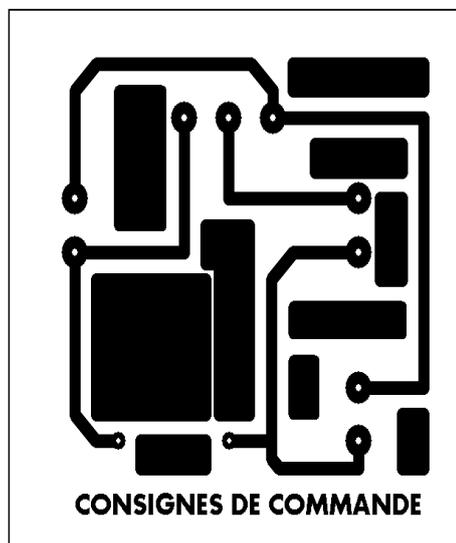
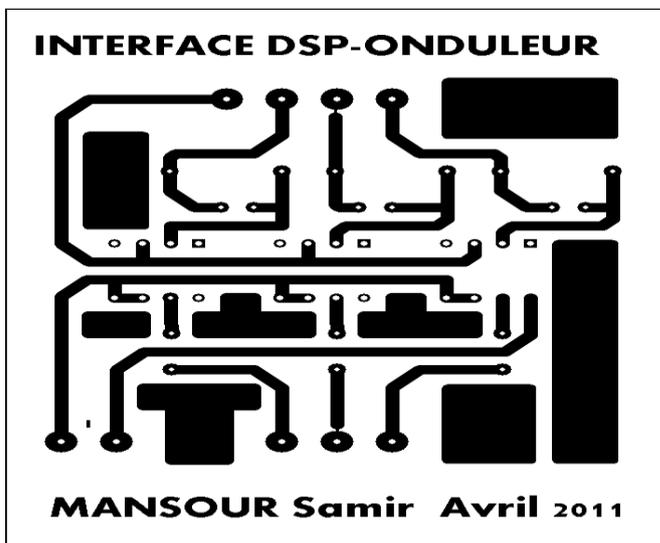
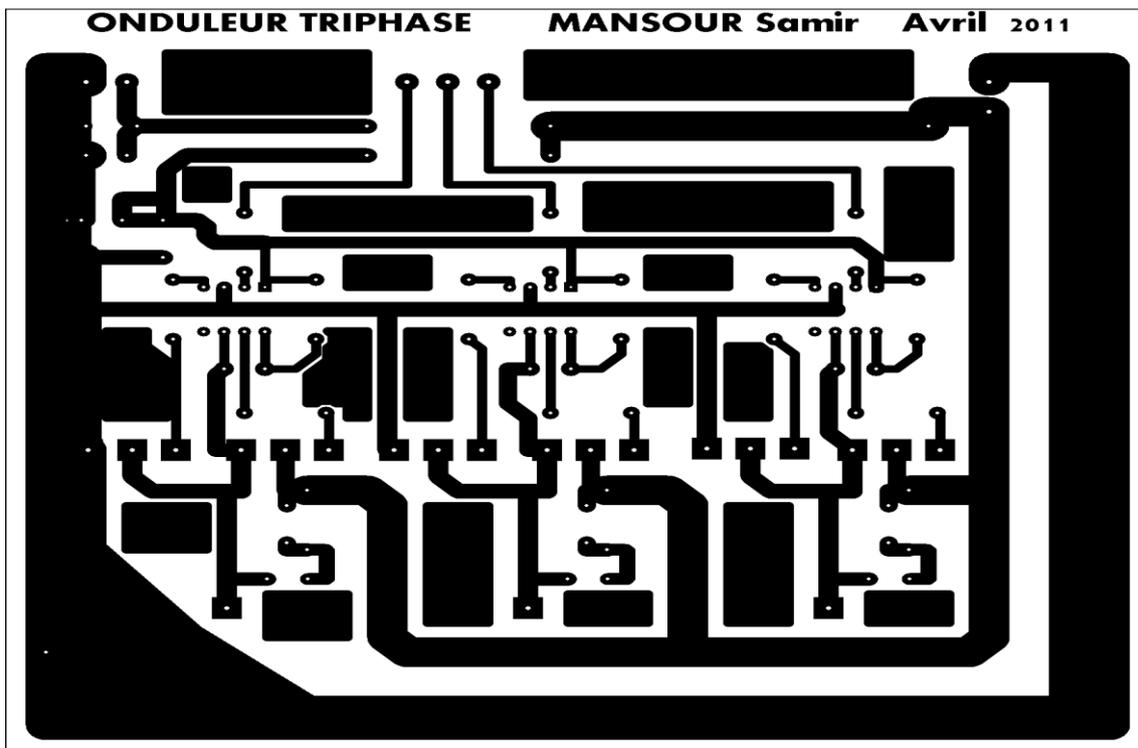
3. Le boîtier du DSP TMS 320F2812



4. La carte eZdsp TMS 320F2812



1. Circuits imprimés



1. Programme

```

//*****
// Description : Générations de signaux PWM1_6 par les unités Full-Compare de l'EVA à base de l'algorithme de la modulation vectorielle
//((SVM) pour la commande en vitesse et en sens de rotation d'un moteur asynchrone.
// Etablissement : Ecole Nationale Polytechnique.
// Auteur : MANSOUR Samir.
// Date : Novembre 2011.
//*****
#include "DSP281x_Device.h"           // Inclure le fichier entête DSP281x _ Device.h
#include "DSP281x_Examples.h"        // Inclure le fichier entete DSP281x Examples.h
#include "math.h"                    // Inclure le fichier entête math.h
#define _2PI 6.2831852                // définir la constante 2*PI

// Déclaration des fonctions d'initialisation et de l'interruption
interrupt void eva_timer1_isr(void);
void init_eva();
void Init_Adc();

//Déclaration des fonctions du programme principal
void cal_Ud_et_Uq();
void cal_X_Y_Z();
void cal_A2_A1_A0();
void cal_secteur();
void cal_T1_T2_T0();
void cal-Ta_Tb_Tc();
void cal_swithing();

// Initialisation des variables globales
unsigned int direction=1;
float Pr_timer1=1875;
unsigned int freq_ref=20;           // freq_ref= {0,1,...,50} (Hz)
float Tech;                         // Tech=50us=0.00005
float theta=0, theta_ref;
float Uref=0.6, Ud, Uq;             // 0<Uref < sqrt(2)/2 (pour éviter la distorsion)
float X, Y, Z;
unsigned int A0, A1, A2, secteur ;
float T0, T1, T2, Ta, Tb, Tc;
unsigned int C1, C2, C3;
char erreur[1];
float tgtheta, tg;

void main(void)
{
// Initialisation du système de contrôle:
InitSysCtrl();
EALLOW;                            // Configurer l'horloge peripherique (HSPCLK)
SysCtrlRegs.HISPCP.all=0x01; // HSPCLK=SYSCLKOUT/2*1=150/ (2*1) =75MHz(AD clock)
EDIS;

// Initialiser GPIO:
EALLOW;                             // 0:GPIO ; 1: PWM
GpioMuxRegs.GPAMUX.all = 0x00FF;     // EVA PWM 1-6 pins .
GpioMuxRegs.GPBMUX.bit.CAP4Q1_GPIOB8=0; // PinB8 (GPIO input)
GpioMuxRegs.GPADIR.all = 0x00FF;    // Set IO pins to be input pins
GpioMuxRegs.GPBDIR.all = 0x0;       // 0: input ; 1:output
EDIS;

// Désactiver les interruption CPU
DINT;

// Initialiser les registres de contrôle PIE
InitPieCtrl();

//Désactiver les interruption CPU
// et clear tous les flags des interruption CPU:

IER = 0x0000;
IFR = 0x0000;

// Initialiser la table vecteur PIE (PIE vector table).
InitPieVectTable();

EALLOW;
PieVectTable.T1PINT = &eva_timer1_isr;
EDIS;

// Pour désactiver l'accès aux registres protégés

// Initialiser EVA et ADC:
init_eva();
initAdc();

PieCtrlRegs.PIEIER2.all = M_INT4; // Valider l'interruption 4 de PIE group 2 pour T1PINT

```

```

IER |= M_INT2;    // Valider l'interruption CPU INT2 de T1PINT:
EINT;             // Valider l' interruption globale
ERTM;            // Valider l'interruption realtime interrupt DBGCM

Tech= (Pr_timer1 * 2)/(75000000);           // Calcul de la periode d'échantillonnage Tech:
                                           // ( 75 Mhz valeur de l'horloge HSPCLK utilisé)

// theta_ref = _2PI* freq_ref* Tech;        // Calcul de theta_ref à partir de la consigne freq_ref
                                           // Tech la période d'échantillonnage qui est la même que la
                                           // période d'appel de l'interruption eva_timer1_isr

for(;;)
{
    // Choix de la direction de rotation
    {if(GpioDataRegs.GPBDAT.bit.GPIOB8==0) // Lire et tester l'état de la pinB8
    direction =1 ;
    else
    direction = 0;
    }

    // Lecture de la consigne freq_ref sur l'entrée ADCINA0
    freq_ref=(AdcRegs.ADCRESULT0>>4)* 0.0122100; // 0.0122100=freqmax/ADCmax= 50/4095

    // Calcul de theta_ref à partir de la consigne freq_ref
    theta_ref = _2PI* freq_ref* Tech; // Tech la période d'échantillonnage qui est la période
    // d'appel de l'interruption eva_timer1_isr

    // Appels les différents sous-programmes (fonctions)
    // appel de la fonction qui calcul Ud et Uq
    cal_Ud_et_Uq();

    // appel de la fonction qui cacul X,Y et Z
    cal_X_Y_Z();

    // appel de la fonction qui calcul A2,A1 et A0
    cal_A2_A1_A0();

    // appel de la fonction qui calcul le secteur
    cal_secteur();

    // appel de la fonction qui calcul T1, T2 et T0
    cal_T1_T2_T0();

    // appel de la fonction qui calcul Ta, Tb, et Tc
    cal_Ta_Tb_Tc();

    // appel de la fonction qui détermine les séquences de commutation
    cal_swithing();
}
}
interrupt void eva_timer1_isr(void)
{
    // Calculer la valeur actualisée de l'angle theta
    {{
    if (theta > _2PI)
    theta = theta - _2PI;}
    theta = theta + theta_ref;
    }

    // Charger les registre compare CMPRx avec les valeurs calculées
    {if (freq_ref>0)
    {EvaRegs.CMPR1 = C1 ;
    EvaRegs.CMPR2 = C2 ;
    EvaRegs.CMPR3 = C3 ;}
    else
    {EvaRegs.CMPR1 = 0;
    EvaRegs.CMPR2 = 0 ;
    EvaRegs.CMPR3 = 0 ;}
    }

    // Valider encore l'interruption de timer1
    EvaRegs.EVAIMRA.bit.T1PINT = 1;

    // Reset le bit flag du timer1
    // BIT7: Flag T1PINT
    EvaRegs.EVAIFRA.all = BIT7;

    // Acknowledge interrupt (accusé de réception) pour recevoir
    // plus d'interruption à partir de PIE group 2
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP2;
}

void init_eva()
{
    // Initialiser EVA Timer1 & Configurer PWM1-PWM6
    // period=Pr_timer1*2/75M=xxxus=50us
    // Initialiser Timer1 counter à 0
    EvaRegs.T1PR = Pr_timer1;
    EvaRegs.T1CNT = 0;
}

```

```

EvaRegs.GPTCONA.bit.T1CMPOE=1; // Valider timer compare
EvaRegs.GPTCONA.bit.T1PIN=1; // Sortie timer compare
EvaRegs.EVAIMRA.bit.T1PINT = 1; // Valider Timer1 period interrupt
EvaRegs.EVAIFRA.bit.T1PINT=1; // Clear le flag Timer1 period interrupt
EvaRegs.T1CON.all = 0x0842; // Configure GP Timer1
//0x8802 // bit15-14(00) : affectée par l'émulation.
// bit13(x):réservé
// bits12-11(01):sélection du mode continuous-up/down count.
// bits10-8(000): Prescaler = /1
// bit7(x) : réservé.
// bit6(0): ne pas valider l'opération compter de timer1.
// bits5-4(00): choisir la source d'horloge interne (HSPCLK)
// bits3-2(00): Recharger les registres compter si compter est0
// bit1(1): le registre Timer compare est validé.
// bit0(x): réservé.

// Initialiser les registres compare

EvaRegs.CMPR1 = 0;
EvaRegs.CMPR2 = 0;
EvaRegs.CMPR3 = 0;

EvaRegs.ACTRA.all = 0x0666; // Définir les polarités des sorties PWM à l'aide du
// Registre ACTRA (Compare Action Control Register A).
// bits15(0): SVRDIR Direction positive (CCW).
// bits14-12(000): D2D1D0 = 000 (bits de basic Space vector).
// bits11-10(01): Pin PWM6/CMP6 est active low;.
// bits9-8 (10): pin PWM5/CMP5 est active high.
// bits7-6 (01): pin PWM4/CMP4 est active low.
// bits5-4 (10): pin PWM3/CMP3 est active high.
// bits3-2 (01): pin PWM2/CMP2 est active low.
// bits1-0 (10): pin PWM1/CMP1 est active high.

// EvaRegs.DBTCONA.all = 0x05FC; // 1us (1/(150M/32))*5 // n'est pas nécessaire dans notre cas.
EvaRegs.COMCONA.all = 0x8E00; // Configurer le registre COMCONA
//0x83E7 // valider full compare 1010 0110 0000 0000
// bit15(1): valider l'operation Compare/PWM.
// bits14-13(00): Recharger COMPx (x=1,2,3) quand
// T1CNT==0(en underflaow).
// bit12(0): désactiver le mode Space Vector PWM.
// bits11-10(00): Recharger ACTRA quand T1CNT==0.
// bit9(1): validerr la logique compare correspondante.
// bit8(1): Retourne l'état actuel de la pin PDPINTA.
// bit7(1): les sorties Full Compare 3,PWM5/6, sont gérées
// par la logique Full Compare 3.
// bits6(1): les sorties Full Compare 2,PWM3/4, sont gérées
// par la logique Full Compare 2.
// bits5(1): les sorties Full Compare 1,PWM1/2, sont gérées
// par la logique Full Compare 1.
// bit4-3(00): Reservés.
// bit2(1): valider Full Compare 3 trip(C3TRIP).
// bit1(1): valider Full Compare 2 trip(C2TRIP).
// bit0(1): valider Full Compare 1 trip(C1TRIP).
}

void InitAdc(void)
{
AdcRegs.ADCTRL1.bit.RESET=1; //Reset ADC
asm(" rpt #10 || nop ");
AdcRegs.ADCTRL1.bit.RESET=0; //No reset
AdcRegs.ADCTRL1.bit.SUSMOD=3; //
AdcRegs.ADCTRL1.bit.ACQ_PS=0;
AdcRegs.ADCTRL1.bit.CPS=0; // AD_clock=HSPCLK/1=25MHz
AdcRegs.ADCTRL1.bit.CONT_RUN=0;
AdcRegs.ADCTRL1.bit.SEQ_CASC=1; //
AdcRegs.ADCTRL3.bit.ADCCLKPS=0; //ADC module clock = HSPCLK/1= 25MHz/(1)= 25MHz
AdcRegs.ADCTRL3.bit.SMODE_SEL=0; //
AdcRegs.ADCMAXCONV.all=0x0000; // (0+1) conversion
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0; // Configurer comme SEQ1 conv.
AdcRegs.ADCTRL2.bit.SOC_SEQ1=1; // Valider EVASOC pour le start SEQ1
}

// Les fonctions du programme principal

void cal_Ud_et_Uq()
{
//tgtheta= tan(theta);

```

