Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

A

ADC: Analog/Digital Converter

API: Application Programmable Interface

ARM: Advanced RISC Machine

B

BLDC: Brushless Direct Current

C

CAN : Convertisseur Analogique Numérique

CMMI: Capability Maturity Model Integration

CMSIS: Cortex Microcontroller Software Interface Standard

CPU: Central Processing Unit

D

DAC: Digital Analogic Converter

F

FEM: Force ElectroMotrice

FLL: Frequency Locked Loop

FOC: Field Oriented Control

FTM: Flex Timer

G

GPRs: General Purpose Registers

GUI: Graphical User Interface

I

I²C: Inter Integrated Circuit

I-PMSM: Interior Permanent Magnent Synchronous Motor

J

JTAG: Joint

L

LCD: Liquid Crystal Display

M

MC: Motor Control

MCG: Multi Clock Generator

MCU: Microcontroller Unit

MDT: Million Dinars Tunisien

MLI: Modulation Largeur d'Impulsion

MS: Machine Synchrone

N

NVIC: Nested Vectored Interrupt Controller

O

oMCI: object Motor Control Interface

oMCT: object Motor Control Tunning

P

PC: Program Counter

PI: Proportionnel Intégral

PIT: Periodic Interrupt Timer

PLL: Phase Locked Loop

PMSM: Permanent Magnent Synchronous Motor

PWM: Pulse Width Modulation

R

RAM: Random Access Memory

RTC: Real Time Clock

S

SDK: Software Development Kit

SDIO: Secure Digital Input Output

SM-PMSM: Surface Mounted Permanent Magnent Motor

SP: Stack Pointer

SPI: Serial Peripheral Interface

ST: STMicroelectronics

StdLib: Standard Library

SVPWM: Space Vector Pulse Width Modulation

U

UART: Universal Asynchronous Receiver Transmitter

UI: User Interface

USART: Universal Synchronous Asynchronous Receiver Transmitter

USB: Universal Serial Bus

Introduction générale

Durant ces dernières années, et grâce à la révolution technologique en microélectronique, les commandes numériques des systèmes électriques sont devenues plus utilisées et plus performantes dans le secteur industriel. Particulièrement, la commande numérique des moteurs électriques a permis l'ouverture de nouveaux horizons tant sur le plan logiciel que sur le plan technologique.

La société Telnet Holding, un leader mondial dans l'ingénierie produit et le domaine des hautes technologies, participe à ce progrès par la proposition d'une large gamme de solutions logicielles dans le domaine de la commande numérique en se basant sur des microcontrôleurs assez robuste pour satisfaire aux besoins de sa clientèle.

Dans ce cadre s'inclut notre projet qui porte sur la conception et le développement d'une application de commande des moteurs brushless ou moteurs sans balais dont les dispositifs de contrôle offrent d'immenses potentiels et de grandes opportunités en économies d'énergie, en haute précision et en amélioration significative en compatibilité électromagnétique (CEM). Le choix de ce type de moteurs est dû au faut qu'ils sont robustes et ne nécessitent pas d'entretien.

Ce projet est essentiellement constitué de deux grandes parties : la première est consacrée à la réalisation du contrôle vectoriel d'un brushless en se basant sur le microcontrôleur STM32 fournie par la société STMicroelectronics. La deuxième vise à commander un autre type des moteurs sans balais en utilisant le microcontrôleur K60N512 de la famille Kinetis fournie par Freescale.

Ce présent rapport s'articule en chapitres :

Dans le premier chapitre, nous allons présenter l'organisme d'accueil Telnet Holding et la division au sein de laquelle s'est déroulé ce travail. Ensuite, nous allons énoncer le cahier des charges. Nous allons passer après à une étude des types et des caractéristiques des moteurs sans balais.

Quant à la dernière partie de ce chapitre, elle expose l'environnement de travail matériel et logiciel pour les deux types de moteurs brushless

Dans le deuxième chapitre, nous allons vérifier tout d'abord la faisabilité des commandes pour les types des moteurs brushless considérés en simulant leurs modèles sur l'environnement logiciel Matlab Simulink. On passe par la suite à une description détaillée des différentes étapes suivies lors de la conception et l'implémentation des deux modèles de commande spécifiques à chaque type de moteur brushless

Chapitre 1

Présentation générale et étude des moteurs brushless

1.1 Introduction

Ce chapitre s'articule autour de trois parties principales. La première partie est consacrée à la présentation du cadre général du projet, comprenant l'entreprise d'accueil Telnet Holding, une étude détaillée de la problématique proposée et les différentes étapes à suivre pour l'accomplissement de ce projet.

Quant à la deuxième partie, elle traite des généralités sur les moteurs brushless portant sur les caractéristiques de leurs différents types et les principaux capteurs de positions qui peuvent être utilisés dans leur pilotage. Cette partie présente aussi les modèles de ces moteurs.

Finalement, la troisième partie de ce chapitre contient une description des différents composants constituant l'environnement du travail, y compris les différentes cartes nécessaires dans le pilotage, leurs caractéristiques et les outils logiciels utilisés.

1.2 Cadre du projet

1.2.1 Présentation de l'entreprise d'accueil

- **Mission**

« Développement de logiciels pour les technologies de l'information, l'informatique, les domaines scientifiques et techniques et des applications temps réel embarqué, en plus des tests logiciels utilisés pour les applications critiques » [1].

- **Profil**

Telnet Holding est un groupe de sociétés tunisien qui a été crée en 1994 par le fondateur principal actionnaire Mr Mohamed FRIKHA dans le but de satisfaire les besoin en développements de logiciels demandés par le multinationales du secteur des télécommunications, [2].

- **Place sur le marché**

Telnet Holding est un leader dans l'ingénierie produit et le conseil dans l'innovation et le domaine des hautes technologies. Son chiffre d'affaire augmente de 29 MDT en 2010 à 33 ,6 MDT en 2011.

De plus, c'est le premier groupe d'Afrique certifié CMMI niveau 5 ainsi qu'il dispose de la certification ISO 9001, version 2008 ce qui lui garanti plus de confidentialité. La clientèle de

ce groupe couvre plusieurs entreprises internationales telles que Johnson Controls, Zodiac Aerospace, Thomson, Sagem Communication.

- **Organisation**

Le groupe Telnet Holding est organisé par secteur de spécialité et par groupes managériaux comme est présenté à la figure 1.1.

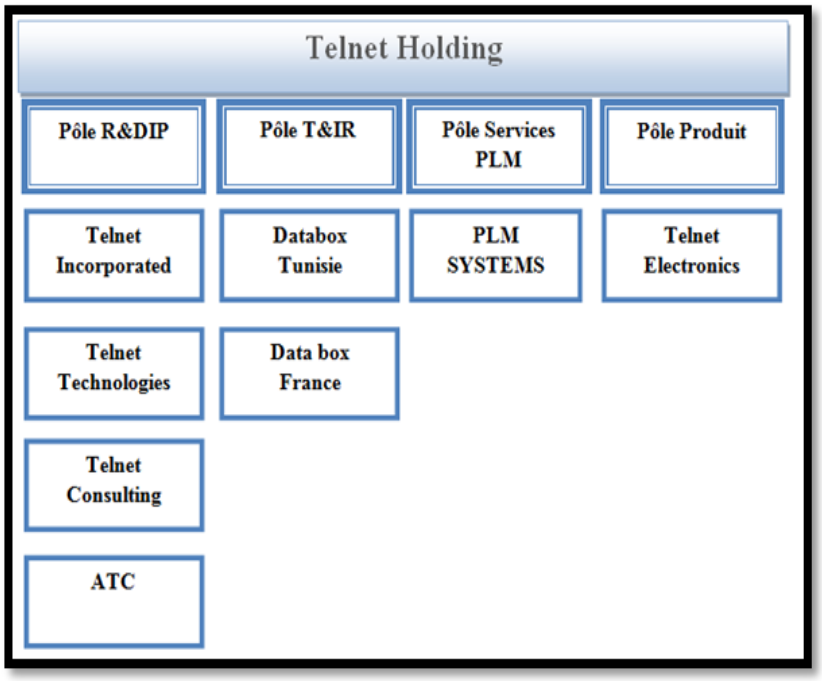


Figure 1.1 - Organigramme générale du groupe Telnet Holding

- **Secteurs d'activité**

Les activités de Telnet sont orientées essentiellement vers le design électronique et microélectronique, le développement embarqué et la conception des systèmes mécaniques. Ce groupe œuvre dans des différents secteurs tels que la « Télécommunications et Multimédia », le « Transport et Automotive », « Défense et Avionique », la « Sécurité et carte à puce », « Ingénierie mécanique » et «Electronique et industrie ».

Nous avons effectué notre projet de fin d'étude au sein de l'équipe industrie. C'est une équipe formée par des différentes plateformes de développement, au sein des quelles plusieurs projets ont été réalisés. Ces projets portent essentiellement sur les compteurs domestiques et industriels ainsi que les répétiteurs sonores. Le projet du moteur brushless est nouveau dans cette activité.

1.2.2 Présentation du projet

1.2.2.1 Problématique

Les moteurs sans balais ou brushless sont très utilisés dans différents domaines tels que l'automatisation industrielle, l'outillage électrique ainsi que les applications domotiques…

Dans ce cadre, et afin d'améliorer encore plus leurs performances, il faut bien choisir la commande de ce type du moteur. La question qui se pose, comment peut on piloter le brushless tout en garantissant la performance et la fiabilité des résultats ?

Les clients de Telnet Holding demandent parfois des applications de commande pour leurs moteurs brushless aussi complexes sans connaitre les limites de leurs faisabilités.

Pour satisfaire leurs clients, les industriels de Telnet Holding ont opté à l'introduction des deux types de commande : l'une est basée sur le principe de la transformation vectorielle et qui est réalisée autour du microcontrôleur STM32 en se référant à une librairie normalisée fournie par STMicroelectronics. L'autre exploite le pilotage par six étapes développé autour du microcontrôleur K60N512 de la famille Kinetis.

1.2.2.2 Définition du projet

- **Objectifs**

Ce projet de fin d'étude a pour thématique la conception et le développement des applications de commande d'un moteur sans balais ou moteur brushless.

- **Condition de réalisation du projet**

Avant de réaliser les applications de pilotage du moteur sans balais, il faut étudier la faisabilité du projet par :

- Etudier les caractéristiques et les différents types des moteurs brushless existants.
- Vérifier les résultats de simulation obtenus en appliquant quelques types de commande sur les modèles de brushless qui existent en utilisant l'environnement logiciel Matlab Simulink.
- Fixer le type de commande à appliquer selon les caractéristiques du moteur sans balais à piloter.

Après la spécification du type de commande adéquate avec les caractéristiques des moteurs, on passe à la réalisation des applications de commande selon le type de brushless.

- **Méthodologie**

La réalisation de ces applications passe par trois étapes principales :

- Conception
- Développement
- Validation

Ces étapes sont décrites dans ce qui suit en détail.

1.3 Présentation des moteurs sans balais

1.3.1 Types des moteurs brushless

Les moteurs sans balais ont une structure semblable aux machines à courant continues. Ils incluent :

-***Une partie fixe (stator) :*** C'est un système formé de trois enroulements à p paires de pôles.

-***Une partie mobile (rotor) :*** Il peut être formé par des aimants permanents ou par un bobinage alimenté par un courant continu, appelé aussi électro-aimant.

Lors de la rotation du moteur, le phénomène d'auto-induction se produit :

Lorsque les trois bobines statoriques sont parcourues par des courants électriques, on aura une génération d'un champ magnétique.

La variation de ce flux Φ au cours du temps engendre une force électromotrice FEM d'après la loi de Lenz-Faraday [3] :

$$E = -\frac{d\Phi}{dt} \qquad \textbf{(1.1)}$$

Ces FEM sont réparties sur le circuit magnétique interne du brushless et selon sa structure, les formes de ces forces se distinguent. Ceci permet de classer les moteurs sans balais en deux grandes familles : les BLDC et les PMSM, [4].

1.3.1.1 Les moteurs BLDC

C'est la famille des moteurs sans balais à courant continu, elle est caractérisée par une forme trapézoïdale du circuit magnétique comme est présentée dans la figure 1.2.

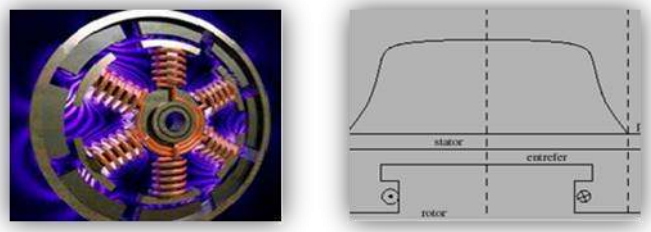


Figure 1.2 – Forme du circuit magnétique du BLDC

De ce fait, les lignes de champs sont réparties selon cette forme du circuit magnétique et on aura des FEM trapézoïdales comme est montré dans la figure 1.3.

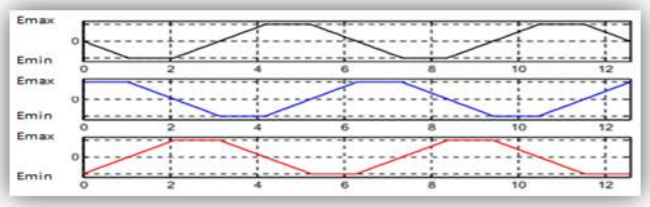


Figure 1.3 - FEM trapézoïdales du BLDC

Le moteur BLDC est caractérisé par un système triphasé équilibré des courants. Ces courants ont une forme en créneaux ou quasi-rectangulaires.

1.3.1.2 Les moteurs PMSM

C'est la famille des moteurs sans balais synchrone à aimants permanents. Elle est caractérisée par une forme circulaire du circuit magnétique comme est présenté dans la figure 1.4.

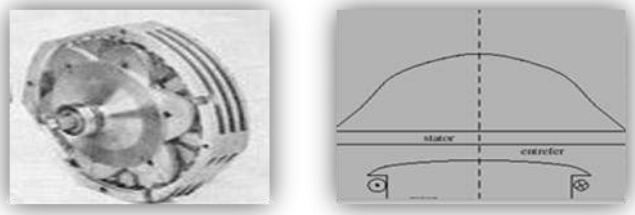


Figure 1.4 - Forme du circuit magnétique du PMSM

A cause de cette forme, les lignes de champs sont réparties en suivant l'allure circulaire de l'entrefer. Les FEM ont donc une forme sinusoïdale comme est montré dans la figure 1.5.

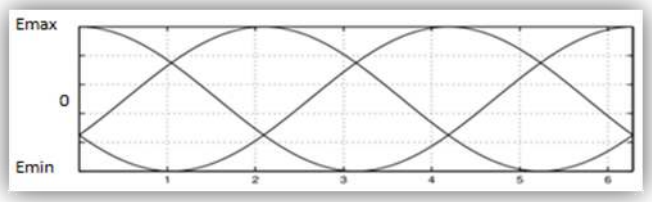


Figure 1.5 - FEM sinusoïdales du PMSM

Le PMSM est caractérisé par un système triphasé et équilibré de courants qui ont une forme sinusoïdale. Pour cette famille des brushless, le rotor peut avoir deux formes :

- Les aimants du rotor peuvent être collés à sa surface, ce sont les brushless à pôles lisses connus sous le nom SM-PMSM
- Comme on peut avoir des aimants incorporés dans la structure du rotor, c'est la caractéristique des brushless à pôles saillants ou les IPMSM.

1.3.2 Les capteurs de position

Les brushless sont généralement équipés par des capteurs de position qui retournent des informations sur le rotor : sa position, sa vitesse…

Notre étude est basée essentiellement sur deux types de capteurs : les capteurs à effet Hall et l'encodeur incrémental [5].

1.3.2.1 Les capteurs à effet Hall

Ils sont utilisés pour détecter le champ d'induction produit par les aimants et pour ainsi mesurer par secteur la position du rotor.

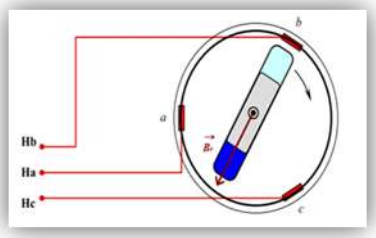


Figure 1.6 - Capteurs à effet Hall

Les capteurs à effet Hall délivrent trois signaux carrés décalés de 120° chacun par rapport à l'autre comme est montré sur la figure 1.7.

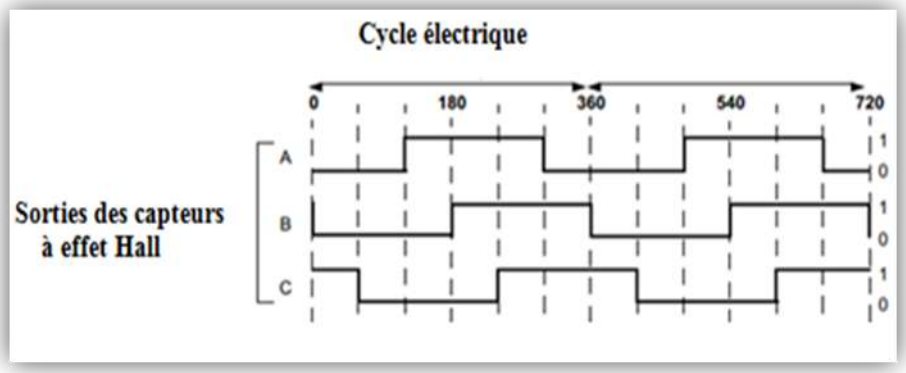


Figure 1.7 - Les signaux délivrés par les capteurs à effet Hall

Ces capteurs ne sont pas très précis car ils délivrent la position du rotor par secteur de 60°.

1.3.2.2 L'encodeur incrémental en quadrature

Il est constitué d'un disque rotatif fixé sur l'axe du rotor, deux pistes de code dont les secteurs sont décalés de 90 degrés d'une piste à l'autre, une source de lumière et un capteur, comme est présenté dans la figure 1.8.

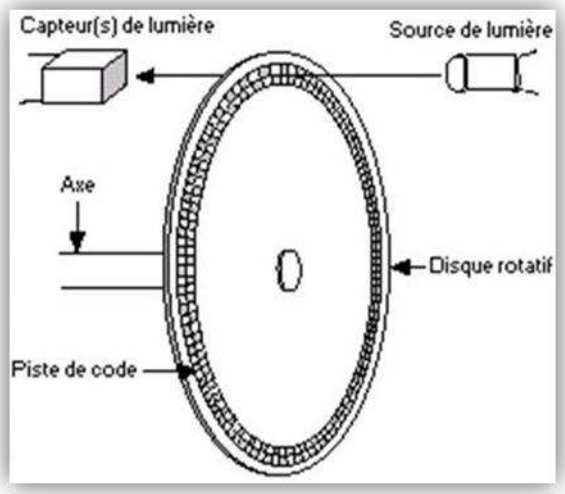


Figure 1.8 - Constitution de l'encodeur incrémental

Lorsque le disque tourne, les segments opaques bloquent la lumière alors que ceux où le verre est clair la laissent passer.

Ceci génère des impulsions d'onde carrée comme est montré dans la figure 1.9 et qui peuvent ensuite être décodées par un compteur et interprétées comme position.

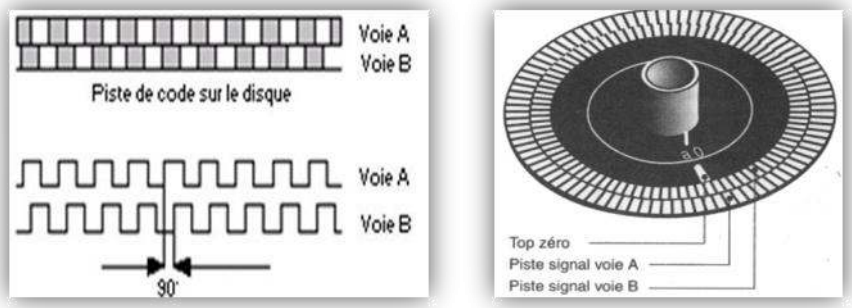


Figure 1.9 - Les signaux des voies A et B

Le sens de rotation du disque dépend de la position des deux pistes : si A devance B d'un quart de période, le disque tourne dans le sens positif (Le sens de l'aiguille d'une montre), si non, il tourne dans le sens inverse.

Ces capteurs sont très précis et délivrent la position du rotor avec une bonne résolution.

1.3.3 Modélisation d'un PMSM

Le moteur à modéliser est une machine synchrone à aimants permanents. Ce moteur possède trois enroulements statoriques, comme est représenté par la figure 1.10. Ces enroulements sont à alimenter par des courants alternatifs.

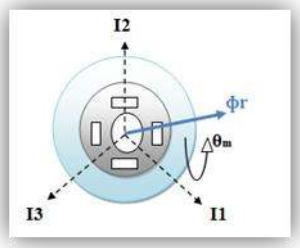


Figure 1.10 - Modèle triphasé du PMSM

La phase rotorique est décalée de la première phase statorique d'un angle mécanique **θm** qui indique la position angulaire du rotor par rapport au stator.

Le stator d'un PMSM est la partie fixe alors que le rotor est la partie mobile et tourne à une vitesse mécanique **Ω**.

La position électrique du rotor **θr** ainsi que sa vitesse électrique **ωr** sont obtenues en multipliant les grandeurs électriques par le nombre de paire de pôle p.

$$\Omega = d\,\theta_m / dt\ ;\ \theta r = p\,\theta r\ ;\ \omega r = p\,\Omega \qquad \textbf{(1.2)}$$

Les courants triphasés i1, i2 et i3 peuvent être représenté par un vecteur spatial $\vec{I}$ qui constitue la base de la commande vectorielle.

$$\vec{I} = \frac{2}{3}\left(i1 + e^{j\frac{2\pi}{3}} i2 + e^{j\frac{4\pi}{3}} i3 \right) \qquad \textbf{(1.3)}$$

Pour passer d'un système de courant triphasé i1, i2 et i3 aux composantes complexes du vecteur spatial $I\alpha$ et $I\beta$, on fait la transformation de Clarke donnée par la relation suivante :

$$\begin{bmatrix} I\alpha \\ I\beta \\ Io \end{bmatrix} = [C] \begin{bmatrix} i1 \\ i2 \\ i3 \end{bmatrix}\ ;\ [C] = \frac{2}{3} \begin{pmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{pmatrix} :\ \text{Matrice de Clarke} \qquad \textbf{(1.4)}$$

Où Io est la composante homopolaire. Pour un système triphasé équilibré de courant, c'est-à-dire pour i1+i2+i3 = 0 ce qui est notre cas, cette composante est nulle.

La transformation de Clarke sera donc simplifiée et il suffit d'avoir deux courants statoriques afin de calculer les composantes $I\alpha$ et $I\beta$.

On obtient ainsi l'expression simplifiée de la transformation de Clarke donnée par la relation suivante :

$$\begin{bmatrix} I\alpha \\ I\beta \end{bmatrix} = [C] \begin{bmatrix} i1 \\ i2 \end{bmatrix},\ [C] = \begin{pmatrix} 1 & 0 \\ 1/\sqrt{3} & 2/\sqrt{3} \end{pmatrix} :\ \text{Clarke simplifiée} \qquad \textbf{(1.5)}$$

La figure 1.11 présente le modèle du moteur synchrone dans le système de coordonnée (α, β).

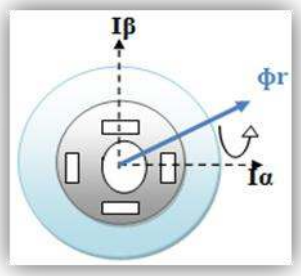


Figure 1.11 - Modèle du PMSM dans le système de coordonnées (α, β)

Pour revenir du système biphasé à un système triphasé, on effectue la transformation de Clarke inverse.

$$\begin{bmatrix} i1 \\ i2 \\ i3 \end{bmatrix} = \left[C^{-1}\right] \begin{bmatrix} I\alpha \\ I\beta \end{bmatrix}, \left[C^{-1}\right] = \begin{pmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{pmatrix} : \text{ Clarke inverse} \qquad \textbf{(1.6)}$$

La commande vectorielle permet de transformer les grandeurs sinusoïdales en des grandeurs continues.

Pour ce faire, on modélise le PMSM dans un système de coordonnées commun **dq** calé sur l'axe de l'enroulement rotorique.

Ceci est réalisé en effectuant une rotation d'un angle **θdq** égale à **θr** du système biphasé obtenu sur la figure 1.11 : Ce changement de repère est assuré via la transformation du Park donnée par :

$$\begin{bmatrix} Id \\ Iq \end{bmatrix} = \left[R(\theta_{dq})\right] \begin{bmatrix} I\alpha \\ I\beta \end{bmatrix} ; \left[R(\theta_{dq})\right] = \begin{pmatrix} \sin(\theta_{dq}) & \cos(\theta_{dq}) \\ -\cos(\theta_{dq}) & \sin(\theta_{dq}) \end{pmatrix} : \text{ Matrice de Park} \qquad \textbf{(1.7)}$$

Le modèle électrique d'un PMSM dans le plan dq est obtenu en faisant les transformations de Park et Clarke, et ca permet d'avoir des courants continus faciles à régler par des PI.

La figure 1.12 représente la transformation dans le plan (d, q).

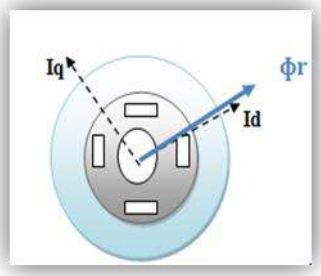


Figure 1.12 - Modèle du PMSM à pôles lisses dans le plan (d, q)

L'expression du couple électromagnétique dans ce système de coordonnées commun est donnée par les relations suivantes.

-Pour un moteur synchrone à aimant permanant et à pôles lisses : SM-PMSM

$$Cem = \frac{3}{2} p \, \Phi m \, iqs \qquad \textbf{(1.8)}$$

-Pour un moteur synchrone à aimant permanant et à pôles saillants : I-PMSM

$$Cem = \frac{3}{2} p \, \Phi m \, iqs + \frac{3}{2} p \, Lds - Lqs \, iqs \, ids \qquad \textbf{(1.9)}$$

$$Créluctant = \frac{3}{2} p \, Lds - Lqs \, iqs \, ids \qquad \textbf{(1.10)}$$

Avec

p : Le nombre de paire de pôles

Lds : Inductance statorique suivant l'axe directe d

Lqs : Inductance statorique suivant l'axe en quadrature q

Φm : Flux magnétique des aimants permanents rotoriques

Créluctant : Couple réluctant

Quant au modèle mécanique de ce moteur, il est donné par la relation suivante :

$$J\frac{d\omega r}{dt} = p(Cem - Cr) - f\omega r \qquad \textbf{(1.11)}$$

Avec

J : Moment d'inertie

Cr : Couple résistant imposé par la charge mécanique

f : Coefficient de frottement visqueux

Les constantes du temps du moteur sont données par :

Constante de temps mécanique : $Tm = J / f$ **(1.12)**

Constante de temps électrique selon l'axe d : $Ted = Lds/Rs$ **(1.13)**

Constante de temps électrique selon l'axe q : $Teq = Lqs/Rs$ **(1.14)**

A cause de la saillance de la machine, un couple réluctant s'ajoute à l'expression du couple d'un moteur à pole lisses. Il est donné par de la relation (1.10).

A partir du modèle mécanique du PMSM donné par la relation (1.11), on remarque qu'il faut avoir toujours un couple électromagnétique maximal au démarrage pour éviter tout risque de décrochage et conserver le fonctionnement moteur.

De ce fait, et pour un SM-PMSM, le couple électromagnétique devient maximal dans le cas ou son courant en quadrature est égal à son courant nominal.

Pour un I-PMSM, les deux composantes, directe et quadrature, du courant ont une influence sur le couple électromagnétique pour qu'il soit maximal avec la plus petite valeur de courant.

C'est l'idée d'ajouter une table qui traite ce phénomène dans les différentes techniques de commande.

1 .4 Environnement de travail

1.4.1 Environnement de travail pour le PMSM

1.4.1.1 Partie commande

Le support utilisé pour l'implémentation de l'algorithme de commande du brushless PMSM est une solution matérielle basée sur la carte STM3210E-EVAL.

C'est une plateforme de développement fournie par STMicroelectronics qui est basée sur le STM32F103ZDT6 de la famille STM32F des microcontrôleurs 32 bit à haute performance.

- **Caractéristiques du STM32**

STM32, est un microcontrôleur produit par STMicroelectronics. Il est fondé sur le cœur standard ARM Cortex™-M3, de fréquence 72MHz. Ce cœur est caractérisé par un ensemble de fonctionnalités dédiées aux applications embarquées qui se manifestent par la haute performance, la réduction de la consommation et la faiblesse du coût.

La figure 1.13 présente son architecture interne.

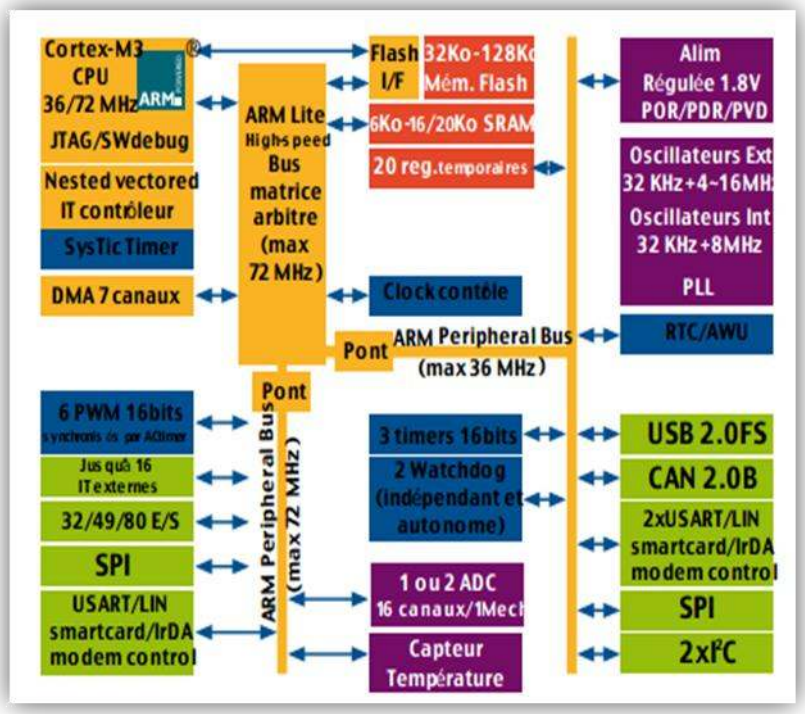


Figure 1.13 - Architecture interne du STM32 [6]

Ce microcontrôleur inclut deux types de mémoires : la mémoire Flash qui supporte jusqu'à 512K octets de données et la RAM de taille 64K octets.

Ses périphériques sont aussi variés et son subdivisés an deux catégories :

-Périphériques analogiques **:** Trois convertisseurs analogiques numériques 12-bit avec une période d'échantillonnage de 1µs et deux convertisseurs numériques analogiques 12-bit.

-Périphériques numériques **:** Les périphériques de communication sont très variés afin de répondre au plus grand nombre d'application: SPI, USART, I²C, SDIO, USB, CAN.

- **Caractéristiques de la carte STM3210E-EVAL**

La maquette STM3210E-EVAL inclut deux oscillateurs, le premier est externe et délivre une horloge temps réel (RTC) de fréquence allant jusqu'à 16MHz et le deuxième est interne ayant une horloge de CPU lors de la réinitialisation de 8MHz. Ses entrées/sorties possèdent un niveau logique 0-3.3 V. Les différents constituants sont représentés à la figure 1.14.

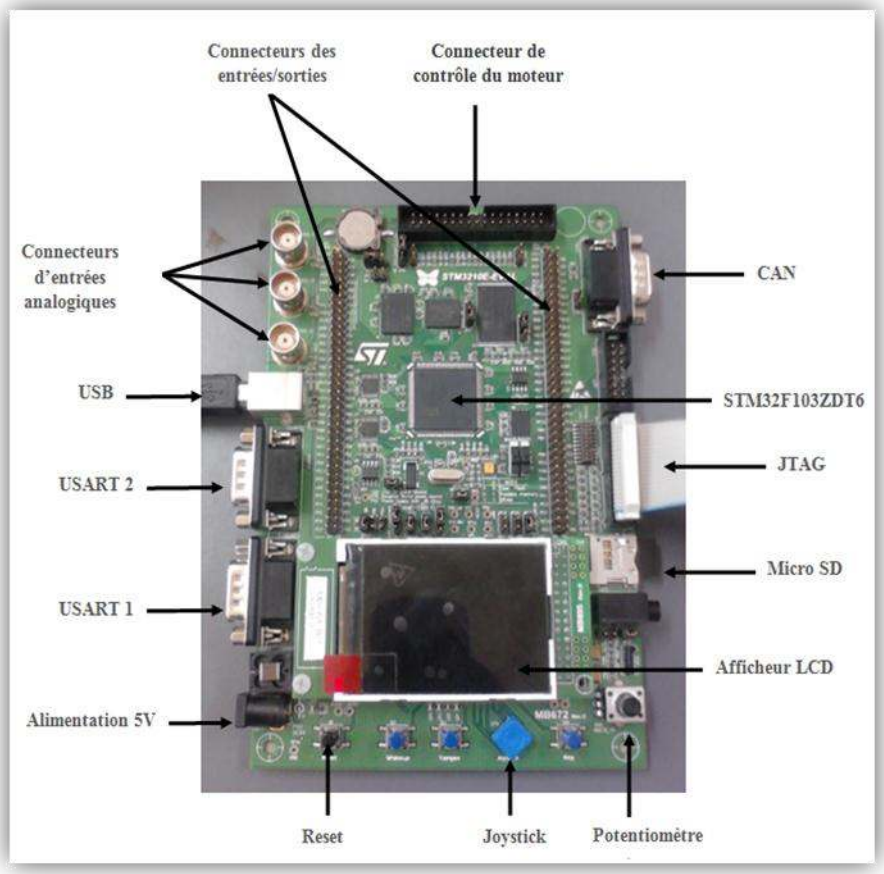


Figure 1.14 - Carte de commande STM3210E-EVAL

La carte STM3210E-EVAL est composée de plusieurs blocs ayant différentes fonctions.

-***Connecteur RS232*** **:** Il est représenté par les deux interfaces séries, USART 1 et USART 2. L'interface USART 2 est multiplexée avec les connecteurs de contrôle moteur.

-Connecteur de contrôle du moteur : La carte STM3210E-EVAL permet de commander un brushless triphasé en utilisant 34 pines qui permettent l'échange des signaux avec la carte de puissance. On peut mesurer également le signal d'arrêt d'urgence, le signal de la vitesse du moteur, les signaux de ses courants, le signal de la tension de bus continue, un signal de la température du moteur, on trouve également deux pines pour la connexion du capteur de position incrémental ainsi que sept canaux pour les signaux PWM délivrés pour l'étage de puissance.

-Capteur de température : Il est représenté par une interface I²C, qui supporte une température de -55°C jusqu'à 125°C connecté à l'interface I²C du microcontrôleur.

-***Entrées Analogiques :*** Ce sont trois connecteurs analogiques, CN2, CN3 et CN5 reliés directement aux entrées analogiques externes PC3, PC2 et PC1 du microcontrôleur.

-***Support de débogage :*** Le moyen utilisé pour le chargement de notre application est l'interface JTAG.

-Afficheur LCD : On peut visualiser différentes pages qui communiquent avec le code à saisir pour faciliter la communication pendant le pilotage du moteur, le Joystick possède quatre direction et permet la sélection des différentes options dans les pages à programmer .

-***Connecteurs d'entrées/sorties :*** On trouve 112 entrées /sorties valables permettant la liaison entre la carte de commande et la carte de puissance pour piloter le moteur.

-***Contrôleur d'interruptions (NVIC) :*** Le Cortex-M3 du microcontrôleur implémente un contrôleur d'interruption capable de gérer jusqu'à 60 canaux d'interruption masqués et 16 niveaux de priorité.

-Temporisateurs : Les principaux Timers du microcontrôleur qui s'interviennent dans les différentes parties de la commande peuvent être classés en quatre familles.

Advanced-control timers (TIM1 et TIM8) : Configurés en 16 bit pour la génération des signaux PWM avec des temps morts spécifiés.

General-purpose timers (TIMx : TIM2, TIM3, TIM4 et TIM5) : Ils sont à usage général permettant de générer les sorties PWM et manipuler les signaux de l'encodeur.

Basic timers (TIM6 and TIM7) : Utilisés généralement pour la génération des convertisseurs numérique analogique DAC.

SysTick timer : Dédié à la gestion de la base du temps (real time operating system)

En addition de cette architecture spécifique de la carte STM3210E-EVAL et ses différentes fonctionnalités, l'une de ses principales caractéristiques est qu'elle permette de commander deux brushless PMSM à la fois avec ou sans capteurs selon les besoins.

L'application réalisée sera par la suite chargée dans cette carte via une interface Jlink.

- ***Interface J-Link***

L'interface J-Link assure la liaison entre le microcontrôleur et l'outil de débogage. Du côté de la carte, la communication des informations se fait grâce au port JTAG et du côté du PC, la communication se fait à travers le port USB comme le montre la figure 1.15.

Figure 1.15 - Interface J-Link

Les signaux de commande sont générés par cette maquette et seront envoyés par la suite vers la partie puissance pour piloter le PMSM.

1.4.1.2 Partie puissance

- **Critère de choix de la carte de puissance**

Le dispositif de puissance commandé est un onduleur de tension triphasé qui est un convertisseur de tension continue alternatif : Cet élément de puissance va recevoir les signaux de commande générés par la maquette STM3210E-EVAL et en commutant ses différents interrupteurs, il va faire passer la tension nécessaire aux phases du moteur et ainsi le piloter.

Le cahier des charges exige un fonctionnement quatre quadrants du brushless présenté par la figure 1.16, de ce fait, nous avons à choisir une carte de puissance qui doit satisfaire cette condition tout en étant compatible avec la maquette de commande présentée.

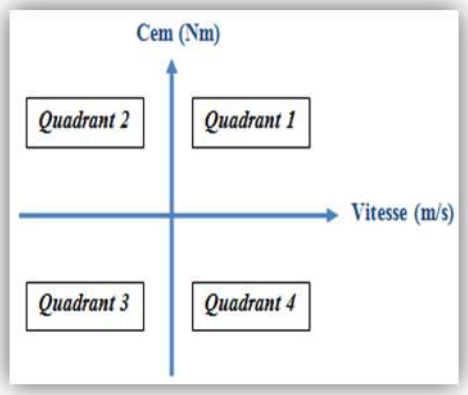


Figure 1.16 - Principe de fonctionnement quatre quadrants

Ce principe de fonctionnement est présenté dans [7] par :

Quadrant 1 : La puissance est positive, il s'agit d'un fonctionnement moteur, et le couple électromagnétique est positif donc c'est une marche avant

Quadrant 2 : La puissance est négative, il s'agit d'un fonctionnement génératrice (ou frein), et le courant délivré est positif donc il s'agit d'un marche avant.

Quadrant 3 : La puissance est positive, c'est le principe de l'accélération, et le courant est négatif donc c'est une marche arrière.

Quadrant 4 : La puissance et le couple sont négatifs, c'est alors un freinage en marche arrière.

Le fonctionnement moteur est déjà assuré par l'onduleur triphasé. En cas de freinage, le courant va retourner dans le sens inverse de sorte que le brushless le génère.

Le pont redresseur est unidirectionnel et il ne permet pas le passage du courant qui va circuler dans la capacité et la surcharger.

Dans ce cas, il faut avoir un mécanisme permettant d'absorber ce courant de retour, c'est ce qu'on appelle le freinage dissipatif ou le hacheur de freinage [8]. La figure 1.17 montre le bloc de freinage.

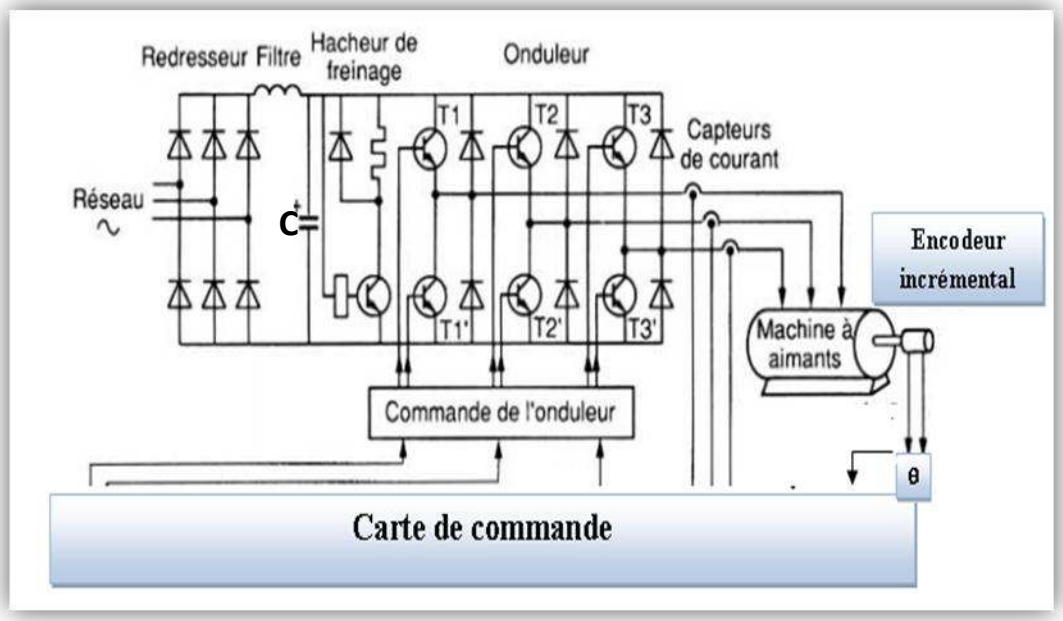


Figure 1.17 - Entrainement électrique d'une MS incluant la fonction de freinage

La tension présentée aux bornes de la capacité C est celle qui va être commuté dans les phases statoriques, appelée la tension de bus continu.

Dans un premier temps, le moteur tourne avec une puissance P1= Cem. Vitesse1, lors d'un freinage brusque la vitesse du moteur diminue et il commence à tourner avec une puissance P2 inférieure à P1, il y a donc une évacuation d'énergie.

Dés que la tension aux bornes de la capacité dépasse un certain seuil, le transistor se ferme et la dissipation de courant se fait à travers la résistance généralement fabriquée en céramique pour pouvoir supporter les pertes joules.

En se basant sur ce principe, nous avons choisi la carte de puissance **: STEVAL-IHM023V2.**

- **Caractéristiques de la carte choisie**

La STEVAL-IHM023V2 est dédiée au PMSM triphasé de puissance allant jusqu'à 1KW avec ou sans capteur. Elle est composée des différents blocs comme montre la figure 1.18.

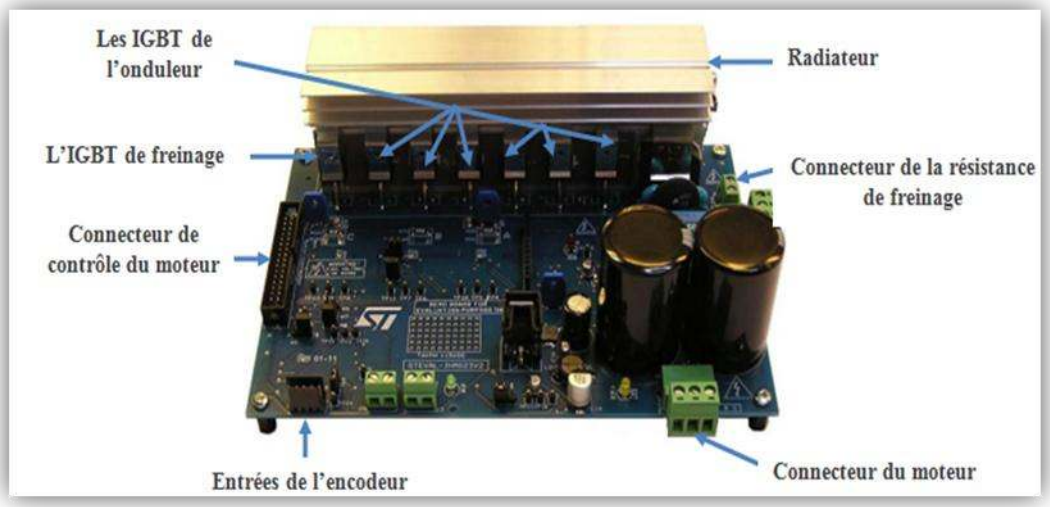


Figure 1.18 - Carte de puissance STEVAL-IHM023V2

-Bloc de freinage (dissipative brake) : Ce bloc permet au moteur d'évacuer son énergie lors du fonctionnement génératrice en éliminant la surtension provoquée aux bornes de la capacité de bus continu qui est environ 440 VDC, la résistance dissipative est connectée de l'extérieur.

*-**Bloc de commande de l'onduleur :*** Ce bloc va permettre d'envoyer les signaux PWM traités dans la partie commande vers les grilles des transistors de l'onduleur triphasé.

-Capteurs de courant : La lecture des courants de phases du moteur est effectuée par les résistances shunt : la tension appliquée à la borne d'un shunt permet de déterminer le courant qui y traverse selon la loi d'ohm.

*-**Bloc de protection contre les surintensités :*** Ce circuit permet de protéger les résistances shunts dans le cas ou le courant qui y traverse dépasse la valeur maximale fixé à 7A.

- Bloc de protection contre la surchauffe : Il permet la protection des interrupteurs de puissance si la température de leurs jonctions dépasse 70°.

Elle est mesurée par un thermostat placé dans le radiateur et le signal sera envoyé vers la partie commande pour le traiter

1.4.1.3 Outils logiciels du travail

1.4.1.3.1 L'IAR

L'environnement de développement intégré que nous avons utilisé est l'IAR.

- **Historique**

Le système IAR est crée en 1983 par l'ingénieur ***Anders Rundgren***, C'est l'abréviation de « **I**ngenjörsfirman **A**nders **R**undgren » qui veut dire la société d'ingénierie de ***Anders Rundgren.*** Il s'agit d'un ensemble des outils de développement pour les applications embarquées.

- **Outils intégrés**

L'outil IAR Embedded Workbench contient une chaine complète d'outils intégrés : Compilateur ARM IAR C/C++, un assembleur, éditeur de lien IAR XLINK, un éditeur de texte adapté, un gestionnaire de projets, Débogueur à langage haut niveau : IAR C-SPY.

- **Interface d'IAR**

La figure 1.19 représente les différents blocs de l'espace de travail IAR

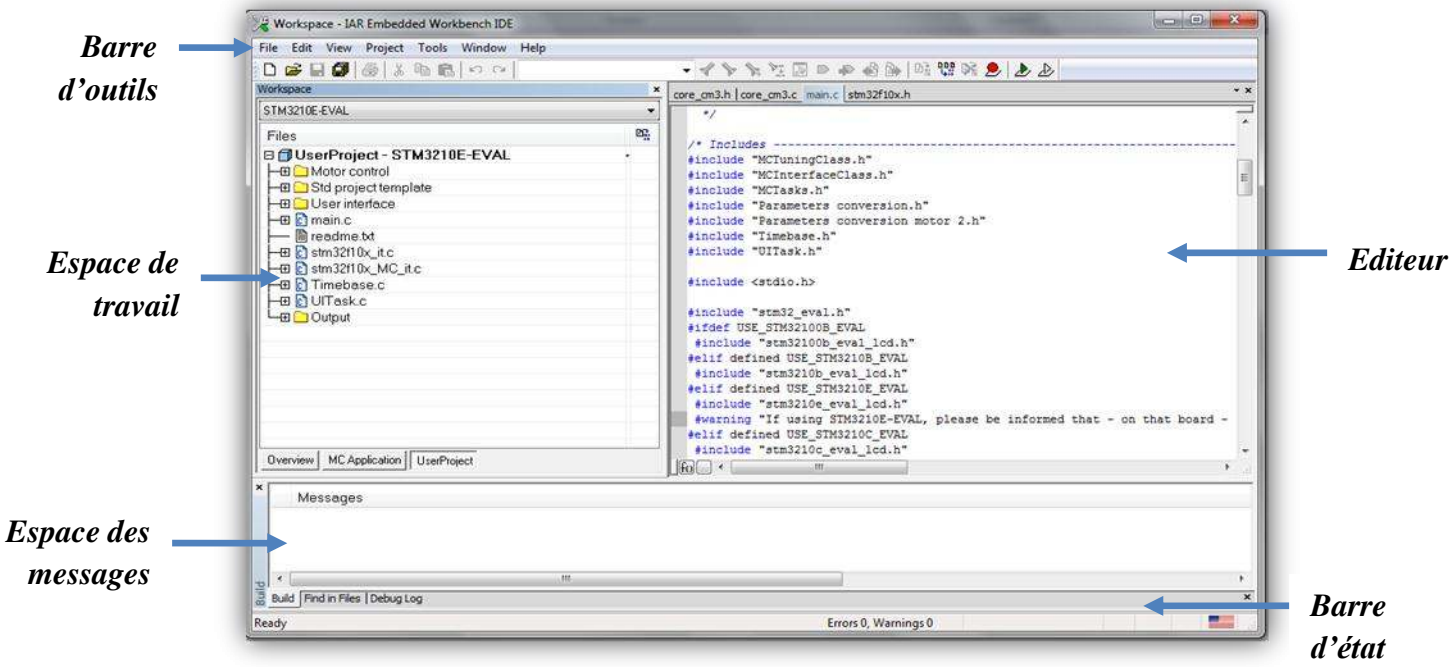


Figure 1.19 - Environnement de travail IAR EWARM

1.4.1.3.2 Le GUI de STMicroelectronics

Un outil (GUI) fourni par STMicroelectronics, permet de saisir les informations nécessaires pour le pilotage du moteur. La figure 1.20 montre l'interface de cet outil.

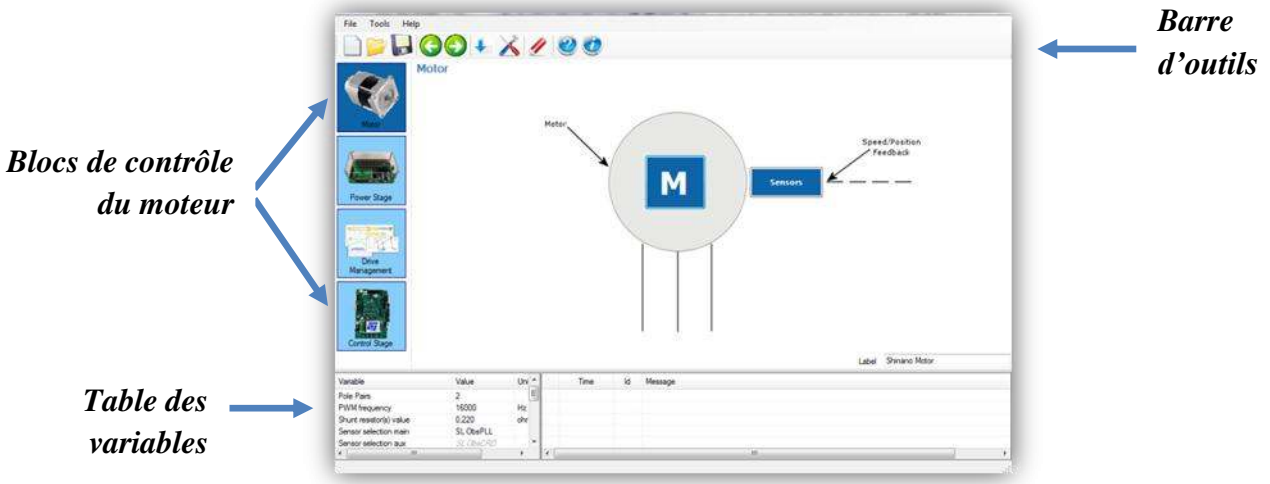


Figure 1.20 - Le GUI de ST pour le contrôle du moteur

Cet outil permet d'entrer les paramètres électriques du moteur, de l'étage de puissance, de la carte de commande et les différents paramètres de contrôle nécessaires au développement de l'application de commande, quatre interfaces permettent de saisir ces paramètres.

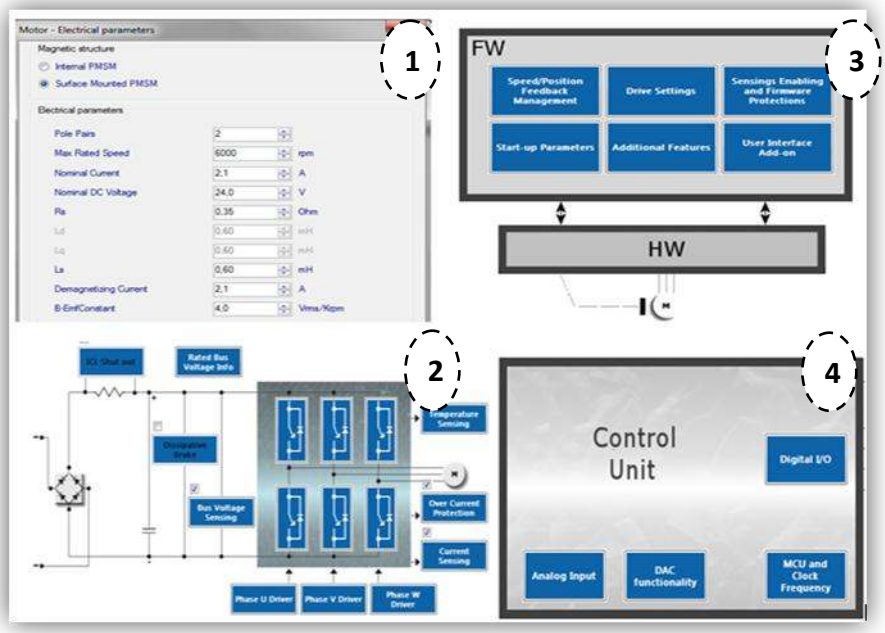


Figure 1.21 - Interfaces de paramétrage

1.4.2 Environnement de travail pour le BLDC

Le support utilisé pour l'implémentation de l'algorithme de commande du deuxième type des moteurs sans balais, le BLDC, est une solution matérielle basée sur la maquette Freescale. Cette maquette est composée d'une carte de commande TWR-K60N512, d'une carte de puissance TWR-MC-LV3PH, une carte d'interface série TWR-SER, deux modules élévateurs, TWR-elevator, permettant l'alimentation et l'interconnexion entre ces différentes cartes, et un moteur BLDC.

1.4.2.1 Partie commande

La carte de commande est une carte TWR-K60X512 basée sur le microcontrôleur K60N512VMD100 de la famille Kinetis 60, (K60), 32 bits.

- **Caractéristique de K60N512**

Ce microcontrôleur est fondé sur le cœur standard ARM Cortex™-M4 de fréquence 100MHz, incluant les fonctionnalités de DSP qui permettent l'exécution des algorithmes complexes et le contrôleur d'interruption qui caractérise l'architecture d'ARM v7-M.

La figure 1.22 montre l'architecture interne de ce microcontrôleur.

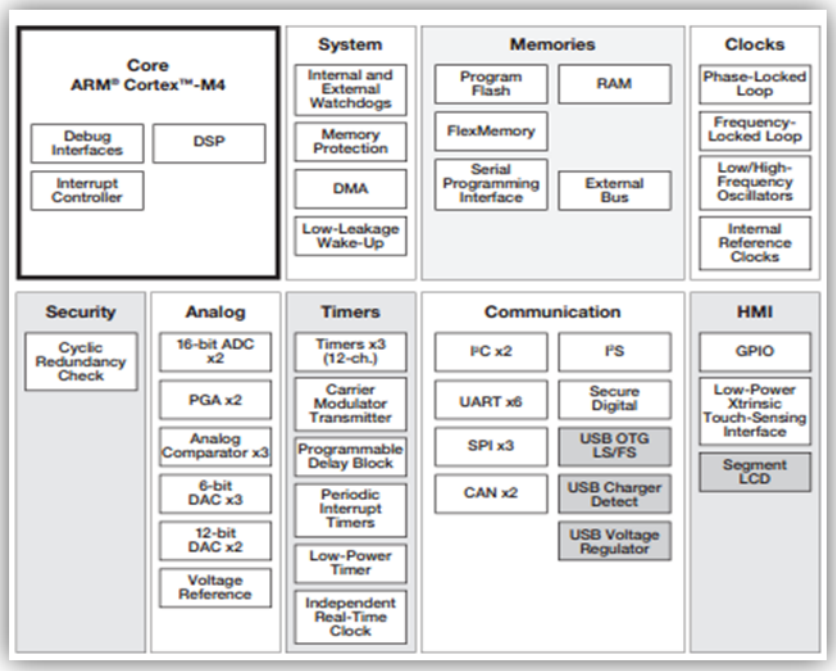


Figure 1.22 - Architecture interne du K60N512 [9]

Le K60N512 inclut trois types de mémoires : la mémoire Flash de taille 512K octets d'où la nomenclature N512, la RAM qui peut supporter jusqu'à 128K octets ainsi qu'une mémoire FlexMemory.

Cette mémoire est constitué d'une FlexNVM qui peut être utilisée comme une Flash additionnelle de taille 256K octets, et d'une, FlexRAM qui peut être utilisée comme une RAM de taille 4K octets.

- **Caractéristiques de la carte TWR- K60N512**

La maquette TWR-K60N512 inclut trois modules d'horloge :

Le module MCG : il génère plusieurs horloges, une horloge délivrée par une PLL à base d'un oscillateur de contrôle numérique, une horloge délivrée par une FLL à base d'un oscillateur de contrôle de tension, ainsi que des horloges de référence interne. Ce module délivre une fréquence entre 32.768KHz et 32MHz.

L'oscillateur de système : Il génère l'horloge de référence pour le MCU

L'oscillateur d'horloge temps réel : Il délivre une horloge temps réel (RTC) de fréquence 32.768KHz et peut remplacer l'oscillateur de système dans certains cas d'application.

Les différents constituants sont détaillés à la figure 1.23

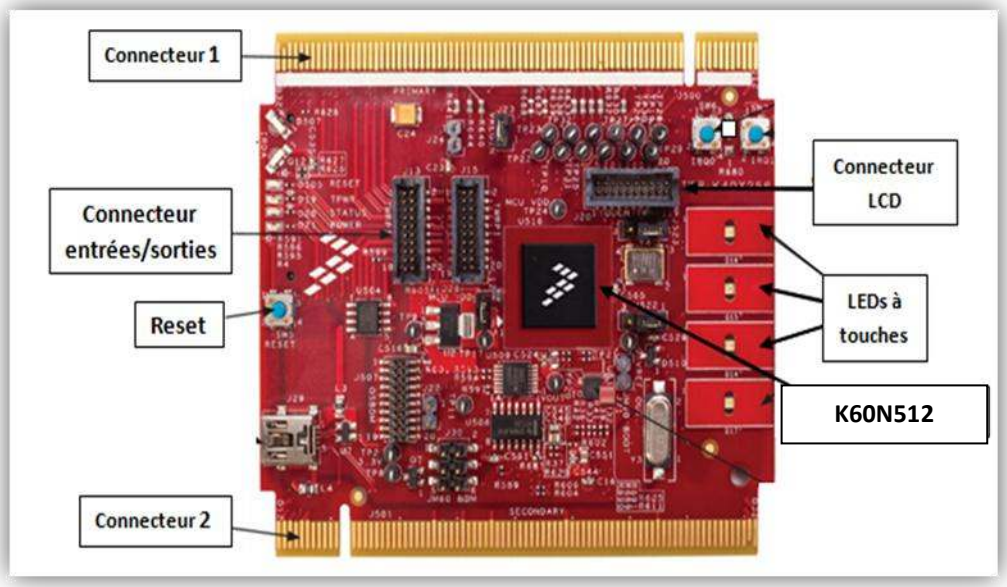


Figure 1.23 - Carte de commande TWR-K60N512

1.4.2.2 Partie puissance

La carte de puissance TWR-MC-LV3PH est alimentée par une tension continue de 12 à 24V.

Elle est composée d'un connecteur pour les capteurs à effet Hall, un connecteur pour le moteur BLDC, un bloc d'onduleur triphasé à base des MOSFETs, commandé par un circuit MC33937, un bloc assurant le principe de fonctionnement quatre quadrants grâce à une résistance dissipative ***R Brake*** comme est décrit au paragraphe (1.4.1.2), ainsi que des blocs de protection contre la surtension et la surintensité.

La figure 1.24 montre ces différents blocs.

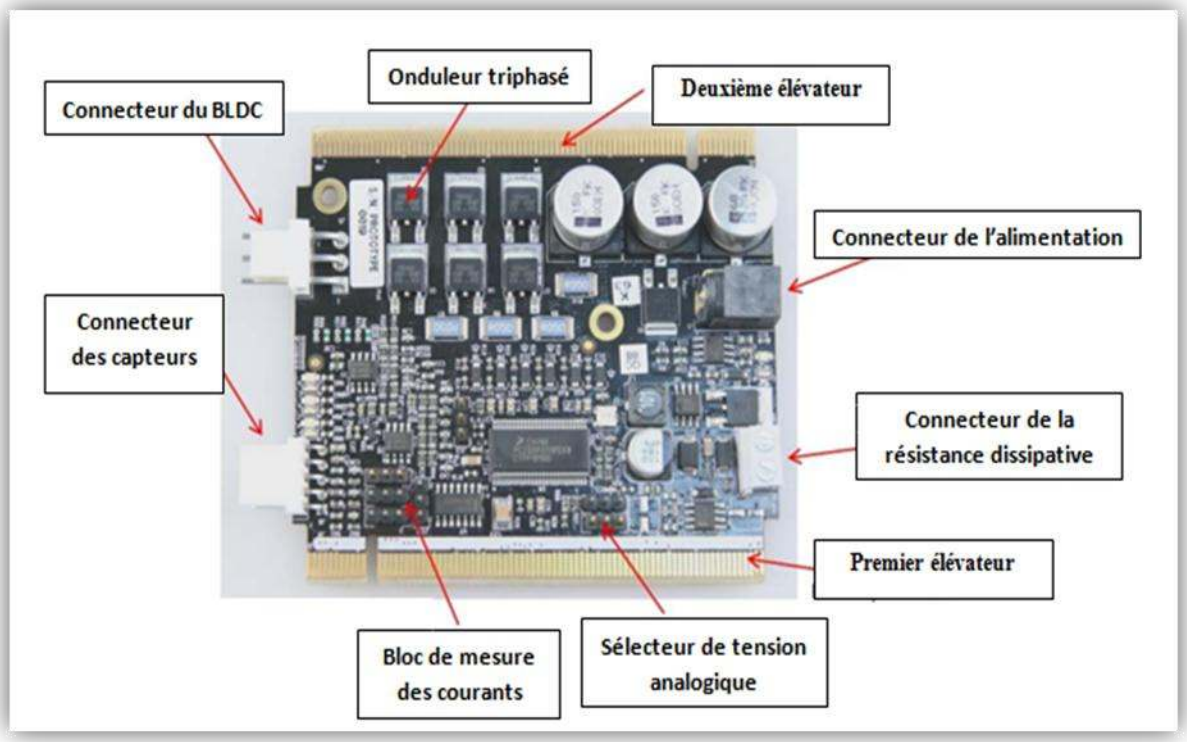


Figure 1.24 - Carte de puissance TWR-MC-LV3PH

Cette carte reçoit les signaux de commande de la partie commande TWR-K60N512, à travers le premier élévateur.

Ces ordres de commande seront traités dans le bloc MC33937A pour générer six signaux MLI qui sont nécessaires pour commuter l'onduleur triphasé et ainsi faire passer la tension nécessaires aux phases du moteur BLDC.

Les différents types de contrôles effectués sur ce bloc tels que le contrôle du bloc dissipatif, les signaux filtrés des secteurs générés par les capteurs à effet Hall, les ordres de commande du bloc MC33937A, sont gérés par la carte de commande et la communication est assurée par le premier élévateur.

La figure 1.25 présente schéma simplifié qui explique l'échange des différents flux de données.

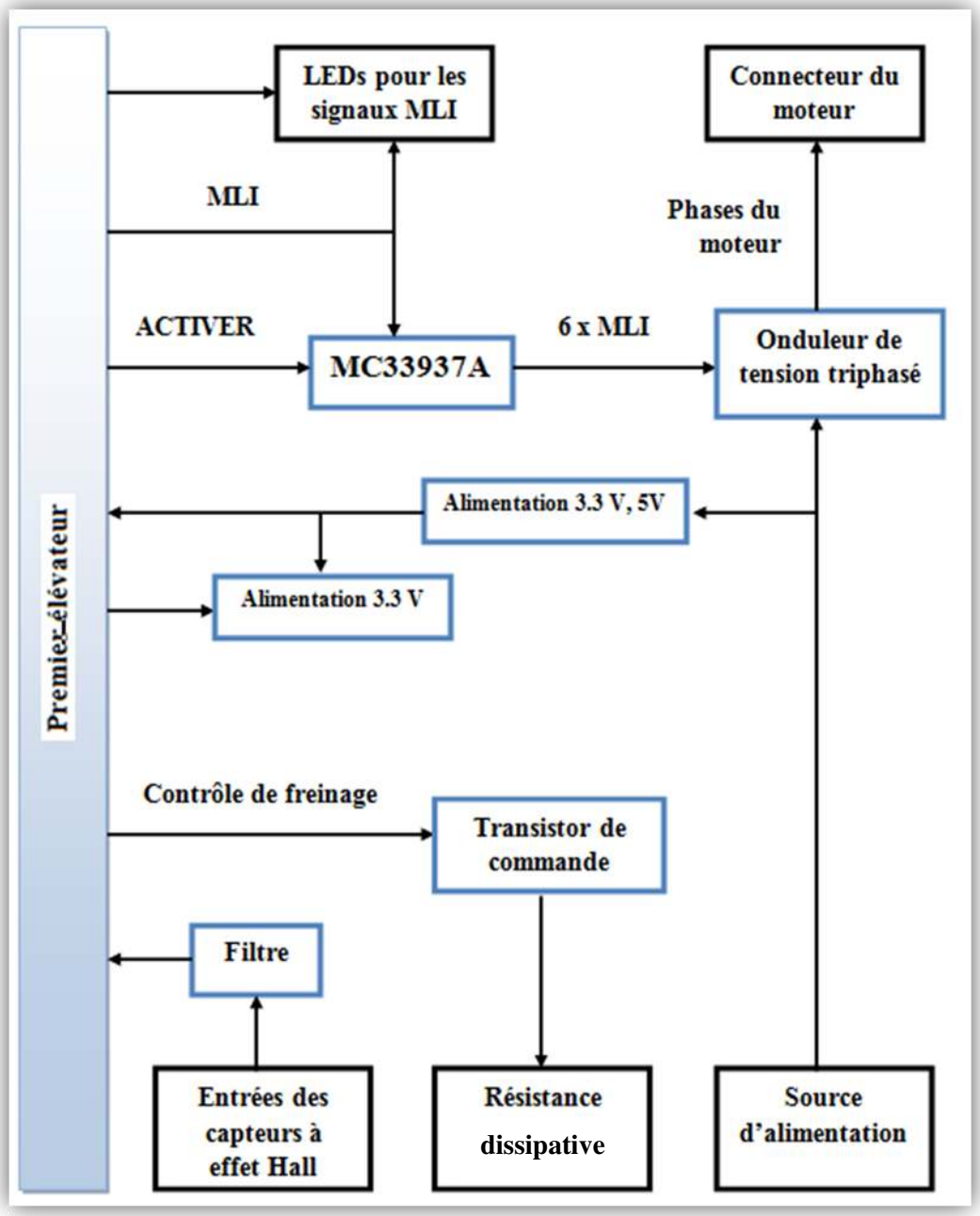


Figure 1.25 - Interaction entre les différents blocs de la carte de puissance

1.4.2.3 Carte série

Cette carte représentée par la figure 1.26 inclut des modules de connexion comme USB, CAN, Ethernet, et RS232. Ces modules sont combinés et utilisés avec le microcontrôleur et les différents périphériques des autres cartes à fin d'assurer la communication avec le BLDC.

Cette interconnexion est assurée par le deuxième élévateur.

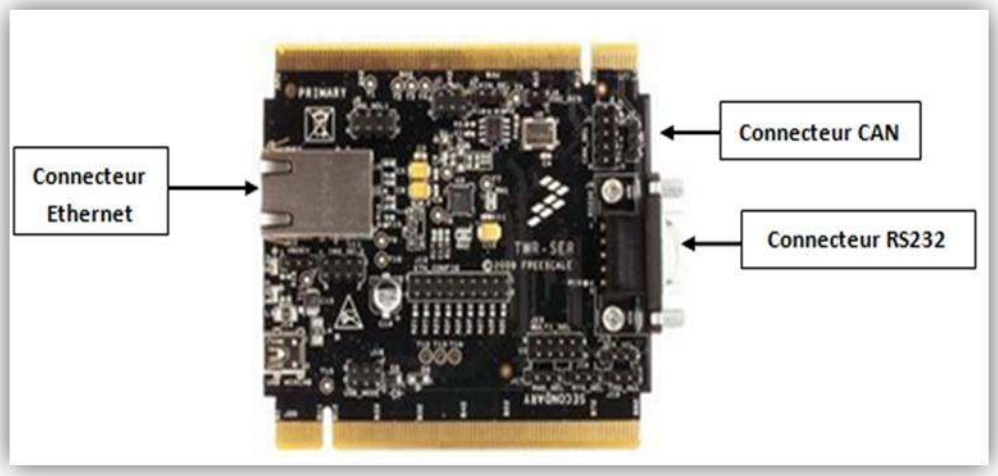


Figure 1.26 - Carte série TWR-SER

1.4.2.4 Outils logiciels de travail

Le développement de l'application du contrôle du moteur BLDC est réalisé autour de le même environnement de travail IAR utilisé pour le PMSM.

De plus, un autre outil logiciel est utilisé afin de commander le BLDC, c'est l'interface FreeMaster. Cet outil permet d'entrer les différents consignes de commande et les paramètres nécessaires pour le pilotage via une connexion RS232 avec la carte série.

1.5 Conclusion

Ce chapitre a été consacré à la présentation du cadre général du projet comprenant l'entreprise d'accueil et la problématique posée par le cahier des charges. Cette présentation est suivie d'une étude des moteurs sans balais à commander et de leurs environnements de travail. La connaissance des caractéristiques de brushless permet par la suite de choisir le type de commande adéquat à appliquer, et c'est la thématique du deuxième chapitre.

Chapitre 2

Spécification de la solution et réalisation de l'application

2.1 Introduction

Ce chapitre comporte deux parties dans lesquelles on va décrire brièvement le principe du choix des commandes de deux types de moteurs brushless : BLDC et PMSM pour passer par la suite à la conception et le développement des différentes étapes du pilotage selon l'environnement matériel et logiciel abordé dans le chapitre précédent.

Comme première partie de ce chapitre, on va construire deux modèles de commandes qui existent déjà et les appliquer sur le PMSM et le BLDC. La vérification de leur faisabilité sera conclue par la suite en exploitant les résultats de simulation sur l'environnement logiciel MATLAB Simulink.

Quant à la deuxième partie, elle présente les différentes étapes à suivre lors de la conception et le développement des applications de pilotage :

- La première application assure le pilotage par étapes du moteur sans balais à courant continu ou le BLDC.

- La deuxième application traite la commande vectorielle du moteur PMSM en se référant à une librairie de commande des moteurs PMSM fournie d'STMicroelectronics.

2.2 Modèles de commande des moteurs sans balais

Deux principaux types de commande existent pour le pilotage des machines synchrones et qui se différent selon le type du capteur de position utilisé et la machine à commander.

Nous avons exploité leurs principes et nous les avons appliqués pour les moteurs brushless.

2.2.1 Modèle de la commande six étapes pour le BLDC

Le principe de fonctionnement du moteur sans balais est basé sur le passage de flux d'excitation rotorique par six points fixes à 60° électriques. Le rotor tourne alors par intervalle de 60° et un tour complet correspond à six intervalles d'où le nom de la commande, six étapes.

La détection de chacun de ces points se fait par les capteurs à effet Hall. Ces capteurs génèrent des signaux carrés découpés en six secteurs.

Ces secteurs délimitent les six positions rotoriques qui sont utilisées pour la synchronisation de la commutation des transistors de l'onduleur de tension triphasé.

La figure 2.1 représente les six secteurs des signaux générés par les capteurs à effet Hall.

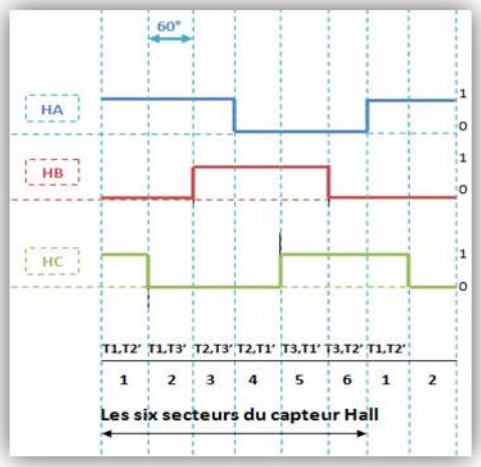


Figure 2.1 - Signaux carrés des capteurs à effet Hall

En se basant sur ce principe, nous avons construit le modèle de la commande d'un brushless BLDC en utilisant l'environnement logiciel Matlab Simulink comme le montre la figure 2.2.

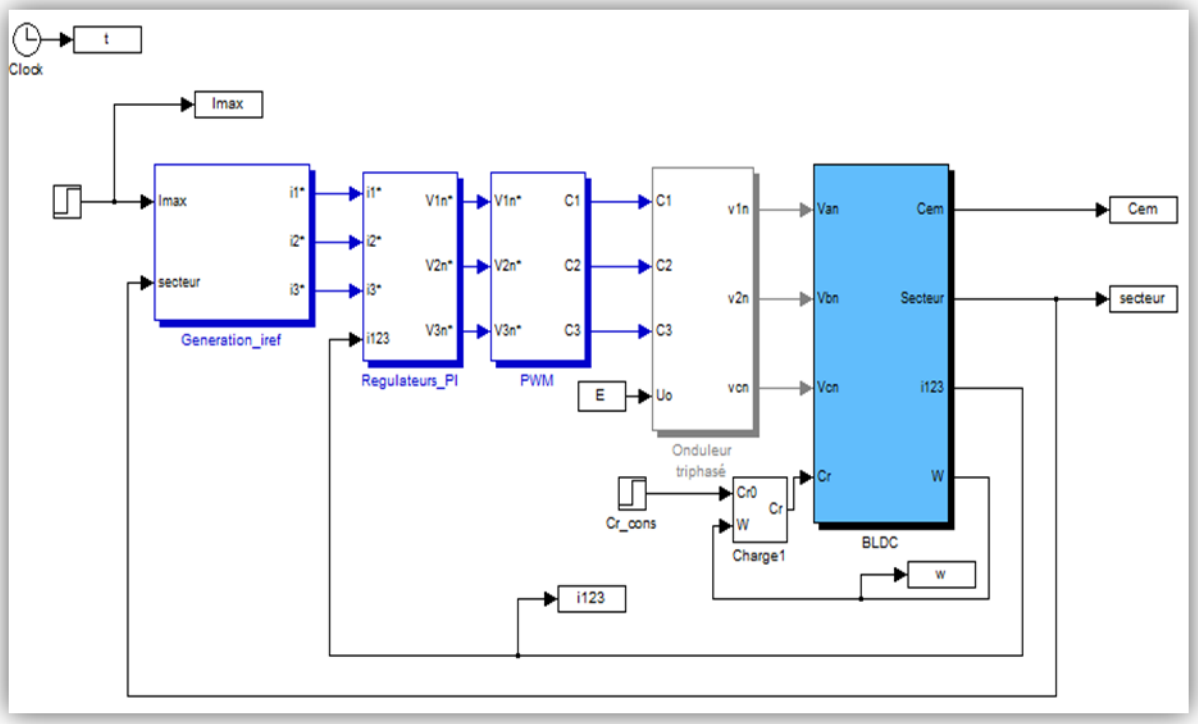


Figure 2.2 - Modèle de commande du BLDC

Comme est représenté sur la figure 2.2, une boucle d'asservissement basée sur des régulateurs PI est mise en œuvre afin d'agir sur le moteur pour maintenir une consigne préétablie. Cette consigne peut être un courant, un couple ou une vitesse.

Ce modèle est constitué de différents blocs :

- ***Bloc de génération des courants de référence*** : Ce bloc permet de générer les consignes triphasées i1*, i2* et i3* en créneaux, selon les secteurs des capteurs Hall, à partir d'une valeur de courant de référence maximale.
- ***Bloc de régulateurs PI :*** La régulation des courants des phases moteurs i1, i2 et i3 est réalisée par des régulateurs PI (Proportionnel Integral) qui génèrent par la suite les tensions de référence triphasées V1n*, V2n* et V3n*.
- ***Bloc PWM :*** Ce bloc représente la technique de Modulation de Largeur d'Impulsion (MLI) ou Pulse Width Modulation (PWM), dont l'objectif est d'imposer à chaque période de hachage une tension moyenne aux bornes de chaque phase du moteur égale à sa tension de référence. Pour ce faire, ce bloc calcule les signaux de commande C1, C2 et C3 de l'onduleur à partir des tensions de référence triphasées.
- ***L'onduleur triphasé :*** Il s'agit d'un convertisseur continu alternatif permettant d'imposer aux bornes du BLDC des tensions de phase via les commandes logiques C1, C2 et C3 délivrées par le bloc MLI. La figure 2.3 montre le schéma de principe de l'onduleur de tension alimentant le BLDC triphasé couplé en étoile, ce convertisseur est équipé de trois bras composés chacun de deux transistors de commutation et alimenté par une tension continue E.

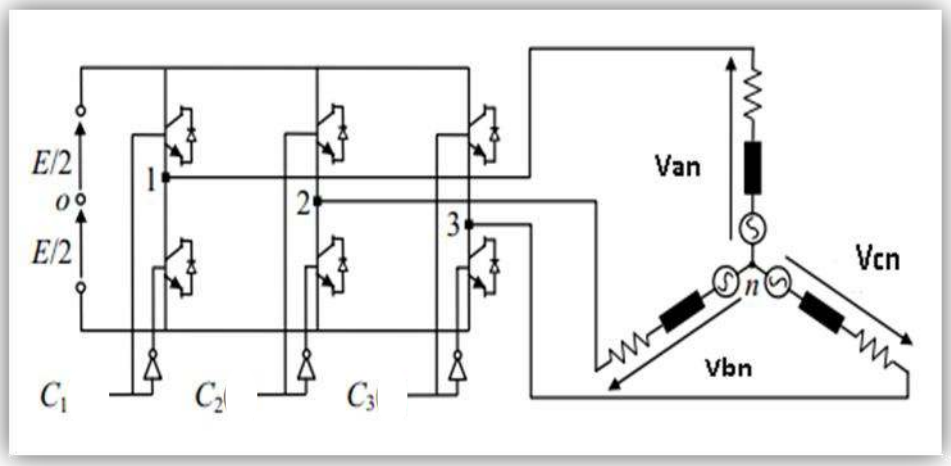


Figure 2.3 - Principe de l'onduleur de tension triphasé

***-Le BLDC* :** Ce bloc contient le modèle du moteur sans balais à FEM trapézoïdales qui est un système de premier ordre comme est représenté par la figure 2.4.

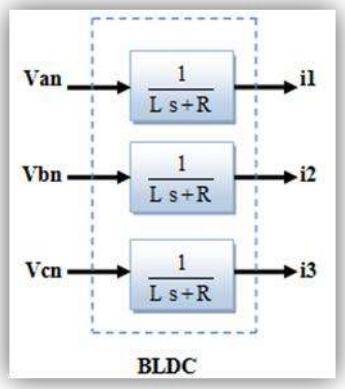


Figure 2.4 - Modèle du BLDC

Nous avons fait des simulations avec ce modèle de commande pour savoir son efficacité pour le pilotage d'un BLDC.

En premier lieu, nous avons visualisé les courants triphasés des phases du moteur, ils sont donnés par la figure 2.5.

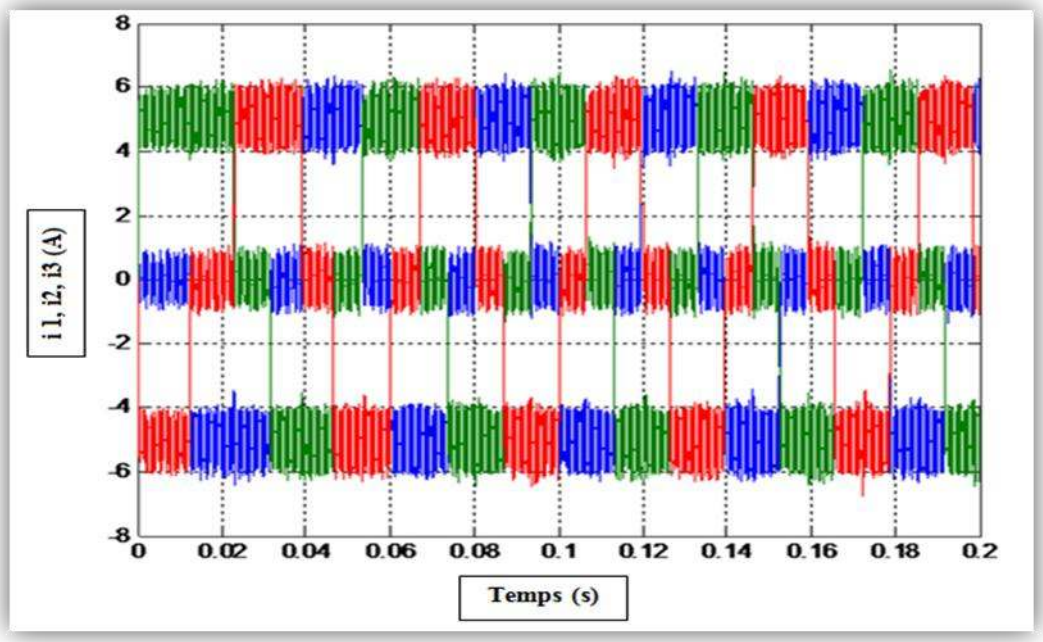


Figure 2.5 - Evolution des courants triphasés du BLDC

Le système obtenu est un système triphasé en créneaux de courants équilibrés.

La figure 2.6 montre très bien que le moteur donne une réponse en courant i1 (en bleu) qui suit la valeur de la consigne de référence i1* préétabli à Imax = 5A, (en vert)

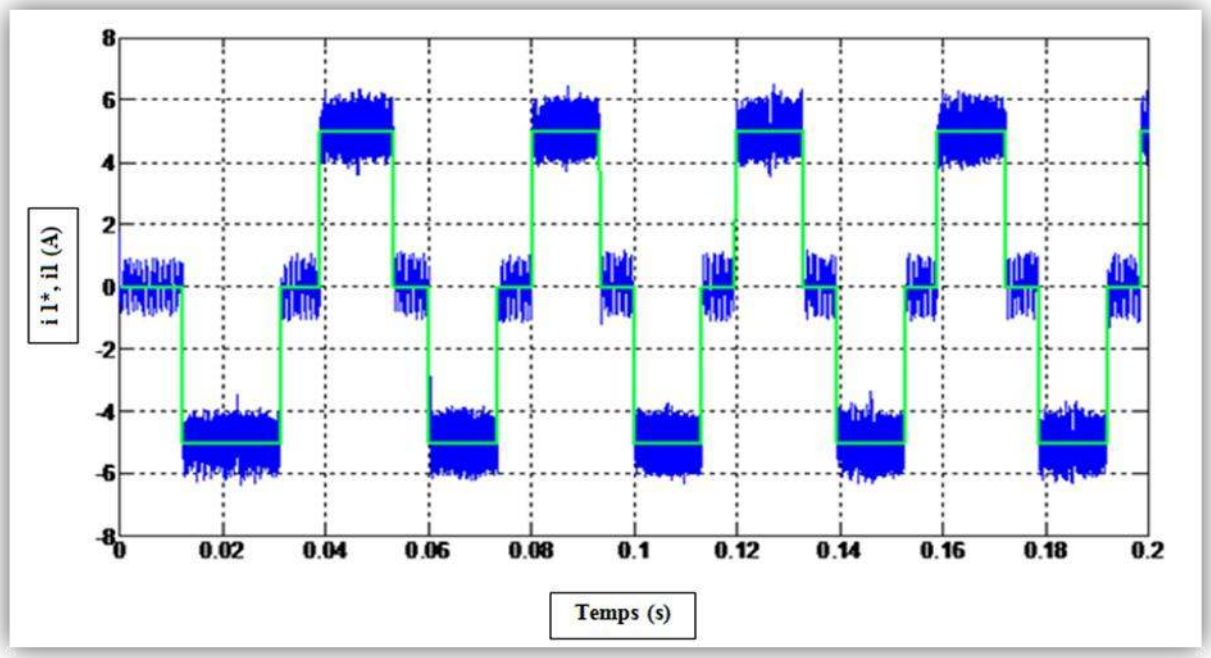


Figure 2.6 - Réponse en courant du BLDC à Imax = 5A

Le couple électromagnétique du BLDC garde aussi une valeur constante avec faible ondulation comme est montré à la figure 2.7.

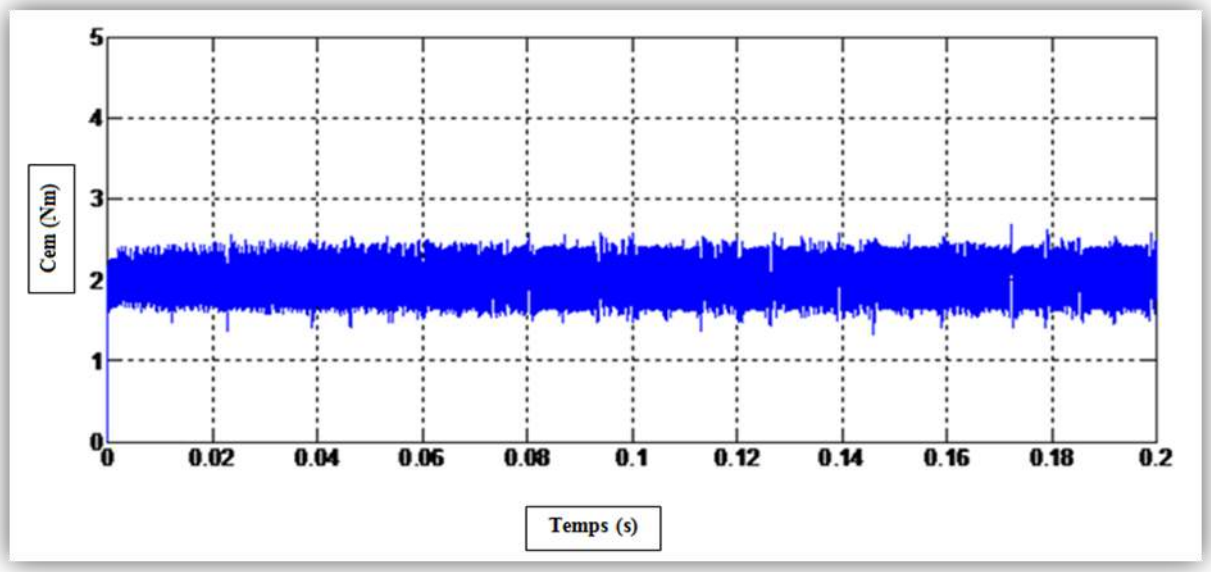


Figure 2.7 - Réponse en couple du BLDC

La réponse en vitesse est quasiment constante en régime permanant comme est montré à la figure 2.8.

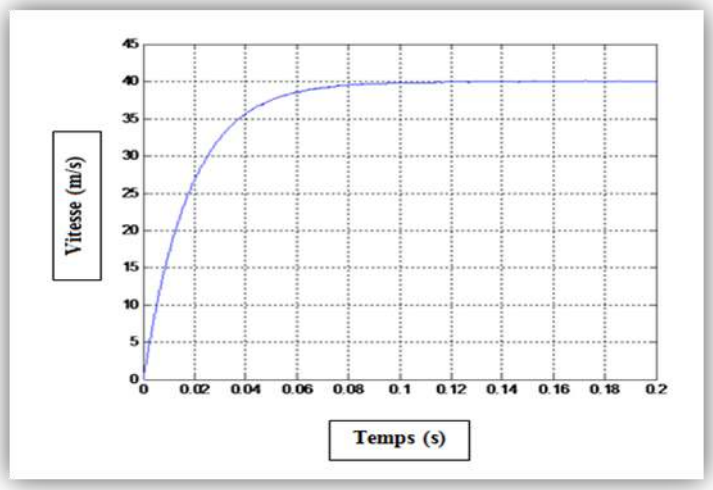


Figure 2.8 - Réponse en vitesse du BLDC

Les résultats de simulation montrent que ce modèle de commande à base des capteurs à effet Hall est très efficace pour un BLDC.

2.2.2 Modèle de la commande vectorielle pour le PMSM

En se basant sur le modèle du moteur synchrone à aimants permanents, nous avons construit les différents blocs de sa commande sur Matlab Simulink à base d'un onduleur de tension triphasé représenté par la figure 2.9.

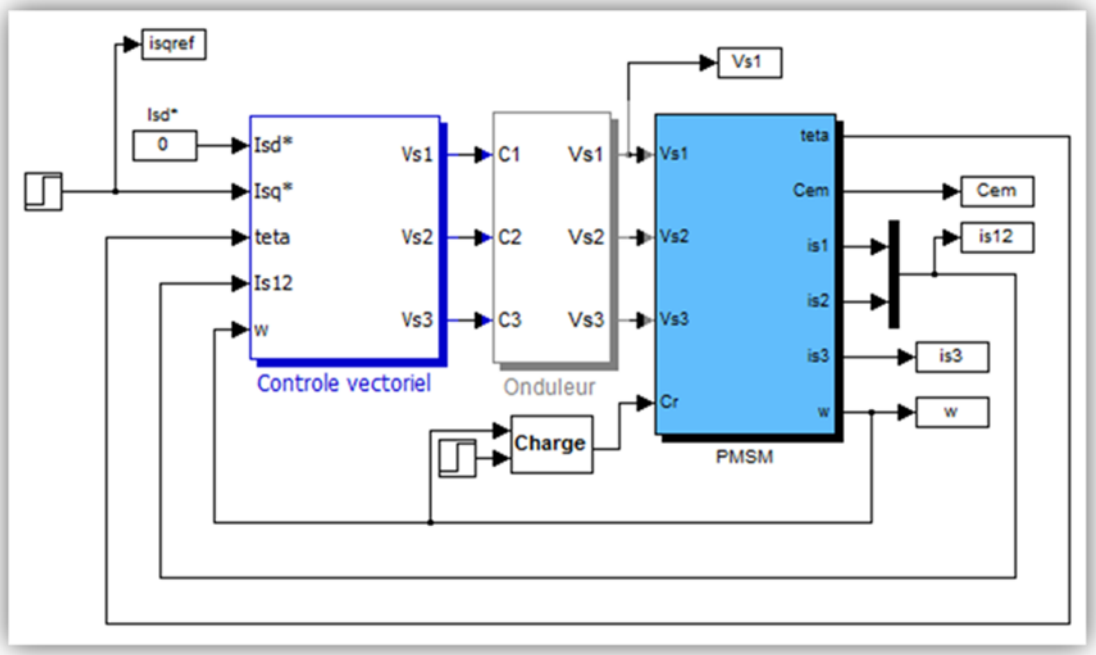


Figure 2.9 - Modèle de commande du PMSM

Le bloc de contrôle vectoriel est responsable de la régulation des courants dans le plan dq par des PI pour fournir par la suite les signaux de commande de l'onduleur de tension triphasé.

Ce bloc est formé par une partie pour les transformations Clarke et Park des courants mesurés i1 et i2 sur les deux phases statoriques , une partie pour leurs régulations, et un bloc PWM pour la génération des signaux de commande comme est décrit par la figure 2.10.

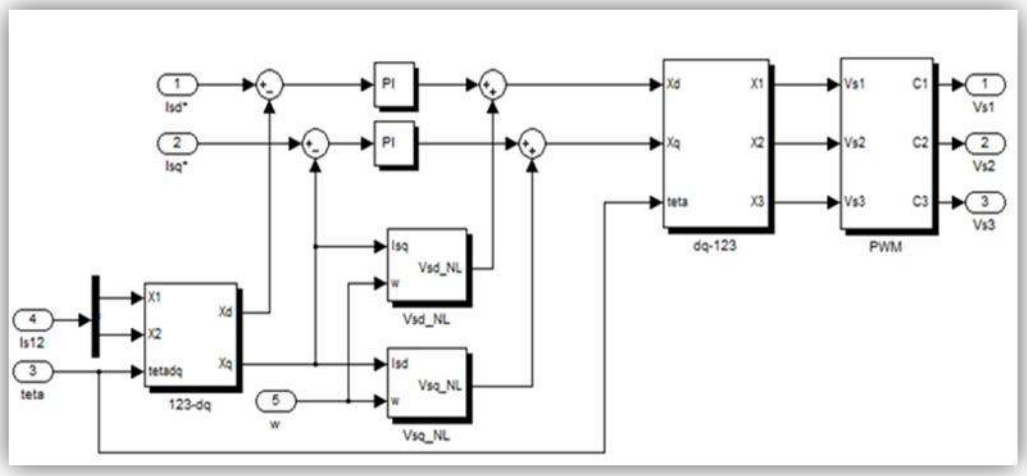


Figure 2.10 - Bloc du contrôle vectoriel

Nous avons alimenté le moteur par une valeur de référence de id* égale à 0A et iq* égale à 2A, nous avons obtenu des courants iq et id parfaitement continus comme est présenté dans la figure 2.11.

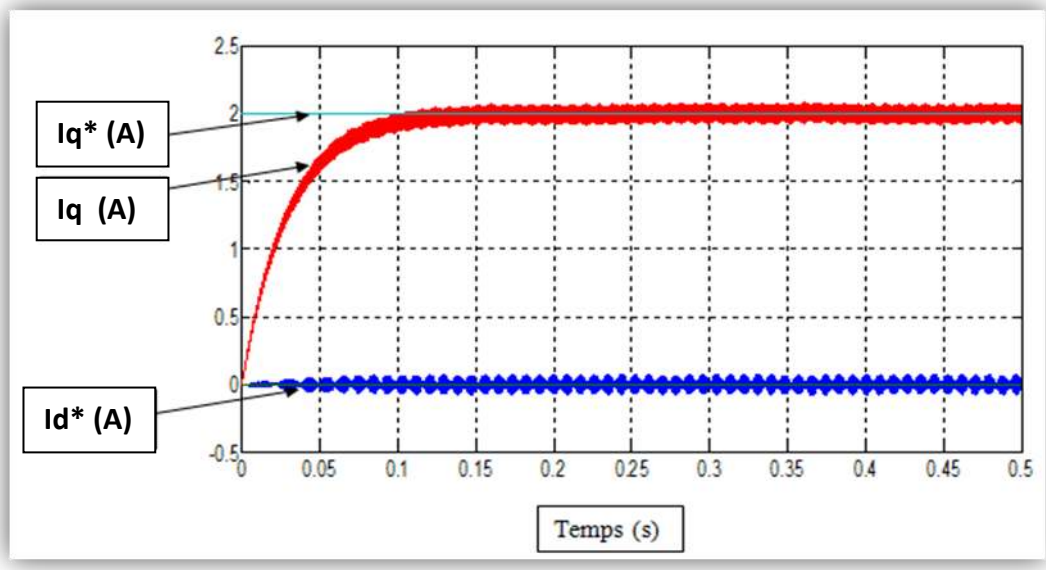


Figure 2.11 - Réponses en courants dans le plan dq du PMSM

Les courants statoriques sont parfaitement sinusoïdaux comme est montré à la figure 2.12.

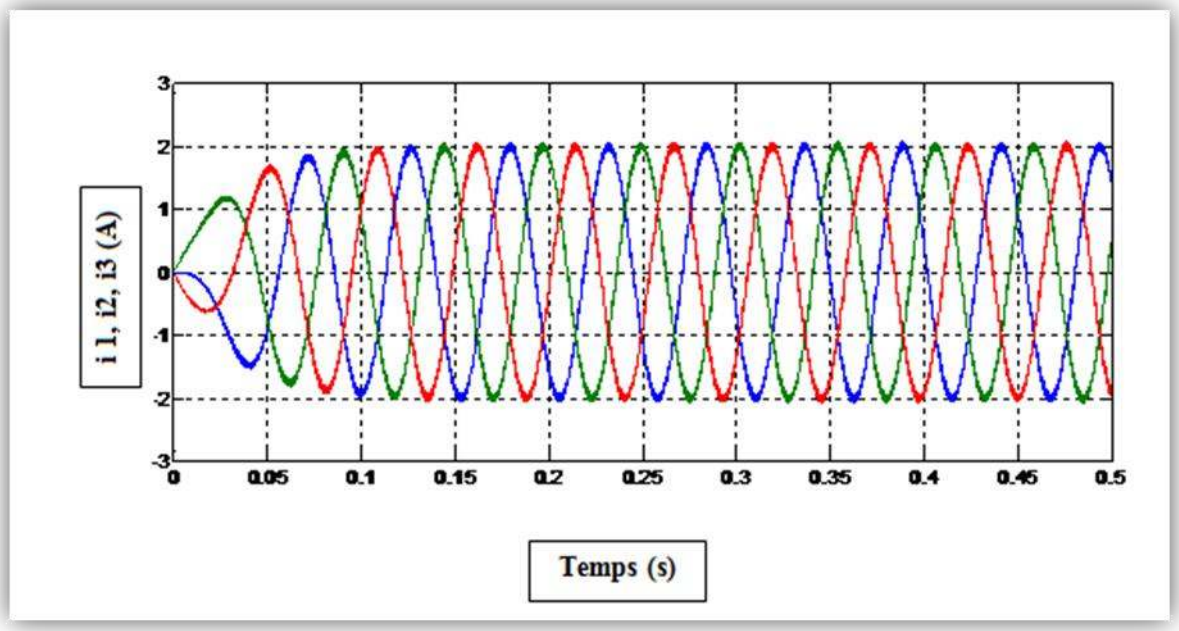


Figure 2.12 - Courants triphasés du PMSM

Le PMSM atteint une valeur constante du couple comme montre la figure 2.13. Il est à noter que les ondulations du couple sont relativement petites et elles dépendent essentiellement de la fréquence MLI utilisée.

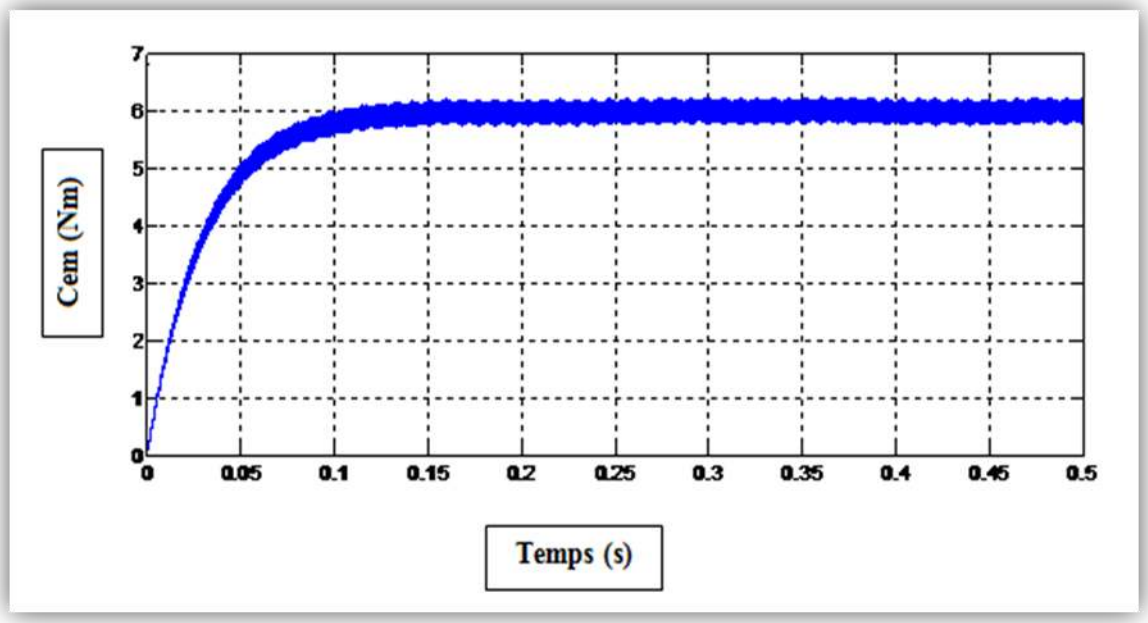


Figure 2.13 - Réponse en couple du PMSM

De plus, ce moteur atteint la vitesse de consigne pré-établie et garde une réponse constante.

D'un autre côté, la vitesse obtenue en régime permanent est quasiment constante. Les résultats de simulation obtenus prouvent donc le bon fonctionnement du contrôle vectoriel développé sur l'environnement logiciel Matlab-Simulink.

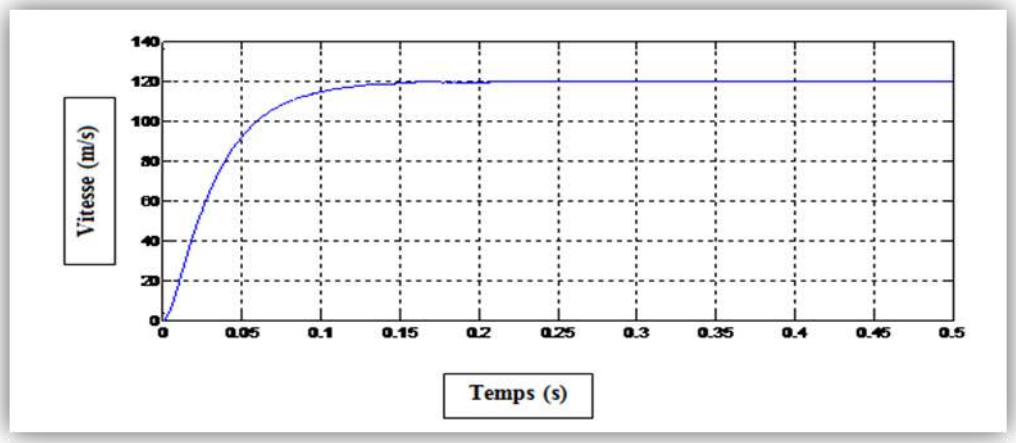


Figure 2.14 - Réponse en vitesse du PMSM

2.2.3 Modèle de la commande vectorielle pour le BLDC

Nous avons appliqué le même principe de la commande vectorielle sur le moteur BLDC. La figure 2.15 montre la réponse en courant du BLDC.

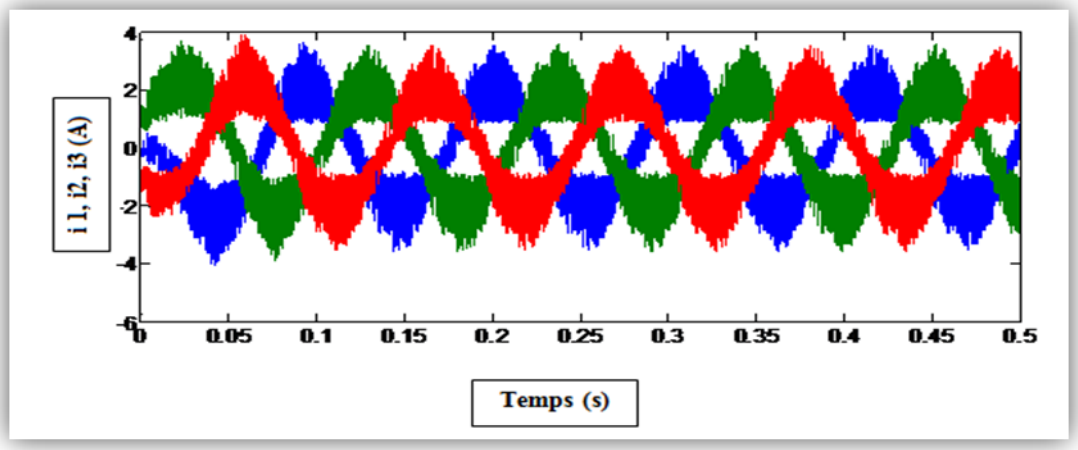


Figure 2.15 - Réponse en courant du BLDC commandé vectoriellement

La réponse obtenue n'est plus un système de courants en créneaux qui caractérisent le moteur BLDC et les ondulations sont fortement remarquables.

Le couple électromagnétique présente des ondulations remarquables durant le régime permanent comme est montré à la figure 2.16. Il en résulte un fonctionnement mal optimisé du BLDC.

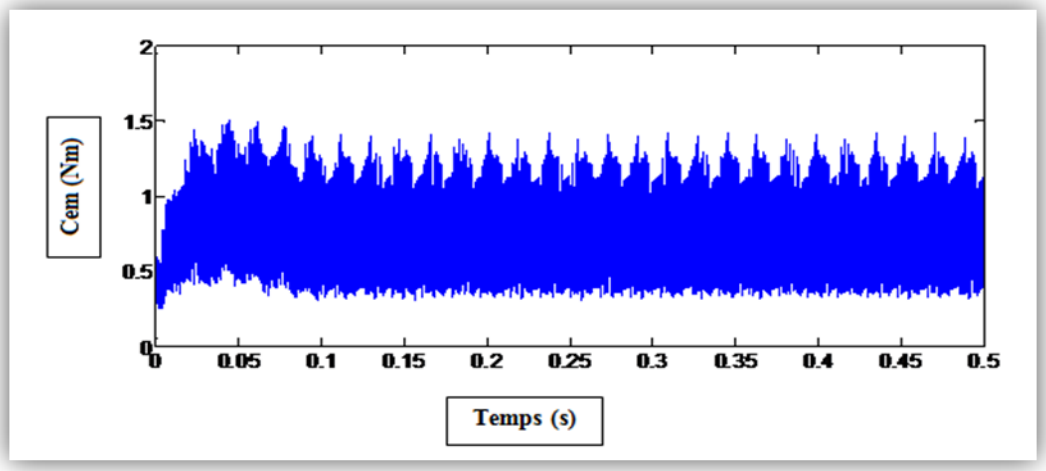


Figure 2.16 - Réponse en couple du BLDC commandé vectoriellement

D'un autre coté, la réponse en vitesse présente des vibrations importantes dues aux ondulations du couple comme est présenté à la figure 2.17.

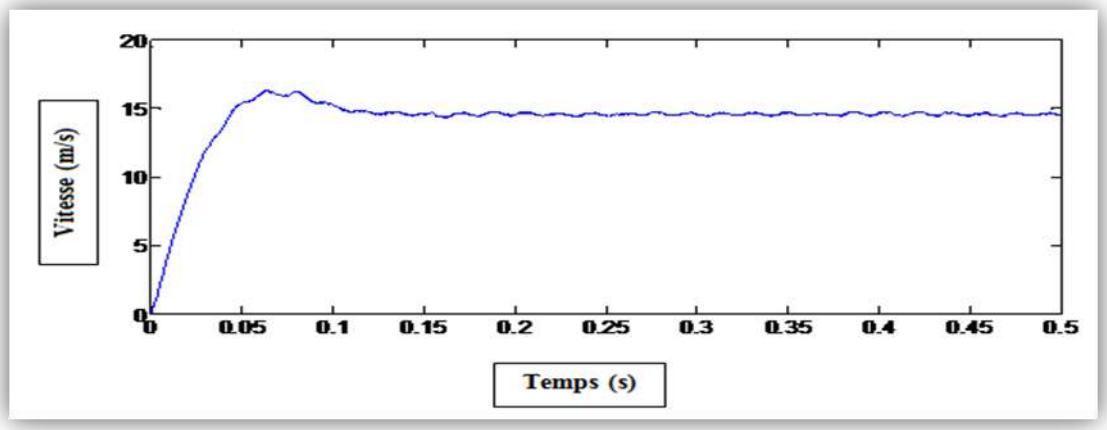


Figure 2.17 - Réponse en vitesse du BLDC commandé vectoriellement

On remarque que l'application du contrôle vectoriel sur un BLDC dégrade ses performances. Ceci est dû aux courants continues de ce moteur qui peuvent être régler directement par des PI sans avoir besoin de la commande vectorielle.

En effet, cette commande ne s'applique que sur des grandeurs parfaitement sinusoïdales et perd son efficacité lorsqu'elle contrôle des grandeurs continues.

2.3 Conception et implémentation de l'application du contrôle vectoriel du PMSM

L'application de commande du moteur PMSM est conçue grâce à une librairie fournie par la société STMicroelectronics. Cette librairie SDK offre différentes fonctions préprogrammées permettant de reconstruire le contrôle vectoriel avec différents types de commandes et en utilisant plusieurs techniques.

Nous avons choisi parmi ces fonctions et ces techniques seulement ceux qui répondent aux besoins du cahier des charges. Particulièrement, on a choisit la commande vectorielle d'un brushless synchrone à aimants permanents et à pôles lisses (SM-PMSM) en utilisant l'encodeur incrémental comme capteur de position et les résistances shunts comme capteurs de courant.

2.3.1 Présentation de la librairie de la commande vectorielle

2.3.1.1 Architecture de la librairie de la commande vectorielle

La libraire que nous avons utilisée pour concevoir l'application de la commande vectorielle nommée « STM32 PMSM FOC Firmware Library ». La figure 2.18 montre l'architecture de cette librairie organisée du bas vers le haut

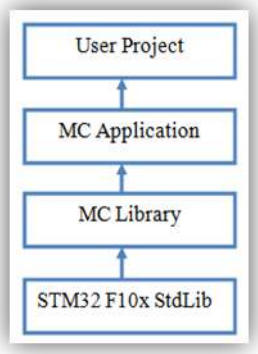


Figure 2.18 - Architecture de la librairie de la commande vectorielle

Cette librairie est composée de quatre couches bien structurées ayant chacune des fonctionnalités précises.

Nous avons tout d'abord appréhendé la structure de ces couches, leurs compositions et nous avons par la suite exploité leurs différentes fonctions pour réaliser notre application.

-La couche STM32F10x STdLib : C'est une couche d'abstraction matérielle pour la série des microcontrôleurs basé sur le Cortex-M, elle contient les pilotes des périphériques de la famille STM32F10x, tels que les GPIO, Timers, USART,…ainsi qu'une interface logicielle standard (CMSIS) permettant la communication avec ces périphériques et les systèmes temps réel.

-La couche MC Library: Cette couche est composée de 37 classes qui représentent tous les éléments intervenant dans le contrôle du moteur PMSM tels que les capteurs, les transformations vectorielles et les blocs de protection. Cette librairie utilise la couche STM32F10x STdLib pour la configuration des périphériques qu'elle utilise.

-La couche MC Application: C'est l'application du contrôle du moteur qui utilise la couche de contrôle du moteur afin de répondre aux commandes de l'utilisateur.

-La couche User Project: C'et la couche applicative de l'utilisateur qui permet l'accès à l'application de pilotage du moteur à travers un ensemble de commande à des instants précis via des interfaces de communication tels que l'afficheur LCD.

2.3.1.2 Organisation de la librairie de la commande vectorielle

La librairie de ST contient un ensemble de fichiers liés entre eux d'une façon bien étudiée.

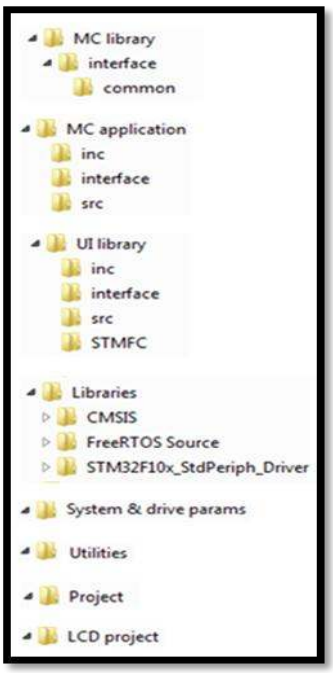


Figure 2.19 – Organisation de la librairie de la commande vectorielle

Dans ce qui suit, on décrit le contenu des répertoires liés à la librairie de la commande vectorielle.

Table 2.1 - Description des répertoires de la librairie de la commande vectorielle

Dossier	Fichier	Description
MC Library		Fichier source de la couche MC Library
	Interface	Ce fichier contient la déclaration des différentes méthodes de toutes les classes de la couche MC Library
	Common	Ce fichier contient la déclaration des fonctions communes (PI, Transformations Park et Clarke..)
MC Application		Fichier source de la couche MC Application
	Inc	Ce fichier contient les structures de données des classes
	Interface	Ce fichier contient la déclaration des différentes méthodes des classes de la couche MC Application
	Src	Ce répertoire contient les fichiers sources ou sera implémentée l'application de commande.
UI Library		Fichier source de la couche utilisateur
	Inc	Ce fichier contient les structures de données des classes
	Interface	Ce fichier contient la déclaration des différentes méthodes des classes de la couche utilisateur
	Src	Ce répertoire contient les fichiers sources
	STMFC	C'est la librairie liée à l'afficheur LCD
Librairies		
	CMSIS	C'est une interface logicielle standard du microcontrôleur CortexM
	STMF10x_StdPeriph_Driver	C'est la librairie des pilotes des périphériques de l'STMF10x
System &Drive **Params**		Ce répertoire contient les différents fichiers des paramètres nécessaires à la commande
Utilities		Ce répertoire contient le code de configuration des différents blocs de la carte de commande
Project		Ce répertoire contient le projet de l'utilisateur
LCD Project		Ce répertoire contient les fichiers sources du programme de l'afficheur LCD

2.3.2 Etapes de conception de l'application de la commande vectorielle

L'application de pilotage du PMSM est réalisée en se basant sur les différentes couches, le principe de la commande vectorielle est expliqué par la figure 2.20.

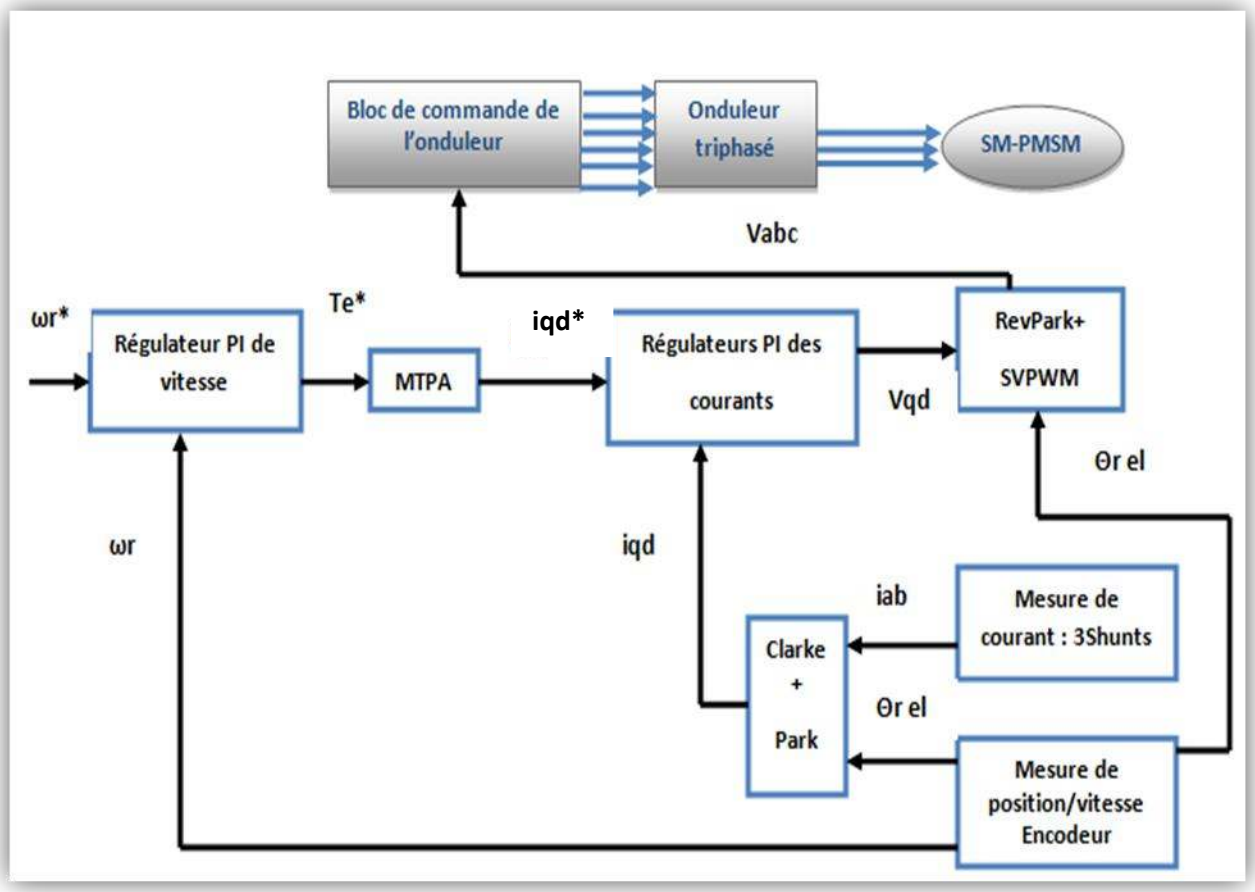
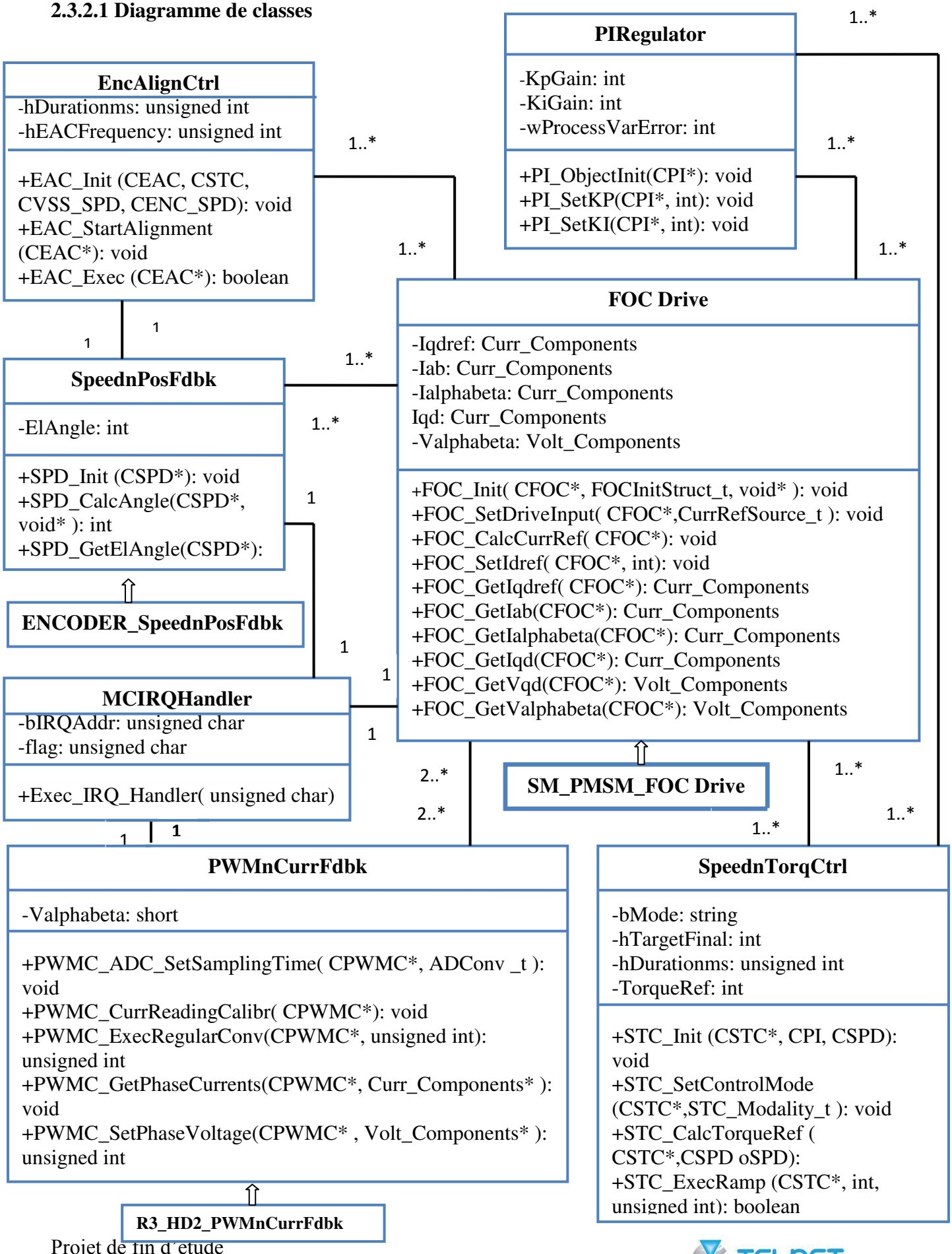


Figure 2.20 - Principe de la commande vectorielle

Comme première étape de la conception de l'application, nous avons sélectionné les classes et les méthodes nécessaires qui s'interviennent dans la commande vectorielle, et nous avons reconstruis ainsi le diagramme des classes.

Comme deuxième étape, nous avons établi l'interaction entre ces différentes classes pour réaliser l'application de contrôle après une analyse détaillée de leurs fonctions préprogrammées.

2.3.2.1 Diagramme de classes



Nous avons reconstrui ce diagramme de classes à partir les 37 classes contenues dans la couche « MC Library », nous avons choisi seulement celles qui vont intervenir dans la commande du PMSM à poles lisses et nous avons élaborés les relations qui existent entre elles.

SpeednTorq Ctrl : C'est la classe qui traite le mode de contrôle. En effet, on peut controler le SM-PMSM soit en courant (en couple) et soit en vitesse.

PI Regulator : C'est la classe de régulateur PI utilisée pour la régulation des courants et de la vitesse selon le type du contrôle choisi.

EncAlignCtrl : C'est la classe du contrôle de l'alignement de l'encodeur et qui vérifie le résultat du calage sur l'axe du rotor.

SpeednPosFdbk : C'est la classe de capteur de position, dans notre cas, nous avons utilisé l'encodeur incrémental dont Encoder_SpeednPosFdbk est la classe dérivée.

MCIRQHandler : C'est la classe des interruptions qui interviennent lors du contrôle moteur.

PWMnCurrFdbk : C'est la classe de génération des signaux PWM pour la conduction de l'onduleur triphasé et elle contient aussi le capteur de courant, dans notre cas ce sont les résistances shunts qui sont représentées par la classe fille R3_HD2_PWMnCurrFdbk.

FOC Drive : Les méthodes de cette classe traitent les différentes étapes du pilotage du brushless basés sur les transformations vectorielles (Park et Clarke), le moteur à commander est un PMSM à pôles lisses d'où le choix de la classe dérivée SM-PMSM.

2.3.2.2. Interactions entre les classes pour la commande du moteur

Il existe différents modes de contrôle du PMSM à pôles lisses, soit on lui commande en couple, soit on lui fait entrer une rampe de vitesse.

L'avantage d'exécuter une rampe de couple ou de vitesse au lieu de donner directement une valeur exacte est d'avoir une bonne qualité démarrage du moteur et limiter le fort appel de courant en atteignant la valeur souhaitée à la fin de la rampe.

- **Contrôle en couple**

C'est le premier type du contrôle, la figure 2.21 explique la communication entre les différentes classes responsables de cette commande contenues dans la première couche.

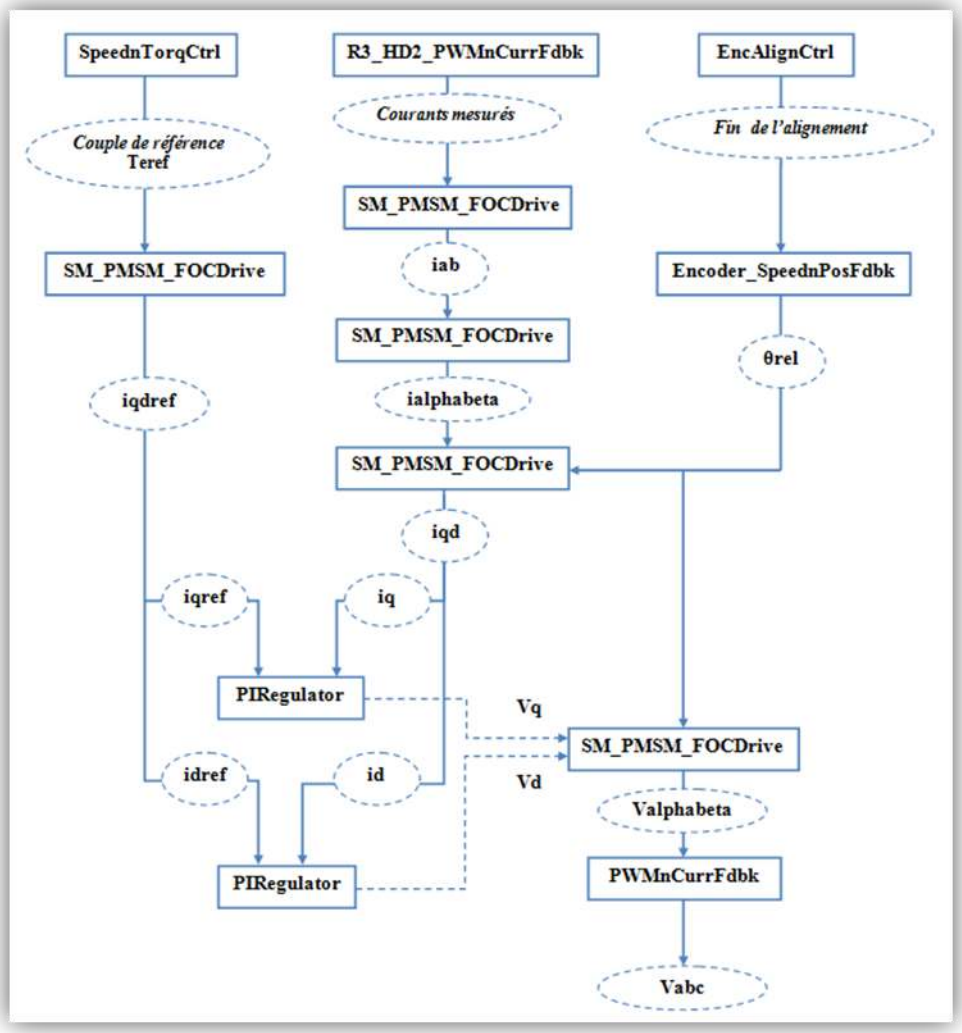


Figure 2.21 - Commande en couple du SM-PMSM

Les étapes de pilotage en couple sont :

- Génération du couple de référence

Nous avons sélectionné les classes et les fonctions adéquates qui vont intervenir dans la génération du couple de référence. Les méthodes de « SpeednTorqCtrl » vont générer le couple de référence utilisé par la suite dans la classe « SM_PMSM_FOCDrive » afin de calculer la valeur des courants vectoriels de références.

De plus, dans le cas d'un moteur PMSM à pôles lisses, la valeur du couple dépend seulement de la composante en quadrature du courant comme est expliqué à la relation (1.8), c'est pourquoi il faut annuler la composante directe.

- Alignement de l'encodeur

Le capteur est couplé aléatoirement avec le rotor et donc son zéro n'est pas forcément calé sur le vrai zéro rotorique. La position indiquée par le capteur est donc probablement différente de la position réelle du rotor, c'est pourquoi, il faut aligner correctement l'encodeur sur l'axe en ajoutant un angle de compensation dans l'algorithme de commande.

Des méthodes précises de la classe « EncAlignCtrl » permettent d'ajouter cet angle d'offset et d'exécuter l'alignement pour calculer par la suite l'angle et la vitesse électrique du rotor par des fonctions préprogrammées de la classe « Encoder_SpeednPosFdbk **».**

- La lecture et la transformation des courants

La lecture numérique des courants statoriques est assurée par des résistances shunts représentées par la classe « R3_HD2_PWMnCurrFdbk », il faut tout d'abord fixer le temps d'échantillonnage des convertisseurs analogique numériques choisies pour la lecture, régler par la suite leur tension d'offset puis détecter les valeurs des courants .

Ces valeurs seront envoyés par la suite vers la classe « SM_PMSM_FOCDrive » pour les transformer en leurs appliquant Park et Clarke grâce à des fonctions préprogrammées spécifiques.

- La régulation des courants

Malgré que nous avons entré la valeur désirée du couple, et par la suite les valeurs de courants de références, le moteur peut délivrer des valeurs différentes de celles souhaitées à cause de la construction interne de la machine, c'est pourquoi, une boucle d'asservissement doit s'intervenir. Des régulateurs PI vont régler les courants dans les phases statoriques. On trouve un PI pour chaque composante de courant : PI_ iq et PI_id dont le principe est basé sur le calcul de l'erreur entre la consigne et la référence, la réduire puis l'annuler.

- La génération des signaux de commande

Après la régulation des courants, les PI génèrent les tensions vectorielles qui seront analysées par les transformations inverses dans la classe « SM_PMSM_FOCDrive », la tension générée Valphabeta à partir du Park inverse est utilisée par la suite pour calculer les tensions de commande qui seront envoyées vers la carte de puissance pour commander l'onduleur.

- **Contrôle en vitesse**

C'est le deuxième type de commande du moteur, on fait entrer une consigne de vitesse.

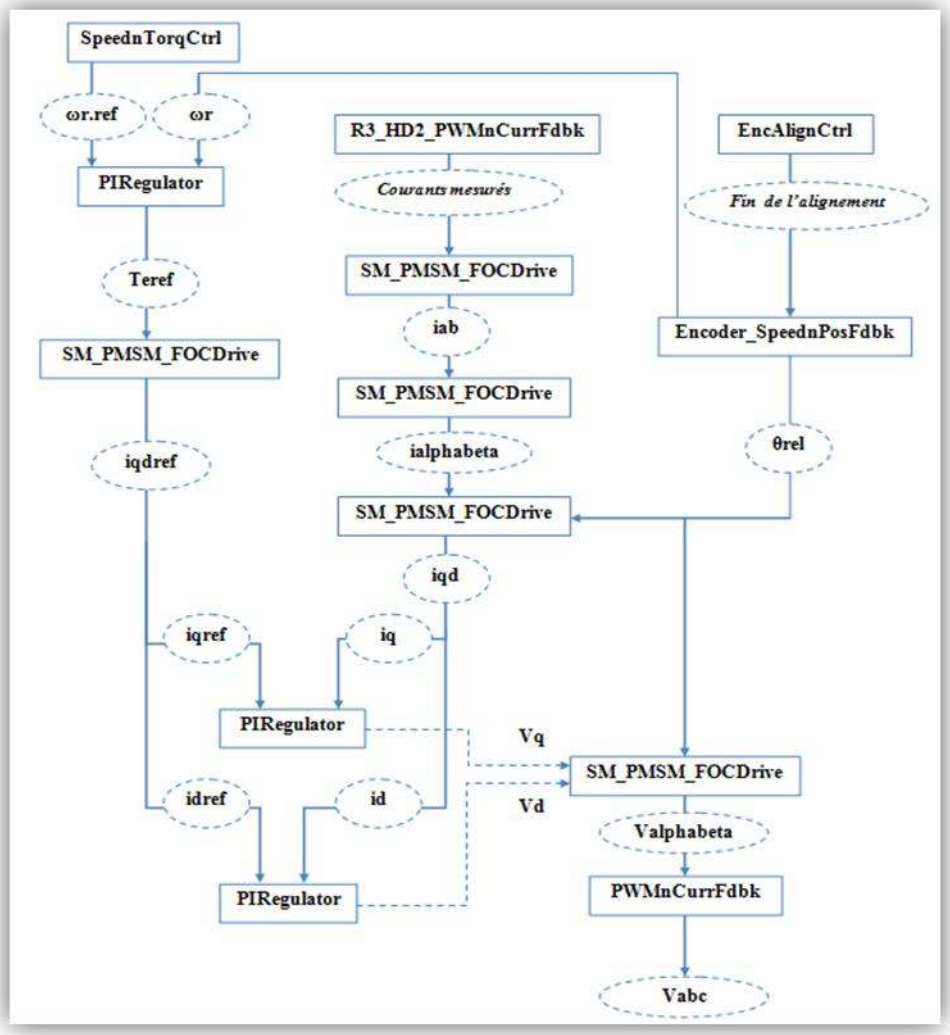


Figure 2.22 - Commande en vitesse du SM-PMSM

Les étapes de contrôle en vitesse sont les mêmes que les contrôles en couple, sauf une étape s'ajoute en amont, il s'agit de la génération de vitesse et sa régulation.

- Génération de la vitesse de référence

Une rampe de vitesse sera exécutée, et en se basant sur le même principe, la vitesse élaborée sur l'axe du moteur est différente de celle désirée, d'où l'intervention d'une boucle d'asservissement basée sur un PI qui va générer après la régulation, le couple de référence nécessaire pour terminer la procédure de commande qui est basée sur les mêmes étapes du contrôle en couple. Ces différents modes de contrôle du moteur exploitent les fonctions préprogrammées des différentes classes et selon un séquencement des états précises.

2.3.3 Implémentation de l'application de contrôle du moteur

L'implémentation de l'application du contrôle moteur est effectuée dans la deuxième couche, « MC Application », selon une architecture bien déterminée.

2.3.3.1 Machine d'état de la commande du moteur

Le moteur passe par plusieurs états lors de son fonctionnement dès son démarrage jusqu'à son arrêt comme est montré à la figure 2.23.

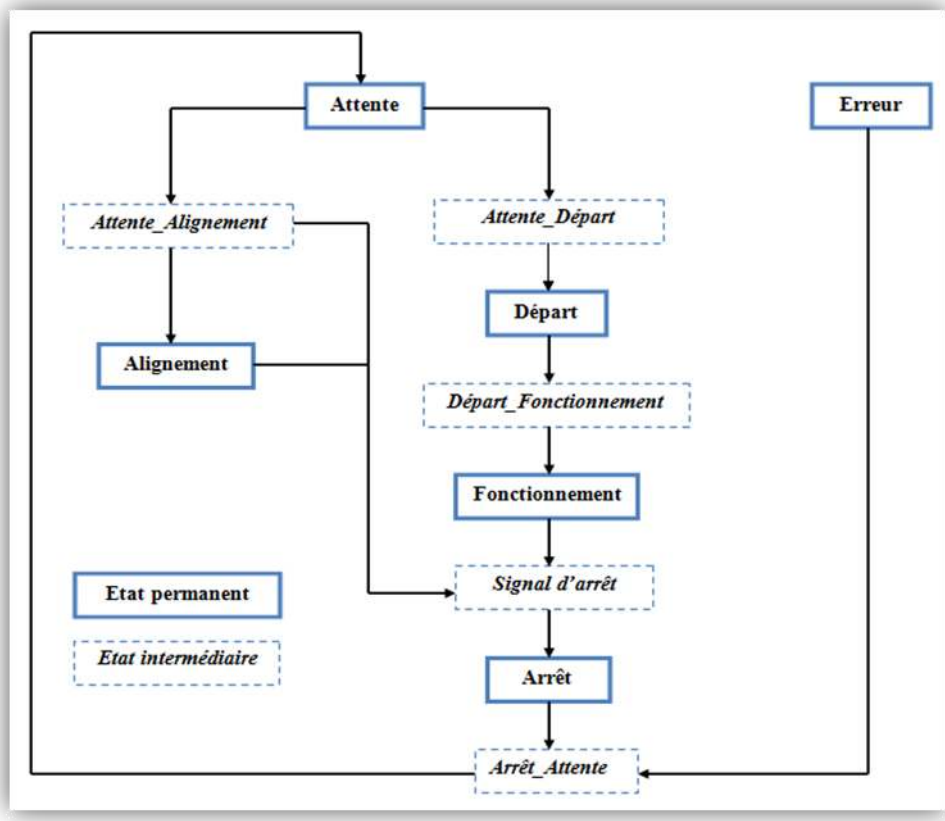


Figure 2.23 - Machine d'état du SM-PMSM

2.3.3.2 Les tâches principales de la commande

L'application de contrôle du moteur est organisée selon des différentes tâches qui sont reliées aux propriétés intrinsèques de la machine.

En effet, la constante de temps mécanique du moteur donnée par la relation **(1.12)** reste inférieure aux constantes de temps électriques exprimées dans les relations **(1.13)** et **(1.14)**.

De plus, la constante de temps mécanique impose la dynamique de la boucle de régulation de la vitesse alors que les constantes de temps électriques s'interviennent dans la régulation de courant comme est expliqué en détail dans [10]. De ce fait, la régulation des courants sera effectuée à une fréquence plus élevée que celle de la régulation de vitesse.

En se basant sur ce principe, l'application de la commande vectorielle du moteur SM-PMSM est composée de cinq tâches.

- **La tâche haute fréquence :** Elle traite la régulation des courants et le contrôle en couple à une fréquence égale à la fréquence MLI, cette tache est exécutée à chaque période d'échantillonnage.
- **La tâche moyenne fréquence :** L'exécution de la régulation de vitesse est réalisée à une fréquence égale à 500Hz.
- **La tâche basse fréquence :** Elle s'occupe des fonctions qui ne nécessitent pas un temps précis tel que le chargement des capacités de démarrage du moteur avec une fréquence égale à 100Hz.
- **La tâche de sécurité :** Cette tâche tient compte des actions à prendre lors d'un défaut déclenché dans la maquette de commande (surtension,..) telles que la désactivation de la génération d'MLI. Elle est exécutée à une fréquence égale à 2KHz.
- **La tâche de l'interface utilisateur :** Elle est exécutée à chaque 100ms et permet la gestion de l'afficheur LCD.

Les paramètres nécessaires pour l'implémentation de l'application du contrôle de moteur sont générés par le GUI d'STMicroelectronics. Ces paramètres constituent les informations techniques du moteur et du capteur, ainsi les différents outils de contrôle utilisés.

Le remplissage de ces paramètres grâce au GUI de ST permet de générer quatre fichiers « .h » utilisés pour mettre à jour l'application de commande du moteur.

Ces fichiers contiennent bien évidemment les paramètres de chaque étage, ce sont :

PMSM motor parameter.h

Contient les paramètres du moteur, tels que le type du moteur, le nombre de paire de pôles, les valeurs de la résistance et de l'inductance statoriques, la vitesse nominale, les paramètres de l'encodeur incrémental.

Control stage parameters.h

Contient les paramètres de la carte de commande, comme la fréquence d'échantillonnage de l'MLI, le mode de contrôle (vitesse ou couple), les valeurs des gains des régulateurs PI, ainsi que le mode de commande de la carte, dans notre cas, on va utiliser l'écran LCD.

Power stage parameters.h

Contient les paramètres de l'étage de puissance. En effet, on choisit les différents blocs disponibles dans la carte de puissance tels que le bloc de frein dissipatif et le bloc de protection contre la surchauffe, et on fait entrer leurs paramètres tels que la valeur des shunts et la fréquence de commutation des interrupteurs.

Drive parameters.h

Contient des différents paramètres de commande. On choisie les canaux de lecture de courant, et les canaux des différents périphériques ainsi la famille du microcontrôleur utilisé dans le pilotage et c'est dans le but de configurer après la carte de commande à utiliser.

2.3.3.3 Gestion et validation de l'application de la commande du moteur

L'utilisateur peut gérer l'application en faisant appel directement à un ensemble de commande à partir du projet de l'utilisateur. Ces commandes sont appelées MC API.

Ces API sont devisées en deux grandes familles :

- ***MC Interface***

La première famille contient les commandes de niveau hiérarchique plus haut tel que le démarrage ou l'arrêt du moteur, l'alignement de l'encodeur, l'exécution des rampes de vitesse ou de couple.

- ***MC Tuning***

C'est la deuxième famille de commandes permettant de lire et de modifier les informations liées aux objets de l'application, tels que les régulateurs PI, le capteur de position, les courants et les tensions des transformations vectoriels.

Ces deux familles sont représentées par deux objets oMCI et oMCT qui sont deux tableaux ayant la taille égale au nombre de moteurs à commander. Ces deux objets permettent d'accéder aux différents paramètres de l'application tels que les gains proportionnel et intégral, les tensions et les courants de la transformation vectorielle.

Tout d'abord, on appelle les deux représentants des deux familles de commandes, oMCI et oMCT, ceci va permettre de créer les objets adéquats avec les paramètres de configuration du système saisis par le GUI d'STMicroelectronics.

Ensuite, les différentes taches responsables à l'application de commande du moteur sont mises à jour afin de répondre aux commandes utilisateurs.

Les résultats de ces commandes peuvent être affichés sur l'écran LCD dont le programme est déjà près dans le répertoire LCD Project et qui doit être flashé correctement avec l'application de contrôle du moteur. La figure 2.24 résume le processus qui se déroule entre les différentes couches dans le but de répondre aux commandes de l'utilisateur.

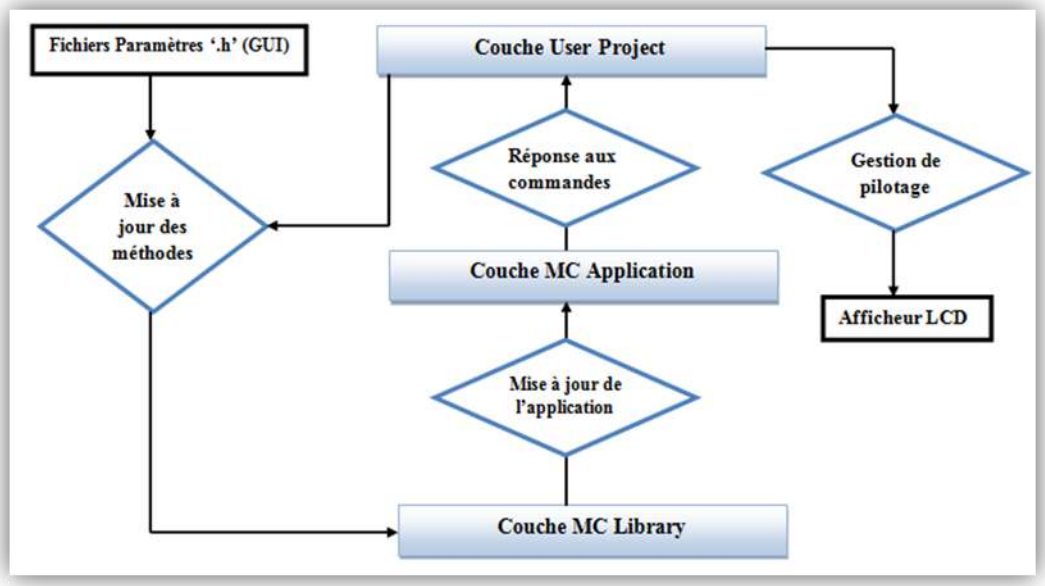


Figure 2.24 – Scénario de la commande

Nous avons construit les commandes utilisateurs selon les différents types de contrôle, soit en couple ou en vitesse. Les différentes étapes de commande sont représentées à la figure 2.25.

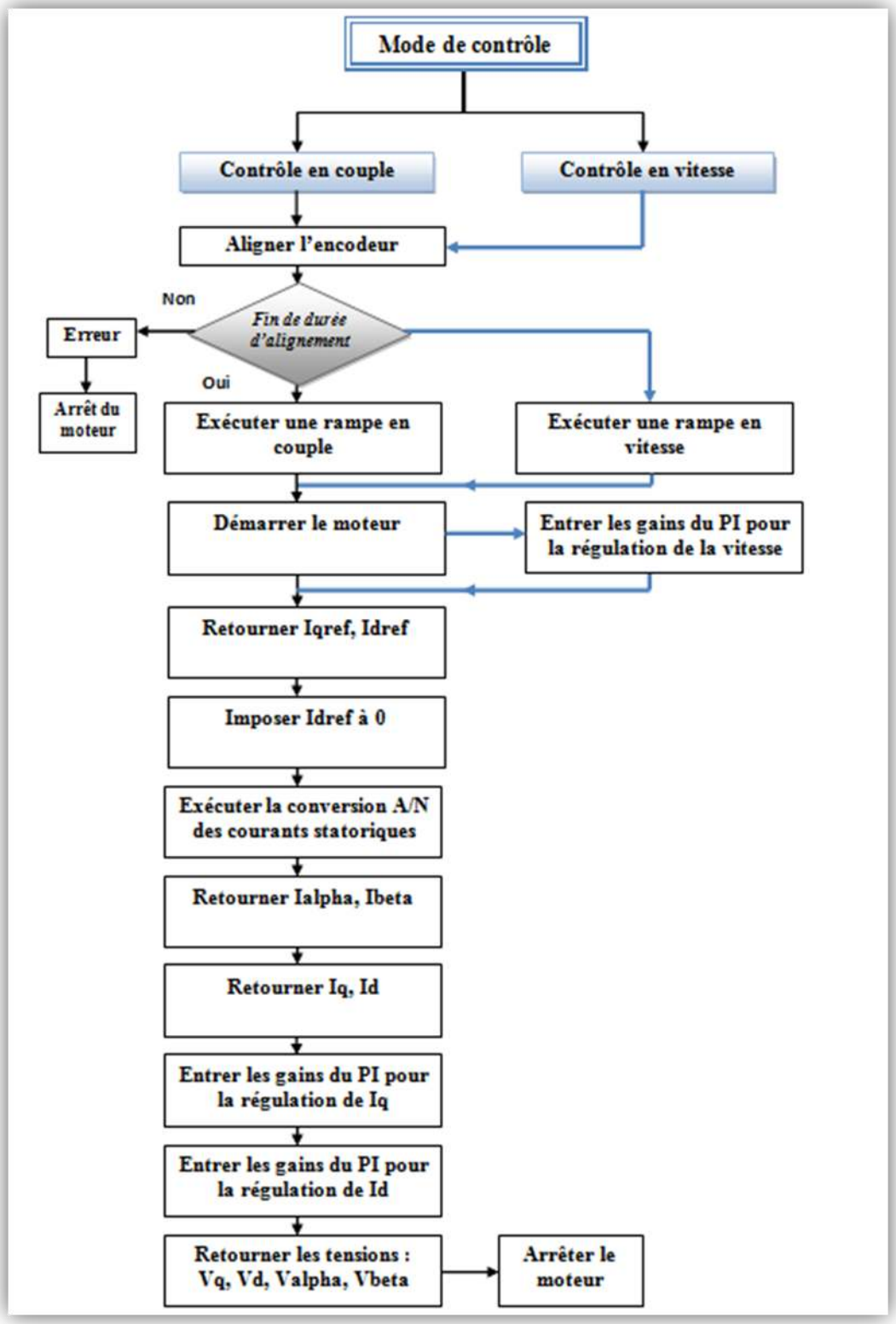


Figure 2.25 - Différents ordres de commande du moteur

Ces commandes vont interagir avec l'application du moteur et ne peuvent être exécutées que si le matériel est bien installé, c'est-à-dire si la carte de commande est connectée correctement à la carte de puissance et au moteur SM-PMSM à commander.

Nous avons flashé tout le projet de la commande vectorielle dans la carte de commande STM3210E-Eval, l'afficheur LCD a fonctionné correctement et nous sommes arrivés à communiquer avec quelques fonctions préprogrammées en utilisant les commandes utilisateurs mais on ne peut pas afficher les valeurs souhaités vu de l'absence du matériel.

La figure 2.26 montre quelques affichages sur l'écran LCD

Figure 2.26 - Afficheur LCD de la carte de commande

Comme est figuré, il ya des défauts dans la carte de commande, c'est évident de les trouver puisqu'elle n'est pas connectée au reste du matériel, ce qui empêche la navigation entre les différentes pages préprogrammées dans le LCD.

L'erreur de température est corrigée grâce au capteur de température qui existe dans la carte et qui a détecté après quelques instants la température ambiante.

2.4 Conception et implémentation de l'application du contrôle six étapes du BLDC

2.4.1 Description générale de l'application

Nous avons conçu l'application du contrôle du moteur BLDC en utilisant les différentes spécificités du microcontrôleur K60N512.

La figure 2.27 présente les principaux blocs constituant la commande du BLDC en se basant sur les différents composants des cartes de pilotage.

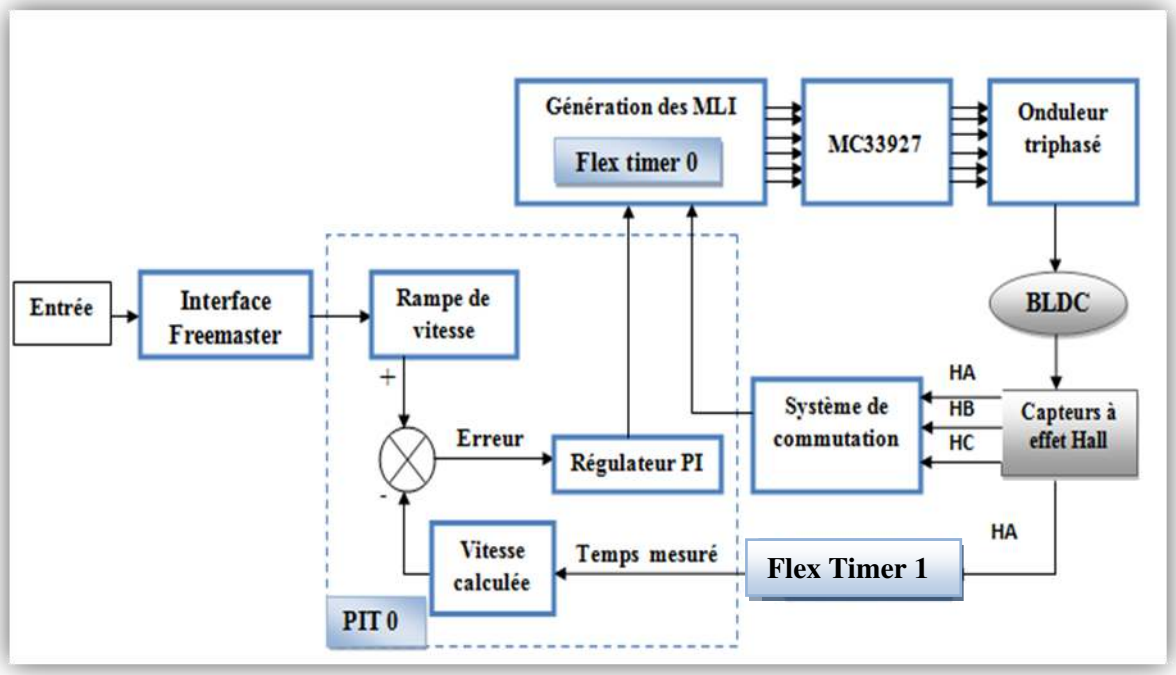


Figure 2.27 – Principe de commande d'un BLDC par K60N51

L'utilisateur fait entrer la valeur souhaitée de la vitesse à travers l'interface Freemaster.

Un capteur à effet Hall (HA) suffit pour capter le secteur dans lequel est incluse la position, et le Timer Flex timer1 FTM1 sert à mesurer la vitesse délivrée par le BLDC.

La comparaison entre la valeur mesurée de la vitesse et la valeur de référence générée par l'utilisateur permet de trouver l'erreur qui sera corrigée par un régulateur PI. Cette régulation est réalisée dans la routine d'interruption du Timer PIT0.

Ensuite, selon les signaux délivrés par les capteurs à effet Hall et la tension de sorties du régulateur PI, on génère les signaux MLI par le Trimer FTM0, ces signaux commandent par la suite l'onduleur triphasé à travers le bloc de commande MC33927.

2.4.2 Implémentation de l'application de la commande du BLDC

Nous avons réalisé l'application de commande vectorielle en se basant sur le diagramme représenté à la figure 2.28.

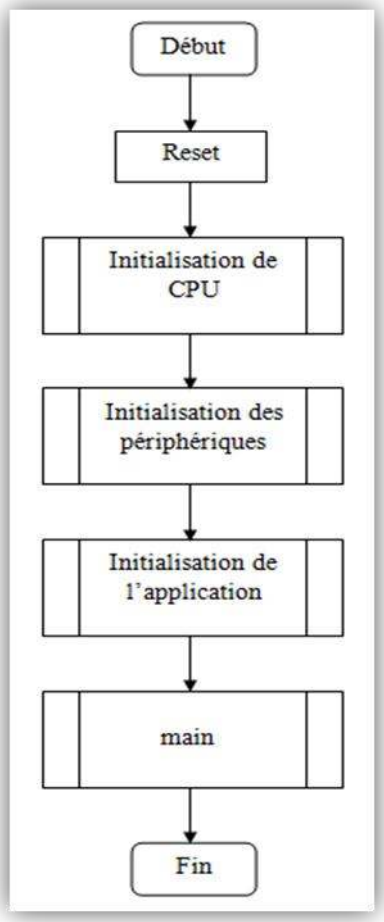


Figure 2.28 – Principe d'implémentation de la commande du BLDC

2.4.2.1 Action Reset

Lors de la réinitialisation (Reset), le processeur récupère le pointeur de pile (SP) et le compteur de programme (PC), il se branche par la suite à l'adresse du PC et commence l'exécution des instructions. De plus, les entrées/sorties numériques et les interruptions sont désactivés, le temporisateur du chien de garde est actif c'est pourquoi il devra être réparé (ou bien désactivé lors du débogage).

2.4.2.1 Initialisation de CPU

Après l'action Reset, on doit initialiser le processeur comme est expliqué à la figure 2.29

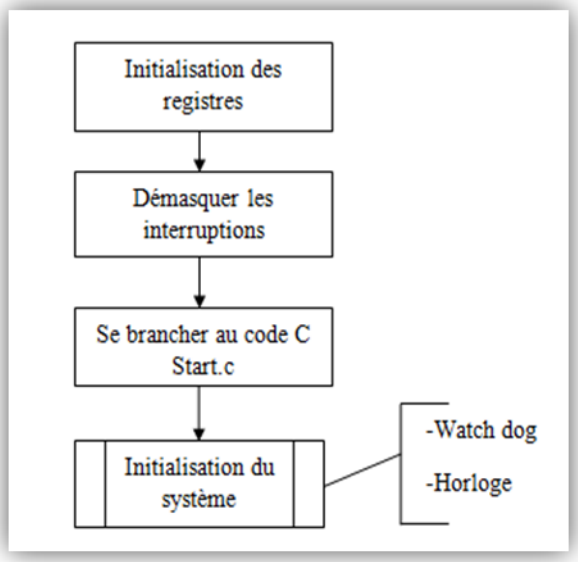


Figure 2.29 - Etapes d'initialisation de CPU

Après l'initialisation des registres du processeur à usage général(GPRs), on démasque les interruptions au cœur ARM et on appel le code C à traiter. Ces fonctions sont exécutées en utilisant le code assembleur. La fonction « Start.c » importée traite les différentes routines d'initialisation du processeur telles que la désactivation du Watch dog ou chien de garde et l'initialisation des horloges.

2.4.2.2 Initialisation des périphériques

Les périphériques qui interviennent dans les différentes étapes de la commande du moteur brushless à courant continu sont :

Timer PIT : Ce Timer s'intervient dans la boucle de vitesse.

Timer FTM0 : Ce Timer est utilisé dans la génération des signaux MLI nécessaires à la commutation des transistors de l'onduleur triphasé.

Timer FTM1 : Ce Timer est utilisé pour la mesure de la vitesse détectée.

Interface SPI : Utilisée dans l'application pour assurer la communication entre le bloc de commande de l'onduleur triphasé MC33927 et le microcontrôleur MK60N512.

Ports A, B, C : On choisie les pines nécessaires comme entrées des capteurs à effet Hall.

Interface UART : Cette interface est utilisée pour réaliser l'interfaçage entre l'application de commande embarquée et l'interface de contrôle Freemaster installée sur le PC. La configuration de l'UART est préparée dans le driver du logiciel Freemaster inclut dans le projet.

Les étapes d'initialisation des périphériques sont décrites dans le diagramme de la figure 2.30.

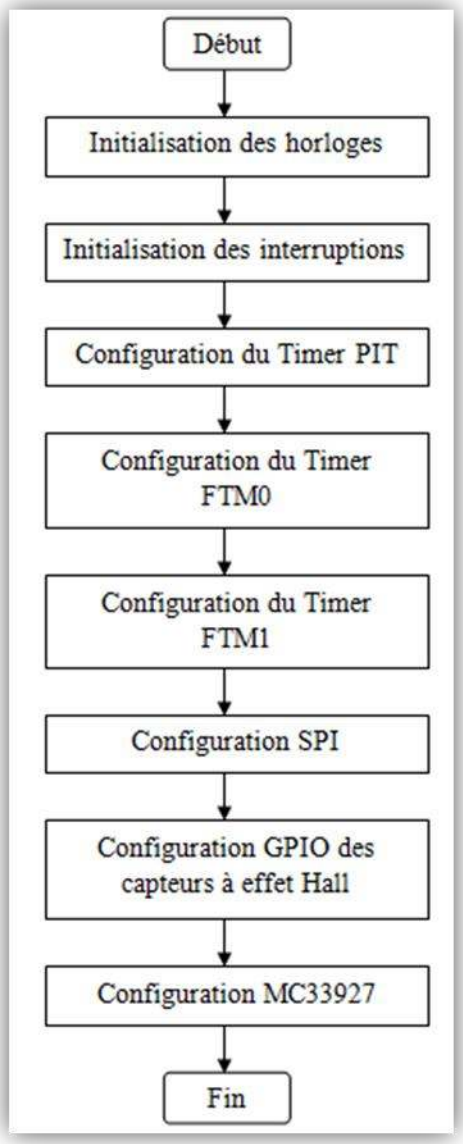


Figure 2.30 – Initialisation des différents périphériques

On commence tout d'abord par l'activation des ports d'horloge des périphériques. On passe par la suite à l'initialisation des interruptions et de leurs priorités en se basant sur la table des vecteurs d'interruption inclut dans le CPU. Ensuite on configure les différents périphériques à utiliser ainsi que le bloc de commande de l'onduleur de tension triphasé MC33927.

2.4.2.3 Initialisation de l'application

Le contrôle du moteur BLDC est généré autour des interruptions.

Interruption Timer FTM1 (Overflow Interrupt Handler) : Cette interruption est la plus prioritaire et elle est utilisée pour la mesure de la vitesse du rotor.

Interruption Timer PIT0 (Periodic interrupt) : L'interruption sur ce timer permet de traiter la régulation de la vitesse par le régulateur PI.

Interruption sur les entrées des capteurs Hall (Hall sensors interrupt): Cette interruption permet d'exécuter le processus de commutation de flux rotorique entre les six positions générées par les secteurs des signaux des capteurs à effet Hall comme est décrit brièvement dans le paragraphe 2.2.1.

A fin d'améliorer les performances de démarrage, une approche est utilisée dans l'algorithme de commande est de générer une rampe de vitesse (***Speed_scaled***) à partir la valeur entrée manuellement par l'utilisateur (***Speed_req***).Cette rampe est calculée grâce à des fonctions précisent contenues dans une librairie GFLIB incluse dans le projet.

Les étapes de la commande du BLDC sont résumées dans le diagramme de la figure 2.31.

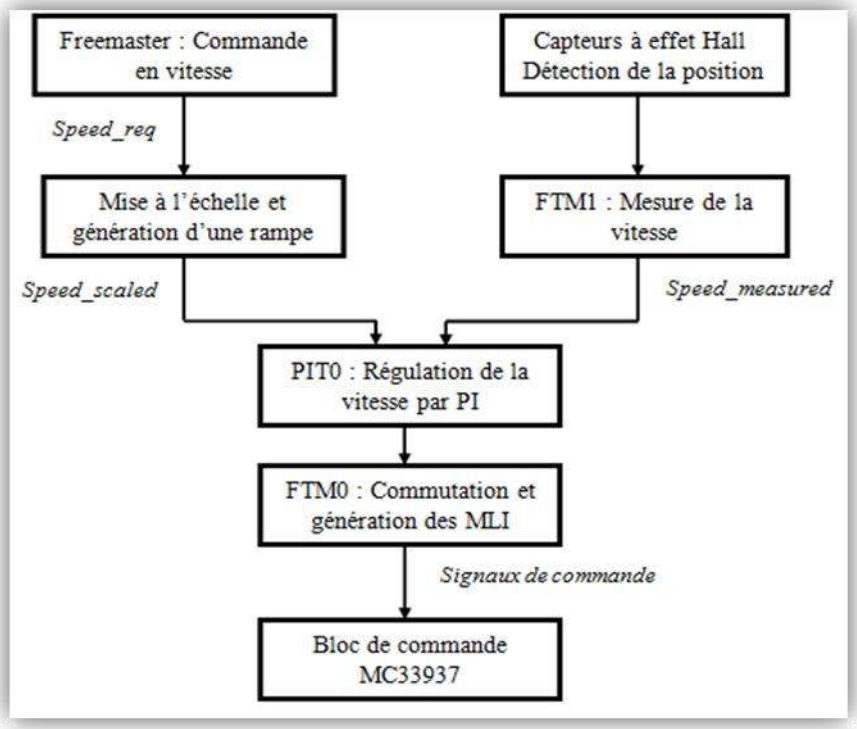


Figure 2.31 - Etapes de la commande du BLDC

2.4.2.4 La fonction main

La fonction main est utilisée pour l'initialisation des variables à utiliser, des horloges et des différents périphériques.

La communication entre l'utilisateur et l'application de commande du BLDC est assurée via une interface logicielle, le Freemaster, dont le driver est inclut dans le projet qui est sous la forme d'un ensemble de fichier assurant la lecture en temps réel des données et la gestion de la communication.

Cette communication est réalisée en appelant d'une façon périodique la fonction Freemaster_Poll. Cette fonction permet la gestion et le décodage des messages reçus de l'UART, qui assure la communication entre le PC et l'application embarquée.

Cette fonction donc sera exécutée dans la boucle infinie de main.

2.5 Conclusion

Dans ce chapitre, nous avons réalisé une étude de faisabilité en s'appuyant sur les résultats de simulations des solutions de contrôle qui existent, pour arriver à concevoir et implémenter les différents algorithmes des commandes de deux types de moteurs brushless : la commande du PMSM à base d'un microcontrôleur STM32, et le contrôle du BLDC par le microcontrôleur K60N512.

Conclusion générale

Dans ce projet, nous avons expliqué les différents étapes à suivre à fin d'implémenter les deux types de commande du moteur sans balais ou brushless. Il s'agit de la commande vectorielle d'un moteur PMSM à pôles lisses à base de microcontrôleur STM32 et la commande six étapes d'un moteur BLDC à base de microcontrôleur Kinetis 60.

Après l'étude des différents besoins de ce projet, deux grandes parties sont identifiées et réalisées, à savoir :

La première partie qui a consisté à exploiter une librairie de commande vectorielle des moteurs sans balais fournie par STMicroelectronics en vue de concevoir et implémenter la commande du moteur PMSM à pôles lisses sur le microcontrôleur STM32 et pouvoir le piloter via une interface LCD de la carte STM3210E-EVAL.

La deuxième partie a porté sur le développement d'une application assurant la commande d'un BLDC à partir des signaux délivrés par un capteur à effet Hall. Cette commande est implémentée par la suite sur le microcontrôleur K60N512 de la famille Kinetis, et l'utilisateur pourra piloter directement son moteur via une interface Freemaster installée sur le PC.

En plus des connaissances techniques, ce projet nous a permis de développer nos compétences dans le domaine de la gestion de projet et la planification des actions en respectant les règles du développement dans le but d'atteindre la qualité mirifique qui satisfait les besoins de la clientèle.

Comme perspective, Telnet Holding propose d'améliorer encore plus les techniques de pilotage des moteurs brushless en développant des applications de hautes technologies permettant la commande à distance de ces moteurs en se basant sur des protocoles de communication aussi développés.

Bibliographie

[1] http://www.groupe-telnet.com

[2] http://www.tustex.com/download/telnetaib.pdf

[3] A.GUILBERT. Machines synchrones. Théorie, *fonctionnement et calcul des machines électriques.* Dunod, Paris, 1965

[4] Margery CONNER. Permanent-Magnet Motors, boost efficiency and power density. EDN, 13 September 2007

[5] Driss YOUSFI.Alimentation et commande des moteurs DC.Ecole Nationale Des Sciences Appliquée de Marrakech. ModuleM.2.2, Janvier 2007.

[6] http://www.st.com/stonline/company/identity/compres.html

[7] François BERNOT. Machines à courant continu : Constitution et fonctionnement. *Techniques de l'Ingénieur, traité Génie électrique*. Pages 08-10. Ref D3-555, 2000.

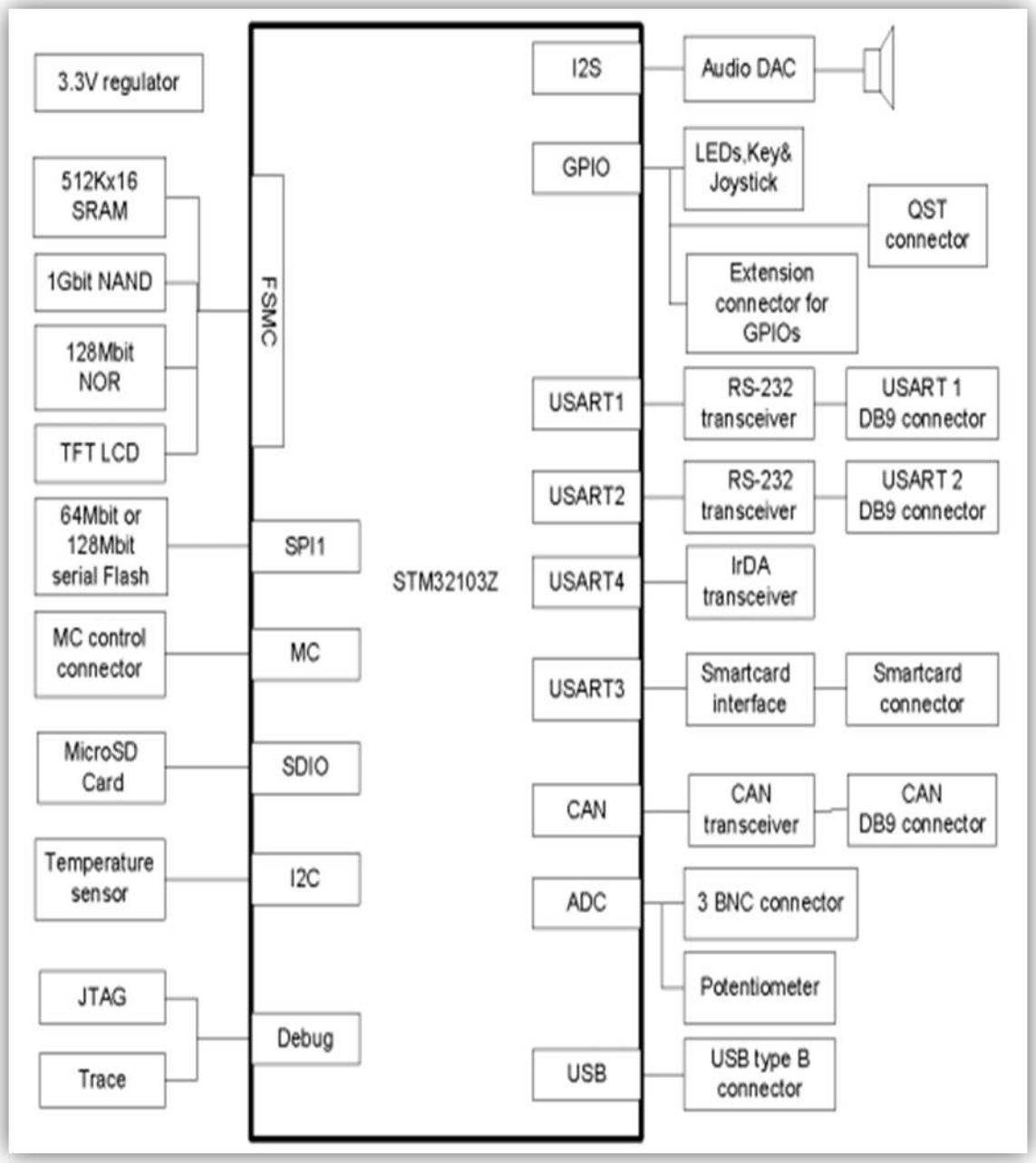
[8] Michel LAJOIE-MAZENC, Philippe VIAROUGE. Alimentation des machines synchrones. *Techniques de l'Ingénieur, traité Génie électrique.* Pages 25-26. Ref D3-630, Février 2001.

[9] http://www.datasheets.org.uk/25DMIPS-datasheet.html

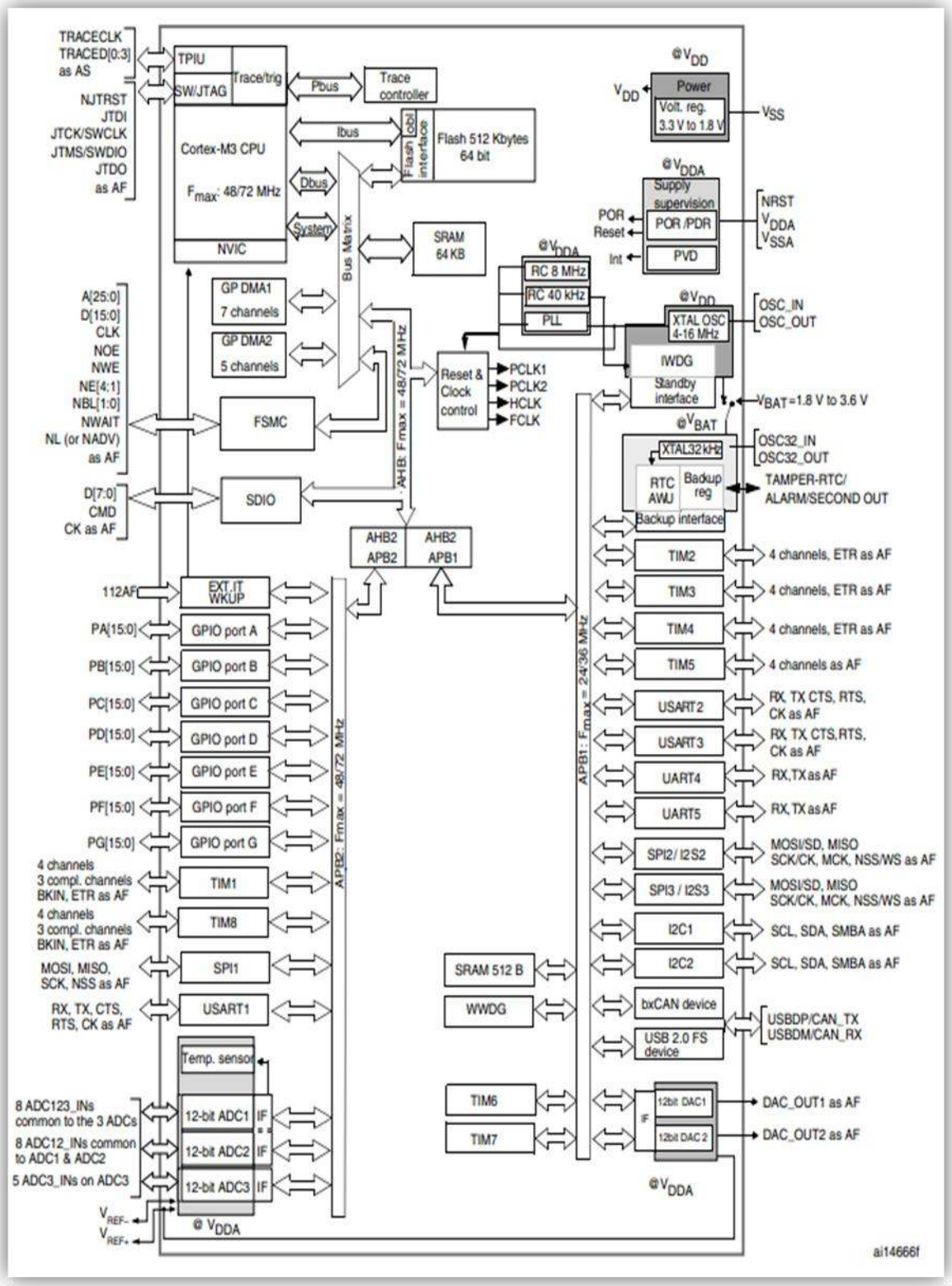
[10] Mohamed Wissem NAOUAR. Commande numérique à base de composants FPGA d'une machine synchrone,algorithmes de contrôle du courant. Thèse de doctorat L'Ecole Nationale d' Ingénieurs de Tunis et l'université de CERGY PONTOISE, Décembre 2007.
Disponible sur Internet <http://www.biblioweb.ucergy.fr/theses/07CERG0344.pdf>

ANNEXES

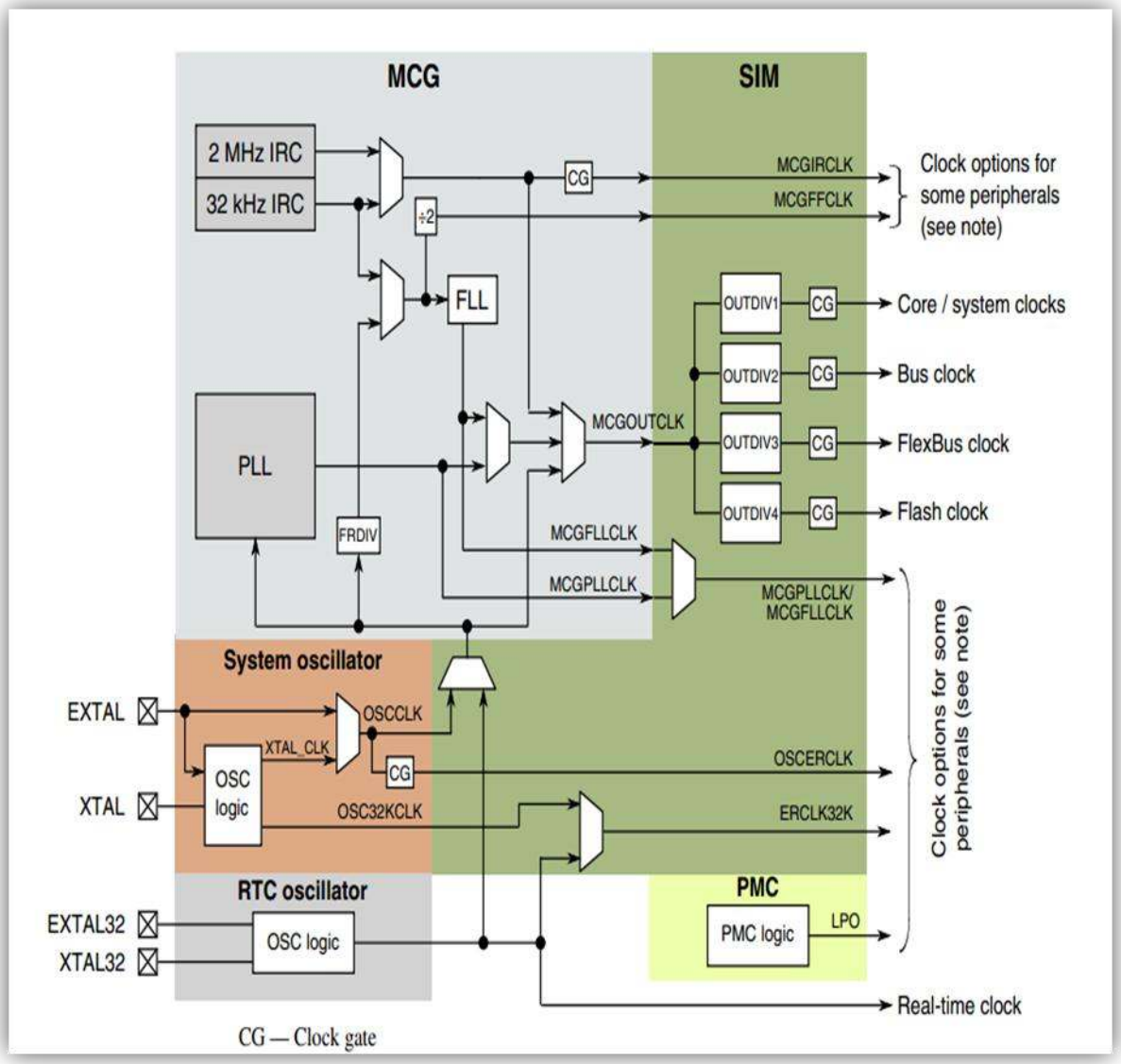
Annexe 1 : Architecture de la carte STM3210E-EVAL



Annexe 2 : Architecture interne du STM32F103



Annexe 3 : Diagramme d'horloge du microcontrôleur K60N512



Annexe 4 : Différents paramètres du GUI pour le moteur PMSM

Nom de la variable (GUI)	*Description*	*Include file*	*#define*
Structure magnétique			
Internal PMSM/Surface Mounted PMSM	la structure magnétique du rotor	-	-
Paramètres électriques			
Pole pairs	Nombre de paires de pôles	PMSM motor parameters.h	POLE_PAIR_NUM
Max Rated Speed (rpm)	Vitesse nominale maximale du moteur en rpm (tr/min)	PMSM motor parameters.h	MOTOR_MAX_SPEED_RPM
Nominal Current (A)	Courant nominal du moteur	PMSM motor parameters.h	NOMINAL_CURRENT
Nominal DC Voltage (V)	Tension efficace (Vrms) nominale entre phases		-
Rs (Ohm)	Résistance statorique (phase)	PMSM motor parameters.h	RS
Ld (mH)	Inductance statorique directe (Ld=Lq=Ls dans le cas d'un SM-PMSM : PMSM à pôles lisses	-	-
Lq (mH)	Inductance statorique en quadrature	PMSM motor parameters.h	LS
Ls (mH)	Inductance statorique (phase)	PMSM motor parameters.h	LS
Demagnetizing Current (A)	Courant de démagnétisation du rotor, il est égal au courant nominal	PMSM motor parameters.h	ID_DEMAG
B-EmfConstant (Vrms/Krpm)	(Tension efficace nominale entre phases) / (Vitesse nominale max du moteur exprimée en Krpm) (1000rpm=1Krpm)	PMSM motor parameters.h	MOTOR_VOLTAGE_CONSTANT

Annexe 5 : Interface utilisateur Freemaster