

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

BRAHMI Abdelghani

&

HAMOUDI Ahmed

pour l'obtention du diplôme de Master en Électronique spécialité Machines Electriques.

Thème

Implémentation du filtre Kalman sur DSP pour l'estimation des grandeurs rotoriques

Proposé par : Mme EL MERRAOUI Khadidja ep. FERDJOUNI

&

Mr FERDJOUNI Abdelaziz

Année Universitaire 2011-2012

Je dédie ce modeste travail

Aux deux êtres les plus chers au monde qui sont :

Ma très chère mère qui a toujours été là pour moi et qui s'est donnée de la peine afin que

j'obtienne mon diplôme.

Mon père qui a été toujours derrière mes exploits, avec son aide matérielle et morale

durant toutes mes études.

A mes chères sœurs Fatima, Malika et Noura que j'apprécie beaucoup.

A mes frères Djamel et Mouhamed que j'estime beaucoup.

A tous ceux qui me sont chers et à toute la famille.

A tous mes amis (es) qui m'ont aidé et

soutenu : Fethi, Daoud, Wahid, Salim, Mouhamed, Sid Ahmed...

A toi mon ami et frère Abdelghani pour les moments précieux que nous avons passé

ensemble et que je n'oublierai jamais.

...Ahmed

Je dédie ce modeste travail

Aux deux êtres les plus chers au monde qui sont :

Ma très chère mère qui a toujours été là pour moi et qui s'est donnée de la peine afin que

j'obtienne mon diplôme.

Mon père qui a été toujours derrière mes exploits, avec son aide matérielle et morale

durant toutes mes études.

A ma chère sœur Amina que j'apprécie beaucoup.

A mes frères Mouhamed, Abdelraouf et Aissa que j'estime beaucoup.

A tous ceux qui me sont chers et à toute la famille.

A tous mes amis (es) qui m'ont aidé et soutenu : Fethi, Abdelwaheb, Aissa, Moustafa, et Yassine

A toi mon ami et frère Ahmed pour les moments précieux que nous avons passé

ensemble et que je n'oublierai jamais.

...Abdelghani

Table des matières

Introduction générale.....	1
----------------------------	---

Chapitre 1 : Modélisation de la MAS

1.1 Introduction.....	3
1.2 Définition.....	3
1.3 construction de MAS.....	4
1.3.1 Le stator.....	4
1.3.2 Le rotor	5
1.4 Principe de fonctionnement de la MAS.....	6
1.5 Modélisation de la machine asynchrone.....	6
1.5.1 Hypothèses simplificatrices.....	7
1.5.2 Modèle de la machine asynchrone triphasée.....	8
1.5.3 Equations en diphasé.....	10
1.5.4 Choix du repère de Park	12
1.5.5 Référentiel lié au stator	13
1.5.6 Le modèle de la MAS dans l'espace d'état	16
1.6 Conclusion.....	19

Chapitre 2 : Filtre de Kalman

2.1 Introduction	20
2.2 L'observabilité	20
2.2.1 Observabilité des systèmes linéaires.....	20
a Critère d'observabilité de Kalman	21
2.2.2 Observabilité des systèmes non-linéaire.....	21
2.3 Observateurs	23
2.3.1 Observateurs des systèmes non linéaires.....	24

2.3.2	Observateurs des systèmes linéaires.....	25
a	Observateur de Lueberger	25
b	Le filtre de Kalman.....	27
2.3.3	Synthèse du filtre de Kalman.....	27
a	Cas des systèmes continus	27
b	Cas des systèmes discrets.....	30
2.4	Application du FK sur la MAS	33
2.4.1	Modèle discret du filtre de Kalman	35
2.5	Conclusion	36

Chapitre 3 : Implémentation du FK sur DSP

3.1	Introduction	37
3.2	Architecture des DSP.....	37
3.2.1	Structure de Von Neuman	38
3.2.2	Structure de Harvard	38
3.3	Principales caractéristiques d'un DSP	38
3.3.1	La fonction MAC.....	38
3.3.2	Double accès à la mémoire	39
3.3.3	Le pipeline.....	39
3.4	Les types de DSP.....	39
3.4.1	Représentation binaire des nombres fractionnaires en format virgule fixe.....	40
a	Les valeurs extrêmes en virgule fixe sur N bits avec un format Q_k	40
b	L'addition de nombres fractionnaires en virgule fixe.....	41
c	Multiplication de 2 nombres fractionnaires en virgules fixe.....	41
3.4.2	Représentation binaire des nombres fractionnaires en format virgule flottante.....	41
a	Plage des nombres représentables en format virgule flottante.....	41
3.5	La famille TMS320.....	42
3.6	Le TMS320F2812.....	43
3.6.1	L'unité centrale (cœur du DSP).....	46

a	Le multiplieur.....	46
b	L'unité arithmétique et logique UAL.....	46
c	Les shifters	46
d	L'interface JTAG	46
e	Le gestionnaire d'interruption.....	47
f	Les temporisateurs	47
g	L'unité arithmétique et logique auxiliaire et ces registres.....	47
3.6.2	Les modes d'adressage	47
a	L'adressage direct	47
b	L'adressage indirect	48
3.6.3	Les interruptions du TMS320F2812.....	48
3.6.4	Le convertisseur analogique numérique (ADC).....	51
3.6.5	Le watchdog	52
3.6.6	Gestionnaire d'évènement EV (Event Manager).....	53
a	Le temporisateur à usage général (General-Purpose Timers).....	53
b	Unité de comparaison (Full Compare).....	54
c	L'unité de capture.....	55
d	Encodeur quadrature QEP.....	56
3.7	Code Composer Studio.....	57
3.7.1	Outils de génération du code.....	57
3.8	Implémentation du FK sur DSP	60
3.8.1	La programmation en C.....	60
3.8.2	La programmation en utilisant La bibliothèque IQ Math.....	61
a	Quelques fonctions de la bibliothèque IQ Math.....	62
3.9	Conclusion	67

Chapitre 4: Résultats et commentaires

4.1	Introduction	68
4.2	Résultat d'application du FK (simulation Matlab) à la MAS.....	68
4.3	Résultat d'application d'algorithme du FK écrit en langage C sur le DSP F2812.....	72

4.4	Résultat d'application d'algorithme du FK écrit en utilisant la bibliothèque IQ Math sur le DSP F2812.....	75
4.5	Comparaison entre la programmation en C et la programmation en utilisant IQ Math.....	77
4.6	Application du FK sur la MAS réelle.....	78
4.6.1	Acquisition de la vitesse	79
4.6.2	Acquisition des courants et des tensions	80
4.6.3	Visualisation des grandeurs estimées.....	80
4.7	La carte d'interface	80
4.7.1	Le circuit d'atténuation.....	80
4.7.2	Le circuit de conversion.....	81
4.8	Conclusion	81
	Conclusion générale.....	83
	Annexes	
	Bibliographie	

ملخص:

ركزنا في هذه المذكرة على تركيب مرشح كالمان من أجل ملاحظة تيارات الساكن والتدفق الدوار لمحرك لا تزامني ثلاثي الطور. تطبيق الخوارزمية المحصل عليها سيكون على مستوى معالج الإشارة الرقمي (DSP TMS320F2812) المطور من قبل (Texas Instruments) بغرض ملاحظة الإشارات الكهربائية و المغناطيسية في الزمن الحقيقي.

كلمات المفاتيح: المحرك اللاتزامني، مرشح كالمان، معالج الإشارة الرقمي، الملاحظ

Résumé :

Ce présent mémoire consiste en la synthèse du filtre de Kalman afin d'estimer les courants statoriques et les flux rotoriques de la machine asynchrone. L'implémentation de l'algorithme obtenue est réalisée sur le DSP TMS320F2812 de TEXAS INSTRUMENTS pour une observation des signaux électriques et magnétiques en temps réel.

Mots clés : Filtre de Kalman, observateurs, Digital Signal Processor, Machine Asynchrone

Abstract :

This present report consists of the synthesis of the filter of Kalman in order to estimate the stator currents and rotor flows of the induction machine. The implementation of the algorithm obtained is carried out on DSP TMS320F2812 of TEXAS INSTRUMENTS for an observation of the electric and magnetic signals in real time.

Keywords : Filter of Kalman, observers, Digital Signal Processor, Induction Machine

Liste des figures

Figure : 1.1 moteur asynchrone	4
Figure : 1.2 stator de la machine asynchrone.....	5
Figure : 1.3 rotor à cage d'écureuil	5
Figure : 1.4 illustration de la disposition des enroulements statoriques et rotoriques	8
Figure : 1.5 position des axes dq par rapport aux axes de la machine	11
Figure : 2.1 Principe de l'observateur	24
Figure : 2.2 Schéma fonctionnel de l'observateur de Luenberger	26
Figure : 2.3 L'algorithme du filtre de kalman	33
Figure : 3.1 Représentation schématique des structures de Von-Neuman et de Harvard	37
Figure : 3.2 Schéma bloc des périphériques du DSC F2812.....	44
Figure : 3.3 La carte mémoire du TMS320F2812.....	45
Figure : 3.4 Différentes sources d'interruptions	48
Figure : 3.5 masquage des interruptions	49
Figure : 3.6 Périphérique d'extension des interruptions	50
Figure : 3.7 Schéma bloc de l'ADC en mode dual séquenceurs	52
Figure : 3.8 Schéma logique de réinitialisation du DSP	52
Figure : 3.9 Schéma bloc du gestionnaire d'évènements	53
Figure : 3.10 Schéma bloc temporisateur à usage général 1,2.....	54
Figure : 3.11 Schéma bloc de l'unité de comparaison	55
Figure : 3.12 Principe l'encodeur quadra que	56
Figure : 3.13 Détermination de position par un signal quadra que	57
Figure : 3.14 processus de développement de code exécutable.....	58
Figure : 3.15 modèle éclaté du filtre de Kalman	66
Figure : 4.1 La vitesse mécanique Ω et le couple électromagnétique.....	69
Figure : 4.2 la superposition des courants I_{qs} et \hat{I}_{qs} et l'erreur entre eux	70
Figure : 4.3 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux	70
Figure : 4.4 la superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux	71
Figure : 4.5 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux	71

Figure : 4.6 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux	72
Figure : 4.7 la superposition des courants I_{qs} et \hat{I}_{qs} et l'erreur entre eux	73
Figure : 4.8 la superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux	73
Figure : 4.9 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux	74
Figure : 4.10 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux	75
Figure : 4.11 la superposition des courants I_{qs} et \hat{I}_{qs} et l'erreur entre eux	76
Figure : 4.12 a superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux	76
Figure : 4.13 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux.....	77
Figure : 4.14 schéma fonctionnel du banc d'essai	79

Chapitre 1 Modélisation de la MAS

1.1 Introduction

Le concept de la machine asynchrone (MAS) d'induction date depuis 120 ans. Depuis, cette machine s'est imposée dans l'industrie. Elle possède en effet plusieurs avantages : simplicité, robustesse, prix peu élevé et entretien facile.

Relié au réseau électrique à fréquence fixe, la MAS permet la réalisation de la majorité des entraînements à vitesse constante, et cela dans une très large gamme de puissances et avec une vitesse « quasi constante »; en effet ce moteur présente un gros intérêt, lorsqu'on lui demande un effort supplémentaire, mais présente la caractéristique de glissement (diminution légère de sa vitesse de fonctionnement).

Vu ces qualités, on l'a utilisée dans le cas de vitesse variable. Il y a quelques dizaines d'années, les solutions techniques de l'époque permirent d'obtenir une variation de vitesse en faisant varier le glissement. Actuellement, ce sont les alimentations électroniques de puissance à fréquence variable qui sont amplement utilisées pour obtenir cette variation, certes au prix d'une certaine complexité des alimentations et des commandes associées [1] [3].

L'objectif de ce chapitre est de présenter la modélisation de la machine asynchrone sous forme de modèle d'état selon un repère de Park lié au stator, déterminé à partir du modèle triphasé obtenu de la MAS.

1.2 Définition

Selon le vocabulaire électrotechnique international, « une machine asynchrone est une machine à courant alternatif dont la vitesse en charge et la fréquence du

réseau auquel elle est reliée, ne sont pas dans un rapport constant ». Ce vocable est le seul utilisé en France alors que les pays anglo-saxons désignent par machine à induction « une machine asynchrone dont le circuit magnétique associé à deux ou plus de deux circuits électriques se déplacent l'un par rapport à l'autre et dans laquelle l'énergie est transférée de la partie fixe à la partie mobile, ou inversement, par l'induction électromagnétique » [2].

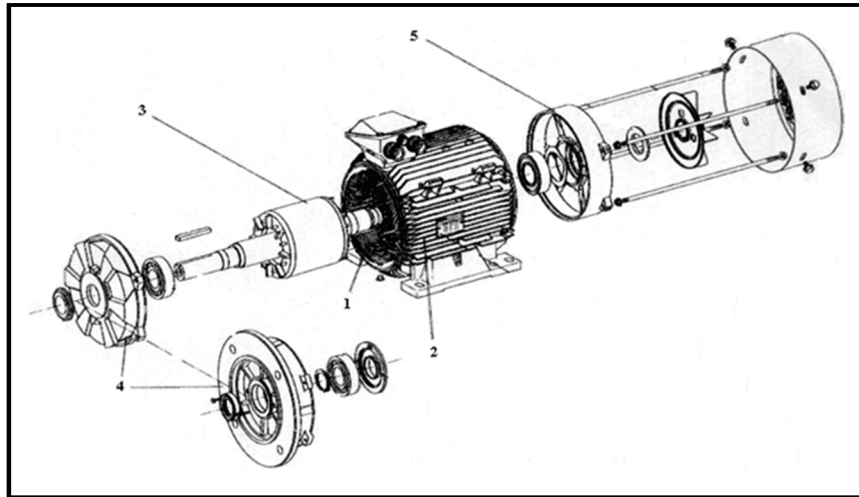


Figure : 1.1 moteur asynchrone

1.3 Constitution de la MAS

La machine asynchrone triphasée comprend deux parties : le stator (partie fixe) et le rotor (partie tournante).

1.3.1 Le stator

Le stator comporte une carcasse en acier renfermant un empilage de tôles identiques qui constituent un cylindre vide ; ces tôles identiques sont percées de trous à leur périphérie intérieure. L'alignement de ces trous forme des encoches dans lesquelles on loge un bobinage triphasé.



Figure : 1.2 stator de la machine asynchrone

1.3.2 Le rotor

Le rotor se compose d'un cylindre de tôles poinçonnées à leur périphérie extérieur pour former les encoches destinées à recevoir des conducteurs. Il est séparé du stator par un entrefer très court de l'ordre de 0.4 à 2 mm seulement. Il existe deux types de rotor :

Le rotor à cage d'écureuil et le rotor bobiné. L'enroulement de la cage d'écureuil est constitué de barres de cuivre nues introduites dans les encoches ; ces barres sont soudées à chaque extrémité à deux anneaux qui les court-circuitent. L'ensemble ressemble à une cage d'écureuil d'où le nom de rotor à cage. Dans les moteurs de petite et moyenne puissance, les anneaux sont formés d'un seul bloc d'aluminium coulé.

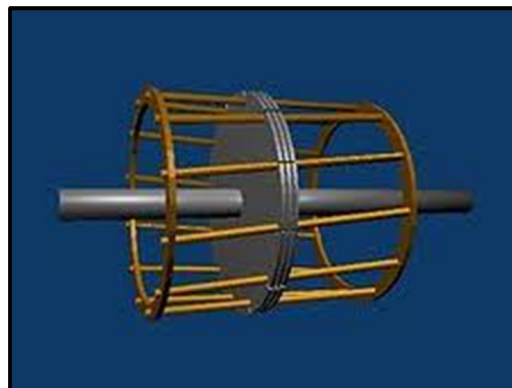


Figure : 1.3 rotor à cage d'écureuil

Le rotor bobiné comprend un bobinage triphasé, semblable à celui du stator, placé dans les encoches. Il est composé de trois enroulements raccordés en étoile ; l'extrémité libre de chaque enroulement est reliée à une bague tournante avec l'arbre. Ces bagues permettent, par l'intermédiaire de trois balais, d'insérer une résistance extérieure en série avec chacun des trois enroulements lors de démarrage du moteur. En fonctionnements normal, les trois balais sont court-circuités.

1.4 Principe de fonctionnement de la MAS

Le fonctionnement de la machine asynchrone triphasée est basé sur l'application des principes de l'électromagnétisme. Le champ tournant créé par l'ensemble des courants triphasés circulant dans les trois enroulements du stator coupe les conducteurs du rotor, selon la loi de FARADAY une tension est induite dans chacun des conducteurs coupé par le champ. D'autre part, puisque la cage d'écureuil forme un circuit fermé, un courant I se met à circuler dans le conducteur qui est momentanément en dessous de l'aimant. Ce courant traverse le champ magnétique généré par les enroulements statoriques, de sorte que, d'après le principe de LAPLACE, le conducteur est soumis à une force mécanique qui agit toujours dans une direction telle qu'elle entraîne le conducteur dans le sens de déplacement du champ, étant donné que le rotor est libre de se déplacer il accélérerait. Ce pendant à mesure qu'il gagne de la vitesse, la coupure des conducteurs par le champ ce fait de moins rapidement et la tension induite diminue, de même que le courant I . Par conséquent, la force agissant sur les conducteurs situés en dessous de l'aimant diminue. Si le rotor tourne à la même vitesse que le champ tournant, la tension induite le courant I et la force mécanique deviendront nuls (la variation du flux est nulle, implique l'annulation de la tension) [1].

1.5 Modélisation de la machine asynchrone

Un modèle est un outil mathématique qui permet de représenter ou de reproduire un système réel donné. L'intérêt d'un modèle est l'analyse et la prédiction du comportement en régime statique et dynamique du système physique.

L'objectif de ce chapitre est de modéliser la machine asynchrone triphasée sous forme d'équations d'état [4].

La machine asynchrone présente un système d'équations très complexe à étudier. Il est donc nécessaire de développer un modèle plus simple. Le modèle mathématique de la MAS est un système de six équations différentielles. La résolution d'un tel système est difficile, pour contourner cette difficulté on utilise la transformation de PARK sous certaines hypothèses simplificatrices. Cette dernière permet une représentation biphasée équivalente de la machine triphasée ce qui réduit considérablement la complexité du modèle [4].

1.5.1 Hypothèses simplificatrices

Les hypothèses simplificatrices admises dans le modèle de la machine asynchrone sont :

- La parfaite symétrie de la machine.
- La répartition spatiale sinusoïdale des différents champs magnétiques le long de l'entrefer.
- On admet de plus que la force magnétomotrice créée par chacune des phases des deux armatures est à répartition sinusoïdale.
- L'absence de saturation et de pertes dans le circuit magnétique.
- Assimilation de la cage à un bobinage en court-circuit de même nombre de phases que le bobinage statorique (c'est-à-dire 3).
- Les résistances des enroulements ne varient pas avec la température.
- On considère une densité de courant uniforme dans la section des conducteurs élémentaires.
- L'entrefer est d'épaisseur uniforme et l'effet d'encoche est négligé, les inductances propres sont constantes et les inductances mutuelles sont des fonctions sinusoïdales de l'angle entre les axes rotoriques et statoriques.

1.5.2 Modèle de la machine asynchrone triphasée

Elle se compose d'un :

-Circuit statorique fixe comportant trois phases identiques décalées entre elles de 120° .

-Circuit rotorique mobile comportant trois phases identiques en court-circuit décalées entre elles de 120° .

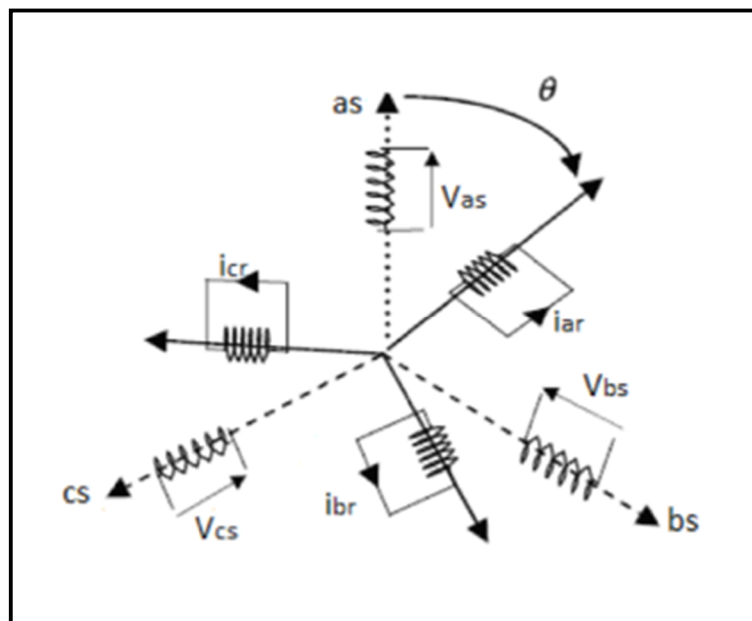


Figure : 1.4 illustration de la disposition des enroulements statoriques et rotoriques

Dans le repère triphasé, les trois vecteurs (Figure 1.4) sont orientés selon les axes des trois enroulements statoriques de la machine. Il est de même pour le rotor.

L'axe a_s est considéré comme axe de référence, et l'angle θ définit la position du rotor par rapport au stator [3].

En tenant compte des hypothèses mentionnées précédemment les équations électriques des tensions statoriques et rotoriques peuvent s'écrire sous une forme matricielle en appliquant la loi d'Ohm comme suit :

Au niveau du stator

$$[V_{abcs}] = [r_s][I_{abcs}] + \frac{d[\phi_{abcs}]}{dt} \quad (1.1)$$

Au niveau du rotor

$$[V_{abcr}] = 0 = [r_r][I_{abcr}] + \frac{d[\phi_{abcr}]}{dt} \quad (1.2)$$

Avec :

$$[V_{abcs}] = \begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} \quad [I_{abcs}] = \begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \end{bmatrix} \quad [\phi_{abcs}] = \begin{bmatrix} \phi_{as} \\ \phi_{bs} \\ \phi_{cs} \end{bmatrix}$$

et

$$[V_{abcr}] = \begin{bmatrix} V_{ar} \\ V_{br} \\ V_{cr} \end{bmatrix} \quad [I_{abcr}] = \begin{bmatrix} I_{ar} \\ I_{br} \\ I_{cr} \end{bmatrix} \quad [\phi_{abcr}] = \begin{bmatrix} \phi_{ar} \\ \phi_{br} \\ \phi_{cr} \end{bmatrix}$$

et

$$[r_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \quad [r_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}$$

Les flux totaux de la machine sont en relation avec les courants par l'intermédiaire des équations suivantes :

$$\begin{bmatrix} \phi_{as} \\ \phi_{bs} \\ \phi_{cs} \\ \phi_{ar} \\ \phi_{br} \\ \phi_{cr} \end{bmatrix} = \begin{bmatrix} l_{aa} & l_{ab} & l_{ab} & l_1 & l_2 & l_3 \\ l_{ab} & l_{aa} & l_{ab} & l_3 & l_1 & l_2 \\ l_{ab} & l_{ab} & l_{aa} & l_2 & l_3 & l_1 \\ l_1 & l_2 & l_3 & l_{AA} & l_{AB} & l_{AB} \\ l_3 & l_1 & l_2 & l_{AB} & l_{AA} & l_{AB} \\ l_2 & l_3 & l_1 & l_{AB} & l_{AB} & l_{AA} \end{bmatrix} \begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \\ I_{ar} \\ I_{br} \\ I_{cr} \end{bmatrix} \quad (1.3)$$

Donc :

$$[\phi_{abcs}] = [L_s][I_{abcs}] + [M_{sr}][I_{abcr}] \quad (1.4)$$

$$[\phi_{abcr}] = [L_r][I_{abcr}] + [M_{rs}][I_{abcs}] \quad (1.5)$$

Avec la matrice inductance:

$$[L_s] = \begin{bmatrix} l_{aa} & l_{ab} & l_{ab} \\ l_{ab} & l_{aa} & l_{ab} \\ l_{ab} & l_{ab} & l_{aa} \end{bmatrix} \quad [L_r] = \begin{bmatrix} l_{AA} & l_{AB} & l_{AB} \\ l_{AB} & l_{AA} & l_{AB} \\ l_{AB} & l_{AB} & l_{AA} \end{bmatrix}$$

$$[M_{sr}] = l_{aA} \begin{bmatrix} \cos \theta & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \cos \theta & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos \theta \end{bmatrix}$$

Ou :

$[L_s]$: Matrice des inductances propres et mutuelles entre phases statoriques.

$[L_r]$: Matrice des inductances propres et mutuelles entre phase rotoriques.

$[M_{sr}]$: Matrice des inductances mutuelles entre phases statoriques et rotoriques.

l_{aa} : Inductance propre d'un enroulement statorique.

l_{AA} : Inductance propre d'un enroulement rotorique.

l_{ab} : Inductance mutuelle entre deux enroulements statoriques.

l_{AB} : Inductance mutuelle entre deux enroulements rotoriques.

l_{aA} : Inductance mutuelle maximale entre une phase du stator et une phase de rotor.

Notons que, la matrice des inductances mutuelles rotoriques n'est d'autre que la transposée de celle des inductances mutuelles statoriques ; $[M_{rs}] = [M_{sr}]^T$

1.5.3 Equations en diphasé

Le modèle diphasé de la MAS est obtenu par une transformation du triphasé en un repère diphasé, c'est en fait un changement de base appliqué sur les grandeurs électriques et magnétiques (tensions, flux, et courants). Des relations indépendantes de l'angle θ et une réduction de l'ordre des équations de la machine sont obtenues. La transformation la plus connue par les électrotechniciens est celle de *Park* [3].

La figure 1.4 met en relief l'axe direct d du référentiel de *Park*, et l'axe en quadrature d'indice q .

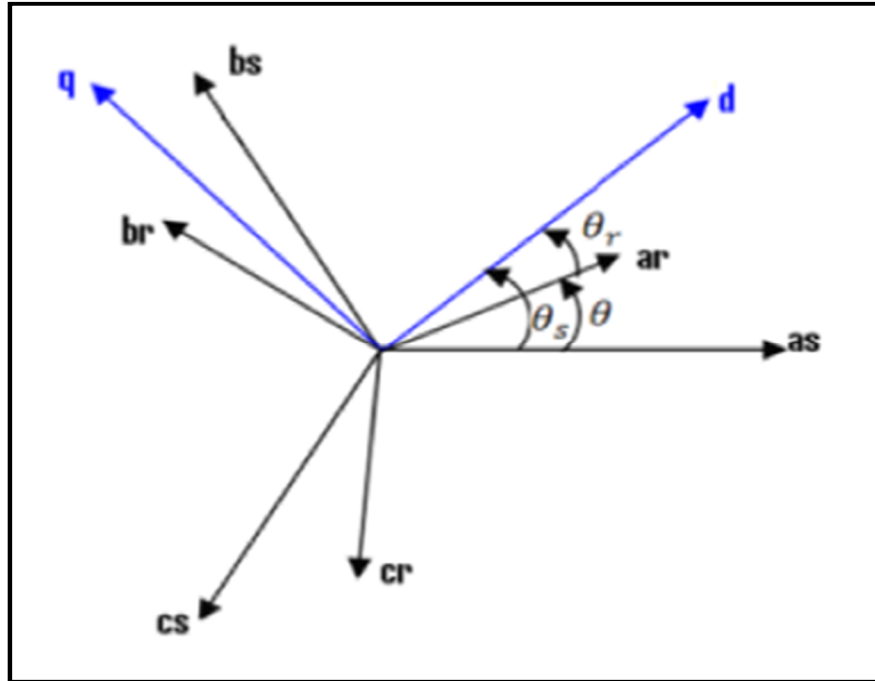


Figure : 1.5 position des axes dq par rapport aux axes de la machine

Avec :

θ : L'angle entre l'axe as et ar .

θ_s : L'angle entre l'axe as et d .

θ_r : L'angle entre l'axe ar et d .

La matrice de transformation de *Park* $[P_i]$ est définie comme suit :

$$[P_i] = \frac{2}{3} \begin{bmatrix} \cos \theta_i & \cos \left(\theta_i - \frac{2\pi}{3} \right) & \cos \left(\theta_i + \frac{2\pi}{3} \right) \\ -\sin \theta_i & -\sin \left(\theta_i - \frac{2\pi}{3} \right) & -\sin \left(\theta_i + \frac{2\pi}{3} \right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (1.6)$$

On a choisi $2/3$, pour conserver l'amplitude des tensions, courants, et flux. θ_i est l'angle entre l'axe d et l'axe de référence (statorique ou rotorique) dans le système triphasé.

La transformation directe est alors :

$$\begin{bmatrix} x_d \\ x_q \\ x_o \end{bmatrix} = [P_i] \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (1.7)$$

Où, x_{abc} représente les variables considérées de la machine qui sont tensions, courants ou flux. La variable x_o représente la composante homopolaire, ajoutée pour rendre la transformation réversible.

Pour revenir aux grandeurs triphasées, la transformation inverse de *Park* est appliquée.

$$[P_i]^{-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 1 \\ \cos\left(\theta_i - \frac{2\pi}{3}\right) & -\sin\left(\theta_i - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_i + \frac{2\pi}{3}\right) & -\sin\left(\theta_i + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad (1.8)$$

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = [P_i]^{-1} \begin{bmatrix} x_d \\ x_q \\ x_o \end{bmatrix} \quad (1.9)$$

1.5.4 Choix du repère de *Park*

Selon l'objectif de l'application, il existe trois choix possible pour l'orientation du repère dq , il peut être fixé au stator, au rotor ou au champ tournant :

- Repère d'axes dq fixe lié au stator ou repère stationnaire ($\theta_s = 0$). Les grandeurs électriques évoluent en régime permanent électrique à la pulsation statorique ω_s . Une méthode souvent retenue dans l'étude des observateurs [3].
- Repère d'axes dq lié au rotor ($\theta_r = 0$). Les grandeurs évoluent en régime permanent électrique à la pulsation des courants rotoriques ω_r .

- Repère d'axes **dq** lié à l'un des flux de la machine. Le modèle est simplifié par l'utilisation d'équations plus simples. En régime permanent électrique les grandeurs du modèle sont continus, une méthode souvent utilisée dans l'étude de la commande [3].

Par la suite et dans notre travail, nous utilisons la transformation de Park liée au repère statorique.

1.5.5 Référentiel lié au stator

C'est un cas particulier de la transformation de *Park*, elle est obtenue quand le repère **dq** est confondu avec le repère **ab**, c.-à-d. en prenant $\theta_s = 0$, la transformation directe se fait alors comme suit :

$$[P_s] = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (1.10)$$

$$[P_r] = \frac{2}{3} \begin{bmatrix} \cos \theta_r & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin \theta_r & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (1.11)$$

Avec :

$$\theta_r = \theta_s - \theta$$

Où, **x** représente les tensions, courants ou flux.

La transformée inverse est de la forme :

$$[P_s]^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \quad (1.12)$$

$$[P_r]^{-1} = \begin{bmatrix} \cos \theta_r & -\sin \theta_r & 1 \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad (1.13)$$

Equations électriques

Les équations électriques de la MAS dans un repère de *Park* lié au stator seront :

Pour les grandeurs statoriques :

$$[V_{dqos}] = [r_s][I_{dqos}] + \frac{d[\phi_{dqos}]}{dt} + [P_s] \frac{d[P_s]^{-1}}{dt} [\phi_{dqos}] \quad (1.14)$$

Après calcul on obtient

$$\begin{cases} V_{ds} = R_s I_{ds} + \frac{d\phi_{ds}}{dt} \\ V_{qs} = R_s I_{qs} + \frac{d\phi_{qs}}{dt} \end{cases} \quad (1.15)$$

Avec les équations des flux statoriques :

$$[\phi_{dqos}] = [P_s][L_s][P_s]^{-1}[I_{dqos}] + [P_s][M_{sr}][P_r]^{-1}[I_{dqor}] \quad (1.16)$$

Après calcul on obtient

$$\begin{cases} \phi_{ds} = L_s I_{ds} + M_{sr} I_{dr} \\ \phi_{qs} = L_s I_{qs} + M_{sr} I_{qr} \end{cases} \quad (1.17)$$

Pour les grandeurs rotoriques:

$$[V_{dqor}] = [r_r][I_{dqor}] + \frac{d[\phi_{dqor}]}{dt} + [P_r] \frac{d[P_r]^{-1}}{dt} [\phi_{dqor}] = 0 \quad (1.18)$$

Après calcul on obtient

$$\begin{cases} V_{dr} = R_r I_{dr} + \frac{d\phi_{dr}}{dt} + p\Omega\phi_{qr} = 0 \\ V_{qr} = R_r I_{qr} + \frac{d\phi_{qr}}{dt} - p\Omega\phi_{dr} = 0 \end{cases} \quad (1.19)$$

Avec les équations des flux rotoriques:

$$[\phi_{dqor}] = [P_r][L_r][P_r]^{-1}[I_{dqor}] + [P_r][M_{rs}][P_s]^{-1}[I_{dqos}] \quad (1.20)$$

Après calcul on obtient

$$\begin{cases} \phi_{dr} = L_r I_{dr} + M_{sr} I_{ds} \\ \phi_{qr} = L_r I_{qr} + M_{sr} I_{qs} \end{cases} \quad (1.21)$$

Avec:

$L_s = l_{aa} - l_{ab}$: Inductance cyclique statorique.

$L_r = l_{AA} - l_{AB}$: Inductance cyclique rotorique.

$M_{sr} = \frac{3}{2} l_{aA}$: Inductance mutuelle cyclique entre stator et rotor [2].

Equations mécaniques

Le modèle électrique doit être complété par l'équation du couple électromagnétique et de la vitesse mécanique.

Le couple électromagnétique T_e peut se mettre sous plusieurs formes :

$$\begin{cases} T_e = \frac{3}{2} p (\phi_{ds} I_{qs} - \phi_{qs} I_{ds}) \\ T_e = \frac{3}{2} p (\phi_{qr} I_{dr} - \phi_{dr} I_{qr}) \\ T_e = \frac{3}{2} p M_{sr} (I_{qs} I_{dr} - I_{ds} I_{qr}) \end{cases} \quad (1.22)$$

$$T_e = \frac{3}{2} \frac{M_{sr}}{L_r} p (\phi_{dr} I_{qs} - \phi_{qr} I_{ds}) \quad (1.23)$$

Ou:

p : Le nombre de paires de pole.

La relation (1. 23) sera retenue, car elle dépend des variables d'état adoptées.

La vitesse de rotation mécanique est déduite de la loi fondamentale de la mécanique :

$$\sum \text{couples} = J \frac{d\Omega}{dt}$$

Elle s'écrit donc:

$$J \frac{d\Omega}{dt} = T_e - C_r - f_v \Omega \quad (1.24)$$

Où $\Omega = \frac{\omega_m}{p}$ est la vitesse mécanique de la MAS.

J : Est l'inertie de toute les masses tournantes ramenées à l'arbre de la machine.

C_r : Le couple résistant de charge.

f_v : Coefficient du frottement visqueux.

1.5.6 Le modèle de la MAS dans l'espace d'état

Maintenant on va réécrire les équations de la machine en modèle d'état, en prenant les courants statoriques, et les flux rotoriques comme variables d'états.

A partir de l'équation (1.21) on obtient

$$\begin{cases} I_{dr} = \frac{1}{L_r} \Phi_{dr} - \frac{M_{sr}}{L_r} I_{ds} \\ I_{qr} = \frac{1}{L_r} \Phi_{qr} - \frac{M_{sr}}{L_r} I_{qs} \end{cases} \quad (1.25)$$

Remplaçons les courants rotoriques (1.25) dans les équations (1.17) on obtient:

$$\begin{cases} \Phi_{ds} = \left(L_s - \frac{M_{sr}^2}{L_r} \right) I_{ds} + \frac{M_{sr}}{L_r} \Phi_{dr} \\ \Phi_{qs} = \left(L_s - \frac{M_{sr}^2}{L_r} \right) I_{qs} + \frac{M_{sr}}{L_r} \Phi_{qr} \end{cases} \quad (1.26)$$

On remplace l'équation (1.25) dans (1.19), on trouve :

$$\begin{cases} \frac{d\Phi_{dr}}{dt} = \frac{R_r M_{sr}}{L_r} I_{ds} - \frac{R_r}{L_r} \Phi_{dr} - p\Omega \Phi_{qr} \\ \frac{d\Phi_{qr}}{dt} = \frac{R_r M_{sr}}{L_r} I_{qs} - \frac{R_r}{L_r} \Phi_{qr} + p\Omega \Phi_{dr} \end{cases} \quad (1.27)$$

On remplace l'équation (1.26) dans (1.15), on trouve :

$$\begin{cases} V_{ds} = R_s I_{ds} + \left(L_s - \frac{M_{sr}^2}{L_r} \right) \frac{dI_{ds}}{dt} + \frac{M_{sr}}{L_r} \frac{d\Phi_{dr}}{dt} \\ V_{qs} = R_s I_{qs} + \left(L_s - \frac{M_{sr}^2}{L_r} \right) \frac{dI_{qs}}{dt} + \frac{M_{sr}}{L_r} \frac{d\Phi_{qr}}{dt} \end{cases} \quad (1.28)$$

A partir de l'équation (1.27) et (1.28) on obtient

$$\begin{cases} V_{ds} = \left(L_s - \frac{M_{sr}^2}{L_r} \right) \frac{dI_{ds}}{dt} + \left(R_s + \frac{R_r M_{sr}^2}{L_r^2} \right) I_{ds} - \frac{M_{sr} R_r}{L_r^2} \Phi_{dr} - \frac{M_{sr}}{L_r} p\Omega \Phi_{qr} \\ V_{qs} = \left(L_s - \frac{M_{sr}^2}{L_r} \right) \frac{dI_{qs}}{dt} + \left(R_s + \frac{R_r M_{sr}^2}{L_r^2} \right) I_{qs} - \frac{M_{sr} R_r}{L_r^2} \Phi_{qr} + \frac{M_{sr}}{L_r} p\Omega \Phi_{dr} \end{cases} \quad (1.29)$$

En rapportant ces dernières relations dans (1.27) et (1.29), on aboutira au système d'équations d'états suivant :

$$\begin{cases} \frac{d}{dt} I_{ds} = - \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) I_{ds} + \left(\frac{1-\sigma}{\sigma M_{sr} T_r} \right) \Phi_{dr} + \frac{(1-\sigma)}{\sigma M_{sr}} p\Omega \Phi_{qr} + \frac{V_{ds}}{\sigma L_s} \\ \frac{d}{dt} I_{qs} = - \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) I_{qs} - \frac{(1-\sigma)}{\sigma M_{sr}} p\Omega \Phi_{dr} + \left(\frac{1-\sigma}{\sigma M_{sr} T_r} \right) \Phi_{qr} + \frac{V_{qs}}{\sigma L_s} \\ \frac{d}{dt} \Phi_{dr} = \frac{M_{sr}}{T_r} I_{ds} - \frac{1}{T_r} \Phi_{dr} - p\Omega \Phi_{qr} \\ \frac{d}{dt} \Phi_{qr} = \frac{M_{sr}}{T_r} I_{qs} + p\Omega \Phi_{dr} - \frac{1}{T_r} \Phi_{qr} \end{cases} \quad (1.30)$$

Où :

$\sigma = 1 - \frac{M_{sr}^2}{L_s L_r}$: est le coefficient de dispersion.

$T_r = \frac{L_r}{R_r}$: Constante de temps rotorique

$T_s = \frac{L_s}{R_s}$: Constante de temps statorique.

Avec l'équation de la vitesse mécanique donnée par:

$$\frac{d}{dt}\Omega = (T_e - C_r) \frac{1}{J} - \frac{f_v}{J} \Omega \quad (1.31)$$

En rapportant la relation (1.23) dans (1.31), on aboutira à la relation de la vitesse:

$$\frac{d}{dt}\Omega = \frac{3}{2} \frac{M_{sr}}{L_r} p (\phi_{dr} I_{qs} - \phi_{qr} I_{ds}) \frac{1}{J} - (C_r + f_v \Omega) \frac{1}{J} \quad (1.32)$$

Le modèle d'état de la machine peut s'écrire sous la forme matricielle suivante:

$$\frac{d}{dt}X = AX + BU$$

Où X représente le vecteur d'état.

$$X = \begin{bmatrix} I_{ds} \\ I_{qs} \\ \phi_{dr} \\ \phi_{qr} \end{bmatrix}$$

A: La matrice de transition d'état du système.

$$A = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & 0 & \left(\frac{1-\sigma}{\sigma M_{sr} T_r}\right) & \frac{(1-\sigma)}{\sigma M_{sr}} p\Omega \\ 0 & -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & -\frac{(1-\sigma)}{\sigma M_{sr}} p\Omega & \left(\frac{1-\sigma}{\sigma M_{sr} T_r}\right) \\ \frac{M_{sr}}{T_r} & 0 & -\frac{1}{T_r} & -p\Omega \\ 0 & \frac{M_{sr}}{T_r} & p\Omega & -\frac{1}{T_r} \end{bmatrix}$$

U : est le vecteur de commande.

$$U = \begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix}$$

B : La matrice d'application de la commande.

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

1.6 Conclusion

Dans ce chapitre on a donné une brève description de la machine asynchrone triphasée et de son principe de fonctionnement. Ensuite, on a présenté le modèle mathématique de la machine triphasé. On a procédé à la transformation dans le repère de Park lié au stator, ce qui aboutit au modèle simplifié.

Pour que le modèle final se rapproche du système réel, nous considérons par la suite les bruits qui s'ajoutent au modèle d'état biphasé de la MAS.

Dans le chapitre qui suit on applique le filtre de Kalman sur le modèle de la MAS dans le but de construire une estimation des flux rotoriques.

Chapitre 2 Filtre de Kalman

2.1 Introduction

La réalisation de la commande d'un système physique passe par la connaissance de vecteur d'état à chaque instant, et la qualité de la solution apportée dépend au moins en partie de la précision avec laquelle l'état est connu. Dans le cas d'une machine asynchrone, il faut disposer de mesures des grandeurs électromagnétiques (courants, tension, flux, couple de charge). Il existe des capteurs physiques qui donnent directement ces mesures, seulement ils sont coûteux, fragiles et encombrant. Pour contourner cette difficulté, on a recouru à des algorithmes d'estimation en ligne des variables non mesurables. Ces algorithmes, dits observateurs, reposent sur la connaissance de modèle du système et sur les mesures fournies par les capteurs physiques disponibles. Parmi les observateurs les plus répandus dans la littérature, on peut citer l'observateur de Kalman qui prend en compte les bruits.

Aussi le système 'machine asynchrone' est un système fortement bruité par l'onduleur (MLI) ou les processus environnants, et donc l'utilisation du filtre de Kalman pour la reconstruction du flux rotorique est bien approprié.

2.2 L'observabilité

De façon générale, la propriété structurelle d'un système appelée observabilité est une condition nécessaire à l'existence d'un observateur.

2.2.1 Observabilité des systèmes linéaires

On considère le système linéaire suivant :

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases} \quad (2.1)$$

Ou :

$X \in R^n$: Vecteur qui représente les n variables d'états.

$u \in R^m$: Vecteur qui représente les m commandes.

$Y \in R^p$: Vecteur qui représente les p mesures.

Le système (2.1) est dit observable, si et seulement si on peut reconstruire l'état X par la seule connaissance de l'entrée u et de la sortie Y .

a Critère d'observabilité de Kalman

Le système (2.1) est observable si et seulement si la matrice d'observabilité $O_{(A,C)}$ est régulière ou encore: le rang d'observabilité du système (2.1) est égal à n ; n étant la dimension de l'espace d'état. On dit alors que la paire (A, C) est observable.

$$\text{Rang } O_{(A,C)} = \text{rang}[C \ CA \ CA^2 \ \dots \ CA^{n-1}]^T = n \quad (2.2)$$

Il s'agit d'une condition nécessaire et suffisante. Si la matrice d'observabilité $O_{(A,C)}$ est inversible (son déterminant est non nul), alors le système (2.1) est complètement observable [5] [6]

$$\det[O_{(A,C)}] = \det[C \ CA \ CA^2 \ \dots \ CA^{n-1}]^T \neq 0 \quad (2.3)$$

2.2.2 Observabilité des systèmes non-linéaire

Soit un système non linéaire de la forme :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \quad (2.4)$$

Ou $x(t) \in R^n$ représente l'état, $u(t) \in R^m$ l'entrée et $y(t) \in R^p$ la sortie. $f(\dots)$ et $h(\dots)$ sont des fonctions analytiques.

La notion d'observabilité d'un système non linéaire peut être définie à partir de la notion d'indiscernabilité d'une paire d'état, tel que le système non linéaire (2.4) est dit observable s'il n'admet pas de paires indiscernables.

Deux états initiaux $x_1(t_0) = x_1$ et $x_2(t_0) = x_2$ sont dit indiscernables pour le système (2.4) si $\forall t \in [t_0, t_1]$ les sorties correspondantes $y_1(t)$ et $y_2(t)$ sont indiscernables quelle que soit l'entrée admissible $u(t)$ du système (2.4).

Donc, contrairement au cas linéaire, l'observabilité d'un système non linéaire dépend de l'entrée appliquée. On doit donc prendre en compte le problème des entrées.

On appelle entrée universelle, l'entrée qui permet de discerner tout couple d'états initiaux par examen de la sortie. Dans le cas contraire, elle sera appelée entrée singulière. Lorsque le système non linéaire est dépourvu d'entrées singulières, il est appelé système uniformément observable [5] [6].

Les critères permettant de déterminer l'observabilité d'un système non linéaire sont plus compliqués que dans le cas linéaire. Cependant, une notion d'observabilité faible locale peut être caractérisée par une condition de rang équivalant au cas linéaire.

On dit que le système non linéaire (2.4) satisfaisant la condition de rang d'observabilité, si :

$$\forall x \in R^n : \dim(dO(x)) = n \quad (2.5)$$

Tel que :

dO : L'espace des différentielles des éléments de O .

O : L'espace d'observabilité.

L'espace d'observabilité noté ' O ' est le plus petit sous espace vectoriel de fonction de R^n à valeur dans l'espace de sortie, contenant les sorties h_1, h_2, \dots, h_p et qui soit fermé sous l'opération de la dérivation de Lie par rapport au champ de vecteur $f(x, u)$, u étant fixée.

Définition: dérivée de Lie.

Pour tout $u \in U$, notons par f_u le champ de vecteurs défini par $f_u(x) = f(x, u)$. Si φ est une fonction différentiable sur M , la dérivée de Lie de φ par rapport à f_u est notée par $L_{f_u}(\varphi)$

Dans un système de coordonnées (x_1, x_2, \dots, x_n) le champ s'écrit :

$$f_u = \sum_{i=1}^n \frac{\partial}{\partial x_i} \quad (2.6)$$

Alors :

$$L_{f_u}(\varphi) = \sum_{i=1}^n \frac{\partial \varphi}{\partial x_i} \quad (2.7)$$

Si O est un espace d'observabilité générique on peut dire que le système (2.4) est génériquement observable si seulement si :

$$\text{Dim}(O) = n \quad (2.8)$$

Cette condition est appelée condition de rang d'observabilité générique. Si cette condition est satisfaite, on peut alors vérifier :

$$\text{Rang} \begin{bmatrix} dh \\ dL_f h \\ \vdots \\ dL_f^{n-1} h \end{bmatrix} = n \quad (2.9)$$

L_f est l'opération de dérivée de Lie.

Cela implique que l'état x peut être déduit de la connaissance de la sortie et d'un nombre finie de ses dérivées.

Un seul critère est suffisant est que le Jacobien :

$$\frac{\partial(y, \dots, y^{(n-1)})}{\partial(x_1, \dots, x_n)} \quad (2.10)$$

Soit de rang plein [5] [6].

2.3 Observateurs

On appelle observateur d'état un système dynamique capable de reproduire les états non mesurés d'un système à partir de la connaissance des entrées et sorties, et éventuellement des états mesurables du système réel. [7]

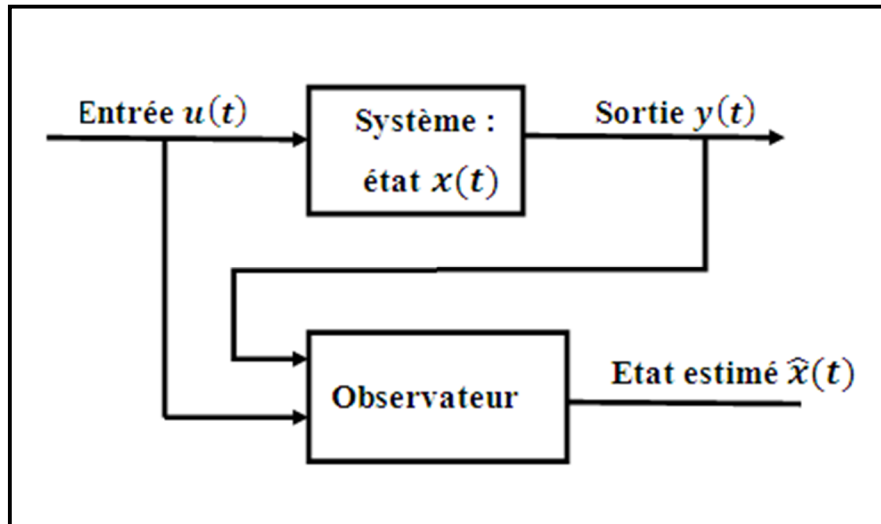


Figure : 2.1 Principe de l'observateur

L'observation se fait en deux phases ; la première est une étape d'estimation et la seconde est une étape de correction.

L'estimation se fait par le calcul des grandeurs d'états à l'aide d'un modèle du système :

$$\dot{\hat{X}}(t) = A\hat{X}(t) + Bu(t) \quad (2.11)$$

La correction se fait par l'addition de la différence entre les états estimés et ceux mesurés (erreur d'estimation) que l'on multiplie par un gain K_f :

$$\dot{\hat{X}}(t) = A\hat{X}(t) + Bu(t) + K_f(Y(t) - C\hat{X}(t)) \quad (2.12)$$

Les observateurs peuvent être divisés en deux catégories, observateurs linéaires et non linéaires.

2.3.1 Observateurs des systèmes non linéaires

Les systèmes physiques sont généralement non linéaires, les méthodes suivies pour la conception d'observateurs non linéaire sont multiples et s'inspirent des méthodes linéaires déjà citées avec des hypothèses très fortes sur la structure des non linéarité parmi ces observateurs, on cite [8]

- Observateur dits à « grand gain ».
- Le filtre de Kalman étendu.

- Observateurs adaptatifs.

2.3.2 Observateurs des systèmes linéaires

Une solution simple au problème de l'estimation de l'état des systèmes linéaires a été proposée par Luenberger dans le cadre déterministe, et par Kalman dans le cadre stochastique.

a Observateur de Lueberger

On considère le modèle dynamique d'un système linéaire défini comme suit :

$$\begin{cases} \dot{X}(t) = AX(t) + BU(t) \\ Y(t) = CX(t) \end{cases} \quad (2.13)$$

Où $t \geq 0, X(t) \in R^n, U(t) \in R^m, Y(t) \in R^p$

Les matrices A, B et C du système (2.13) sont constantes et de dimensions appropriées.

On se place dans le cadre déterministe (les bruits sont négligés). L'observabilité du système (2.13) doit être confirmée en premier lieu, par le critère de Kalman.

La structure de l'observateur linéaire de Luenberger est celle indiquée sur la figure (figure 2.2). Elle fait intervenir tout d'abord un estimateur fonctionnant en boucle ouverte qui est caractérisé par la même dynamique que celle de système. La structure fonctionnant en boucle fermée obtenue par l'introduction d'une matrice de gain L permet d'imposer la dynamique propre à cet observateur.

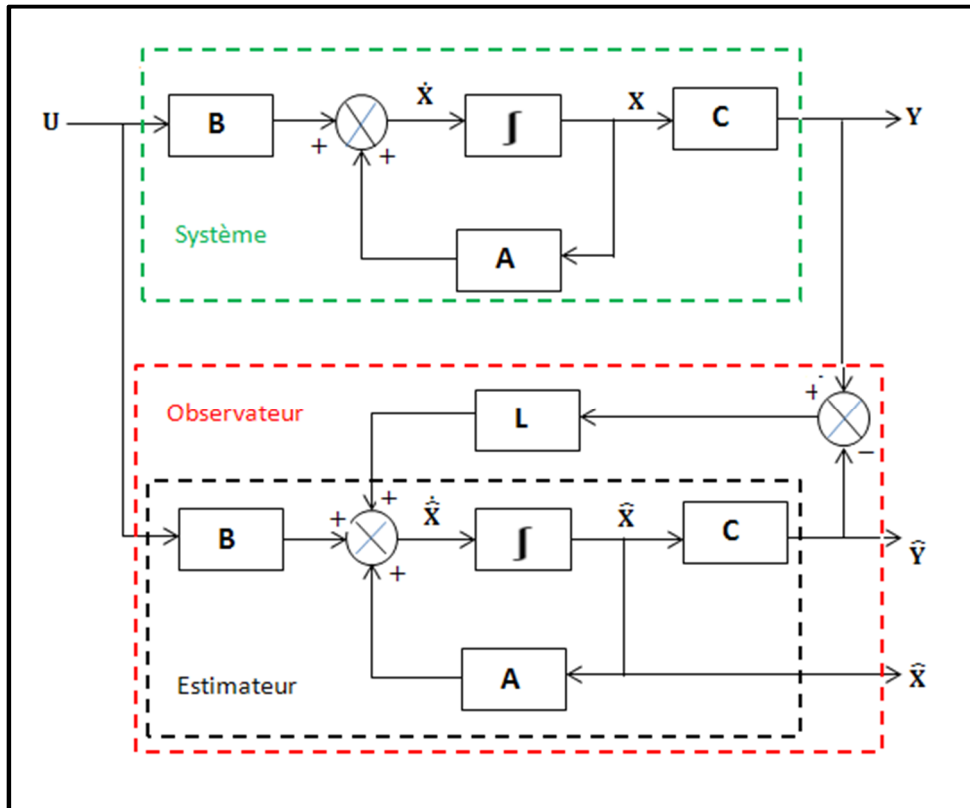


Figure : 2.2 Schéma fonctionnel de l'observateur de Luenberger

U : représente le vecteur d'entrée du système réel (2.13) et de l'observateur.

X : représente le vecteur d'état constitué des grandeurs à observer.

Y : représente le vecteur de sortie dans les composantes sont mesurables.

Le dernier vecteur (Y) est comparé au vecteur équivalent (\hat{Y}) donné par l'observateur pour assurer le fonctionnement en boucle fermée.

$$\varepsilon = (\hat{Y} - Y) = C(\hat{X} - X) \quad (2.14)$$

Cette différence est multipliée par une matrice de gain L et envoyée à l'entrée de l'observateur pour influencer les états estimés.

L'observateur de Luenberger est décrit par l'équation suivante :

$$\hat{\dot{X}}(t) = (A - LC)\hat{X}(t) + BU(t) \quad (2.15)$$

Ainsi, par un choix judicieux de la matrices de gain L tel que les valeurs propres de $(A - LC)$ soient à parties réelles strictement négatives, on peut modifier la dynamique

é de l'observateur, et par conséquent faire évoluer la vitesse de convergence de l'erreur vers zéros plus au moins rapidement [5].

b Le filtre de Kalman

Le filtre de kalman est un observateur d'état optimal pour un contexte stochastique défini : il permet la reconstruction de l'état d'un système à partir des signaux d'entrer et de mesures, à l'aide de son modèle dynamique échantillonné. Le vecteur d'état de processus est l'information que l'on cherche à extraire du bruit additionnel. Le filtre de Kalman est un observateur donnant la prédiction du vecteur d'état. Dans un environnement stochastique lorsque les variances des bruits sont connues, c'est un observateur optimal .de plus, si les bruits peuvent être considéré comme blancs et gaussiens, il est optimal, c'est -à-dire que de tous les observateurs, c'est celui dont la variance de l'erreur d'observa on sera la plus faible [9].

2.3.3 Synthèse du filtre de Kalman

Dans le cadre non déterministe, on définit le système dynamique par le modèle d'équations suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Mw(t) \\ y(t) = Cx(t) + v(t) \end{cases} \quad (2.16)$$

$w(t)$ et $v(t)$ sont deux bruits blancs.

La théorie de l'observateur de Kalman nécessite, quant à elle, la résolution d'une équation de Riccati et utilise les propriétés statiques des bruits w et v et propose une structure d'observateur dans le cas continu et discret :

a Cas des systèmes continus

Soit le modèle stochastique continu décrit par le système d'équations d'état et de mesure (2.16) :

H1 : la paire (A, C) est détectable, c'est -à-dire qu'il n'y a pas de mode instable et inobservable dans le système (2.16).

H2 : les signaux $w(t)$ et $v(t)$ sont des bruits blancs gaussiens centrés de Densité Spectrale de Puissance (DSP) W et V respectivement, c'est-à-dire :

$$\begin{cases} E[w(t)w(t + \tau)^T] = W\delta(\tau) \\ E[v(t)v(t + \tau)^T] = V\delta(\tau) \\ E[w(t)v(t + \tau)^T] = 0 \end{cases}$$

Cette dernière relation traduit l'indépendance stochastique des bruits $w(t)$ et $v(t)$.

H3 : V est inversible (il y a autant de sources de bruits blancs indépendantes que de mesures dans l'équation de mesure).

Le filtre de Kalman est décrit par :

$$\begin{cases} \dot{\hat{x}}(t) = A_f \hat{x}(t) + B_f u(t) + K_f y(t) \\ \hat{y}(t) = C \hat{x}(t) \end{cases} \quad (2.17)$$

Détermination des matrices A_f et B_f du filtre de Kalman :

On définit l'erreur de l'estimation $\varepsilon(t)$ par :

$$\varepsilon(t) = x(t) - \hat{x}(t) \quad (2.18)$$

A partir de l'équation d'état du modèle (2.16) et celle du filtre (2.17), on obtient l'équation de la dynamique de l'erreur d'estimation :

$$\begin{aligned} \dot{\varepsilon}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= (A - K_f C)\varepsilon(t) + (A - K_f C - A_f)\hat{x}(t) + (B - B_f)u(t) + Mw(t) - K_f v(t) \end{aligned} \quad (2.19)$$

On souhaite que la moyenne de l'erreur d'estimation tende vers 0 lorsque t tend vers l'infini.

Les bruits w et v étant centrés, on a :

$$E[\dot{\varepsilon}(t)] = (A - K_f C)E[\varepsilon(t)] + (A - K_f C - A_f)E[\hat{x}(t)] + (B - B_f)u(t) \quad (2.20)$$

Alors :

$\forall u(t), \forall E[\hat{x}(t)] : \lim_{t \rightarrow \infty} E[\varepsilon(t)] = 0$, si et seulement si :

1. $A_f = A - K_f C$
2. $B_f = B$
3. $A - K_f(t)C$ soit stable

La dynamique de l'erreur d'estimation devient.

$$\dot{\varepsilon}(t) = (A - K_f(t)C)\varepsilon(t) + Mw(t) - K_f(t)v(t) \quad (2.21)$$

Et l'équation du filtre de Kalman s'écrit

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K_f(y(t) - C\hat{x}(t)) \quad (2.22)$$

Détermination du gain du filtre de Kalman

On cherche donc le gain de FK K_f qui minimise le critère :

$$\begin{aligned} J_k(t) &= \sum_{i=0}^n E[\varepsilon_i(t)^2] \\ &= E[\varepsilon(t)^T \varepsilon(t)] \\ &= \text{trace} E[\varepsilon(t)\varepsilon(t)^T] \\ &= \text{trace} P(t) \end{aligned} \quad (2.23)$$

Pour minimiser $\text{trace} P(t)$, il suffit de minimiser $\dot{P}(t)$.

Et :

$$\begin{aligned} P(t) &= E \left[(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T \right] \\ &= e^{A_f(t-t_0)} \left[P(t_0) + \int_{t_0}^t e^{A_f(t_0-\tau)} [M \quad -K_f] \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} M^T \\ -K_f^T \end{bmatrix} e^{A_f(t_0-\tau)} d\tau \right] e^{A_f(t-t_0)} \end{aligned} \quad (2.24)$$

Après calcul :

$$\dot{P}(t) = (A - K_f C)P(t) + P(t)(A - K_f C)^T + MWM^T + K_f V K_f^T \quad (2.25)$$

C'est l'équation différentielle de Riccati.

Cherchons $K_f(t)$ qui minimise $trace \dot{P}(t)$.

$$\frac{\partial (trace \dot{P}(t))}{\partial K_f} = -P(t)C^T - P(t)C^T + 2K_fV \quad (2.26)$$

En annulant l'équation (2.26), nous obtenons :

$$K_f = P(t)C^T V^{-1} \quad (2.27)$$

b Cas des systèmes discrets

Considérons le système stochastique (2.16), une discrétisation de ce dernier est nécessaire avant de considérer le filtre de Kalman (discret). Après calcul, on obtient le système d'équations d'état discret suivant :

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + M_d w_d(k) \\ y(k) = C_d x(k) + v_d(k) \end{cases} \quad (2.28)$$

Avec :

$$\begin{cases} A_d = e^{At_e} = I_n + At_e + \frac{(At_e)^2}{2} + \dots + \frac{(At_e)^n}{n!} \\ B_d = \int_0^{t_e} e^{At} B dt = A^{-1} (e^{At} B) \Big|_0^{t_e} = \left(I_n t_e + \frac{At_e^2}{2} + \dots + \frac{A^{n-1} t_e^n}{n!} \right) B \\ W_d = \int_0^{t_e} e^{At} M W M^T e^{A^T t} dt \\ V_d = \frac{V}{t_e} \\ M_d = I_n \\ C_d = C \end{cases} \quad (2.29)$$

Et :

M_d : représente la matrice identité (n x n).

t_e : représente la période d'échantillonnage.

I_n : Matrice identité (n x n).

Nous supposons que :

H1 : la paire (A_d, C_d) est détectable.

H2 : les signaux $w_d(k)$ et $v_d(k)$ sont des bruits pseudo-blancs gaussiens centrés de matrices de covariance W_d et V_d respectivement, c'est-à-dire :

$$\begin{cases} E[w_d(k)w_d(k + \tau)^T] = W_d\delta(\tau) \\ E[v_d(k)v_d(k + \tau)^T] = V_d\delta(\tau) \\ E[w_d(k)v_d(k + \tau)^T] = 0 \end{cases}$$

H3 : V_d est inversible.

Le filtre de Kalman opère donc en deux étapes d'une manière récursive, après une étape d'initialisation :

1. Une étape de prédiction des variables d'état du système.

2. Une étape de correction de la prévision une fois qu'une nouvelle observation est disponible [10].

Pour déterminer l'état estimé final $\hat{x}(k + 1/k + 1)$, on doit calculer la prédiction $\hat{x}(k + 1/k)$ de $x(k + 1)$, cette valeur prédite sera exprimée par l'équation du modèle déterministe :

$$\hat{x}(k + 1/k) = A_d\hat{x}(k/k) + B_d u(k) \quad (2.30)$$

Où $\hat{x}(k/k)$ représente l'état estimé à l'instant k .

On associe à $P(k + 1/k)$ la matrice de covariance de l'erreur de prédiction notée :

$$P(k + 1/k) = E \left[(x(k + 1) - \hat{x}(k + 1/k))(x(k + 1) - \hat{x}(k + 1/k))^T \right] \quad (2.31)$$

Après calcul :

$$P(k + 1/k) = A_d P(k/k) A_d^T + M_d W_d M_d^T \quad (2.32)$$

Où :

$P(k/k)$: représente la matrice de la covariance de l'erreur d'estimation.

La covariance $P(k + 1/k)$ indique le degré de confiance que l'on a dans la prédiction $\hat{x}(k + 1/k)$.

Comme dans le cas continu on cherche $K_f(k + 1)$ qui minimise $\text{trace } P(k + 1/k + 1)$

$$K_f(k + 1) = P(k + 1/k)C_d^T(C_dP(k + 1/k)C_d^T + V_d)^{-1} \quad (2.33)$$

L'estimé final $\hat{x}(k + 1/k + 1)$ s'exprimera par:

$$\hat{x}(k + 1/k + 1) = \hat{x}(k + 1/k) + K_f(k + 1)(Y(k) - C_d\hat{x}(k + 1/k)) \quad (2.34)$$

Et la matrice de covariance de l'erreur d'estimation doit être aussi à jours, on trouve donc :

$$P(k + 1/k + 1) = E \left[(x(k + 1) - \hat{x}(k + 1/k + 1))(x(k + 1) - \hat{x}(k + 1/k + 1))^T \right] \quad (2.35)$$

On obtient le résultat suivant :

$$P(k + 1/k + 1) = (I_n - K_f(k + 1)C_d)P(k + 1/k) \quad (2.36)$$

Les étapes du FK décrites auparavant sont résumées dans le diagramme de la figure 2.3 :

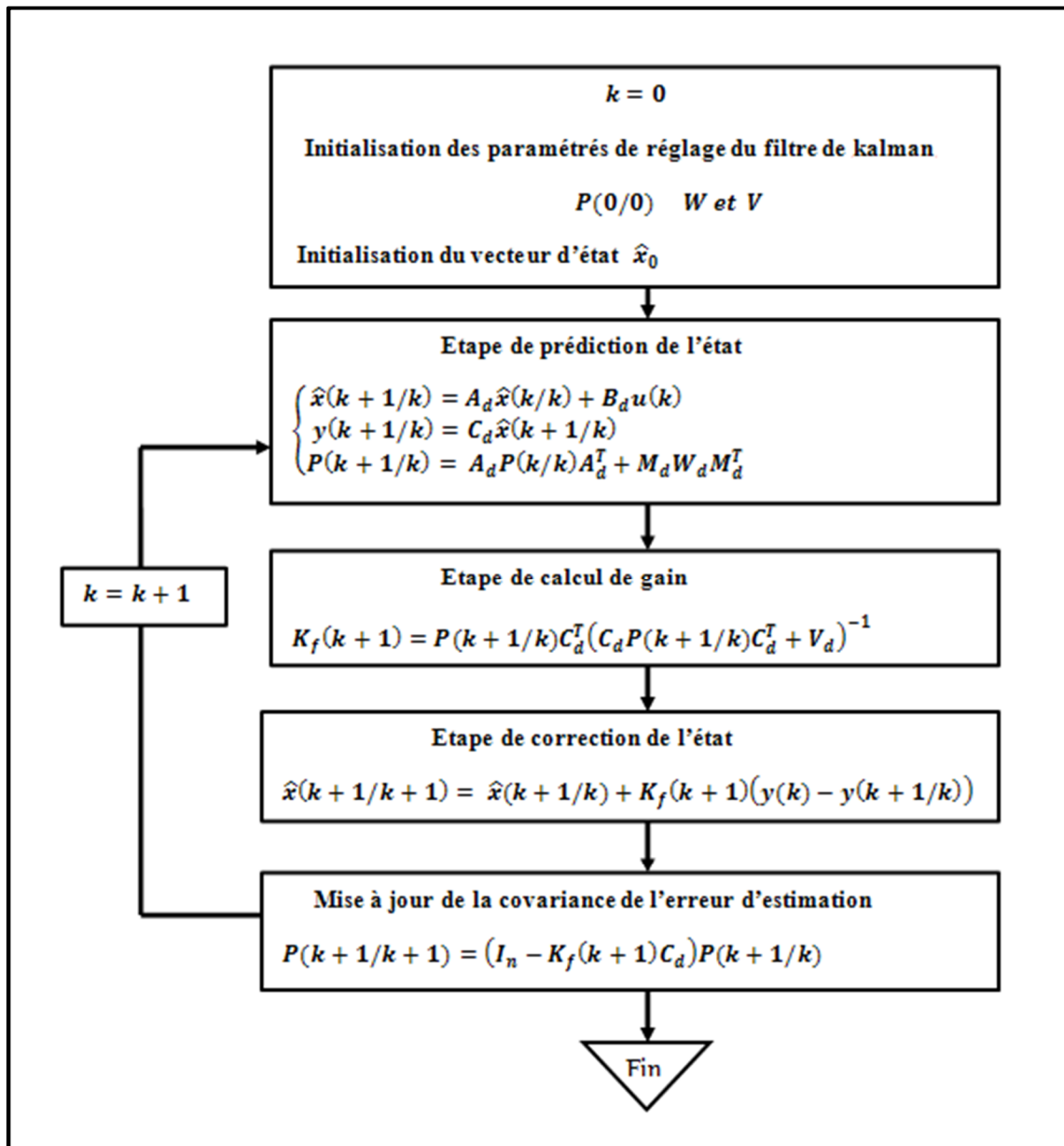


Figure : 2. 3 L'algorithme du filtre de kalman

2.4 Application du FK à la MAS

Notre travail se basera sur l'algorithme du FK effectué pour estimer les grandeurs électriques et magnétiques (courants statoriques et flux rotoriques) de la MAS.

La machine asynchrone est alimentée par une alimentation triphasée. Après avoir modélisé et appliqué la transformation de Park à cette dernière, on obtient le modèle stochastique décrit par l'ensemble d'équations suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Mw(t) \\ y(t) = Cx(t) + v(t) \end{cases} \quad (2.37)$$

w et v sont deux bruits blancs de covariance W et V.

Avec :

$$A = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & 0 & \left(\frac{1-\sigma}{\sigma M_{sr} T_r}\right) & \frac{(1-\sigma)}{\sigma M_{sr}} p\Omega \\ 0 & -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & -\frac{(1-\sigma)}{\sigma M_{sr}} p\Omega & \left(\frac{1-\sigma}{\sigma M_{sr} T_r}\right) \\ \frac{M_{sr}}{T_r} & 0 & -\frac{1}{T_r} & -p\Omega \\ 0 & \frac{M_{sr}}{T_r} & p\Omega & -\frac{1}{T_r} \end{bmatrix} \quad (2.38)$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.39)$$

Les courants statoriques I_{ds} et I_{qs} sont les variables d'états mesurées

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.40)$$

$$u = \begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} \quad (2.41)$$

$$x = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \phi_{dr} \\ \phi_{qr} \end{bmatrix} \quad (2.42)$$

$$y = \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} \quad (2.43)$$

L'implémentation du FK sera faite sur un DSP. Nous allons donc considérer que la sortie du modèle continu est échantillonnée à la cadence t_e et nous supposerons que

des bloqueurs d'ordre 0 sont placés sur les signaux déterministes u (CAN) et nous allons chercher une représentation d'état discrète de ce modèle.

2.4.1 Modèle discret du filtre de Kalman

Le modèle de la machine asynchrone étant un modèle d'état continu et les systèmes temps réel utilisent des modèles discrétisés des processus, il est important d'obtenir l'équation d'état discrétisée du système.

Par analogie directe avec le cas continu le modèle de la MAS discret s'écrit :

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + M_d w_d(k) \\ y(k) = C_d x(k) + v_d(k) \end{cases} \quad (2.44)$$

Avec :

$$\begin{cases} A_d = e^{At_e} \\ B_d = \int_0^{t_e} e^{At} B dt \\ C_d = C \\ W_d = \int_0^{t_e} e^{At} M W M^T e^{A^T t} dt \\ V_d = \frac{V}{t_e} \end{cases} \quad (2.45)$$

t_e : Temps d'échantillonnage.

I_4 : Matrice identité 4x4.

M_d : Matrice identité 4x4

La matrice A_d se calcule par :

$$A_d = \frac{(At_e)^3}{6} + \frac{(At_e)^2}{2} + At_e + I_4 \quad (2.46)$$

La matrice B_d se calcule par :

$$B_d = B \left(I + \frac{At_e}{2} + \frac{(At_e)^2}{6} \right) t_e \quad (2.47)$$

Les matrices A_d et B_d sont des fonctions du troisième ordre par rapport à la vitesse de rotation du moteur.

Remarque

- La vitesse rotorique Ω est supposée constante entre deux instants d'échantillonnage.
- Les courants rotorique ainsi que la vitesse mécanique sont mesurable, les flux rotoriques seront estimés.
- Un développement limité de e^{At_e} à l'ordre trois avec une période d'échantillonnage t_e égale à 10^{-4} s.

Le filtre de Kalman est une copie du modèle d'état linéaire qui met en équation l'évolution du signal utile, et sa relation avec le signal mesuré. Le filtre obtenu par kalman est récursif, sa réponse en chaque instant n'est en effet fonction que du signal observé en son entrée et de sa réponse à l'itération précédente. Ainsi, le filtre de Kalman ne nécessite pas toutes les données passées pour produire une estimation à l'instant courant. Il ne nécessite donc pas de mise en mémoire des données. Cet avantage rend possible l'implémentation du FK sur un DSP.

2.5 Conclusion

Dans ce chapitre, nous avons rappelé quelques notions relatives à l'observabilité et aux observateurs des systèmes linéaires et non linéaires.

Notre étude s'est basée sur la synthèse du filtre de Kalman, observateur stochastique qui prend en compte les bruits de mesure et les bruits d'état.

L'algorithme du filtre de Kalman obtenu sera appliqué à la MAS et implémenté par la suite sur le DSP TMS320F2812.

Chapitre 3 Implémentation du FK sur DSP

3.1 Introduction

De nos jours le DSP (Digital Signal Processor) présente une puissance importante de calcul et d'implémentation des techniques numériques de traitement de signal. Cette puissance est due à un ensemble de fonctions spéciales et un jeu d'instructions optimisé pour le traitement numérique du signal. En effet, tout système fondé autour d'un DSP bénéficie des avantages dérivants de ses particularités architecturales et de programmation : rapidité, flexibilité et la réduction du coût.

Le DSC (Digital Signal Controller) est une nouvelle version de microcontrôleur dont la puissance de traitement est délivrée par un cœur DSP relié à des périphériques embarqués dans la même puce.

3.2 Architecture des DSP

L'architecture d'un DSP est un élément important qui conditionne directement ces performances [11]. Il existe deux types de structures fondamentales Figure 3.1, dites de « Von Neuman », et de « Harvard ».

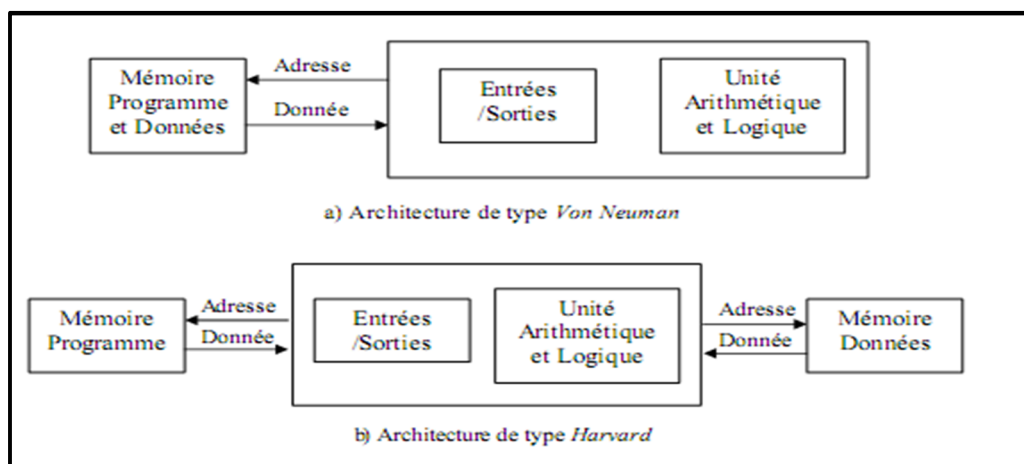


Figure : 3.1 Représentation schématique des structures de Von-Neuman et de Harvard

3.2.1 Structure de Von Neuman

La structure de Von Neuman consiste à mettre les données et le programme dans la même zone mémoire, et donc utiliser un seul bus de données et d'adresses pour les données et le programme ; on ne peut lire, en un seul cycle d'horloge, qu'une donnée ou une instruction.

3.2.2 Structure de Harvard

Cette structure se distingue de celle de Von Neuman par le fait que les mémoires programme et données sont séparées. L'accès à chacune des deux mémoires se fait via deux chemins distincts. Cette organisation permet de transférer une instruction et une donnée simultanément, elle est considérée comme deux fois plus rapide que celle de Von Neuman, mais ce gain de vitesse se fait au prix d'une sérieuse complication de l'électronique puisque toute l'architecture des bus est doublée.

Pour réduire le coût de la structure Harvard, certains DSP utilisent la structure de Harvard modifiée. A l'extérieur, le DSP ne propose qu'un seul bus de donnée et un d'adresse, comme la structure Von Neuman. Toutefois, à l'intérieur, la puce DSP dispose de deux bus distincts de données et de deux bus distincts d'adresses. Le transfert des données entre les bus internes et les bus externes est effectué par un multiplexage temporel.

3.3 Principales caractéristiques d'un DSP

3.3.1 La fonction MAC

Le DSP diffère des autres processeurs par l'opération MAC (*MULTIPLY AND ACUMULATE*), ou un simple processeur nécessite plusieurs cycles d'horloge pour effectuer un tel calcul, elle est réalisée en un seul cycle d'horloge dans les DSP.

Le premier opérande de la multiplication est chargé dans la mémoire donnée tandis que le deuxième dans la mémoire programme, la multiplication étant une opération asynchrone le résultat (de la multiplication) sera additionné au contenu de l'accumulateur le tout en un seul cycle d'horloge.

3.3.2 Double accès à la mémoire

Cette caractéristique est assurée grâce à la séparation entre le bus de donnée et le bus de programme ce qui permet de récupérer deux données de la mémoire simultanément, ce type d'opérations simultanées est généralement limité à des instructions spéciales, qui utilisent un mode d'adressage restreint, c'est-à-dire ne portant que sur la mémoire vive intégrée au DSP [11].

Une autre caractéristique de DSP est la puissance de calcul, elle est proportionnelle à la fréquence d'horloge et à la largeur de bus de donnée.

3.3.3 Le pipeline

Pour atteindre de bonnes performances de vitesse, les DSP utilisent le parallélisme et une certaine forme de pipeline. Le pipeline consiste à découper chaque instruction en certain nombre de tâches élémentaires ordonnées en phases :

Fetch : aller chercher l'instruction en mémoire programme.

Decode : réaliser le décodage de l'instruction, et des adresses des opérandes.

Read : lire les opérandes en mémoire de données.

Execute : exécuter l'opération et écrire le résultat.

Elles sont exécutées sur des ressources différentes du DSP. Il s'agit en fait du principe de travail à la chaîne. Plusieurs instructions peuvent donc s'exécuter en parallèle sur le processeur, qui occupe chacune des phases différentes [11].

3.4 Les types de DSP

La classification la plus générale des DSP est la classification selon le format, format virgule fixe et format virgule flottante. Les DSP sont conçus pour travailler avec un de ces formats. Toutefois, ils peuvent faire les calculs avec un deuxième format mais d'une façon peu efficace [11].

3.4.1 Représentation binaire des nombres fractionnaires en format virgule fixe

On appelle représentation en virgule fixe des nombres fractionnaires, ou plus généralement des nombres réels avec une précision finie, une représentation comprenant une partie entière suivie d'une partie fractionnaire correspondante à des bits après la virgule.

On utilise l'expression « format Q » pour indiquer une représentation comportant k bit après la virgule.

Un nombre réel étant rarement en précision finie, sa représentation sur un nombre fini de bits introduit une erreur, en virgule fixe sur N bits avec un format Q, cette erreur est inférieure à 2^{-k} [11].

a Les valeurs extrêmes en virgule fixe sur N bits avec un format Q_k

Les valeurs extrêmes représentables sont :

$$\text{Max} = 2^{N-1-k} - 2^{-k}$$

$$\text{Min} = -2^{N-1-k}$$

Le tableau suivant donne la représentation en format Q_5 sur 8 bits de quelques réels et leurs équivalents en décimal.

Représentation binaire Virgule fixe, format Q_5	Valeur décimale
011 10000	3.5
001 10100	1.625
110 10001	-1.046875
100 00000	-4
011 11111	3.96875

Tableau : 3.1 représentations en format fixe Q_5 pour $N=8$ bits

Les valeurs extrêmes représentables sont -4 et 3.96875 , l'erreur commise est 0.03125 .

b L'addition de nombres fractionnaires en virgule fixe

Lors de l'addition de nombres fractionnaires en virgule fixe, il faut comme en décimal aligner les virgules. La somme de 2 nombres en format Q_k donne un résultat en format Q_k .

$$Q_k + Q_k \Rightarrow Q_k$$

c Multiplication de 2 nombres fractionnaires en virgules fixe

Le produit de 2 nombres en virgules fixe sur N bits donne un résultat sur $2N - 1$ bits.

Comme en décimal, le nombre de bits après la virgule du résultat est égal à la somme des nombres de bits derrière la virgule des 2 opérandes.

$$Q_k * Q_k \Rightarrow Q_{k+k}$$

3.4.2 Représentation binaire des nombres fractionnaires en format virgule flottante

Dans la représentation binaire en virgule flottante en précision finie sur N bits, les nombres sont représentés par une mantisse M et un exposant E :

$$x = M * 2^E$$

Pour une représentation sur N bits, la mantisse M est exprimée sur m bits, et l'exposant E sur e bits, avec $N = m + e$.

a Plage des nombres représentables en format virgule flottante

Sur N bits, avec m bits pour la mantisse entre 0.5 et 1 , et e bits pour l'exposant. On peut représenter des nombres dont la valeur absolue est comprise dans l'intervalle :

$$\left[\frac{1}{2} 2^{-(2^{e-1})}, (1 - 2^{1-m}) 2^{2^{e-1}-1} \right]$$

Overflow si $|x| > (1 - 2^{1-m}) 2^{2^{e-1}-1}$

Underflow si $|x| > \frac{1}{2}2^{-(2^{e-1}-1)}$

Dans ce cas la précision relative est à peu près constante, et l'erreur de représentation est inférieure à $\frac{1}{2}2^{-(2^{e-1}-1)}2^{2-m}$

Le tableau suivant donne la représentation en virgule flottante de quelques réels et leurs équivalents en décimal.

Représentation binaire Mantisse M, exposant E	Valeur décimale $M 2^E$
01 110 010	1.75
01 100 100	0.046875
100 10 011	-7
01 000 100	0.03125
01 111 011	7.5

Tableau : 3.2 représentation en virgule flottante sur 8 bits avec $m = 5$ et $e = 3$.

3.5 La famille TMS320

La famille des DSP TMS320, produite par Texas Instruments, se compose de processeurs à virgule fixe, à virgule flottante et de mul processeurs[11]. L'architecture des DSP TMS320 est adaptée au traitement de signal en temps réel, elle offre des approches plus adaptables aux problèmes de traitement de signal, les caractéristiques suivantes font de cette famille le choix idéal pour une large gamme de traitement d'applications :

- Ensemble d'instructions très flexible.
- Flexibilité opérationnelle inhérente.
- Exécution à grande vitesse.

Cette famille contient trois plates - formes principales selon l'application, et ces plates-formes à leur tour englobent plusieurs familles.

- La plate-forme C6000, spécialisé dans le traitement d'image et le traitement de la parole est formée de familles C62x et C67x :

C62x : une famille de DSP qui travaille en 16 bits, format virgule fixe. Elle est utilisée dans les applications exigeantes en vitesse telle que les stations de base des réseaux de télécommunications mobiles, liaison ADSL et les équipements de radio diffusion.

C67x : des DSP de 32 bits à virgule flottante, architecture VLIW, d'une puissance qui varie entre 600 MFLOPS jusqu'à 1 GFLOPS, utilisé dans les applications exigeantes en précision, dynamique et vitesse telle que les antennes adaptives des stations de base, imagerie médicale, graphismes.

- La plate-forme C5000 : on trouve dans celle-là la famille C54x, optimisée en consommation, utilisée dans les applications de télécommunication exigeante en coût et en consommation telle que les terminaux mobiles.
- La plate - forme C2000, optimisée pour les applications de contrôle, elle est formée de trois familles :

C20x : des DSP de 16 bits, virgule fixe, destinés aux applications de grand volume telle que la téléphonie, l'électronique professionnelle, les appareils photos.

C24x : des DSP de 16 bits virgule fixe, utilisés dans l'automatique, robotique et le contrôle de moteur.

C28x : des DSP de 32 bits virgule fixe, utilisés dans les mêmes applications que celle de la famille C24x [11].

3.6 Le TMS320F2812

Le TMS320F2812 est un DSP contrôleur de 32 bit à virgule fixe avec la mémoire flash intégrée. L'unité centrale de traitement fonctionne avec un signal d'horloge de

150 MHz. La génération des DSP C28x est efficace en exécutant des programmes en C et en C++, ce qui permet de développer des algorithmes de commande dans des langages de niveau élevé. En plus de la mémoire flash, le F2812 embarque sur la même puce des périphériques utilisés pour la commande et la communication, telle qu'un gestionnaire d'évènement (générateur des signaux PWM), un convertisseur analogique numérique, des entrées / sorties numériques à usage général, et une interface de communication série. La figure 3.2 présente le schéma bloc des périphériques du DSC F2812 [12].

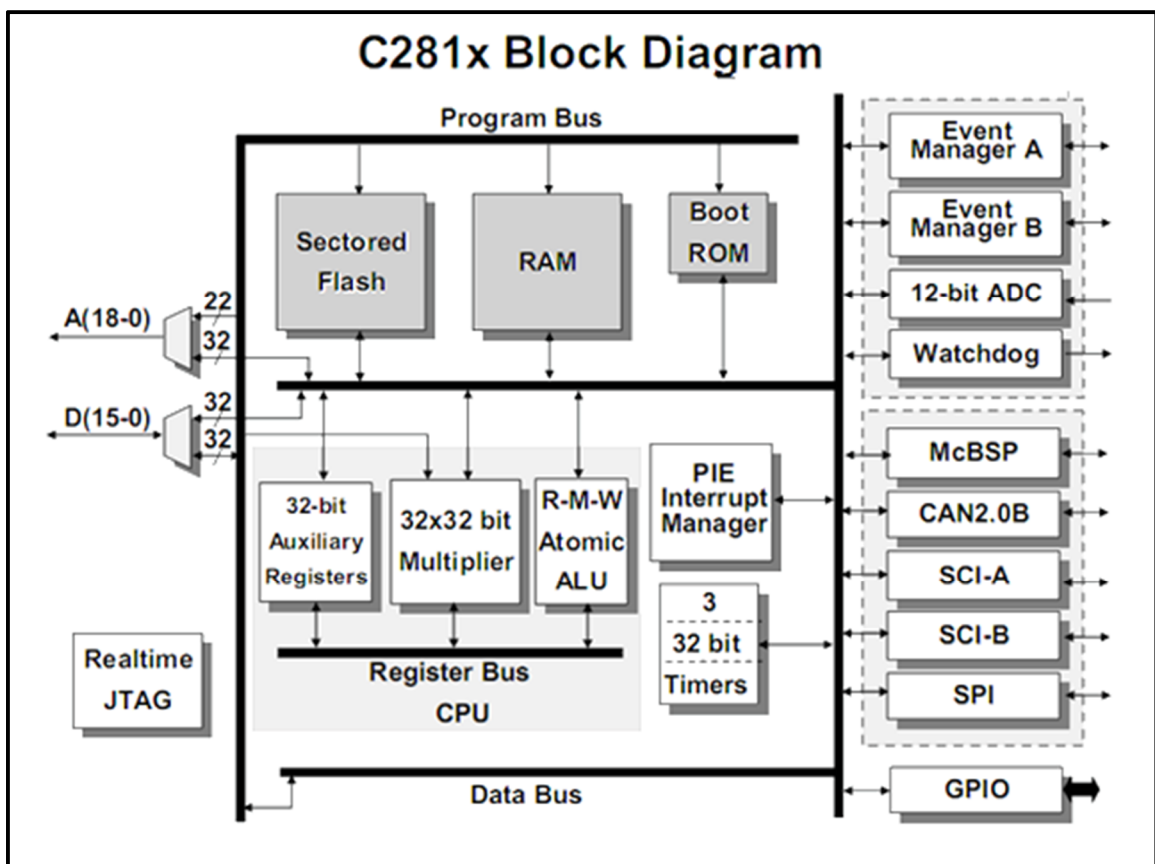


Figure : 3.2 Schéma bloc des périphériques du DSC F2812

Le DSP F2812 utilise un bus d'adresse de donnée de 32 bits et un bus d'adresse de programme de 22 bits. Ceci tient compte d'une extension pour une mémoire donnée totale de 4G mots, et une mémoire programme totale de 4M mots. La carte mémoire pour le TMS320F2812 est montrée sur la figure 3.3, elle est divisée en trois segments : mémoire programme/donnée, vecteurs d'interruption d'unité centrale de traitement et mémoire réservée [12].

La mémoire programme/donnée de TMS320F2812 est composée d'une mémoire non volatile (flash), de cinq zones (M0, M1, L0, L1 et H0) de mémoire à accès unique par cycle (SARAM), d'une mémoire de démarrage (BOOTROM) et d'une mémoire OTP (One Time Programmable), certaines de ces zones mémoires (L0 SARAM, L1 SARAM, OTP, mémoire flash) sont protégées par des mots de passe (Code Security Module) [12].

Tous les registres de périphériques sont regroupés dans des frames périphériques (PF0, PF1, PF2), le frame PF0 inclut des registres pour commander la vitesse interne de la mémoire flash ; le frame PF1 contient la plupart des registres des périphériques alors que le frame PF2 est réservé pour le bloc du registre CAN.

La figure 3.3 présente l'organisation générale de la mémoire.

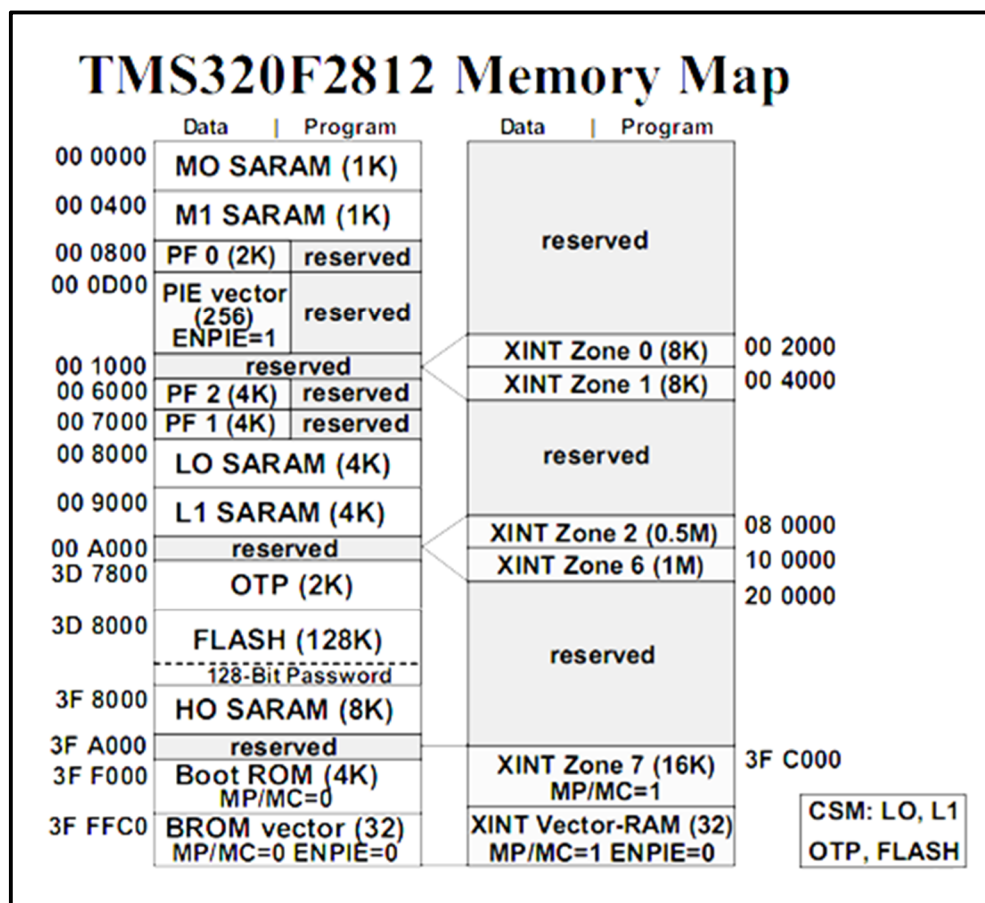


Figure : 3.3 La carte mémoire du TMS320F2812

3.6.1 L'unité centrale (cœur du DSP)

L'unité centrale du DSP (CPU) se compose d'une unité arithmétique et logique, d'un multiplicateur hardware de 32 bits, trois temporisateurs 32 bits pour des utilisations spécifiques, des registres auxiliaires, JTAG et un gestionnaire d'interruption [13].

a Le multiplieur

La multiplication dans le F2812 est le produit de deux opérandes l'un chargé de la mémoire et l'autre logé dans le registre XT. Si l'opérande de XT provient de la mémoire donnée et le deuxième opérande de la mémoire programme la multiplication aura lieu en un seul cycle. Le résultat de la multiplication est chargé dans le registre P (produit), ou directement dans l'accumulateur. Pour le produit de 32x32 bits qu'est 64 bits la moitié de mot sera stocké dans l'accumulateur et l'autre moitié dans le registre P [13].

b L'unité arithmétique et logique UAL

Elle assure le reste des opérations mathématiques. L'accumulateur contient toujours le premier opérande de l'opération tandis que le deuxième peut être chargé de la mémoire (donnée/programme), du registre P ou directement à partir de l'unité de multiplication [13].

c Les shifters

Trois unités de décalage peuvent être employées en parallèle à d'autres unités de CPU, elles sont habituellement utilisées dans les calculs intermédiaires ou la multiplication/la division par 2^n [13].

d L'interface d'émulation JTAG

Le JTAG (joint test action group) est un outil très puissant d'échange de données en temps réel entre le DSP et un poste externe. L'opérateur peut accéder et changer le contenu de différents registres, périphériques, et mémoire tandis que le programme s'exécute.

Une application indispensable dans la phase de développement qui est l'exécution pas à pas du programme, est assuré aussi par l'interface JTAG [13].

e Le gestionnaire d'interruption

Le gestionnaire d'interruption (PIE) permet une réponse rapide aux différentes interruptions générées par des signaux externes ou bien par des événements internes [13].

f Les temporisateurs

Le timer 0 est utilisé pour cadencer les périphériques, compter les événements, générer des impulsions de commande, et interrompre le CPU, tandis que les timers 1 et 2 sont utilisés dans le système "DSP/BIOS" [13].

g L'unité arithmétique et logique auxiliaire et ces registres

L'ARAU (Auxiliary Register Arithmetic Unit) génère des adresses données quand une instruction emploie l'adressage indirect, elle est associée à huit registres de 32 bits utilisés pour le stockage temporaire des adresses et des données [13].

3.6.2 Les modes d'adressage

Les modes d'adressages déterminent la façon d'accéder à la mémoire. Ils spécifient aussi la manière avec laquelle on manipule les données, comme la recherche d'un opérande indirectement à partir d'une position mémoire.

a L'adressage direct

Dans le mode d'adressage direct, le processeur emploie les 16 bits de registre DP (Data Page Pointer) pour sélectionner une page de la mémoire et emploie 6 autres (retirer de l'instruction) pour trouver une adresse particulière à cette page. L'avantage de ce mode est la possibilité d'accéder à n'importe quelle page dans n'importe quel ordre.

Pour adresser une case mémoire dans une autre page on sera obligé de changer la valeur de registre DP.

***b* L'adressage indirect**

Le mode d'adressage indirect utilise 8 registres de 32 bits (XARn) pour pointer les cases mémoires.

3.6.3 Les interruptions du TMS320F2812

L'interruption, qui est une rupture de séquence asynchrone, est générée suite à un événement interne ou externe (le plus probable). Lors d'une interruption, le traitement courant du processeur est arrêté de manière à exécuter la tâche associée; il s'agit d'une routine d'interruption surveillée par le service des routines d'interruption (ISR); et cela tout en sauvegardant les conditions et les registres du programme pour être restaurés lors du retour. Le système d'interruption du C28x se compose de 16 lignes d'interruptions; la figure 3.4 présente les différentes sources d'interruptions [14].

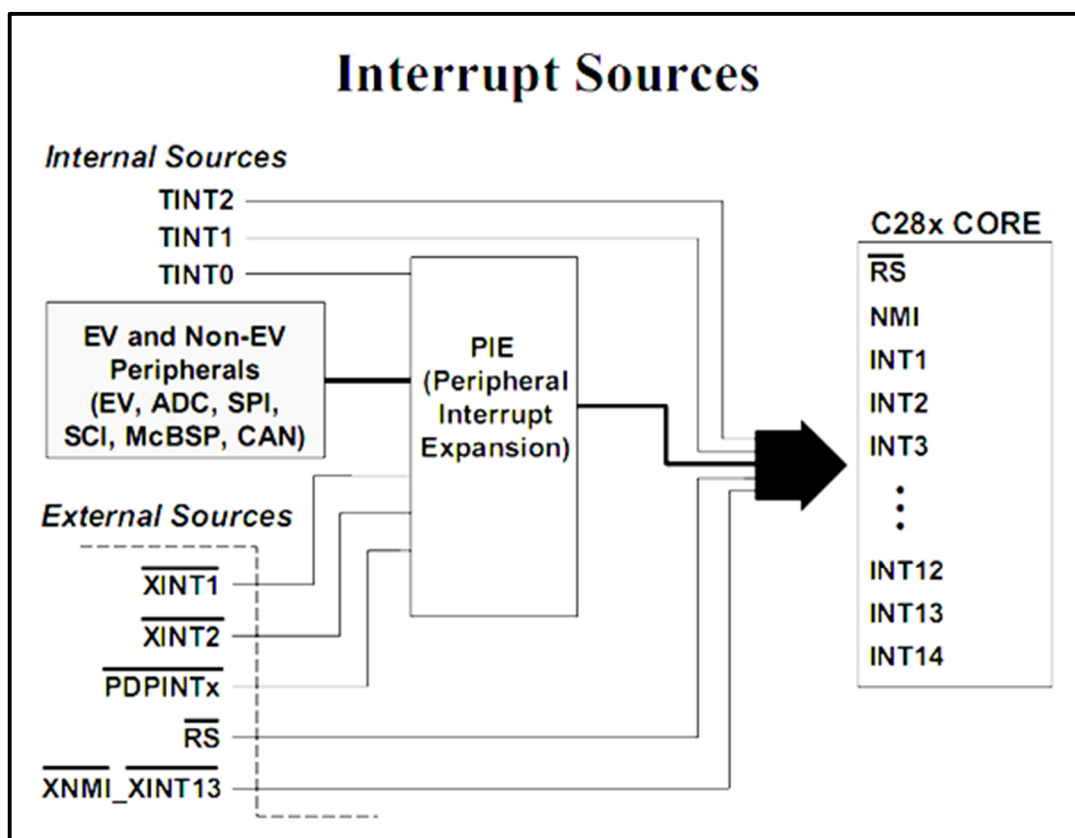


Figure : 3.4 Différentes sources d'interruptions

Le RESET et la NMI sont des interruptions "Non-Masquable". Les 14 autres lignes sont "Masquable" ce qui signifie que le programmeur peut laisser ou rejette des interruptions de ces 14 lignes, en activant ou en désactivant deux interrupteurs (IER et INTM), la figure 3.5 explique le processus de masquage des interruptions.

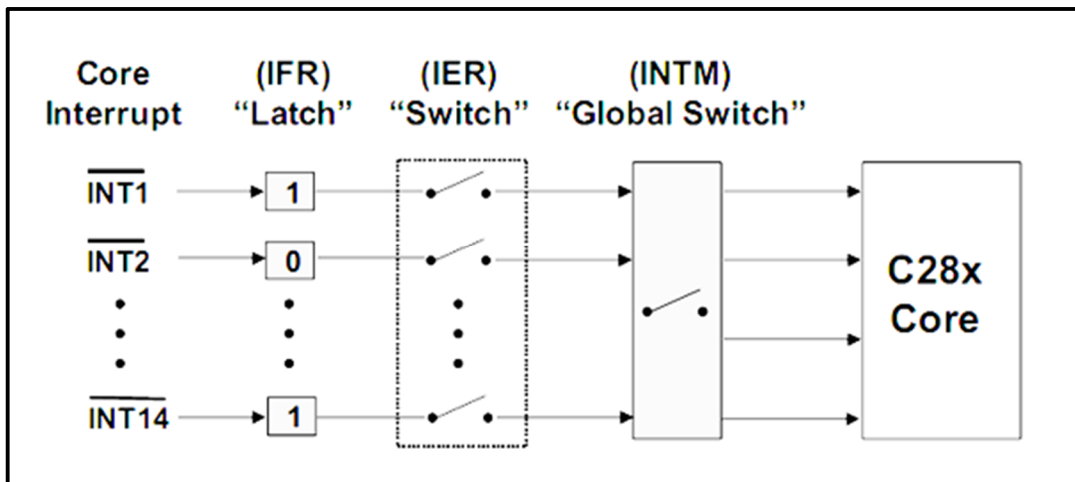


Figure : 3.5 masquage des interruptions

L'unité centrale de traitement n'a pas assez de possibilités pour répondre à toutes les demandes périphériques au niveau d'unité centrale de traitement, un contrôleur centralisé "Peripheral Interrupt Expansion" est utilisé pour arbitrer les demandes d'interrup ons [14].

Le périphérique d'extension des interruptions regroupe 96 sources d'interrup ons masquables possible dans 12 lignes, 8 sources par lignes, pour ac ver une source nous devons activer un groupe de registres propre à cette ligne (PIE Interrupt Flag Register and PIE Interrupt Enable Register), comme le montre la figure 3.6.

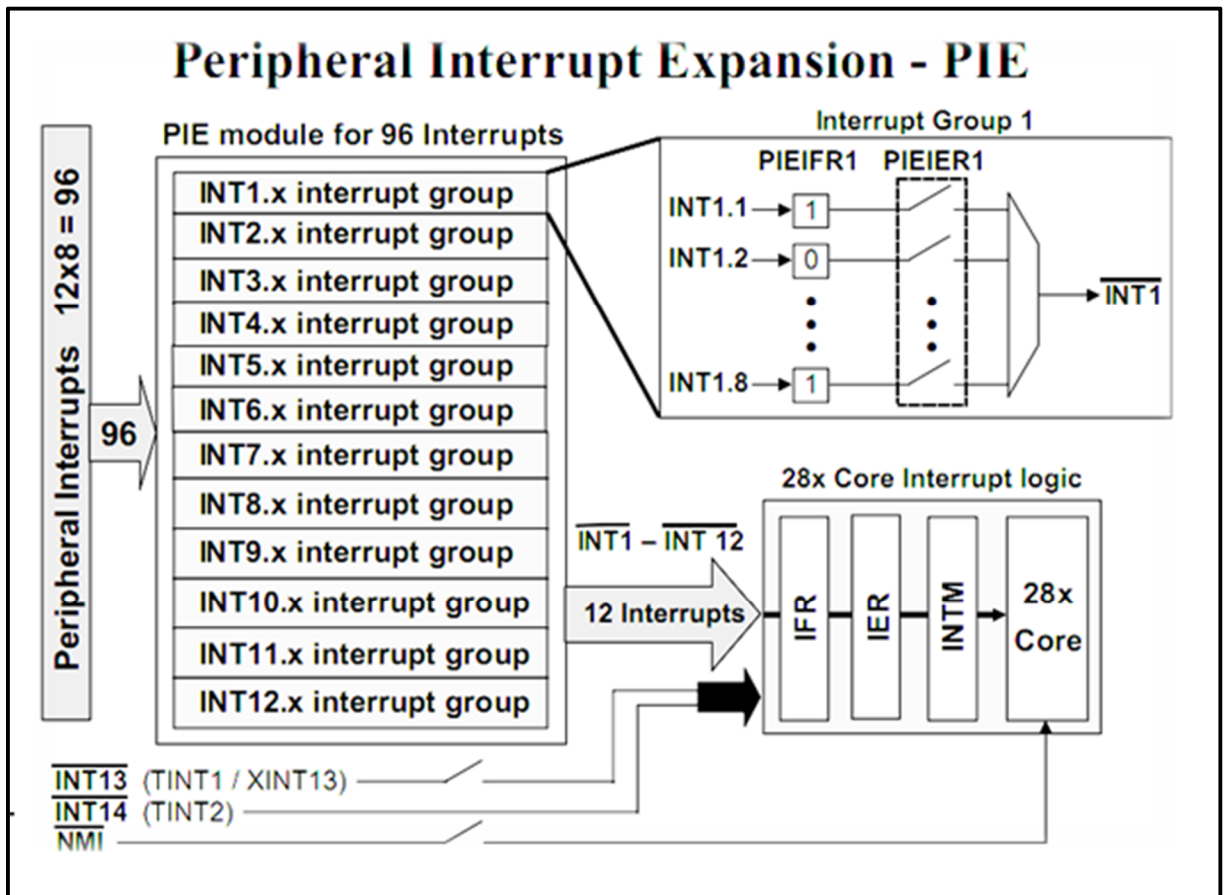


Figure : 3.6 Périphérique d'extension des interruptions

Les interruptions sont exécutées suivant leurs priorités, si deux interruptions déclenchent au même temps, c'est la plus prioritaire qui sera exécutée. Une fois une interruption exécutée c'est au développeur de remettre un accusé de réception à la fin de sa routine pour permettre la prise en considération d'autres interruptions.

Le Reset

Un front descendant appliqué sur la branche externe "/RS" causera une remise à zéro du DSP. Cet événement forcera le DSP à commencer à partir de l'adresse (0x3F FFC0), il effacera également tous les contenus des registres internes, les flags de CPU, et désactivera les 16 lignes d'interruptions. Une autre source de Reset est le débordement du temporisateur de chien de garde "watchdog timer" [14].

3.6.4 Le convertisseur analogique numérique (ADC)

La plupart des signaux physiques telle que la température, le courant électrique et l'accélération sont des signaux analogiques. Ils peuvent être représentés par une tension électrique proportionnelle à leurs valeurs.

L'ADC est une unité périphérique qui convertit la valeur d'une tension (limité entre 0v et 3v) à un nombre binaire de 12 bits [15] [12].

Dans certaines applications on a besoin de plusieurs signaux de mesure, par conséquent, le C28x est équipé de 16 bornes consacrées d'entrée aux tensions analogiques de mesure. Ces 16 signaux sont multiplexés intérieurement, et échantillonnés par deux unités d'échantillonnage et de blocage (S&H) utilisable en même temps (deux signaux échantillonnés simultanément). La conversion est assurée par l'auto séquenceur qui convertit les 16 signaux d'entrée et stocke le résultat dans les registres "Result" d'une manière automatique [15].

Deux modes de fonctionnements sont envisagés pour l'ADC, le "cascade mode" et le "dual sequencer mode". Dans le premier mode un seul séquenceur contrôle le flux de la conversion tandis que dans le deuxième l'ADC est divisé en deux séquenceurs de huit états SEQ1 et SEQ2, chaque séquenceur est activé indépendamment par un générateur d'évènement (EVA pour SEQ1 et EVB pour SEQ2). La figure 3.7 présente le schéma bloc de l'ADC en mode dual séquenceur [15].

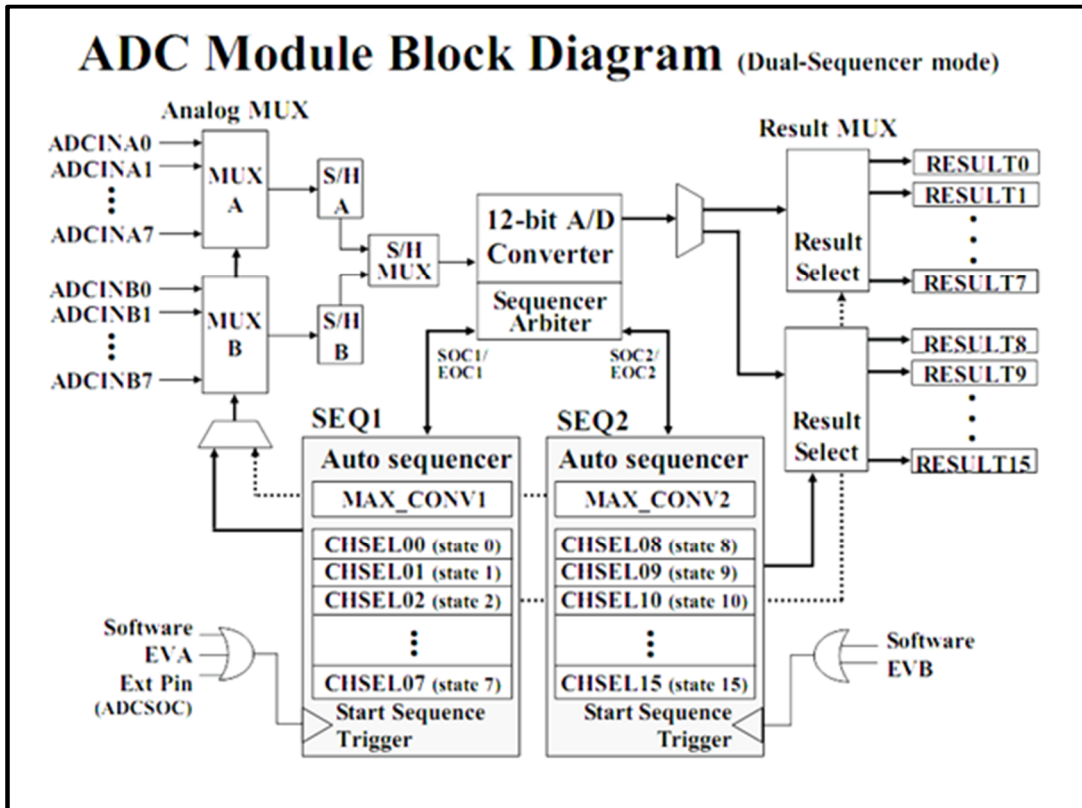


Figure : 3.7 Schéma bloc de l'ADC en mode dual séquenceurs

3.6.5 Le watchdog

C'est un temporisateur relié à l'entrée RESET du noyau DSP, à chaque cycle complet du temporisateur il génère une impulsion qui réinitialise le DSP. C'est à l'utilisateur de réinitialiser le compteur du chien de garde par programme à un certain délai. L'utilisation de chien de garde permet d'éviter les problèmes de blocages qui peuvent se produire dans certaines situations. La figure 3.8 présente le schéma logique de réinitialisation du DSP.

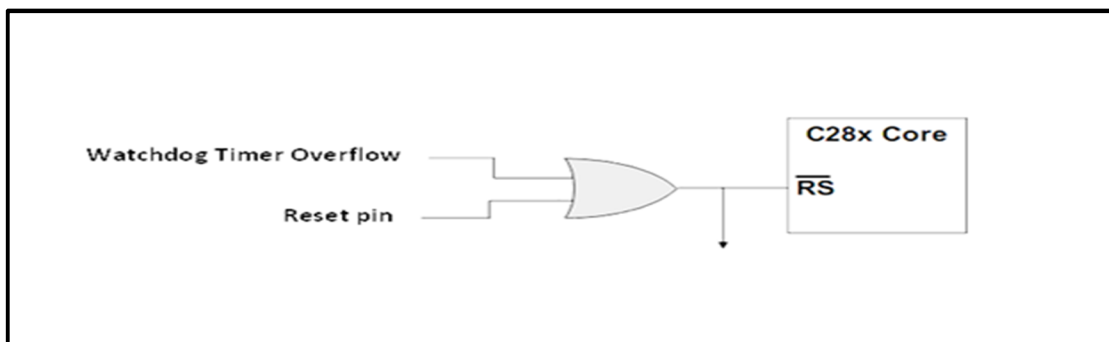


Figure : 3.8 Schéma logique de réinitialisation du DSP

3.6.6 Gestionnaire d'évènement EV (Event Manager)

Les modules du gestionnaire d'évènement fournissent une large gamme de fonctions qui sont particulièrement utiles dans les applications de commandes des moteurs et de commandes des convertisseurs de puissances. Les modules d'Event Manager incluent les temporisateurs à usage général, les unités de comparaison, les unités de capture, et l'encodeur quadrature(QEP). Les gestionnaires d'évènements EVA et EVB sont des périphériques identiques, destinés aux applications de contrôle, ils fournissent chacun 8 signaux PWM simultanément. La figure 3.9 présente le schéma bloc du gestionnaire d'évènements [16].

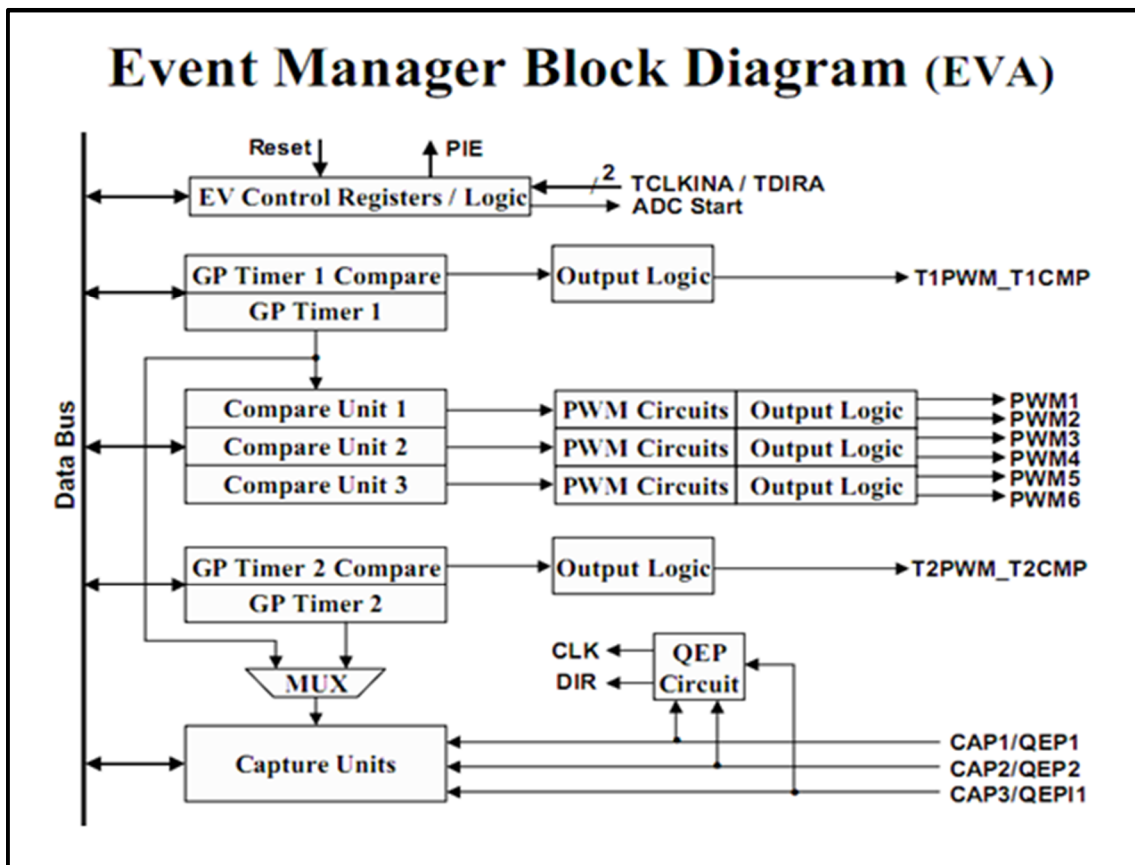


Figure : 3.9 Schéma bloc du gestionnaire d'évènements

a Le temporisateur à usage général (General-Purpose Timers)

Le EVA possède deux temporisateurs à usage général le GPT1 et le GPT2 qui servent comme base de temps pour les différents blocs du gestionnaire d'évènements.

Le bloc de comparaison est la centrale logique d'un GPT. Cette unité compare d'une façon connue la valeur d'un compteur de 16 bits TxCNT avec le contenu de deux registres (registre de comparaison TxCMPR et le registre de période TxPR) afin de générer à la sortie TxPWM un signal PWM. La figure 9.10 présente le schéma bloc d'un générateur à usage général.

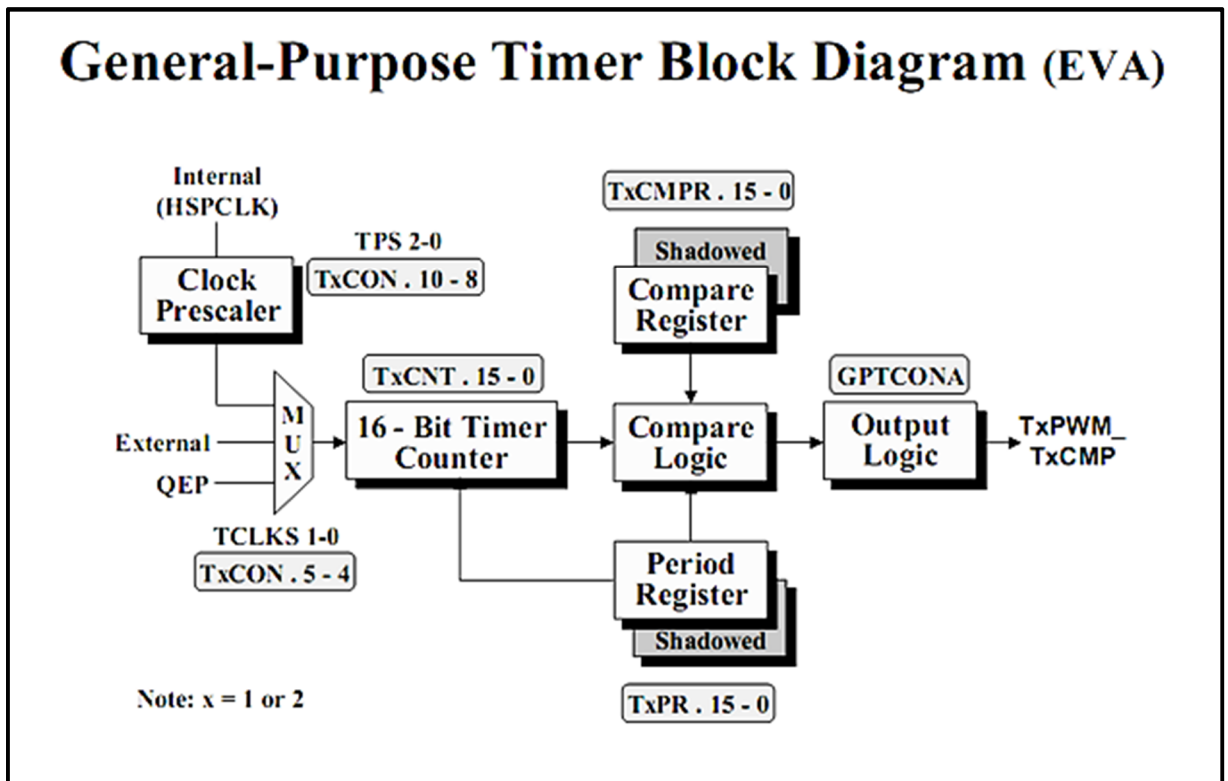


Figure : 3.10 Schéma bloc temporisateur à usage général 1,2

Pour quelques applications, il est nécessaire de changer la valeur des registres TxCMPR et TxPR dans chaque période, cette manœuvre est possible grâce aux registres fantômes (Shadow), qui permette de charger une nouvelle valeur dans les deux registres sans effacer la valeur utilisable dans la période courante [16].

b Unité de comparaison (Full Compare)

Chaque EV possède trois unités de comparaison qui utilisent le GPT pour générer deux signaux PWM complémentaire. Grâce à l'unité de génération de bande morte assignée

à chaque comparateur, une zone morte réglable peut être générée dans les deux signaux complémentaires.

La figure 3.11 présente le schéma bloc de l'unité de comparaison.

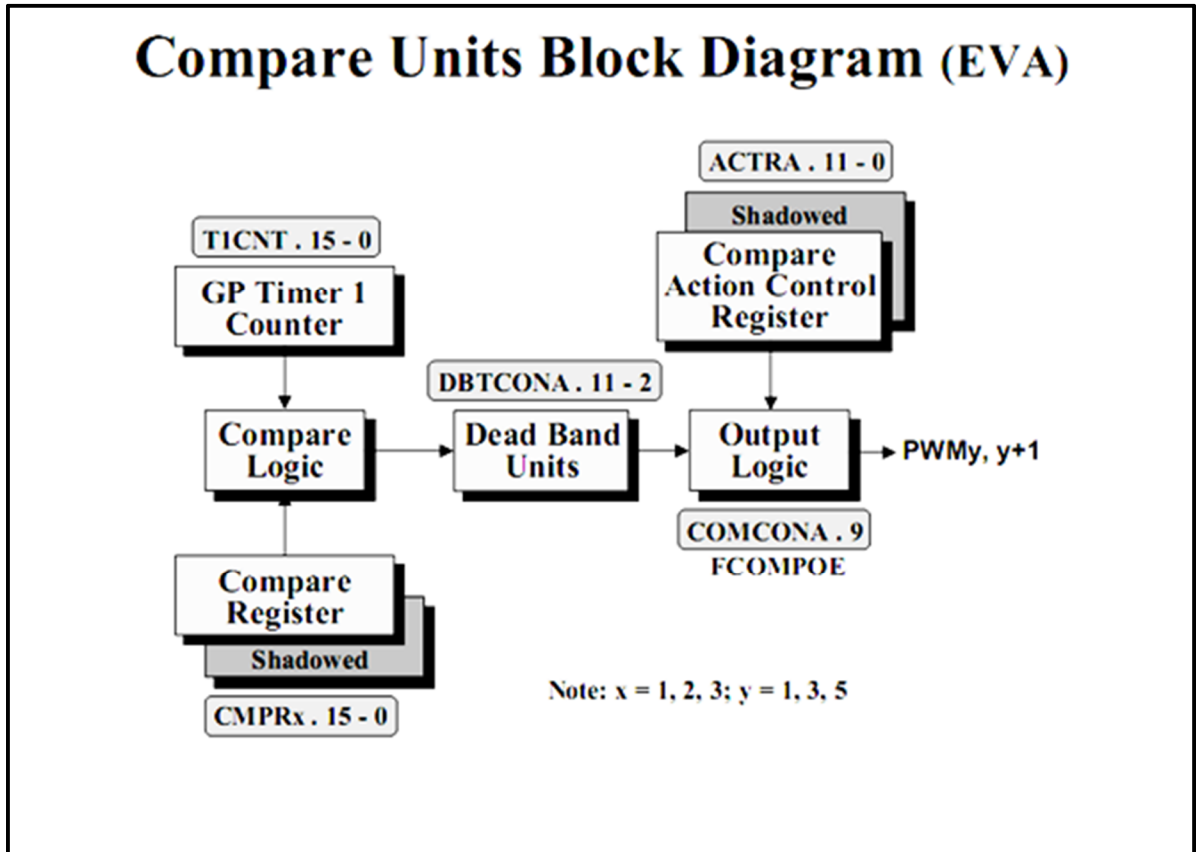


Figure : 3.11 Schéma bloc de l'unité de comparaison

c L'unité de capture

L'unité de capture permet l'acquisition d'un signal logique, chaque gestionnaire d'évènement possède une unité de capture de trois entrées, chaque entrée est associée à une broche de la puce.

La base de temps est sélectionnable par un multiplexeur pour être soit le temporisateur à usage général 1 ou 2 (pour EVA), ou bien le temporisateur à usage général 3 et 4 (pour EVB).

Les utilisations possibles pour l'unité de capture sont :

- Mesure d'une largeur d'impulsion ou un signal numérique.

- Estimation de la vitesse (basse) d'un axe tournant.
- Démarrage automatique d'une conversion analogique numérique (ADC) par le biais de la troisième entrée de l'unité de capture CAP3 [16].

d Encodeur quadrature QEP

L'unité QEP est utilisée pour connaître la position et la vitesse d'un moteur à partir des signaux provenant d'un circuit encodeur monté sur l'axe de rotation du moteur.

Le circuit encodeur se compose d'une roue équipée de trous espacés d'un angle θ , d'une source lumineuse (LED) et de deux photodiodes espacées de $\theta/4$ comme le montre la figure 3.12.

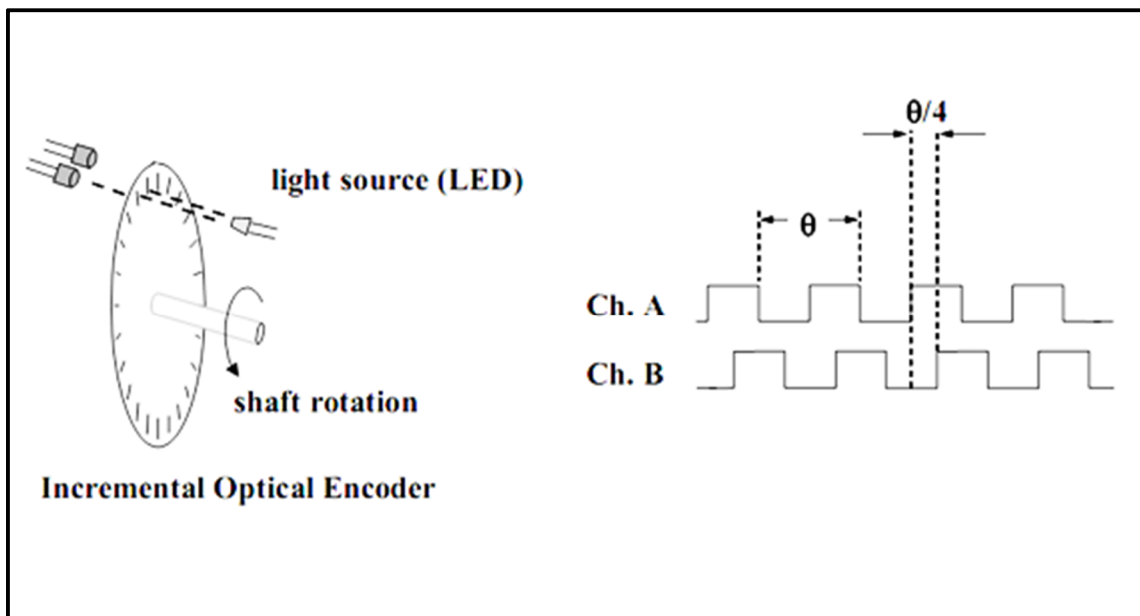


Figure : 3.12 Principe l'encodeur quadratique

La relation de temps entre les sorties du canal A et B conduit à une machine à quatre états. Selon la séquence d'états et la vitesse d'alternance, le GPT 2/4 est décrémenté ou incrémenté. En lisant et en comparant cette information à un intervalle fixe, on obtient donc la vitesse et/ou la position.

Deux entrées de capture CAP1 et CAP2 pour l'EVA et CAP4 et CAP5 pour l'EVB peuvent être utilisées comme interface entre le circuit QEP et l'encodeur en quadrature avec une parfaite synchronisation [16].

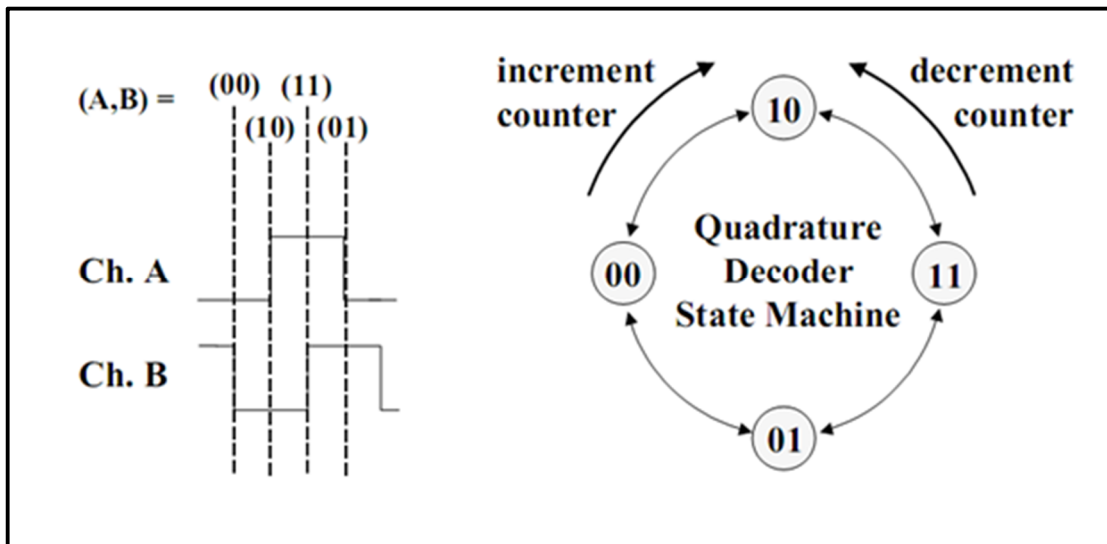


Figure : 3.13 Détermination de position par un signal quadratique

3.7 Code Composer Studio

Code Composer Studio (CCS) est un environnement intégré de développement de code du DSP de Texas Instrument. Il est fourni en standard avec la carte de développement du DSP.

CCS fournit plusieurs outils pour faciliter la construction et la mise au point des programmes de DSP. Il comprend un éditeur de code source, un compilateur de langage C/C++, un assembleur de code source, un éditeur de liens et un environnement d'exécution qui permet de télécharger un programme exécutable sur une carte cible, de l'exécuter et de le déboguer au besoin. CCS comprend aussi des outils qui permettent l'analyse en temps réel d'un programme en cours d'exécution et des résultats produit. Finalement, il fournit un environnement de gestion de fichiers qui facilite la construction et la mise au point des programmes [17].

3.7.1 Outils de génération du code

Le Code Composer Studio offre une interface graphique adéquate pour l'utilisation des outils de génération du code. Il garde en permanence toutes les informations ou les fichiers utilisés pour la création ou la compilation d'un programme ou d'une librairie.

La figure suivante montre le processus du développement software.

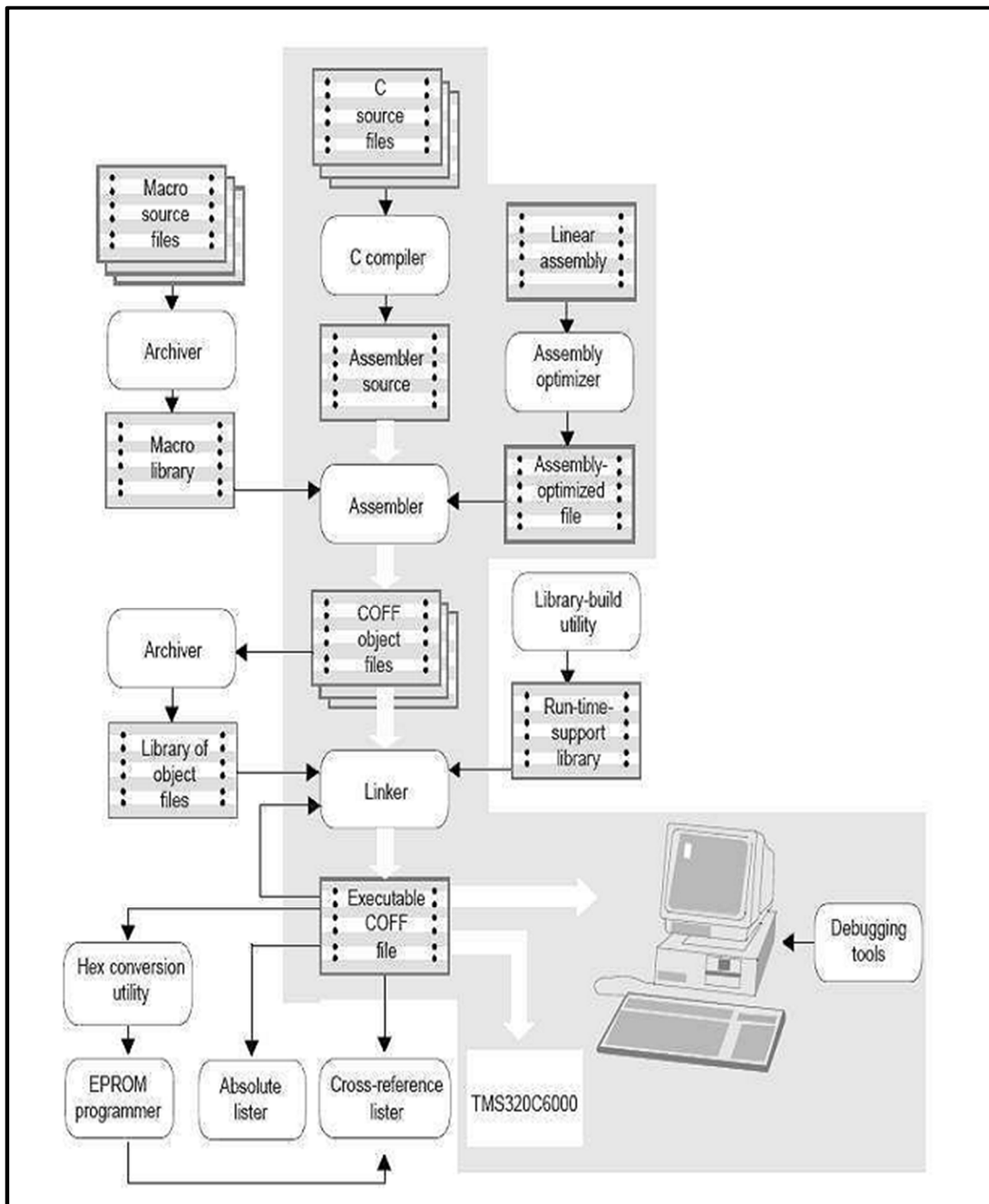


Figure : 3.14 processus de développement de code exécutable

- **Outils de développement pour l'assembleur :**

Les outils de développement de langage d'assembleur sont les suivants :

Assembleur : l'assembleur traduit les fichiers source, écrits en langage assembleur, en fichiers objet de langage machine. Le langage machine est basé sur le format de fichier « common object file format »(COFF).

Archiveur : l'archiveur permet de rassembler un groupe de fichiers dans un seul fichier d'archives simple appelé une bibliothèque. De plus, il permet de modifier une bibliothèque en supprimant, remplaçant, extrayant, ou ajoutant des membres (routines). Une des applications les plus utilisées de l'archiver est de construire une bibliothèque des modules d'objet.

Éditeur de liens (Linker) : l'éditeur de liens combine les fichiers objet dans un fichier objet exécutable par la cible. Pendant la création du fichier exécutable, il exécute la mise en mémoire adressable disponible, selon la cible, des fonctions définies par le fichier de commande et associe les références externes.

Absolute Lister : il accepte les fichiers exécutables comme entrée et crée des fichiers .abs en sortie. En assemblant ce fichier on obtient une liste du code contenant les adresses absolues des instructions du code exécutable.

Cross-référence Lister : le générateur de liste de correspondance emploie les fichiers exécutables pour produire une liste de correspondance montrant des symboles, leurs définitions et leurs références dans les fichiers source liés. C'est donc un rapport de l'opération exécutée par le linker.

Utilitaire de conversion hexadécimal : son utilité réside dans la conversion de fichier exécutable de format COFF en format adapté aux EPROM Ti-Étiqueté, d'ASCII-hex, d'Intel, de Motorola-S, ou de Tektronix. L'EPROM en question fait l'objet d'une mémoire de stockage du code exécutable via laquelle le DSP est amorcé.

- **Outils de développement pour C/C++ :**

Les outils de développement pour C/C++ sont :

Compilateur C/C++ : le compilateur de C/C++ accepte le code source C/C++ et produit le code source en langage assembleur. Donc il s'agit de compiler et faire l'assemblage des fichiers ; si une optimisation est invoquée l'optimisateur de code C intervient.

Optimisateur du code Assembleur : il permet de prendre en considération le code écrit en assembleur linéaire .sa (Linear assembler) c'est-à-dire qu'il est écrit sans tenir compte du parallélisme ou une optimisation d'utilisation des registre. Son rôle est d'optimiser cette structure en employant les registres appropriés, ainsi qu'un autre apport d'optimisation pour les boucles, et cela pour obtenir un code fortement parallélisé.

Utilitaire de construction de bibliothèques : utilisé pour réaliser un archivage de routines fréquemment utilisées sous forme de bibliothèques, cet utilitaire génère un fichier d'extension .lib.

Des bibliothèques sont disponibles sont fournies par le concepteur du logiciel CCS (TI), telle que la bibliothèque rts2800.lib utilisée par le linker afin de construire le fichier exécutable adéquat à une cible de la famille C28x.

3.8 Implémentation du FK sur DSP

3.8.1 La programmation en C

La recherche de célérité des algorithmes de commande et d'estimation contraint le plus souvent à utiliser le langage assembleur propre au DSP afin d'optimiser les performances du processeur. Malgré cela, nous avons choisi le langage C, plus évolué, générique, lisible, compréhensible et portable en dépit de la baisse de performances liée à la programmation de plus haut niveau.

Afin de réduire le temps d'exécution du programme, on a respecté certains points de programmation :

- Eviter de laisser dans le code des calculs mathématiques invariants dans le temps :

$$\text{ex : } y = 1.23 * x / 2.5 \quad \Leftrightarrow \quad 0.492 * x$$

- Aucune constante numérique ne doit apparaître dans le code à part la valeur zéro utilisée comme valeur initiale de comptage. Toutes les constantes sont donc définies par des symboles avec l'instruction « #define » [19].

- Eviter d'utiliser la fonction puissance, et la remplacer par une série de multiplication.

3.8.2 La programmation en utilisant La bibliothèque IQ Math

Dans les plateformes de simulation, telle que le Matlab, les algorithmes sont écrits en virgule flottante, ces algorithmes peuvent être facilement utilisés par un DSP à virgule flottante (possédant une unité centrale qui soutient les opérations à point flottant). Ces algorithmes peuvent être implémentés sur un DSP à virgule fixe, mais pour que ce dernier effectue les opérations mathématiques telle que l'addition, la multiplication et la division, il doit exécuter des sous-programmes qui prennent énormément de temps si les calculs sont intenses. Pour le cas du DSP F2812 la multiplication est réalisée par un sous-programme de 90 mots, tandis que l'addition est réalisée par un sous-programme de 120 mots.

L'avantage d'utiliser des variables de types float et long (64 bits) est la précision avec laquelle les opérations sont menées, cette précision se paye au prix de temps d'exécution.

Les DSP à virgule fixe possèdent une unité de traitement à virgule fixe, qui peut exécuter différents types d'opérations mathématiques d'une manière très efficace, en utilisant seulement quelques cycles d'horloge. La bibliothèque IQ Math fournit un ensemble de fonctions permettant d'exploiter d'une manière optimale les ressources internes de DSP F2812.

La librairie IQ Math du DSP F2812 soutient les formats de Q1 à Q30, pour des nombres représentables qui varient entre 1.999999999 et -2 avec une précision de 10^{-9} dans le cas de Q30, et des nombres qui varient entre 1073741823.5 et -1073741824 avec une précision de 0.5 pour Q1. Cette bibliothèque nous donne la possibilité de choisir un format pour chaque variable d'une façon indépendante, dans notre choix de format on doit prendre en considération la valeur maximale que prend la variable, elle nous donne aussi la possibilité de passer d'un format à l'autre pour effectuer une addition ou une soustraction [18].

a Quelques fonctions de la bibliothèque IQ Math :

- **Les fonctions de transformations :**

IQ Math mais à la disposition du programmeur un ensemble de fonctions de transformation, de type de donnée à un autre, et de format de donnée à un autre format.

- La fonction `_IQN` : cette C-Macro convertit les constantes et les variables à virgule flottante à un format IQ.
- La fonction `_IQtoF` : cette fonction convertit une valeur IQ à sa équivalente de type float (représenté dans la norme IEEE 754).

- **Les fonctions de multiplications :**

Plusieurs fonctions de multiplication sont disponibles, telle que :

- La fonction `IQNmpy` : la multiplication de deux nombre de même format, et l'arrangement du résultat dans le même format, cette opération prend la moindre quantité de cycles d'horloge et de nombre d'instructions, elle devrait être employée le plus souvent.
- La fonction `IQNmpyIQX` : la multiplication de deux nombres possédants deux formats différents et l'arrangement de résultat dans un troisième format.

Remarque : La multiplication de deux nombres de 32 bits représentés en format Q_k génère un résultat intermédiaire de 64 bits, le résultat retenu sera sur 32 bits présenté en format $Q_{k'}$, en gardant k' bits du poids fort de la partie fractionnaire, et $(32 - k')$ bits du poids faible de la partie entière [18].

Détermination des éléments de matrices de FK :

Le passage de la forme matricielle à la forme élémentaire est une étape primordiale pour l'implémentation du FK sur un calculateur numérique. Une étape de simplification des équations réduit sensiblement le temps de calcul.

Nous déterminons les équations des éléments des matrices suivantes : matrice transition d'état discrète A_d , matrice d'entrée B_d , matrice de bruit d'état w_d , matrice

de covariance de l'erreur de prédiction $P[k + 1/k]$, matrice de covariance de l'erreur d'estimation $P[k/k]$ et matrice de gain $K_f[k + 1]$.

La matrice A_d est définie par l'équation (3.1)

$$A_d = \begin{bmatrix} A_{d11} & A_{d12} & A_{d13} & A_{d14} \\ -A_{d12} & A_{d11} & -A_{d14} & A_{d13} \\ A_{d31} & A_{d32} & A_{d33} & A_{d34} \\ -A_{d32} & A_{d31} & -A_{d34} & A_{d33} \end{bmatrix} \quad (3.1)$$

Après simplification nous obtenons les expressions de chaque éléments de A_d :

$$\begin{cases} A_{d11} = 1.9581 * 10^{-7}x_1 + 0.981 \\ A_{d12} = 5.8741 * 10^{-5}x - 4.9618 * 10^{-10}x_2 \\ A_{d13} = 1.1574 * 10^{-10}x_2 + 1.373 * 10^{-3}x_1 + 0.032 \\ A_{d14} = 4.5744 * 10^{-6}x_2 + 0.27 \\ A_{d31} = 4.2437 * 10^{-4} - 7.062 * 10^{-9}x_1 \\ A_{d32} = 1.7862 * 10^{-11}x_2 - 2.1198 * 10^{-6}x \\ A_{d33} = -4.1667 * 10^{-12}x_2 - 4.9548 * 10^{-5}x_1 + 0.9988 \\ A_{d34} = 1.6498 * 10^{-7}x_2 - 9.9293 * 10^{-3} * x \end{cases} \quad (3.2)$$

Avec :

$x = \Omega$: la vitesse mécanique.

$$x_1 = \Omega * \Omega$$

$$x_2 = x_1 * \Omega$$

La matrice B_d est définie par l'équation (3.3)

Avec :

$$B_d = \begin{bmatrix} B_{d11} & B_{d12} \\ -B_{d12} & B_{d11} \\ B_{d31} & B_{d32} \\ -B_{d32} & B_{d31} \end{bmatrix} \quad (3.3)$$

Alors après calcul :

$$\begin{cases} B_{d11} = 0.28 + 1.3783 * 10^{-8}x_1 \\ B_{d12} = 5.4571 * 10^{-6} * x \\ B_{d31} = -4.9618 * 10^{-10}x_1 + 5.9140 * 10^{-5} \\ B_{d32} = -1.9682 * 10^{-7}x \end{cases} \quad (3.4)$$

La matrice w_d caractérise les bruits d'état du processus, elle est définie par l'équation (3.5) et :

$$W_d = \begin{bmatrix} W_{d11} & W_{d12} & W_{d13} & W_{d14} \\ -W_{d12} & W_{d11} & -W_{d14} & W_{d13} \\ W_{d31} & W_{d32} & W_{d33} & W_{d34} \\ -W_{d32} & W_{d31} & -W_{d34} & W_{d33} \end{bmatrix} \quad (3.5)$$

Alors:

$$\begin{cases} W_{d11} = -3.8576 * 10^{-4} x_1 - 9.5058 * 10^{-6} \\ W_{d12} = -9.9236 * 10^{-11} x \\ W_{d13} = 1.3888 * 10^{-5} x_1 + 4.1102 * 10^{-7} \\ W_{d14} = 1.3890 * 10^{-7} x \\ W_{d31} = W_{d13} \\ W_{d32} = -1.379 * 10^{-7} x \\ W_{d33} = 9.8746 * 10^{-7} - 4.9997 * 10^{-7} x_1 \\ W_{d34} = -2.3730 * 10^{-11} x \end{cases} \quad (3.6)$$

La matrice V_d représente les bruits issus de la mesure des courants, elle est obtenue par la division de la matrice V par la constante d'échantillonnage t_e . Ces éléments sont constants, donc leurs valeurs seront injectées directement dans le calcul du gain K_f .

La matrice de variance-covariance de l'erreur de prédiction donnée par (3.7) est définie par :

$$P[k + 1/k] = \begin{bmatrix} P_{11}[k + 1/k] & P_{12}[k + 1/k] & P_{13}[k + 1/k] & P_{14}[k + 1/k] \\ -P_{12}[k + 1/k] & P_{11}[k + 1/k] & -P_{14}[k + 1/k] & P_{13}[k + 1/k] \\ P_{31}[k + 1/k] & P_{32}[k + 1/k] & P_{33}[k + 1/k] & P_{34}[k + 1/k] \\ -P_{32}[k + 1/k] & P_{31}[k + 1/k] & -P_{34}[k + 1/k] & P_{33}[k + 1/k] \end{bmatrix} \quad (3.7)$$

Les symétries et les antisymétries présentes dans La matrice de variance-covariance de l'erreur de prédiction sont conservées dans la matrice de variance-covariance de l'erreur d'estimation, cette dernière est donnée par:

$$P[k/k] = \begin{bmatrix} P_{11}[k/k] & P_{12}[k/k] & P_{13}[k/k] & P_{14}[k/k] \\ -P_{12}[k/k] & P_{11}[k/k] & -P_{14}[k/k] & P_{13}[k/k] \\ P_{31}[k/k] & P_{32}[k/k] & P_{33}[k/k] & P_{34}[k/k] \\ -P_{32}[k/k] & P_{31}[k/k] & -P_{34}[k/k] & P_{33}[k/k] \end{bmatrix} \quad (3.8)$$

Le gain de correction $K_f[k + 1]$:

La matrice de gain $K_f[k + 1]$ est une matrice de dimension 4x2 définie par (3.9) est donnée par :

$$K_f[k + 1] = \begin{bmatrix} K_{11}[k + 1] & K_{12}[k + 1] \\ -K_{12}[k + 1] & K_{11}[k + 1] \\ K_{31}[k + 1] & K_{32}[k + 1] \\ -K_{32}[k + 1] & K_{31}[k + 1] \end{bmatrix} \quad (3.9)$$

La complexité des calculs des matrices du FK :

La complexité des calculs donnés par le tableau (3.1) est évaluée en comptant le nombre de multiplications et d'additions du modèle discrétisé.

Matrice	nombre de multiplications et de division	nombre d'additions et de soustraction
A_d	16	10
B_d	4	2
W_d	7	3
V_d	0	0
$\hat{x}[k + 1/k]$	24	20
$P[k + 1/k]$	64	56
$K[k + 1]$	13	5
$P[k + 1/k + 1]$	16	13
$\hat{x}[k + 1/k + 1]$	8	8

Tableau : 3.3 nombre d'opérations pour le calcul du filtre de Kalman.

Le coût algorithmique du filtre de Kalman :

Le coût algorithmique du filtre de Kalman est évalué en comptant le nombre de multiplications et d'additions du modèle discrétisé. 152 opérations d'addition sera effectuées en une seule itération, 42% d'entre elle pour calculer la matrice de covariance de l'erreur de prédiction.

Pour la multiplication 117 opérations sont envisagées, 47.86% d'entre elle pour calculer la matrice de covariance de l'erreur de prédiction.

Le schéma de la figure suivante donne les grandeurs des variables d'état

estimé $\begin{bmatrix} \hat{i}_{ds} \\ \hat{i}_{qs} \\ \hat{\Phi}_{dr} \\ \hat{\Phi}_{qr} \end{bmatrix}$, résultats des différentes opérations arithmétiques, effectuées pour

l'algorithme du FK.

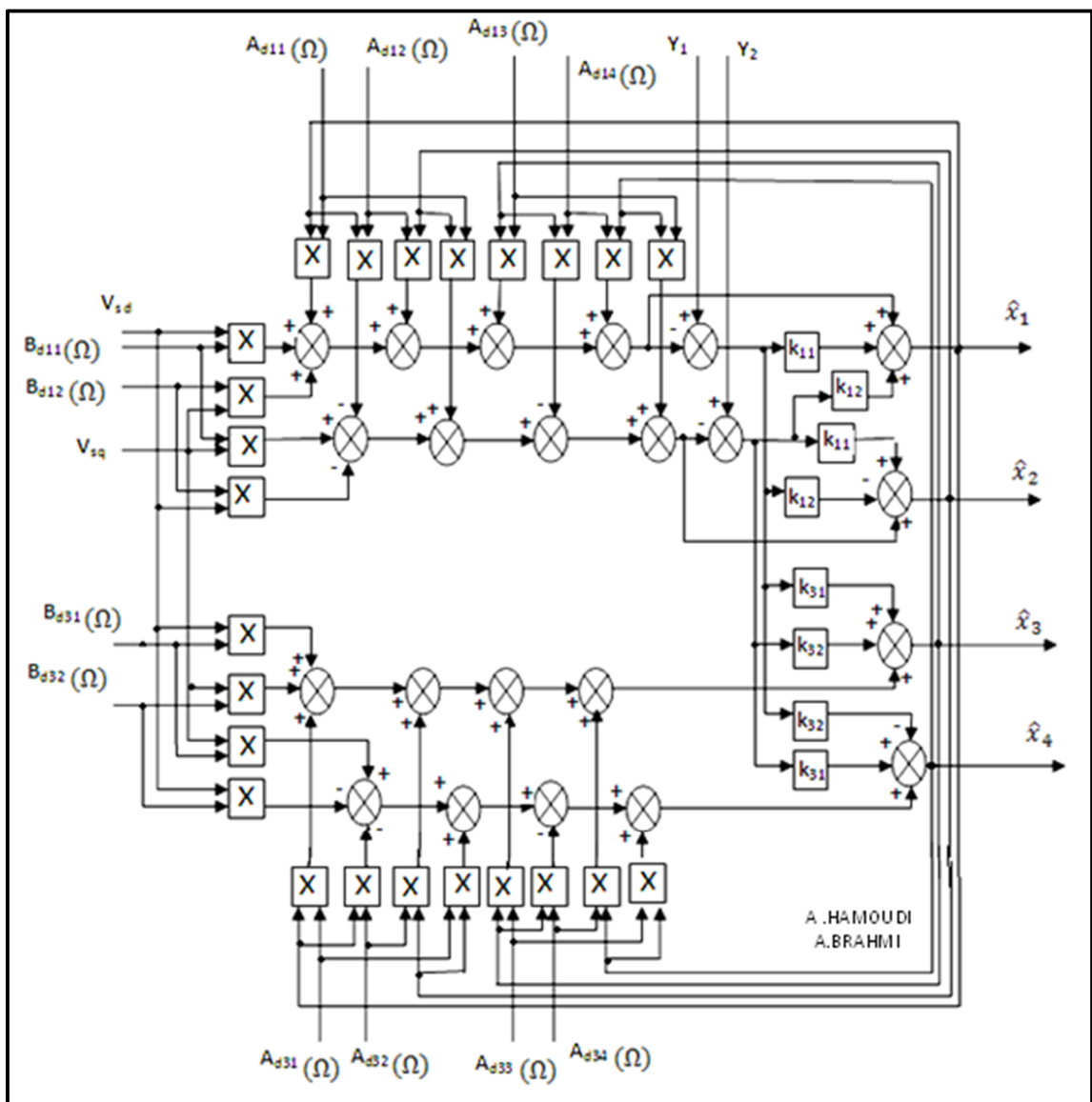


Figure: 3.15 modèle éclaté du filtre de Kalman

3.9 Conclusion

On a présenté dans ce chapitre les caractéristiques générales des DSP, et les formats utilisés pour les représentations binaires des réels.

Les périphériques de commande d'un DSP déterminent le cadre général d'utilisation de ce dernier, en effet pour la commande des machines électriques le TMS320F2812 est la solution idéale. L'ADC et le gestionnaire d'évènement nous permet d'acquérir la vitesse, les courants et les tensions électriques mesurées.

Dans la deuxième partie de ce chapitre en a parlé des outils de développements logiciels offerts par le CCS, et l'intérêt de la programmation en langage C et en IQ Math.

L'algorithme du filtre de Kalman développé à la fin de ce chapitre sera programmé en utilisant les deux outils précédents.

Chapitre 4 Résultats et commentaires

4.1 Introduction

L'implémentation du filtre de Kalman sur DSP sera faite sur plusieurs étapes. En commence par effectuer le programme du FK sur Matlab pour simuler l'ensemble MAS et filtre. La deuxième étape consiste à implémenter l'algorithme du FK écrit en langage C sur le DSP F2812 et tester ses performances. Dans ce cas les tensions, les courants statoriques, et la vitesse sont issus de la simulation de la MAS sur matlab. Un troisième programme du filtre est ensuite implémenté sur le DSP F2812 utilisant des ressources importantes (la bibliothèque IQ Math).

La dernière étape consiste à faire appliquer les deux algorithmes du FK sur une machine réel, en faisant l'acquisition des courants statoriques, les tensions d'alimentation et la vitesse mécanique.

Une carte d'interface est réalisée pour visualiser les flux et les courants estimés sur un oscilloscope.

4.2 Résultats de l'application du FK (simulation Matlab) à la MAS

Le model de la MAS est simulé sur logiciel Matlab (programme m.file), alimenté de trois tensions sinusoïdales de valeur efficace 220 volts et de fréquence 50 hertz.

Les bruits d'état et de mesure sont habituellement considérés comme des signaux gaussiens centrés; leurs variances sont généralement estimées. Ces grandeurs sont importantes pour une bonne convergence du FK. Dans notre travail la valeur des composantes de la matrice bruit d'état et du vecteur de mesure sont déterminées par la méthode essai-erreur.

La matrice de bruit d'état est :

$$w = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

Le vecteur de bruit de mesure est :

$$v = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

Dans les essais en temps réel ces matrices seront ré estimées pour le modèle de la MAS utilisé.

Un couple de charge de 10 N.m est appliqué à la machine en simulation à $t=0.3$ s.

Les figures ci-dessous montrent l'évolution, du couple électromagnétique et de la vitesse mécanique.

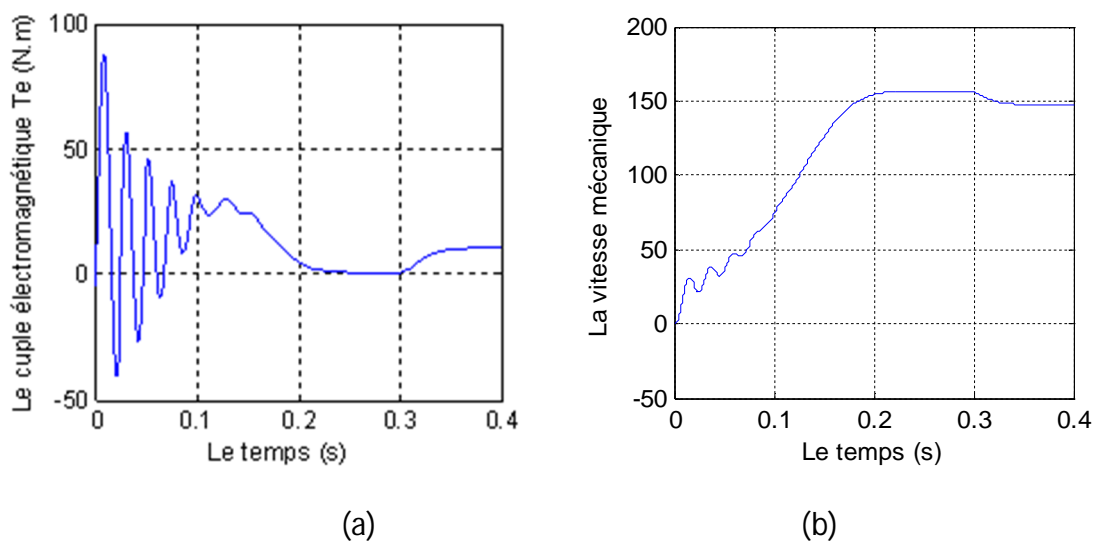


Figure 4.1 La vitesse mécanique Ω et le couple électromagnétique

Les figures ci-dessous illustrent l'évolution des variables d'état mesurées (courants statoriques) et non mesurées (flux rotoriques) et les erreurs d'estimations correspondantes.

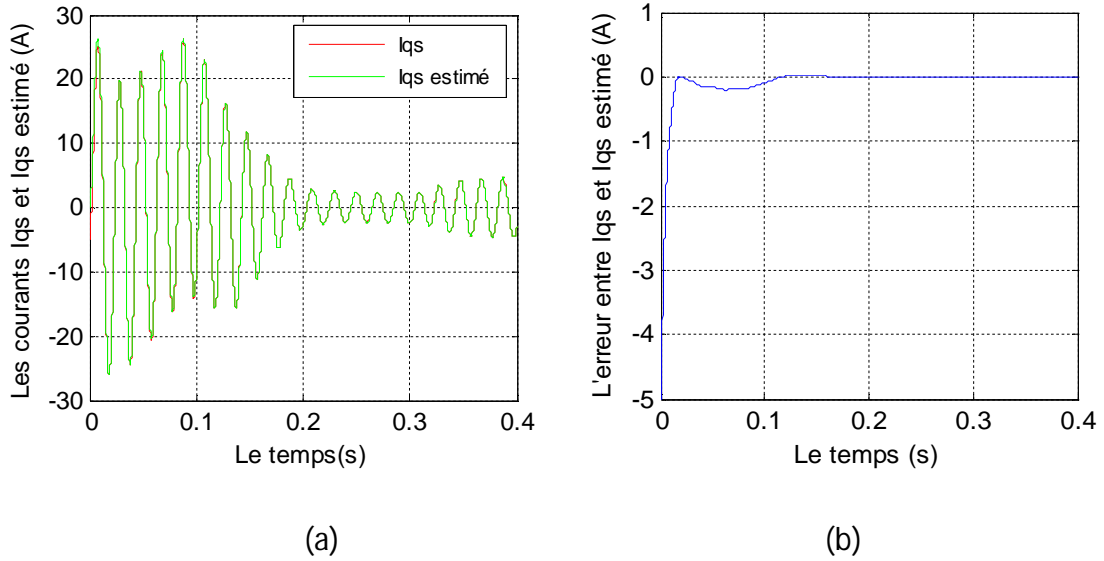


Figure 4.2 la superposition des courants I_{qs} et \hat{I}_{qs} et l'erreur entre eux

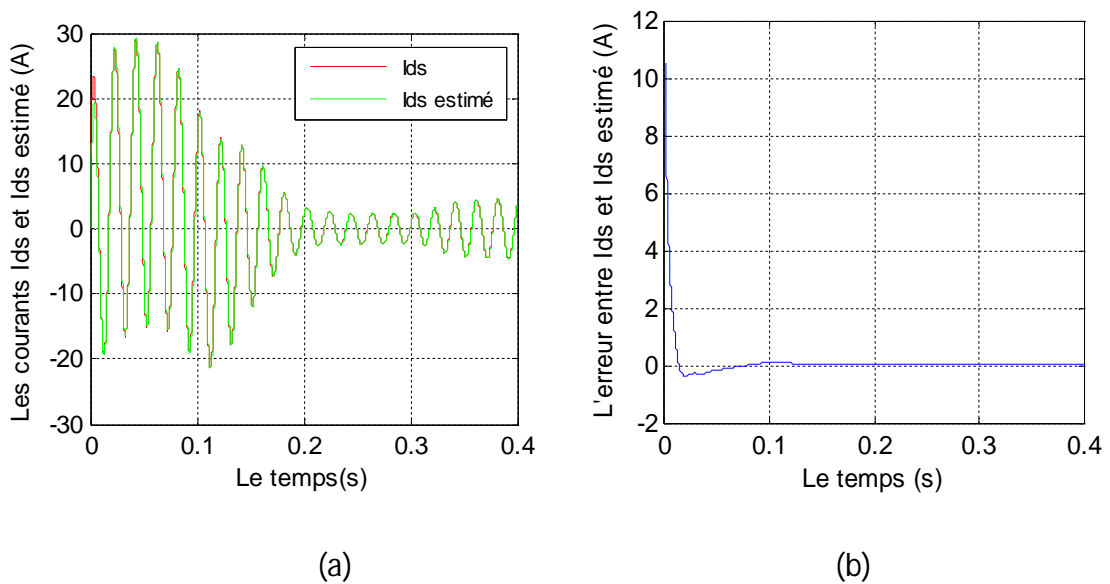
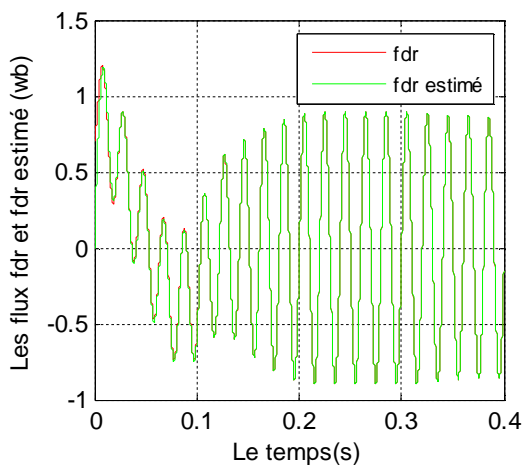
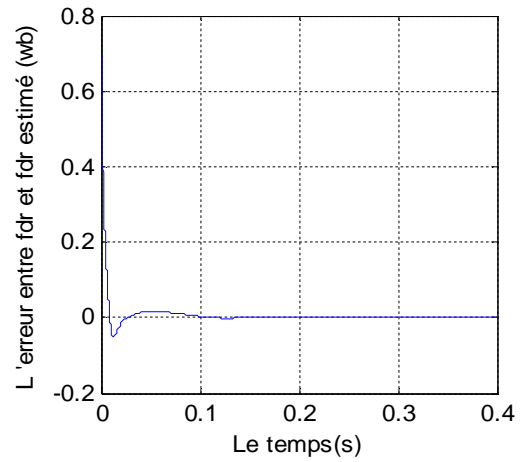


Figure 4.3 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux

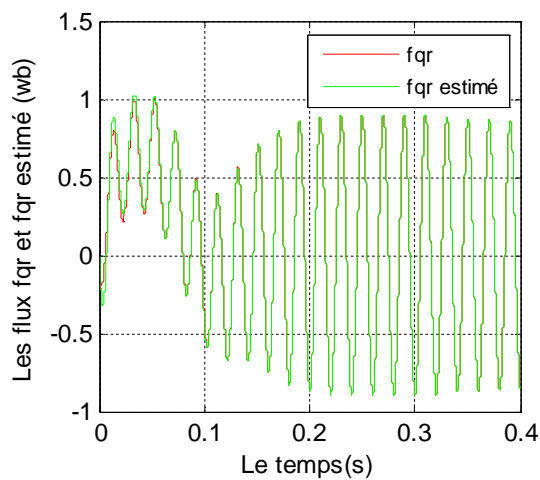


(a)

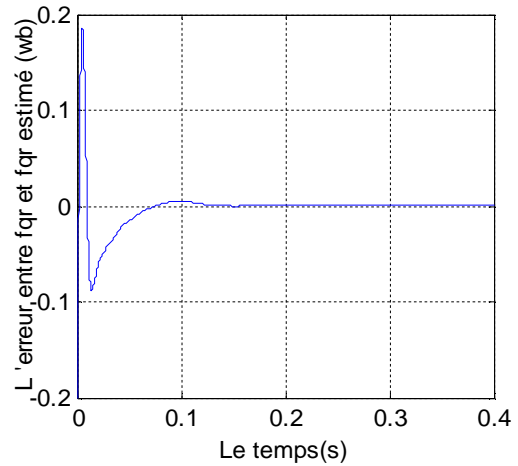


(b)

Figure 4.4 la superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux



(a)



(b)

Figure 4.5 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux

Commentaires

Les figures (4.3 a), (4.2 a) représentent la superposition entre les courants (I_{ds} , I_{qs}) et leurs estimés (\hat{I}_{ds} , \hat{I}_{qs}). Ces derniers suivent parfaitement les courants du modèle. les

figures (4.5 a) et (4.4 a) décrivent la superposition des flux rotoriques (ϕ_{dr}, ϕ_{qr}) et de leurs estimés ($\hat{\phi}_{dr}, \hat{\phi}_{qr}$).

Les figures (4.2 b) et (4.3 b) représentent l'erreur d'estimation entre les courants et leurs estimés. Pendant le régime transitoire l'erreur d'estimation est non nulle cela est due à la différence entre les conditions initiales prises sur les variables d'état du filtre et celles qui sont prises sur les variables d'état de la MAS. En régime permanent, cette erreur tend vers le zéro pour les courants (I_{ds}, I_{qs}). Les figures (4.4 b) et (4.5 b) représentent l'erreur d'estimation entre les flux rotoriques et leurs estimés ; cette erreur tend vers le zéro. Ceci montre clairement la convergence du filtre de Kalman.

Les résultats obtenus sont dus à l'implémentation de l'algorithme du filtre de Kalman sans simplification de calcul, et à la nature du calculateur de l'ordinateur, qui est un micro-processeur à virgule flottante réalisant des calculs avec une grande précision .

4.3 Résultats de l'application de l'algorithme du FK écrit en langage C sur le DSP F2812 :

Les figures ci-dessous illustrent l'évolution des variables d'état mesurées (courants statoriques) et non mesurées (flux rotoriques) et leurs estimés.

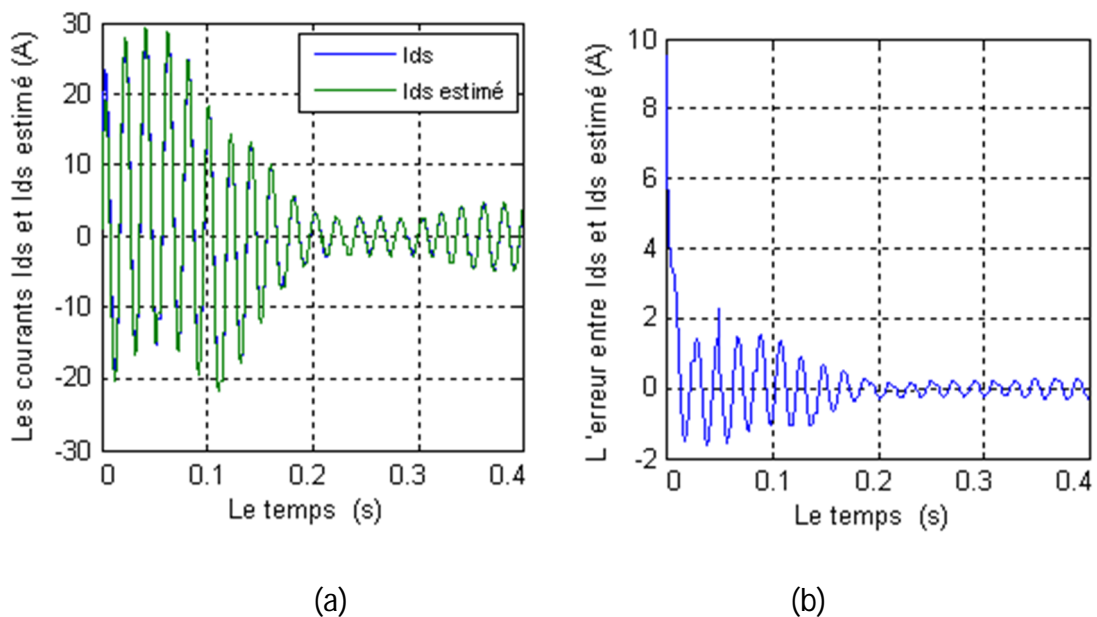
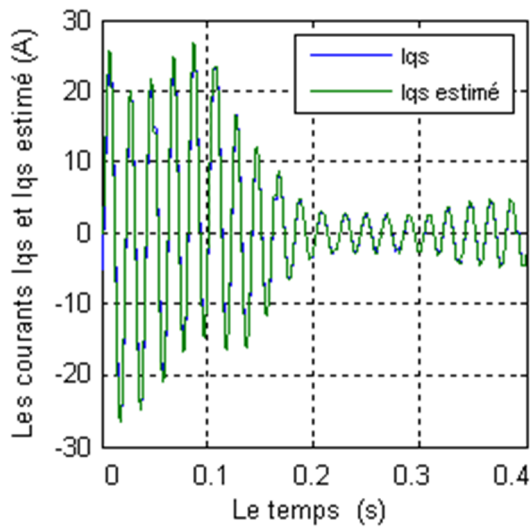
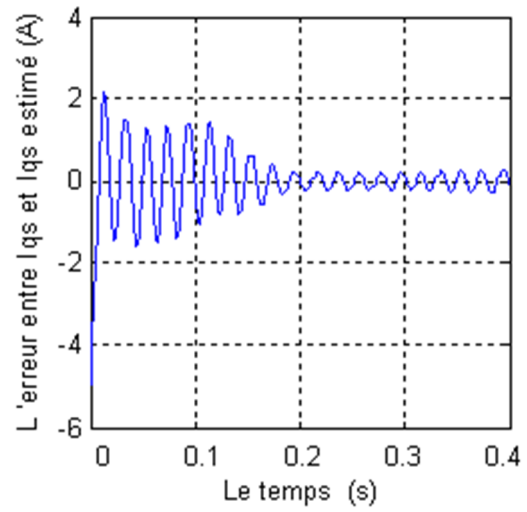


Figure 4.6 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux

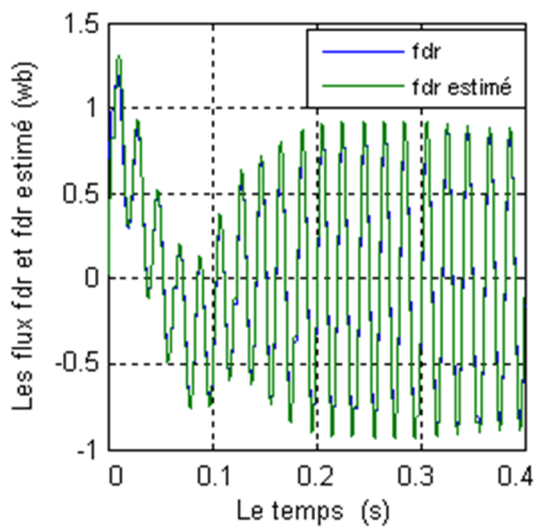


(a)

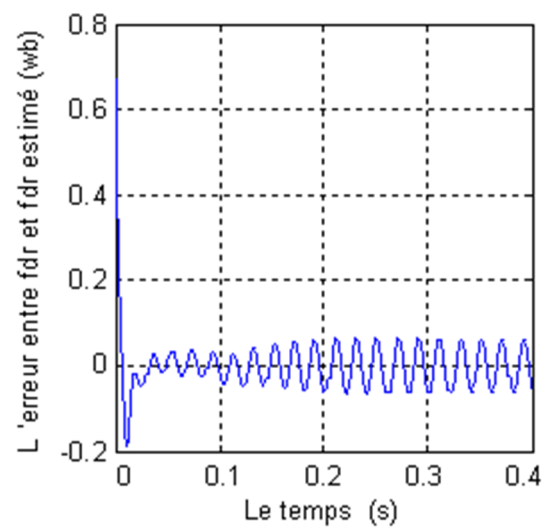


(b)

Figure 4.7 la superposition des courants i_{qs} et \hat{i}_{qs} et l'erreur entre eux



(a)



(b)

Figure 4.8 la superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux

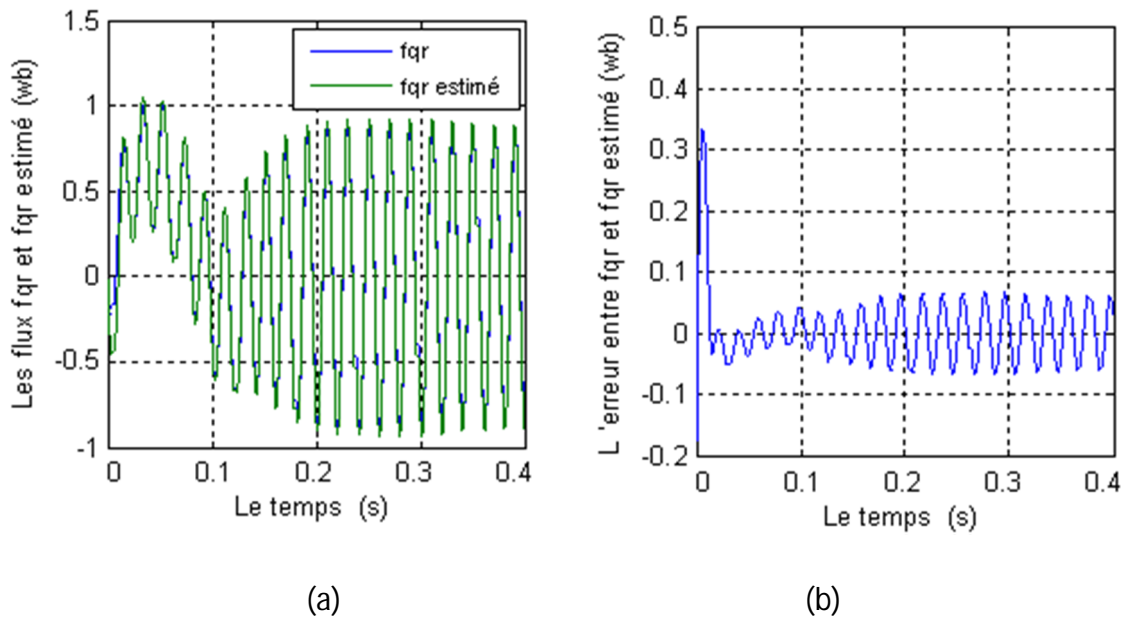


Figure 4.9 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux

Commentaires

Les figures (4.6 a), (4.7 a) représentent la superposition entre les courants (I_{ds}, I_{qs}) et leurs estimés ($\hat{I}_{ds}, \hat{I}_{qs}$). Ces derniers suivent parfaitement les courants du modèle. Les figures (4.8 a) et (4.9 a) décrivent la superposition des flux rotoriques (ϕ_{dr}, ϕ_{qr}) et de leurs estimés ($\hat{\phi}_{dr}, \hat{\phi}_{qr}$).

Les figures (4.6 b) et (4.7 b) représentent l'erreur d'estimation entre les courants et leurs estimés. Au démarrage elle atteint la valeur de '5 A', cela est dû à la différence entre les conditions initiales prises sur les variables d'état du filtre et celles qui sont prises sur les variables d'état de la MAS. Pendant le régime transitoire cette erreur prend de grandes valeurs qui arrivent à 1.5 A et diminue après, pour atteindre 0.24 A au régime permanent.

Les figures (4.8 b) et (4.9 b) représentent l'erreur d'estimation entre les flux rotoriques et leurs estimés, elle est importante au démarrage après elle prend de petites valeurs qui tendent vers 0.04 wb dans le régime transitoire, cette erreur augmente dans le régime permanent (0.06 wb) ou la valeur du flux augmente, et diminue après l'application du couple de charge ou la valeur de flux diminue.

L'erreur est en rapport répulsif avec la valeur des variables à estimer, ou les grandes valeurs sont estimées avec une grande marge d'erreur et vice versa.

La valeur de l'erreur obtenue en régime permanent montre clairement la convergence du FK écrit en langage C.

4.4 Résultats de l'application de l'algorithme du FK écrit en utilisant la bibliothèque IQ Math sur le DSP F2812

Les figures ci-dessous illustrent l'évolution des variables d'état mesurées (courants statoriques) et non mesurées (flux rotoriques) et leurs estimés.

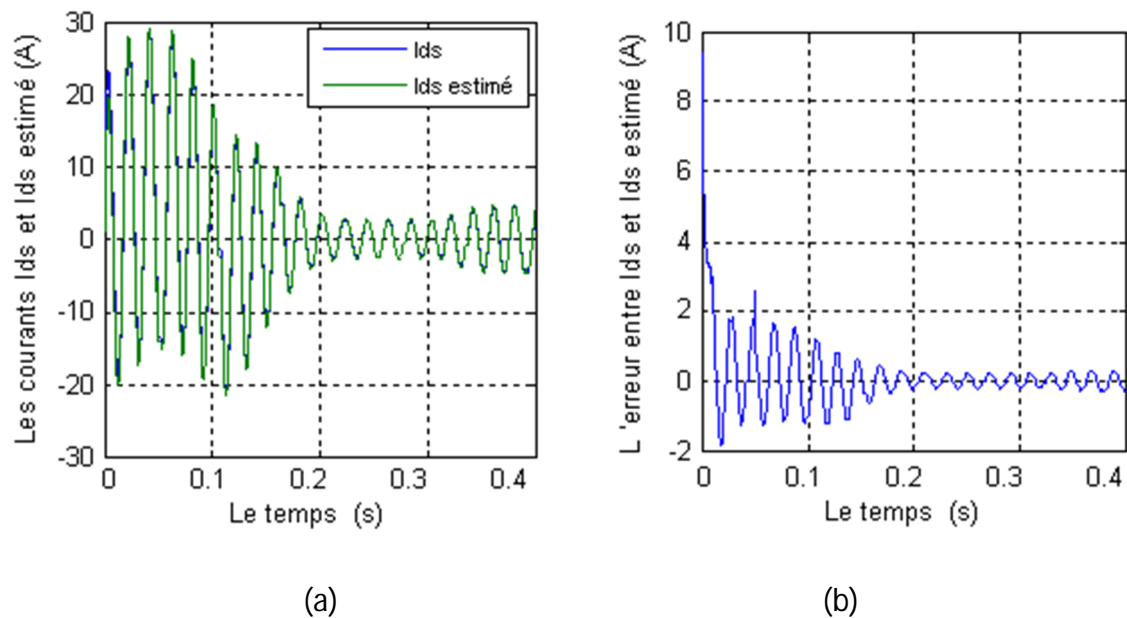
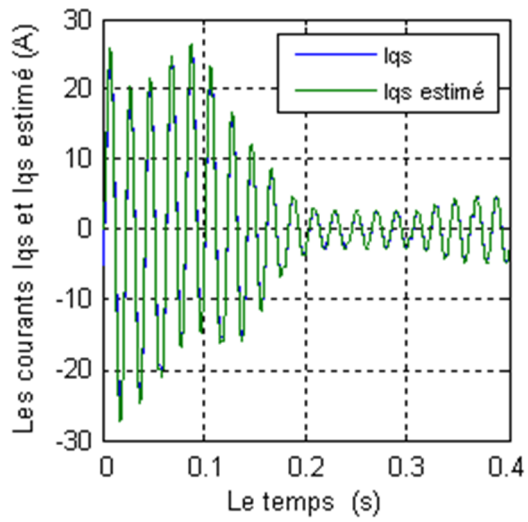
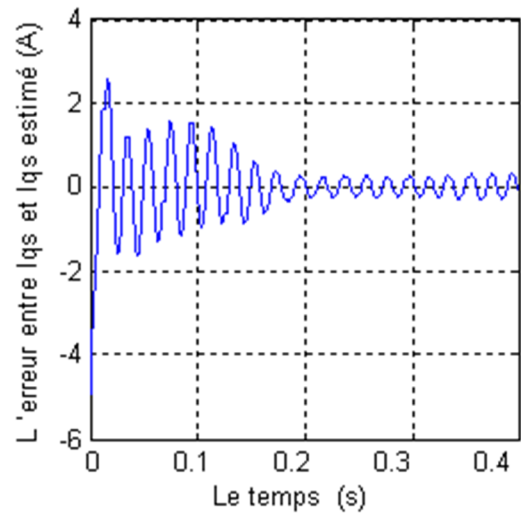


Figure : 4.10 la superposition des courants I_{ds} et \hat{I}_{ds} et l'erreur entre eux

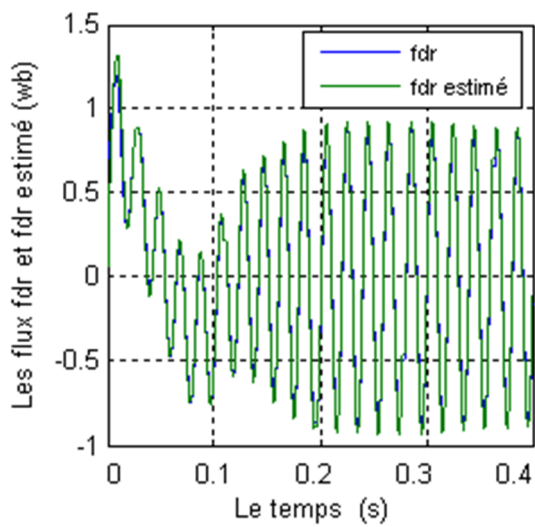


(a)

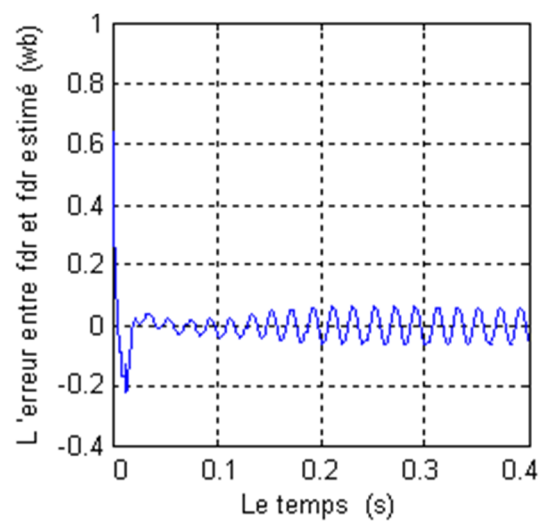


(b)

Figure : 4.11 la superposition des courants I_{qs} et \hat{I}_{qs} et l'erreur entre eux



(a)



(b)

Figure : 4.12 la superposition des flux ϕ_{dr} et $\hat{\phi}_{dr}$ et l'erreur entre eux

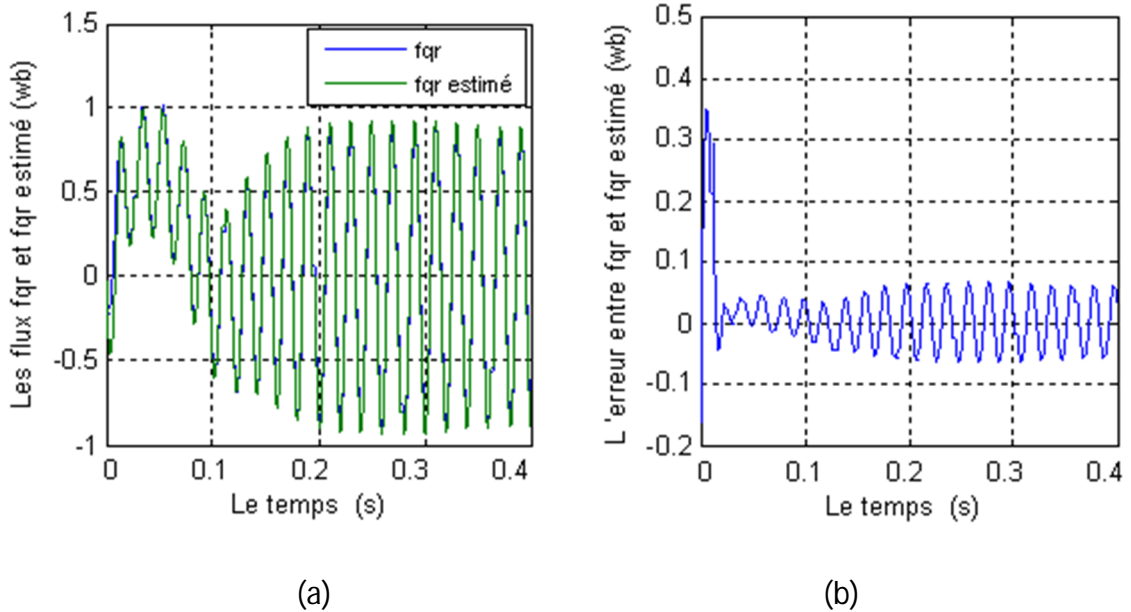


Figure : 4.13 la superposition des flux ϕ_{qr} et $\hat{\phi}_{qr}$ et l'erreur entre eux

Les mêmes résultats sont obtenus avec l'algorithme du filtre écrit en IQ Math, ce qui montre l'efficacité des fonctions de la bibliothèque IQ Math.

Remarque :

Les résultats obtenus par le FK simulé sur MATLAB sont meilleurs que ceux obtenus par implémentation sur DSP F2812. Cette différence revient à la simplification des matrices discrétisées (A_d, B_d, W_d) et à la précision avec laquelle les calculs sont faits par le DSP F2812 qui est un micro-processeur à virgule fixe.

4.5 Comparaison entre la programmation en C et la programmation en utilisant IQ Math

Occupation de l'espace mémoire :

L'algorithme du filtre écrit en programme C occupe un espace mémoire de 6642 mots:

- 2978 mots utilisés pour loger le code exécutable et les sous-programmes de l'addition, la multiplication, et la division.
- 3664 mots utilisés pour les variables globales et locales.

L'algorithme écrit en programme C utilise pour exécuter une addition un sous-programme de 120 mots.

L'algorithme du FK écrit en utilisant les fonctions d'IQ Math occupe un espace mémoire de 3084 mots :

- 2602 mots utilisés pour le code exécutable et les sous-programmes de la multiplication, la division ...
- 434 mots pour les variables globales et locales, et 40 pour des sous-programmes de transformation de format IQ au format virgule flottante.

L'espace occupé par l'algorithme écrit en programme C, est deux fois plus supérieur que celui occupé par le programme IQ.

Le temps d'exécution des deux algorithmes :

Sachons qu'une période d'échantillonnage de 10^{-3} (s) reflète la physique de la MAS.

Le temps d'exécution du programme du FK IQ pour une seule itération est de $3 * 10^{-5}$ (s), il est suffisamment petit pour exécuté en parallèle d'autre programme exemple programme d'acquisition, régulation et visualisation des résultats. Ce temps réduit est dû à l'utilisation des fonctions de la bibliothèque IQ optimisées en temps et en précision alors que pour l'algorithme écrit en programme C ce temps est $2.28 * 10^{-4}$ (s), il est sept fois plus grand que celui du programme IQ.

4.6 Application du FK à la MAS réelle

Cette étape consiste à appliquer le FK à la MAS alimenté du réseau triphasé pour l'estimation des flux rotoriques. Une acquisition des tensions d'alimentation, des courants absorbés par la MAS et une capture de la vitesse mécanique sont réalisées par une carte d'interface.

Le schéma suivant explique les différentes étapes d'estimation des flux rotoriques, et d'observation du résultat d'estimation.

4.6.1 Acquisition de la vitesse

Pour la reconstruction de l'état de la machine par le filtre de Kalman implémenté sur le DSP F2812, la vitesse mécanique est mesurée. Pour cela on va utiliser une des unités de capture d'un des gestionnaires d'évènement. Ces unités mesurent la fréquence du signal TTL délivré par l'encodeur en quadrature intégré dans la machine, et à partir de cette fréquence on déduit la vitesse mécanique de la machine.

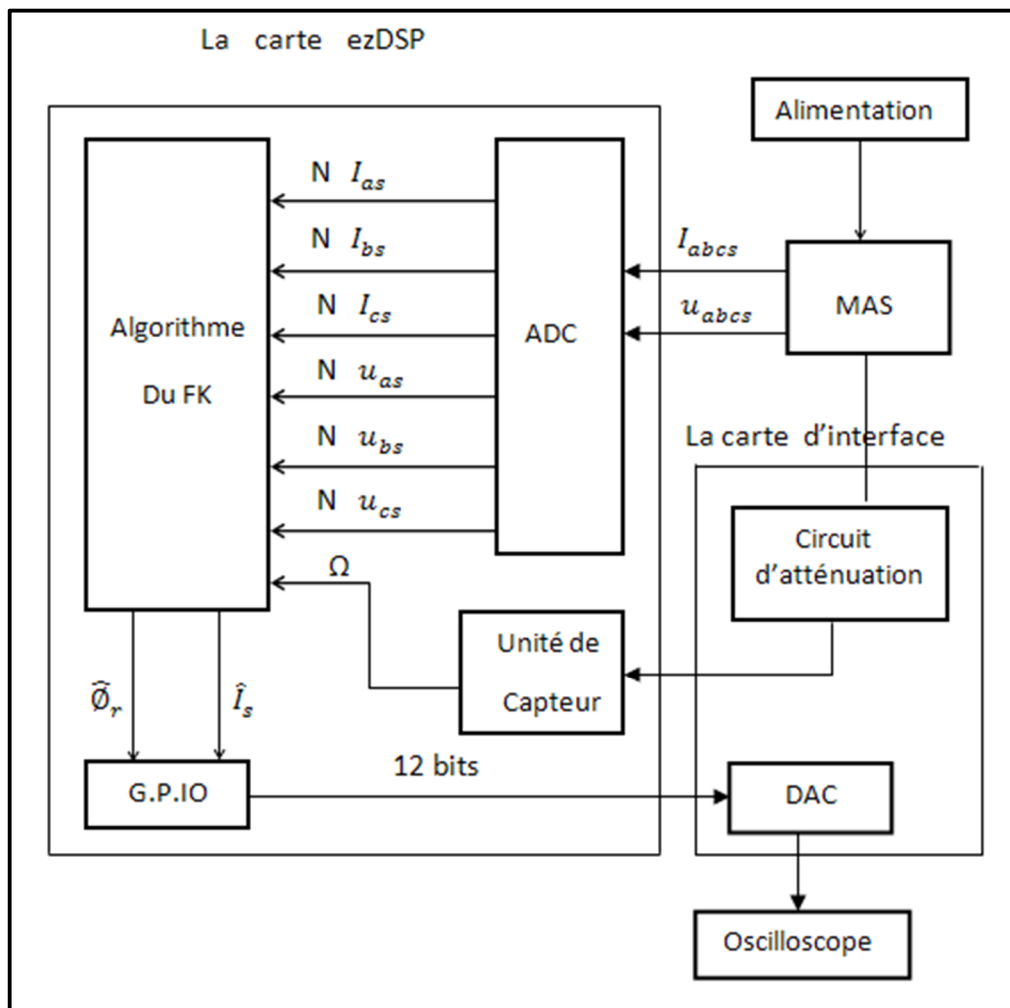


Figure : 4.14 schéma fonctionnel du banc d'essai

Un premier front montant du signal TTL permet l'enregistrement de la valeur du compteur du timer, un deuxième front montant est marqué par l'enregistrement

d'une deuxième valeur. La fréquence de fonctionnement du timer divisé par la différence entre ces deux valeurs représente la fréquence f_c du signal TTL capturé.

La période $1/f_c$ représente le temps prêt par le rotor pour tourner d'un angle élémentaire ϑ , sachant que $\vartheta = 2 * \pi/1024$, le temps prêt par le rotor pour effectuer une rotation est $1024/f_c$, d'où la vitesse de rotation en tour par seconde égale à : $v = f_c/1024$.

4.6.2 Acquisition des courants et des tensions

L'acquisition des courants statoriques et des tensions d'alimentation est réalisée par le convertisseur analogique numérique (ADC). Une carte de mise en forme des signaux captés (courants et tensions) est utilisée pour régler le niveau des signaux à l'entrée du DSP (interface analogique P5/P9).

4.6.3 Visualisation des grandeurs estimées

Les grandeurs estimées et signaux mesurés sont envoyés sur le port GPIO afin de les visualiser et les étudier. Une conversion numérique analogique externe est réalisée à l'aide d'une carte à base de convertisseur numérique analogique.

4.7 La carte d'interface

La conversion numérique analogique des signaux (courants et flux) est effectuée par une carte d'interface composée de deux circuits :

4.7.1 Le circuit d'atténuation

Le signal TTL délivré par l'encodeur de la machine (capteur de vitesse) est de valeur maximale de 5 volts, tandis que le DSP reçoit à ces entrées des tensions maximales de 3.3 volts. Une atténuation du signal TTL est effectuée par un circuit composé de trois résistances variables (de type multi tour) et d'ampli-up TL 084.

Les résistances variables : on a choisi des résistances multi tour pour leurs précisions, elles sont utilisées comme diviseur de tension pour atténuer le signal TTL de 5v à 3.3v.

Les amplis-up : ils sont utilisés comme suiveurs pour assurer que la tension des signaux TTL ne chute pas au-dessous de 3.3v quand on les injecte dans la carte ezdsp.

4.7.2 Le circuit de conversion

Etant donné que le DSP F2812 ne possède pas un convertisseur analogique numérique, un circuit de conversion externe doit être envisagé afin de convertir les données numériques sortantes du DSP.

Ce circuit est réalisé et est composé d'un convertisseur analogique numérique de 12 bits (AD7545).

4.8 Conclusion

Dans ce chapitre, nous venons de présenter :

Les résultats d'une simulation sur logiciel Matlab du filtre de Kalman appliqué à la machine asynchrone triphasée, pour l'estimation des courants statoriques et des flux rotoriques. La convergence du filtre a été obtenue avec une erreur négligeable.

Les résultats d'estimation des courants statoriques et des flux rotoriques par deux algorithmes du filtre de Kalman différents, le premier écrit en langage C et le deuxième écrit en IQ Math, implantés sur le DSP TMS320F2812. Ils sont très satisfaisants, avec une erreur très acceptable.

La comparaison entre les deux algorithmes en termes de temps d'exécution et d'occupation d'espace mémoire, a permis de voir les grandes performances de l'algorithme du filtre écrit en IQ Math.

Introduction générale

La machine asynchrone (MAS), de par sa simplicité de conception et d'entretien, est convoitée par les industriels depuis son invention par N. TESLA, quand il découvrit les champs magnétiques tournants engendrés par un système de courants polyphasés. Cette simplicité s'accompagne toutefois d'une grande complexité physique, liée aux interactions électromagnétiques entre le stator et le rotor.

De point de vue automatique, la machine asynchrone est un système dynamique qui pose un certain nombre de problèmes de commande à cause de ses caractéristiques : système non-linéaire, multi-variable et fortement couplé, dont les paramètres résistifs et inductifs varient aussi bien que la charge. De plus, certaines variables ne sont pas mesurables, notamment les flux, ce qui implique l'utilisation d'observateurs pour la commande ou l'observation.

Dans ce présent travail, on s'intéressera à l'utilisation du filtre de Kalman pour l'estimation des variables d'état mesurables (les courants statoriques) et non mesurables (les flux rotoriques). Pour cela on utilisera le DSP TMS320F2812 qui présente un ensemble de caractéristiques indispensables pour la commande des machines électriques.

Dans ce contexte, les travaux présentés ont porté sur l'application du filtre de Kalman optimal à la MAS, qui passe par la phase d'étude théorique, la simulation de modèle obtenue sur le logiciel Matlab, puis en dernière phase l'implémentation du filtre sur DSP F2812.

Ce mémoire est composé de quatre chapitres, encadrés par une introduction générale et par une conclusion. Les chapitres sont organisés de la façon suivante :

Dans le premier chapitre, on va tout d'abord présenter la machine asynchrone avec ses différents constituants. Puis, on passera à sa modélisation à l'aide des équations de base (les équations des tensions aux bornes des bobines ainsi que les équations des flux). Enfin, l'application de la transformation de Park nous permet d'obtenir un modèle à coefficients constants que l'on va utiliser pour la suite du travail.

Dans le deuxième chapitre, on va rappeler quelques notions sur l'observabilité des systèmes linéaires et non-linéaires. Puis, on citera les différents types des observateurs concernant les systèmes linéaires et non-linéaires. Après, on s'intéressera au filtre de Kalman optimal qui est l'élément central de ce travail et que l'on va appliquer à la MAS pour l'estimation des flux rotoriques.

Dans le troisième chapitre, une étude des DSP, en particulier le TMS320F2812, sera faite ; dans lequel nous exposerons également les outils de développement logiciels fournie par TEXAS INSTRUMENTS. A la fin on simplifiera le modèle du filtre écrit sous forme élémentaire, afin de l'implémenter sur le DSP F2812.

Dans le dernier chapitre, on va tirer des commentaires des graphes obtenus après avoir simulé le filtre de Kalman appliqué au modèle de la machine sur logiciel Matlab, ainsi des graphes obtenus par l'implémentation d'algorithme de FK écrit en langage C et d'algorithme de FK écrit en utilisant les fonctions de la bibliothèque IQ Math, une comparaison entre les deux programmes sera faite. A la fin nous envisageons d'appliquer l'algorithme optimal à une MAS réelle.

Liste des tableaux

Tableau : 3.1 représentations en format fixe Q5 pour N =8 bits	40
Tableau : 3.2 représentation en virgule flottante sur 8 bits avec m = 5 et e = 3.....	42
Tableau : 3.3 nombre d'opérations pour le calcul du filtre de Kalman.....	65

Listes des acronymes et abréviations

$[V_{abcs}][V_{abcr}]$: Vecteur tensions appliquées aux phases statoriques et rotoriques.

$[I_{abcs}][I_{abcr}]$: Vecteur courants appliquées aux phases statoriques et rotoriques.

$[\Phi_{abcs}][\Phi_{abcr}]$: Vecteur flux statoriques et rotoriques.

$R_s ; R_r$: Résistance d'une phase statorique et rotorique.

$[L_s][L_r]$: Matrice des inductances propres et mutuelles entre phase statoriques et rotoriques.

$[M_{sr}]$: Matrice des inductances mutuelles entre phases statoriques et rotoriques.

$[M_{rs}]$: Matrice des inductances mutuelles entre phases rotoriques et statoriques.

$[P_i]$: La matrice de transformation de Park.

θ : L'angle entre stator et rotor

θ_s : L'angle entre stator et l'axe **d**.

θ_r : L'angle entre rotor et l'axe **d**.

ω_s : La pulsation statorique.

ω_r : La pulsation rotorique.

p : Le nombre de paires de pôles.

Ω : La vitesse mécanique.

T_e : Le couple électromagnétique.

C_r : Le couple résistant de charge.

J : Moment d'inertie.

f_v : Coefficient du frottement visqueux.

σ : Le coefficient de dispersion.

T_r : Constante de temps rotorique

T_s : Constante de temps statorique.

MAS : Machine Asynchrone triphasée.

FK : Filtre de Kalman.

DSP : Digital Signal Processor.

MFLOPS : mega float operation per second.

GFLOPS : giga float operation per second.

JTAG : Joint Test Action Group.

XT : Multiplcand Register.

P : product Register.

PIE : Peripheral Interrupt Expansion.

IER : Interrupt Enable Register.

INTM : global interrupt mask bit.

ADC : Analogue Digital Converter.

SEQ : Sequencer.

EVA : Event Manager A.

EVB : Event Manager B.

PWM : Pulse-Width Modulation

QEP : Quadrature Encoder Pulse.

CCS : code composer studio.

Conclusion générale

L'objectif de notre travail est l'estimation des courants statoriques et des flux rotoriques de la machine asynchrone.

Nous avons commencé par effectuer la modélisation de la MAS dans le repère de Park (en considérant les conditions habituelles de simplification). Le model obtenu étant observable, nous avons conçu le filtre de Kalman.

La simulation sur logiciel Matlab du filtre de Kalman appliqué à la MAS a donné des résultats intéressants (erreur d'estimation très acceptable), puis on a effectué l'implémentation du FK sur un DSP F2812 à travers deux programmes différents, en l'écrivant en langage C, puis en utilisant la bibliothèque IQ Math.

Les deux programmes implémentés sur DSP, ont donné à leurs tours des résultats intéressants en termes d'erreur. Pour le temps d'exécution d'une seule itération (de programme), le temps consommé par le programme IQ Math est sept fois plus petit que celui consommé par le programme C. Tandis que l'espace mémoire consommé par ce dernier est deux fois plus grand que celui consommé par le programme IQ. L'espace mémoire occupé et le temps d'exécution optimisés par l'utilisation de la bibliothèque IQ Math, rend possible l'implémentation sur DSP d'algorithmes plus complexes telle que le filtre de Kalman étendu, et le filtre de Kalman adaptatif.

En fin on a réalisé l'environnement nécessaire pour intégrer le programme d'estimation des courants statoriques et des flux rotoriques pour une commande de la MAS réelle (insertion de régulateurs, commande vectorielle, DTC..).

Bibliographie

[1] : Bertrand Nogarede, Michel Lajoie-Mazenc, Bernard Trannoy, Henri Foch et Jean Faucher : principes d'électrotechnique, Dunod, Paris, 2005.

[2] :J. Chatelain : Machines électriques Tome 1, Dunod, 1984.

[3] : Rezegui Salah Eddine : commande de machine électrique en environnement Matlab/Simulink et Temps Réel, thèse de magister en électrotechnique, université Mentouri de Constantine.

[4] : Dr A.Meroufel : contrôle de la machine asynchrone : commande scalaire, commande vectorielle, commande directe du couple, Intelligent Control & Electrical Power Systems Laboratory (ICEPS) Université Djillali Liabès –Sidi Bel-Abbès, 2008/2009.

[5] : Malek Ghanes : Observation et commande de la machine asynchrone sans capteur mécanique, Thèse de doctorat 2005 à l'école centrale de Nantes.

[6] : Ricardo Alvares Salas : Développement de lois de commande avec observateur de la machine asynchrone, Thèse de doctorat 2002.

[7] : Dorléans, Observabilité-observateurs, Université de Caen, Master 1 année, AII & ESCI

[8] : Estelle CHERRIER : Estimation de l'état et des entrées inconnues pour une classe de systèmes non linéaires, Thèse de doctorat, INP Lorraine, France (2006)

[9] : Zedong Zheng : Commande à haute performance et sans capteur mécanique du moteur synchrone à aimants permanents, Thèse de doctorat, INP de Toulouse.

[10] : Ibrahim Hoteit : Filtre de Kalman réduits et efficaces pour l'assimilation de données en océanographie, Université de Joseph Fourier –Grenoble.

[11] :G. Baudoin et F.Violieu : Les DSP Famille TMS320C54x : Développement d'applications, Dunod, Paris, 2000.

[12] : Liping Guo : design and implementation of digital controllers for buck and boost converters using linear and nonlinear control methods, Auburn, Alabama, 2006.

[13]: Texas Instruments Teaching Materials, DSP28-Architecture, 2002.

[14]: Texas Instruments Teaching Materials, DSP28-Interrupt System, 2002.

[15]: Texas Instruments Teaching Materials, DSP28-C28x Analogue Digital Converter, 2002.

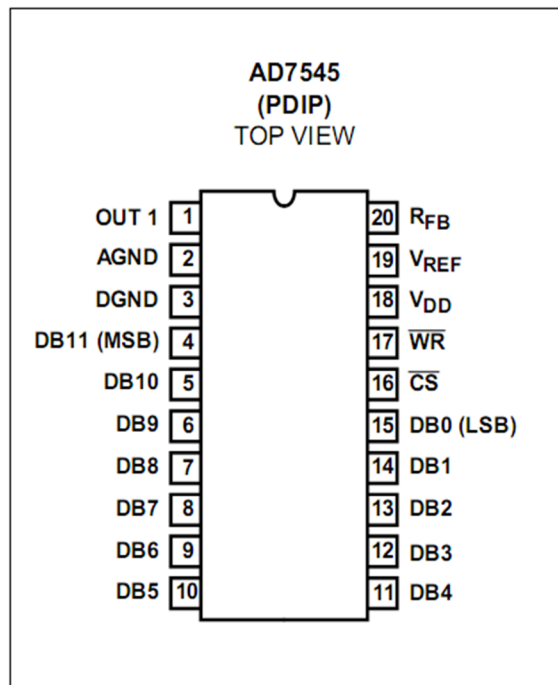
[16]: Texas Instruments Teaching Materials, DSP28-C28x Event Manager, 2002.

[17] : LOUNIS Ahmed Abd elkarim : Implémentation d'un système de communication FM par l'utilisation du DSP TMS320VC5402, ENP Alger, 2005.

[18]: Texas Instruments Teaching Materials, DSP28-IQ-Math Library, 2002.

[19]: Mickael HILAIRET : Application des outils du traitement du signal à la commande des machines tournantes, Thèse de Doctorat de l'Université de Nantes, France, 2001.

Le convertisseur analogique numérique :



La figure présente le convertisseur numérique analogique AD7545. Il peut être alimenté par une tension qui prend des valeurs de 5 à 15 volts

Schéma du circuit de conversion (numérique/analogique)

réalisé :

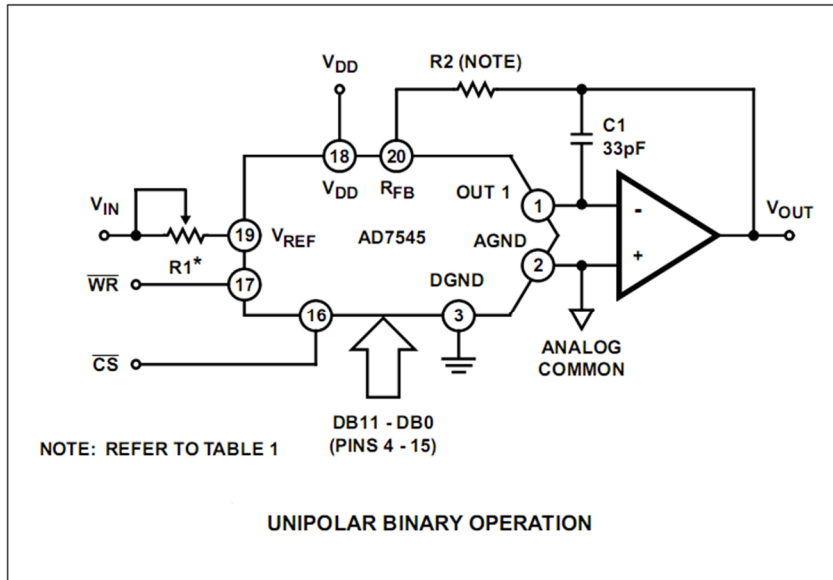


Schéma synoptique du circuit de conversion numérique/analogique :

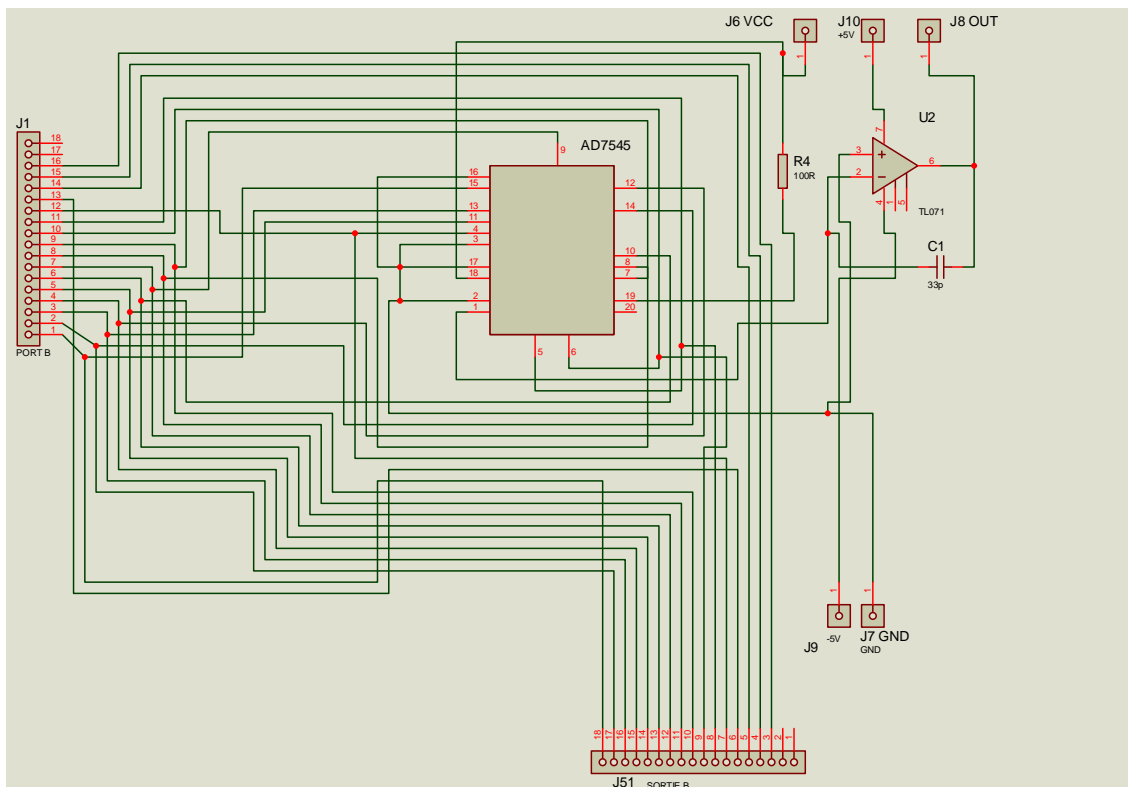
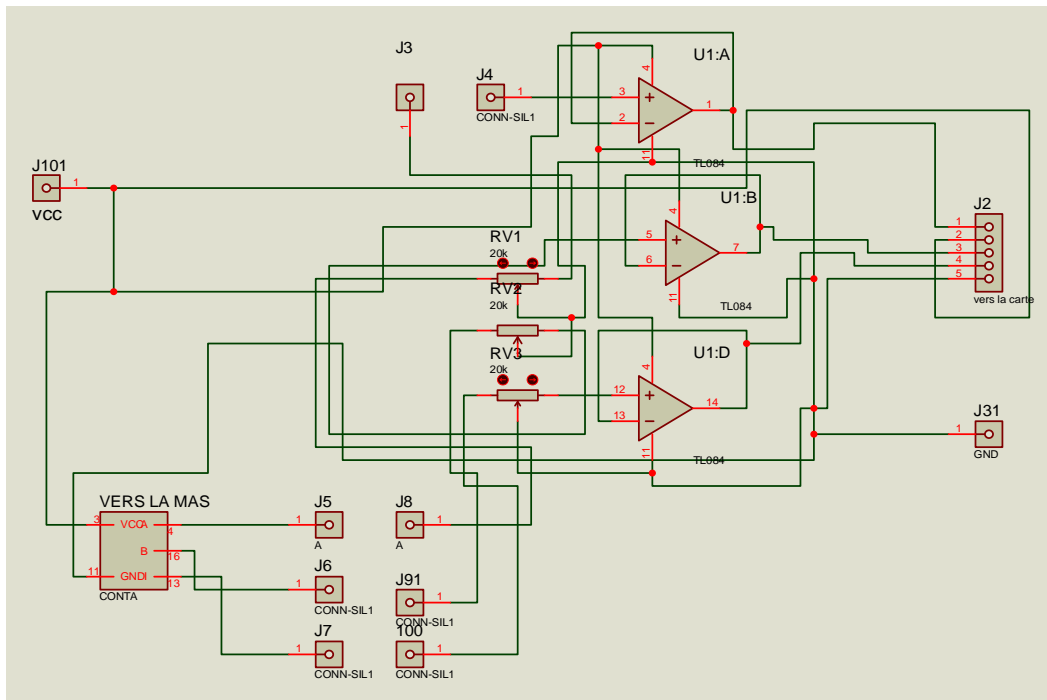
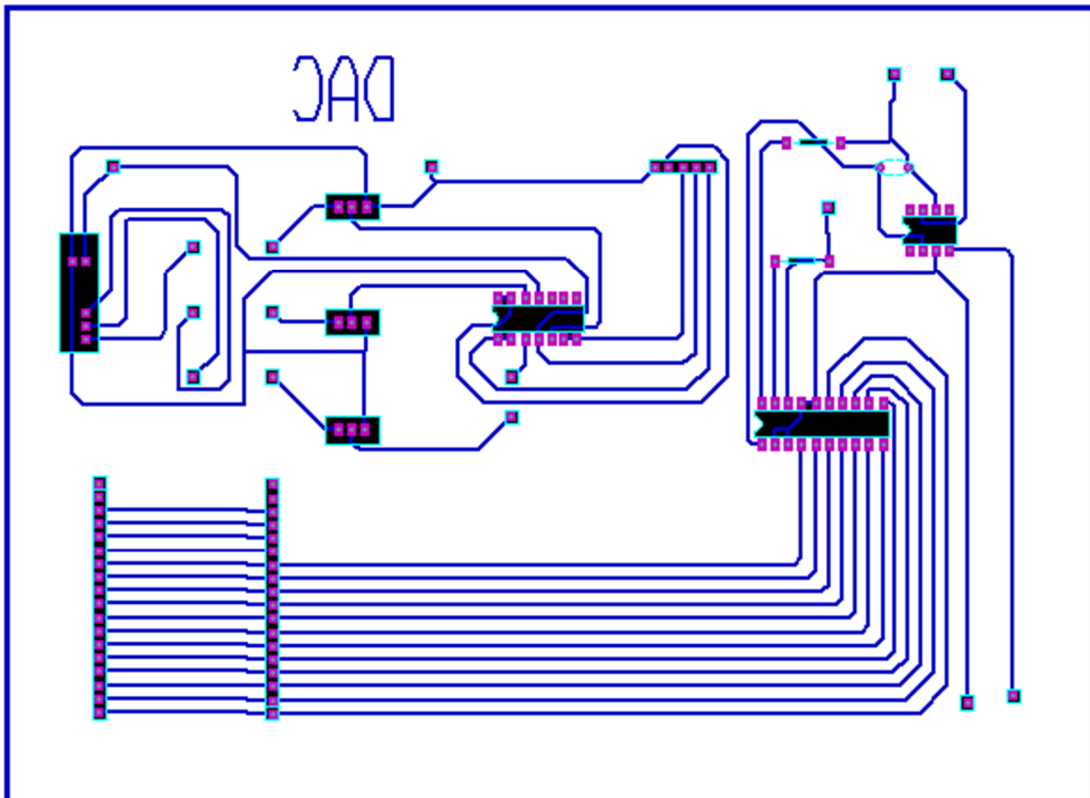


Schéma synoptique du circuit d'atténuation :



Circuit imprimé de la carte d'interface :



La carte eZdsp :

2.3 eZdsp™ F2812 Connectors

The eZdsp™ F2812 has five connectors. Pin 1 of each connector is identified by a square solder pad. The function of each connector is shown in the table below:

Table 2: eZdsp™ F2812 Connectors

Connector	Function
P1	JTAG Interface
P2	Expansion
P3	Parallel Port/JTAG Controller Interface
P4/P8/P7	I/O Interface
P5/P9	Analog Interface
P6	Power Connector

The diagram below shows the position of each connector

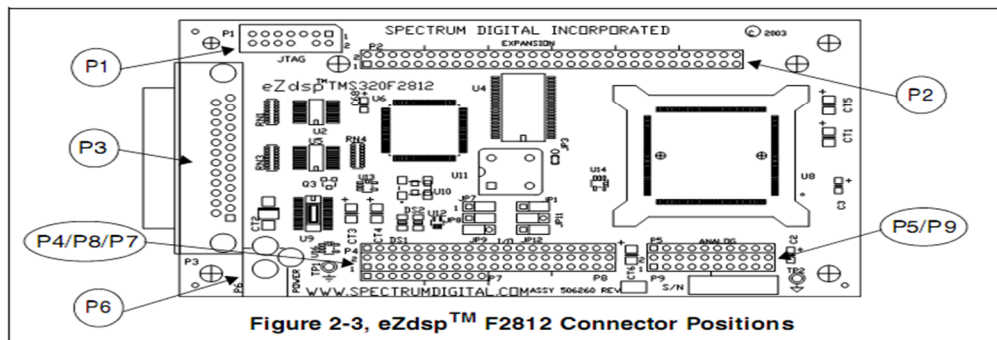
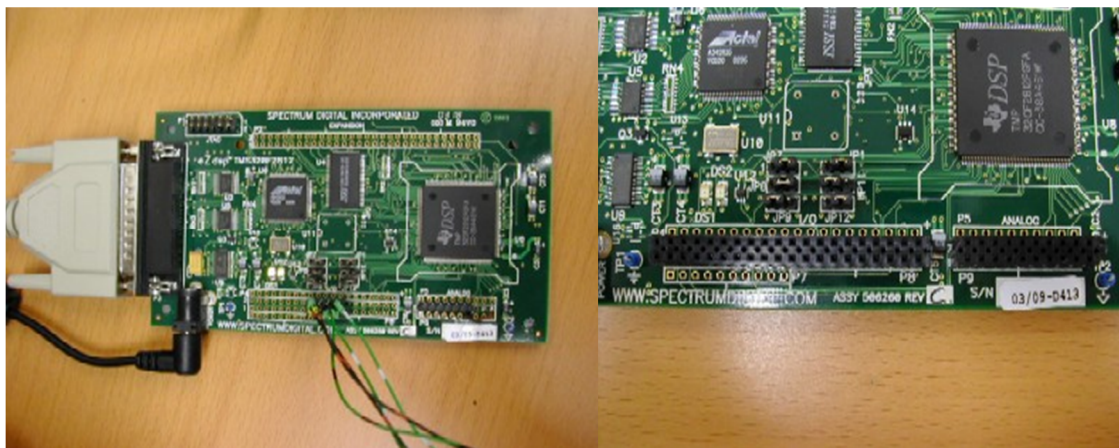
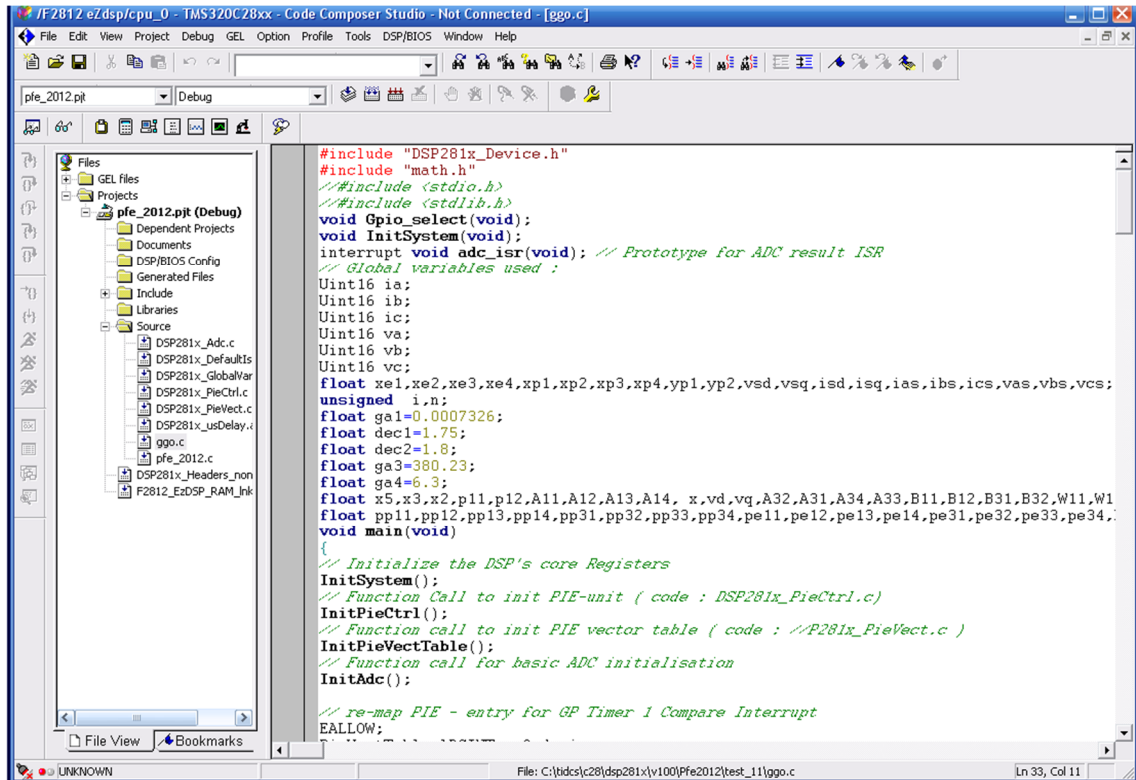


Figure 2-3, eZdsp™ F2812 Connector Positions

Image de la carte eZdsp :



Code Composer Studio :



```
#include "DSP281x_Device.h"
#include "math.h"
#include <stdio.h>
#include <stdlib.h>
void Gpio_select(void);
void InitSystem(void);
interrupt void adc_isr(void); // Prototype for ADC result ISR
// Global variables used :
Uint16 ia;
Uint16 ib;
Uint16 ic;
Uint16 va;
Uint16 vb;
Uint16 vc;
float xe1,xe2,xe3,xe4,xp1,xp2,xp3,xp4,yp1,yp2,vsd,vsq,isd,isq,ias,ibs,ics,vas,vbs,vcs;
unsigned i,n;
float ga1=0.0007326;
float dec1=1.75;
float dec2=1.8;
float ga3=380.23;
float ga4=6.3;
float x5,x3,x2,p11,p12,A11,A12,A13,A14, x,vd,vq,A32,A31,A34,A33,B11,B12,B31,B32,W11,W1
float pp11,pp12,pp13,pp14,pp31,pp32,pp33,pp34,pe11,pe12,pe13,pe14,pe31,pe32,pe33,pe34.
void main(void)
{
// Initialize the DSP's core Registers
InitSystem();
// Function Call to init PIE-unit ( code : DSP281x_PieCtrl.c)
InitPieCtrl();
// Function call to init PIE vector table ( code : //P281x_PieVect.c )
InitPieVectTable();
// Function call for basic ADC initialisation
InitAdc();

// re-map PIE - entry for GP Timer 1 Compare Interrupt
EALLOW;
```