

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

**UNIVESIE MOHAMED KHEIDER BISKRA**  
**FACULTE DES SCIENCES ET SCIENCES DE L'INGENIEUR**  
**DEPARTEMENT D'ELECTRONIQUE**

## **MEMOIRE**

Présenté pour l'obtention de diplôme de magistère en  
**ELECTRONIQUE**

**Option**  
**Architecture Des Systèmes**

### **THEME**

**IMPLANTATION D'UNE IDENTIFICATION EN TEMPS REEL**  
**DE LA MACHINE ASYNCHRONE A CAGE SUR LE**  
**DSP TMS320 LF 2407 A**

PAR  
**AOURAGH NABIL**

#### **Soutenu devant le jury**

<b>Président</b>	<b>: N.E SENGOUGA</b>	<b>Prof.</b>	<b>Univ de Biskra</b>
<b>Rapporteur</b>	<b>: A.R DEBILOU</b>	<b>Dr.</b>	<b>Univ de Biskra</b>
<b>Examineurs</b>	<b>: M BEDDA</b>	<b>Prof.</b>	<b>Univ de Annaba</b>
	<b>A.H BEN AKCHA</b>	<b>M.C.</b>	<b>Univ de Biskra</b>
	<b>A KHELIFA</b>	<b>Dr.</b>	<b>Univ de Biskra</b>

**2003/2004**

## ***Résumé***

Ce travail a pour objectif l'implantation d'une technique d'identification et d'estimation des grandeurs non accessibles à la mesure par filtre de Kalman linéaire et non linéaire de la machine asynchrone. Les algorithmes d'identification, développés conformément à la formulation du filtre de Kalman appliqué à la machine, sont validés par simulation sous MATLAB SIMULINK.

Autour du système de développement à processeur dédié aux applications de contrôle et de commande des machines électriques le « DSP TMS 320 LF 2407 EVM » de *Texas Instrument*, nous avons développé un banc d'essais pour l'étude et la mise au point d'algorithmes en temps réel de contrôle de la machine. Le banc comprenant en plus de la machine asynchrone, deux modules électroniques pour le filtrage et de la mise en forme des signaux mesurables et un variateur de vitesse le « L100 » de HITACHI. L'algorithme d'estimation en temps réel de l'état rotorique ( flux rotoriques ) de la machine a été implanté avec succès sur ce dispositif.

**Mots clés :** identification, temps réel, machine asynchrone, filtre de Kalman linéaire, filtre de Kalman étendu, processeur de traitement de signal TMS320LF2407.

## ***Abstract***

This work is for objective the development of identification and estimation technics of a state which are not directly measured by linear Kalman filtering and extended Kalman filtering of the asynchronous machine. The identification algorithms by Kalman filter applied to the induction machine are tested for validation by simulation under MATLAB SIMULINK environment.

Around a development system based on digital signal processor " DSP TMS 320 LF 2407 EVM " of *Texas Instrument*, for application to monitoring and control of electric machines, we have developed a test bench for study and integration of real time monitoring algorithms of induction machine. The device including in addition to the asynchronous machine, two electronics modules for filtering and adjustment of measurable signals and a " L100 " voltage inverter of HITACHI. The real time estimation algorithm of the rotor state ( rotor flux ) of the induction machine was successfully developed on this device.

**Key words** : identification, real time, induction motor, linear Kalman filter, extended Kalman filter, signal processing processor TMS320LF2407.

## *Remerciements*

Le travail présenté dans ce mémoire a été effectué au « Laboratoire des Systèmes Experts, l'Imagerie et leurs Application dans l'ingénierie ( LESIA ) ». Je remercie vivement son directeur Monsieur Z.E. BAARIR de m'avoir permis d'effectuer ce travail dans les meilleures conditions.

Je remercie tout particulièrement mon promoteur Monsieur A.R DEBILOU qui m'a soutenu scientifiquement et pédagogiquement tout au long de la réalisation de ce mémoire. Je le remercie encore pour son amitié, sa disponibilité, son honnêteté et ses qualités humaines qui m'ont permis de réaliser ce travail dans les meilleurs conditions.

Je tiens à remercier les membres du jury : Messieurs N.E. SENGOUGA, A.H. BEN AKCHA M. BEDDA et A. KHELIFA pour avoir accepté de juger mon travail.

Je remercie beaucoup ma famille, surtout mes parents pour leurs aides et soutien.

Je remercie également M<sup>lle</sup> A. GARGAZI, membre de l'équipe de recherche, pour ses conseils et soutien durons la période de mémoire.

Enfin , un grand merci à mes amis qui m'ont supporté tout long de ces deux ans de travail, Djamel TAIBI, Khaled YAHIA, Azzdine CHAIBA , Tahar CHATOIH, Samir KOUTI, Tarek MEDDOUR, Nacer-Eddine RAHMANI, SALAH RAHMANI, Abd-Razzak RAHMANI, A.A. ABD-ALLAIDOU, F. BENDAHANE, Mohamed MADDOUR, Yacine HADDANE ... etc.

# *Dédicace*

*Je dédie ce mémoire :*

*A ma mère.*

*A mon père.*

*A mes frères et sœurs.*

*A la mémoire de ma sœur Iman.*

*A tout mes amis.*

*Nabil Souragh*

# Sommaire

<b>Introduction générale</b> .....	1
<b>Chapitre 1 : Filtre de Kalman linéaire</b>	
1.1. Introduction .....	3
1.2. Approche de Kalman .....	3
1.3. Formulation et Equations du filtre de Kalman discret .....	6
1.3.1. Expression de $P(k,k)$ .....	7
1.3.2. Détermination du gain $K(k)$ .....	8
1.4. Mise en œuvre du filtre de Kalman .....	9
1.5. Algorithme du filtre de Kalman .....	10
1.6. Choix des valeurs initiales .....	11
1.7. Exemple d'application du filtre de Kalman .....	12
1.7.1. Modèle du système .....	12
1.7.2. Equation et paramètres du filtre .....	12
1.7.3. Simulation du système sous MATLAB .....	13
1.7.4. Signal d'entrée du filtre de Kalman .....	14
1.7.5. Résultats de simulation .....	15
1.8. Conclusion .....	17
<b>Chapitre 2 : Estimation des flux rotoriques de la machine asynchrone à cage par le filtre de Kalman</b>	
2.1. Introduction .....	18
2.2. Modélisation de la machine asynchrone à cage .....	19
2.2.1. Description du moteur asynchrone à cage .....	19
2.2.2. Hypothèses de simplification .....	19

2.2.3. Modèle de la machine asynchrone dans le plan $a, b, c$ .....	20
2.2.3.1. Equations électriques .....	20
2.2.3.2. Equations magnétiques .....	21
2.2.3.3. Equation mécanique .....	22
2.2.3.4. Transformation de PARK .....	22
2.2.4. Modèle de la machine asynchrone dans le repère de Park .....	24
2.2.5. Modèle d'état de la machine asynchrone .....	26
2.3. Simulation d'un démarrage à vide de la machine asynchrone alimentée par réseau électrique triphasé .....	28
2.3.1. Schéma en bloc de simulation .....	28
2.3.2. Résultats de la simulation .....	30
2.3.3. Commentaires .....	32
2.4. Estimation des flux rotoriques de la machine par le filtre de Kalman .....	32
2.4.1. Définition du vecteur d'état .....	32
2.4.2. Discrétisation du modèle d'état de la machine asynchrone .....	33
2.4.3. Equations du filtre de Kalman linéaire .....	34
2.4.5. Bruitage des signaux .....	36
2.4.6. Schéma en bloc de la simulation de l'estimation des flux rotoriques de la machine asynchrone par le filtre de Kalman linéaire .....	36
2.4.7. Temps d'échantillonnage .....	37
2.4.8. Résultats de simulation .....	37
2.4.8.1. Grandeurs d'entrées pour le filtre de Kalman .....	37
2.4.8.2. Résultats d'estimation .....	39
2.5. Conclusion .....	41

### **Chapitre 3 : Estimation de la vitesse de la machine asynchrone par le filtre de kalman étendu**

3.1. Introduction .....	42
3.2. Filtre de Kalman étendu .....	42
3.2.1. Equations du filtre de Kalman étendu .....	44
3.2.2. Algorithme du filtre de Kalman étendu .....	45
3.3. Estimation de la vitesse rotorique de la machine asynchrone .....	47

3.3.1. Modèle d'état de la machine pour le filtre de Kalman étendu .....	48
3.3.2. Modèle discret de la machine pour filtre de Kalman étendu .....	48
3.3.3. Equations du filtre de Kalman étendu appliqué a la machine .....	50
a - Equation de prédiction d'état .....	50
b - Equation de prédiction de covariance .....	50
c - Equation de calcul du gain de filtre de Kalman .....	51
d - Equation de correction d'état estimé .....	51
e - Equation de correction de la covariance de l'erreur d'estimation .....	51
3.4. Simulation de l'estimation en boucle ouverte de la vitesse rotorique par filtre de Kalman étendu .....	51
3.4.1. Résultats de la simulation .....	52
3.5. Conclusion .....	55

## Chapitre 4 : Dispositif expérimental

4.1. Introduction .....	57
4.2. Plate-forme expérimentale .....	57
4.2.1. La machine asynchrone à cage .....	58
4.2.2. L'onduleur triphasé .....	59
4.2.2.1. Principe de fonctionnement de la modulation de largeur d'impulsion .....	60
4.2.2.2. Modulation MLI a 3 phases .....	60
4.2.3. Module de développement TMS 320 LF 2407 EVM .....	62
4.2.3.1. Caractéristiques du module de développement TMS320 LF 2407 EVM .....	63
4.2.3.2. Schéma fonctionnel du module TMS320 LF 2407 EVM .....	63
4.2.3.3. Architecture interne du DSP TMS 320 LF 2407 .....	65
4.2.3.4. Organisation de la mémoire du DSP TMS 320 LF 2407 .....	67
4.2.4. Module des capteurs .....	68
4.2.4.1. Fonctionnement des capteurs utilisés .....	68
4.2.4.2. Choix de la résistance de sortie des capteurs de courant .....	69
4.2.5. Carte d'adaptation et de filtrage .....	70
4.2.5.1. Fonction de transfert du filtre .....	71
4.2.5.2. Calcul valeurs des composants .....	73
4.2.5.3. Test des filtres .....	74



4.2.5.4. Adaptation des tensions .....	75
4.2.6. Mesure de la vitesse .....	76
4.3. Ajustement des grandeurs mesurées .....	77
4.3.1. Ajustement des tensions .....	77
4.3.2. Ajustement et calibrage des courants .....	77
4.3.3. Calibrage de la valeur de la vitesse rotorique .....	78
4.4. Validation de la plate-forme expérimental .....	78
4.5. Conclusion .....	80

## **Chapitre 5 : Implantation des algorithmes d'estimation des flux rotoriques par le filtre de Kalman linéaire**

5.1. Introduction .....	81
5.2. Outil de développement Code Composer Studio « CCS » .....	81
5.2.1. Utilitaire de génération de code .....	82
5.2.2. Emulation hardware et échange de données en temps réel .....	83
5.2.3. Exemples d'application .....	84
5.3. Schéma d'implantation du filtre de Kalman appliqué a la machine .....	85
5.4. Organigramme d'estimation .....	86
5.4.1. Partie initialisation .....	87
5.4.1.1. Configuration du DSP .....	87
5.4.1.2. Choix de la fréquence d'échantillonnage .....	88
a - Entre du TIMER .....	89
b - Soties du TIMER .....	89
c - Contrôle du TIMER .....	90
5.4.1.3. Initialisation des entrées analogiques .....	90
5.4.2. Acquisition des grandeurs analogiques pour le filtre de Kalman .....	92
5.4.3. Algorithme de filtre de Kalman .....	92
5.4.4. Temps d'exécution de l'algorithme développer .....	96
5.5. Initialisation du filtre de Kalman .....	96
5.6. Chronogramme d'exécution de l'algorithme du filtre de Kalman .....	97
5.7. Résultat expérimental de l'estimation .....	99
5.8. Conclusion .....	102

<b>Conclusion et perspectives</b> .....	103
Annexe A .....	105
Annexe B .....	111
Bibliographie .....	121

## Liste des figures

<b>Figure ( 1.1 )</b>	: Représentation de filtre de Wiener .....	4
<b>Figure ( 1.2 )</b>	: Etapes de prédiction et de correction du filtre de Kalman .....	9
<b>Figure ( 1.3 )</b>	: Processus itératif du filtre de Kalman linéaire .....	10
<b>Figure ( 1.4 )</b>	: Organigramme du filtre de Kalman linéaire .....	11
<b>Figure ( 1.5 )</b>	: Schéma bloc d'ensemble système + filtre de Kalman .....	13
<b>Figure ( 1.6 )</b>	: Signal d'entrée original ( sinusoïdal ) .....	14
<b>Figure ( 1.7 )</b>	: Bruit blanc injecté .....	14
<b>Figure ( 1.8 )</b>	: Signal d'entrée du filtre de Kalman ( signal mesuré) .....	14
<b>Figure ( 1.9 )</b>	: Signal estimé ( - ) avec le signal original ( ..) .....	15
<b>Figure ( 1.10 )</b>	: Erreur d'estimation .....	15
<b>Figure ( 1.11 )</b>	: Covariance de l'erreur d'estimation .....	15
<b>Figure ( 1.12 )</b>	: Signal estimé ( - ) avec le signal original ( ..) pour $R = 10$ .....	16
<b>Figure ( 1.13 )</b>	: Erreur d'estimation pour $R = 10$ .....	16
<b>Figure ( 1.14 )</b>	: Signal estimé ( - ) avec Le signal original ( ..) pour $R = 0.1$ .....	16
<b>Figure ( 1.15 )</b>	: Erreur d'estimation pour $R = 0.1$ .....	17
<b>Figure ( 2.1 )</b>	: Structure de la machine asynchrone à cage .....	19
<b>Figure ( 2.2 )</b>	: Représentation de la machine asynchrone à cage .....	20
<b>Figure ( 2.3 )</b>	: Représentation physique de la transformation de Park pour une machine asynchrone .....	24
<b>Figure ( 2.4 )</b>	: Schéma équivalent de la machine asynchrone à cage .....	24
<b>Figure ( 2.5 )</b>	: Schéma en bloc de simulation de la machine asynchrone .....	28
<b>Figure ( 2.6 )</b>	: Schéma du « bloc alimentation » .....	28
<b>Figure ( 2.7 )</b>	: Schéma du bloc « modèle de la machine asynchrone » .....	29
<b>Figure ( 2.8 )</b>	: Tensions d'alimentation $v_a, v_b$ et $v_c$ .....	30
<b>Figure ( 2.9 )</b>	: Transformation de Park des tensions d'alimentation $u_{ds}$ $u_{qs}$ .....	30

<b>Figure ( 2.10 )</b>	: Courants statoriques $i_{ds}$ et $i_{qs}$ .....	31
<b>Figure ( 2.11 )</b>	: Flux rotorique $\psi_{dr}$ et $\psi_{qr}$ .....	31
<b>Figure ( 2.12 )</b>	: Vitesse de rotation rotorique .....	31
<b>Figure ( 2.13 )</b>	: Couple électromagnétique .....	32
<b>Figure ( 2.14 )</b>	: Diagramme en bloc du filtre de Kalman appliqué a la machine asynchrone .....	35
<b>Figure ( 2.15 )</b>	: Bruit blanc .....	36
<b>Figure ( 2.16 )</b>	: Schéma en blocs de la simulation de l'estimation des flux rotoriques .....	37
<b>Figure ( 2.17 )</b>	: Tensions d'entrées bruité du filtre de Kalman .....	38
<b>Figure ( 2.18 )</b>	: Courant d'entrées bruité du filtre de Kalman .....	38
<b>Figure ( 2.19 )</b>	: Vitesse d'entrées bruité du filtre de Kalman .....	38
<b>Figure ( 2.20 )</b>	: Courants statoriques $i_{ds}$ simulé ( .. ) et estimé ( - ) .....	39
<b>Figure ( 2.21 )</b>	: Courants statoriques $i_{qs}$ simulé ( .. ) et estimé ( - ) .....	39
<b>Figure ( 2.22 )</b>	: Flux rotorique $\psi_{dr}$ simulé ( .. ) et estimé ( - ) .....	40
<b>Figure ( 2.23 )</b>	: Flux rotorique $\psi_{qr}$ simulé ( .. ) et estimé ( - ) .....	40
<b>Figure ( 2.24 )</b>	: Erreurs d'estimation $i_{ds}$ ( .. ) et $i_{qs}$ ( - ) .....	40
<b>Figure ( 2.25 )</b>	: Erreurs d'estimation $\psi_{dr}$ ( .. ) et $\psi_{qr}$ ( - ) .....	41
<b>Figure ( 3.1 )</b>	: Processus itératif du filtre de Kalman étendu .....	46
<b>Figure ( 3.2 )</b>	: Organigramme du filtre de Kalman étendu .....	47
<b>Figure ( 3.3 )</b>	: Schéma en bloc de simulation du EKF appliqué sur la machine asynchrone .....	52
<b>Figure ( 3.4 )</b>	: Courants statoriques $i_{ds}$ simulé ( .. ) et estimé ( - ) .....	53
<b>Figure ( 3.5 )</b>	: Courants statoriques $i_{qs}$ simulé ( .. ) et estimé ( - ) .....	53
<b>Figure ( 3.6 )</b>	: Erreurs d'estimation $i_{ds}$ ( .. ) et $i_{qs}$ ( - ) .....	53
<b>Figure ( 3.7 )</b>	: Flux rotorique $\psi_{dr}$ simulé ( .. ) et estimé ( - ) .....	54
<b>Figure ( 3.8 )</b>	: Flux rotorique $\psi_{qr}$ simulé ( .. ) et estimé ( - ) .....	54
<b>Figure ( 3.9 )</b>	: Erreurs d'estimation $\psi_{dr}$ ( .. ) et $\psi_{qr}$ ( - ) .....	54
<b>Figure ( 3.10 )</b>	: Vitesse rotorique $w_r$ simulé ( .. ) et estimé ( - ) .....	55
<b>Figure ( 3.11 )</b>	: Erreurs d'estimation de la vitesse rotorique .....	55
<b>Figure ( 4.1 )</b>	: Plate-forme expérimental .....	58
<b>Figure ( 4.2 )</b>	: Structure d'un onduleur MLI .....	59
<b>Figure ( 4.3 )</b>	: Principe de génération du signal MLI par Comparaison .....	60
<b>Figure ( 4.4 )</b>	: L'onduleur « L100 » de <i>HITACHI</i> .....	62

<b>Figure ( 4.5 )</b>	: Schéma bloc de la carte DSP .....	64
<b>Figure ( 4.6 )</b>	: Vu de haut de la carte DSP .....	64
<b>Figure ( 4.7 )</b>	: Schéma fonctionnel de l'architecture interne du DSP TMS 320 LF 2407 .....	66
<b>Figure ( 4.8 )</b>	: Organisation de la mémoire du DSP TMS 320 LF 2407 .....	67
<b>Figure ( 4.9 )</b>	: Canal d'acquisition pour chaque capteur .....	68
<b>Figure ( 4.10 )</b>	: Brochage du capteur de courant .....	69
<b>Figure ( 4.11 )</b>	: Brochage du capteur de tension .....	69
<b>Figure ( 4.12 )</b>	: Courbe courant résistance du capteur de courant LEM .....	69
<b>Figure ( 4.13 )</b>	: Vu de haut de la carte capteur .....	70
<b>Figure ( 4.14 )</b>	: Tension et courant de sortie de l'onduleur alimentant la machine .....	71
<b>Figure ( 4.15 )</b>	: Structure de Filtre passe bas .....	71
<b>Figure ( 4.16 )</b>	: Cellule biquadratique de Sallen et Key .....	72
<b>Figure ( 4.17 )</b>	: Diagramme de Bode de la fonction de transfert Du filtre .....	74
<b>Figure ( 4.18 )</b>	: Représentation des signaux d'entrées et de sorties des filtres réalisé ....	75
<b>Figure ( 4.19 )</b>	: Schéma électrique de l'adaptateur des signaux .....	75
<b>Figure ( 4.20 )</b>	: Mise en forme et adaptation des signaux .....	76
<b>Figure ( 4.21 )</b>	: Carte de mise en forme et adaptation .....	76
<b>Figure ( 4.22 )</b>	: Vue d'ensemble de la plate-forme expérimental réalisé .....	79
<b>Figure (4.23)</b>	: Tensions d'alimentation statoriques de la machine asynchrone .....	79
<b>Figure (4.24)</b>	: Courants statoriques de la machine asynchrone .....	80
<b>Figure ( 5.1 )</b>	: Etapes de développement d'une application sur DSP sous CCS .....	82
<b>Figure ( 5.2 )</b>	: Déroulement typique de génération de code exécutable d'une application sur DSP sous CCS .....	83
<b>Figure ( 5.3 )</b>	: Exemple d'application sur CCS C2000 .....	84
<b>Figure ( 5.4 )</b>	: Schéma de mise en œuvre de l'implantation .....	85
<b>Figure ( 5.5 )</b>	: Organigramme d'initialisation .....	86
<b>Figure ( 5.6 )</b>	: Organigramme de filtre de Kalman .....	87
<b>Figure ( 5.7 )</b>	: Registres de configuration du DSP .....	88
<b>Figure ( 5.8 )</b>	: Schéma bloc du temporisateur ( TIMER ) .....	89
<b>Figure ( 5.9 )</b>	: Registres de contrôles du TIMER .....	90
<b>Figure ( 5.10 )</b>	: Structure du module de conversion ADC .....	91
<b>Figure ( 5.11 )</b>	: Registres de contrôle du module ADC .....	92

<b>Figure ( 5.12 )</b>	: Temps d'exécution des différent segments d'algorithme développer ....	98
<b>Figure (5.13)</b>	: Courants statoriques $i_{ds}$ et $i_{qs}$ dans le plan de Park .....	99
<b>Figure (5.14)</b>	: Tensions statoriques $u_{ds}$ et $u_{qs}$ dans le plan de Park .....	100
<b>Figure (5.15)</b>	: Courant statorique mesuré $i_{ds}$ et estimé $\hat{i}_{ds}$ .....	100
<b>Figure (5.16)</b>	: Courant statorique mesuré $i_{qs}$ et estimé $\hat{i}_{qs}$ .....	101
<b>Figure (5.17)</b>	: Flux rotoriques estimés $\hat{\psi}_{dr}$ et $\hat{\psi}_{qr}$ .....	101

## *Liste des tableaux*

<b>Tableau (3.1)</b>	: Equations de prédiction .....	46
<b>Tableau (3.2)</b>	: Equations de correction .....	46
<b>Tableau (5.1)</b>	: Opérations de multiplications et d'addition de l'équation $\hat{x}(k+1, k)$ .....	93
<b>Tableau (5.2)</b>	: Opérations de multiplication et d'addition de l'équation $P(k+1, k)$ .....	94
<b>Tableau (5.3)</b>	: Opérations de multiplication et d'addition de 3 <sup>eme</sup> équation $K(k+1)$ .....	94
<b>Tableau (5.4)</b>	: Opérations de multiplication et d'addition de l'équation $\hat{x}(k+1, k+1)$ ...	95
<b>Tableau (5.5)</b>	: Opérations de multiplication et d'addition de 'équation $P(k+1, k+1)$ ....	95
<b>Tableau (5.6)</b>	: Affectation des états des <i>LEDs</i> pour chaque segment de programme .....	97

## ***Notations***

$m(t)$	: signal observable.
$x$	: vecteur d'état.
$S(t)$	: Signal utile.
$b(t)$	: Bruit blanc.
$\hat{S}(t)$	: Estimation du signal $S(t)$ .
$u(k)$	: Signal de commande.
$x_0$	: état initial.
$W(k)$	: Bruit d'état.
$V(k)$	: Bruits de mesure.
$\hat{x}(k/n)$	: Estimation de l'état à l'instant $k$ .
$\{y(1), y(2), \dots, y(n)\}$	: Mesures
$\Phi(k+1, k)$	: Matrice de transition.
$\hat{x}(k+1, k)$	: Prédiction de l'état $\hat{x}(k, k)$ pour l'instant $k+1$ .
$K(k+1)$	: Gain du filtre de Kalman.
$\tilde{x}(k)$	: Erreur d'estimation.
$P(k, k)$	: Matrice de la covariance de l'erreur d'estimation.
$P(k+1, k)$	: Matrice de la covariance de l'erreur d'estimation à priori.
$P(k+1, k+1)$	: Matrice de la covariance de l'erreur d'estimation a posteriori.
$E[ \ ]$	: Espérance mathématique.
$J$	: trace d'une matrice.
$A$	: Matrice d'évolution.
$B$	: Matrice de commande.
$C$	: Matrice d'observation.
$R$	: Matrice de covariance du bruit de mesure.
MLI	: Modulation de largeur d'impulsion.
$[u_s]$	: Vecteurs des tensions statoriques.



---

$[ u_r ]$	: Vecteurs des tensions rotoriques.
$[ i_s ]$	: Vecteurs des courants statoriques.
$[ i_r ]$	: Vecteurs des courants rotoriques.
$[ \psi_s ]$	: Vecteurs des flux statoriques.
$[ \psi_r ]$	: Vecteurs des flux rotoriques.
$[ R_s ]$	: Matrices des résistances statoriques.
$[ R_r ]$	: Matrices des résistances rotoriques.
$L_s$	: Inductances propres statoriques.
$L_r$	: Inductances propres rotoriques.
$M_s$	: Inductance mutuelle entre phases statoriques.
$M_r$	: Inductance mutuelle entre phases rotoriques.
$M_{sr}$	: Inductance mutuelle Maximale entre une phase du stator et les phases correspondantes du rotor.
$C_r$	: Couple résistant de la charge.
$J$	: Moment d'inertie total de l'arbre de moteur.
$f_r$	: Coefficient de frottement.
$\Omega_r = \frac{\omega_r}{p}$	: La vitesse de rotation mécanique.
$p$	: Le nombre de paires de pôles.
$d$	: Axe direct.
$q$	: Axe en quadrature.
$O$	: Axe homopolaire.
$[A]$	: Matrice de transformation de PARK.
$C_e$	: Couple électromagnétique.
$p$	: Opérateur différentiel.
$u = [V_{ds} \quad V_{qs}]^T$	: Grandeurs d'entrée.
$y = [i_{ds} \quad i_{qs}]^T$	: Grandeurs mesurés.
$e_i(k)$	: Harmoniques injectées.
$U_i^*(k)$	: Signal fondamental.

$E$	: Tension d'alimentation contenue.
$T$	: Période d'échantillonnage.
$A_d$	: Matrices d'évolution discrétisés.
$B_d$	: Matrices de commande discrétisés.
$C_d$	: Matrices d'observation discrétisés.

## ***Introduction générale***

L'amélioration principale des processus industriels modernes au cours des 50 dernières années peut être en grande partie attribuée à l'évolution de l'électronique de puissance et à l'apparition des convertisseurs statiques et à la technologie des processeurs de traitement de signal « DSP ». Texas Instrument, après son premier né le TMS 32010 en 1982, a présentée différents membres de cette famille pour répondre aux besoins existants de traitement en temps réel. Ces *DSPs* conçus d'origine pour la téléphonie ont entourés, par la suite, une grande variété d'applications de traitement numérique de signal tel que le filtrage numérique, le traitement des signaux audio, le traitement d'images et de la vidéo et les applications de commande numérique des systèmes. Toutes ces applications partagent quelques caractéristiques communes, comme le besoin d'un temps de calcul assez réduit d'une part, d'autre part leurs algorithmes comprennent beaucoup de termes mathématiques complexes.

L'électrotechnique a beaucoup bénéficié des possibilités de la technologie *DSP* dans le domaine des entraînements par machine électrique en terme de commande ou de contrôle. Avant les années 50 la plupart des usines ont utilisées des moteurs à courant continu parce que les moteurs à induction triphasés pourraient seulement fonctionner à vitesse fixe. Maintenant la machine asynchrone à cage, grâce à la technologie avancée des alimentations électriques et l'utilisation des *DSPs* en plus les avantages présentés par cette machine en terme de robustesse, structure compacte et fiabilité, est d'une importance indoutable dans l'industrie moderne.

L'arrivée de la technologie des *DSPs* a relancé les théories d'estimation et d'identification des systèmes développées depuis des années en vue d'une application en ligne pour l'amélioration des performances des machines électriques. L'équipe de recherche du laboratoire LESIA « Les systèmes experts, l'imagerie et leur application dans l'ingénierie » s'intéresse à l'implantation des algorithmes d'estimation et d'identification des machines électriques en temps réel sur *DSP*.

Notre travail s'inscrit dans le cadre des activités de cette équipe de recherche. Nous avons choisi le développement des algorithmes d'estimation en temps réel des grandeurs non accessibles à la mesure direct de la machine asynchrone par le filtre de Kalman. La connaissance en ligne de ces paramètres est nécessaire pour la commande et le contrôle de la machine. Le filtre de Kalman s'adapte efficacement à l'implantation sur calculateur numérique vu sont formalisme récursif, en plus, ce filtre prend en charge toutes les sources d'erreurs et d'influence causées par les instruments et les capteurs. Comme processeur de traitement nous avons choisi Le TMS 320 LF 2407 dernier né de Texas Instrument conçu spécialement pour les applications de commande et de contrôle des machines électriques.

Dans ce travail, nous avons réalise un banc d'essai piloté par le DSP TMS320LF2407 EVM ( module de développement ) pour effectuer les testes expérimentaux de la tectonique d'estimation adoptée. Le système expérimental est valable pour le développement d'autres techniques d'estimation ou de contrôle.

Le présent mémoire est organisé en cinq chapitres :

Le premier chapitre est consacré à la présentation du formalisme du filtre de Kalman et il se termine par un exemple d'application qui montre l'efficacité du ce filtre.

Le second chapitre présente la modélisation dynamique de la machine asynchrone à cage. La transformation de Park est utilisée pour déterminer le modèle d'état. Les résultats de simulation d'un démarrage à vide de la machine sont présentés ainsi que les courbe d'estimation des flux rotoriques par le filtre de Kalman.

Le troisième chapitre expose la description du filtre de Kalman étendu et son application pour l'estimation des flux et la vitesse rotorique de la machine asynchrone.

Le quatrième chapitre développe le dispositif expérimental réalisé, son utilisation et installation. Ce chapitre se termine par des résultats de teste de validation du dispositif expérimental.

Le dernier chapitre traite l'implantation du filtre de Kalman sur de dispositif mise au point autour du DSP TMS320LF2407 pour une estimation en temps réel des flux rotoriques de la machine asynchrone.

## *Filtre de Kalman linéaire*

### **1.1. Introduction**

Un signal brut est généralement composé d'une partie utile ( signal utile ) et d'un bruit. Le filtrage est une description mathématique des différentes opérations de traitement que subit le signal mélangé au bruit, dans l'objectif d'estimer le signal utile [3]. Nous considérons dans notre cas des signaux déterministes noyés dans des bruits blancs avec l'hypothèse que les seules informations disponibles sur les signaux mis en jeux sont leurs paramètres statistiques.

Il y a deux types de filtres qui permettent d'extraire l'information utile ( déterministe ou aléatoire ) noyée dans des bruits :

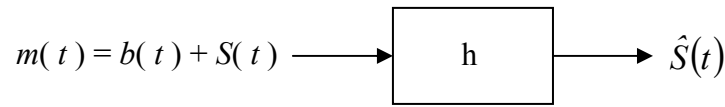
- Filtre de Wiener.
- Filtre de Kalman.

Dans le cas où la connaissance des informations a priori sur les signaux et les bruits nous fait défaut, on aura comme alternatif l'utilisation des filtres adaptatifs. Ces derniers apprennent les caractéristiques des signaux au fur et à mesure du déroulement du filtrage [3].

L'objectif de ce chapitre est de présenter l'approche, le formalisme du filtre de Kalman et de définir l'environnement où il peut être appliqué. Un exemple théorique de synthèse est développé permettant de voir la mise en œuvre de la procédure du filtrage de Kalman.

### **1.2. Approche de Kalman :**

Soit un signal  $m(t)$  observable qui contient une information utile traduit par le signal  $S(t)$  en addition avec un bruit  $b(t)$ . On désire avoir une méthode ou un filtre qui peut discriminer le signal du bruit. Le filtre Wiener permet de construire à partir du signal observé  $m(t)$ , une estimation  $\hat{S}(t)$  du signal recherché  $S(t)$ , figure (1.1).



**Figure ( 1.1 ) :** Représentation de filtre de Wiener.

L'estimation  $\hat{S}(t)$  proposée par Wiener du signal  $S(t)$  est donnée par :

$$\hat{S}(t) = \int_{t_0}^t h(t, \tau) m(t, \tau) d\tau \quad (1.1)$$

L'équation (1.1) est d'une manière générale difficile à résoudre dès que l'on aborde des cas non triviaux. Elle est en outre peut adaptée au traitement par ordinateur numérique à cause de son caractère non récursif. Il existe une autre approche ; celle du filtre de Kalman plus appropriée au traitement par ordinateur.

Kalman a introduit un filtre dès 1960 à partir de la représentation des systèmes par des équations différentielles matricielles [1][17]. Ce filtre est basé sur le fait qu'un processus aléatoire peut être modélisé comme étant la sortie d'un système linéaire ayant comme entrée un bruit blanc, alors que dans le cas de Wiener les systèmes sont représentés par des équations de covariance. L'approche de Kalman consiste à transformer l'équation intégrale (1.1), décrivant le filtre de Wiener dans le cas continu en équation différentielle récursive facile à implanter sur ordinateur numérique [2][3].

Etant donné le système linéaire stochastique dont l'évolution dynamique est modélisée à l'aide du système d'équations d'états stochastiques suivant :

$$\begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) + W(k+1) \\ y(k) = C(k)x(k) + V(k) \end{cases} \quad (1.2)$$

Où  $x(k) \in \mathfrak{R}^n$ ,  $u(k) \in \mathfrak{R}^m$ ,  $W(k) \in \mathfrak{R}^n$ ,  $y(k) \in \mathfrak{R}^p$  et  $V(k) \in \mathfrak{R}^p$ .

Le signal  $u(k)$  est le signal de commande supposé déterministe et connu, l'état initial  $x_0$  est supposé gaussien :

$$x_0 \sim N(\bar{x}, P_0) \quad (1.3)$$

Les bruits  $W(k)$  et  $V(k)$  sont supposés des bruits blancs, gaussiens, mutuellement indépendants et aussi indépendants de  $x_0$  [3][5]:

$$W \sim N(0, Q) \quad (1.4)$$

$$V \sim N(0, R) \quad (1.5)$$

Les bruits  $W(k)$  et  $V(k)$  ont des spécifications typiquement différentes ;  $W(k)$  est un bruit d'état qui exprime les phénomènes agissants sur le processus ainsi que les erreurs de modélisation, tandis que  $V(k)$  est un bruit de mesure qui exprime l'incertitude sur les instruments de mesure et aussi le bruit de quantification [2]. Les matrices  $A$ ,  $B$ ,  $C$ ,  $Q$  et  $R$  sont de dimensions convenables pour éviter toute confusion.

Le problème d'estimation au sens de Kalman consiste à estimer le vecteur d'état du système compte tenu des informations disponibles à l'instant  $n$ , postérieure, antérieure et identique à l'instant  $k$ . Nous considérons les trois cas suivants [1][2][3] :

- $k = n$  : Il s'agit dans ce cas de déterminer une estimation de l'état, compte tenu de toutes les mesures disponibles à l'instant  $n$ , c'est le cas du filtrage.
- $k < n$  : On ne prendra ici en considération qu'une partie des mesures disponibles, c'est le cas d'un lissage ou interpolation.
- $k > n$  : Il s'agit dans ce cas de prévoir une estimation du vecteur d'état en se basant sur des mesures disponibles jusqu'à l'instant  $n$  seulement, c'est le cas de prédiction.

Le filtre de Kalman peut résoudre facilement le problème de prédiction et de filtrage mais le problème de lissage nécessite un traitement plus complexe [4].

Le problème de filtrage au sens de Kalman est de trouver, pour un système dynamique à observer, la meilleure estimation notée par  $\hat{x}(k/n)$  de l'état à l'instant  $k$ , à partir des observations effectuées jusqu'à l'instant  $n$  au sens de critère de variance conditionnelle minimale.

### 1.3. Formulation et Equations du filtre de Kalman discret

L'estimation de l'état est définie par la valeur moyenne du vecteur d'état compte tenu des mesures  $\{y(1), y(2), \dots, y(n)\}$ . Cette estimation est exprimée par l'espérance mathématique conditionnelle notée par :

$$\hat{x}(k/n) = E[x(k) / y(1), y(2), \dots, y(n)] \quad (1.6)$$

L'évolution du vecteur d'état est caractérisée par la matrice de transition  $\Phi(k+1, k)$ . On peut admettre que l'évolution de l'estimation de l'état sera aussi caractérisée par la même matrice de transition, en d'autre terme nous aurons [3] :

$$\hat{x}(k/n) = \Phi(k, k-1)\hat{x}(k-1, k-1) \quad (1.7)$$

Cette relation peut être obtenue d'une manière rigoureuse en appliquant la définition donnée par l'équation (1.6) dans l'équation (1.7) soit :

$$\begin{aligned} E[x(k) / y(1), y(2), \dots, y(k-1)] &= \Phi(k, k-1) E[x(k) / y(1), y(2), \dots, y(k-1)] \\ &+ B(k)E[u(k) / y(1), y(2), \dots, y(k-1)] \end{aligned} \quad (1.8)$$

L'objectif du filtre de Kalman [3] est d'avoir une estimation récursive du vecteur d'état à partir d'une estimation à l'instant  $k$  avec la prise en compte des mesures disponibles à l'instant  $k+1$  pour donner une nouvelle estimation du vecteur d'état à cet instant. Kalman [1] a adopté une estimation linéaire d'état de la forme :

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1)[y(k+1) - H(k)\hat{x}(k, k)] \quad (1.9)$$

Où  $\hat{x}(k+1, k)$  est la prédiction de l'état estimée  $x$  à partir de  $\hat{x}(k, k)$  pour l'instant  $k+1$ .  $K(k+1)$  est appelé le gain de filtre de Kalman. Il est déterminé de sorte que l'erreur entre l'état réel et l'état estimé soit minimale au sens des moindres carrés, soit :

$$\tilde{x}(k) = x(k) - \hat{x}(k, k) \quad (1.10)$$

$P(k, k)$  est la matrice de covariance de l'erreur d'estimation donnée par :

$$\begin{aligned} P(k, k) &= E[(x(k) - \hat{x}(k, k))(x(k) - \hat{x}(k, k))^T] \\ &= E[\tilde{x}(k)\tilde{x}^T(k)] \end{aligned} \quad (1.11)$$



La minimisation de  $P(k, k)$  exprime l'optimalité de l'estimateur, on dit que le filtre de Kalman est un filtre à minimum de variance d'erreur d'estimation [3][6].

### 1.3.1 Expression de $P(k, k)$

Nous allons à partir des équations d'état et des équations d'estimation exprimer la matrice  $P(k, k)$  en fonction du gain  $K(k+1)$  :

Reprenons les relations :

$$x(k) = \Phi(k, k-1)x(k-1) + B(k-1)u(k-1) \quad (1.12)$$

$$\hat{x}(k, k-1) = \Phi(k, k-1)\hat{x}(k-1, k-1) \quad (1.13)$$

En les soustrayant, nous obtenons :

$$\begin{aligned} x(k) - \hat{x}(k, k-1) &= \Phi(k, k-1)[x(k-1) - \hat{x}(k-1, k-1)] \\ &\quad + B(k-1)u(k-1) \end{aligned} \quad (1.14)$$

Par définition nous avons :

$$P(k, k-1) = E[(x(k-1) - \hat{x}(k, k-1))(x(k-1) - \hat{x}(k, k-1))^T] \quad (1.15)$$

En introduisant l'espérance mathématique sur la relation (1.14) elle devient :

$$P(k, k-1) = \Phi(k, k-1)E[(x(k-1) - \hat{x}(k-1, k-1))(x(k-1) - \hat{x}(k-1, k-1))^T] \quad (1.16)$$

On a aussi :

$$E[(x(k-1) - \hat{x}(k-1, k-1))(x(k-1) - \hat{x}(k-1, k-1))^T] = P(k-1, k-1) \quad (1.17)$$

Donc :

$$P(k, k-1) = \Phi(k, k-1)P(k-1, k-1)\Phi^T(k, k-1) + B(k-1)QB^T(k-1) \quad (1.18)$$

En d'autre terme :

$$\tilde{x}(k) = x(k) - \hat{x}(k, k) \quad (1.19)$$

Et :

$$\hat{x}(k, k) = \hat{x}(k, k-1) + K(k)[y(k) - H(k)\hat{x}(k, k-1)] \quad (1.20)$$

Soit :

$$\begin{aligned}\tilde{x}(k, k) &= x(k) - \hat{x}(k, k-1) - K(k)[H(k)x(k) + V(k) - H(k)\hat{x}(k, k-1)] \\ &= (I - K(k))(x(k) - \hat{x}(k, k-1)) - K(k)V(k)\end{aligned}\quad (1.21)$$

Avec :

$$P(k, k) = E[\tilde{x}(k)\tilde{x}^T(k)] \quad (1.22)$$

En substituant l'expression (1.19) dans la relation (1.20), on obtient l'expression de  $P(k, k)$  en fonction de  $P(k, k-1)$  et de  $K(k)$  où :

$$P(k, k) = (I - K(k)H(k))P(k, k-1)(I - K(k)H(k))^T + K(k)R(k)K^T(k) \quad (1.23)$$

Ou bien :

$$\begin{aligned}P(k+1, k+1) &= (I - K(k+1)H(k))P(k+1, k)(I - K(k+1)H(k))^T \\ &\quad + K(k)R(k)K^T(k)\end{aligned}\quad (1.24)$$

### 1.3.2. Détermination du gain $K(k)$

Le gain du filtre de kalman est calculé dans le sens de minimiser la covariance de l'erreur d'estimation  $P(k, k)$  [3], ou bien la minimisation du critère  $J$  où :

$$J = [\tilde{x}^T(k)\tilde{x}(k)] = \text{trace}(P(k, k)) \quad (1.25)$$

On peut obtenir l'expression de  $K(k)$  en résolvant l'équation suivante :

$$\frac{\partial J(k)}{\partial K(k)} = 0 \quad (1.26)$$

En tenant compte de la relation  $\frac{\partial}{\partial K(k)}(\text{trace} KAK) = 2KA$  on aura :

$$-[I - K(k)H(k)]P(k, k-1)H^T(k) + K(k)K(k) = 0 \quad (1.27)$$

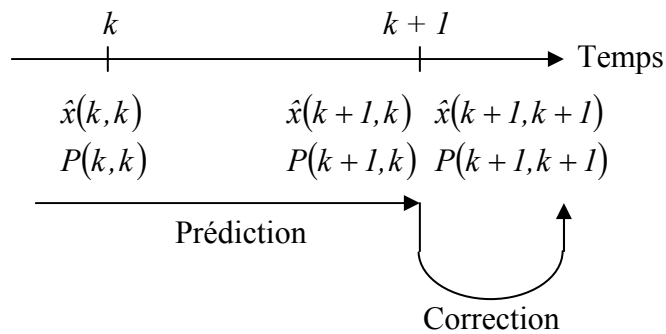
La solution de l'équation matricielle ci-dessus est donnée par :

$$K(k) = P(k, k-1)H^T(k)[H(k)P(k, k-1)H^T(k) + R(k)]^{-1} \quad (1.28)$$

Cette expression est la forme de gain du filtre de Kalman qui minimise la covariance de l'erreur d'estimation.

#### 1.4. Mise en œuvre du filtre de Kalman

L'estimation récurrente du filtre de Kalman peut se décomposer en deux étapes dites étape de prédiction et étape de correction [6], figure (1,2).



**Figure ( 1.2 )** : Etapes de fonctionnement du filtre de Kalman.

L'étape de prédiction correspond au calcul de l'estimation à priori (la prédiction  $\hat{x}(k+1, k)$ ) et de sa covariance d'erreur à priori  $P(k+1, k)$  par les équations suivantes :

##### a - Equation de prédiction de l'état :

$$\hat{x}(k+1, k) = A\hat{x}(k, k) + Bu(k) \quad (1.29)$$

##### b - Equation de prédiction de la covariance d'erreur à priori :

$$P(k+1, k) = AP(k, k)A^T + BQB^T \quad (1.30)$$

L'étape de correction permet la détermination de l'estimation a posteriori et de sa covariance d'erreur après le calcul du gain du filtre de Kalman selon les équations suivantes :

##### c - Calcul du gain de filtre de Kalman

$$K(k+1) = P(k+1, k)H^T(k) [H(k)P(k+1, k)H^T(k) + R]^{-1} \quad (1.31)$$

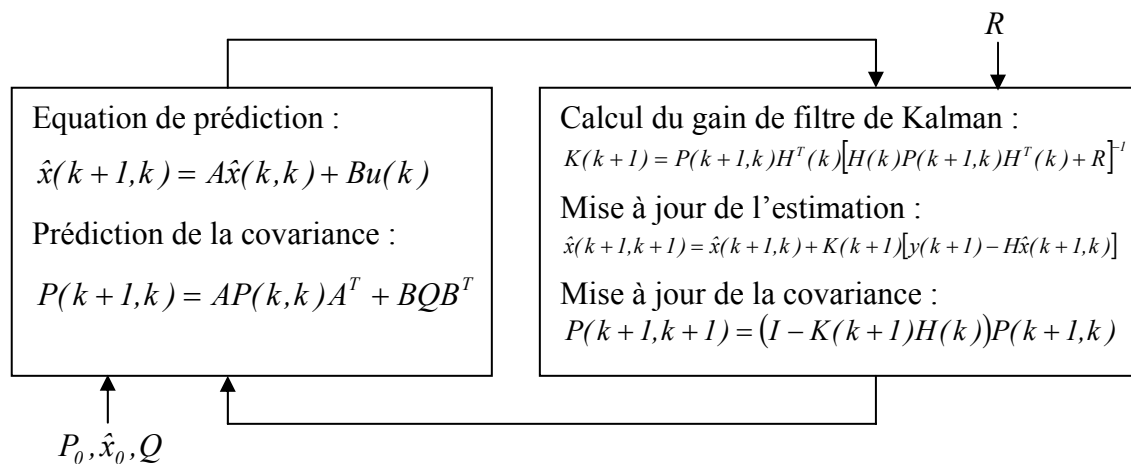
**d - Mise à jour de l'estimation avec les mesures  $y(k+1)$** 

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1)[y(k+1) - H(k)\hat{x}(k+1, k)] \quad (1.32)$$

**e - Mise à jour de la covariance de l'erreur**

$$P(k+1, k+1) = (I - K(k+1)H(k))P(k+1, k) \quad (1.33)$$

La figure (1.3) illustre le fonctionnement du processus itératif de ce filtre :

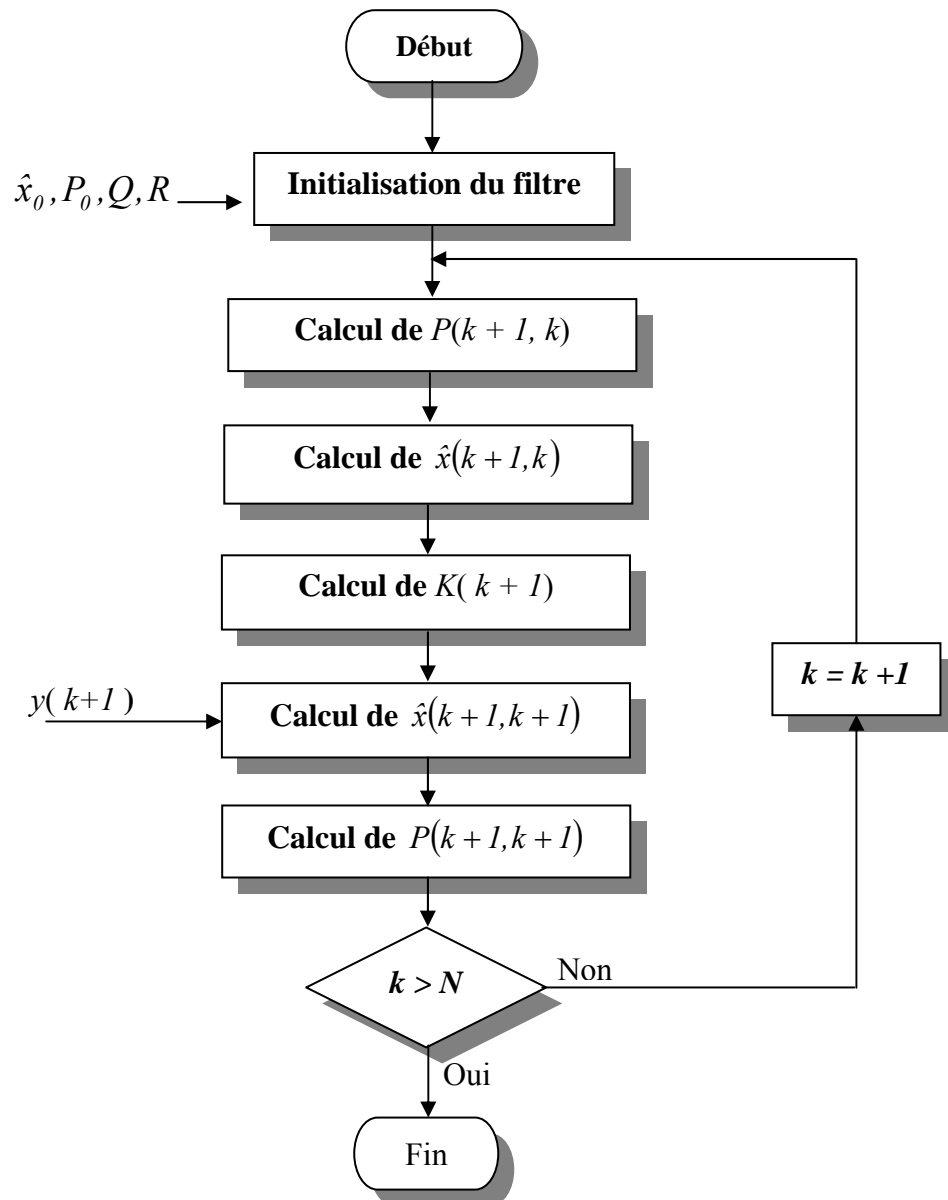


**Figure ( 1.3 )** : Processus itératif du filtre de Kalman linéaire.

### 1.5. Algorithme du filtre de Kalman

Lors de l'implantation de l'algorithme du filtre de Kalman, on doit commencer par les équations de prédiction. La première itération est caractérisée par l'initialisation d'un certain nombre de valeurs, tels que la covariance de l'erreur  $P_0$  et la valeur initiale  $x_0$  de l'état estimé, ainsi que les matrices de covariance des bruits d'état et de mesure, puis on passe aux étapes de prédiction et de correction.

L'organigramme de la figure ( 1.4 ) nous montre la succession des équations de filtre de Kalman avec l'initialisation et l'introduction des mesures :



**Figure ( 1.4 )** : Organigramme du filtre de Kalman linéaire.

### 1.6. Choix des valeurs initiales

Le choix des valeurs initiales du filtre de Kalman est délicat [3], un mauvais choix de  $\hat{x}_0$  n'est pas catastrophique au sens que l'algorithme apportera toutes les corrections nécessaires. Par contre le traitement des mesures n'améliore pas la covariance de l'erreur au fur et à mesure de son traitement, nous pouvons traduire notre ignorance de  $P_0$  en adaptant une valeur élevée.

Le choix des valeurs des matrices  $Q$  et  $R$  est une des principales difficultés de la mise en œuvre du filtre de Kalman. Elle réside dans la résolution de l'équation de Riccati donnant les matrices de covariance. Ce travail implique de nombreuses opérations arithmétiques qui peuvent entraîner la non positivité ou la non symétrie de ces matrices. Un choix arbitraire de ces conditions initiales donne souvent des résultats insuffisants ou conduit à la divergence du filtre [4].

### 1.7. Exemple d'application du filtre de Kalman

Considérons la situation suivante, on a un signal  $x(k)$  et un bruit  $b(k)$ , seule la somme  $m(k) = x(k) + b(k)$  peut être observé, nous essayons d'obtenir  $\hat{x}(k)$  ; l'estimation du signal  $x(k)$  en utilisant le filtre de Kalman.

#### 1.7.1. Modèle du système

Notre système est régi par le système d'équations d'état suivant :

$$\begin{aligned}x(k+1) &= A(k)x(k) + B(k)u(k) + W(k+1) \\ y(k) &= C(k)x(k)\end{aligned}\tag{1.34}$$

Dans ce système, on a pas d'entrée de commande, donc  $u(k) = 0$  ; le système devient alors :

$$\begin{aligned}x(k+1) &= x(k) + W(k+1) \\ y(k) &= x(k) + V(k)\end{aligned}\tag{1.35}$$

#### 1.7.2. Equation et paramètres du filtre

Equations de prédiction :

$$\hat{x}(k+1, k) = \hat{x}(k, k)\tag{1.36}$$

$$P(k+1, k) = P(k, k) + Q\tag{1.37}$$

Equations de correction :

$$K(k+1) = P(k+1, k)(P(k+1, k) + R)^{-1}\tag{1.38}$$

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1)(y(k+1) - \hat{x}(k+1, k))\tag{1.39}$$

$$P(k+1, k+1) = (I - K(k+1))P(k+1, k) \tag{1.40}$$

### 1.7.3. Simulation du système sous MATLAB

Dans cet exemple, nous utilisons un signal sinusoïdal bruité avec un bruit gaussien de moyenne nulle et de variance constante connue. La figure (1.5) donne un schéma en bloc de la simulation du système associé au filtre de Kalman :

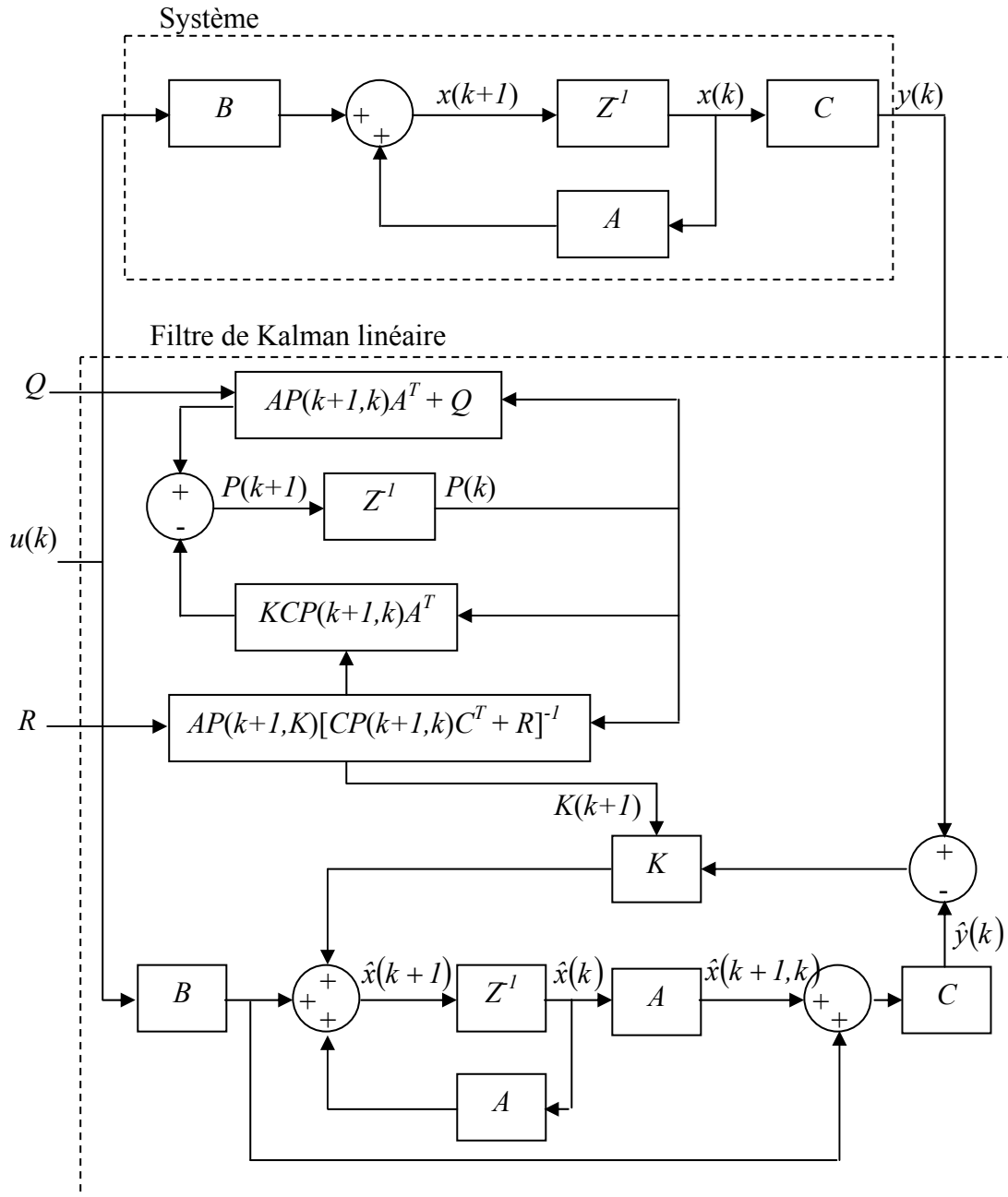
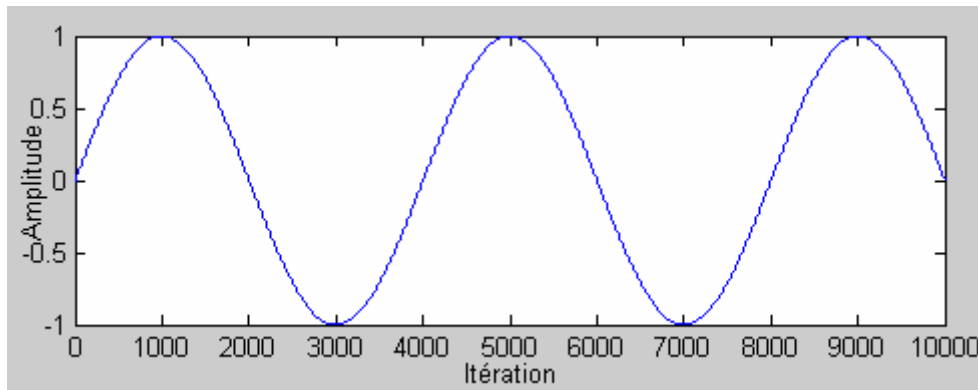


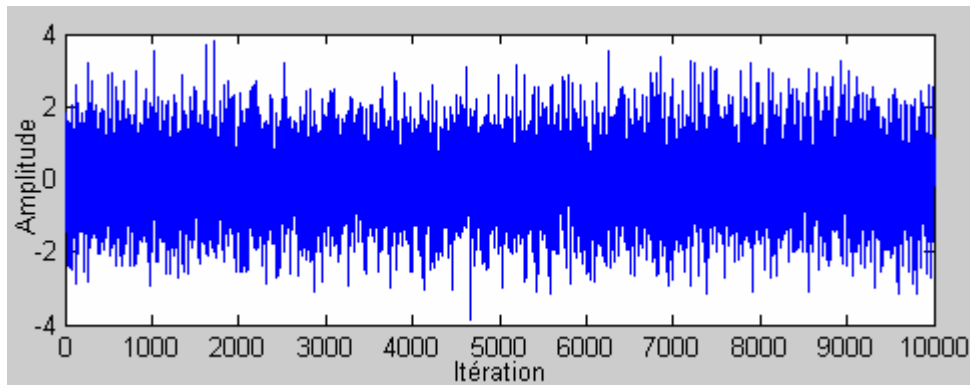
Figure ( 1.5 ) : Schéma en bloc d'ensemble système + filtre de Kalman.

#### 1.7.4. Signal d'entrée du filtre de Kalman

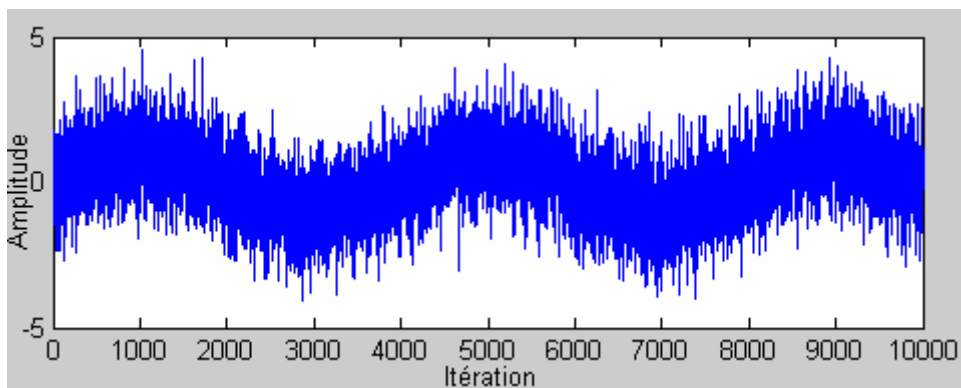
Nous avons choisi comme signal d'entrée du système, un signal sinusoïdal figure (1.6) de fréquence égale à  $1\text{ Hz}$  et d'amplitude unitaire. Le signal mesuré  $m(k)$ , figure (1.8) est donc un signal sinusoïdal bruité par un bruit gaussien de moyenne nul et de variance unitaire (bruit blanc), figure (1.7).



**Figure (1.6) :** Signal d'entrée original (sinusoïdal).



**Figure (1.7) :** Bruit blanc injecté.

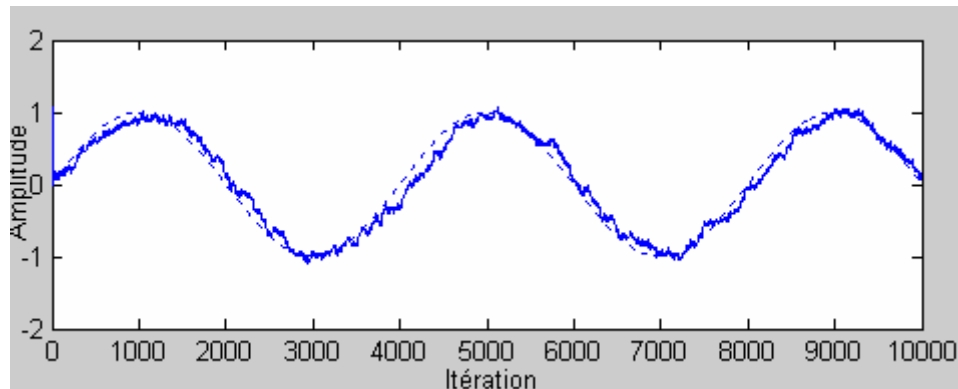


**Figure (1.8) :** Signal d'entrée du filtre de Kalman  
(Signal mesuré).

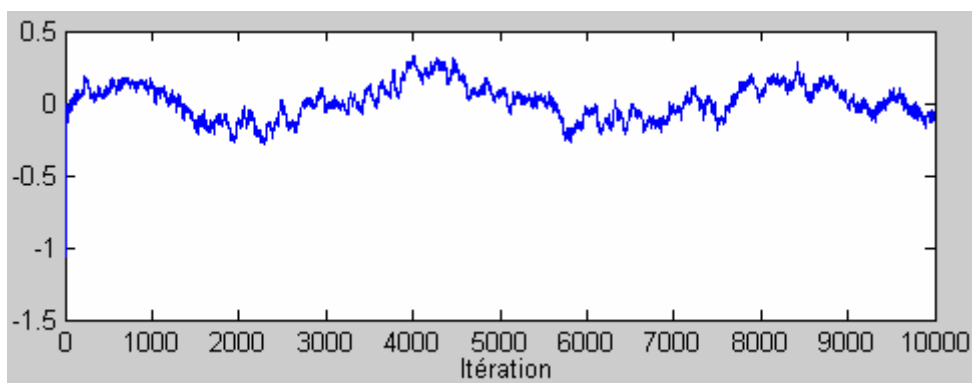


### 1.7.5. Résultats de simulation

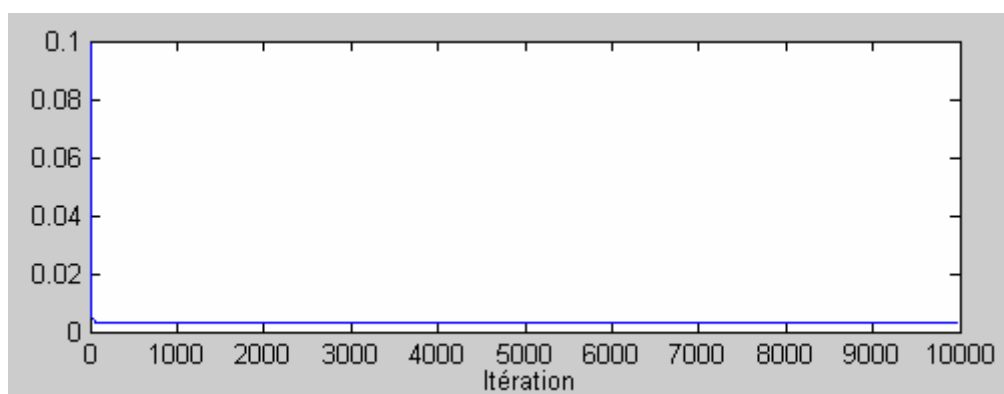
Les résultats obtenus de la simulation de cet exemple sont le signal estimé représenté avec le signal original non bruité, figure (1.9), l'erreur d'estimation correspond à la différence entre le signal estime le signal original, figure (1.10) et en fin la covariance de l'erreur d'estimation pour montrer que le filtre de Kalman fonctionne dans le bon sens, figure (1.11).



**Figure (1.9) :** Signal estimé (-) avec le signal original (..).



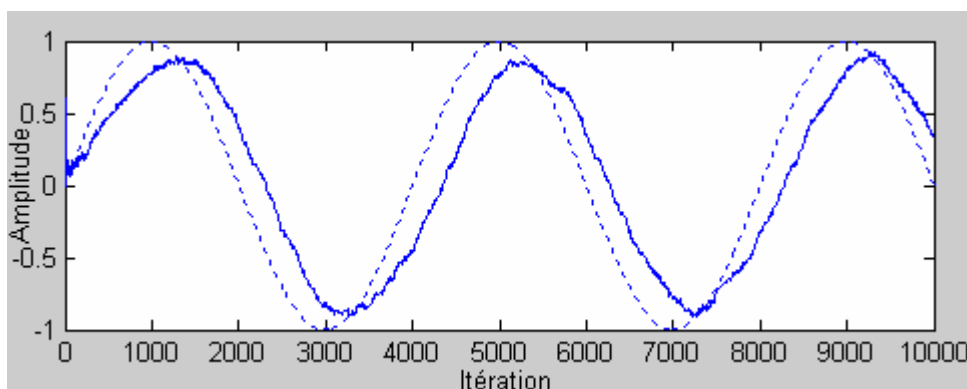
**Figure (1.10) :** Erreur d'estimation.



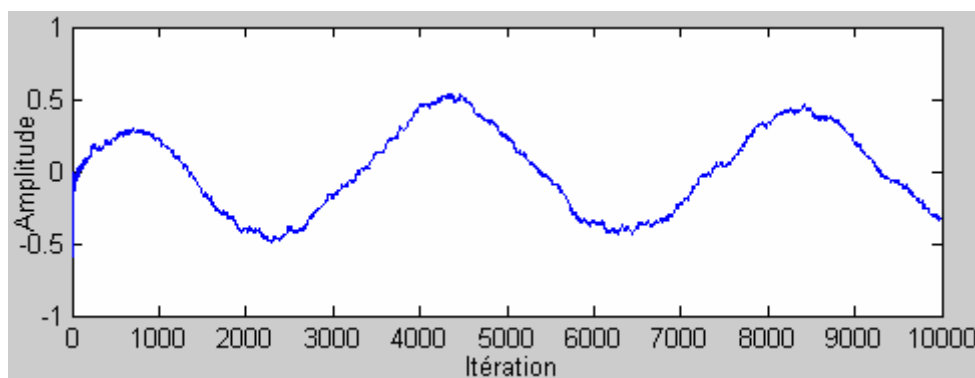
**Figure (1.11) :** Covariance de l'erreur d'estimation.

Pour voir le comportement du filtre de Kalman dans les cas où les paramètres statistiques du bruit ne sont pas parfaitement connus, nous avons injecté le même signal bruité au filtre de Kalman pour deux valeurs de covariances différentes de la valeur exacte de covariance du bruit généré. Nous présentons comme résultats le signal estimé son erreur d'estimation pour chaque cas.

1<sup>ère</sup> Cas pour  $R = V^2 = 10$  :

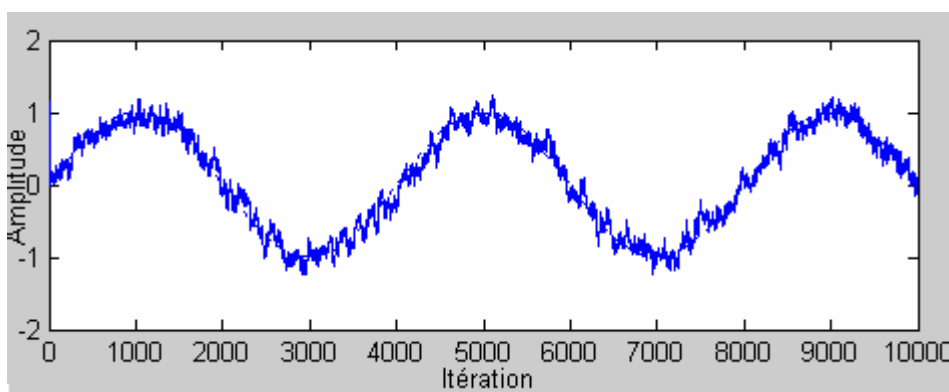


**Figure (1.12) :** Signal estimé (-) avec le signal original (..) pour  $R = 10$ .

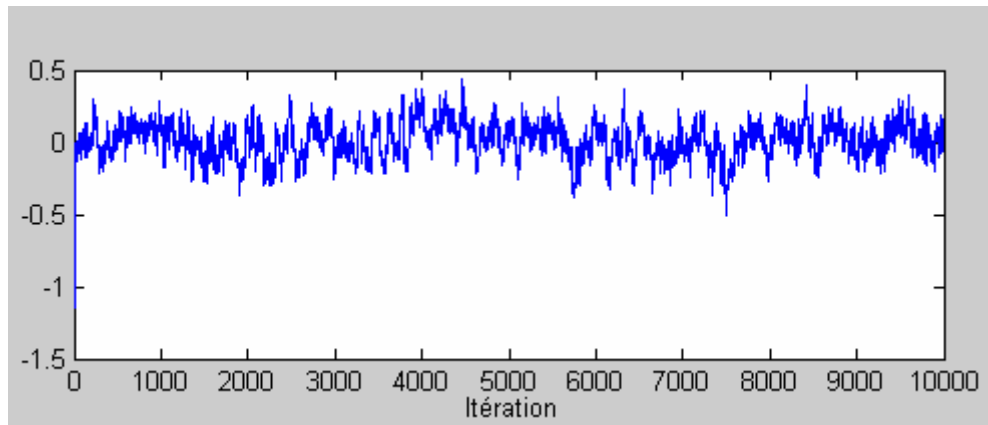


**Figure (1.13) :** Erreur d'estimation pour  $R = 10$ .

2<sup>ème</sup> Cas pour  $R = V^2 = 0.1$



**Figure (1.14) :** Signal estimé (-) avec Le signal original (..) pour  $R = 0.1$ .



**Figure (1.15) :** Erreur d'estimation pour  $R = 0.1$ .

### 1.8. Conclusion

Dans ce chapitre nous avons exposé le formalisme du filtre de Kalman linéaire. Cela a permis de voir que ce filtre s'adapte bien à une implantation sur ordinateur numérique vu son caractère récursif. Le comportement stochastique du filtre permet d'accepter ou d'autoriser une marge d'erreur sur le modèle du système et sur ces grandeurs mesurables.

L'exemple de simulation que nous avons développé sous MATLAB a montré que le filtre de Kalman est capable de réduire l'effet des grandeurs d'influences sur le système à étudier, néanmoins ce filtre demande une bonne connaissance des caractéristiques stochastiques de ces grandeurs. Cela nécessite une étude préalable du système avant toute implantation.

## *Estimation des flux rotoriques de la machine asynchrone à cage par le filtre de Kalman*

### **2.1. Introduction**

Le filtre de Kalman linéaire est un estimateur de vecteur d'état d'un système linéaire. Il permet de donner une estimation optimale des grandeurs non accessibles directement à la mesure [8]. Le filtre de Kalman, sous sa forme récurrente, peut être implanté sur ordinateur numérique pour un fonctionnement en ligne (temps réel) dans l'objectif de déterminer des paramètres nécessaires à la commande du système, ou hors ligne dans le but de simuler et étudier le comportement des paramètres liés au système.

L'application du filtre de Kalman sur un système linéaire passe par une modélisation en équations d'état. L'incertitude sur la modélisation du système et sur l'exactitude des mesures est compensée par l'addition de grandeurs stochastiques (bruits) aux variables du système. La mise en œuvre du filtre de Kalman nécessite des connaissances préalables sur les caractéristiques statistiques de ces bruits pour aboutir à des résultats optimaux.

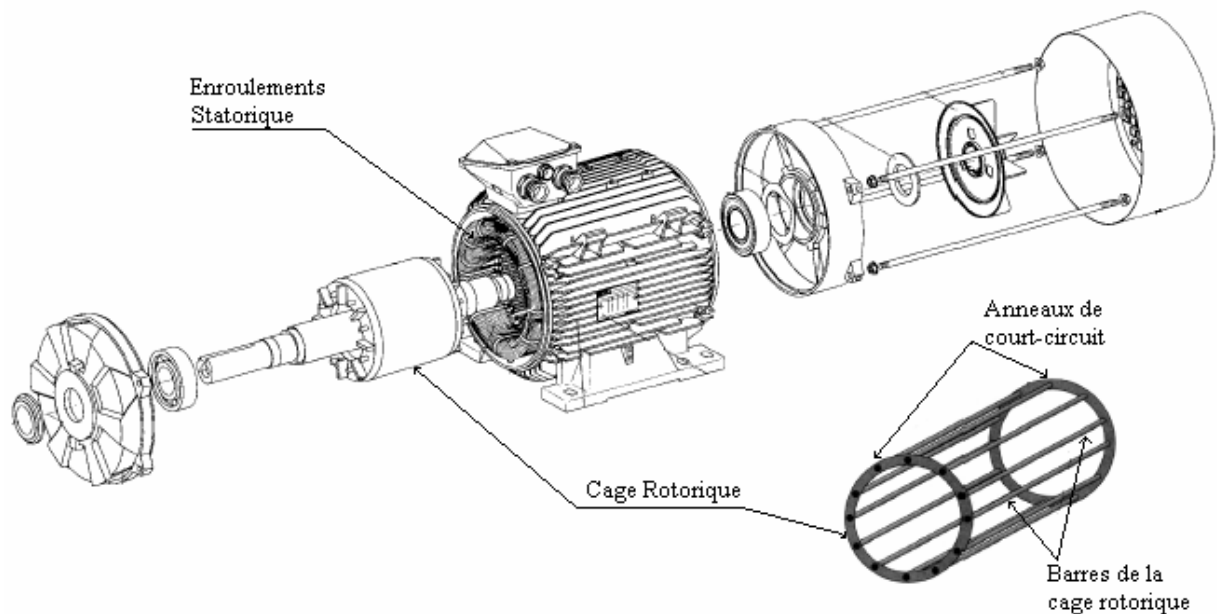
En industrie, dans le domaine des entraînements à vitesse variable, la machine asynchrone est largement utilisée pour des raisons de coût, taille, robustesse et fiabilité [5][7]. Le filtre de Kalman est utilisé pour l'estimation des flux rotoriques, des courants rotoriques ou du couple électromagnétique ; grandeurs non accessibles directement à la mesure. Ces grandeurs seront exploitées par les algorithmes de commande qui sont devenus complexes et nécessitent de la rapidité de calcul. Nous citons comme exemple d'algorithme la commande vectorielle qui exige la connaissance temporelle de la variation des flux rotoriques [8][9].

L'objectif de ce chapitre est de formuler le modèle de la machine asynchrone en équations d'état discrètes et d'appliquer le filtre de Kalman pour estimer, par simulation, les flux rotoriques de la machine asynchrone fonctionnant en boucle ouverte.

## 2.2. Modélisation de la machine asynchrone à cage

### 2.2.1 Description du moteur asynchrone à cage

Un moteur asynchrone à cage se présente sous la forme d'un carter enroulant les circuits magnétiques et ferromagnétiques statoriques dans des encoches d'enroulement triphasé bobiné de cuivre isolé. A l'intérieur de ce circuit magnétique, se présente le circuit magnétique rotorique qui accueille dans ces encoches, les barreaux de la cage rotorique en Aluminium ou Cuivre, ces barreaux sont court-circuités à chaque extrémité par des anneaux du même matériaux [11], figure (2.1).



**Figure (2.1) :** Structure de la machine asynchrone à cage.

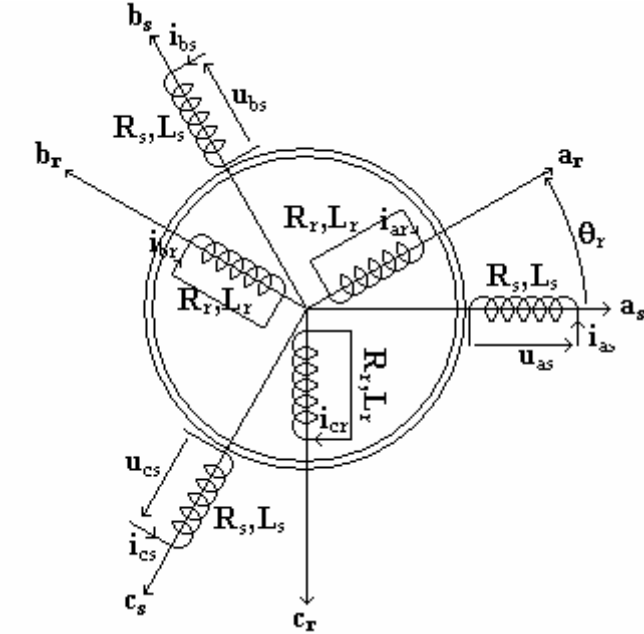
### 2.2.2. Hypothèses de simplification

La modélisation de la machine asynchrone s'appuie sur un certain nombre d'hypothèses simplificatrices :

- Entrefer constant.
- Effet des encoches négligé.
- Distribution spatiale des champs magnétiques le long de l'entrefer.
- Circuit magnétique non saturé et à perméabilité constante.
- Pertes ferromagnétiques négligeables.

**2.2.3. Modèle de la machine asynchrone dans le plan  $a, b, c$**

Electriquement la machine asynchrone à cage se présente comme l'indique la figure (2.2).



**Figure ( 2.2 )** : Représentation de la machine asynchrone à cage.

Les grandeurs et les paramètres régissant le comportement de la machine sont :

$R_s, R_r$  : Résistance statorique et rotorique.

$L_s, L_r$  : Inductance statorique et rotorique.

**2.2.3.1. Equations électriques**

Les équations de tensions des trois phases statoriques ( $a, b, c$ ) et rotoriques ( $A, B, C$ ) S'écrivent sous forme matricielle :

$$[u_s] = [R_s][i_s] + \frac{d}{dt}[\psi_s] \tag{2.1}$$

$$[u_r] = [R_r][i_r] + \frac{d}{dt}[\psi_r] \tag{2.2}$$

Avec :

$$[u_s] = [u_a \ u_b \ u_c]^T, \quad [u_r] = [u_A \ u_B \ u_C]^T, \quad [i_s] = [i_a \ i_b \ i_c]^T$$

Et :

$$[i_r] = [i_A \ i_B \ i_C]^T, \quad [\psi_s] = [\psi_a \ \psi_b \ \psi_c]^T, \quad [\psi_r] = [\psi_A \ \psi_B \ \psi_C]^T$$

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

Où :

$[u_s], [u_r]$  : Vecteurs des tensions statoriques et rotoriques.

$[i_s], [i_r]$  : Vecteurs des courants statorique et rotoriques.

$[\psi_s], [\psi_r]$  : Vecteurs des flux statoriques et rotoriques.

$[R_s], [R_r]$  : Matrices des résistances statoriques et rotoriques.

### 2.2.3.2. Equations magnétiques

Les flux qui traversent les surfaces statoriques et rotoriques s'expriment sous forme :

$$[\psi_s] = [L_s][i_s] + [M_{sr}][i_r] \quad (2.3)$$

$$[\psi_r] = [L_r][i_r] + [M_{rs}][i_s] \quad (2.4)$$

Les matrices d'inductance statorique, et rotorique sont données par :

$$[L_s] = \begin{bmatrix} L_s & M_s & M_s \\ M_s & L_s & M_s \\ M_s & M_s & L_s \end{bmatrix}, \quad [L_r] = \begin{bmatrix} L_r & M_r & M_r \\ M_r & L_r & M_r \\ M_r & M_r & L_r \end{bmatrix}$$

$L_s, L_r$  : Inductances propres statoriques et rotoriques.

$M_s$  : Inductance mutuelle entre phases statoriques.

$M_r$  : Inductance mutuelle entre phases rotoriques.

La matrice des inductances mutuelles entre phases du stator et du rotor dépend de la position angulaire  $\theta$  entre l'axe du stator et celui du rotor [12][13] et s'écrit :

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

$M_{sr}$  : Maximum de l'inductance mutuelle entre une phase du stator et les phases correspondantes du rotor (leurs axes magnétiques sont alors alignés).

### 2.2.3.3. Equation mécanique

Le comportement mécanique de la machine asynchrone est régi par l'équation suivante :

$$Ce - Cr = J \frac{d\Omega_r}{dt} + f_r \Omega_r \quad (2.5)$$

Où :

- $Cr$  : Le couple résistant de la charge.
- $J$  : Le moment d'inertie total de l'arbre de moteur.
- $f_r$  : Le coefficient de frottement visqueux.
- $\Omega_r = \frac{\omega_r}{p}$  : La vitesse de rotation mécanique.
- $p$  : Le nombre de paires de pôles.

Le couple électromagnétique développé par le champ tournant est exprimé par la relation :

$$Ce = \frac{3}{2} p \frac{M}{L_r} (\psi_{rd} i_{sq} - \psi_{rq} i_{sd}) \quad (2.6)$$

### 2.2.3.4. Transformation de PARK

La transformation de PARK [5][13] consiste à appliquer aux courants, tensions et flux, un changement de variables faisant intervenir l'angle entre l'axe des enroulements et les axes  $d$



et  $q$ . Ceci peut être interprété comme la substitution des enroulements réels avec des d'enroulements fictifs  $d_s, q_s, d_r$  et  $q_r$  dont les axes magnétiques sont liés aux axes  $d$  et  $q$  conformément à la figure (2.3).

On transforme donc l'enroulement triphasé ( $a, b, c$ ) en trois enroulements orthogonaux ( $d, q, O$ ), Avec :

- $d$  : Axe direct.
- $q$  : Axe en quadrature.
- $O$  : Axe homopolaire.

La matrice de transformation de PARK est définie comme suit :

$$[A] = \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{2}\right) & \cos\left(\theta + \frac{2\pi}{2}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{2}\right) & -\sin\left(\theta + \frac{2\pi}{2}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.7)$$

La transformation inverse est donnée par :

$$[A]^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos\left(\theta - \frac{2\pi}{2}\right) & -\sin\left(\theta - \frac{2\pi}{2}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{2}\right) & -\sin\left(\theta + \frac{2\pi}{2}\right) & 1 \end{bmatrix} \quad (2.8)$$

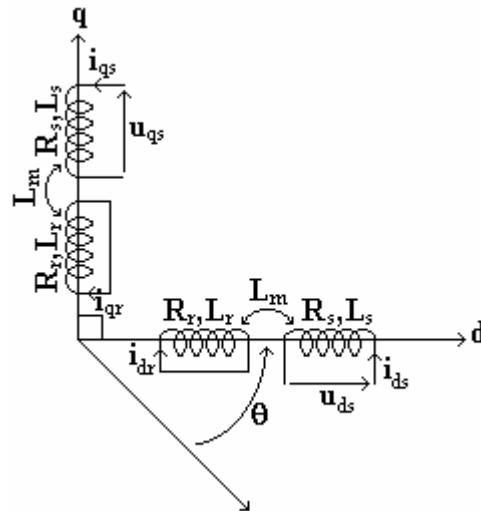
Pour le stator :

$$\begin{bmatrix} x_d \\ x_q \\ x_o \end{bmatrix} = [A(\theta = \theta_s)] \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (2.9)$$

Pour le rotor :

$$\begin{bmatrix} x_d \\ x_q \\ x_o \end{bmatrix} = [A(\theta = \theta_r)] \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix} \quad (2.10)$$

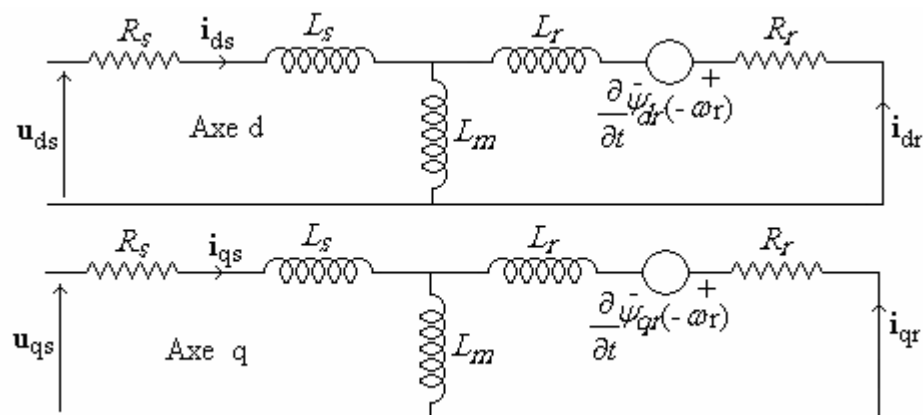
$x$  : Correspond aux courant, tension ou flux.



**Figure ( 2.3 ) :** Représentation physique de la transformation de Park pour une machine asynchrone.

### 2.2.4. Modèle de la machine asynchrone dans le repère de Park

Le circuit équivalent de la machine asynchrone dans le repère Park (  $d, q$  ) [5][10] est représenté sur la figure (2.4) :



**Figure ( 2.4 ) :** Schéma équivalent de la machine asynchrone à cage dans le plan de Park.

D'après le circuit équivalent de la machine asynchrone, figure (2.4), les équations électriques et magnétiques dans le repère  $(d, q)$  s'écrivent :

- Equations électriques du stator et rotor

$$\begin{aligned}
 u_{sd} &= R_s i_{sd} + \frac{d\psi_{sd}}{dt} - \omega_a \psi_{sq} \\
 u_{sq} &= R_s i_{sq} + \frac{d\psi_{sq}}{dt} + \omega_a \psi_{sd} \\
 u_{rd} &= 0 = R_r i_{rd} + \frac{d\psi_{rd}}{dt} - (\omega_a - \omega_r) \psi_{rq} \\
 u_{rq} &= 0 = R_r i_{rq} + \frac{d\psi_{rq}}{dt} + (\omega_a - \omega_r) \psi_{rd}
 \end{aligned} \tag{2.11}$$

- Equations magnétiques du stator et rotor

$$\begin{aligned}
 \psi_{sd} &= L_s i_{sd} + M i_{rd} \\
 \psi_{sq} &= L_s i_{sq} + M i_{rq} \\
 \psi_{rd} &= L_r i_{rd} + M i_{sd} \\
 \psi_{rq} &= L_r i_{rq} + M i_{sq}
 \end{aligned} \tag{2.12}$$

Avec :

$$\begin{aligned}
 L_s &= L_s - M_s \\
 L_r &= L_r - M_r \\
 M &= \frac{3}{2} M_{sr}
 \end{aligned}$$

Le couple électromagnétique développé par les champs tournants est exprimé par la relation :

$$Ce = \frac{3}{2} p \frac{M}{L_r} (\psi_{rd} i_{sq} - \psi_{rq} i_{sd}) \tag{2.13}$$

Afin de concevoir le modèle complet, on ajoute au système d'équations électriques (2.11), l'équation mécanique suivante :

$$Ce - Cr = J \frac{d\Omega_r}{dt} + f_r \Omega_r \tag{2.14}$$

Par substitution des équations (2.12) dans les équations (2.11), on obtient :

$$\begin{bmatrix} V_{ds} \\ V_{qs} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + pL_s & 0 & pL_m & 0 \\ 0 & R_s + pL_s & 0 & pL_m \\ pL_m & \omega_r L_m & R_r + pL_r & \omega_r L_r \\ -\omega_r L_m & pL_m & -\omega_r L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \quad (2.15)$$

Où :

$$p : \text{Est un opérateur différentiel défini par : } p = \frac{d}{dt}$$

On peut aussi représenter le système des équations différentielles comme suit :

$$\begin{aligned} L_s \frac{d}{dt} i_{ds} + L_m \frac{d}{dt} i_{dr} &= V_{ds} - R_s i_{ds} \\ L_s \frac{d}{dt} i_{qs} + L_m \frac{d}{dt} i_{qr} &= V_{qs} - R_s i_{qs} \\ L_r \frac{d}{dt} i_{ds} + L_m \frac{d}{dt} i_{dr} &= -R_r i_{dr} - \omega_r (L_r i_{qr} + L_m i_{qs}) \\ L_r \frac{d}{dt} i_{qr} + L_m \frac{d}{dt} i_{qs} &= -R_r i_{qr} - \omega_r (L_r i_{dr} + L_m i_{ds}) \end{aligned} \quad (2.16)$$

En se basant sur le système d'équations (2.16), il est possible de donner plusieurs représentations du modèle de la machine sous forme d'équations d'état selon le vecteur d'état choisi pour l'application envisagée.

### 2.2.5. Modèle d'état de la machine asynchrone

L'estimation des flux rotoriques de la machine asynchrone par le filtre de Kalman nécessite la formulation d'un modèle adapté au vecteur d'état choisi pour l'application envisagée.

Le vecteur d'état le plus adapté à l'estimation des flux rotoriques de la machine est :

$$x = \begin{bmatrix} i_{ds} & i_{qs} & \psi_{dr} & \psi_{qr} \end{bmatrix}^T$$

Avec :

- Les grandeurs d'entrée :  $u = \begin{bmatrix} V_{ds} & V_{qs} \end{bmatrix}^T$ .
- Les grandeurs mesurées :  $y = \begin{bmatrix} i_{ds} & i_{qs} \end{bmatrix}^T$ .

Le modèle d'état de la machine asynchrone sera de la forme :

$$\begin{aligned}\dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t)\end{aligned}\quad (2.17)$$

Où :

$$x = [i_{ds} \quad i_{qs} \quad \psi_{dr} \quad \psi_{qr}]^T$$

$$u = [V_{ds} \quad V_{qs}]^T$$

$$A = \begin{bmatrix} -\frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} & 0 & \frac{L_m L_r}{\sigma L_s L_r^2} & \frac{L_m \omega_r}{\sigma L_s L_r} \\ 0 & -\frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} & -\frac{L_m \omega_r}{\sigma L_s L_r} & \frac{L_m L_r}{\sigma L_s L_r^2} \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & -\omega_r \\ 0 & \frac{L_m R_r}{L_r} & \omega_r & -\frac{R_r}{L_r} \end{bmatrix}\quad (2.18)$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\quad (2.19)$$

Avec :

$$\sigma = 1 - \frac{M^2}{L_s L_r} : \text{Le coefficient de dispersion [12].}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\quad (2.19)$$

### 2.3. Simulation d'un démarrage à vide de la machine asynchrone alimentée par réseau électrique triphasé

La simulation d'un démarrage de la machine peut être étudié à partir des équations du modèle d'état ou du système d'équations (2.16). Nous avons exploité le logiciel SIMULINK de MATWORK destiné à la simulation des systèmes dynamiques pour simuler un démarrage à vide d'une machine asynchrone. Ce logiciel est basé sur une représentation graphique des équations du modèle du système [16].

#### 2.3.1. Schéma en bloc de simulation

Le schéma en bloc de simulation d'un démarrage de la machine asynchrone peut se divisé en deux sous blocs, figure (2.5) ; le sous bloc d'alimentation en tension triphasé incluant la transformation de Park, figure (2.6) et le sous bloc du modèle de la machine, figure (2.7).

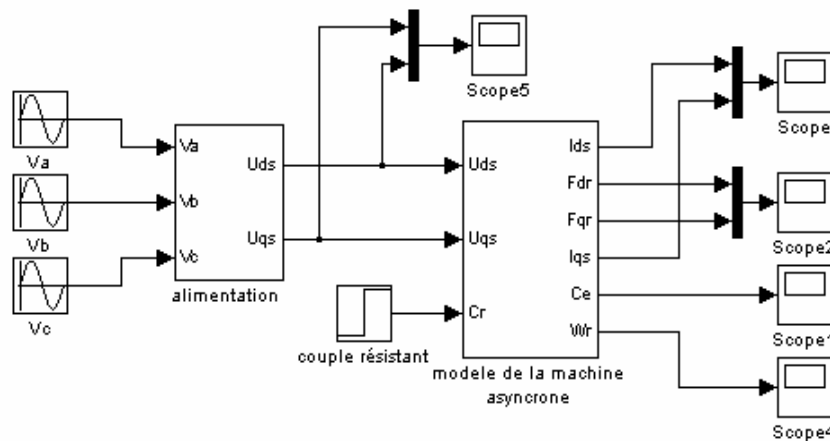


Figure ( 2.5 ) : Schéma en bloc de simulation de la machine asynchrone.

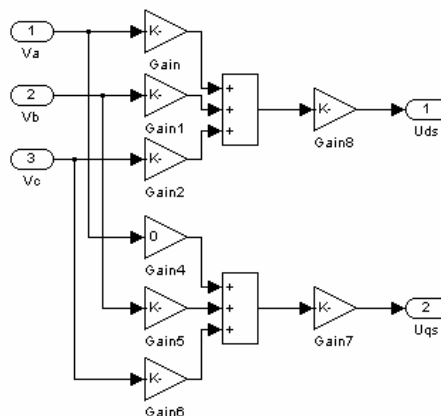


Figure ( 2.6 ) : Schéma du bloc « alimentation ».

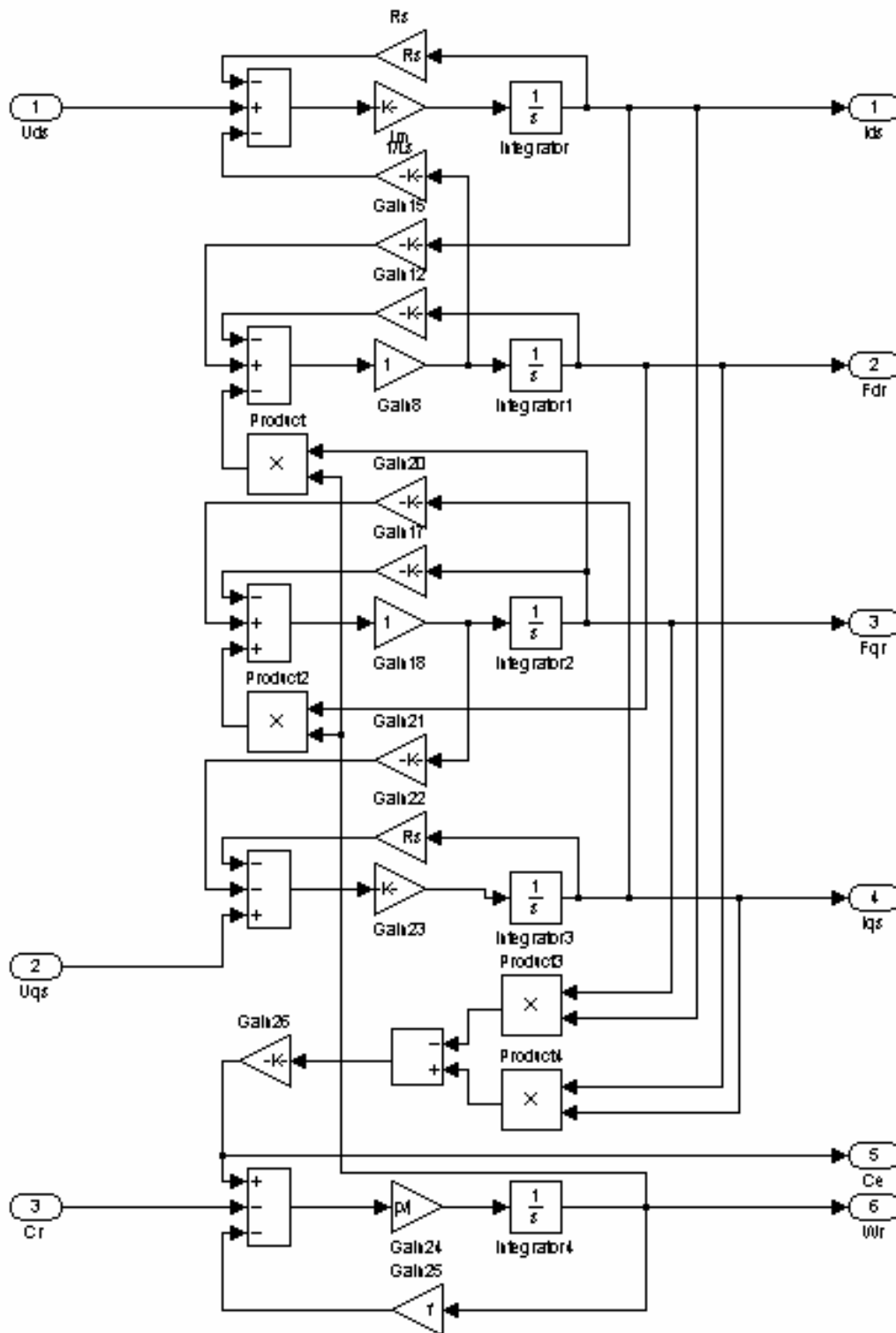
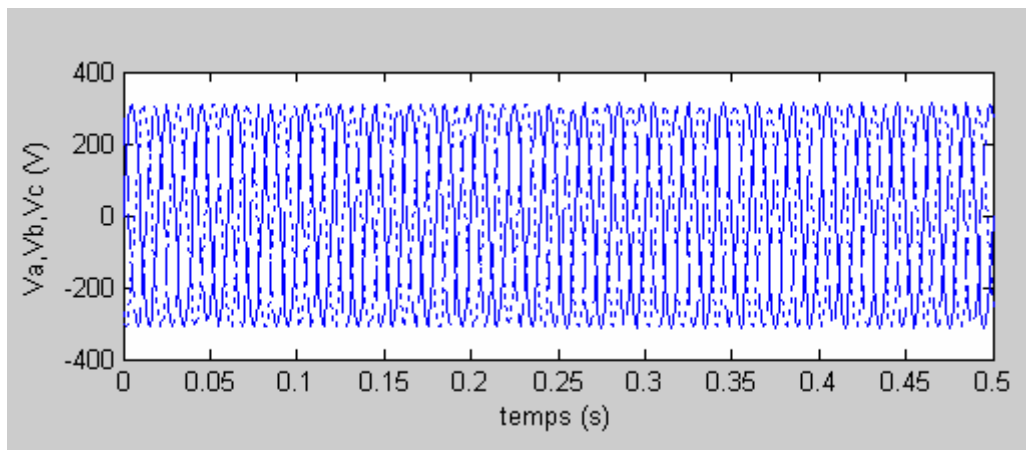


Figure ( 2.7 ) : Schéma du bloc « modèle de la machine asynchrone ».

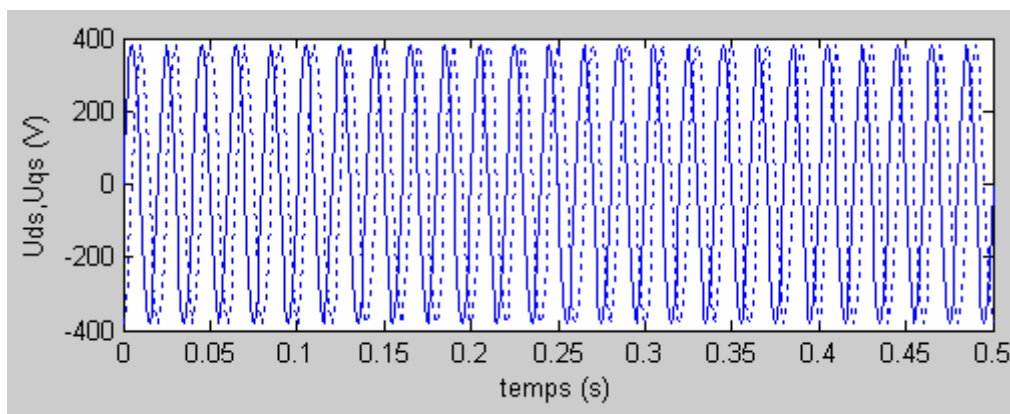
### 2.3.2. Résultats de la simulation

Nous avons représentés les trois tensions d'alimentations triphasé  $u_a, u_b, u_c$ , figure (2.8), et leur transformation de Park  $u_{ds}, u_{qs}$ , figure (2.9), les deux courants statoriques  $i_{ds}$  et  $i_{qs}$  figure (2.10), et les deux flux rotoriques  $\psi_{dr}$  et  $\psi_{qr}$ , figure (2.11) exprimés dans le plan de Park ainsi que la vitesse et le couple électromagnétique représentés respectivement sur la figure (2.12) et la figure (2.13).



**Figure ( 2.8 )** : Tensions d'alimentation statoriques

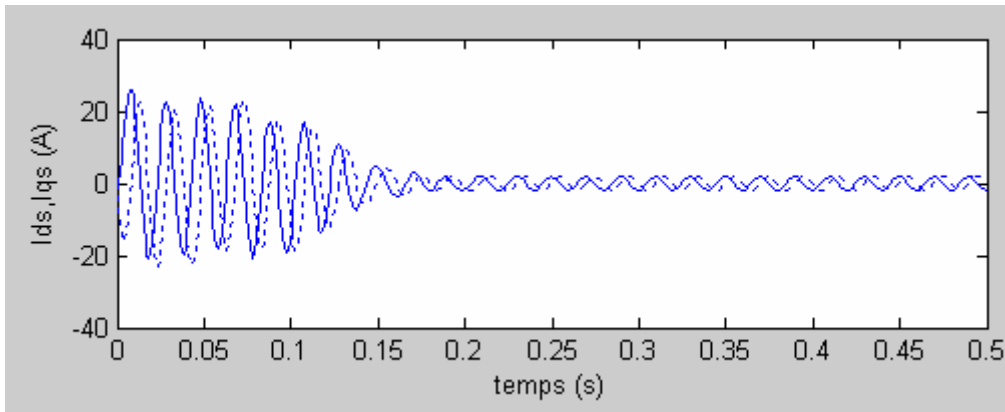
$v_a, v_b$  et  $v_c$ .



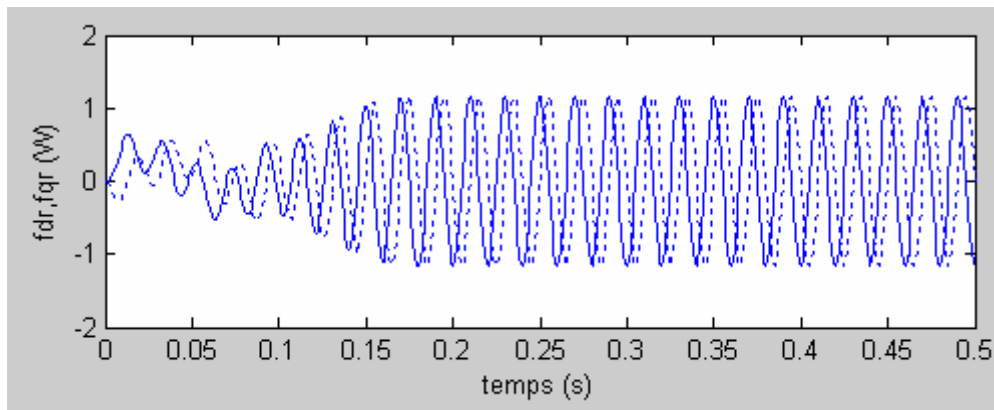
**Figure ( 2.9 )** : Transformation de Park des tensions

d'alimentation  $u_{ds}, u_{qs}$ .

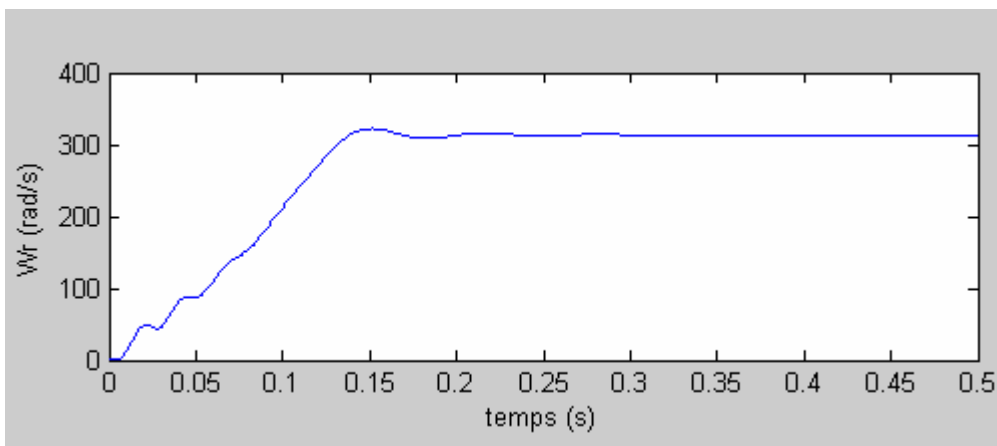




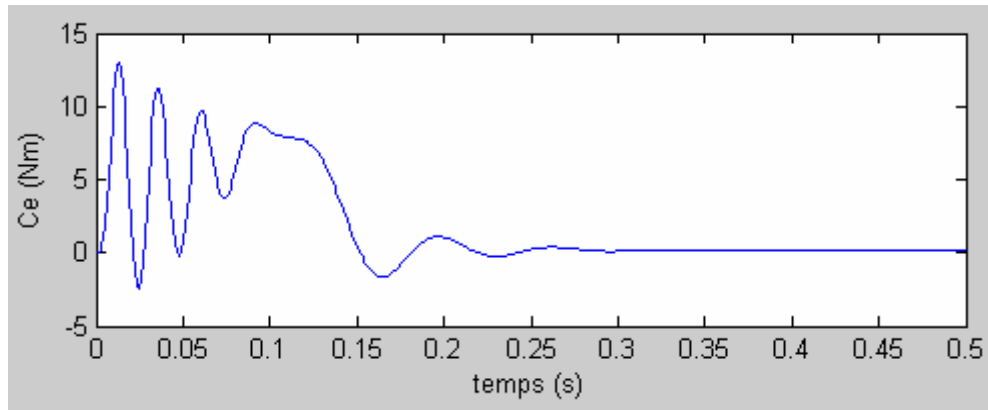
**Figure ( 2.10 )** : Courants statoriques  $i_{ds}$  et  $i_{qs}$  .



**Figure ( 2.11 )** : Flux rotoriques  $\psi_{dr}$  et  $\psi_{qr}$  .



**Figure ( 2.12 )** : Vitesse de rotation rotorique.



**Figure ( 2.13 )** : Couple électromagnétique.

### 2.3.3. Commentaires

L'analyse des courbes des différentes grandeurs mises en jeu lors et après le démarrage de la machine permet de faire les constatations suivantes :

- Un appel important de courant statorique au démarrage et stabilisation de ces grandeurs après le régime transitoire.
- Croissance de la vitesse et stabilisation après régime transitoire.
- Couple de démarrage important pour compenser l'inertie de la machine. Ce couple tend vers zéro en régime stable.
- Perturbation des flux au démarrage durons le régime transitoire ensuite stabilisation de ces flux durons le régime permanent.

## 2.4. Estimation des flux rotoriques de la machine par le filtre de Kalman

### 2.4.1 Définition du vecteur d'état

Le vecteur d'état choisi pour l'estimation des flux rotorique de la machine asynchrone est formé par les deux courants statoriques dans le plan de Park, mesuré directement à l'aide des capteurs de courant et des deux flux rotoriques constituant les grandeurs non accessibles à la mesure directe et à estimé par le filtre de Kalman. Le vecteur d'état est donc :

$$x(t) = [i_{ds}(t) \quad i_{qs}(t) \quad \psi_{dr}(t) \quad \psi_{qr}(t)]^T$$

### 2.4.2. Discrétisation du modèle d'état de la machine asynchrone

La mise en œuvre pratique du filtre de Kalman nécessite la discrétisation de équations d'état (2.17) de la machine asynchrone d'où :

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned} \quad (2.19)$$

$A_d$ ,  $B_d$  et  $C_d$  sont les matrices discrétisées de transition, de commande et d'observation respectivement. Elles sont données selon [5] par :

$$\begin{aligned} A_d &= e^{AT} \approx I + AT + \frac{(AT)^2}{2} \\ B &\approx BT + \frac{(ABT)^2}{2} \end{aligned} \quad (2.19)$$

Où :

$T$  : Est la période d'échantillonnage.

Par substitution des expressions des matrices A, B et C du modèle d'état, équation (2.17), les matrices discrétisés  $A_d$ ,  $B_d$  et  $C_d$  seront :

$$A_d = \begin{bmatrix} a_{11} & b_{11} & a_{12} & b_{12} \\ -b_{11} & a_{11} & -b_{12} & a_{12} \\ a_{21} & b_{21} & a_{22} & b_{22} \\ -b_{21} & a_{21} & -b_{22} & a_{22} \end{bmatrix}$$

$$B_d = \begin{bmatrix} a_1 & 0 & a_2 & 0 \\ 0 & a_1 & 0 & a_2 \end{bmatrix}^T$$

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Avec :

$$a_{11} = 1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T + \left( \left( \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} \right)^2 + \frac{L_m L_r}{\sigma L_s L_r^2} \frac{L_m R_r}{L_r} \right) \frac{T^2}{2}$$

$$\begin{aligned}
a_{12} &= \frac{L_m L_r}{\sigma L_s L_r^2} T + \left( \frac{L_m L_r}{\sigma L_s L_r^2} \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} - \frac{L_m L_r}{\sigma L_s L_r^2} \frac{R_r}{L_r} + \frac{L_m \omega_r^2}{\sigma L_s L_r} \right) \frac{T^2}{2} \\
a_{21} &= \frac{L_m R_r}{L_r} T + \left( \frac{L_m R_r}{L_r} \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} - \frac{L_m R_r}{L_r} \frac{R_r}{L_r} \right) \frac{T^2}{2} \\
a_{22} &= 1 - \frac{R_r}{L_r} T + \left( \frac{L_m R_r}{L_r} \frac{L_m L_r}{\sigma L_s L_r^2} + \left( \frac{R_r}{L_r} \right)^2 - \omega_r^2 \right) \frac{T^2}{2} \\
b_{11} &= \frac{L_m R_r}{L_r} \frac{L_m \omega_r}{\sigma L_s L_r} \frac{T^2}{2} \\
b_{12} &= \frac{L_m \omega_r}{\sigma L_s L_r} T + \left( \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} \frac{L_m \omega_r}{\sigma L_s L_r} - \frac{L_m L_r}{\sigma L_s L_r^2} \omega_r - \frac{L_m \omega_r}{\sigma L_s L_r} \frac{R_r}{L_r} \right) \frac{T^2}{2} \\
b_{21} &= -\frac{L_m R_r}{L_r} \omega_r \frac{T^2}{2} \\
b_{22} &= -\omega_r T + \left( \frac{L_m R_r}{L_r} \frac{L_m \omega_r}{\sigma L_s L_r} + 2 \frac{R_r}{L_r} \omega_r \right) \frac{T^2}{2} \\
a_1 &= \frac{T}{\sigma L_s} + \left( \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} \frac{1}{\sigma L_s} \right)^2 \frac{T^2}{2} \\
a_2 &= \left( \frac{1}{\sigma L_s} \frac{L_m R_r}{L_r} \right)^2 \frac{T^2}{2}
\end{aligned}$$

### 2.4.3. Equations du filtre de Kalman linéaire

Les équations du filtre de Kalman linéaire sont formées de deux groupes de calcul ; le premier groupe d'équations constitue la phase de prédiction des grandeurs à estimer à partir des données disponibles à l'instant courant ( $k$ ). Ces équations de prédiction sont :

- Equation de prédiction d'état

$$\hat{x}(k+1, k) = A_d \hat{x}(k, k) + B_d u(k)$$

- Equation de prédiction de la covariance de l'erreur

$$P(k+1, k) = AP(k, k)A_d^T + Q$$

Le deuxième groupe d'équations constitue la phase de correction. Il utilise les valeurs prédites calculées par les équations de prédiction. Il donne les valeurs estimées en utilisant les équations de correction et les grandeurs mesurées disponibles à l'instant de calcul ( $k + 1$ ). Ces équations de correction sont :

- Equation de calcul du gain de filtre de Kalman

$$K(k + 1) = P(k + 1, k)H(k)^T [H(k)P(k + 1, k)H^T(k) + R]^{-1}$$

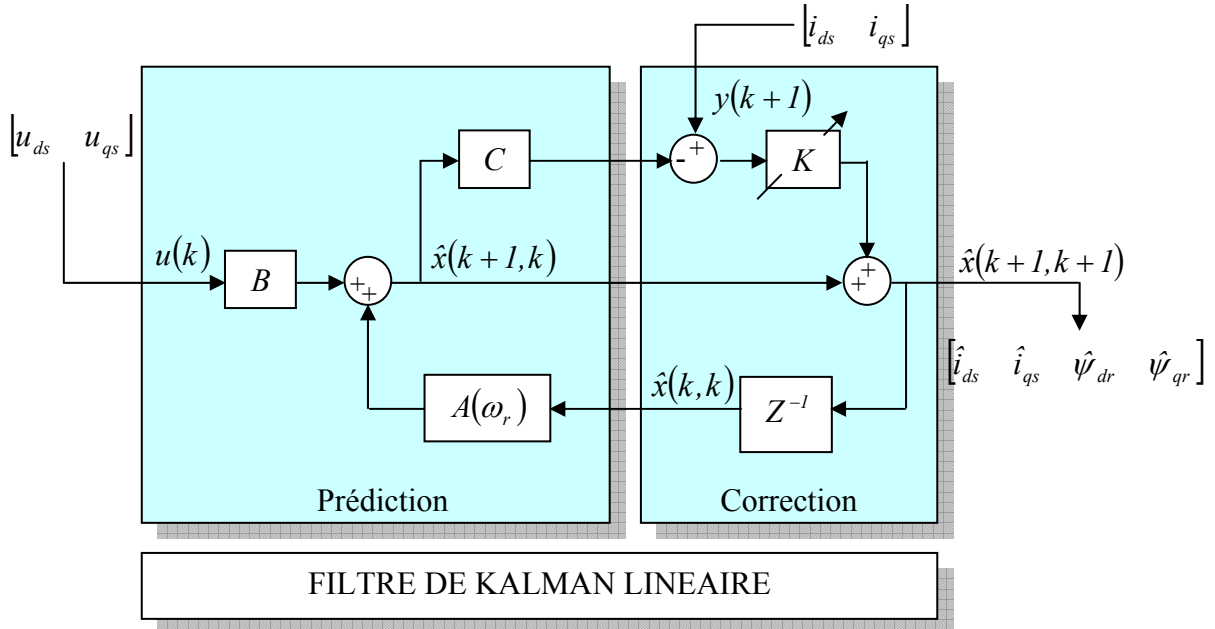
- Equation de correction d'état estimé

$$\hat{x}(k + 1, k + 1) = \hat{x}(k + 1, k) + K(k + 1)[y(k + 1) - H(k)\hat{x}(k + 1, k)]$$

- Equation de correction de la covariance de l'erreur d'estimation

$$P(k + 1, k + 1) = [I - K(k + 1)H(k)]P(k + 1, k)$$

Les équations de prédiction et de correction formant le filtre de Kalman peuvent se présenter schématiquement par le bloc diagramme de la figure (2.14).



**Figure (2.14) :** Diagramme en bloc du filtre de Kalman appliqué à la machine asynchrone.

### 2.4.5. Bruitage des signaux

Le filtre de Kalman linéaire est applicable à des grandeurs stochastiques entachées de bruit blanc gaussien, figure (2.15). Pour une simulation proche du réel, les signaux ou les grandeurs mis en jeu doivent être bruités. Nous avons additionné des bruits blancs gaussiens à des caractéristiques statistiques connues aux grandeurs mesurables de la machine asynchrone ( les tensions et les courants d'alimentation ), cela permet de prendre en compte les bruits dû aux instruments de mesure et le bruit de quantification des signaux.

Le bruit d'état est implicite à cause des hypothèses simplificatrices sur les équations du modèle de la machine. Par conséquent, nous avons jugé que l'addition de bruit au vecteur d'état est inutile.

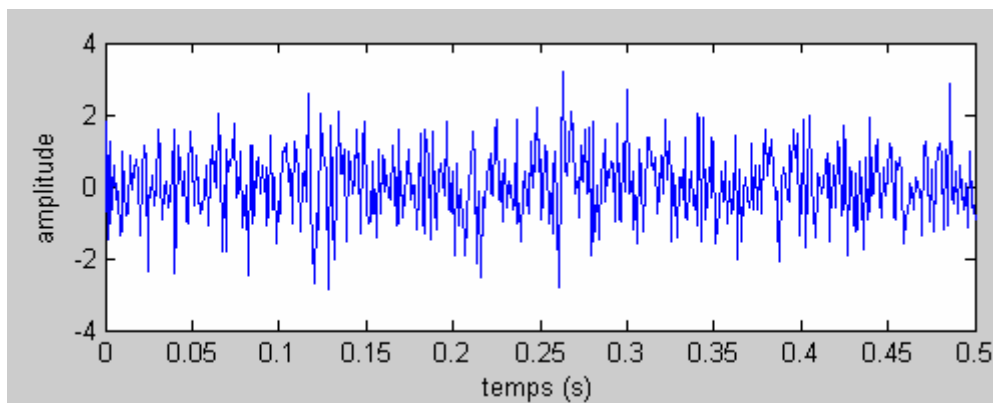
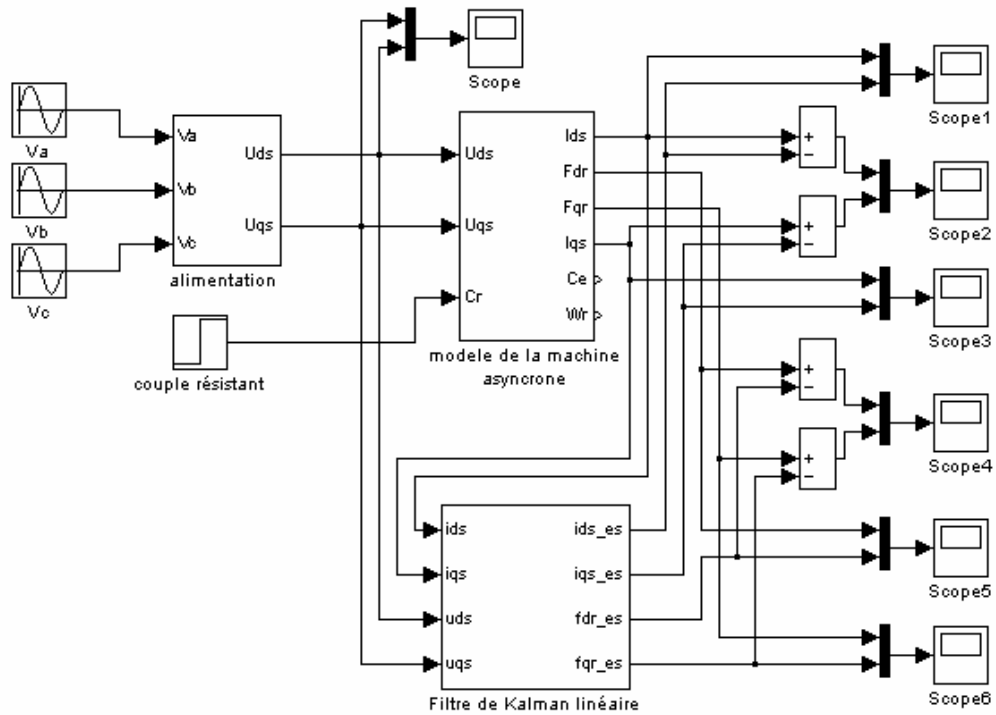


Figure (2.15) : Bruit blanc.

### 2.4.6. Schéma en bloc de la simulation de l'estimation des flux rotoriques de la machine asynchrone par le filtre de Kalman linéaire

Nous avons exploité les possibilités du logiciel SIMULINK pour la simulation de la machine. A ce modèle nous avons associé un module intégrant l'ensemble des équations du filtre de Kalman formulé pour l'estimation des flux rotoriques de la machine. Le diagramme complet de simulation est donné sur la figure (2.16).



**Figure (2.16) :** Schéma en blocs de la simulation de l'estimation des flux rotoriques.

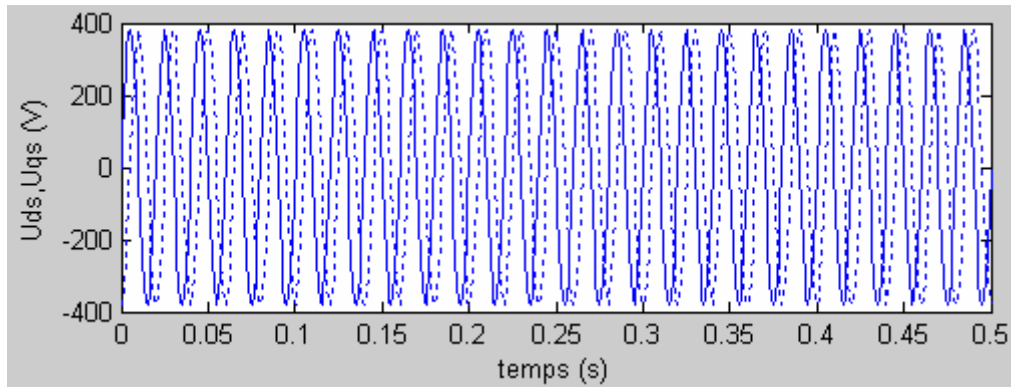
### 2.4.7. Temps d'échantillonnage

Des travaux de recherche effectués sur la dynamique de la machine asynchrone ont montré que le temps d'échantillonnage doit être inférieur ou égal à la valeur de la constante rotorique de la machine pour aboutir à des résultats satisfaisants [5]. Le temps d'échantillonnage choisi est de 1ms.

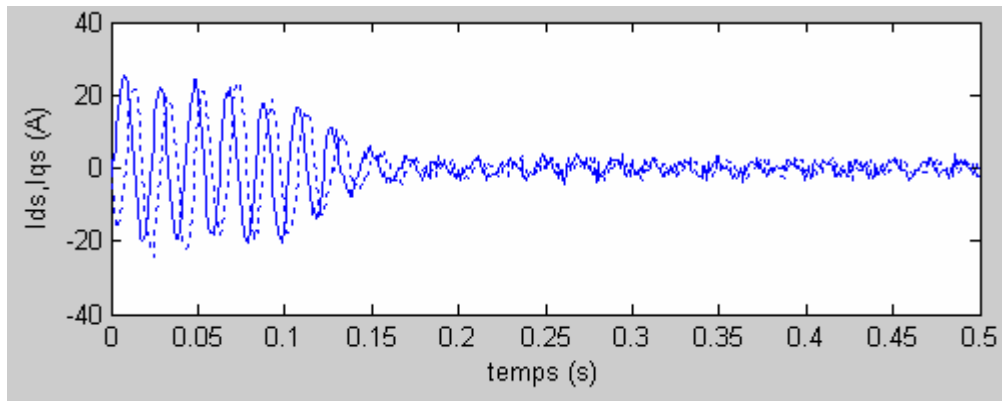
### 2.4.8. Résultats de simulation

#### 2.4.8.1. Grandeurs d'entrée pour le filtre de Kalman

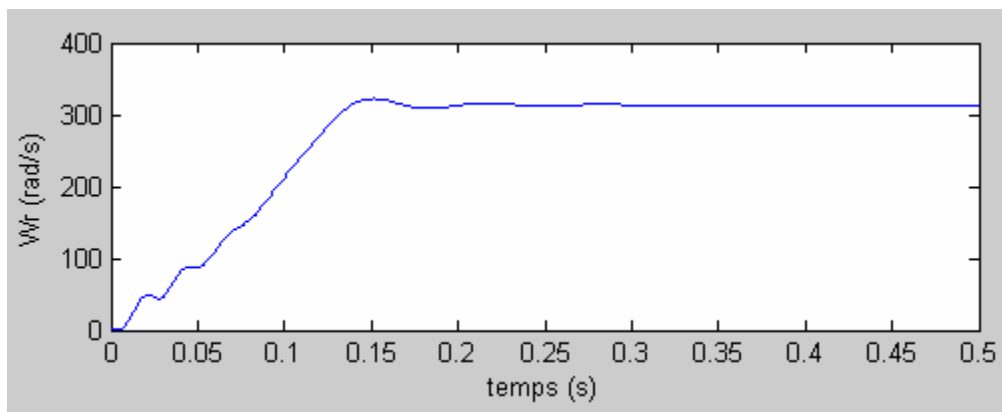
Le filtre de Kalman linéaire appliqué à la machine à comme grandeur d'entrée les deux tensions statoriques bruitées et les deux courants statoriques bruités dans le plan de Park ; figure (2.17) et figure (2.18), respectivement, ainsi que la pulsation rotorique ; figure (2.19).



**Figure (2.17) :** Tensions d'entrée bruitées du filtre de Kalman.



**Figure (2.18) :** Courant d'entrée bruités du filtre de Kalman.

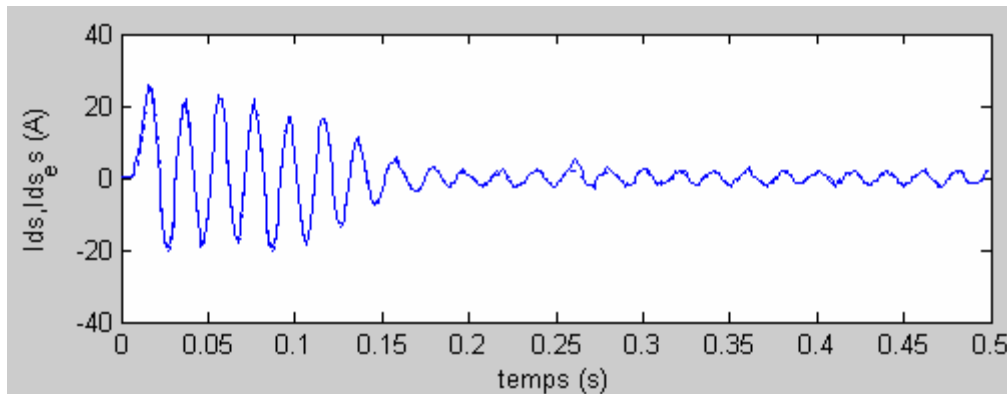


**Figure (2.19) :** Vitesse d'entrée du filtre de Kalman.

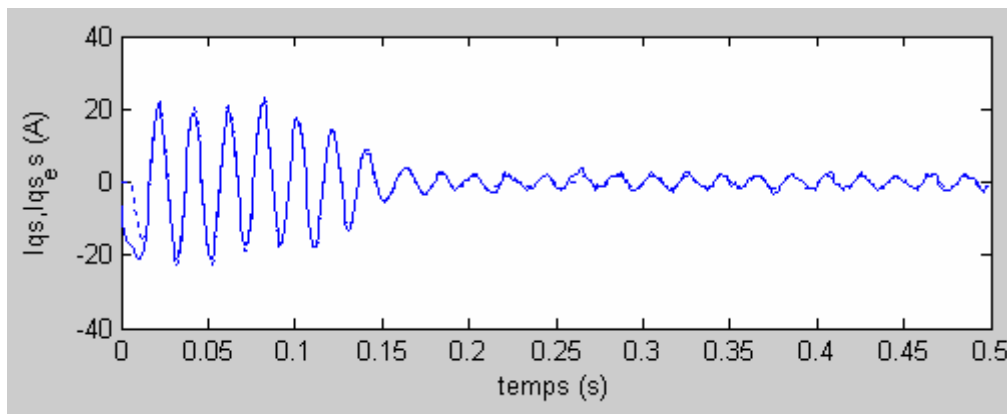


### 2.4.8.2. Résultats d'estimation

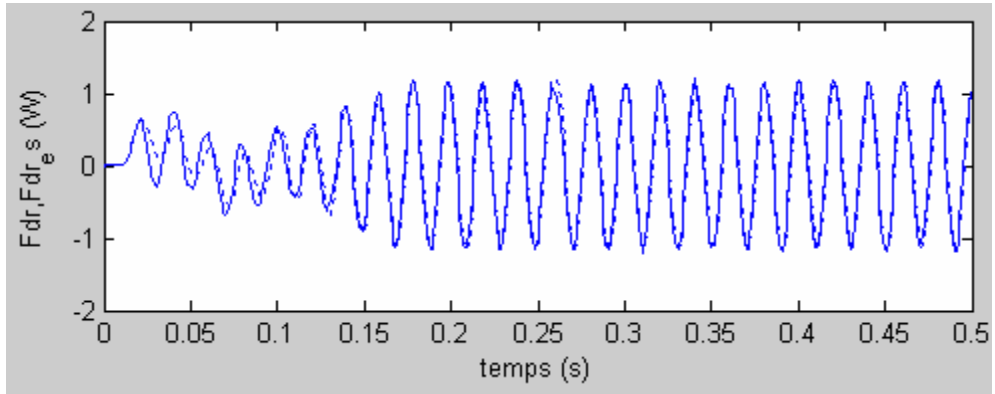
Les résultats d'estimation, obtenus par application de filtre de Kalman sur la machine asynchrone, sont les courants statoriques estimé et des flux rotorique estimé représentés respectivement sur les figures (2.20), figure (2.21), figure (2.22) et figure (2.23), avec le grandeur obtenus pas simulation du modele dynamique de la machine.



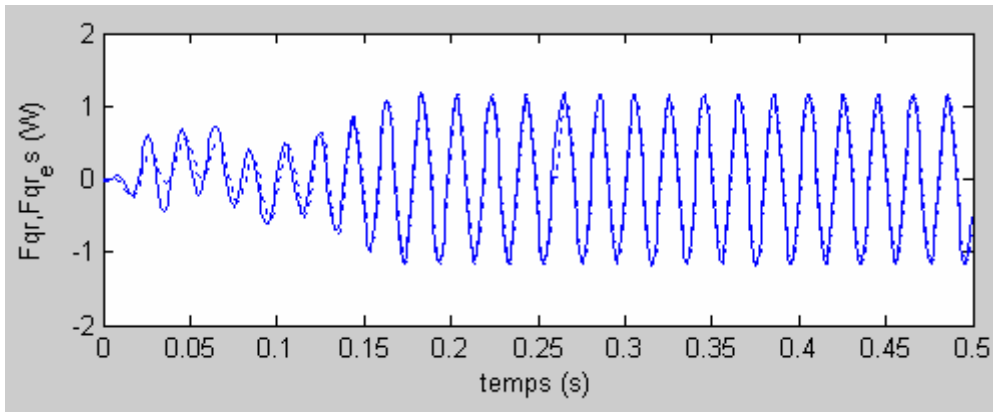
**Figure (2.20)** : Courants statoriques  $i_{ds}$  simulé ( .. )  
et estimé ( - ).



**Figure (2.21)** : Courants statoriques  $i_{qs}$  simulé ( .. )  
et estimé ( - ).

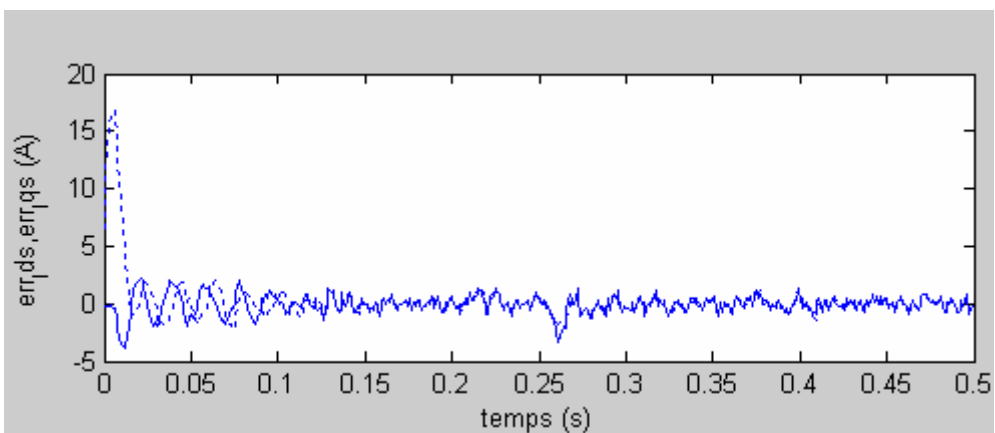


**Figure (2.22)** : Flux rotorique  $\psi_{dr}$  simulé (..) et estimé (-).

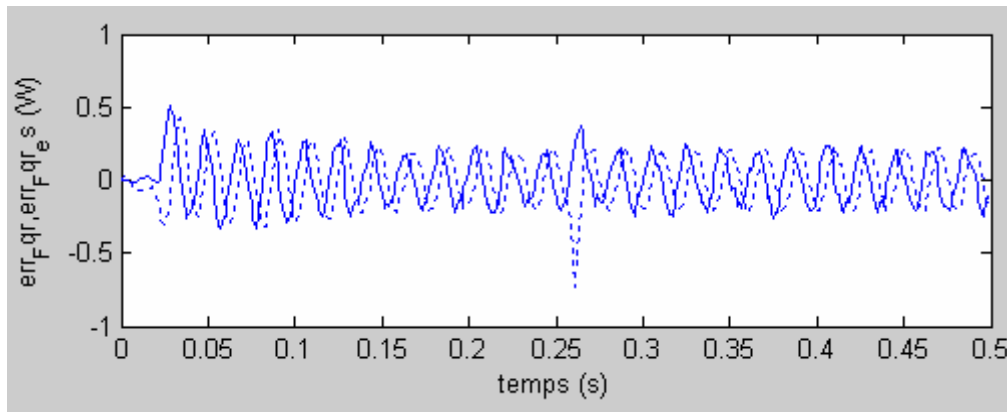


**Figure (2.23)** : Flux rotorique  $\psi_{qr}$  simulé (..) et estimé (-).

Les erreurs d'estimation des courants et des flux sont données sur les figures (2.24), (2.25).



**Figure (2.24)** : Erreurs d'estimation  $i_{ds}$  (..) et  $i_{qs}$  (-).



**Figure (2.25) :** Erreurs d'estimation  $\psi_{dr}$  (..) et  $\psi_{qr}$  (-).

## 2.5. Conclusion

Le développement d'algorithmes d'estimation nécessite la modélisation du système objet d'étude. Dans ce chapitre nous avons présenté le modèle dynamique de la machine asynchrone. Ce modèle est utilisé pour simuler, sous MATLAB SIMULINK, un démarrage à vide de la machine. Les résultats de simulation sont conformes aux grandeurs mesurées en cours de fonctionnement de la machine ( tensions, courants et vitesse ), cela permet la validation du modèle théorique adopté.

Sur ce modèle théorique, caractérisant le comportement de la machine, nous avons formulé le filtre de Kalman linéaire pour l'estimation des grandeurs non accessibles à la mesure directe ( flux rotoriques ). Les résultats de simulation obtenus par le filtre de Kalman sont satisfaisants et ils sont valides pour une implantation réaliste.

## ***Estimation de la vitesse rotorique de la machine asynchrone par le filtre de Kalman étendu***

### **3.1. Introduction**

Dans le modèle de la machine asynchrone présenté dans le deuxième chapitre, nous avons supposé que les paramètres de la machine sont constants et invariables dans le temps. Réellement ces grandeurs changent en fonction de plusieurs paramètres tel que :

- La variation des constantes de temps rotorique et statorique en fonction de la température (  $T_r(T)$ ,  $T_s(T)$  ).
- La variation de la vitesse rotorique en fonction du temps ( régime transitoire ).

Il est évident que dans ces conditions, que le modèle de la machine asynchrone ne peut être considéré linéaire. Le filtre de Kalman linéaire est par conséquent est non applicable pour l'estimation de ces paramètres. L'utilisation de filtre de Kalman étendu est indispensable. Ce filtre permet l'estimation de l'état d'un système non linéaire et ces paramètres.

Dans ce chapitre nous exposons le filtre de Kalman étendu et ces principes, puis son application à l'estimation de la vitesse rotorique de la machine asynchrone à cage. Les résultats de simulation sous MATLAB SIMULINK seront également présentés.

### **3.2. Filtre de Kalman étendu**

Le filtre de Kalman standard décrit dans le premier chapitre, permet d'estimer l'état d'un système linéaire ( les coefficients des équations d'état sont constants ). Dans le cas contraire où les coefficients sont variables ( système non linéaire ), l'estimation des coefficients ( paramètres ) consiste à étendre le vecteur d'état aux coefficients voulus. Le mode devient alors non linéaire. Certain des applications les plus intéressantes et les plus réussies du filtre de Kalman ont été une telle situation ou le système est non linéaire [3].

La linéarisation des équations du système autour d'un point de fonctionnement en utilisant les séries de Taylor, nous permet d'aboutir à des équations similaires au cas du filtre de Kalman linéaire [13]. Le filtre de Kalman linéarisé autour du point de fonctionnement actuel est le filtre de Kalman étendu [17].

Les équations du filtre de Kalman linéaire, se trouvent légèrement modifiées. Les équations d'observation et de mesure dans le cas d'un système non linéaire s'écrivent :

$$\begin{aligned} x(k+1) &= f(x(k), u(k), W(k+1)) \\ y(k) &= h(x(k), V(k)) \end{aligned} \tag{3.1}$$

$W(k)$  et  $V(k)$  représentent toujours les bruits d'état et de mesure. Les hypothèses faites précédemment sur ces bruits restent toujours valables. La fonction non linéaire  $f$  relie l'état précédent du système à l'état actuel. Elle inclut l'état et les paramètres du système.

Nous définissons le vecteur des paramètres à estimer ainsi par:

$$\theta = [\theta_1(k), \theta_2(k), \dots, \theta_m(k)] \tag{3.2}$$

L'état étendu du système sera :

$$x(k) = [x_1(k), x_2(k), \dots, x_n(k), \theta_1(k), \theta_2(k), \dots, \theta_m(k)] \tag{3.3}$$

Selon [13] le modèle d'état stochastique est :

$$\begin{bmatrix} x(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} A(x(k), \theta(k)) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} B(x(k), \theta(k)) \\ 0 \end{bmatrix} + \begin{bmatrix} W_x(k) \\ W_\theta(k) \end{bmatrix} \tag{3.4}$$

$$y(k) = [C \quad 0] \begin{bmatrix} x(k) \\ \theta(k) \end{bmatrix} + V(k)$$

$Q_x = [W_x(k) \cdot W_x(k)^T]$  représente la matrice de covariance d'état. En absence d'information supplémentaire, on considère les variances des paramètres comme aléatoires et on les modélise par des bruits blancs centrés. La dynamique des paramètres est caractérisée par la matrice de covariance  $Q$ . Plus une variance est grande, plus le paramètre correspond est

important [13]. En supposant que les paramètres et les états sont indépendants, on définit une matrice diagonale de covariance de bruit d'état et des paramètres comme suit :

$$Q = \begin{bmatrix} Q_x & 0 \\ 0 & Q_\theta \end{bmatrix} = \begin{bmatrix} Q_{x1} & 0 & & \dots & & 0 \\ 0 & Q_{x2} & & & & \\ & & \ddots & & & \\ & & & Q_{xn} & & \vdots \\ \vdots & & & & Q_{\theta 1} & \\ & & & & & Q_{\theta 2} & \\ & & & & & & \ddots & 0 \\ 0 & & \dots & & & & 0 & Q_{\theta n} \end{bmatrix}$$

En pratique, les valeurs des bruits à chaque pas d'échantillonnage sont inaccessibles à la mesure directe. C'est pour cela, il faut utiliser des valeurs estimées qui ne tiennent pas compte des valeurs des bruits. Donc, les équations (3.1) sont approximées aux équations suivantes :

$$\begin{aligned} \tilde{\hat{x}}(k+1) &= f(\hat{x}(k), u(k), \theta) \\ y(k) &= h(\hat{x}(k), \theta) \end{aligned} \tag{3.5}$$

Où  $\hat{x}(k)$  est l'estimation à posteriori de l'état calculé à l'instant discret  $k$ . Cette nouvelle forme d'équations d'état du système nous aide à déterminer les équations du filtre de Kalman étendu.

### 3.2.1. Equations du filtre de Kalman étendu

Les équations du filtre de Kalman étendu sont de forme similaire aux équations du filtre de Kalman standard à l'exception de l'équation de prédiction d'état qui est remplacée par une équation non linéaire. La seule différence entre les deux groupes d'équations est la linéarisation des fonctions non linéaires par le développement de Taylor du premier ordre [3][13], c'est à dire les dérivés partielles premières des fonctions non linéaires par rapport à l'état et les paramètres du système.

Les prédictions de l'état et de la covariance d'état sont données par les équations suivantes :

$$\begin{aligned}\hat{x}(k+1, k) &= f(\hat{x}(k), u(k)) \\ P(k+1, k) &= F(k)P(k, k)F^T(k) + Q(k)\end{aligned}\quad (3.6)$$

Où :

$$F(k) = \left. \frac{\partial f(\hat{x}(k), u(k))}{\partial x} \right|_{x=\hat{x}(k, k)}$$

$f$ : la fonction non linéaire régissant le système.

La matrice de gain du filtre de Kalman est calculée par l'équation suivante :

$$K(k+1) = P(k+1, k) \cdot H^T(k+1) [H(k+1)P(k+1, k)H^T(k) + R(k)]^{-1} \quad (3.7)$$

Où  $R$  représente la matrice de covariance du bruit de mesure et :

$$H(k+1) = \left. \frac{\partial h(\hat{x}(k), \theta)}{\partial x} \right|_{x=\hat{x}(k+1, k)}$$

Finalement, l'estimation de l'état est donnée par :

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1)[y(k+1) - H(k+1)\hat{x}(k+1, k)] \quad (3.8)$$

Sachant que notre filtre est récursif, l'actualisation de la covariance de l'erreur est donnée par :

$$P(k+1, k+1) = [I - K(k+1)H(k+1)]P(k+1, k) \quad (3.9)$$

Les équations citées ci-dessus forment le filtre de Kalman étendu destiné à l'estimation de l'état et des paramètres du système à observer.

### 3.2.2. Algorithme du filtre de Kalman étendu

Comme le filtre de Kalman standard, les équations du filtre de Kalman étendu se regroupent en deux ensembles : les équations de prédiction, tableau (3.1), composées de l'équation de prédiction d'état qui prévoit la valeur de l'état à l'instant  $(k+1)$  en se basant sur

des informations disponibles à l'instant  $k$ , et aussi l'équation de prédiction de la covariance de l'erreur d'estimation qui va être utilisée dans le calcul du gain du filtre.

**Tableau (3.1) :** Equations de prédiction.

$$\hat{x}(k+1, k) = f(\hat{x}(k), u(k))$$

$$P(k+1, k) = F(k)P(k, k)F^T(k) + Q(k)$$

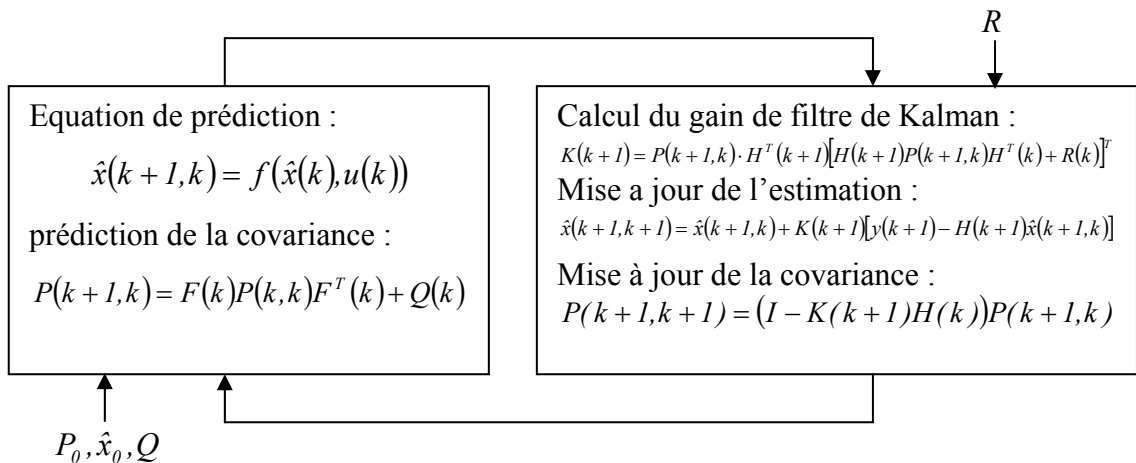
**Tableau (3.2) :** Equations de correction.

$$K(k+1) = P(k+1, k) \cdot H^T(k+1) [H(k+1)P(k+1, k)H^T(k) + R(k)]^{-1}$$

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1) [y(k+1) - H(k+1)\hat{x}(k+1, k)]$$

$$P(k+1, k+1) = [I - K(k+1)H(k+1)]P(k+1, k)$$

L'ensemble des équations de correction est donné dans le tableau (3.1). Il est composé de l'équation du gain, l'actualisation de l'estimation et l'actualisation de la covariance de l'erreur. Contrairement au filtre de Kalman standard les matrices F et H dans le filtre de Kalman étendu sont variables. Cela veut dire qu'elles sont actualisées à chaque itération selon la variation des paramètres estimés. L'organigramme de la figure (3.2) nous présente le séquençement et l'évaluation des équations du filtre de Kalman étendu, tandis que la figure (3.1) nous donne les étapes de prédiction et de correction :



**Figure (3.1) :** Processus itératif du filtre de Kalman étendu.



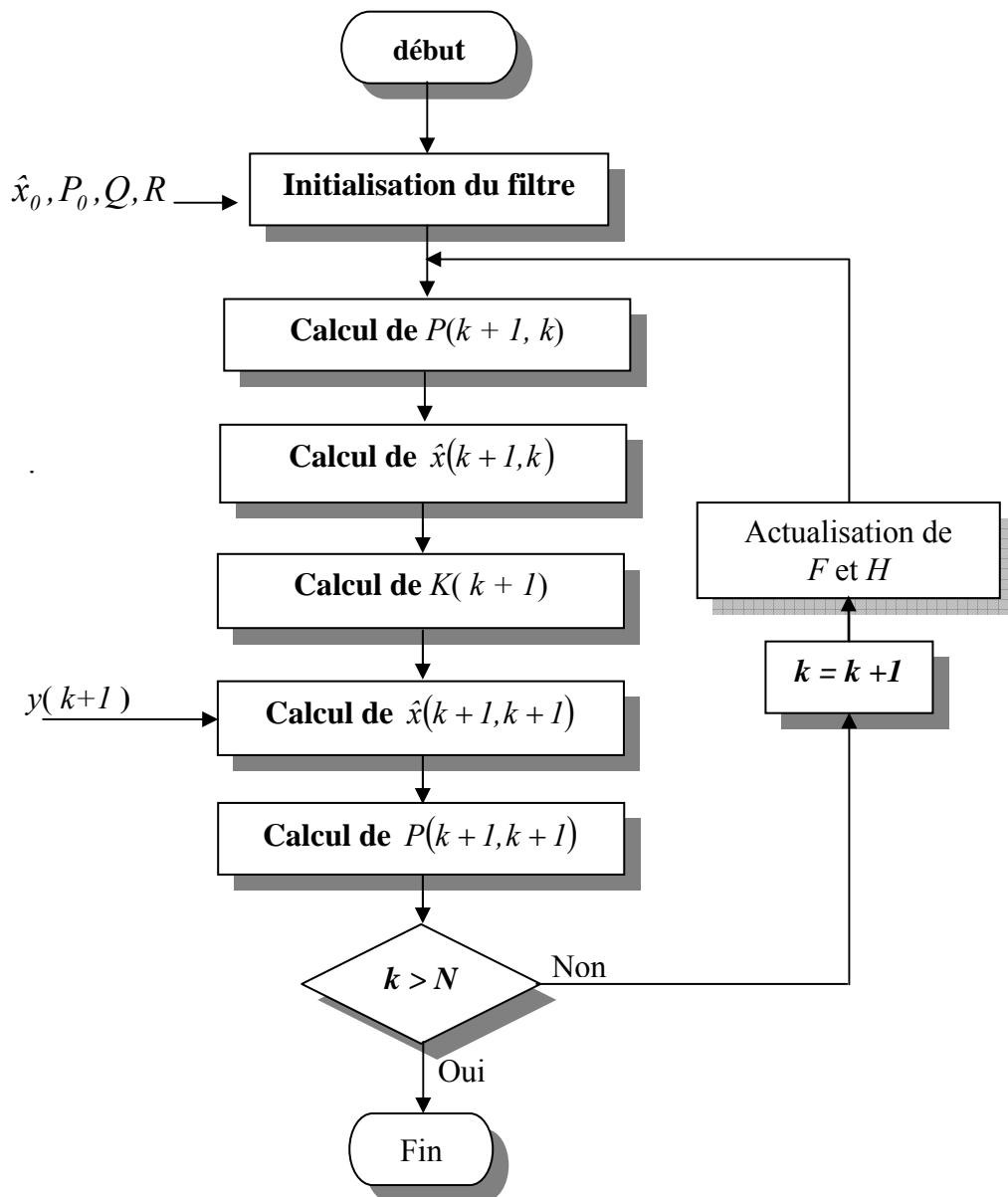


Figure (3.2) : Organigramme du filtre de Kalman étendu.

### 3.3. Estimation de la vitesse rotorique de la machine asynchrone

Les stratégies de commande de la machine asynchrone se basent en totalité sur la mesure de la vitesse rotorique. Cette mesure est faite à l'aide d'une génératrice tachymétrique ou d'un capteur de vitesse optique à base d'un disque perforé. Par fois, l'utilisation d'un capteur présente quelques inconvénients [5] [8] :

- Il est très cher.
- Il peut prendre beaucoup de place dans les milieux encombrés.
- Il est dans un environnement agressif; ce qui influe sur la mesure ou être endommagé. D'où, il représente une partie faible du système.

La solution qui permet de surmonter ces inconvénients, et surtout le dernier qui réduit beaucoup les performances du système, est l'utilisation d'une commande sans capteur de vitesse. La vitesse est donc estimée, en mesurant les tensions et les courants de la machine asynchrone, par utilisation du filtre de Kalman étendu.

### 3.3.1. Modèle d'état de la machine pour le filtre de Kalman étendu

Le modèle d'état de la machine développé dans un référentiel lié aux stator est donné par :

$$\frac{\partial}{\partial t} \begin{bmatrix} i_{ds} \\ i_{qs} \\ \psi_{dr} \\ \psi_{qr} \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} & 0 & \frac{L_m L_r}{\sigma L_s L_r^2} & \frac{L_m \omega_r}{\sigma L_s L_r} & 0 \\ 0 & -\frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} & -\frac{L_m \omega_r}{\sigma L_s L_r} & \frac{L_m L_r}{\sigma L_s L_r^2} & 0 \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & -\omega_r & 0 \\ 0 & \frac{L_m R_r}{L_r} & \omega_r & -\frac{R_r}{L_r} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ \psi_{dr} \\ \psi_{qr} \\ \omega_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} \quad (3.10)$$

$$\begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ \psi_{dr} \\ \psi_{qr} \\ \omega_r \end{bmatrix} \quad (3.11)$$

### 3.3.2. Modèle discret de la machine pour filtre de Kalman étendu

Les équations données ci-dessus doivent être discrétisées pour une implantation numérique des algorithmes du filtre de Kalman étendu où :

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned} \quad (3.12)$$

$A_d$  et  $B_d$  sont les matrices discrétisées du système et de mesure respectivement. Elles sont données par :

$$\begin{aligned} A_d &= e^{AT} \approx I + AT + \frac{(AT)^2}{2} \\ B_d &\approx BT + \frac{(ABT)^2}{2} \end{aligned} \quad (3.13)$$

La meilleure approximation est assurée par l'utilisation de développement en série de deuxième ordre. Mais, cette approximation augmente beaucoup le temps d'exécution en cas d'une implantation temps réel. Le choix de l'approximation doit satisfaire le temps l'exécution réduit et l'exactitude et la stabilité du filtre [5]. Notre choix dans cette application est une approximation du premier ordre tel que :

$$\begin{aligned} A_d &\approx I + AT \\ B &\approx BT \end{aligned} \quad (3.12)$$

Avec :

$$A_d = \begin{bmatrix} 1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T & 0 & \frac{L_m L_r}{\sigma L_s L_r^2} T & \frac{L_m \omega_r}{\sigma L_s L_r} T & 0 \\ 0 & 1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T & \frac{L_m \omega_r}{\sigma L_s L_r} T & \frac{L_m L_r}{\sigma L_s L_r^2} T & 0 \\ \frac{L_m R_r}{L_r} T & 0 & 1 - \frac{R_r}{L_r} T & -\omega_r T & 0 \\ 0 & \frac{L_m R_r}{L_r} T & \omega_r T & 1 - \frac{R_r}{L_r} T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$C_d = C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

### 3.3.3. Equations du filtre de Kalman étendu appliqué a la machine

#### a - Equation de prédiction d'état

L'équation de prédiction d'état du filtre de Kalman étendu s'écrit sous la forme :

$$\hat{x}(k+1, k) = f(\hat{x}(k), u(k)) \quad (3.13)$$

Où :

$$f = \begin{bmatrix} \left(1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T\right) \hat{i}_{ds} + \left(\frac{L_m L_r}{\sigma L_s L_r^2} T\right) \hat{\psi}_{dr} + \left(\frac{L_m \omega_r}{\sigma L_s L_r} T\right) \hat{\psi}_{qr} \\ \left(1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T\right) \hat{i}_{qs} + \left(\frac{L_m \omega_r}{\sigma L_s L_r} T\right) \hat{\psi}_{dr} + \left(\frac{L_m L_r}{\sigma L_s L_r^2} T\right) \hat{\psi}_{qr} \\ \left(\frac{L_m R_r}{L_r} T\right) \hat{i}_{ds} + \left(1 - \frac{R_r}{L_r} T\right) \hat{\psi}_{dr} + (-\omega_r T) \hat{\psi}_{qr} \\ \left(\frac{L_m R_r}{L_r} T\right) \hat{i}_{qs} + (\omega_r T) \hat{\psi}_{dr} + \left(1 - \frac{R_r}{L_r} T\right) \hat{\psi}_{qr} \\ \omega_r \end{bmatrix}$$

#### b - Equation de prédiction de covariance

L'équation de prédiction de covariance de l'erreur d'estimation est donnée par :

$$P(k+1, k) = F(k)P(k, k)F^T(k) + Q(k) \quad (3.13)$$

Où :

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T & 0 & \frac{L_m L_r}{\sigma L_s L_r^2} T & \frac{L_m \omega_r}{\sigma L_s L_r} & T \frac{L_m}{\sigma L_s L_r} \psi_{qr} \\ 0 & 1 - \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2} T & \frac{L_m \omega_r}{\sigma L_s L_r} T & \frac{L_m L_r}{\sigma L_s L_r^2} T & T \frac{L_m}{\sigma L_s L_r} \psi_{dr} \\ \frac{L_m R_r}{L_r} T & 0 & 1 - \frac{R_r}{L_r} T & -\omega_r T & -T \psi_{qr} \\ 0 & \frac{L_m R_r}{L_r} T & \omega_r T & 1 - \frac{R_r}{L_r} T & T \psi_{dr} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**c - Equation de calcul du gain de filtre de Kalman**

$$K(k+1) = P(k+1, k) H(k)^T [H(k) P(k+1, k) H(k)^T + R]^{-1}$$

**d - Equation de correction d'état estimé**

$$\hat{x}(k+1, k+1) = \hat{x}(k+1, k) + K(k+1) [y(k+1) - H(k) \hat{x}(k+1, k)]$$

**e - Equation de correction de la covariance de l'erreur d'estimation**

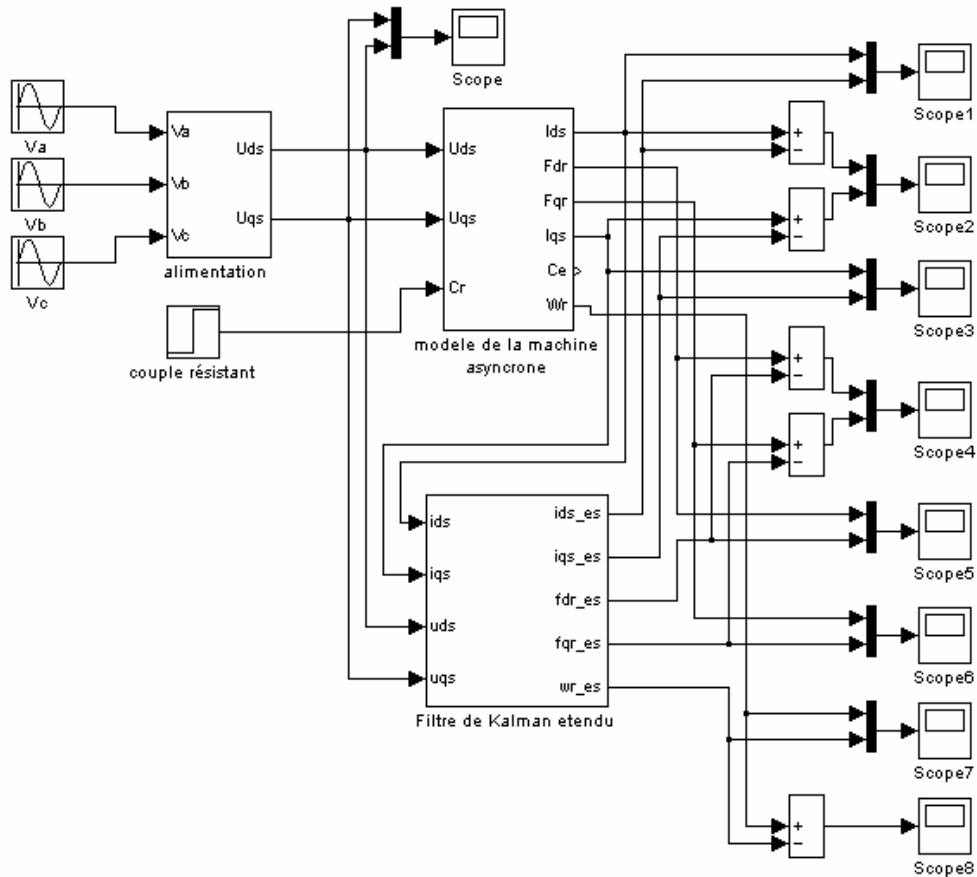
$$P(k+1, k+1) = [I - K(k+1) H(k)] P(k+1, k)$$

Où :

$$H(k+1) = \left. \frac{\partial h(\hat{x}(k), 0)}{\partial x} \right|_{x=\hat{x}(k+1, k)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

**3.4. Simulation de l'estimation en boucle ouverte de la vitesse rotorique par filtre de Kalman étendu**

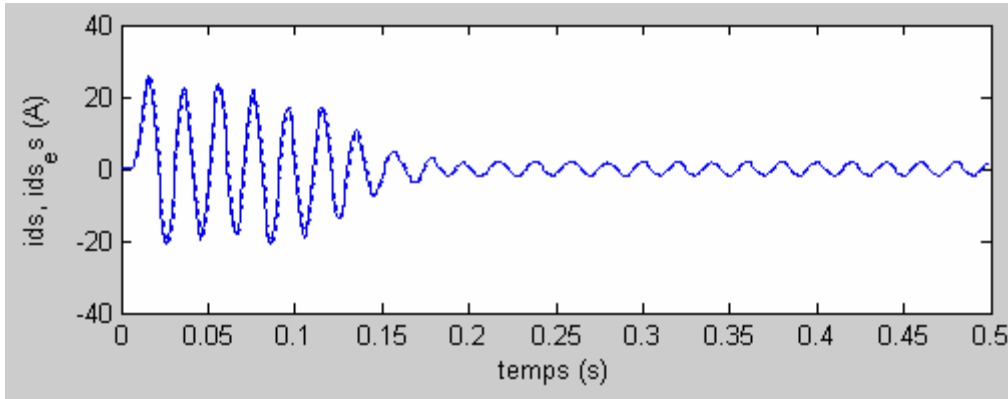
Dans cette partie, nous allons implanter la totalité des équations du filtre de Kalman étendu avec la simulation de la machine asynchrone sur MATLAB/SIMULINK pour estimer les flux et la vitesse rotoriques. Les grandeurs estimées peuvent être affichées seulement ( cas d'une estimation en boucle ouverte ) ou utilisées pour calculer d'autres variables ( cas d'une commande vectorielle par exemple ). Le diagramme bloc de l'estimation par le filtre de Kalman étendu est donné dans la figure (3.3) ; où les tensions d'entrée et les courants de sortie de la machine sont aiguillés vers le filtre de Kalman étendu pour estimer les flux et la vitesse rotoriques de la machine.



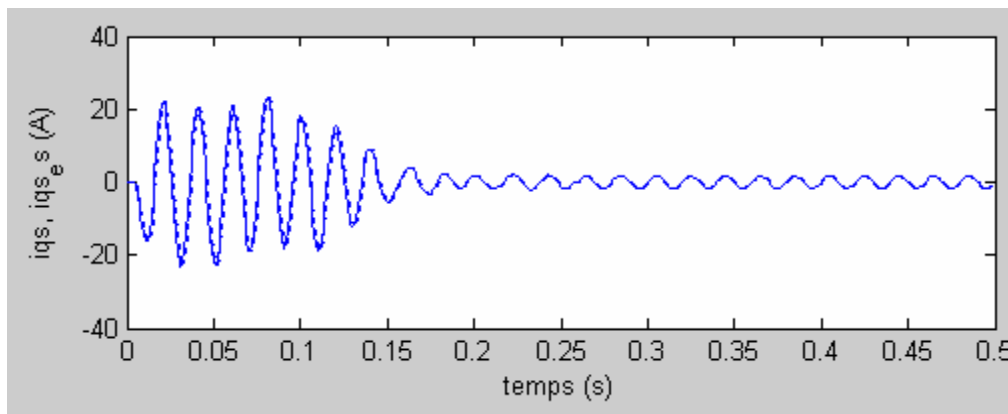
**Figure (3.3) :** Schéma en bloc de simulation du EKF appliqué sur la machine asynchrone.

### 3.4.1. Résultats de simulation

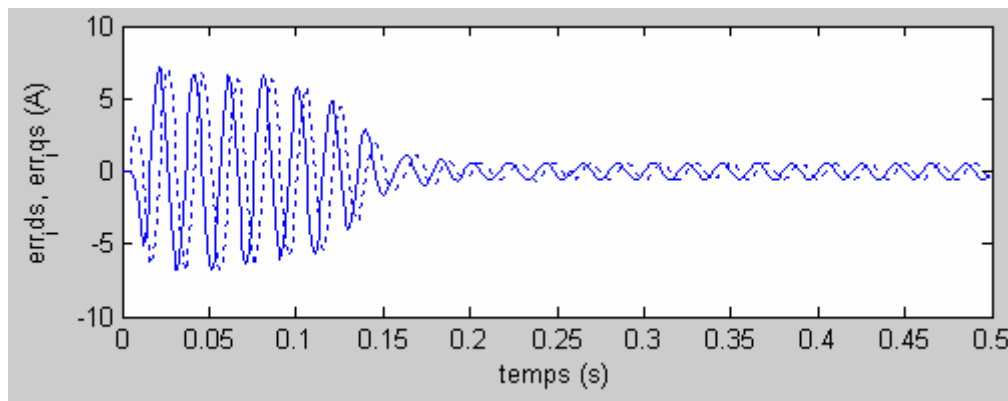
Les résultats obtenus de la simulation du filtre de Kalman étendu se résument dans les courants statoriques estimés et simulés ; figure (3.4) et figure (3.5) avec les erreurs d'estimation associées ; figure (3.6), les flux rotoriques estimés et simulés ; figure (3.7) et figure (3.8) avec les erreurs d'estimations correspondantes, figure (3.9) et en fin la vitesse rotorique estimée et simulé, figure (3.10) accompagnée de l'erreur d'estimation, figure (3.11).



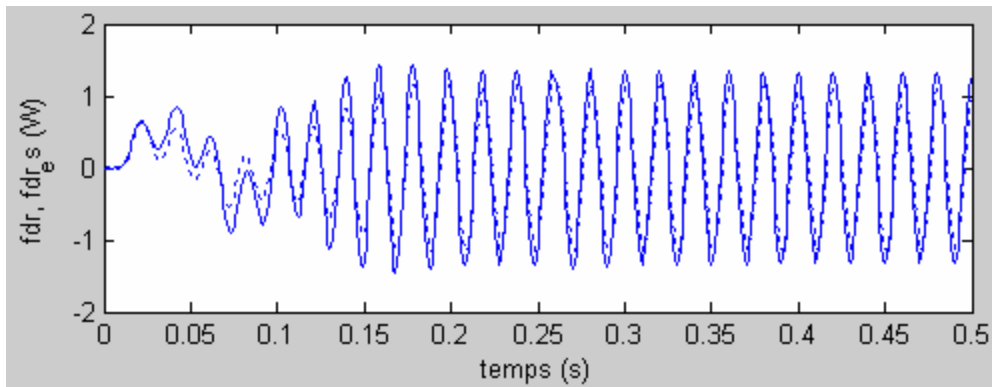
**Figure (3.4)** : Courants statoriques  $i_{ds}$  simulé (..) et estimé (-).



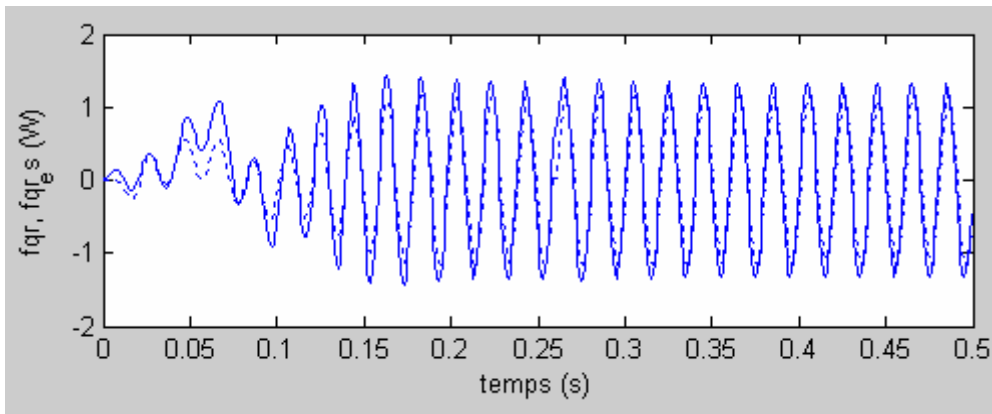
**Figure (3.5)** : Courants statoriques  $i_{qs}$  simulé (..) et estimé (-).



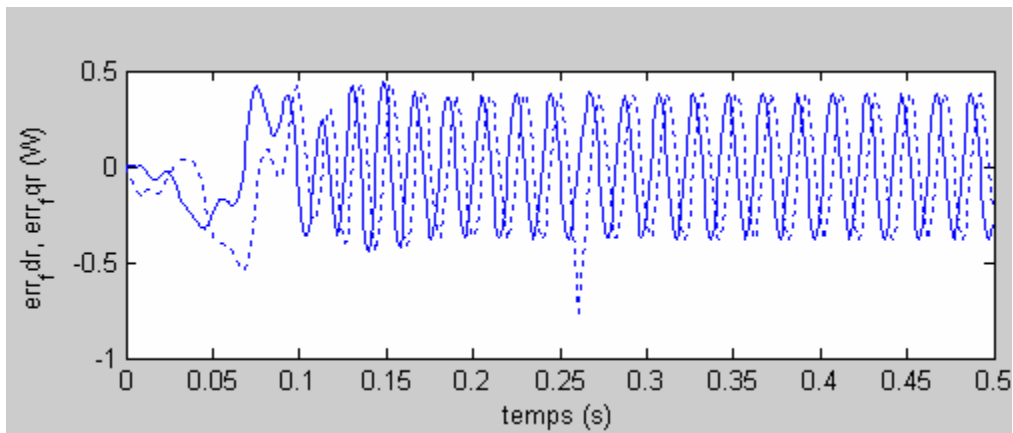
**Figure (3.6)** : Erreurs d'estimation  $i_{ds}$  (..) et  $i_{qs}$  (-).



**Figure (3.7) :** Flux rotorique  $\psi_{dr}$  simulé ( .. )  
et estimé ( - ).

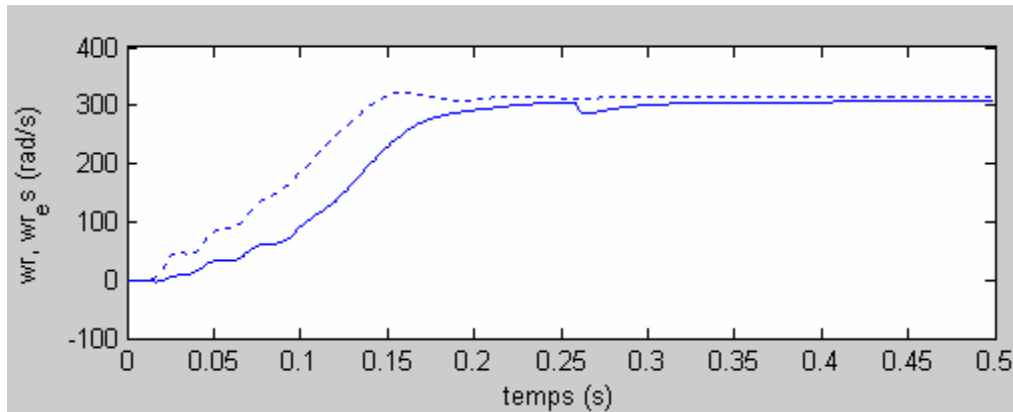


**Figure (3.8) :** Flux rotorique  $\psi_{qr}$  simulé ( .. )  
et estimé ( - ).

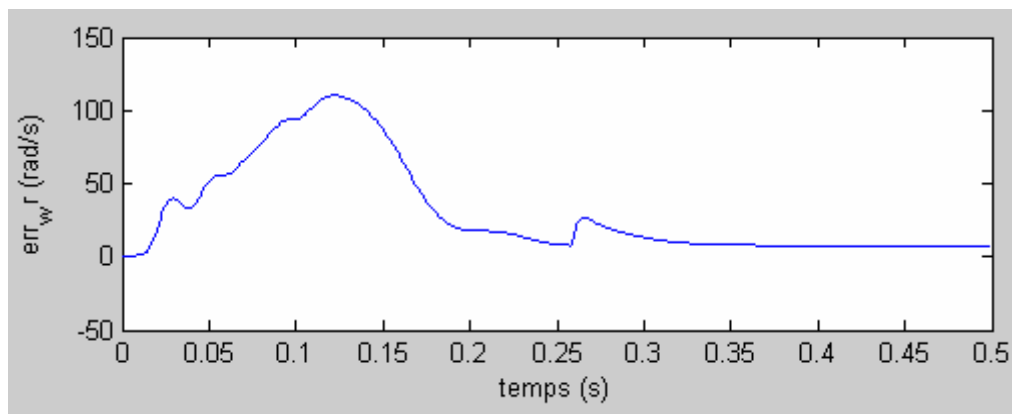


**Figure (3.9) :** Erreurs d'estimation  $\psi_{dr}$  ( .. ) et  $\psi_{qr}$  ( - ).





**Figure (3.10) :** Vitesse rotorique  $w_r$  simulé ( .. )  
et estimé ( - ).



**Figure (3.11) :** Erreurs d'estimation de la vitesse rotorique.

### 3.5. Conclusion

Dans le chapitre précédent nous avons supposé que le modèle de la machine asynchrone est linéaire ce qui rend l'application du filtre de Kalman linéaire valable. Réellement la variation des paramètres de la machine ( $R_r$ ,  $R_s$ ,  $L_r$ ,  $L_s$  et  $w_r$ ) en fonction du temps et ou la température aboutis à un modèle non linéaire, le filtre de Kalman étendu est alors utilisé pour une estimation simultanée du vecteur d'état et des paramètres électriques ou mécaniques de la machine.

Dans ce chapitre nous avons exploité et adopté le filtre de Kalman étendu pour l'estimation des flux rotoriques de la machine ainsi que la vitesse de rotation rotorique. Les

résultats de simulation obtenus présente des erreurs d'estimation remarquable dans le régime transitoire, l'erreur décroît rapidement vers une valeur minimale proche de zéro après établissement du régime permanent. Ces erreurs sont dues essentiellement au choix arbitraire des paramètres initiaux du filtre. Cela présente le problème major rencontré dans cette phase d'estimation.

## *Dispositif expérimental*

### **4.1. Introduction**

La validation pratique des différents algorithmes d'estimation, présentés dans les chapitres précédents, nécessite des tests sur un banc d'essais. Un dispositif expérimental est mis au point autour d'une machine asynchrone à cage alimentée par un onduleur de tension (MLI), des capteurs de courant, des capteurs de tension et des modules électroniques d'adaptation permettent à un calculateur DSP, convenablement choisi, l'acquisition et le traitement des grandeurs électriques de la machine pour l'estimation des flux rotoriques.

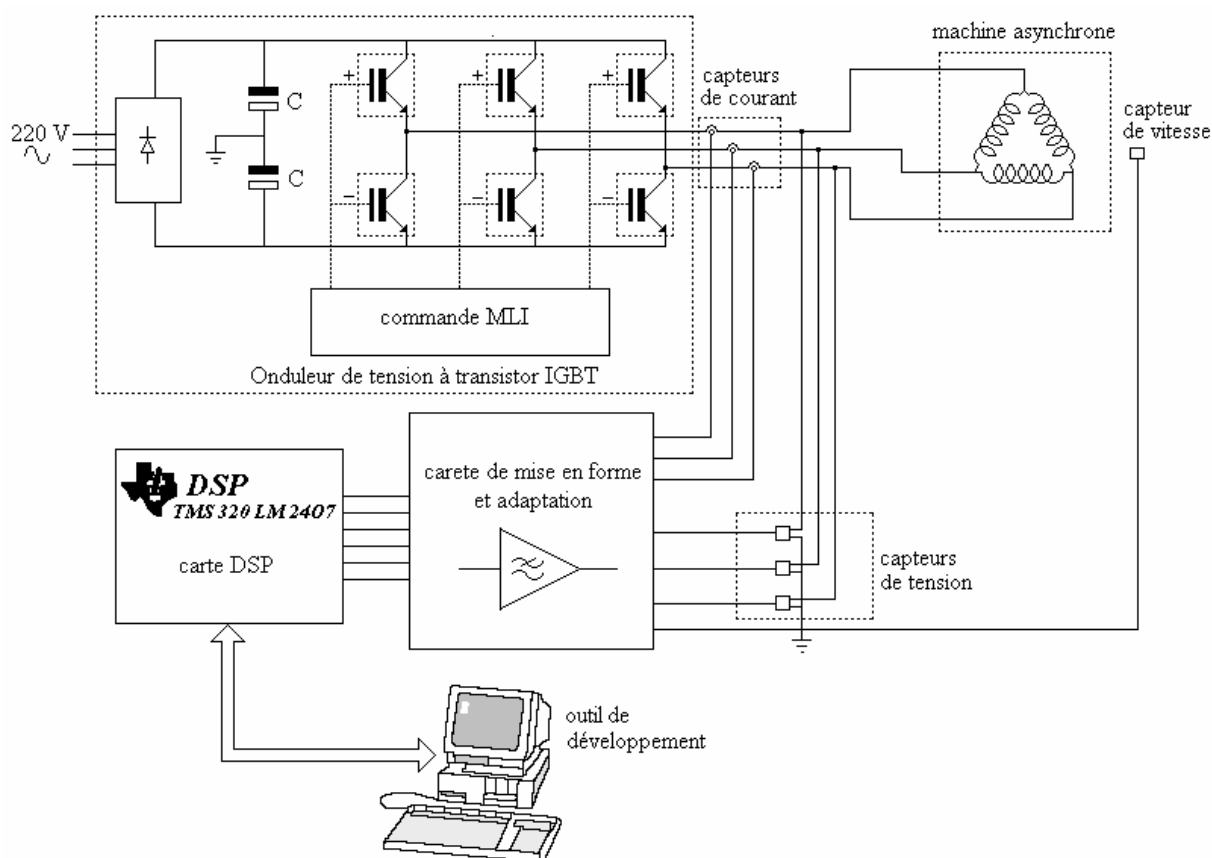
Ce chapitre donne la description du dispositif expérimental réalisé et ces différents modules électroniques et électriques. Ce dispositif permet le développement des algorithmes d'estimation de la machine asynchrone.

### **4.2. Plate-forme expérimentale**

La plate-forme expérimentale utilisée pour valider les algorithmes d'estimation, figure (4.1), est réalisée autour d'un processeur de traitement de signal « DSP », le *TMS320LF2407A* de Texas Instruments. Ce processeur est le résultat d'une amélioration du DSP *TMS320C240* [18]. Le module de développement que nous avons utilisé comme module de calcul, est le *TMS320LF2407 EVM* créé par *SPECTRUM DIGITAL* [19]. Ce module intègre des ports de communication, un convertisseur numérique analogique, des ports d'entrée/sortie dans une seule carte pour former un système d'évaluation destiné aux applications de commande ou de contrôle des machines électriques.

Une machine asynchrone à cage de l'unité de production de *AZAZGA* constitue le système à observer, associée à une génératrice tachymétrique pour la mesure de la vitesse rotorique. Elle est alimentée par un onduleur triphasé fonctionnant en modulation de largeur d'impulsion à base de transistors *IGBT* de la firme *HITACHI JAPAN*. Une carte de capteurs de courant et capteurs de tension et une carte de mise en forme et d'adaptation de ces signaux sont réalisées pour transmettre les grandeurs analogiques à la carte DSP en toute sécurité. Cette

dernière est reliée à un micro-ordinateur contenant un outil de développement software ; Code Composer Studio « CCS » pour la saisie et le débogage des algorithmes destinés aux traitement en temps réel. Nous décrivons par la suite les différents modules du dispositif expérimental.



**Figure (4.1) :** Plate-forme expérimentale.

#### 4.2.1. La machine asynchrone à cage

Le moteur utilisé est une machine asynchrone à cage fabriquée par *Electro-Industriel* de AZAZGA d'une puissance de 3K avec une paire de pôle. Les caractéristiques données par le constructeur sont :

Puissance nominale	: $P_{nom} = 3 \text{ kW}$ .
Fréquence nominale	: $f_{nom} = 50 \text{ Hz}$ .
Nombre de paire de pôles	: $P = 1$ .
Tension d'alimentation	: $V_{nom} = 380 \text{ V}$ .
Courant nominal	: $I_{nom} = 6.4 \text{ A}$ .
Vitesse de rotation nominale	: $N_{nom} = 2895 \text{ tr/min}$ .

Cette machine est associée à une génératrice tachymétrique pour la mesure de vitesse. Les paramètres électriques de la machine sont obtenus suite à des essais de fonctionnement à vide et à rotor bloqué. Les résultats obtenus sont :

Résistance statorique	: $R_s = 5.8500 \Omega$
Résistance rotorique	: $R_r = 3.3650 \Omega$
Inductance cyclique statorique	: $L_s = 0.6578 H$
Inductance cyclique rotorique	
Inductance mutuelle cyclique stator-rotor	: $L_r = 0.6578 H$
Moment d'inertie	: $L_m = 0.6578 H$
Coefficient de frottement	: $J = 2.690 \cdot 10^{-3} \text{ kgm}^2$
	: $f = 0.611 \cdot 10^{-3} \text{ kgm}^2/\text{s}$

#### 4.2.2. L'onduleur triphasé

L'onduleur triphasé qui alimente le stator en tension et courant avec fréquence et amplitude ajustables est un élément important dans la chaîne de variation de vitesse de la machine asynchrone. L'onduleur avec ses composants électroniques de commutation à grande puissance forme un convertisseur de courant continu en courant alternatif [14]. Le principe de conversion se repose sur le découpage de la tension continue en impulsions positives et négatives pour avoir une tension impulsionnelle et alternative.

Il y'a plusieurs techniques pour la commande de l'onduleur, mais la plus connue et la plus utilisée est la modulation de largeur d'impulsion qui applique des tensions aux bases des transistors ( MOSFET, IGBT ) à des temps différents pour produire les signaux de sortie désirés, figure (4.2).

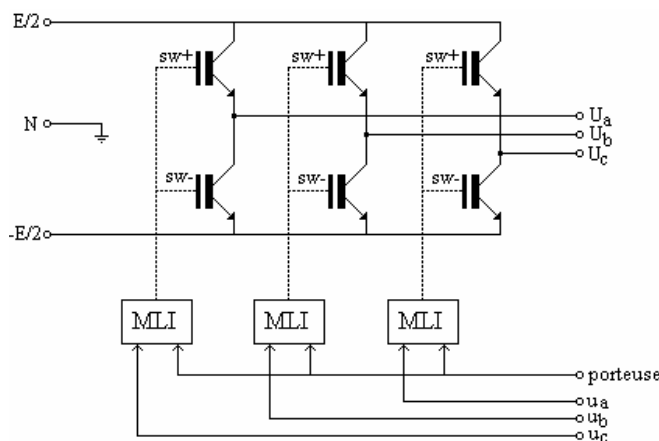
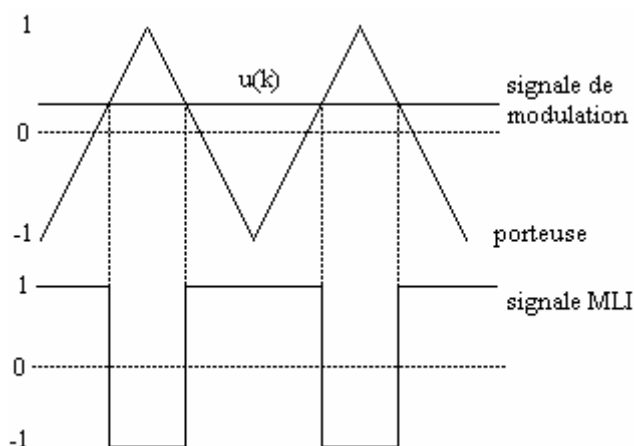


Figure (4.2) : Structure d'un onduleur MLI.

### 4.2.2.1. Principe de fonctionnement de la modulation de largeur d'impulsion

La modulation de largeur d'impulsion se compose de deux signaux ; un signal dit porteuse, généralement triangulaire de fréquence habituellement entre 10 et 20 kHz comparé à un signal sinusoïdal de fréquence variable entre 0 et 50 Hz pour produire des impulsions de largeur variable ; figure (4.3). Les largeurs et les espacements entre les impulsions sont arrangés de sorte que le signal de sortie produit une enveloppe sinusoïdale. La variation de l'amplitude et la fréquence du signal MLI sont assurées par la variation de l'amplitude et la fréquence du signal de modulation sinusoïdal [13]. Les impulsions résultantes de cette comparaison sont conduites par suite à la partie puissance. Cette dernière est composée des transistors de commutation pour délivrer les tensions d'alimentation de la machine électrique utilisée.



**Figure (4.3) :** Principe de génération du signal MLI par Comparaison.

### 4.2.2.2. Modulation MLI a trois phases

La représentation générale des signaux triphasés modulés en MLI est comme suit [15] :

$$U_i(k) = U_i^*(k) + e_i(k) \quad (4.1)$$

Où :

$e_i(k)$  : Est les harmoniques injectées pour chaque phase.

$U_i^*(k)$  : Est appelé le signal fondamental.

Les signaux triphasés de modulation sont :

$$\begin{aligned} U_a^*(k) &= m \cdot \sin(\omega t) \\ U_b^*(k) &= m \cdot \sin\left(\omega t + \frac{2\pi}{3}\right) \\ U_c^*(k) &= m \cdot \sin\left(\omega t + \frac{4\pi}{3}\right) \end{aligned} \quad (4.2)$$

Avec  $m$  : indice de modulation et  $U_a^* + U_b^* + U_c^* = 0$

Les signaux à la sortie de l'onduleur deviennent :

$$\begin{aligned} U_{aN}(t) &= \frac{E}{2} [m \cdot \sin(\omega t) + e_a(t)] \\ U_{bN}(t) &= \frac{E}{2} \left[ m \cdot \sin\left(\omega t + \frac{2\pi}{3}\right) + e_b(t) \right] \\ U_{cN}(t) &= \frac{E}{2} \left[ m \cdot \sin\left(\omega t + \frac{4\pi}{3}\right) + e_c(t) \right] \end{aligned} \quad (4.3)$$

Où :

$E$  : Est la tension d'alimentation continue.

Les tensions de sortie phase à phase sont :

$$\begin{aligned} U_{ab}(t) &= U_{aN} - U_{bN} \\ &= \frac{E}{3} \sqrt{3} m \cdot \sin\left(\omega t + \frac{\pi}{6}\right) \\ U_{bc}(t) &= \frac{E}{3} \sqrt{3} m \cdot \sin\left(\omega t + \frac{5\pi}{6}\right) \\ U_{ca}(t) &= \frac{E}{3} \sqrt{3} m \cdot \sin\left(\omega t + \frac{5\pi}{6}\right) \end{aligned} \quad (4.4)$$

Pour avoir un fonctionnement à vitesse variable de la machine, la tension d'alimentation doit avoir une fréquence variable dans une plage de 0,5 à 50 Hz selon la plaque signalétique de la machine

Par conséquent, La machine utilisée est alimentée par un onduleur de tension triphasé industriel. Nous avons utilisé le « L100 » ; figure (4.4), qui est un onduleur MLI construit par « HITACHI » JAPAN.



**Figure (4.4) :** L'onduleur « L100 » de *HITACHI*.

« L100 » est un onduleur de tension à transistors IGBT commandés en modulation de largeur d'impulsion (MLI). Il délivre une tension triphasée de fréquence allant de 0,5 à 50 Hz avec sortie non filtrée. La variation de fréquence se fait soit par un potentiomètre situé sur la face avant de l'onduleur, soit par une tension variante entre 0 et 10V appliquée à une entrée spécifique accessible.

Les caractéristiques données par le constructeur sont :

Tension d'entrée : 3 Phases, 380 à 460 Volts, 10 A.

Tension de sortie : 3 Phases, 380 à 460 Volts, 7.5 A.

Puissance délivré : 3 KWatt.

Fréquence de la tension de sortie : 0.5 à 50 Hz.

Possibilité de fonctionnement en  $V/F$  constant.

Possibilité de contrôle par PC à travers un port de communication RS 422.

#### **4.2.3. Module de développement TMS 320 LF 2407 EVM**

Le cœur du module de développement LF2407A est un DSP 16 bits à point fixe. C'est le TMS320LF2407A de Texas Instruments, avec 544 mots de mémoire de données internes et une mémoire Flash contenant les instructions à exécuter. La mémoire totale du module est de 128k mots de 16 bits. Les entrées analogiques sont intégrées dans le DSP ainsi que des ports de communication série synchrone et asynchrone [21].



Le module contient aussi un convertisseur numérique-analogique avec quatre sorties et quatre ports d'extension analogiques et numériques pour la communication avec d'autres cartes.

#### **4.2.3.1. Caractéristiques du module de développement TMS320 LF 2407 EVM**

Le module de développement LF2407EVM est caractérisé par :

- DSP TMS320LF2407 travaillant avec une vitesse de (30MIPS) et une mémoire de 128k mots de 16 bits dont la majorité des instructions s'exécutent dans un seul cycle machine.
- 16 entrées analogiques multiplexées avec 10 bits de précision intégrées dans le DSP.
- Deux compteurs d'évènements qui génèrent des signaux MLI pour la commande directe des onduleurs et des entrées pour capteurs numériques de vitesse.
- Un convertisseur numérique-analogique DAC7625 avec quatre sorties analogiques.
- Un contrôleur UART avec port de communication RS232.
- 32K mots de mémoire Flash intégrée.
- Quatre interrupteurs et quatre LEDs utilisateurs.
- Un connecteur d'émulation selon la norme IEEE11491.
- Une alimentation 5V.

#### **4.2.3.2. Schéma fonctionnel du module TMS320 LF 2407 EVM**

La figure (4.5) présente le schéma en bloc de la configuration principale du module TMS320LF2407EVM [19]. La carte inclut des éléments externes comme l'interface analogique, les ports de communication ... etc. Chaque composant logique ou programmable externe est identifié par une adresse binaire sur 16 bits fixée par le constructeur, les autres périphériques internes sont contrôlés par des registres internes (sur DSP) mappés en mémoire de données.

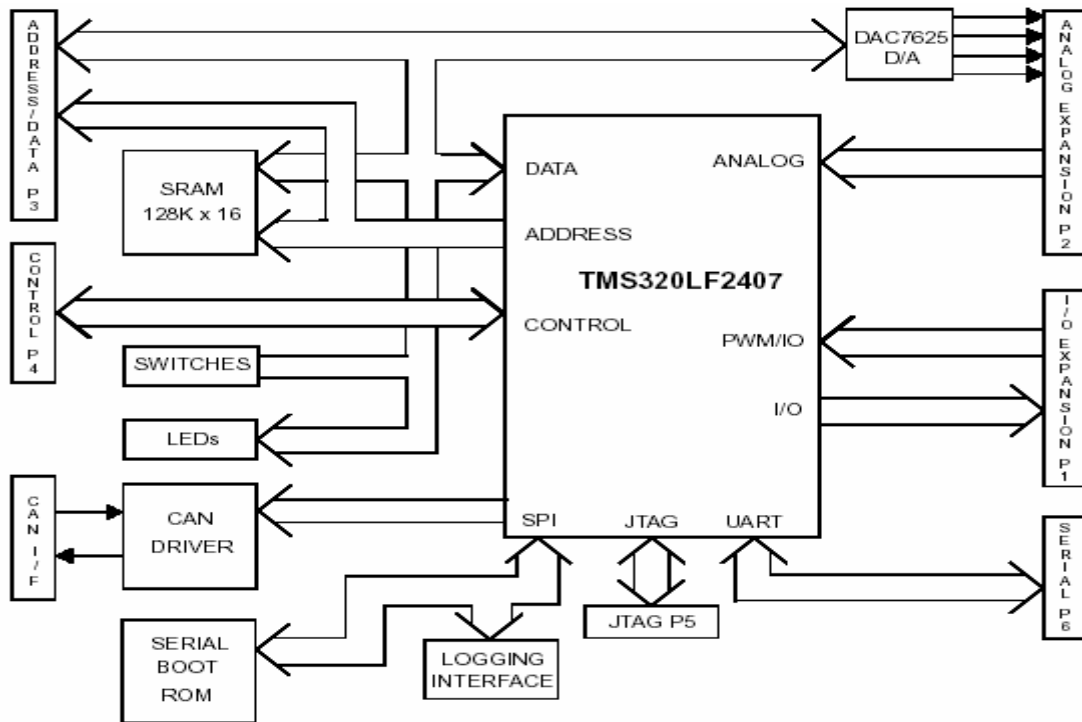


Figure (4.5) : Schéma bloc de la carte DSP.

La figure (4.6) représente une vue de haut de la carte *LF 2407 EVM* alimentée par une tension de *5Volts* et montrant les composants externes associés au DSP.



Figure (4.6) : Vu de haut de la carte DSP.

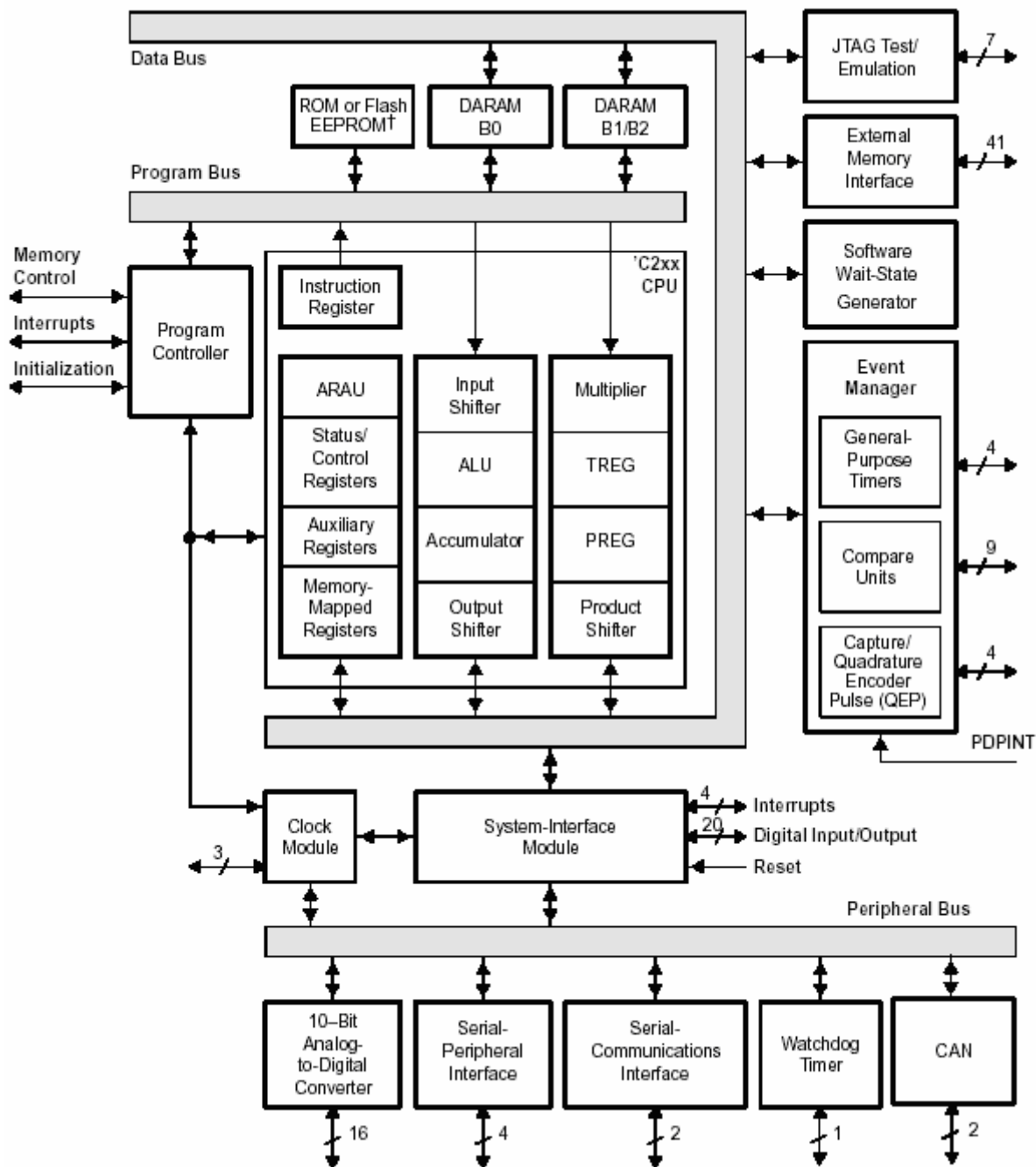
### 4.2.3.3. Architecture interne du DSP TMS 320 LF 2407

Le *DSP TMS320 LF 2407* est conçu spécialement pour traiter une large variante d'applications de contrôle ou de commande numériques des moteurs électriques et pour d'autres applications de commande embarquées [18].

L'architecture interne est basée sur une architecture standard dite « *C2xLP ASIC* » à point fixe, bus internes de 16 bits et à consommation faible complimenté par quelque périphérique intégrée largement utilisé dans le application de commande, des mémoire de programme de type *ROM* ou *FLASH* et en plus des mémoire *RAM* a double accès.

La figure (4.7) représente un schéma fonctionnel de l'architecture interne du *DSP TSM 320 LF 2407* basé sur une architecture Harvard modifié, qui supporte les structures séparées de bus de la mémoire de programme et celui de la mémoire de données.

Un troisième espace, l'espace des entrées/sorties est également disponible et accessible à travers le bus d'interface externe. Pour supporter la sélection d'un grand nombre de périphériques, un bus périphérique est utilisé avec mappage sur la mémoire de données. Il est connecté au bus de données par un module spécial qui offre la possibilité de traiter les registres des périphériques par les mêmes instructions adressant la mémoire de données.



**Figure (4.7) :** Schéma fonctionnel de l'architecture interne du DSP TMS 320 LF 2407.

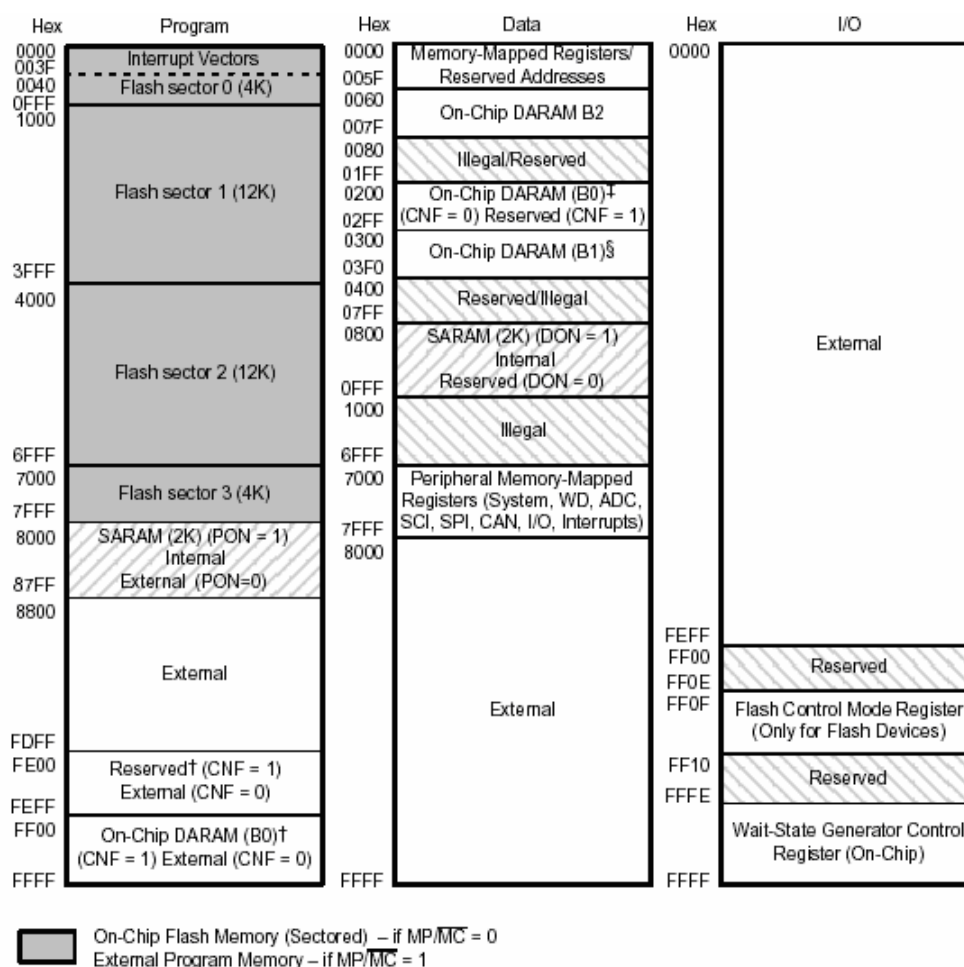
L'architecture de l'unité arithmétique et logique centrale et de l'unité arithmétique des registres auxiliaires est donnée dans l'annexe A.

### 4.2.3.4. Organisation de la mémoire du DSP TMS 320 LF 2407

La mémoire du *DSP TMS 320 LF 2407* est organisée en trois espaces adressables individuellement qui sont :

- **Mémoire de programme ( 64 Kmots de 16 bits )** : contient les instructions à exécuter ainsi que les opérandes immédiates utilisées durant d'exécution du programme.
- **Mémoire de données ( 64 Kmots de 16 bits )** : utilisée pour le stockage des données à manipuler par les instructions.
- **Espace entrées/sorties ( 64 Kmots de 16 bits )** : utilisé pour l'interfaçage avec le périphériques externes ainsi que quelque registres internes du DSP.

Ces trois espaces fournissent un espace total de travail allons jusqu'à 192 Kmots répartis en blocs spécifiques, figure (4.8).

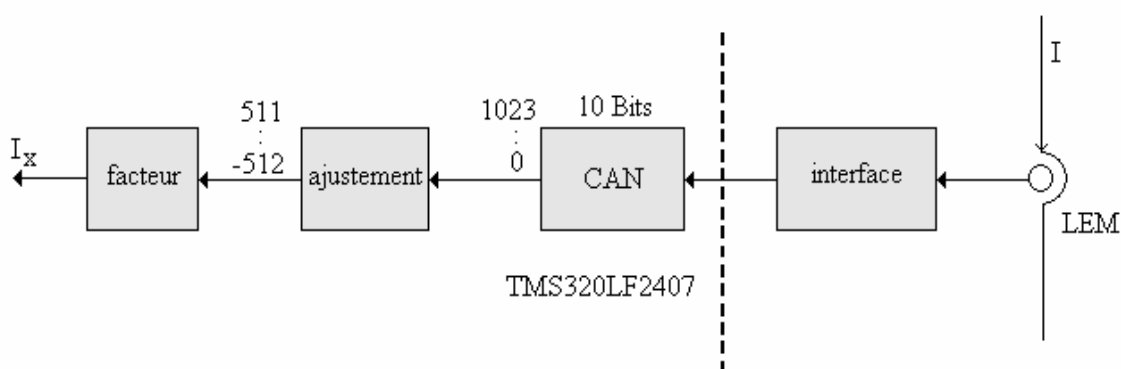


**Figure (4.8) :** Organisation de la mémoire du DSP TMS 320 LF 2407.

#### 4.2.4. Module des capteurs

L'application de filtre de Kalman sur la machine asynchrone exige la mesure des trois tensions des trois phases issues de l'onduleur et des trois courants qui traversent ces phases statoriques de la machine [22].

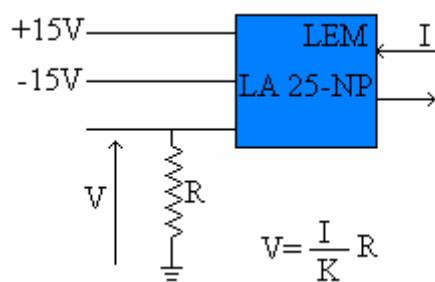
Dans notre application, nous avons utilisé des capteurs de courants et des capteurs de tensions du constructeur « LEM » [26]. Ces capteurs fonctionnent selon le principe de l'effet de Hall ; basé sur la mesure des champs magnétiques résultants du courant électrique. En plus de la mesure des courants et des tensions, les capteurs LEM offrent une autre fonction ; celle de l'isolation entre la partie haute tension et basse tension du système expérimental. Les capteurs sont séparés de la carte DSP par une interface d'adaptation et de mise en forme. La valeur issue de cette dernière doit être convertie en valeur numérique, ajustée puis multipliée par le facteur d'atténuation pour qu'elle soit utilisable par l'algorithme d'estimation. La figure (4.9) nous donne le schéma en bloc d'un canal d'acquisition pour chaque grandeur analogique issue d'un capteur de tension ou de courant.



**Figure (4.9) :** Canal d'acquisition pour chaque capteur.

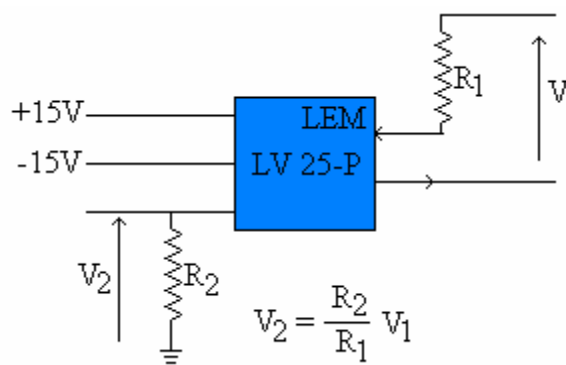
##### 4.2.4.1. Fonctionnement des capteurs utilisés

Les courants qui traversent les phases statoriques sont mesurés par des capteurs de courant LA25-NP de LEM qui peuvent supporter un courant maximal de 36A. La sortie de chaque capteur délivre un courant proportionnel à celui de l'entrée avec un facteur d'atténuation de 1000. Pour avoir une tension à la sortie du capteur, on utilise une résistance entre la masse et sa sortie où la tension sera proportionnelle au courant d'entrée entre les bornes de cette résistance, figure (4.10).



**Figure (4.10)** : Brochage du capteur de courant.

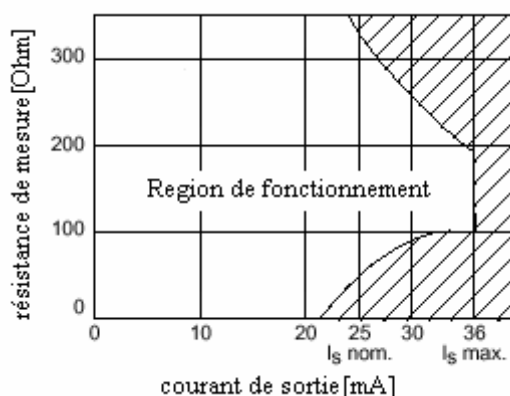
Les tensions des phases statoriques sont mesurées par des capteurs LV 25-P de LEM. Ils ont le même principe de fonctionnement que les LA25-NP, mais avec un courant d'entrée maximal de  $10\text{ mA}$  et un facteur d'atténuation de  $0,4$ . Ils sont reliés avec les phases par une résistance de grande valeur pour limiter le courant maximal d'entrée à  $10\text{ mA}$ , figure (4.11).



**Figure (4.11)** : Brochage du capteur de tension.

#### 4.2.4.2. Choix de la résistance de sortie des capteurs de courant :

La valeur de la résistance de shunt est choisie en fonction du courant d'entrée maximal voulu. Le diagramme courant-résistance donné par le constructeur des capteurs est représenté dans sur La figure (4.12).



**Figure (4.12)** : Courbe courant résistance du capteur de courant LEM.

En respectant les conditions d'utilisation des capteurs décrites précédemment, nous avons réalisé le module « carte capteur » présenté par la figure (4.13) (avec alimentation incluse) capable de délivrer des tensions image proportionnelles aux trois tensions et aux trois courants réels de la machine asynchrone.

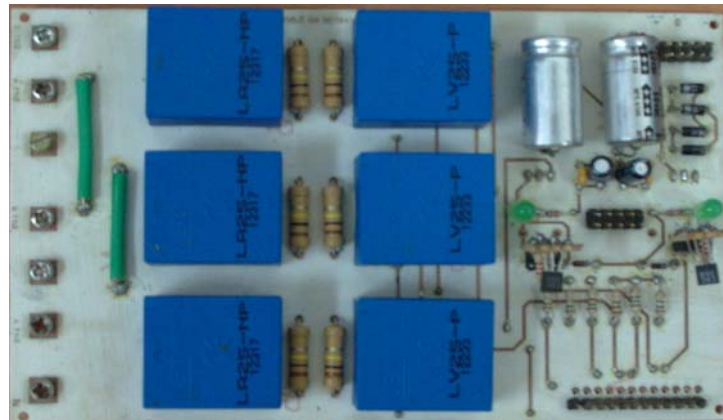


Figure (4.13) : Vu de haut de la carte capteur.

#### 4.2.5. Carte d'adaptation et de filtrage :

Les signaux issus des capteurs de courant et de tension contiennent des harmoniques indésirables dû essentiellement au fonctionnement en commutation (MLI) des transistors IGBT de l'onduleurs [23], figure (4.14) *a* et *b*. Ces harmoniques influent sur l'efficacité et sur les performances de l'algorithme d'estimation [10].

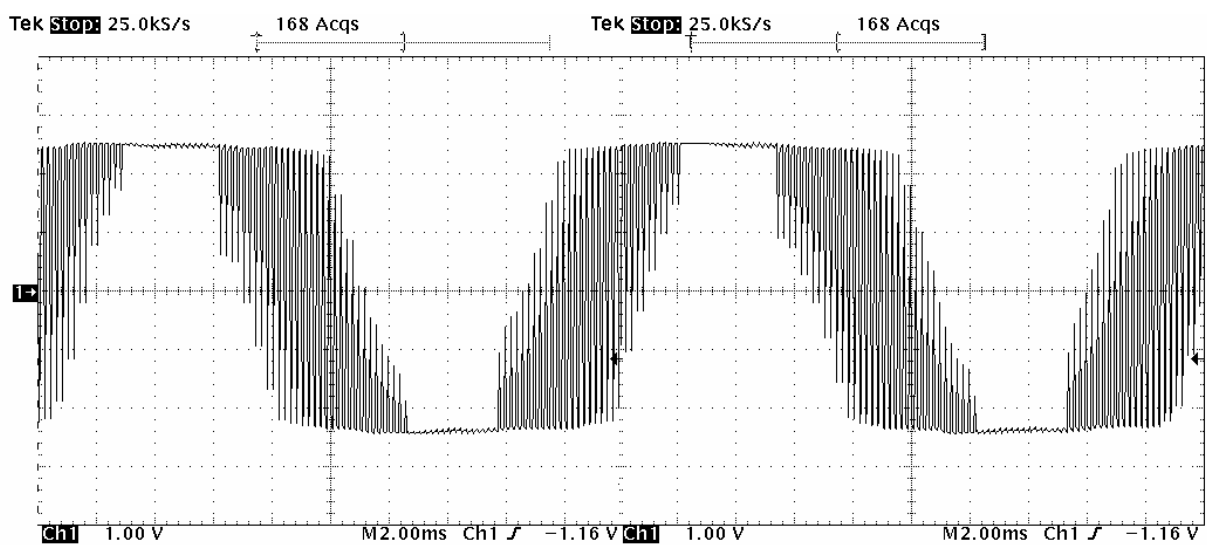


Figure (4.14)-a : Tension.



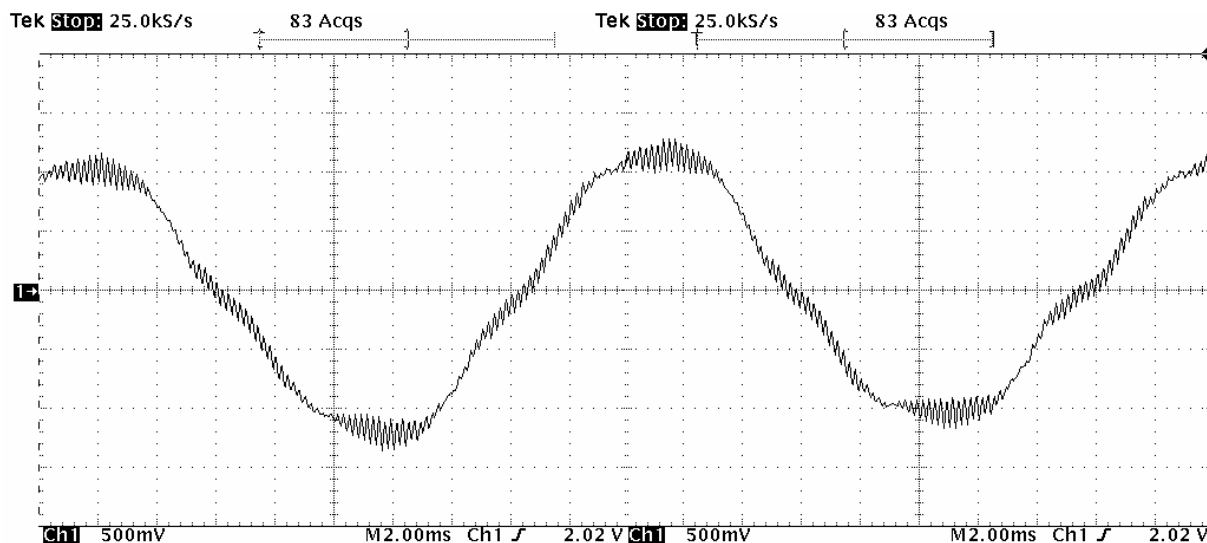
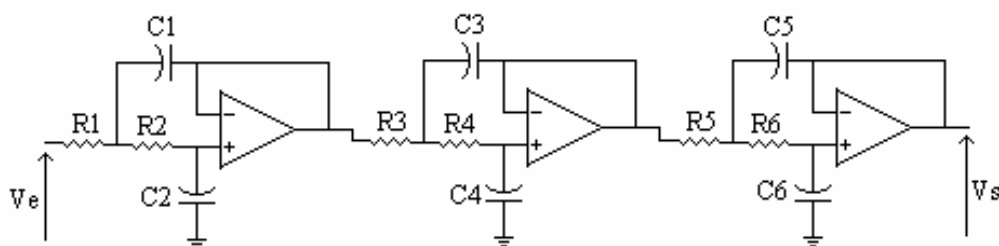


Figure (4.14)-b : Courant.

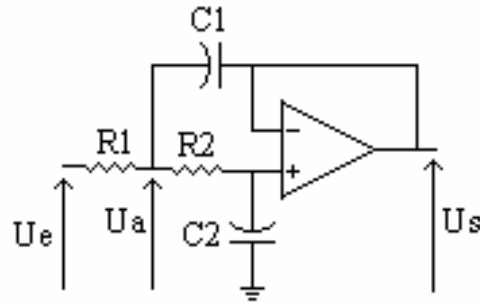
**Figure (4.14)** : Tensions et courants de sortie de l'onduleur

Nous avons utilisé un filtre passe bas pour chaque grandeur dans l'objectif de réduire les harmoniques de haute fréquence. La fréquence de coupure de ce filtre doit être supérieure à la fréquence maximale de la tension et du courant statorique (50 Hz). Les filtres passe bas choisis pour accomplir la tâche de mise en forme sont développés par l'approximation de Butterworth ayant une fréquence de coupure de 150 Hz. Ces filtres sont d'ordre 6 réalisés selon la structure de Sallen et Key [24], figure (4.15), en utilisant les amplificateurs opérationnels.

**Figure (4.15)** : Structure de Filtre passe bas.

#### 4.2.5.1. Fonction de transfert du filtre

Le filtre représenté sur la figure (4.15) est constitué de trois cellules biquadratiques en cascade. Pour le calcul de la fonction de transfert, on se limite à étudier une seule cellule, figure (4.16) puis on généralise sur tout le filtre.



**Figure (4.16) :** Cellule biquadratique de Sallen et Key.

On a :

$$F(p) = \frac{U_s(p)}{U_e(p)} \quad (4.5)$$

$$U_a(p) = \frac{U_e(p) \frac{1}{C_1 p} + U_s(p) R_1}{R_1 + \frac{1}{C_1 p}} \quad (4.6)$$

Par simplification on obtient :

$$U_a(p) = \frac{U_e(p) + C_1 R_1 p U_s(p)}{C_1 R_1 p + 1} \quad (4.7)$$

On a aussi :  $e_- = e_+$

Et :  $e_- = U_s$

$$e_- = \frac{\frac{1}{C_2 p}}{R_2 + \frac{1}{C_2 p}} U_a(p) = \frac{1}{C_2 R_2 p + 1} U_a(p) \quad (4.8)$$

alors :

$$U_s(p) = \frac{1}{C_2 R_2 p + 1} U_a(p) \quad (4.9)$$

$$U_a(p) = (C_2 R_2 p + 1) U_s(p) \quad (4.10)$$

donc :

$$\frac{U_e(p) + C_1 R_1 U_s(p)}{C_1 R_1 p + 1} = (C_2 R_2 p + 1) U_s(p) \quad (4.11)$$

On a :  $R_1 = R_2 = R$  ; donc la fonction de transfert devient :

$$F(p) = \frac{U_s(p)}{U_e(p)} = \frac{I}{C_1 C_2 R^2 p^2 + 2C_2 R p + I} \quad (4.12)$$

Avec :  $\omega_0 = \frac{I}{\sqrt{C_1 C_2 R^2}}$  : La pulsation de coupure du filtre.

#### 4.2.5.2. Calcul des valeurs des composants

D'après le tableau des valeurs normalisées de Butter Worth [25], on a les fonctions de transfert des trois cellules :

$$F_1(p) = \frac{I}{p^2 + 1.9318 \cdot P + I} \quad (4.13)$$

$$F_2(p) = \frac{I}{p^2 + 1.4142 \cdot P + I} \quad (4.14)$$

$$F_3(p) = \frac{I}{p^2 + 0.5176 \cdot P + I} \quad (4.15)$$

Première cellule : on a  $F_0 = 150 \text{ Hz} \Rightarrow \omega_0 942.4777 \text{ rad/s}$

dénormalisation :  $\frac{p}{\omega_0} \rightarrow p$

$F_1(p)$  Devient :  $F_1(p) = \frac{I}{1.12579 \cdot 10^{-6} + 2.0497 \cdot 10^{-3} p + I}$

Alors :

$$C_1 C_2 R^2 = 1.12579 \cdot 10^{-6}$$

Et :

$$2 C_2 R = 2.0497 \cdot 10^{-3}$$

Si on donne une valeur arbitraire pour la résistance  $R$  ; par exemple  $33 \text{ k}\Omega$  , on obtient les valeurs des capacités  $C_1 = 33 \text{ nf}$  et  $C_2 = 33 \text{ nf}$ . Par la même méthode, on obtient les valeurs des autres composants :

2<sup>ème</sup> cellule :

$$R = 33 \text{ k}\Omega, \quad C_1 = 47 \text{ nf}, \quad C_2 = 22 \text{ nf}$$

3<sup>ème</sup> cellule :

$$R = 13 \text{ k}\Omega, \quad C_1 = 330 \text{ nf}, \quad C_2 = 22 \text{ nf}$$

La fonction de transfert globale du filtre est le produit des trois fonctions de transfert des trois cellules en cascade. Après le calcul on obtient la fonction simplifiée suivante :

$$F(p) = \frac{1}{1.427 \cdot 10^{-18} p^6 + 5.195 \cdot 10^{-15} p^5 + 9.459 \cdot 10^{-12} p^4 + 1.092 \cdot 10^{-8} p^3 + 8.402 \cdot 10^{-6} p^2 + 0.0049 p + 1}$$

La figure (4.17) représente le diagramme de *Bode* de la fonction de transfert du filtre réalisé :

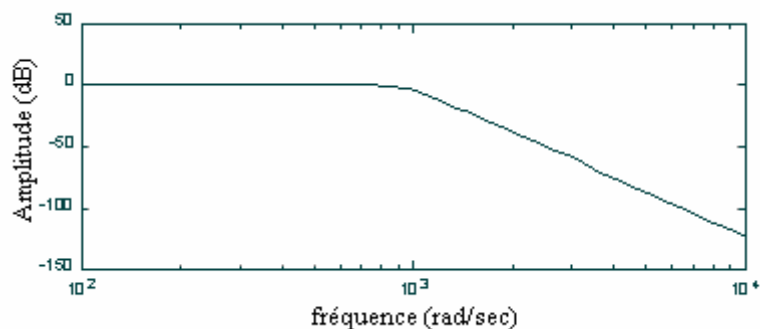


Figure (4.17) : **Diagramme de Bode de la fonction de transfert Du filtre**

#### 4.2.5.3. Test des filtres

Pour confirmer la validité de la structure des filtres choisi et de leurs fonctions de transfert calculées, la figure (4.18) *a* et *b* représente les signaux à l'entrée et à la sortie du filtre avec le spectre en fréquence de chaque signal. Nous constatons que les signaux de sortie ; tension ou courant, sont nettement améliorés.

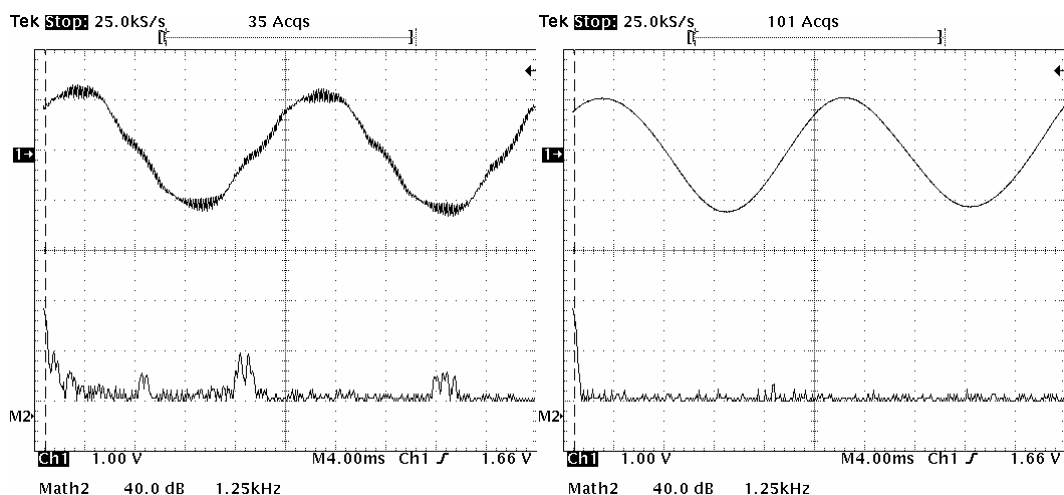


Figure (4.18)-a : courant

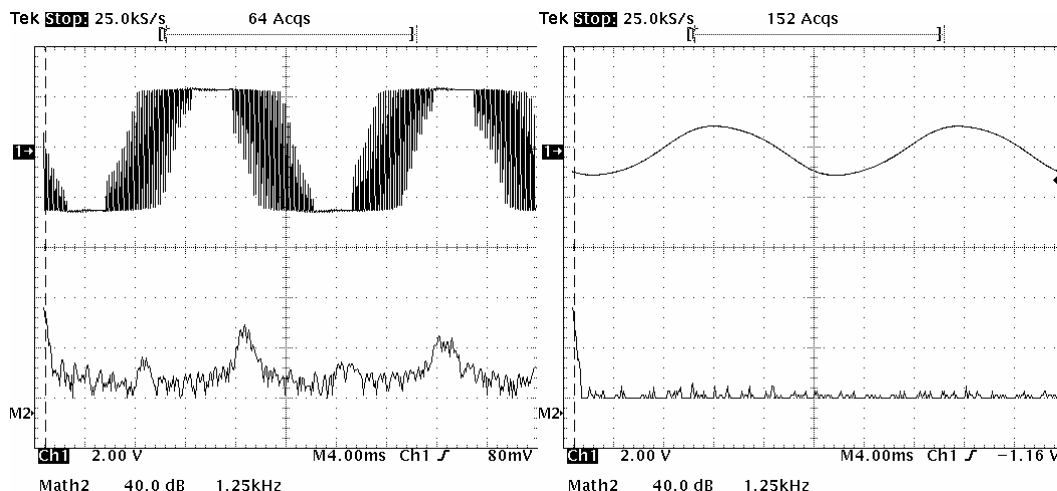


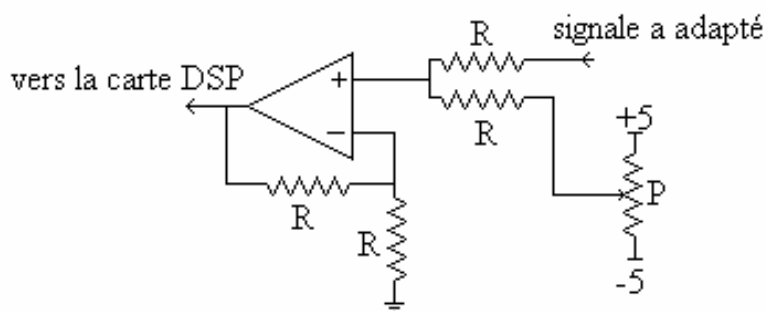
Figure (4.18)-b : Tension.

**Figure (4.18)** : Représentation des signaux d’entrée et de sortie des filtres réalisés.

#### 4.2.5.4. Adaptation des tensions

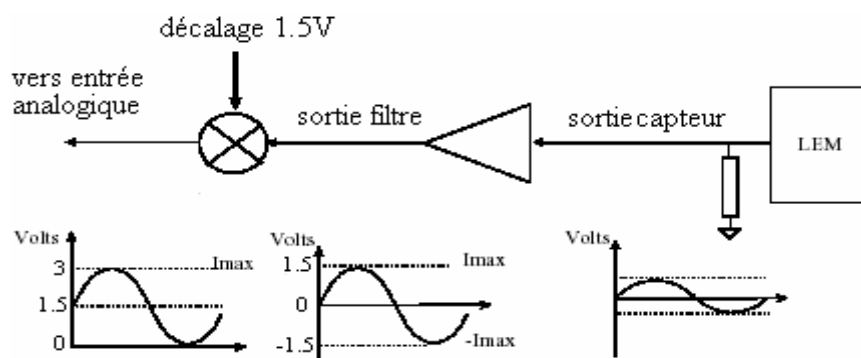
Les capteurs LEM délivrent des tensions positives ou négatives selon la direction des courants d’entrée. L’entrée analogique de la carte DSP accepte des tensions variant entre 0 et 3,3 V, donc un ajustement d’échelle est imposé.

Pour transformer les tensions bipolaires en tensions unipolaires on ajoute une composante continue de 1,5 Volts. Les signaux à la sortie des filtres se trouvent limités dans une plage de tension allant 0 à 3 Volts, figure (4.19).



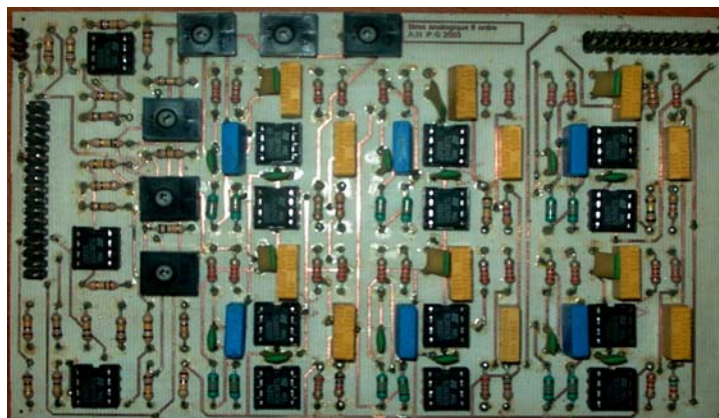
**Figure (4.19)** : Schéma électrique de l’adaptateur des signaux.

Les niveaux maximaux des tensions à partir des capteurs jusqu’aux carte DSP sont représentés par la figure (4.20).



**Figure (4.20) :** Mise en forme et adaptation des signaux.

La carte d'ajustement et mise en forme tel que nous l'avons réalisée est présentée par la figure suivante ; figure (4.21).



**Figure (4.21) :** Carte de mise en forme et adaptation.

#### 4.2.6. Mesure de la vitesse

La carte DSP dispose d'une entrée capteur destinée pour la mesure numérique de la vitesse avec un capteur optique *LASER* à disque perforé [22]. La précision de mesure nécessite un capteur à disque de 5000 ou 10000 trous. Malheureusement ce dernier n'est pas disponible. Cependant, ce capteur est remplacé par une génératrice tachymétrique. Elle est constituée par un moteur à courant continu, qui transforme l'énergie mécanique en énergie électrique traduite par une tension proportionnelle à la vitesse de rotation. Cette génératrice délivre 1Volt pour 1000 tours par minute.

### 4.3. Ajustement des grandeurs mesurées

La plage de tension autorisée par les entrées analogiques de la carte DSP est de 0 à 3,3 Volts. Les courants et les tensions de la machine asynchrone issues des capteurs, constituant les entrées analogique de la carte DSP doivent être ajustés ( amplifier ou atténuer ) pour une acquisition correcte par le processeur de traitement ; figure (4.20).

#### 4.3.1. Ajustement des tensions

Avec une résistance d'entrée de  $50\text{ k}\Omega$  et une résistance de sortie de  $150\ \Omega$ , une tension de  $300\text{ V}$  à l'entrée du capteur devient  $2,25\text{ Volts}$  à la sortie du même capteur. Donc on aura un signal de sortie varié entre  $-2,25\text{ V}$  et  $2,25\text{ V}$ . Celui-là est une image d'un autre signal variant entre  $-300\text{ V}$  et  $300\text{ V}$  avec un rapport d'atténuation de  $133,3$ .

L'ajustement est fait par un ajout de  $1,5\text{ V}$  à la tension de sortie du filtre. La correction est faite donc par la soustraction d'une valeur numérique de (  $465$  ) du résultat de conversion de chaque tension. La tension réelle à utiliser par l'algorithme de filtre de Kalman sera calculée par la relation suivante :

$$V_i = \left( \frac{v_x}{64} - 465 \right) \times 1.0909 \quad (4.16)$$

Avec :

$v_x$  : Est la valeur numérique de la conversion donnée par le convertisseur A/N sur  $16\text{ bits}$ .

$i$  : C'est l'indice correspond aux nombre de phase d'alimentation de la machine.

#### 4.3.2. Ajustement et calibrage des courants

Le courant maximal supporté par un capteur de type LA 25-NP de LEM est  $36\text{ A}$ . cette valeur est traduite par une valeur de tension égale à  $5,4\text{ V}$  avec une résistance de sortie de  $150\ \Omega$ , c'est à dire ; un courant entre  $-36\text{ A}$  et  $36\text{ A}$  se traduit par une tension entre  $-5,4$  et  $5,4$  et un rapport d'atténuation est  $6.66$ . Par une autre atténuation de  $3,6$  par un diviseur de tension on obtient un signal variant entre  $-1,5\text{ Volts}$  et  $1,5\text{ Volts}$ . Le courant réel à utiliser dans l'application, sera calculé par la relation suivante :

$$I_i = \left( \frac{i_x}{64} - 465 \right) \times 0.02146 \quad (4.17)$$

Avec :

$i_x$  : Est la valeur numérique de conversion donnée par le convertisseur A/N sur 16 bits.

$i$  : C'est l'indice correspond au nombre de phases d'alimentation de la machine.

### 4.3.3. Calibrage de la valeur de la vitesse rotorique

La génératrice tachymétrique délivre 1 Volts pour 1000 tours/min ou bien 1V pour 104.7 rad/s, cette tension est conduite directement à l'entrée analogique du DSP. Le résultat de conversion sera présenté sur 10 bits et la pulsation rotorique réel à utiliser dans l'estimation est donnée par la relation suivante :

$$\omega_r = w \times 0.02107 \quad (4.18)$$

Avec  $w$  : La valeur numérique de conversion donné par le convertisseur A/N sur 16 bits.

## 4.4. Validation de la plate-forme expérimental

La plat-forme expérimentale réalisée autour du DSP TMS 320 LF 2407, figure (4.22), est dessinée dans notre travail pour implantation des algorithmes d'estimation ou contrôle en temps réel des machines électriques alimentées par un onduleur.

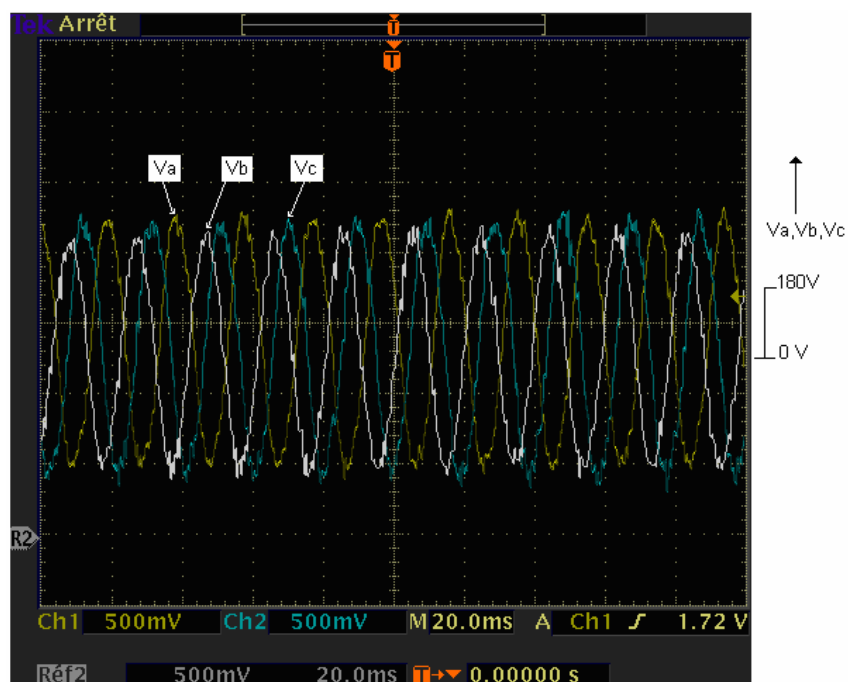
Avant d'entamé le développement de notre application nous avons effectué quelques testes de validation pour voir si les trois tensions et les trois courants de fonctionnement de la machine asynchrone arrivent correctement aux entrées analogique de la carte DSP avec la mise en forme et l'ajustement nécessaire.

Les résultats de teste obtenus sont les trois tensions d'alimentation statorique, figure (4.23) et les trois courants statoriques, figure (4.24) mesurés aux sorties des étages d'adaptations par un oscilloscope numérique à mémoire.





**Figure (4.22) :** Vue d'ensemble de la plate-forme expérimentale réalisée.



**Figure (4.23) :** Tensions d'alimentation statoriques de la machine asynchrone.

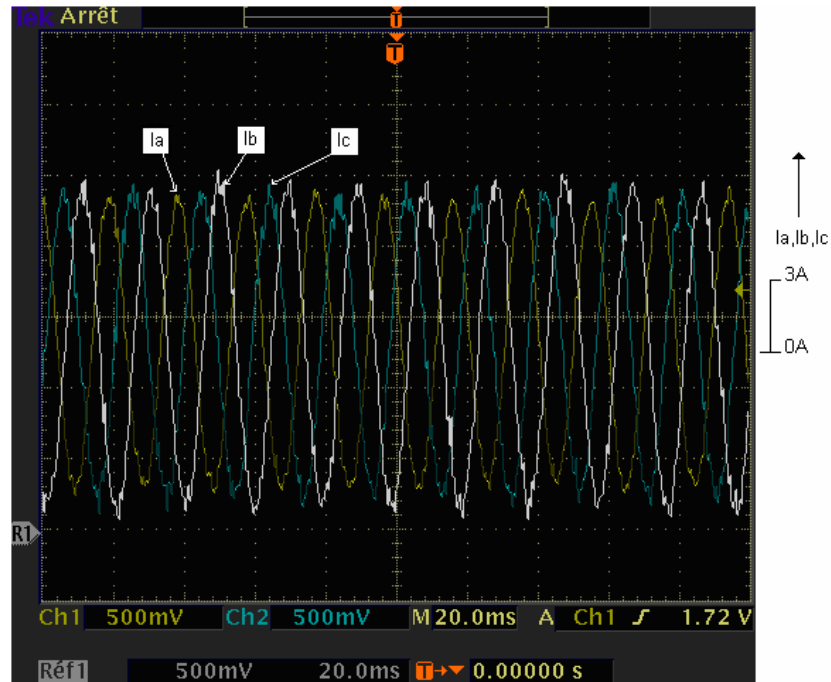


Figure (4.24) : Courants statoriques de la machine asynchrone.

#### 4.5. Conclusion

Dans ce chapitre nous avons présenté le dispositif expérimental mis en œuvre et ces différents modules réalisés :

- Module des capteurs et d'isolation.
- Module de filtrage, d'adaptation et d'ajustement des signaux.

Les valeurs des grandeurs électriques et mécaniques mesurées via ces modules sont correctes. Les signaux issus de ces modules constituent les entrées analogiques de la carte DSP TMS 320 LF 2407. L'ajustement des signaux à permet de définir des facteur de multiplication ou d'atténuation permettant le passage, dans le programme, des signaux acquis par la carte aux valeurs réelles des grandeurs électriques et mécaniques mises en jeux. En conclusion le dispositif expérimental développer est prêt pour l'implantation des algorithmes souhaités.

## ***Implantation des algorithmes d'estimation des flux rotoriques par le filtre de Kalman linéaire***

### **5.1. Introduction**

La machine asynchrone a été préférée pour une grande variété d'applications industrielles à cause de sa nature robuste et simplicité de commande. Dans ces applications, le moteur est directement couplé aux réseaux électriques ou contrôlé par la méthode  $V/f$  ( Rapport tension fréquence ) constant.

L'apparition des processeurs spécialisés de traitement tel que le *TMS320LF2407* et la modélisation dynamique de la machine asynchrone avec l'évolution de l'électronique de puissance ont permit le développement de systèmes à haute fiabilité pour la commande précise de la vitesse et du couple électromagnétique de la machine.

Dans ce chapitre, on va aborder l'implantation en temps réel du filtre de Kalman linéaire sur le *DSP TMS 320 LF 2407* pour estimer les flux rotoriques de la machine asynchrone. Nous commençons, tout d'abord par une présentation succincte sur l'outil de développement Software « Code Composer Studio » pour la mise en œuvre de l'application autour du *DSP TMS 320 LF 2407*.

### **5.2. Outil de développement Code Composer Studio « CCS »**

L'outil de développement « CCS C2000 » est un logiciel crée par *Texas Instrument* pour faciliter la tache de création et de mise en œuvre d'applications sur les *DSPs* de la famille C2xxx. Il intègre l'ensemble des utilitaires software pour la génération des codes avec des possibilités d'analyse et débogage des programmes. Le cycle de développement d'une application doit passer par quatre phases [28] :

**1- Conception ou phase design :**

C'est la phase préliminaire de création et de planification de projet.

**2- Code et compilation :**

Création de projet et écriture du code source en langage C ou en Assembleur des programmes et configuration des fichiers.

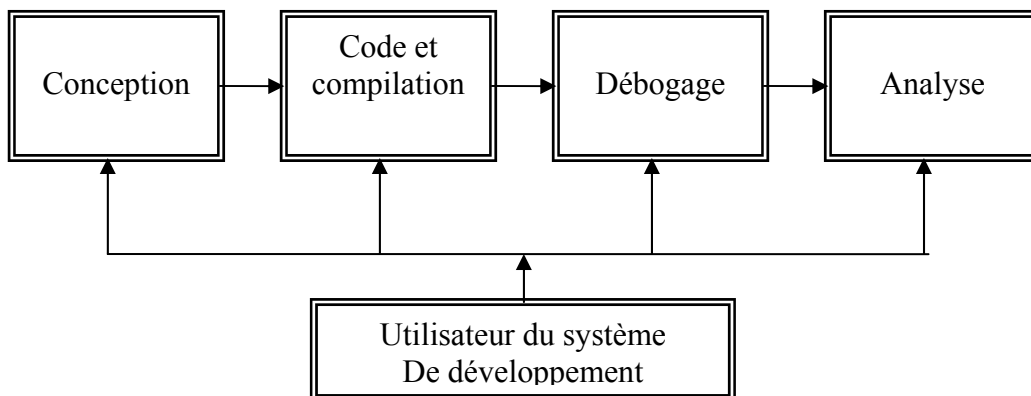
**3- Déboguage :**

Phase de détection des erreurs, donnant la possibilité d'exécution avec des points d'arrêt en mode trace.

**4- Analyse :**

Possibilité d'exécution du programme en temps réel avec affichage des courbes.

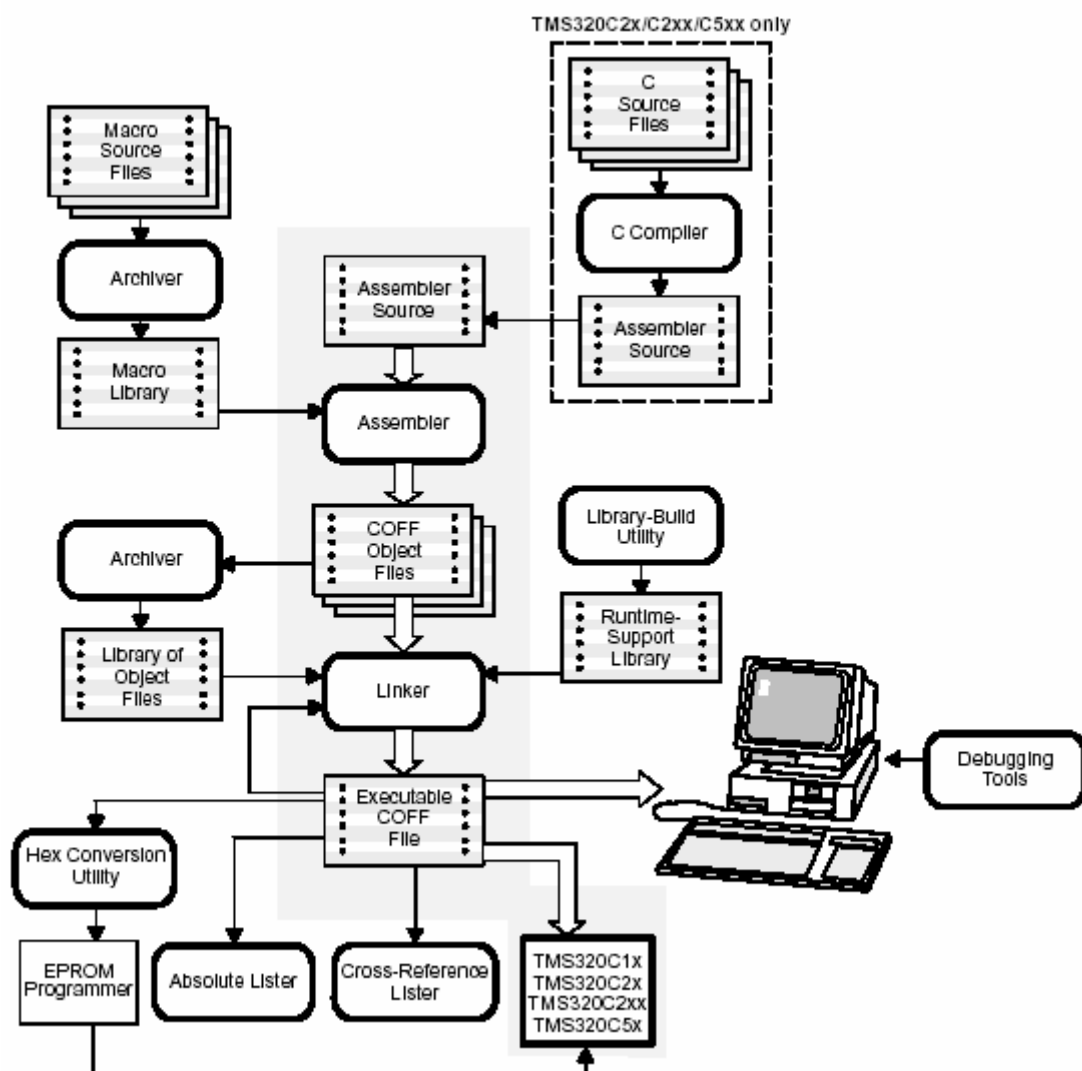
La figure (5.1) résume les différentes phases de développement :



**Figure (5.1) :** Etapes de développement d'une application sur DSP sous CCS.

**5.2.1. Utilitaire de génération de code :**

Les utilitaires de génération des codes constituent la base de l'environnement du développement des programmes sous « CCS ». Une application sous CCS peut être développés en langage C selon l'ouvrage [27] ou en assembleur caractérisant chaque processeur *DSP* de la famille C2xxx. Un développement typique d'une application est présenté sur l'organigramme de la figure (5.2).



**Figure (5.2) :** Déroulement typique de génération de code exécutable d'une application sur DSP sous CCS.

### 5.2.2. Emulation hardware et échange de données en temps réel

Les *DSPs* de *Texas Instrument* comporte un support d'émulation intégré permettant au « CCS » de contrôler l'exécution des programmes en temps réel. Cette émulation intégrée fournit des différentes possibilités hardware tel que :

- Démarrage, arrêt et remise a zéro du DSP.
- Chargement des codes depuis ou vers le DSP.
- Consultation des registres ou de la mémoire du DSP.

- Fixation des points d'arrêt dans le programme exécutable.
- Plusieurs possibilités de comptage.
- Chargement de données en temps réel ( RTDX ) entre Host et DSP.

Le ( RTDX , Real Time Data eXchange ) permet au utilisateur du système de transférer les données entre le micro-ordinateur et le dispositif « DSP » sans arrêt de l'application en cours d'exécution sur le DSP [28].

### 5.2.3. Exemples d'application

Nous avons commencé par une application simple qui consiste à acquérir les sept signaux analogiques de la machine électrique, trois tension et trois courants. Les grandeurs acquises sont représentées sur la figure (5.3).

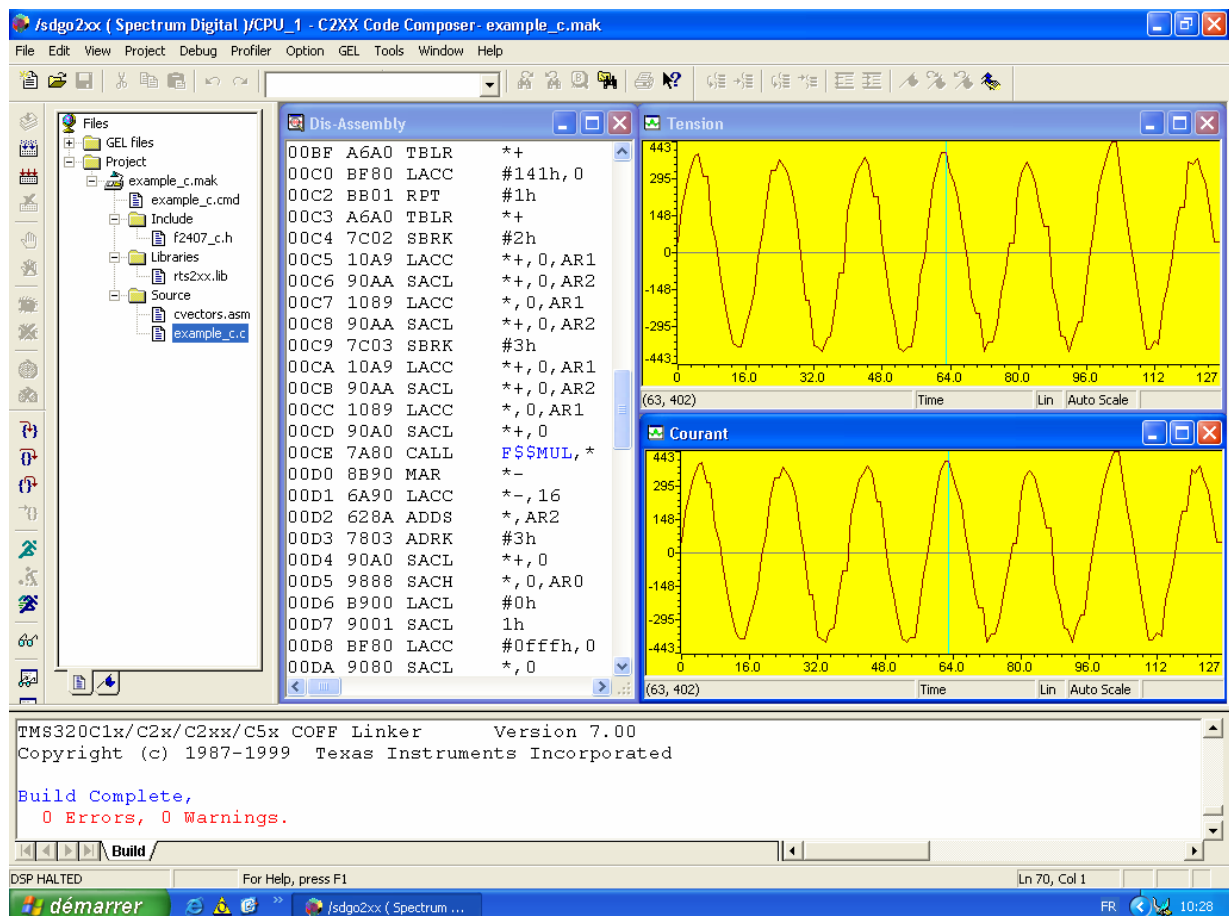
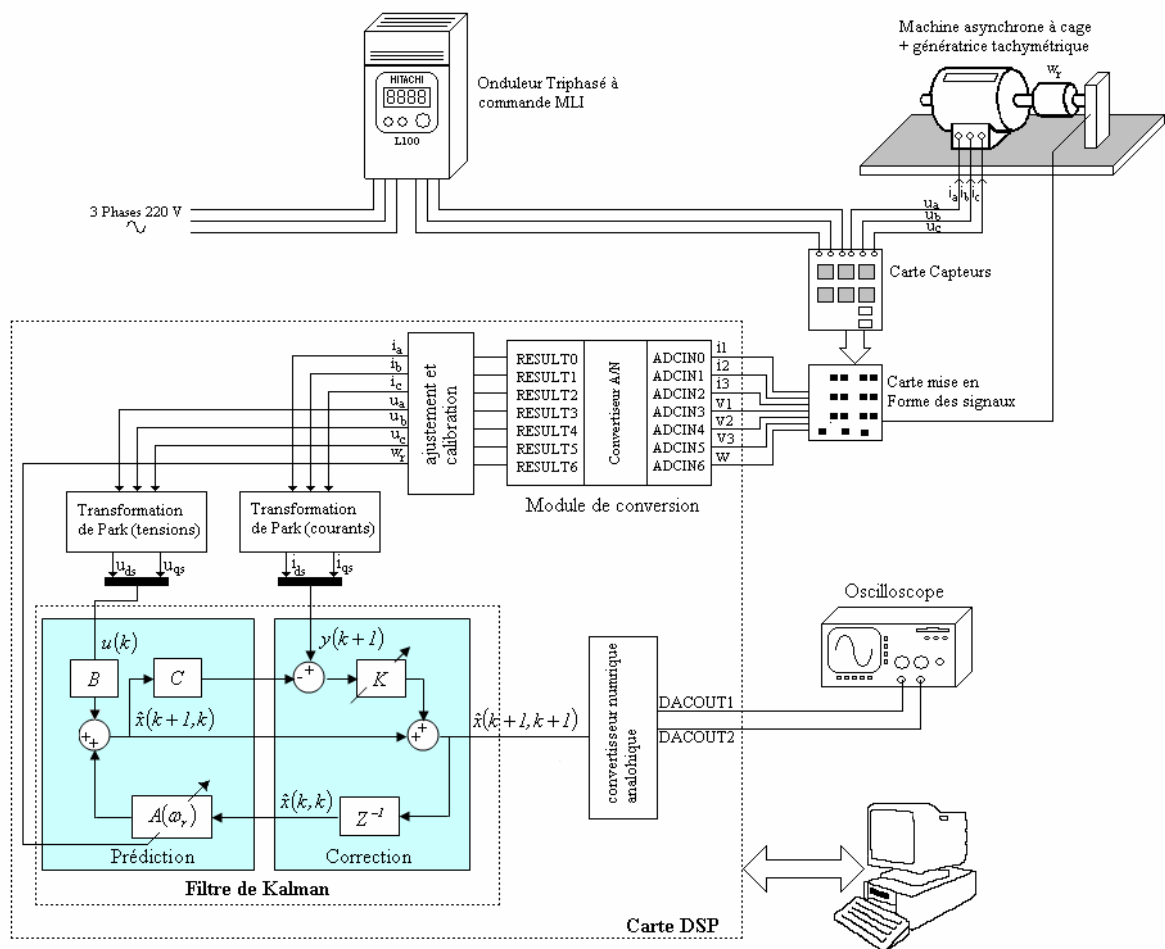


Figure (5.3) : Exemple d'application sur CCS C2000.

**5.3. Schéma d'implantation du filtre de Kalman appliqué à la machine asynchrone**

Dans notre démarche, on a besoin d'acquérir six grandeurs électriques et une grandeur mécanique ; figure (5.4) : les trois courants statoriques de la machine, les trois tensions d'alimentation et la vitesse rotorique de la machine asynchrone. La carte DSP dispose de 16 entrées analogiques multiplexées avec une plage de conversion allant de 0 et 3,3 Volts. Les résultats de conversion délivrer par le convertisseur analogique numérique est sur 10 bits.



**Figure (5.4) :** Schéma de mise en œuvre de l'implantation.

Les sept grandeurs à mesurer sont affectées à 7 entrées analogiques de la carte DSP de ADCIN0 à ADCIN6 où les entrées ADCIN0 à ADCIN2 correspondent aux trois courants statoriques, ADCIN3 à ADCIN5 aux tensions d'alimentation statoriques et ADCIN6 pour la vitesse de rotorique.

On a utilisé aussi les sorties DACOUT1 et DACOUT2 pour générer des tensions proportionnelles au flux estimés ou aux autre grandeurs voulus. Ces tensions seront visualiser sur un oscilloscope.

#### 5.4. Organigramme d'estimation

L'algorithme développé peut se divisé en deux parties : phase d'initialisation et phase d'estimation, figure (5.5) et figure (5.6) respectivement. L'initialisation constitue la configuration du DSP et l'initialisation des variables.

La partie estimation contient la section d'acquisition de données analogiques avec l'ajustement et le calibrage pour la section d'estimation, Les équations du filtre de Kalman et en fin la section conversion numérique-analogique où on traduit les valeurs numériques résultantes de l'estimation et calculer par des valeurs analogiques à visualiser sur oscilloscope. Cette dernière partie se termine par une instruction de retour à la boucle d'attente. La partie estimation se déclanche par une interruption délivré par le temporisateur ( TIMER ) utilisé pour le choix de la période d'échantillonnage.

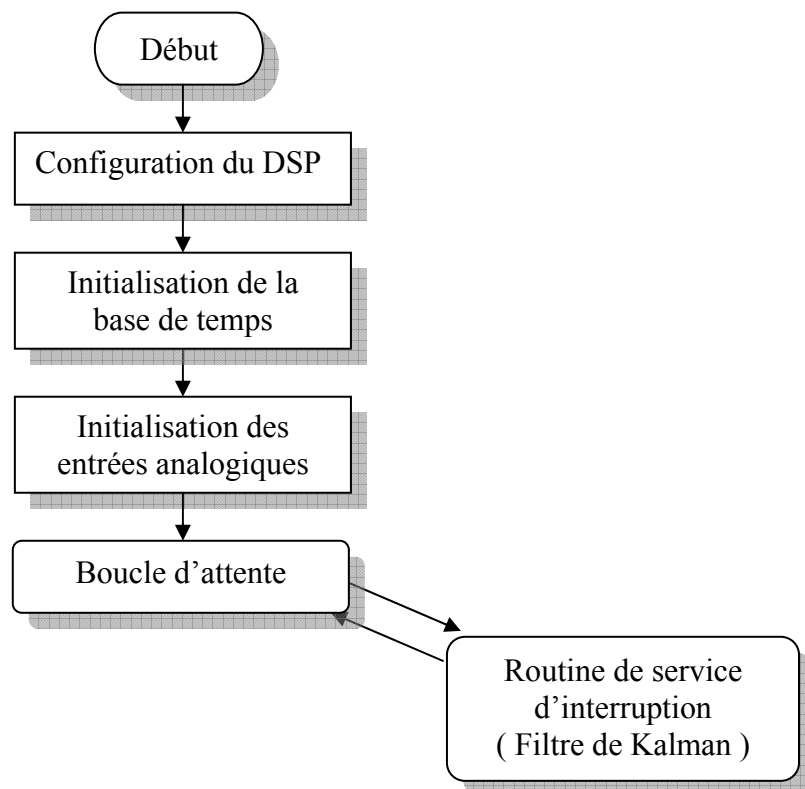


Figure (5.5) : Organigramme d'initialisation.



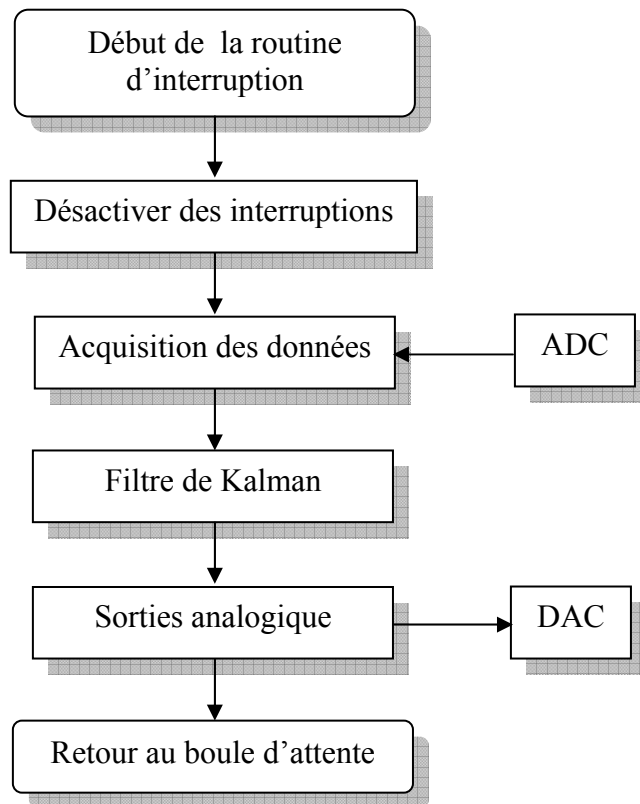
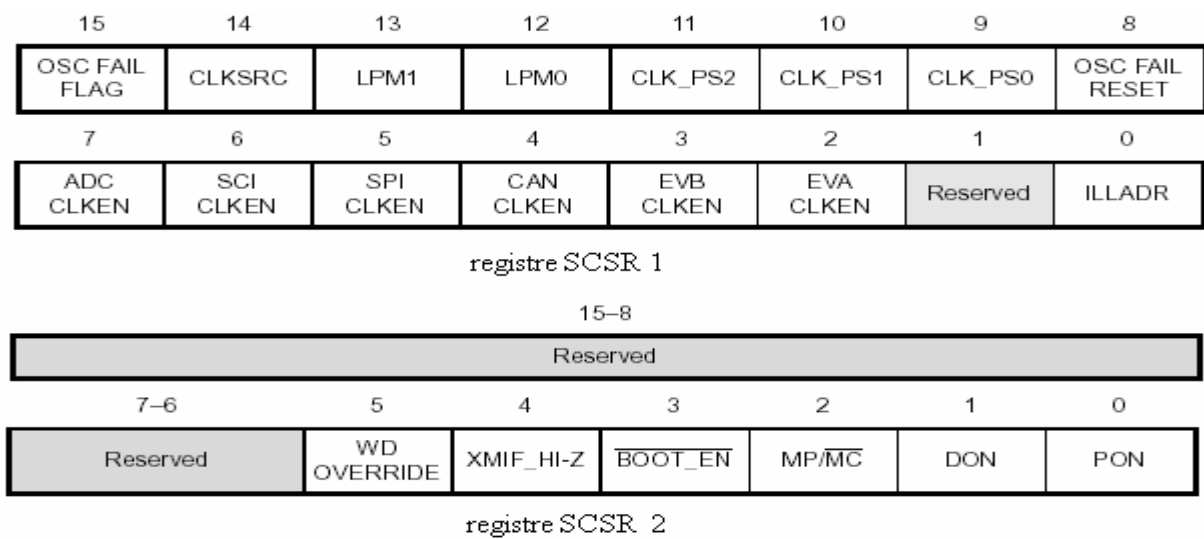


Figure (5.6) : Organigramme de filtre de Kalman.

#### 5.4.1. Partie initialisation

##### 5.4.1.1. Configuration du DSP

Toute application sur *DSP TMS 320 LF 2407* commence par la configuration du système ; c'est à dire choisir la fréquence de travail et sélectionner les éléments et organes du système concerné par l'application envisagée. Cette configuration se fait par l'écriture d'un mot de contrôle bien définie dans les deux registres de configuration du système SCSR1 et SCSR2 ; figure (5.7) :



**Figure (5.7) :** Registres de configuration du DSP.

Ces registres doivent être écrits au début de chaque application pour un bon fonctionnement du système. Les détails de la configuration de ces registres sont donnés dans l'annexe B.

Dans notre application nous avons configuré le DSP pour travailler à une fréquence de 40 MHz pour atteindre un temps de cycle machine de 25 ns. Nous avons également validé le système d'acquisition des grandeurs analogiques et les compteurs d'évènements nécessaires au paramétrage de la fréquence d'échantillonnage.

#### 5.4.1.2. Choix de la fréquence d'échantillonnage

Le Choix de la fréquence d'échantillonnage se fait par un temporisateur ( TIMER ) situé dans le compteur d'évènements. Le TIMER provoque une interruption à la fin de décomptage d'un nombre de pas fixé auparavant. Le TIMER peut être alimenté par une l'horloge interne après un passage sur le pré diviseur ou par une horloge externe à travers la broche TCLKINA/B. L'interruption déclenchée par le TIMER peut être utilisée pour exécuter une routine de service ou même démarrer le convertisseur analogique numérique. Le TIMER se caractérise par ces entrées, sorties et de contrôle, figure (5.8).

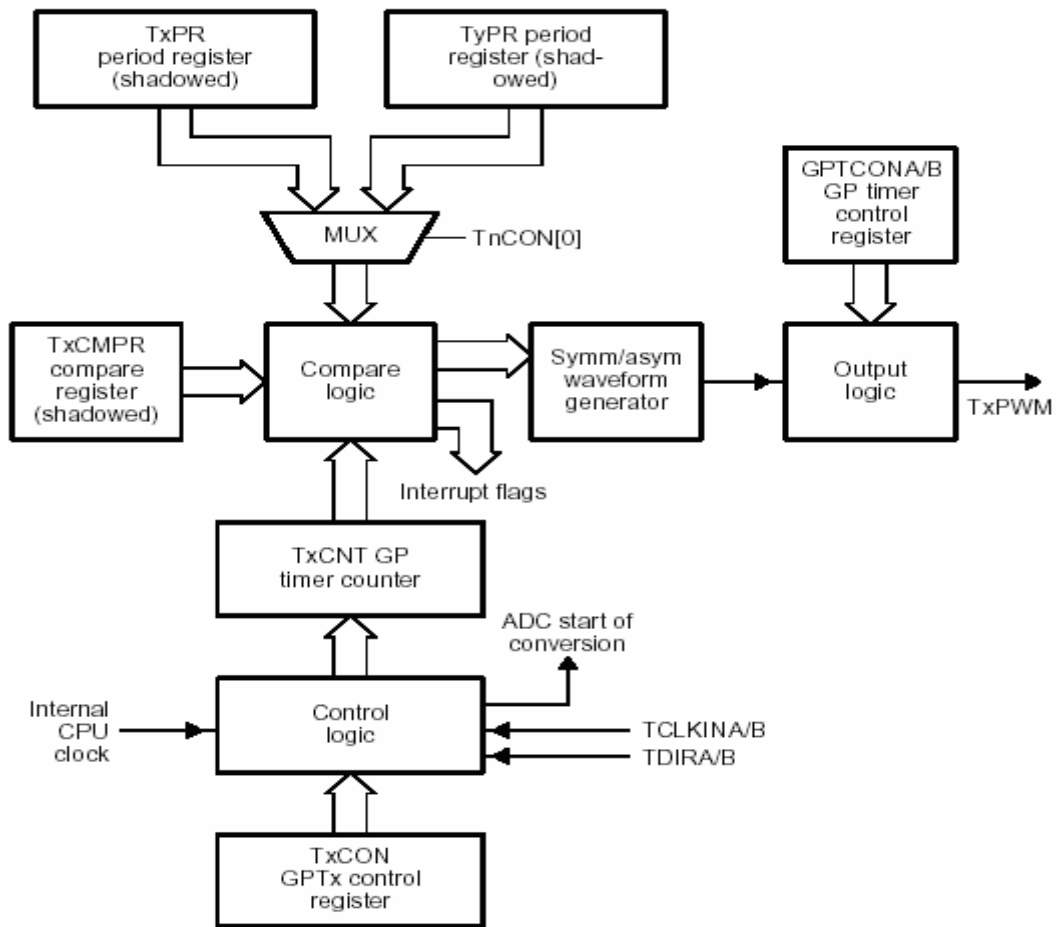


Figure (5.8) : Schéma bloc du temporisateur ( TIMER ).

**a - Entrées du TIMER:**

Les entrées du TIMER à usage général sont :

- l'horloge interne CLK.
- L'horloge externe TCLKINA/B.
- L'entrée direction TDIRA/B pour comptage ou décomptage.
- Signal de remise à zéro.

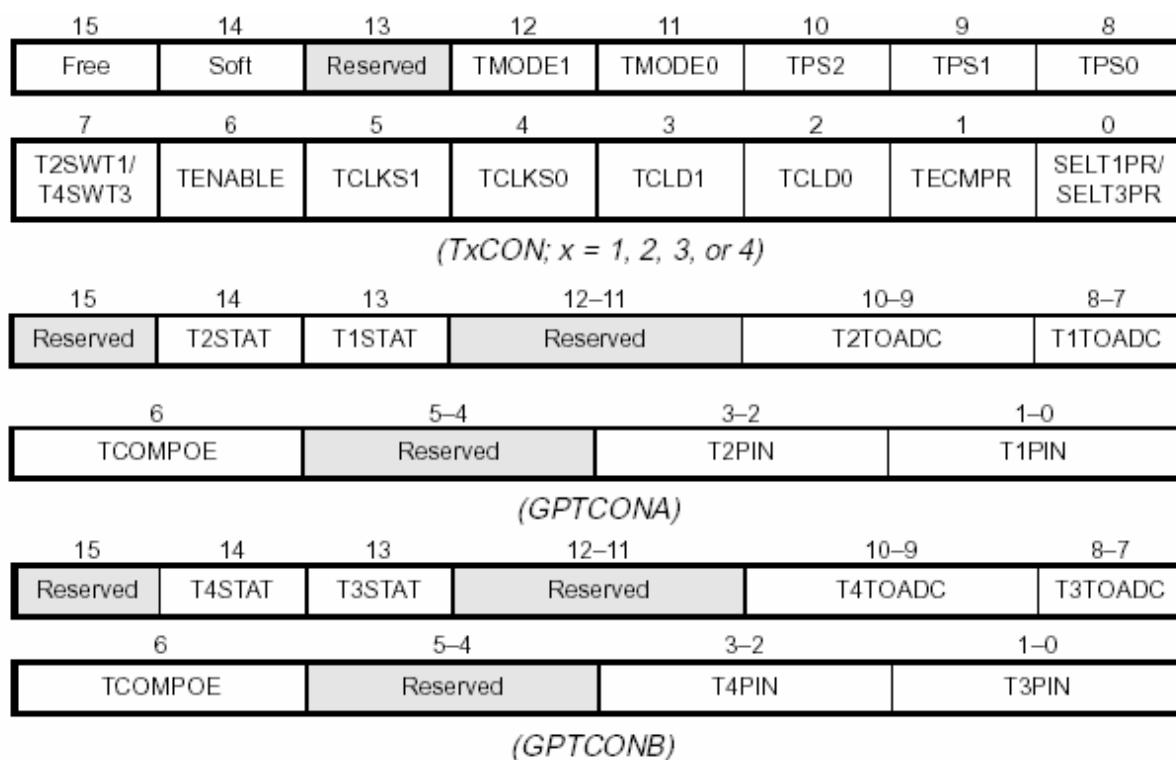
**b - Sorties du TIMER:**

Les sorties du TIMER sont :

- Sortie de comparaison.
- Démarrer le convertisseur analogique numérique.
- Indicateur de dépassement.
- Indicateur de direction.

**c - Contrôle du TIMER:**

Le TIMER peut être configuré et contrôlé par trois registres, figure (5.9). Les détails de la configuration sont donnés dans l'annexe B.



**Figure (5.9) :** Registres de contrôles du TIMER.

**5.4.1.3. Initialisation des entrées analogiques**

Le processeur de traitement de signal *DSP TMS 320LF2407* comporte 16 entrées analogiques ADCIN0 à ADCIN15 multiplexées vers un seul convertisseur analogique numérique de temps de conversion égal à 500 ns [18]. Le module de conversion peut travailler en deux modes ; le mode simple conversion et le mode séquence de conversion.

Le mode séquence de conversion se compose de deux séquences indépendantes de 8 états pour chaque séquence. Ces deux séquences peuvent se fusionner pour former une seule séquence de 16 états. La figure (5.10) nous donne la structure générale du module de conversion. Les résultats numériques de conversion sont stockés dans les registres RESULT0 à RESULT15 approprié pour chaque entrée.

A chaque début d'une application utilisant le module ADC on doit configurer ce dernier pour répondre aux besoins envisagés. Cette configuration est faite avec des mots binaires choisis conformément aux significations des bits de registres de contrôle ADCTRL1 et ADCTRL2 du module de conversion, figure (5.11). Les détails de configuration sont expliqués dans l'annexe B.

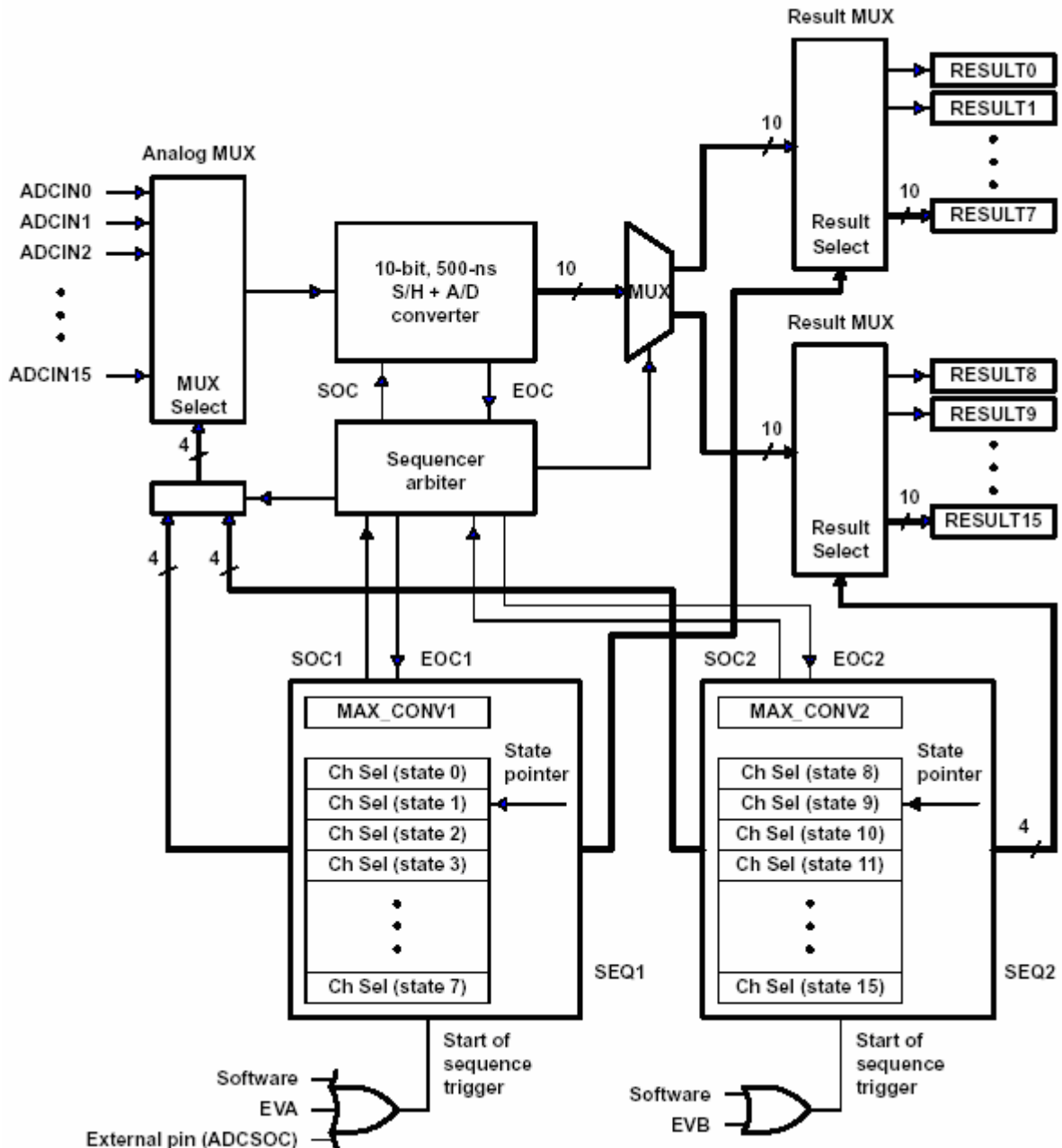


Figure (5.10) : Structure du module de conversion ADC.

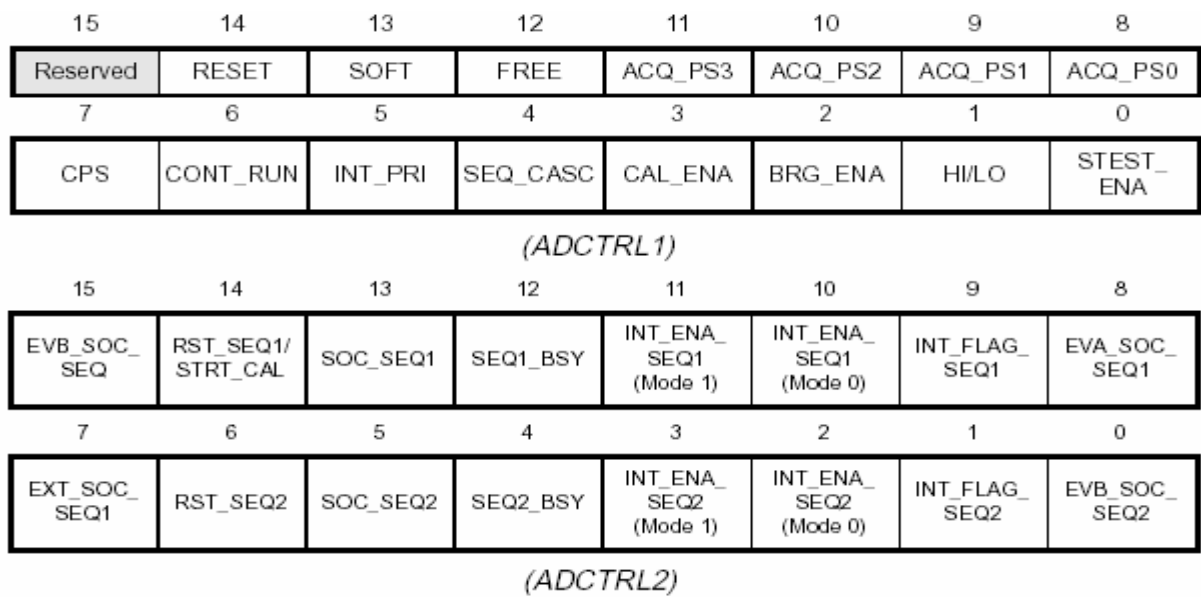


Figure (5.11) : Registres de contrôle du module ADC.

### 5.4.2. Acquisition des grandeurs analogiques pour le filtre de Kalman

La mise en œuvre du filtrage de Kalman nécessite l'acquisition des grandeurs analogiques courants, tensions et vitesse de la machine asynchrone. Le module de conversion est configuré pour réaliser une séquence de conversion de huit états selon l'ordonnement choisi, par exemple :  $I_1, I_2, I_3, V_1, V_2, V_3$  et  $w_r$ . Les conversions se feront avec un espacement temporel de  $500\text{ ns}$  ; c'est à dire un temps total de conversion égal à  $3,5\ \mu\text{s}$ . Les valeurs converties seront stockées dans les registres RESULT0 à RESULT6 après la fin de la séquence de conversion. Ces valeurs doivent être ajustées et calibrées pour nous donner les valeurs réelles des grandeurs mesurées.

### 5.4.3. Algorithme de filtre de Kalman

Le filtre de Kalman linéaire se compose de cinq équations matricielles, deux équations pour la prédiction et trois pour la correction de l'estimation d'état.

#### - Equations de prédiction

a – Prédiction de l'état

$$\hat{x}(k+1, k) = A_d \hat{x}(k, k) + B_d u(k)$$

[4,1]
[4,4]
[4,1]
[4,2]
[2,1]

b – Prédiction de la covariance d'erreur

$$\begin{array}{cccccc}
 P(k+1,k) & = & A_d & P(k,k) & A_d^T & + & Q \\
 [4,4,] & & [4,4] & [4,4] & [4,4] & & [4,4]
 \end{array}$$

**- Equations de correction**

a – Gain du filtre de Kalman

$$\begin{array}{ccccccc}
 K(k+1) & = & P(k+1,k) & C_d(k)^T & [C_d(k) & P(k+1,k) & C_d(k)^T & + & R]^{-1} \\
 [4,2] & & [4,4] & [4,2] & [2,4] & [4,4] & [4,2] & & [2,2]
 \end{array}$$

b – Correction de l'état

$$\begin{array}{cccccc}
 \hat{x}(k+1,k+1) & = & \hat{x}(k+1,k) & + & K(k+1) & [y(k+1) - C_d(k) \hat{x}(k+1,k)] \\
 [4,1] & & [4,1] & & [4,2] & [2,1] & [2,4] & & [4,1]
 \end{array}$$

c – Correction de la covariance

$$\begin{array}{ccccc}
 P(k+1,k+1) & = & [I & - & K(k+1) & C_d(k)] & P(k+1,k) \\
 [4,4] & & [4,4] & & [4,2] & [2,4] & [4,4]
 \end{array}$$

Dans chaque équation nous avons présenté les dimensions de chaque matrice. L'implantation de toutes algorithmes en temps réel nécessite la connaissance préalable de nombre d'opération élémentaire à effectuer entre deux acquisition successive dans l'objectif de voir si le processeur pilote supportera ou non le temps réel pour l'application envisagée. Les tableaux ci-dessous résument les différentes opérations de multiplication matricielle et les opérations élémentaires à effectuer pour chaque équation du filtre.

Multiplication Matricielle		Nombre d'opérations élémentaires	
Matrice1 × Matrice 2	Résultat	Multiplications	Additions
[4,4] [4,1]	[4,1]	16	12
[4,2] [2,1]	[4,1]	8	4
Addition Matricielle			
Matrice1 + Matrice 2	Résultat		
[4,1] [4,1]	[4,1]	/	4
Total		24	20

**Tableau (5.1) :** Opérations de multiplications et d'addition de l'équation  $\hat{x}(k+1,k)$ .

Multiplication Matriciel		Nombre d'opérations élémentaires	
Matrice1 × Matrice 2	Résultat	Multiplication	Addition
[4,4] [4,4]	[4,4]	64	48
[4,4] [4,4]	[4,4]	64	48
Addition Matriciel			
Matrice1 + Matrice 2	Résultat		
[4,4] [4,4]	[4,4]	/	16
Total		128	144

**Tableau (5.2) :** Opérations de multiplication  
et d'addition de l'équation  $P(k+1, k)$ .

Multiplication Matriciel		Nombre d'opérations élémentaires	
Matrice1 × Matrice 2	Résultat	Multiplication	Addition
[4,4] [4,2]	[4,2]	32	24
[2,4] [4,4]	[2,4]	32	24
[2,4] [4,2]	[2,2]	16	12
[4,2] [2,2]	[4,2]	16	8
Inversion matrice [2,2]	[2,2]	6	1
Addition Matricielle			
Matrice1 + Matrice 2	Résultat		
[2,2] [2,2]	[2,2]	/	4
Total		102	73

**Tableau (5.3) :** Opérations de multiplication  
et d'addition de 3<sup>ème</sup> équation  $K(k+1)$ .



Multiplication Matricielle		Nombre d'opérations élémentaires	
Matrice 1 × Matrice 2	Résultat	Multiplication	Addition
[2,4] [4,1]	[2,1]	8	6
[4,2] [2,1]	[4,1]	8	4
Addition Matricielle			
Matrice 1 + Matrice 2	Résultat		
[2,1] [2,1]	[2,1]	/	4
[4,1] [4,1]	[4,1]	/	4
Total		16	18

**Tableau (5.4) :** Opérations de multiplication  
et d'addition de l'équation  $\hat{x}(k+1, k+1)$ .

Multiplication Matricielle		Nombre d'opérations élémentaires	
Matrice 1 × Matrice 2	Résultat	Multiplication	Addition
[4,2] [2,4]	[4,4]	32	16
[4,4] [4,4]	[4,4]	64	48
Addition Matricielle			
Matrice 1 + Matrice 2	Résultat		
[4,4] [4,4]	[4,4]	/	16
Total		96	80

**Tableau (5.5) :** Opérations de multiplication  
et d'addition de l'équation  $P(k+1, k+1)$ .

L'opération d'inversion de la matrice [2,2] dans l'équation de calcul du gain de filtre de Kalman est calculée selon la relation :

$$A^{-1} = \frac{co-matrice(A)}{det(A)}$$

#### 5.4.4. Temps d'exécution de l'algorithme développer

Le nombre total d'opérations élémentaire à effectuer pour une seule exécution du filtre de Kalman est :

366 : Opérations de multiplication à virgule flottante sur 32 Bits.

335 : Opérations d'addition à virgule flottante sur 32 Bits.

Sachons qu'une opération de multiplication de deux nombres à virgule flottante se réalise en 126 cycles machine sur le *DSP TMS 320 LF 2407* et 155 cycles pour une opération d'addition. Dans notre application nous avons configurer le *DSP TMS320LF2407* pour travailler à une fréquence de 40 MHz ; c'est à dire un temps de cycle de 25 ns, donc l'algorithme du filtre de Kalman est exécuté dans un temps estimé égal à :

$$(366 \times 126 + 335 \times 155) \times 25 \text{ ns} = 2451 \mu\text{s}.$$

Sans compter les instructions utilisées pour réaliser les boucles et les changements d'indices des matrices avec le temps nécessaire pour l'acquisition des grandeurs analogiques fixé à 3,5  $\mu\text{s}$ , nous aurons un temps estimé d'exécution de l'algorithme égal à : 2455  $\mu\text{s}$ .

#### 5.5. Initialisation du filtre de Kalman

Les matrices discrétisées d'évolution, de commande et d'observation de la machine asynchrone pour une période d'échantillonnage égal à 2 ms sont donnés par :

$$A_d = \begin{bmatrix} 0.5171 & 0 & 0.2110 & 0.0412\omega_r \\ 0 & 0.7586 & -0.0412\omega_r & 0.2110 \\ 0.0065 & 0 & 0.9898 & -0.002\omega_r \\ 0 & 0.0065 & 0.002\omega_r & 0.9898 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0428 & 0 \\ 0 & 0.0428 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Les valeurs initiales des matrices de covariance des bruits et d'erreur d'estimation ainsi que l'état initial estimé sont choisis arbitrairement comme suit :

$$Q = \begin{bmatrix} 1.5 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

$$R = \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix}$$

$$\hat{x}_0 = [0 \ 0 \ 0 \ 0]^T$$

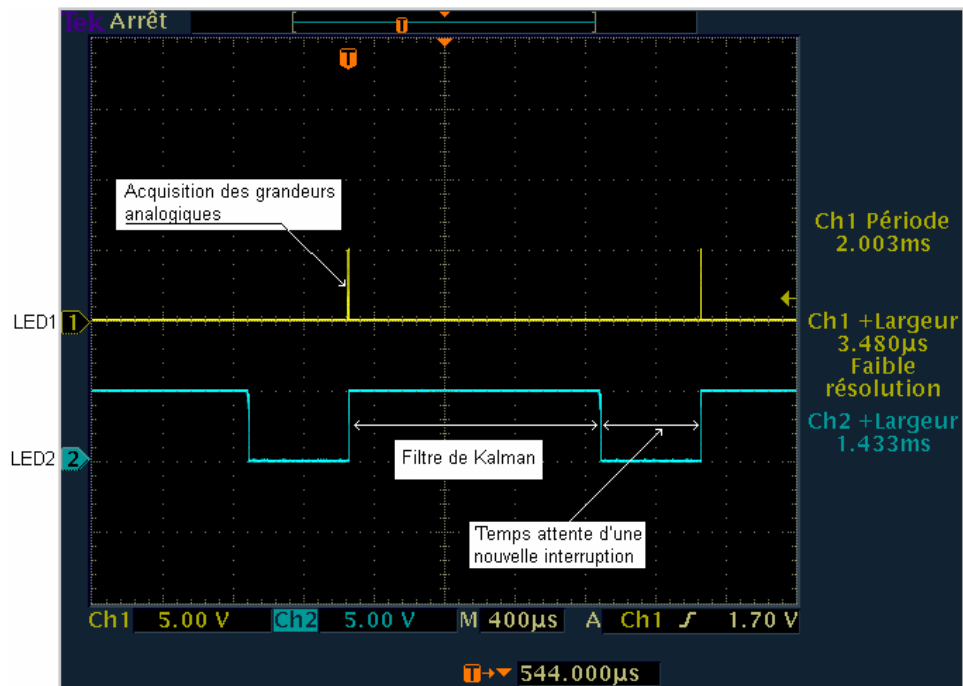
$$P_0 = \begin{bmatrix} 0.11 & 0 & 0 & 0 \\ 0 & 0.11 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

### 5.6. Chronogramme d'exécution de l'algorithme du filtre de Kalman

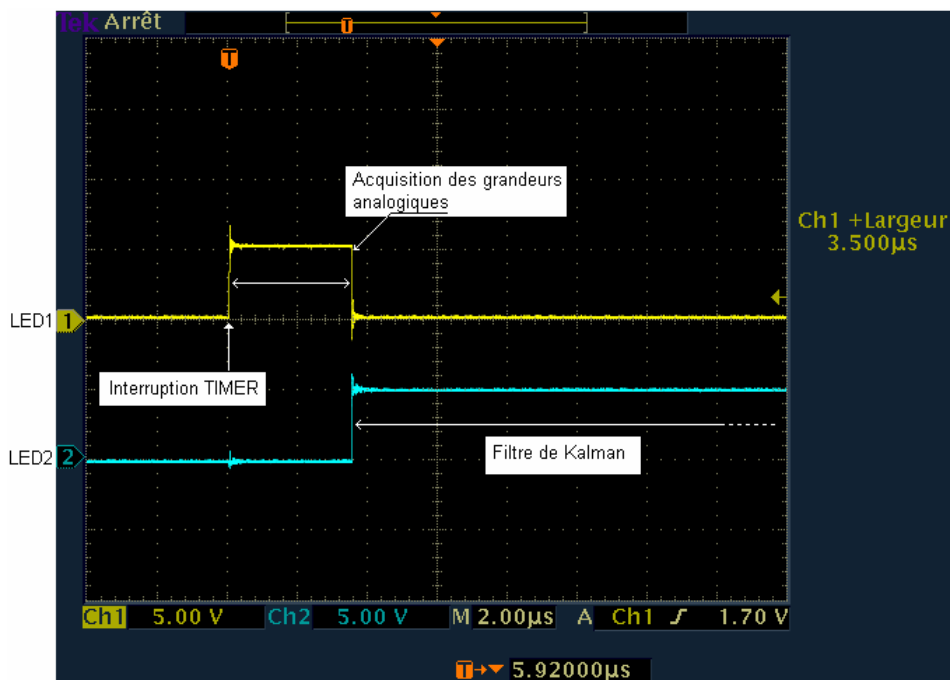
Un algorithme développé pour fonctionner en temps réel doit être exécuté dans une durée inférieure à la période d'échantillonnage exigée par l'application. Pour voir réellement les temps d'exécution des différents segments de l'algorithme développé et les comparer avec le temps estimé, nous avons utilisé les LEDs utilisateur disponibles sur la carte *DSP* pour s'informer sur l'état d'exécution du programme selon le tableau (5.6). Ces signaux seront en suite affichés sur un oscilloscope pour extraire les temps d'exécution réel de chaque segment de programme, figure (5.9).

Etat des LEDs		Segment correspondant du programme
LED 1	LED 2	
0	0	Le programme est en attente d'une interruption.
1	0	Acquisition des grandeurs analogiques.
0	1	Exécution des équations du filtre de Kalman.

**Tableau (5.6) :** Affectation des états des LEDs pour chaque segment de programme.



a – Temps d'exécution total de l'algorithme.



b – Agrandissement de la partie acquisition.

**Figure (5.12) :** Temps d'exécution des différents segments de l'algorithme développé.

D'après les deux signaux de la figure (5.12) *a* et *b*, on constate que le temps d'exécution total de l'algorithme de filtre de Kalman est :  $1,436\text{ ms}$  dont  $3,5\ \mu\text{s}$  sont exploités pour l'acquisition des grandeurs analogiques,  $1,425\text{ ms}$  le temps de calcul des équations du filtre de Kalman et le reste du temps jusqu'à  $2\text{ ms}$  ( la période d'échantillonnage ) ; c'est à dire  $564\ \mu\text{s}$  est le temps d'attente d'une nouvelle interruption a généré par le TIMER.

### 5.7. Résultat expérimental de l'estimation

Nous commençons la représentation des résultats expérimentaux par les grandeurs d'entrée utilisé par l'algorithme de filtre de Kalman qui sont le deux courant statotique ( $i_{ds}$ ,  $i_{qs}$ ) mesuré dans le plans de Park ; figure (5.13) et les deux tensions statoriques ( $u_{ds}$ ,  $u_{qs}$ ) dans le même plan, figure (5.14). Ces résultats sont prélevés par un l'oscilloscope numérique TEKTRONIX TDS 3052.

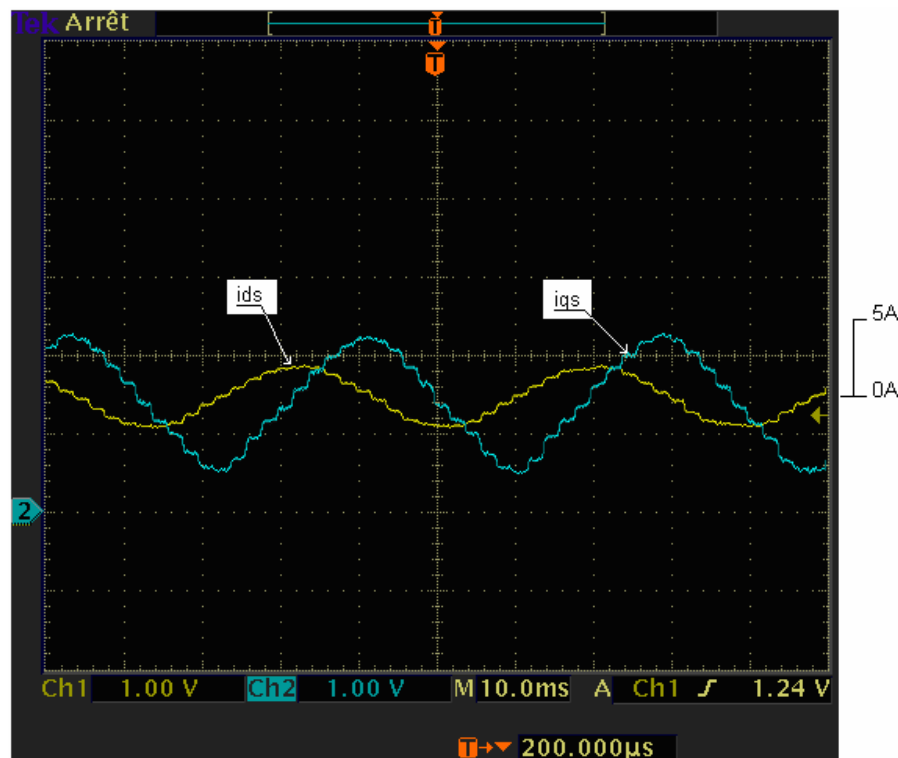
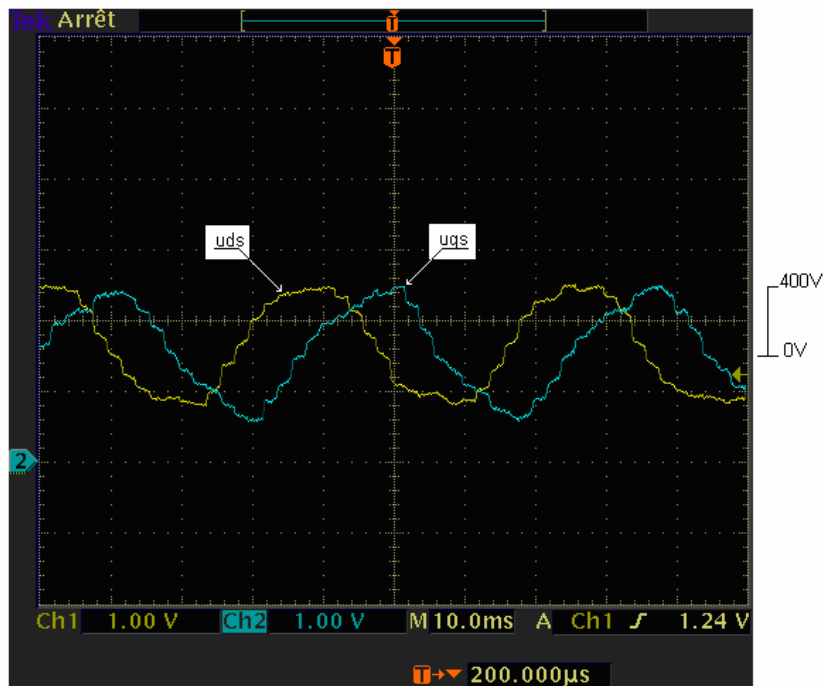
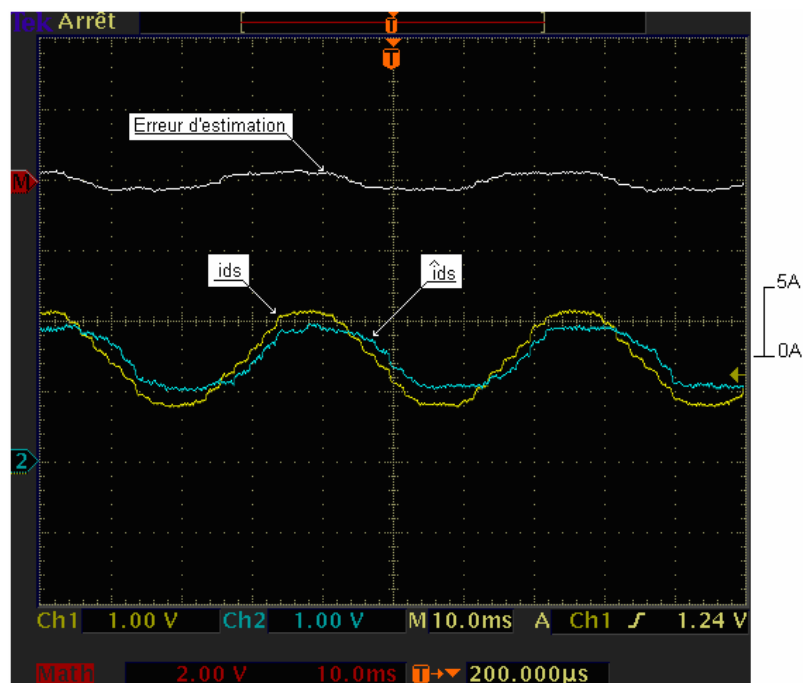


Figure (5.13) : Courants statoriques  $i_{ds}$ , et  $i_{qs}$  dans le plan de Park (  $5\text{ A/div}$  ).



**Figure (5.14) :** Tensions statoriques  $u_{ds}$  et  $u_{qs}$  dans le plan de Park (400 V/div).

Les résultats expérimentaux d'estimation obtenus par le filtre de Kalman se résument dans le courant statorique mesuré  $i_{ds}$  et estimé  $\hat{i}_{ds}$  avec l'erreur d'estimation ; figure (5.15), le courant statorique mesuré  $i_{qs}$  et estimé  $\hat{i}_{qs}$  et son erreur d'estimation ; figure (5.16).



**Figure (5.15) :** Courant statorique mesuré  $i_{ds}$  et estimé  $\hat{i}_{ds}$  (5 A/div).

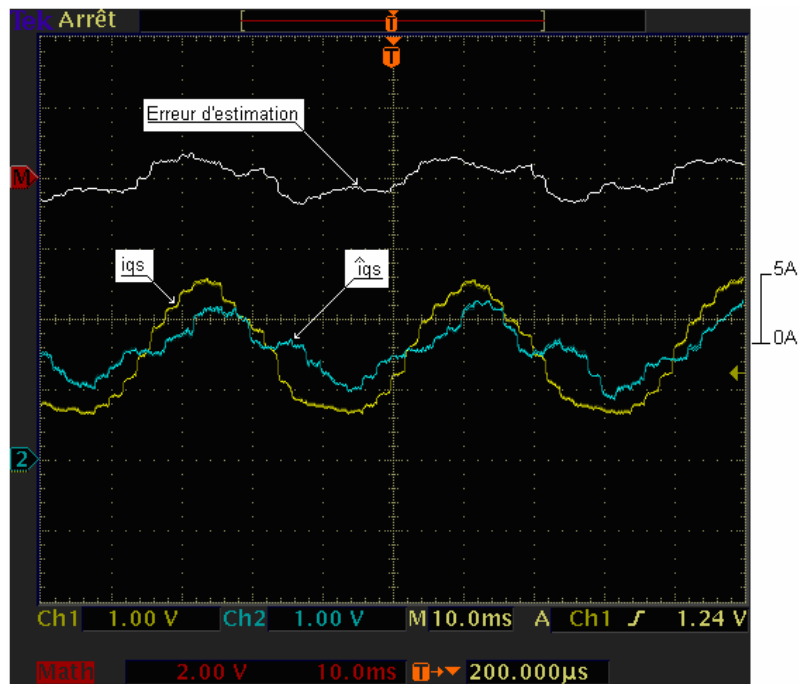


Figure (5.16) : Courant statorique mesuré  $i_{qs}$  et estimé  $\hat{i}_{qs}$  (5 A/div).

Les grandeurs non accessible à la mesure estimés par le filtre de Kalman sont les deux flux rotorique ( $\hat{\psi}_{dr}$  et  $\hat{\psi}_{qr}$ ) dans le plan de Park ; figure (5.17), représenté par des tensions correspondante.

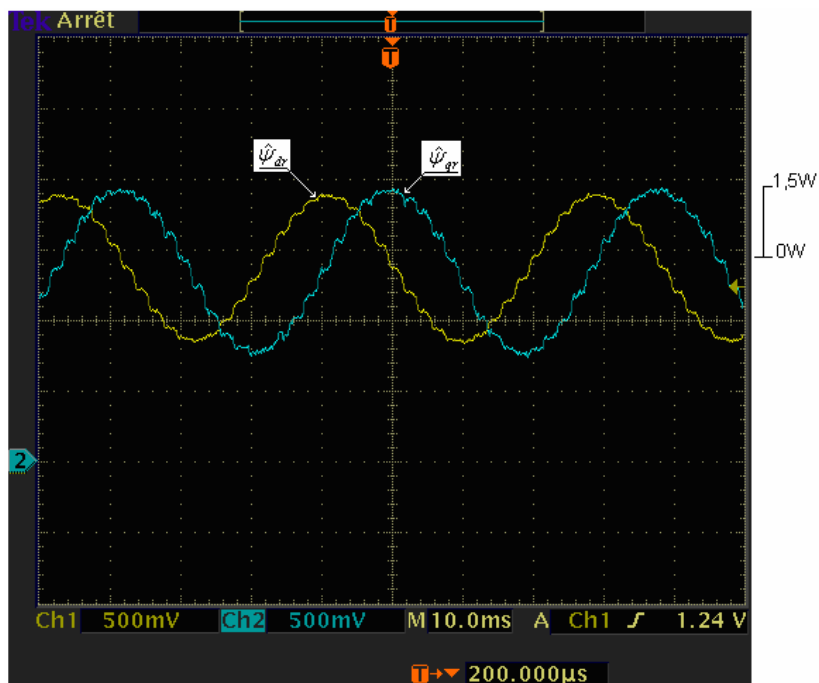


Figure (5.17) : Flux rotoriques estimés  $\hat{\psi}_{dr}$  et  $\hat{\psi}_{qr}$  (1,5 W/div).

## 5.8. Conclusion

Dans ce chapitre nous avons exposé les différentes étapes de développement software d'une estimation en temps réel des flux rotoriques de la machine asynchrone à cage par le filtre de Kalman linéaire. La mise en place soigneuse de la plate-forme expérimentale a permis l'implantation de l'algorithme d'estimation de l'état rotorique de la machine asynchrone. Le temps d'échantillonnage valable dans cette application doit être inférieur ou égal à la constante du temps rotorique de la machine, ( $Tr = Lr/Rr = 100 \text{ ms}$ ), conformément à plusieurs références bibliographiques. Le temps de calcul du filtre de Kalman est estimé à 2,5 ms pour une fréquence d'horloge du DSP égale à 40 MHz. Ce temps inclus d'exécution de 366 opérations de multiplications, 335 opérations d'additions ainsi que le temps d'acquisition de sept signaux analogiques.

Dans ces conditions le temps d'échantillonnage minimal, pour une programmation en C du filtre de Kalman, est de 1,433 ms. Les résultats obtenus pour un fonctionnement en temps réel sont satisfaisants et peuvent être améliorés en optimisant le programme dans le but de réduire la fréquence d'échantillonnage.

Quelques troubles relatifs à la propagation des bruits de commutations en plus quelques limitations aux niveaux du langage de programmation utilisé causent par fois un déséquilibre de l'algorithme.



## ***Conclusion et perspectives***

Dans ces dernière années, le filtre de Kalman appliqué à la machine asynchrone a fait l'objet de plusieurs travaux de recherche, sur le plan théorique ( modélisation et formulation des équations de filtre du filtre pour la machine ) ou sur le plan expérimental ( implantation en temps réel et estimation en ligne des grandeurs ).

Notre travail à pour objectif la mise en œuvre pratique d'une indentification par filtre de Kalman des grandeurs rotoriques non accessibles directement à la mesure. C'est autour du processeur TMS 320 LF 2407, processeur dédié aux applications de commande ou de contrôle des machines électriques, que notre banc d'essais à été développé est validé. Les algorithmes d'identification par filtre de Kalman linéaire développés et mis au point par simulation sous MATLAB SIMULINK, après implantation sur DSP et à partir des mesures correctes des différentes grandeurs accessibles ( courant, tension et vitesse ), permettent l'identification en ligne des flux rotoriques de la machine asynchrone à cage.

Pour un temps d'échantillonnage égal à 2 ms les résultats obtenus sont satisfaisants. Ces résultats peuvent être amélioré par amélioration du temps d'échantillonnage qui est limité par le temps de calcule des équations du filtre. Ce temps est lié à la complexité des équations, à la fréquence d'horloge du processeur et aux performances du compilateur/optimisateur utilisé (outil de développement software ) en plus à la rapidité du convertisseur analogique-numérique.

L'implantation du filtre de Kalman étendu pour l'estimation de la vitesse repose sur le même principe avec un temps de calcul très important pour des équations plus complexes. Pour des raisons de délais de préparation de mémoire cette partie a été seulement simulée.

Comme perspective nous proposons l'exploration du banc d'essais réalisé pour :

- Exploiter les signaux de retour pour implanter différentes stratégies de commande de la machine asynchrone.

- Introduire d'autres algorithmes d'indentification ( flous, algorithmes génétiques, MRAS, .... etc. ).
- Développer les routines d'indentification en assembleur pour une meilleure optimisation des temps d'exécution.

## Annexe A

### 1. Structure des bus interne du DSP TMS 320 LF 2407

Le DSP TMS 320 LF 2407 contient six bus interne de programme et de données. Ces bus sont repartie en bus d'adresse et de données, de lecture et écriture avec une largeur de 16 bits. Les six bus interne du DSP sont, schéma (1) :

- **PAB** : Bus d'adresse de programme, fourni des adresse lecture et écriture dans la mémoire de programme.
- **DRAB** : Bus d'adresse pour la lecture des données,. Il est utilisé pour l'adressage des à lire de la mémoire de données.
- **DWAB** : Bus d'adresse pour écriture des données, fourni des adresse pour l'écriture sur la mémoire de données.
- **PRDB** : Bus de lecture de programme, utilisé pour la lecture des instruction ou les opérandes immédiates vers l'unité centrale de calcul ( CPU ).
- **DRDB** : Bus de lecture de données, porte des données de la mémoire de donnée à l'unité arithmétique et logique centrale ( CALU ) et l'unité arithmétique des registres auxiliaires ( ARAU ).
- **DWEB** : Bus d'écriture des données, porte les données vers le mémoire de programme et la mémoire de données.

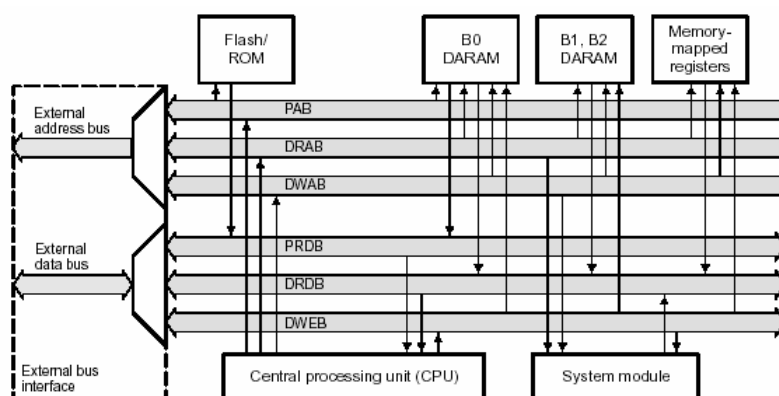


Schéma (1) : Structure des bus interne du DSP TMS 320 LF 2407.

## 2. Unité centrale de traitement

L'unité centrale de traitement se compose de trois sections fondamentales présentées dans le schéma (2), une entrée à décalage, une unité arithmétique et logique centrale et une unité de multiplication.

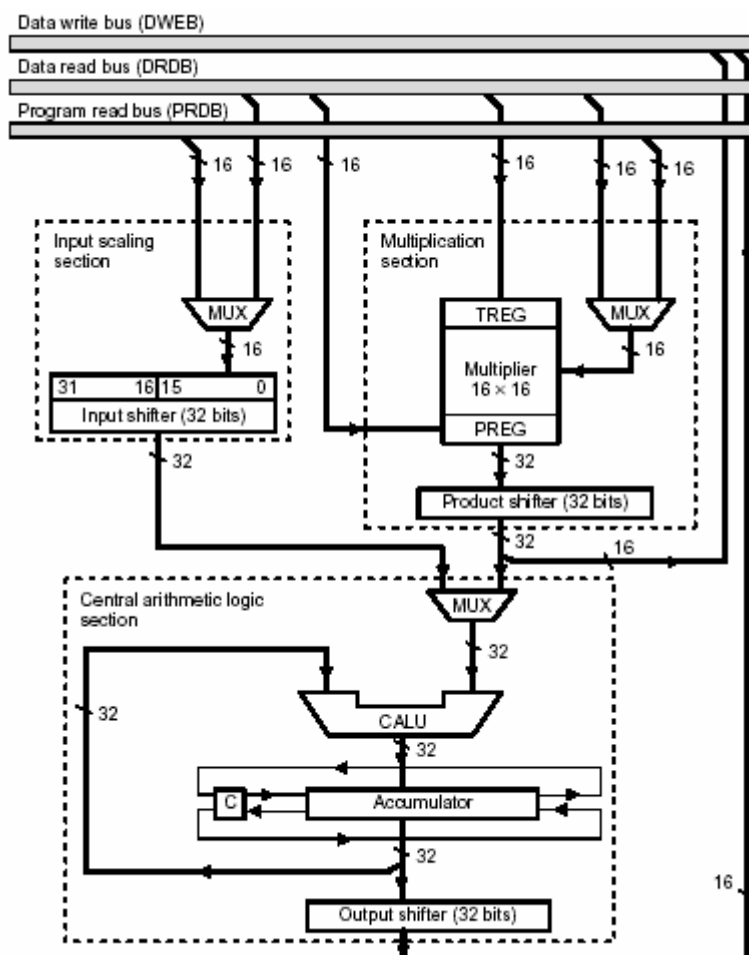
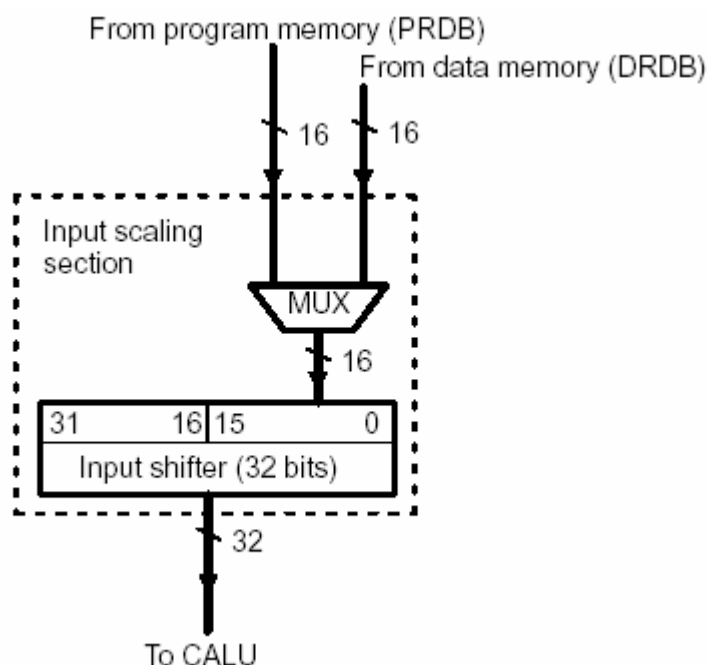


Schéma (2) : Unité centrale de traitement.

## 3. Section d'entrée à décalage 32 bits

L'entrée à décalage, schéma (2), est utilisée pour aligner les données de 16 bits provenant de la mémoire de données à des données de 32 bits pour être traitées par l'unité arithmétique et logique centrale (CALU). Ce décalage est nécessaire pour le calibrage des données et aussi bien pour l'alignement des masques dans les opérations logiques. L'entrée à décalage travaille comme un point d'accès entre la mémoire de programme ou de données et l'unité arithmétique et logique centrale. Les données peuvent arriver de mémoire de programme à travers le bus

( PRDB ) ou de la mémoire de données à travers le bus ( DRDB ) et il sera décalé sans avoir besoin d'une autre cycle machine.



**Schéma (3)** : Section d'entrée à décalage.

#### 4. Section de multiplication

La section de multiplication, schéma (4), se compose d'un multiplieur câblé capable de faire une opération de multiplication 16 Bits  $\times$  16 Bits en une seul cycle machine. La section de multiplication comporte :

- Une registre temporaire ( TREG ) utilise pour stocker l'un des opérandes.
- Le multiplieur, qui multiplier la valeur de TREG par une autre donnée arrive de la mémoire de programme ou de donnée.
- Un registre de produit ( PREG ) qui reçoit le résultat de multiplication.
- Un registre à décalage de produit pour le calibrage des résultats avant de passer à l'unité arithmétique et logique centrale ( CALU ).

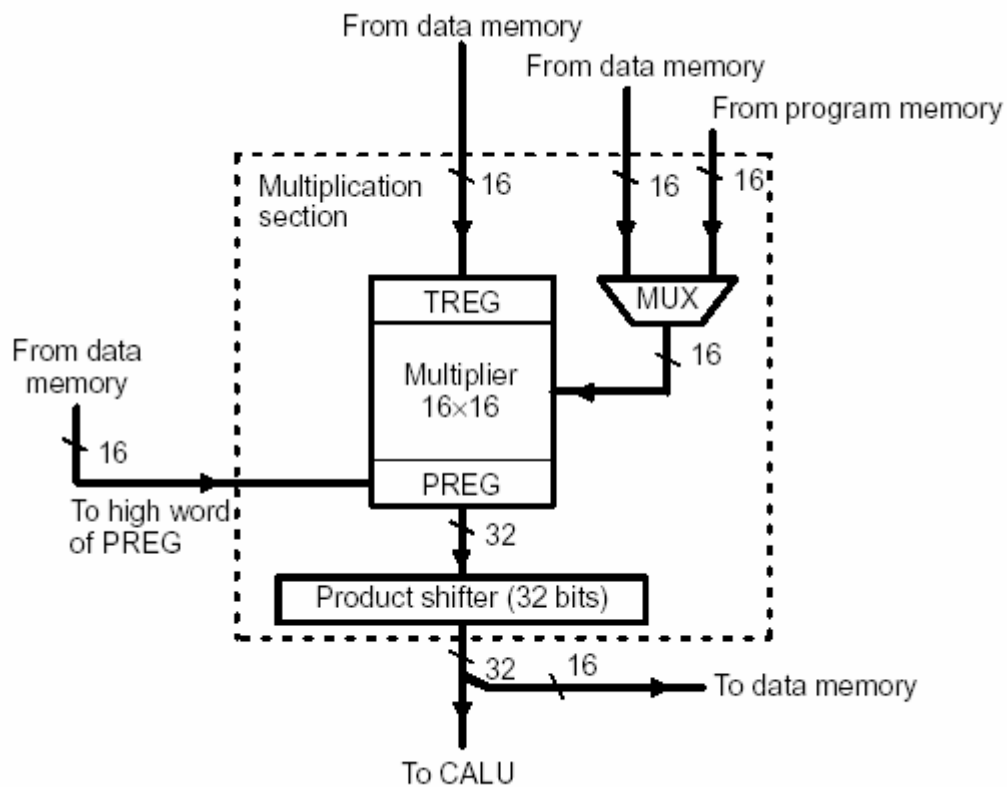
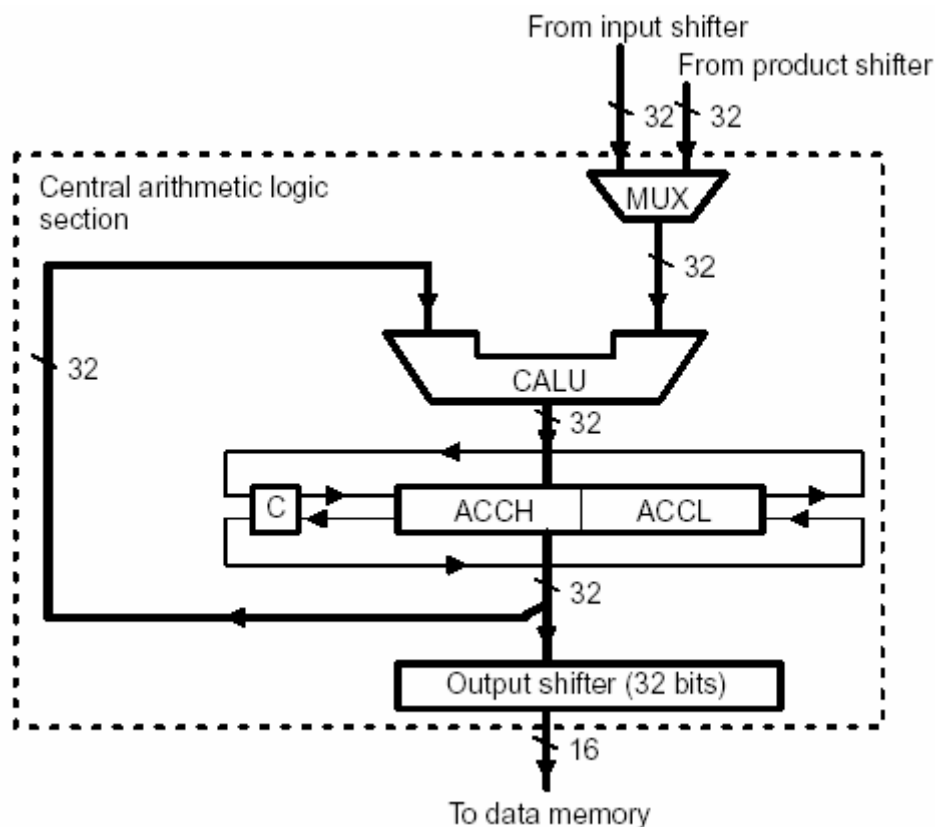


Schéma (4) : Section de multiplication.

## 5. Section d'unité arithmétique et logique centrale

Les composants principaux de la section d'unité arithmétique et logique sont montrés dans le schéma (5). Ces composants sont :

- L'unité arithmétique et logique centrale ( CALU ) qui implémente une variante de fonctions arithmétiques et logiques.
- L'accumulateur ( ACC ) qui reçoit la sortie du CALU et capable de faire des opérations de décalage par bits dans les deux sens avec l'aide des bits de retenue ( C ).
- Le registre à décalage de sortie qui peut faire une copie du mot de poids faible ou du mot de poids fort avant l'envoyer vers la mémoire de données pour stockage.



**Schéma (5) :** Unité arithmétique et logique central.

L'unité arithmétique et logique centrale peut effectuer les opérations suivantes :

- Addition 16 bits.
- Soustraction 16 bits.
- Opération logique ( AND, OR , XOR )
- Teste des Bits, décalage et rotation.

## 6. Unité arithmétique des registres auxiliaires

L'unité centrale de traitement contient aussi une unité arithmétique indépendante de l'unité arithmétique et logique centrale, c'est l'unité arithmétique des registres auxiliaires ( ARAU ), schéma (6). Sa principale fonction est de faire des opérations arithmétiques sur les huit registres auxiliaires (AR0 à AR7) en parallèle avec les opérations déroulant dans l'unité arithmétique et logique centrale ( CALU ). Les huit registres auxiliaires fournissent un outil puissant et flexible dans l'adressage indirect. N'importe quel emplacement mémoire dans l'espace mémoire des données peut être consulté en utilisant une adresse de 16 bits contenue dans un registre auxiliaire.

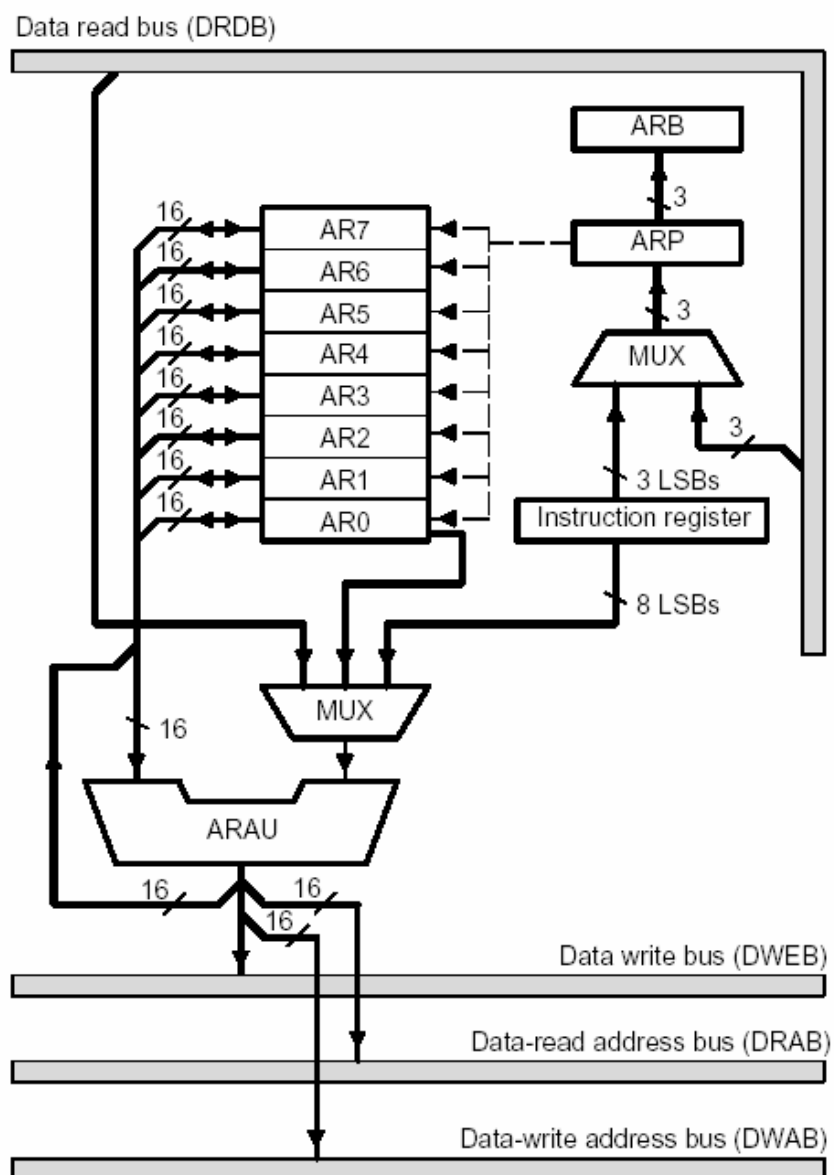


Schéma (6) : Unité arithmétique de registre auxiliaire.



## Annexe B

## Configuration Registers

## 1. System Control and Status Registers 1 and 2 (SCSR1, SCSR2)

## 1.1. System Control and Status Register 1 (SCSR1) — Address 07018h

15	14	13	12	11	10	9	8
OSC FAIL FLAG	CLKSRC	LPM1	LPM0	CLK_PS2	CLK_PS1	CLK_PS0	OSC FAIL RESET
RW-0	RW-0	RW-0	RW-0	RW-1	RW-1	RW-1	RW-0
7	6	5	4	3	2	1	0
ADC CLKEN	SCI CLKEN	SPI CLKEN	CAN CLKEN	EVB CLKEN	EVA CLKEN	Reserved	ILLADR
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	R-0	RW-0

**Bit 15 Oscillator Fail Flag**

When set to 1, indicates that the external reference (oscillator or crystal) is operating at a frequency too slow to be functional when compared to an internal reference oscillator. Power on reset value is 0

- 0 Operating at too slow frequency
- 1 Operating normally.

**Bit 14 CLKSRC.** CLKOUT pin source select

- 0 CLKOUT pin has CPU Clock (30 MHz on a 30-MHz device) as the output.
- 1 CLKOUT pin has Watchdog clock as the output

**Bits 13–12 LPM(1:0).** Low-power mode select

These bits indicate which low-power mode is entered when the CPU executes the IDLE instruction. See Table 2–1 for description of the low-power modes.

LPM(1:0)	Low-Power mode selected
00	IDLE1 (LPM0)
01	IDLE2. (LPM1)
1x	HALT (LPM2)

**Bits 11–9 PLL Clock prescale select.** These bits select the PLL multiplication factor for the input clock.

CLK PS2	CLK PS1	CLK PS0	System Clock Frequency
0	0	0	4 x Fin
0	0	1	2 x Fin
0	1	0	1.33 x Fin
0	1	1	1 x Fin
1	0	0	0.8 x Fin
1	0	1	0.66 x Fin
1	1	0	0.57 x Fin
1	1	1	0.5 x Fin

**Bit 8 Reset OSC Fail.** Reset if Oscillator fails

- 0 System reset is NOT initiated if the clock monitor detects a bad oscillator input.
- 1 System reset is initiated if the clock monitor detects a bad oscillator input.

**Bit 7 ADC CLKEN.** ADC module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

**Bit 6 SCI CLKEN.** SCI module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

**Bit 5 SPI CLKEN.** SPI module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

**Bit 4 CAN CLKEN.** CAN module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

**Bit 3 EVB CLKEN.** EVB module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

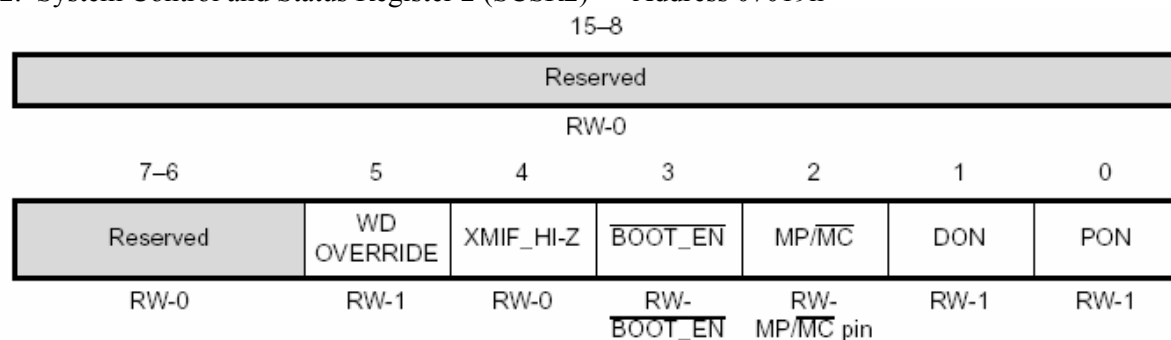
**Bit 2 EVA CLKEN.** EVA module clock enable control bit

- 0 Clock to module is disabled (i.e. shut down to conserve power)
- 1 Clock to module is enabled and running normally

**Bit 1 Reserved****Bit 0 ILLADR.** Illegal Address detect bit

If an illegal address has occurred this bit will be set. It is up to software to clear this bit following an illegal address detect. Note: An illegal address will cause an NMI.

## 1.2. System Control and Status Register 2 (SCSR2) — Address 07019h



**Bits 15–6 Reserved.** Writes have no effect, read are undefined

**Bit 5 Watchdog Override.** (WD protect bit)

After RESET, this bit gives the user the ability to disable the WD function through software (by setting the WDDIS bit = 1 in the WDCR). This bit is a clear only bit and defaults to a 1 after reset, Note: this bit is cleared by writing a 1 to it.

- 0 Protects the WD from being disabled by software. This bit cannot be set to 1 by software. It is a clear- only bit, cleared by writing a 1.
- 1 This is the default reset value and allows the user to disable the WD through the WDDIS bit in the WDCR. Once cleared however, this bit can no longer be set to 1 by software, thereby protecting the integrity of the WD timer.

**Bit 4 XMIF\_Hi-Z Control** This bit controls the state of the external memory interface (XMIF) signals.

- 0 XMIF signals in normal driven mode, i.e., not Hi-Z (high impedance)
- 1 All XMIF signal are forced to Hi-Z state

**Bit 3 Boot Enable**

This bit reflects the state of the BOOT\_EN / XF pin at the time of reset. After reset and device has “Booted up”, this bit can be changed in software to re-enable Flash memory visibility, or return to active Boot ROM.

- 0 Enable Boot ROM — Address space 0000 — 00FF is now occupied by the on-chip Boot ROM Block. Flash memory is totally disabled in this mode.
- 1 Disable Boot ROM — Program address space 0000 — 7FFF is mapped to on-chip Flash Memory, in case of 'LF2407 and 'LF2406. In case of 'LF2402, addresses 0000 – 1FFF are mapped.

**Bit 2 Microprocessor / Microcontroller Select**

This bit reflects the state of the MP/MC pin at time of reset. After reset this bit can be changed in software to allow dynamic mapping of memory on and off chip.

- 0 Set to Microcontroller mode — Program Address range 0000 — 7FFF is mapped internally (i.e. Flash or ROM)
- 1 Set to Microprocessor mode — Program Address range 0000 — 7FFF is mapped externally (i.e. Customer provides external memory device.)

Note: MP/MC pin is available only in 'LF2407.

**Bits 1–0 SARAM Program / Data Space Select**

DON	PON	SARAM status
0	0	SARAM not mapped (disabled), address space allocated to external memory.
0	1	SARAM mapped internally to Program space
1	0	SARAM mapped internally to Data space
1	1	SARAM block mapped internally to both Data and Program spaces. This is the default or reset value.

**2. Individual GP Timer Control Register (TxCON; x = 1, 2, 3, or 4)**

2.1. Timer Control Register (TxCON; x = 1, 2, 3, or 4) — Addresses 7404h (T1), 7408h (T2), 7504h (T3), and 7508h (T4)

15	14	13	12	11	10	9	8
Free	Soft	Reserved	TMODE1	TMODE0	TPS2	TPS1	TPS0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
7	6	5	4	3	2	1	0
T2SWT1/ T4SWT3	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SELT1PR/ SELT3PR
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

**Bits 15–14 Free, Soft.** Emulation control bits.

- 00 Stop immediately on emulation suspend.
- 01 Stop after current timer period is complete on emulation suspend.
- 10 Operation is not affected by emulation suspend.
- 11 Operation is not affected by emulation suspend.

**Bit 13 Reserved.** Reads return zero, writes have no effect.

**Bits 12–11 TMODE1–TMODE0.** Count Mode Selection.

- 00 Stop/Hold
- 01 Continuous-Up/-Down Count Mode
- 10 Continuous-Up Count Mode
- 11 Directional-Up/-Down Count Mode

**Bits 10–8 TPS2–TPS0.** Input Clock Prescaler.

- 000 x/1 100 x/16
- 001 x/2 101 x/32

010 x/4 110 x/64  
 011 x/8 111 x/128  
 x = device (CPU) clock frequency

**Bit 7 T2SWT1.** In case of EVA, this bit is T2SWT1. (GP timer 2 start with GP timer 1.) Start GP timer 2 with GP timer 1's timer enable bit. This bit is reserved in T1CON.

**T4SWT3.** In case of EVB, this bit is T4SWT3. (GP timer 4 start with GP timer 3.) Start GP timer 4 with GP timer 3's timer enable bit. This bit is reserved in T3CON.

0 Use own TENABLE bit.

1 Use TENABLE bit of T1CON to enable and disable operation ignoring own TENABLE bit.

**Bit 6 TENABLE.** Timer enable.

0 Disable timer operation (the timer is put in hold and the prescaler counter is reset).

1 Enable timer operations.

**Bits 5–4 TCLKS1, TCLKS0.** Clock Source Select.

5	4	Timer 1	Timer 2
0	0	Internal	Internal
0	1	External	External
1	0	Reserved	Reserved
1	1	Reserved	QEP Circuit

**Bits 3–2 TCLD1, TCLD0.** Timer Compare Register Reload Condition.

00 When counter is 0.

01 When counter value is 0 or equals period register value.

10 Immediately

11 Reserved

**Bit 1 TECMPR.** Timer compare enable.

0 Disable timer compare operation.

1 Enable timer compare operation.

**Bit 0 SELT1PR.** In case of EVA, this bit is SELT1PR. (Period register select.) This bit is a reserved bit in T1CON.

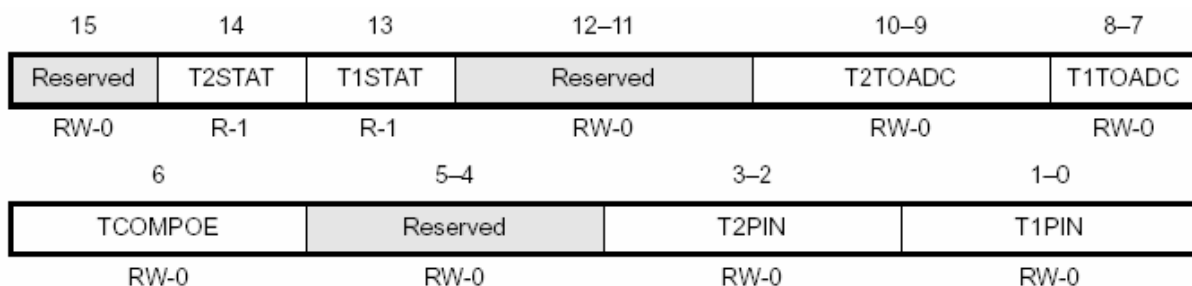
**SELT3PR.** In case of EVB, this bit is SELT3PR. (Period register select.) This bit is a reserved bit in 3CON.

0 Use own period register.

1 Use T1PR (in case of EVA) or T3PR (in case of EVB) as period register ignoring own period register.

### 3. Overall GP Timer Control Register (GPTCONA)

#### 3.1. GP Timer Control Register A (GPTCONA) — Address 7400h



**Bit 15 Reserved.** Reads return zero; writes have no effect.

**Bit 14 T2STAT.** GP timer 2 Status. Read only.

0 Counting downward

1 Counting upward

**Bit 13 T1STAT.** GP timer 1 Status. Read only.

- 0 Counting downward
- 1 Counting upward

**Bits 12–11 Reserved.** Reads return zero; writes have no effect.

**Bits 10–9 T2TOADC.** Start ADC with timer 2 event.

- 00 No event starts ADC.
- 01 Setting of underflow interrupt flag starts ADC.
- 10 Setting of period interrupt flag starts ADC.
- 11 Setting of compare interrupt flag starts ADC.

**Bits 8–7 T1TOADC.** Start ADC with timer 1 event.

- 00 No event starts ADC.
- 01 Setting of underflow interrupt flag starts ADC.
- 10 Setting of period interrupt flag starts ADC.
- 11 Setting of compare interrupt flag starts ADC.

**Bit 6 TCOMPOE.** Compare output enable. If PDPINTx is active this bit is set to zero.

- 0 Disable all GP timer compare outputs (all compare outputs are put in the high-impedance state).
- 1 Enable all GP timer compare outputs.

**Bits 5–4 Reserved.** Reads return zero; writes have no effect.

**Bits 3–2 T2PIN.** Polarity of GP timer 2 compare output.

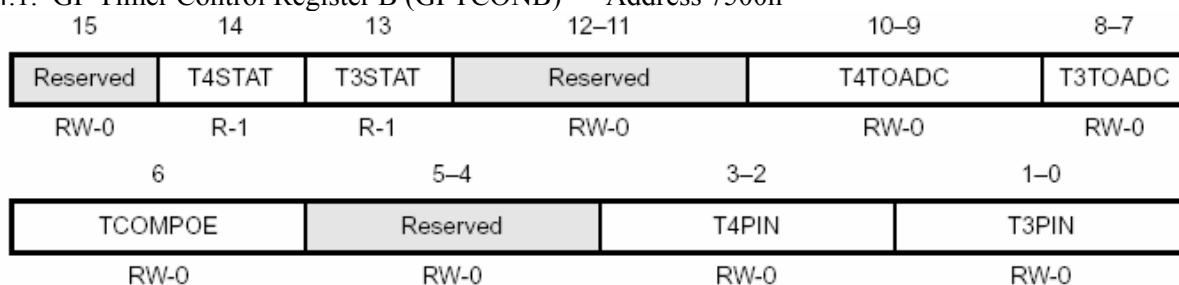
- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

**Bits 1–0 T1PIN.** Polarity of GP timer 1 compare output.

- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

#### 4. Overall GP Timer Control Register (GPTCONB)

4.1. GP Timer Control Register B (GPTCONB) — Address 7500h



**Bit 15 Reserved.** Reads return zero; writes have no effect.

**Bit 14 T4STAT.** GP timer 4 Status. Read only.

- 0 Counting downward
- 1 Counting upward General-Purpose (GP) Timers

**Bit 13 T3STAT.** GP timer 3 Status. Read only.

- 0 Counting downward
- 1 Counting upward

**Bits 12–11 Reserved.** Reads return zero; writes have no effect.

**Bits 10–9 T4TOADC.** Start ADC with timer 4 event.

- 00 No event starts ADC.
- 01 Setting of underflow interrupt flag starts ADC.
- 10 Setting of period interrupt flag starts ADC.
- 11 Setting of compare interrupt flag starts ADC.

**Bits 8–7 T3TOADC.** Start ADC with timer 3 event.

- 00 No event starts ADC.
- 01 Setting of underflow interrupt flag starts ADC.
- 10 Setting of period interrupt flag starts ADC.
- 11 Setting of compare interrupt flag starts ADC.

**Bit 6 TCOMPOE.** Compare output enable. If PDPINTx is active this bit is set to zero.

- 0 Disable all GP timer compare outputs (all compare outputs are put in the high-impedance state).
- 1 Enable all GP timer compare outputs.

**Bits 5–4 Reserved.** Reads return zero; writes have no effect.

**Bits 3–2 T4PIN.** Polarity of GP timer 4 compare output.

- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

**Bits 1–0 T3PIN.** Polarity of GP timer 3 compare output.

- 00 Forced low
- 01 Active low
- 10 Active high
- 11 Forced high

## 5. ADC Control Registers

### 5.1. ADC Control Register 1 (ADCTRL1) — Address 70A0h

15	14	13	12	11	10	9	8
Reserved	RESET	SOFT	FREE	ACQ_PS3	ACQ_PS2	ACQ_PS1	ACQ_PS0
	RS-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
7	6	5	4	3	2	1	0
CPS	CONT_RUN	INT_PRI	SEQ_CASC	CAL_ENA	BRG_ENA	HI/LO	STEST_ENA
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

**Bit 15 Reserved**

**Bit 14 RESET.** ADC module software reset

This bit causes a master reset on the entire ADC module. All register bits and sequencer state machines are reset to the initial state as occurs when the device reset pin is pulled low (or after a power-on reset).

- 0 No effect
- 1 Resets entire ADC module (bit is then set back to 0 by ADC logic)

**Bits 13, 12 SOFT and FREE.** Soft and Free bits

These bits determine what occurs when an emulation-suspend occurs (due to the debugger hitting a breakpoint, for example). In free-run mode, the peripheral can continue with whatever it is doing. In stop mode, the peripheral can either stop immediately or stop when the current operation (i.e., the current conversion) is complete.

- |      |      |                           |
|------|------|---------------------------|
| Soft | Free |                           |
| 0    | 0    | Immediate stop on suspend |

1	0	Complete current conversion before stopping
X	1	Free run, continue operation regardless of suspend

**Bits 11–8 ACQ\_PS3 – ACQ\_PS0.** Acquisition time window – prescale bits 3–0

These bits define the ADC clock prescale factor applied to the acquisition portion of the conversion. The prescale values are defined below:

#	ACQ_ PS3	ACQ_ PS2	ACQ_ PS1	ACQ_ PS0	Pre-scaler (div by)	Acquisition Time Window	Source Z (CPS= 0) ( $\Omega$ )	Source Z (CPS=1) ( $\Omega$ )
0	0	0	0	0	1	2 x Tclk	67	385
1	0	0	0	1	2	4 x Tclk	385	1020
2	0	0	1	0	3	6 x Tclk	702	1655
3	0	0	1	1	4	8 x Tclk	1020	2290
4	0	1	0	0	5	10 x Tclk	1337	2925
5	0	1	0	1	6	12 x Tclk	1655	3560
6	0	1	1	0	7	14 x Tclk	1972	4194
7	0	1	1	1	8	16 x Tclk	2290	4829
8	1	0	0	0	9	18 x Tclk	2607	464
9	1	0	0	1	10	20 x Tclk	2925	6099
A	1	0	1	0	11	22 x Tclk	3242	6734
B	1	0	1	1	12	24 x Tclk	3560	7369
C	1	1	0	0	13	26 x Tclk	3877	8004
D	1	1	0	1	14	28 x Tclk	4194	8639
E	1	1	1	0	15	30 x Tclk	4512	9274
F	1	1	1	1	16	32 x Tclk	4829	9909

**Bit 7 CPS.** Conversion clock prescale

This bit defines the ADC conversion logic clock prescale

0 Fclk = CLK/1

1 Fclk = CLK/2

CLK = CPU clock frequency

**Bit 6 CONT\_RUN.** Continuous run

This bit determines whether the sequencer operates in continuous conversion mode or start-stop mode. This bit can be written while a current conversion sequence is active. This bit will take effect at the end of the current conversion sequence; i.e., software can set/clear this bit until EOS has occurred, for valid action to be taken. In the continuous conversion mode, there is no need to reset the sequencer; however, the sequencer must be reset in the start-stop mode to put the converter in state CONV00.

0 Start-stop mode. Sequencer stops after reaching EOS. This is used for multiple time-sequenced triggers.

1 Continuous conversion mode. After reaching EOS, the sequencer starts all over again from state CONV00 (for SEQ1 and cascaded) or CONV08 (for SEQ2).

**Bit 5 INT\_PRI.** ADC interrupt request priority

0 High priority

1 Low priority

**Bit 4 SEQ\_CASC.** Cascaded sequencer operation

This bit determines whether SEQ1 and SEQ2 operate as two 8-state sequencers or as a single 16-state sequencer (SEQ).

0 Dual-sequencer mode. SEQ1 and SEQ2 operate as two 8-state sequencers.

1 Cascaded mode. SEQ1 and SEQ2 operate as a single 16-state sequencer (SEQ).

**Bit 3 CAL\_ENA.** Offset calibration enable

When set to 1, CAL\_ENA disables the input channel multiplexer, and connects the calibration reference selected by the bits HI/LO and BRG\_ENA to the ADC core inputs. The calibration conversion can then be started by setting bit 14 of ADCTRL2 register (STRT\_CAL) to 1. Note that CAL\_ENA should be set to 1 first before the STRT\_CAL bit can be used.

0 Calibration mode disabled

1 Calibration mode enabled

**Bit 2 BRG\_ENA.** Bridge enable

Together with the HI/LO bit, BRG\_ENA allows a reference voltage to be converted in calibration mode. See the description of the HI/LO bit for reference voltage selections during calibration.

- 0 Full reference voltage is applied to the ADC input
- 1 A reference midpoint voltage is applied to the ADC input

**Bit 1 HI/LO.** VREFHI/VREFLO selection

When the fail self-test mode is enabled (STEST\_ENA = 1), HI/LO defines the test voltage to be connected. In calibration mode, HI/LO defines the reference source polarity; see Table 7–3. In normal operating mode, HI/LO has no effect.

- 0 VREFLO is used as precharge value at ADC input
- 1 VREFHI is used as precharge value at ADC input

**Bit 0 STEST\_ENA.** Self-test function enable

- 0 Self-test mode disabled
- 1 Self-test mode enabled

## 5.2. ADC Control Register 2 (ADCTRL2) — Address 70A1h

15	14	13	12	11	10	9	8
EVB_SOC_SEQ	RST_SEQ1/ STRT_CAL	SOC_SEQ1	SEQ1_BSY	INT_ENA_SEQ1 (Mode 1)	INT_ENA_SEQ1 (Mode 0)	INT_FLAG_SEQ1	EVA_SOC_SEQ1
RW-0	RS-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
7	6	5	4	3	2	1	0
EXT_SOC_SEQ1	RST_SEQ2	SOC_SEQ2	SEQ2_BSY	INT_ENA_SEQ2 (Mode 1)	INT_ENA_SEQ2 (Mode 0)	INT_FLAG_SEQ2	EVB_SOC_SEQ2
RW-0	RS-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

**Bit 15 EVB\_SOC\_SEQ.** EVB SOC enable for cascaded sequencer

- 0 No action
- 1 Setting this bit allows the cascaded sequencer to be started by an Event Manager B signal. The Event Manager can be programmed to start a conversion on various events. See chapter 6, Event Manager (EV), for details.

**Bit 14 RST\_SEQ1 / STRT\_CAL.** Reset Sequencer/Start Calibration**Case: Calibration Disabled (Bit 3 of ADCTRL1) = 0**

Writing a 1 to this bit will reset the sequencer immediately to an initial “pretriggered” state, i.e., waiting for a trigger at CONV00. A currently active conversion sequence will be aborted.

- 0 No action
- 1 Immediately reset sequencer to state CONV00

**Case: Calibration Enabled (Bit 3 of ADCTRL1) = 1**

Writing a 1 to this bit will begin the converter calibration process.

- 0 No action
- 1 Immediately start calibration process

**Bit 13 SOC\_SEQ1.** Start-of-conversion (SOC) trigger for Sequencer 1 (SEQ1). This

bit can be set by the following triggers:

- \_S/W – Software writing a 1 to this bit
- \_EVA – Event Manager A
- \_EVB – Event Manager B (only in cascaded mode)
- \_EXT – External pin (i.e., the ADCSOC pin)

When a trigger occurs, there are 3 possibilities:

**Case 1:** SEQ1 idle and SOC bit clear

SEQ1 starts immediately (under arbiter control). This bit is set and cleared, allowing for any “pending” trigger requests.

**Case 2:** SEQ1 busy and SOC bit clear



Bit is set signifying a trigger request is pending. When SEQ1 finally starts after completing current conversion, this bit will be cleared.

**Case 3:** SEQ1 busy and SOC bit set

Any trigger occurring in this case will be ignored (lost).

0 Clears a pending SOC trigger.

1 Software trigger – Start SEQ1 from currently stopped position (i.e., Idle mode)

**Bit 12 SEQ1\_BSY.** SEQ1 Busy

This bit is set to a 1 while the ADC autoconversion sequence is in progress.

It is cleared when the conversion sequence is complete.

0 Sequencer is Idle (i.e., waiting for trigger).

1 Conversion sequence is in progress.

**Bits 11–10 INT\_ENA\_SEQ1.** Interrupt-mode-enable control for SEQ1

Bit 11	Bit 10	Operation Description
0	0	Interrupt is Disabled
0	1	Interrupt <b>Mode 1</b> Interrupt requested immediately when INT_FLAG_SEQ1 flag is set
1	0	Interrupt <b>Mode 2</b> Interrupt requested only if INT_FLAG_SEQ1 flag is already set. If clear, INT_FLAG_SEQ1 flag is set and INT request is suppressed. (This mode allows Interrupt requests to be generated for every other EOS.)
1	1	Reserved

**Bit 9 INT\_FLAG\_SEQ1.** ADC interrupt flag bit for SEQ1

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event

1 An interrupt event has occurred.

**Bit 8 EVA\_SOC\_SEQ1.** Event Manager A SOC mask bit for SEQ1

0 SEQ1 cannot be started by EVA trigger.

1 Allows SEQ1 to be started by Event Manager A trigger. The Event Manager can be programmed to start a conversion on various events. See chapter 6, Event Manager (EV), for details.

**Bit 7 EXT\_SOC\_SEQ1.** External signal start-of-conversion bit for SEQ1

0 No action

1 Setting this bit enables an ADC autoconversion sequence to be started by a signal from the ADCSOC device pin.

**Bit 6 RST\_SEQ2.** Reset SEQ2

0 No action

1 Immediately resets SEQ2 to an initial “pretriggered” state, i.e., waiting for a trigger at CONV08. A currently active conversion sequence will be aborted.

**Bit 5 SOC\_SEQ2.** Start-of-conversion trigger for Sequencer 2 (SEQ2)

This bit can be set by the following triggers:

\_S/W – Software writing of 1 to this bit

\_EVB – Event Manager B

When a trigger occurs, there are 3 possibilities:

**Case 1:** SEQ2 idle and SOC bit clear

SEQ2 starts immediately (under arbiter control) and the bit is cleared, allowing for any “pending” trigger requests.

**Case 2:** SEQ2 busy and SOC bit clear

Bit is set signifying a trigger request is pending. When SEQ2 finally starts after Completing current conversion, this bit will be cleared.

**Case 3:** SEQ2 busy and SOC bit set

Any trigger occurring in this case will be ignored (lost).

0 Clears a Pending SOC trigger.

1 Software trigger – Start SEQ2 from currently stopped position (i.e., Idle mode).

**Bit 4 SEQ2\_BSY.** SEQ2 Busy

This bit is set to a 1 while the ADC autoconversion sequence is in progress.

It is cleared when the conversion sequence is complete.

0 Sequencer is idle (i.e., waiting for trigger).

1 Conversion sequence is in progress.

**Bits 3–2 INT\_ENA\_SEQ2.** Interrupt-mode-enable control for SEQ2.

<b>Bit 3</b>	<b>Bit 2</b>	<b>Operation Description</b>
0	0	Interrupt is Disabled
0	1	Interrupt <b>Mode 1</b> Interrupt requested immediate on INT_FLAG_SEQ2 flag set
1	0	Interrupt <b>Mode 2</b> Interrupt requested only if INT_FLAG_SEQ2 flag is already set. If clear, INT_FLAG_SEQ2 flag is set and INT request is suppressed. (This mode allows Interrupt requests to be generated for every other EOS)
1	1	Reserved

**Bit 1 INT\_FLAG\_SEQ2.** ADC interrupt flag bit for SEQ2

This bit indicates whether an interrupt event has occurred or not. This bit must be cleared by the user writing a 1 to it.

0 No interrupt event.

1 An interrupt event has occurred.

**Bit 0 EVB\_SOC\_SEQ2.** Event Manager B SOC mask bit for SEQ2

0 SEQ2 cannot be started by EVB trigger.

1 Allows SEQ2 to be started by Event Manager B trigger. The Event Manager can be programmed to start a conversion on various events. See chapter 6, Event Manager (EV), for details.

## ***Bibliographie***

- [1] **R. E. Kalman**, « A New Approach to Linear Filtering and Prediction Problems », Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45. Copyright © 1960 by ASME.
- [2] **M. GEVERS et L. VANDENDORPE**, « Processus stochastiques, estimation et prédiction » Université catholique de Louvain Faculté des Sciences appliquées.
- [3] **M. NAJIM**, « Modilisation et identification en traitement de signal », édition MASSON 1988.
- [4] **Frédéric ROTELLA**, « Filtrage Optimal », Ecole Nationale d'Ingénieurs de Tarbes.
- [5] **Bilal AKIN**, « State estimation technique for speed sensorless field oriented control of induction motors », a thesis of MASTER, the middle east technical university, department of electrical and electronics engineering 2003.
- [6] **Hermann GILLES**, « Asservissement visuel d'une tête robotique par un filtre de Kalman », Rapport de DEA, université de Haut Alsace 2000.
- [7] **M.A. OUHROUCHE**, « EFK-based on-line tuning of rotor time-constant in an induction motor vector control », International Journal of Power and Energy Systems 2000.
- [8] **M.A. OUHROUCHE**, « Estimation of speed, rotor flux and rotor resistance in cage induction motor using the EKF algorithm », International Journal of Power and Energy Systems 2002.
- [9] **Brett HOVINGH, W.W.L KEERTHIPALA et Wei-Yong YAN**, « Sensorless Speed Estimation of an Induction Motor in a Field Orientated Control System », School of Electrical and Computer Engineering Curtin University of Technology, Australia.

- [10] **Sang-uk Kim, Ice-woo Yang, Eul-j ae Lee, Young-bong Kim, Jong-Tai Lee, and Young-seek Kim**, « Robust Speed Estimation for Speed Sensorless Vector Control of Induction Motors », IEEE 1999.
- [11] **Gabriel BUCHE**, « Commande de la machine asynchrone en environnement temps réel MATLAB/SIMULINK » mémoire d'ingénieur C.N.A.M 2001, conservatoire national des arts et métiers, Centre régional associé de Grenoble.
- [12] **Guy STURTZER et Eddie SMIGIEL**, « Modélisation et commande des moteurs triphasés », édition ellipses.
- [13] **Emmanuel SEDDA**, « Estimation en ligne de l'état et des paramètres d'une machine asynchrone par filtrage à erreur bornée et par filtrage de Kalman ». Thèse de Doctorat en Génie Electrique de Université Paris 6 1998.
- [14] **Michael FILIPPICH**, « Digital Control of a Three Phase Induction Motor », thesis of Electrical Engineering, University of Queensland 2002.
- [15] **Keliang Zhou and Danwei Wang**, « Relationship Between Space-Vector Modulation and Three-Phase Carrier-Based PWM: A Comprehensive Analysis », IEEE 2002.
- [16] **Claude RICHARD**, « Simulation des machines électriques en régime dynamique à l'aide du logiciel SIMULINK »
- [17] **Greg WELCH and Gary BISHOP**, « An Introduction to the Kalman Filter », Department of Computer Science University of North Carolina at Chapel Hill 2003.
- [18] **Texas Instruments**, « TMS320LF/LC240x DSP Controllers Reference Guide », Literature Number: SPRU357 January 2000
- [19] **Spectrum Digital**, « TMS320LF2407 Evaluation Module Technical Reference », DSP Development Systems 2000
- [20] MATLAB User's Guide, 1998.
- [21] **Texas Instruments**, « TMS320F/C240 DSP Controllers Reference Guide Peripheral Library and Specific Devices », Literature Number: SPRU161C June 1999.

- [22] **Texas Instruments Europe**, « Implementation of a Speed Field Orientated Control of Three Phase AC Induction Motor using TMS320F240 », Literature Number: BPRA076 March 1998.
- [23] **Gubae KANG, Junha KIM and Kwanghee NAM**, « Parameters estimation of linear induction motor with PWM inverter », IEEE 2001.
- [24] **Paul BILDSTEIN**, « Filtres Actifs », Technique de l'ingénieur E3 I « Electronique », Article N° E3130.
- [25] **Paul BILDSTEIN**, « Filtres Actifs, Données numérique », Technique de l'ingénieur E3 I « Electronique », Article N° E3132.
- [26] **RS** , « Hall effect transducers, current and voltage», Data sheet 232-5250, 1997.
- [27] **Brian W. KERNIGHAN and Dennis M. RITCHIE** « The C programming language», Prentice-Hall 1988.
- [28] **Texas Instruments**, « TMS 320 C2xx / C24x Code Composer User's Guide », Literature Number: SPRU 490 October 2000.