

**UNIVERSITE SAAD DAHLAB DE BLIDA**

**Faculté des Sciences de l'Ingénieur**

Département d'Electronique

## **MEMOIRE DE MAGISTER**

En électronique

Spécialité : Contrôle

### **IMPLEMENTATION SUR UN CIRCUIT FPGA DU FILTRE DE KALMAN (FK) POUR LA MACHINE ASYNCHRONE TRIPHASEE**

Par

**OUNISSI Oussama**

Devant le jury composé de :

M.Bounekhla	Professeur	USD-Blida	Président
H.Salhi	Professeur	USD-Blida	Examineur
K.Benmansour	Maitre de conférences A	UYF-Médéa	Examineur
A.Ferdjouni	Maitre de conférences A	USD-Blida	Promoteur

Blida, Février 2012

## RESUME

Les circuits FPGA interviennent ces dernières années dans presque tous les domaines de recherche grâce à leur reconfigurabilité, leur grande fréquence de traitement ainsi que leur avantage de parallélisme. Dans le présent travail, Ils sont utilisés pour l'implémentation du filtre de Kalman afin de reconstruire les flux rotoriques de la machine asynchrone qui sont inaccessibles à la mesure en se basant sur un modèle mathématique adéquat, ces grandeurs étant essentielles pour la commande et pour le diagnostic de la machine.

Des méthodes méta-heuristiques basées sur des Algorithmes Génétiques sont utilisées pour le pré réglage du filtre Kalman ainsi que pour l'identification des paramètres de la MAS.

## **ABSTRACT**

FPGAs are involved in the last years in almost all domain of research due to their reconfigurability, their high frequency of treatment and their advantage of parallelism. In the present work, they are used for the implementation of the Kalman filter to reconstruct the flux of the asynchronous machine rotor which is inaccessible to measurement using an appropriate mathematical model; these quantities are essentials for control and diagnosis of the machine.

Meta-heuristic methods based on Genetic Algorithms are used to adjusting the Kalman filter and to identify the parameters of the asynchronous machine.

## ملخص

تشارك FPGAs في السنوات الأخيرة في جميع مجالات البحث تقريبا بسبب إمكانية إعادة برمجتها، وتيرتها في معالجة المعلومات وكذا قدرتها على اعتماد مبدأ التوازي في التصميم. وقد استخدمت في هذا البحث لتصميم مرشح كالمان لإعادة تشكيل التدفقات غير القابلة للقياس في الجزء المتحرك من الآلة غير المتزامنة بالاعتماد على نموذج رياضي مناسب، و التي تعتبر ضرورية من أجل التحكم في هذه الآلة وتشخيصها.

كما استخدمنا أساليب حديثة تعتمد على الخوارزميات الجينية لضبط مرشح كالمان وتحديد خصائص الآلة.

## REMERCIEMENTS

Je tiens tout d'abord à exprimer ma profonde reconnaissance et mes sincères remerciements au Professeur H.Salhi, Directeur du laboratoire LABSET pour m'avoir accueilli au sein de son laboratoire. Je remercie également mon promoteur M<sup>r</sup> Ferdjouni Abdelaziz et le co-promoteur M<sup>e</sup>, Elmeraoui Khadidja pour leurs précieux conseils, leurs suivi ainsi que leurs patience.

Je tiens à témoigner toute ma gratitude à Mr Chikhi M.Lazhar pour sa disponibilité et son aide.

J'adresse également des remerciements très particuliers aux membres du jury, qui m'ont fait l'honneur de participer au jugement de ce travail.

Je réserve une pensée toute particulière à mes chers parents qui m'ont toujours soutenu et guidé tout au long de mes études.

## TABLE DES MATIERES

RESUME	0
REMERCIEMENTS	3
TABLE DES MATIERES	4
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	8
INTRODUCTION	11
CHAPITRE 1	14
1.1. Introduction .....	14
1.2. Présentation de la MAS .....	15
1.2.1. Construction .....	15
1.2.2. Principe de fonctionnement .....	17
1.2.3. Caractéristiques mécaniques .....	18
1.3. Hypothèses simplificatrices .....	19
1.4. Modélisation de la MAS .....	19
1.4.1. Repère triphasé .....	19
1.4.2. Repère diphasé .....	21
1.4.3. Choix du vecteur d'état .....	23
1.4.4. Représentation d'état .....	23
1.4.5. Equation mécanique .....	25
1.4.6. Choix du référentiel .....	26
1.5. Identification des paramètres de la MAS .....	28
1.5.1. Modèle de référence .....	29
1.5.2. Modèle à identifier .....	29
1.5.3. Algorithmes génétiques .....	29
1.5.4. Résultats de l'identification .....	33
1.6. Observabilité de la MAS .....	34
1.7. Conclusion .....	37
CHAPITRE 2	38
2.1. Introduction .....	38
2.2. Hypothèses simplificatrices .....	39

2.3. Présentation du filtre de Kalman .....	39
2.3.1. Equation générale du filtre de Kalman.....	40
2.3.2. Calcule d'erreur .....	41
2.3.3. Estimateur non biaisé .....	41
2.3.4. Calcul du gain de Kalman.....	42
2.4. Discrétisation du filtre de Kalman.....	44
2.4.1. Equations récurrentes du système discrétisé .....	44
2.4.2. Bruits d'état et de mesure discrétisés .....	45
2.4.3. Equations récurrentes du filtre de Kalman.....	46
2.5. Extension du filtre de Kalman .....	47
2.6. Application du filtre de Kalman a la mas .....	48
2.6.1. Modèle d'état continu .....	48
2.6.2. Discrétisation du modèle continu.....	48
2.6.3. Préréglage du filtre de Kalman .....	49
2.6.4. Algorithme du filtre de Kalman.....	50
2.7. Conclusion .....	53
CHAPITRE 3 .....	54
3.1. Introduction .....	54
3.2. Les systèmes à microprocesseurs .....	55
3.2.1. Architecture de Von Neumann.....	56
3.2.2. Architecture de Harvard.....	56
3.3. Circuits logiques programmables.....	57
3.3.1. SPLD .....	58
3.3.2. CPLD.....	59
3.3.3. FPGAs.....	59
3.4. Familles des FPGA de Xilinx.....	60
3.5. Langage de description matériel .....	60
3.6. Etapes de conception d'un projet sur FPGA .....	61
3.6.1. Description du circuit .....	61
3.6.2. Simulation comportementale .....	62
3.6.3. Synthèse .....	63
3.6.4. Simulation fonctionnelle.....	63
3.6.5. Implémentation .....	63
3.6.6. Simulation temporelle .....	63

3.6.7. Configuration .....	64
3.7. Applications des FPGA .....	64
3.8. Apport des FPGA aux problèmes de contrôle .....	65
3.9. Conclusion .....	66
CHAPITRE 4 .....	67
4.1. Introduction .....	67
4.2. Représentation des données .....	67
4.3. Troncature des résultats de multiplications .....	69
4.4. Synchronisation des signaux internes.....	70
4.5. Présentation de l'architecture.....	71
4.5.1. Prétraitement des signaux d'entrée .....	71
4.5.2. Transformation de Park .....	73
4.5.3. calcul de la matrice d'état .....	74
4.5.4. Filtrage de Kalman .....	76
4.5.5. Transformation de Park inverse.....	88
4.6. Présentation des résultats.....	89
4.6.1. Résultats de Simulation.....	90
4.6.2. Résultats de Cosimulation HDL.....	91
4.6.3. Résultats de Synthèse.....	94
4.7. Conclusion .....	98
CONCLUSION .....	99
APPENDICE A .....	101
APPENDICE B .....	103
B.1. Bloc d'Entrées/Sorties (IOB).....	104
B.2. Bloc Logique Configurable (CLB) .....	104
B.3. Bloc de mémoire vive (RAM Bloc) .....	106
B.4. Cellule de gestion d'Horloges (CMT).....	106
B.4.1. Gestionnaire d'Horloge Digitale (DCM).....	107
B.4.2. Boucle à Verrouillage de Phase (PLL) .....	107
B.5. Bloc de traitement de signal (DSP48E) .....	107
B.6. Réseau d'interconnexions .....	108
APPENDICE C .....	109
C.1.Détermination de RS .....	110
C.2.Détermination de Rf, Ls, J, f .....	110



C.2.1.Essai à vitesse constante .....	110
C.2.2.Essai au ralentissement.....	111
C.3.Détermination de Rr et M .....	112
C.3.1.Essai à rotor bloqué .....	112
C.3.2.Essai à la vitesse nominale.....	113
REFERENCES	115

## LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

FIGURE 1.1: STATOR DE LA MACHINE ASYNCHRONE .....	15
FIGURE 1.2: ROTOR A CAGE D'ECUREUIL D'UNE MACHINE ASYNCHRONE .....	16
FIGURE 1.3: ENTREFER D'UNE MACHINE ASYNCHRONE .....	16
FIGURE 1.4: CHAMP MAGNETIQUE GENERE DANS UNE BOBINE PAR UN COURANT ALTERNATIF.....	17
FIGURE 1.5 : CHAMP MAGNETIQUE TOURNANT DANS LA MAS.....	17
FIGURE 1.6 : CARACTERISTIQUE MECANIQUE DE LA MAS.....	18
FIGURE 1.7: IDENTIFICATION PAR MODELE DE REFERENCE .....	29
FIGURE 1.8 : ORGANIGRAMME GENERAL D'UN AG .....	30
FIGURE 1.1.9 : COMPARAISON ENTRE LE SIGNAL DE REFERENCE ET CELUI OBTENU PAR LE MODELE .....	33
FIGURE 1.10 : EVOLUTION DE L'ERREUR QUADRATIQUE ENTRE LE SIGNAL DE REFERENCE ET CELUI OBTENU PAR LE MODELE .....	33
FIGURE 2.1 : ORGANIGRAMME DU FILTRE DE KALMAN .....	40
FIGURE 2.2 : SCHEMA BLOC D'UN OBSERVATEUR.....	44
FIGURE 2.3: ALGORITHME DE FILTRAGE DE KALMAN .....	47
FIGURE 3.1: ARCHITECTURE DE VON NEUMANN .....	56
FIGURE 3.2 : ARCHITECTURE DE HARVARD .....	57
FIGURE 3.3 : CLASSIFICATION DES CIRCUITS LOGIQUES PROGRAMMABLES.....	58
FIGURE 3.4 : ARCHITECTURES DES SPLD .....	58
FIGURE 3.5 : ARCHITECTURE D'UN CPLD.....	59
FIGURE 3.6 : ETAPES DE CONCEPTION D'UN PROJET FPGA .....	62
FIGURE 4.1 : REPRESENTATION AVEC UN FORMAT $D + Q/Q$ .....	68
FIGURE 4.2 : SCHEMA BLOC DE L'ARCHITECTURE COMPLETE .....	71
FIGURE 4.3 : AIGUILLAGE DES SIGNAUX .....	72
FIGURE 4.4 : ELIMINATION D'UNE COMPOSANTE CONTINUE PRE-CALCULEE .....	72
FIGURE 4.5 : TRANSFORMATION DE PARK.....	73
FIGURE 4.6 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F01 .....	74
FIGURE 4.7 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F02 .....	75

FIGURE 4.8 : MISE A JOUR DE LA MATRICE D'ETAT .....	75
FIGURE 4.9 : SCHEMA BLOC DU FILTRE DE KALMAN .....	76
FIGURE 4.10 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F1 .....	77
FIGURE 4.11 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F2 .....	77
FIGURE 4.12 : STRUCTURE LOGIQUE DE LA PREMIERE EQUATION DU SYSTEME D'ETAT ...	78
FIGURE 4.13 : STRUCTURE LOGIQUE DE LA DEUXIEME EQUATION DU SYSTEME D'ETAT ...	78
FIGURE 4.14 : STRUCTURE LOGIQUE DE LA TROISIEME EQUATION DU SYSTEME D'ETAT ..	78
FIGURE 4.15 : STRUCTURE LOGIQUE DE LA QUATRIEME EQUATION DU SYSTEME D'ETAT .	79
FIGURE 4.16 : STRUCTURE LOGIQUE DE L'EQUATION D'ETAT DU SYSTEME .....	79
FIGURE 4.17 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F3 .....	81
FIGURE 4.18 : REPRESENTATION DE LA CELLULE ELEMENTAIRE F4 .....	81
FIGURE 4.19 : STRUCTURE LOGIQUE DE L'EQUATION DE PP1 .....	82
FIGURE 4.20 : STRUCTURE LOGIQUE DE L'EQUATION DE PP2 .....	82
FIGURE 4.21 : STRUCTURE LOGIQUE DE L'EQUATION DE PP3.....	83
FIGURE 4.22 : STRUCTURE LOGIQUE DE L'EQUATION DE PP4.....	83
FIGURE 4.23 : STRUCTURE LOGIQUE DE L'EQUATION DE COVARIANCE DE L'ERREUR DE PREDICTION .....	84
FIGURE 4.24 : STRUCTURE LOGIQUE DE L'EQUATION DE L'ERREUR .....	85
FIGURE 4.25 : STRUCTURE LOGIQUE DE L'EQUATION DU GAIN DE KALMAN .....	86
FIGURE 4.26 : STRUCTURE LOGIQUE DE L'EQUATION DE CORRECTION DE L'ETAT.....	87
FIGURE 4.27 : STRUCTURE LOGIQUE DE L'EQUATION DE LA COVARIANCE DE L'ERREUR D'ESTIMATION.....	88
FIGURE 4.28 : TRANSFORMATION DE PARK INVERSE.....	89
FIGURE 4.29 : SIMULATION FONCTIONNELLE DU FILTRE DE KALMAN PAR LE MODELSIM .	90
FIGURE 4.30 : SCHEMA BLOC DE COSIMULATION HDL DU FILTRE DE KALMAN.....	91
FIGURE 4.31 : RESULTATS DE COSIMULATION HDL (PREMIER COURANT STATORIQUE) ..	92
FIGURE 4.32 : RESULTATS DE COSIMULATION HDL (DEUXIEME COURANT STATORIQUE)	92
FIGURE 4.33 : RESULTATS DE COSIMULATION HDL (TROISIEME COURANT STATORIQUE) .....	92
FIGURE 4.34 : RESULTATS DE COSIMULATION HDL (PREMIER FLUX ROTORIQUE).....	93
FIGURE 4.35 : RESULTATS DE COSIMULATION HDL (DEUXIEME FLUX ROTORIQUE) .....	93
FIGURE 4.36 : RESULTATS DE COSIMULATION HDL (TROISIEME FLUX ROTORIQUE) .....	93
FIGURE 4.37 : INTERFACE GRAPHIQUE DE XILINX-ISE .....	94
FIGURE 4.38 : SCHEMA RTL DU BLOC DE PRETRAITEMENT DES ENTREES (MFORME) ....	95

FIGURE 4.39 : SCHEMA RTL DU BLOC DE TRANSFORMATION DE PARK (PARK).....	95
FIGURE 4.40 : SCHEMA RTL DU BLOC PRINCIPAL (KFILTER) .....	96
FIGURE 4.41 : SCHEMA RTL DU BLOC DE TRANSFORMATION DE PARK INVERSE (IPARK).....	96
FIGURE 4.42 : RAPPORT DE SYNTHÈSE – UTILISATION DES RESSOURCES.....	97
FIGURE 4.43 : RAPPORT DE SYNTHÈSE – RAPPORT TEMPOREL .....	97
FIGURE B.1: ARCHITECTURE INTERNE D'UN FPGA.....	103
FIGURE B.2: DIAGRAMME DE BLOC IOB DE BASE.....	104
FIGURE B.3: ARCHITECTURE D'UN CLB.....	105
FIGURE B.4 : REPRESENTATION SIMPLIFIÉE D'UN SLICE DE VIRTEX5 .....	105
FIGURE B.5 : REPRESENTATION D'UNE RAM BLOC.....	106
FIGURE B.6: DIAGRAMME DE BLOC CMT DE VIRTEX5 .....	106
FIGURE B.7: DIAGRAMME DE BLOC SIMPLIFIÉE D'UNE CELLULE DSP48E DE VIRTEX5 ...	107
FIGURE B.8: MATRICE DE ROUTAGE (SM).....	108
FIGURE C.1 : SCHEMA MONOPHASE EQUIVALENT DE LA MAS .....	109
FIGURE C.2 : ESSAI A COURANT CONTINU .....	110
FIGURE C.3 : ESSAI A VIDE .....	110
FIGURE C.4 : COURBE DE RALENTISSEMENT DE LA MACHINE A PARTIR DE LA VITESSE NOMINALE .....	112
FIGURE C.5 : ESSAI A ROTOR BLOQUE .....	113

## INTRODUCTION

Les moteurs électriques sont devenus incontournables dans le monde actuel. Ils sont utilisés dans les applications industrielles et domestiques. La machine asynchrone (MAS) triphasée est la machine employée le plus fréquemment dans l'industrie. Elle possède en effet plusieurs avantages : simplicité, robustesse, prix peu élevé et entretien facile. Elle est utilisée aujourd'hui dans de nombreuses applications industrielles telles que transport, machines-outils... ou domestiques telles que lave-linge, robots ménagers, ventilateurs...

La machine asynchrone est un système physique non linéaire et fortement couplé qui permet la conversion de l'énergie électrique en énergie mécanique et inversement. Lorsqu'elle est branchée directement sur le réseau électrique, elle fonctionne à vitesse fixe. Son alimentation à l'aide d'un onduleur permet de varier sa vitesse. L'obtention de performances élevées, en termes de vitesse et de couple, nécessite des commandes sophistiquées telles que la commande à orientation de flux (FOC), la commande directe du couple (DTC) ou la commande linéarisante, [1, 2, 3]. La mise en œuvre de ce type de commande nécessite la connaissance des grandeurs qui ne sont pas accessibles à la mesure. Ceci a conduit à la synthèse de différents types d'observateurs [4, 5, 6]. Ces derniers sont aussi nécessaires pour le diagnostic de la MAS [7, 8].

Les observateurs d'états peuvent être considérés comme des capteurs logiciels. Ils permettent de reconstruire les informations internes (états ou paramètres inconnus) d'un système dynamique à partir de son modèle, de ses entrées connues et de ses sorties mesurées.

Les observateurs peuvent être linéaires ou non linéaires, comme ils peuvent être aussi stochastiques ou déterministes.

Le filtre de Kalman (**FK**) est l'observateur le plus souvent utilisé, depuis son apparition. Il occupe une place privilégiée dans l'industrie et dans la recherche scientifique. En plus des domaines technologiques, il trouve ces applications dans d'autres domaines comme l'hydrologie [9], l'économétrie [10], les statistiques [11].

Le filtre de Kalman est appliqué à la machine asynchrone pour reconstruire les flux rotoriques dans le cas où la vitesse mécanique est mesurée. Sa version étendue peut être utilisée lorsqu'on veut s'affranchir du capteur mécanique [12, 13, 14, 16, 23]. Elle peut également être utilisée pour suivre les variations de la résistance statorique [15] ou rotorique [5].

La mise en œuvre de l'algorithme du FK nécessite des opérations arithmétiques complexes faisant intervenir la multiplication, l'accumulation, la division, l'inversion matricielle..., ce qui explique la difficulté de l'implémentation de cet algorithme sur un calculateur numérique.

Le **DSP** (*Digital Signal Processor*) a constitué pendant de longues années la solution la plus convenable pour son implémentation [4, 17]. La nécessité à des performances plus élevées (réduction du temps de calcul, Implémentation de plusieurs systèmes sur une même puce et reconfiguration dynamique) a poussé les chercheurs à utiliser des circuits logiques programmables (**FPGA**) [1, 2, 15], [18, 19].

Les **FPGA** (*Field Programmable Gate Arrays*) sont couramment utilisés sur les cartes électroniques pour assurer des fonctions traditionnelles de logique câblée. La technologie a beaucoup évolué ces derniers temps et les FPGA sont devenus aujourd'hui de véritables processeurs numériques des signaux, qui viennent concurrencer les composants DSP et ASIC.

Les FPGA sont aussi beaucoup plus accessibles. Il s'agit en effet de composants standard fabriqués en masse et les coûts initiaux sont répartis sur l'ensemble des utilisateurs. Comme tout composant standard, ils peuvent être achetés en petites quantités.

L'objectif de notre travail est l'utilisation des FPGA pour l'implémentation du filtre de Kalman pour la reconstruction des flux rotoriques de la machine asynchrone. Cette implémentation peut être effectuée en tenant compte de deux critères primordiaux; le premier est la minimisation du temps de calcul et le deuxième est l'optimisation de ressources matérielles utilisées. Un autre paramètre important est le nombre de bits utilisés pour coder les différentes grandeurs. La modularité de l'architecture du FK est une contrainte supplémentaire. En effet, elle permet d'ajuster le niveau de parallélisme pour trouver un judicieux compromis entre le temps de calcul et l'utilisation de ressources.

Le premier chapitre de ce mémoire a été consacré à la présentation générale de la machine asynchrone ainsi que la démonstration du modèle mathématique. L'identification des paramètres de la machine est une étape indispensable. Deux méthodes principales peuvent être utilisées pour la faire ; la méthode classique basée sur les essais expérimentaux, et la méthode numérique basée sur un modèle de référence. Les deux méthodes sont présentées mais seule la deuxième a été exploitée dans ce travail. Ce chapitre se termine par la présentation de quelques définitions de l'observabilité des systèmes non linéaires avant l'étude de l'observabilité de la MAS.

Dans le deuxième chapitre, les équations du filtre de Kalman (FK) ont été démontrées, puis une discrétisation du modèle de Kalman a été introduite de manière à pouvoir l'implémenter dans un calculateur numérique. Ce modèle a été appliqué à la MAS et un algorithme efficace a été proposé pour la reconstruction des flux rotoriques. Le préréglage du filtre se fait par un algorithme génétique.

Le troisième chapitre est consacré aux circuits logiques programmables (FPGA). Il présente les différentes familles de Xilinx, leurs utilisations ainsi que leur apport aux problèmes de contrôle.

Le quatrième chapitre présente les différentes techniques utilisées pour l'implémentation du filtre de Kalman ensuite, une description détaillée a été présentée. Ce chapitre se termine par la présentation des différents résultats obtenus ainsi que les rapports de synthèse du projet.

Finalement nous terminons par une conclusion générale sur l'ensemble de cette étude ainsi que les perspectives de travail.

# CHAPITRE 1

## MODELISATION ET IDENTIFICATION DE LA MACHINE ASYNCHRONE

### 1.1.Introduction

Une machine électrique est un dispositif électromécanique permettant la conversion d'énergie électrique en travail ou énergie mécanique. Connue sous le terme «anglo-saxon» de machine à induction, La **machine asynchrone (MAS)** est une machine à courant alternatif sans contact physique entre le stator et le rotor [21]. Le terme asynchrone provient du fait que la vitesse de cette machine n'est pas nécessairement proportionnelle à la fréquence des courants qui la traversent.

La machine asynchrone triphasée, dont la puissance varie de quelques watts à plusieurs mégawatts est la machine la plus répandue dans l'industrie. Elle est robuste, fiable, normalisée, économique et peu encombrante [7]. Mais l'emploi de ce type de machine est évité en très forte puissance car la consommation de puissance réactive peut devenir alors un handicap.

Associés à des onduleurs de tension ou de courant, les machines asynchrones peuvent fonctionner à vitesse variable avec des performances de couple élevées. Pour aboutir à ces performances, différentes techniques de commande ont été utilisées nécessitant des modèles mathématiques convenables [1, 2, 18, 19, 24]. La détection de défauts et le diagnostic se font aussi avec des modèles appropriées prenant en compte les défauts de la machine. Plusieurs approches ont été présentés dans de différents travaux [7,15,25].

La machine asynchrone est un système physique non linéaire et fortement couplé. Sous certaines hypothèses simplificatrices, l'utilisation de la transformation de PARK, permet de contourner la difficulté à modéliser ce système. Elle permet une représentation biphasée équivalente de la machine triphasée ce qui réduit



considérablement la complexité du modèle [2, 21, 24]. L'utilisation de ce modèle nécessite la détermination des paramètres physiques de la machine. Un certain nombre de mesures sont alors effectuées. L'exploitation des données collectées, soit par les méthodes classiques soit par celle utilisant un modèle de référence conduit à l'identification de la MAS [26, 27, 28, 29]. La modélisation et l'identification de la MAS feront l'objet de ce chapitre.

## 1.2.Présentation de la MAS

### 1.2.1.Construction

Une machine asynchrone se caractérise par le fait qu'elle est constituée d'un stator (**inducteur**) alimenté en courant alternatif et d'un rotor (**induit**) soit en court-circuit (**cage**), soit bobiné dans lequel le courant est créé par induction.

- Le **stator** est la partie fixe liée à la carcasse, il est constitué de trois enroulements identiques décalés de  $(2\pi/3)$  radians les uns par rapport aux autres. L'ensemble de ces enroulements constituent le bobinage statorique. Les conducteurs de ces enroulements sont logés dans les encoches à la périphérie du stator. Lorsqu'ils sont alimentés par un réseau triphasé, ces enroulements créent un champ tournant dans l'entrefer de la machine. Le stator de la MAS est représenté dans la figure 1.1.



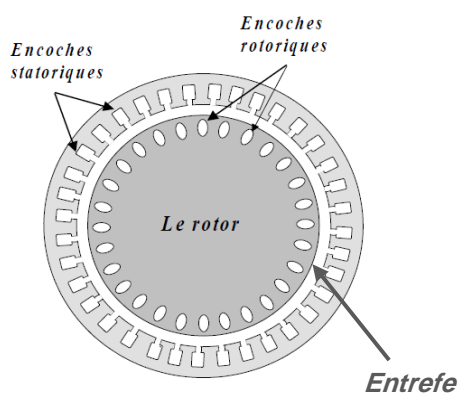
Figure 1.1: Stator de la Machine asynchrone

- Le **rotor**, c'est la partie mobile qui se place dans le centre du stator. Dans un rotor bobiné, plusieurs disques ferromagnétiques sont empilés sur l'arbre. Dans ses encoches tournées vers l'entrefer est logé un bobinage polyphasé accessible de l'extérieur de la machine via un collecteur simplifié à trois bagues, et court-circuité en fonctionnement normal. Mais on peut modifier les propriétés électromécaniques de la machine en agissant sur le rotor par ces connexions. Dans le cas d'un rotor à cage d'écurieil (figure 1.2) des barres conductrices (généralement en Aluminium) sont placées dans les encoches et court-circuitées aux deux extrémités par des connexions conductrices. Ce système présente de nombreux avantages de simplicité, robustesse et faible prix qui expliquent qu'il représente le type de moteur le plus largement utilisé.



**Figure 1.2: Rotor à cage d'écurieil d'une machine asynchrone**

- l'**entrefer** est la partie d'air entre le stator et le rotor (figure 1.3). Il doit être aussi étroit que possible pour éviter les pertes de flux.

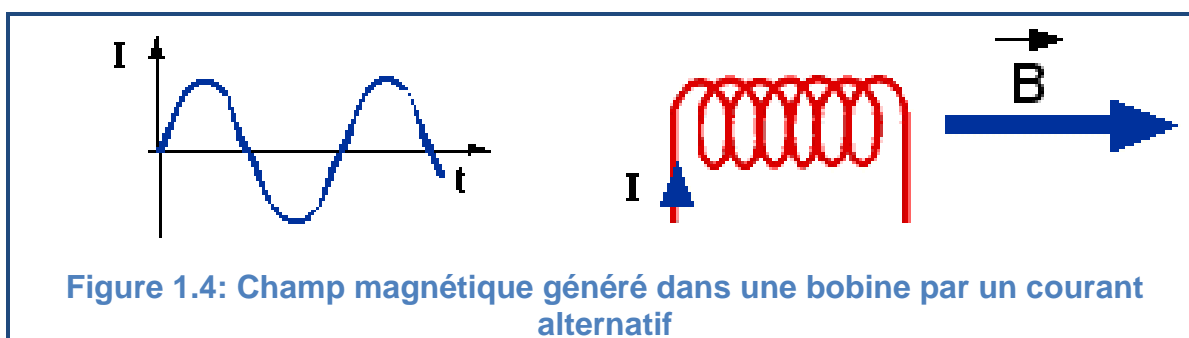


**Figure 1.3: Entrefer d'une machine asynchrone**

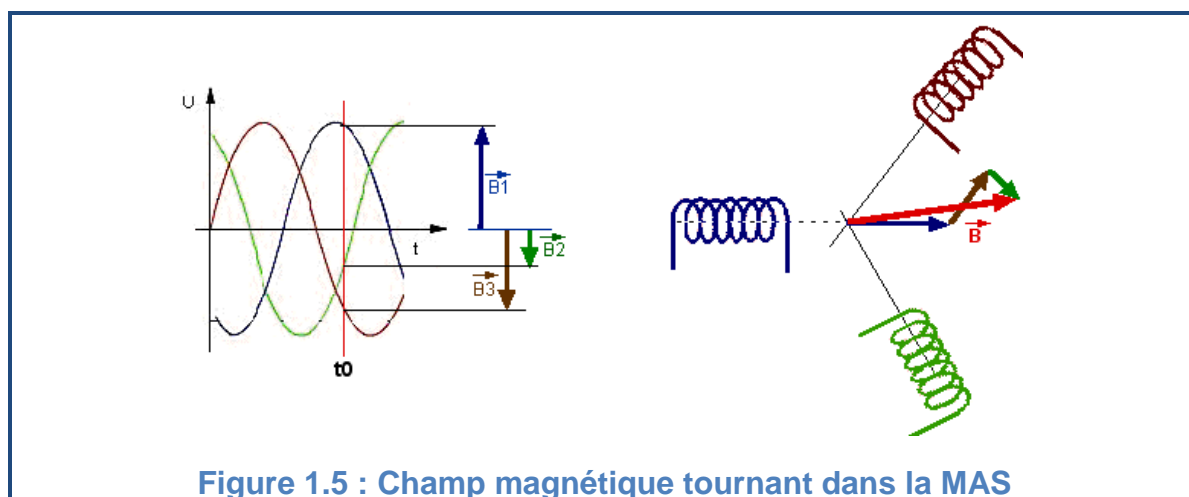
### 1.2.2.Principe de fonctionnement

La circulation d'un courant dans une bobine crée un champ magnétique. Ce champ est dans l'axe de la bobine, sa direction et son intensité sont fonction du courant.

Si le courant est alternatif, le champ magnétique varie en sens et en amplitude à la même fréquence que le courant.



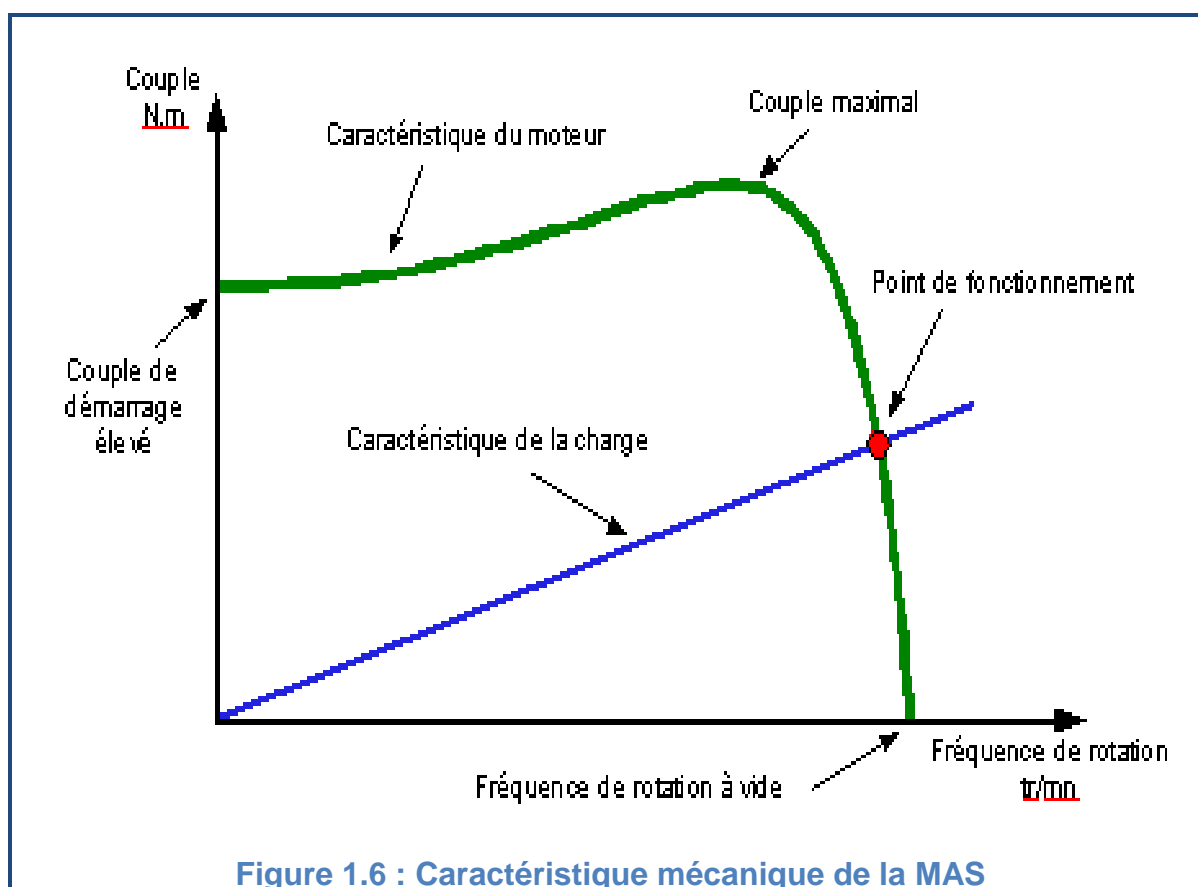
Dans le cas du moteur triphasé, les trois bobines sont disposées dans le stator à  $(2\pi/3)$  les unes des autres, trois champs magnétiques sont ainsi créés. Compte-tenu de la nature du courant sur le réseau triphasé, les trois champs sont déphasés et chacun à son tour passe par un maximum. Le champ magnétique résultant (**champ tournant**) est la somme vectorielle des trois autres (figure 1.5). Il tourne à la même vitesse  $\omega_s$  (**vitesse de synchronisme**) que le courant soit **50 tr/s (3000 tr/mn)**. Si le bobinage du stator est constitué de plusieurs pôles (**p**), cette vitesse sera divisée par le nombre de paires de pôles (**p**).



Le rotor est balayé par ce champ magnétique tournant. Les conducteurs sont alors traversés par un courant de Foucault induit. Des courants circulent dans les anneaux formés par la cage, les forces de Laplace (**forces électromotrices**) qui en résultent exercent un couple sur le rotor. D'après la loi de Lenz les courants induit s'opposent par leurs effets à la cause qui leur a donné naissance. Le rotor tourne alors dans le même sens que le champ tournant mais avec une vitesse  $\omega_m$  (**vitesse mécanique**) légèrement inférieure à la vitesse de synchronisme de ce dernier, ce qui explique le nom asynchrone.

### 1.2.3. Caractéristiques mécaniques

La caractéristique mécanique (couple moteur en fonction de la vitesse :  $C_{em} = f(\omega_m)$ ) est donnée pour une machine en fonction de la tension nominale et de la fréquence d'alimentation. La charge possède elle aussi une caractéristique mécanique :  $C_r = f(\omega_m)$ . Les deux courbes sont représentées dans la **figure 1.6**



### 1.3.Hypothèses simplificatrices

Pour établir des relations liant les tensions d'alimentation de la machine aux courants traversant les bobinages, il faut s'appuyer sur un certain nombre d'hypothèses [30, 31]:

- Parfaite symétrie de construction.
- Assimilation de la cage à un bobinage en court-circuit de même nombre de phases que le bobinage statorique.
- Répartition sinusoïdale, le long de l'entrefer, des champs magnétiques de chaque bobinage.
- Absence de saturation dans le circuit magnétique.
- Le régime homopolaire est nul.

### 1.4.Modélisation de la MAS

La machine asynchrone peut se représenter mathématiquement sous forme de différents modèles d'état selon le choix du repère de travail, du vecteur d'état et des entrées-sorties. Généralement, ces modèles sont définis dans un référentiel diphasé défini à partir du référentiel triphasé naturel de la MAS à l'aide de transformations mathématiques adaptées [32].

#### 1.4.1.Repère triphasé

##### 1.4.1.1.Equations électriques

En appliquant la loi d'Ohm généralisée à chaque enroulement de la machine, nous obtenons :

$$[v_{abc}] = R[i_{abc}] + \frac{d}{dt}[\varphi_{abc}] \quad (1.1)$$

Donc les équations électriques obtenues sont données par (1.5) et (1.6):

$$\begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} = R_s \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \varphi_{sa} \\ \varphi_{sb} \\ \varphi_{sc} \end{bmatrix} \quad (1.2)$$

$$\begin{bmatrix} v_{ra} \\ v_{rb} \\ v_{rc} \end{bmatrix} = R_r \begin{bmatrix} i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \varphi_{ra} \\ \varphi_{rb} \\ \varphi_{rc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (1.3)$$

### 1.4.1.2. Equations magnétiques

Les équations des flux en fonction des courants s'obtiennent à partir des différentes inductances, dont certaines dépendent du temps par l'intermédiaire de l'angle électrique  $\theta_m$  ; position de la phase (a) du rotor par rapport à la phase (a) du stator

$$[\varphi_{abc}] = \begin{bmatrix} L_{ss} & L_{sr} \\ L_{rs} & L_{rr} \end{bmatrix} \begin{bmatrix} i_s \\ i_r \end{bmatrix} \quad (1.4)$$

Les équations statoriques et rotoriques sont données ci-dessous:

$$\begin{bmatrix} \varphi_{sa} \\ \varphi_{sb} \\ \varphi_{sc} \end{bmatrix} = [L_{ss}] \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + [L_{sr}] \begin{bmatrix} i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} \quad [\varphi_{abc}] = \begin{bmatrix} L_{ss} & L_{sr} \\ L_{rs} & L_{rr} \end{bmatrix} \begin{bmatrix} i_s \\ i_r \end{bmatrix} \quad (1.5)$$

$$\begin{bmatrix} \varphi_{ra} \\ \varphi_{rb} \\ \varphi_{rc} \end{bmatrix} = [L_{rs}] \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} + [L_{rr}] \begin{bmatrix} i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} \quad (1.6)$$

Avec :

$$[L_{ss}] = \begin{bmatrix} l_s & l_{ss} & l_{ss} \\ l_{ss} & l_s & l_{ss} \\ l_{ss} & l_{ss} & l_s \end{bmatrix} \quad (1.7)$$

$$[L_{rr}] = \begin{bmatrix} l_r & l_{rr} & l_{rr} \\ l_{rr} & l_r & l_{rr} \\ l_{rr} & l_{rr} & l_r \end{bmatrix} \quad (1.8)$$

$$[L_{sr}] = [L_{rs}]^T = l_{sr} \begin{bmatrix} \cos(\theta_m) & \cos(\theta_m + \frac{2\pi}{3}) & \cos(\theta_m - \frac{2\pi}{3}) \\ \cos(\theta_m - \frac{2\pi}{3}) & \cos(\theta_m) & \cos(\theta_m + \frac{2\pi}{3}) \\ \cos(\theta_m + \frac{2\pi}{3}) & \cos(\theta_m - \frac{2\pi}{3}) & \cos(\theta_m) \end{bmatrix} \quad (1.9)$$

Et :

$l_s$  et  $l_r$  : Inductances propres statorique et rotorique.

$l_{ss}$  et  $l_{rr}$  : Inductances mutuelles entre deux phases statorique ou rotoriques

$l_{sr}$  : Inductance mutuelle maximale entre une phase statorique et rotorique

Afin de rendre toutes les inductances de la machine indépendantes du temps et pour réduire la complexité du modèle nous allons travailler par la suite avec un modèle transformé dans un repère à deux axes orthogonaux.

### 1.4.2. Repère diphasé

Dans cette modélisation la machine asynchrone triphasée réelle est remplacée par une machine diphasée fictive, mais magnétiquement équivalente, grâce à la matrice de transformation de Park

$$T = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\xi) & \cos(\xi - \frac{2\pi}{3}) & \cos(\xi - \frac{4\pi}{3}) \\ -\sin(\xi) & -\sin(\xi - \frac{2\pi}{3}) & -\sin(\xi - \frac{4\pi}{3}) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (1.10)$$

Cette matrice est orthogonale, sa matrice inverse est donc la matrice transposée:

$$[T]^{-1} = [T]^t = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\xi) & -\sin(\xi) & 1/\sqrt{2} \\ \cos(\xi - \frac{2\pi}{3}) & -\sin(\xi - \frac{2\pi}{3}) & 1/\sqrt{2} \\ \cos(\xi - \frac{4\pi}{3}) & -\sin(\xi - \frac{4\pi}{3}) & 1/\sqrt{2} \end{bmatrix} \quad (1.11)$$

L'alimentation triphasée étant équilibrée, la somme des courants  $\sum i_{a,b,c}$  est donc nulle. Le courant et le flux homopolaires s'annulent. Par conséquent, cet axe est rarement pris en compte dans la transformation et la matrice de transformation triphasée-diphasée se résume dans l'équation (1.12) [21]:

$$T_{32} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\xi) & \cos(\xi - \frac{2\pi}{3}) & \cos(\xi - \frac{4\pi}{3}) \\ -\sin(\xi) & -\sin(\xi - \frac{2\pi}{3}) & -\sin(\xi - \frac{4\pi}{3}) \end{bmatrix} \quad (1.12)$$

#### 1.4.2.1. Equations électriques

Les tensions sont représentées dans le repère de Park par l'équation (1.13)

$$[v_{dq}] = [T][v_{abc}] \quad (1.13)$$

$$[v_{dq}] = [T][R][i_{abc}] + [T] \frac{d}{dt} [\varphi_{abc}] \quad (1.14)$$

$$[v_{dq}] = [T][R][T]^{-1}[i_{dq}] + [T] \frac{d}{dt} [[T]^{-1}[\varphi_{dq}]] \quad (1.15)$$

$$[v_{dq}] = [R][T][T]^{-1}[i_{dq}] + [T][T]^{-1} \frac{d}{dt} [\varphi_{dq}] + [T] \left[ \frac{d}{dt} [T]^{-1} \right] [\varphi_{dq}] \quad (1.16)$$

Alors :

$$[v_{dq}] = [R][i_{dq}] + \frac{d}{dt}[\varphi_{dq}] + \frac{d\xi}{dt} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} [\varphi_{dq}] \quad (1.17)$$

Avec :  $\xi = \theta_s$  pour les grandeurs statoriques et  $\xi = \theta_r$  pour les grandeurs rotoriques.

Alors les équations des tensions deviennent:

$$\begin{cases} v_{ds} = R_s i_{ds} + \frac{d}{dt} \varphi_{ds} - \omega_s \varphi_{qs} \\ v_{qs} = R_s i_{qs} + \frac{d}{dt} \varphi_{qs} + \omega_s \varphi_{ds} \end{cases} \quad (1.18)$$

$$\begin{cases} v_{dr} = R_r i_{dr} + \frac{d}{dt} \varphi_{dr} - \omega_r \varphi_{qr} \\ v_{qr} = R_r i_{qr} + \frac{d}{dt} \varphi_{qr} + \omega_r \varphi_{dr} \end{cases} \quad (1.19)$$

Donc :

$$\begin{cases} v_{ds} = R_s i_{ds} + \frac{d}{dt} \varphi_{ds} - \omega_s \varphi_{qs} & (1.20) \\ v_{qs} = R_s i_{qs} + \frac{d}{dt} \varphi_{qs} + \omega_s \varphi_{ds} & (1.21) \\ v_{dr} = R_r i_{dr} + \frac{d}{dt} \varphi_{dr} - \omega_r \varphi_{qr} = 0 & (1.22) \\ v_{qr} = R_r i_{qr} + \frac{d}{dt} \varphi_{qr} + \omega_r \varphi_{dr} = 0 & (1.23) \end{cases}$$

#### 1.4.2.2. Equations magnétiques

Les flux sont représentés dans le repère de Park par l'équation : (1.27)

$$[\varphi_{dq}] = [T][\varphi_{abc}] \quad (1.24)$$

Les équations du stator :

$$[\varphi_{sdq}] = [T][L_{ss}][i_{sabc}] + [T][L_{sr}][i_{rabc}] \quad (1.25)$$

$$[\varphi_{sdq}] = [T][L_{ss}][T]^{-1}[i_{sdq}] + [T][L_{sr}][T]^{-1}[i_{rdq}] \quad (1.26)$$

$$[\varphi_{sdq}] = \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} [i_{sdq}] + \begin{bmatrix} L_m & 0 \\ 0 & L_m \end{bmatrix} [i_{rdq}] \quad (1.27)$$

$$\begin{cases} \varphi_{ds} = L_s i_{ds} + L_m i_{dr} \\ \varphi_{qs} = L_s i_{qs} + L_m i_{qr} \end{cases} \quad (1.28)$$



Les équations du rotor :

$$[\varphi_{rdq}] = [T][L_{rs}][i_{sabc}] + [T][L_{rr}][i_{rabc}] \quad (1.29)$$

$$[\varphi_{rdq}] = [T][L_{rs}][T]^{-1}[i_{sdq}] + [T][L_{rr}][T]^{-1}[i_{rdq}] \quad (1.30)$$

$$[\varphi_{rdq}] = \begin{bmatrix} L_m & 0 \\ 0 & L_m \end{bmatrix} [i_{sdq}] + \begin{bmatrix} L_r & 0 \\ 0 & L_r \end{bmatrix} [i_{rdq}] \quad (1.31)$$

$$\begin{cases} \varphi_{dr} = L_m i_{ds} + L_r i_{dr} \\ \varphi_{qr} = L_m i_{qs} + L_r i_{qr} \end{cases} \quad (1.32)$$

Avec:

$$L_s = l_s - l_{ss} \quad L_r = l_r - l_{rr} \quad L_m = \frac{3}{2} l_{sr} \quad (1.33)$$

Donc:

$$\begin{cases} \varphi_{ds} = L_s i_{ds} + L_m i_{dr} & (1.34) \\ \varphi_{qs} = L_s i_{qs} + L_m i_{qr} & (1.35) \\ \varphi_{dr} = L_m i_{ds} + L_r i_{dr} & (1.36) \\ \varphi_{qr} = L_m i_{qs} + L_r i_{qr} & (1.37) \end{cases}$$

#### 1.4.3.Choix du vecteur d'état

La mise en forme des équations précédentes sous forme d'équations d'état demande dans un premier temps la définition du vecteur d'état  $X$ . On déclare en premier temps un vecteur contient les deux courants statoriques et les deux flux rotoriques et la vitesse mécanique, mais on va considérer par la suite que la vitesse mécanique soit mesurée et on définit le nouveau vecteur d'état contenant seulement les deux courants statoriques et les deux flux rotoriques:

$$X = \begin{bmatrix} I_{ds} \\ I_{qs} \\ F i_{dr} \\ F i_{qr} \end{bmatrix} \quad (1.38)$$

#### 1.4.4.Représentation d'état

Pour établir l'équation d'état, il faut définir aussi le vecteur d'entrée  $U$ . Dans notre cas la machine est alimentée en tensions, donc :

$$U = \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} \quad (1.38)$$

On peut reformuler les équations (1.20) à (1.23) de la manière suivante :

$$\left\{ \begin{array}{l} -i_{ds} - \frac{1}{R_s} \frac{d}{dt} \varphi_{ds} + \frac{1}{R_s} \omega_s \varphi_{qs} + \frac{1}{R_s} v_{ds} = 0 \end{array} \right. \quad (1.39)$$

$$\left\{ \begin{array}{l} -i_{qs} - \frac{1}{R_s} \frac{d}{dt} \varphi_{qs} - \frac{1}{R_s} \omega_s \varphi_{ds} + \frac{1}{R_s} v_{qs} = 0 \end{array} \right. \quad (1.40)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{dr} = -R_r i_{dr} + \omega_r \varphi_{qr} \end{array} \right. \quad (1.41)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{qr} = -R_r i_{qr} - \omega_r \varphi_{dr} \end{array} \right. \quad (1.42)$$

$$\left\{ \begin{array}{l} -i_{ds} - \frac{1}{R_s} \frac{d}{dt} \varphi_{ds} + \frac{1}{R_s} \omega_s \varphi_{qs} + \frac{1}{R_s} v_{ds} = 0 \end{array} \right. \quad (1.43)$$

$$\left\{ \begin{array}{l} -i_{qs} - \frac{1}{R_s} \frac{d}{dt} \varphi_{qs} - \frac{1}{R_s} \omega_s \varphi_{ds} + \frac{1}{R_s} v_{qs} = 0 \end{array} \right. \quad (1.44)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{dr} = -R_r i_{dr} + \omega_r \varphi_{qr} \end{array} \right. \quad (1.45)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{qr} = -R_r i_{qr} - \omega_r \varphi_{dr} \end{array} \right. \quad (1.46)$$

En remplaçant (1.45) et (1.46) dans (1.43) et (1.44) respectivement, on trouve :

$$\varphi_{ds} = L_s i_{ds} - \frac{L_m^2}{L_r} i_{ds} + \frac{L_m^2}{L_r} \varphi_{dr} \quad (1.47)$$

$$\varphi_{qs} = L_s i_{qs} - \frac{L_m^2}{L_r} i_{qs} + \frac{L_m}{L_r} \varphi_{qr} \quad (1.48)$$

En remplaçant ces deux équations dans (1.39) et (1.40) on trouve :

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{ds} = -\frac{R_s L_r}{L_s L_r - L_m^2} i_{ds} + \omega_s i_{qs} - \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} \frac{d}{dt} \varphi_{dr} \\ \quad + \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} \omega_s \varphi_{qr} + \frac{R_s L_r}{L_s L_r - L_m^2} \frac{1}{R_s} v_{ds} \end{array} \right. \quad (1.49)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{qs} = -\omega_s i_{ds} - \frac{R_s L_r}{L_s L_r - L_m^2} i_{qs} - \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} \omega_s \varphi_{dr} \\ \quad - \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} \frac{d}{dt} \varphi_{qr} + \frac{R_s L_r}{L_s L_r - L_m^2} \frac{1}{R_s} v_{qs} \end{array} \right. \quad (1.50)$$

On remplace (1.45) et (1.46) dans (1.41) et (1.42) respectivement. On trouve :

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{dr} = \frac{L_m R_r}{L_r} i_{ds} - R_r \frac{1}{L_r} \varphi_{dr} + \omega_r \varphi_{qr} \end{array} \right. \quad (1.51)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \varphi_{qr} = \frac{L_m R_r}{L_r} i_{qs} - \omega_r \varphi_{dr} - R_r \frac{1}{L_r} \varphi_{qr} \end{array} \right. \quad (1.52)$$

Et finalement si on remplace (1.51) et (1.52) dans (1.49) et (1.50) respectivement on obtient:

$$\left\{ \begin{aligned} \frac{d}{dt} \mathbf{i}_{ds} &= - \left( \frac{R_s L_r}{L_s L_r - L_m^2} + \frac{L_m^2}{L_s L_r - L_m^2} \frac{R_r}{L_r} \right) \mathbf{i}_{ds} + \omega_s \mathbf{i}_{qs} + \frac{L_m}{L_s L_r - L_m^2} \frac{R_r}{L_r} \boldsymbol{\varphi}_{dr} \\ &+ \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} (\omega_s - \omega_r) \boldsymbol{\varphi}_{qr} + \frac{L_r}{L_s L_r - L_m^2} \mathbf{v}_{ds} \end{aligned} \right. \quad (1.53)$$

$$\left\{ \begin{aligned} \frac{d}{dt} \mathbf{i}_{qs} &= -\omega_s \mathbf{i}_{ds} - \left( \frac{R_s L_r}{L_s L_r - L_m^2} + \frac{L_m^2}{L_s L_r - L_m^2} \frac{R_r}{L_r} \right) \mathbf{i}_{qs} \\ &- \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} (\omega_s - \omega_r) \boldsymbol{\varphi}_{dr} + \frac{L_m}{L_s L_r - L_m^2} \frac{R_r}{L_r} \boldsymbol{\varphi}_{qr} + \frac{L_r}{L_s L_r - L_m^2} \mathbf{v}_{qs} \end{aligned} \right. \quad (1.54)$$

Alors (1.51), (1.52), (1.53) et (1.54) donnent :

$$\left\{ \begin{aligned} \frac{d}{dt} \mathbf{i}_{ds} &= - \left( \frac{R_s L_r}{L_s L_r - L_m^2} + \frac{L_m^2}{L_s L_r - L_m^2} \frac{R_r}{L_r} \right) \mathbf{i}_{ds} + \omega_s \mathbf{i}_{qs} + \frac{L_m}{L_s L_r - L_m^2} \frac{R_r}{L_r} \boldsymbol{\varphi}_{dr} \\ &+ \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} (\omega_s - \omega_r) \boldsymbol{\varphi}_{qr} + \frac{L_r}{L_s L_r - L_m^2} \mathbf{v}_{ds} \\ \frac{d}{dt} \mathbf{i}_{qs} &= -\omega_s \mathbf{i}_{ds} - \left( \frac{R_s L_r}{L_s L_r - L_m^2} + \frac{L_m^2}{L_s L_r - L_m^2} \frac{R_r}{L_r} \right) \mathbf{i}_{sq} \\ &- \frac{R_s L_r}{L_s L_r - L_m^2} \frac{L_m}{R_s L_r} (\omega_s - \omega_r) \boldsymbol{\varphi}_{dr} + \frac{L_m}{L_s L_r - L_m^2} \frac{R_r}{L_r} \boldsymbol{\varphi}_{qr} + \frac{L_r}{L_s L_r - L_m^2} \mathbf{v}_{qs} \\ \frac{d}{dt} \boldsymbol{\varphi}_{dr} &= \frac{L_m R_r}{L_r} \mathbf{i}_{ds} - \frac{R_r}{L_r} \boldsymbol{\varphi}_{dr} + \omega_r \boldsymbol{\varphi}_{qr} \\ \frac{d}{dt} \boldsymbol{\varphi}_{qr} &= \frac{L_m R_r}{L_r} \mathbf{i}_{qs} - \omega_r \boldsymbol{\varphi}_{dr} - \frac{R_r}{L_r} \boldsymbol{\varphi}_{qr} \end{aligned} \right. \quad (1.55)$$

#### 1.4.5. Equation mécanique

Comme la somme des couples à l'arbre est équivalente au couple inertiel, il s'ensuit :

$$C_{em} - C_r = J \frac{d\Omega_m}{dt} + f \Omega_m \quad (1.56)$$

Avec :  $C_{em}$  est le Couple électromagnétique.

$C_r$  est le couple de charge.

$\Omega_m$  est la vitesse mécaniques.

Alors :

$$\frac{d\Omega_m}{dt} = \frac{1}{J} C_{em} - \frac{1}{J} C_r - \frac{f}{J} \Omega_m \quad (1.57)$$

Le couple électromagnétique est donné par l'équation (1.58) :

$$C_{em} = \frac{1}{2} [i_s^T \quad i_r^T] \frac{d}{d\theta_m} \left( \begin{bmatrix} L_{ss} & L_{sr} \\ L_{rs} & L_{rr} \end{bmatrix} \begin{bmatrix} i_s \\ i_r \end{bmatrix} \right) = [i_{sabc}]^T \frac{d}{d\theta_m} ([L_{sr}] [i_{rabc}]) \quad (1.58)$$

Son expression dans le repère de Park devient:

$$C_{em} = \left[ [T]^{-1} [i_{sdq}] \right]^T \frac{d}{d\theta_m} \left( [L_{sr}] [T]^{-1} [i_{rdq}] \right) \quad (1.59)$$

La simplification de la relation précédente donne :

$$C_{em} = p \frac{L_m}{L_r} (\varphi_{dr} i_{qs} - \varphi_{qr} i_{ds}) \quad (1.60)$$

Avec :

$$\frac{d\Omega_m}{dt} = \frac{1}{J} C_{em} - \frac{1}{J} C_r - \frac{f}{J} \Omega_m \quad (1.61)$$

Donc :

$$\frac{d\Omega_m}{dt} = \frac{pL_m}{JL_r} (i_{qs}\varphi_{dr} - i_{ds}\varphi_{qr}) - \frac{1}{J} C_r - \frac{f}{J} \Omega_m \quad (1.62)$$

En ajoutant l'équation mécanique représentée par (1.62) au système (1.55) et en introduisant les constantes représentées par (1.63), le modèle de la MAS prend la représentation (1.64) :

$$\sigma = \frac{L_r L_s - L_m^2}{L_r L_s} \quad T_r = \frac{L_r}{R_r} \quad T_s = \frac{L_s}{R_s} \quad k_m = \frac{L_m}{JL_r} \quad m_1 = \frac{f}{J} \quad m_2 = \frac{1}{J} \quad (1.63)$$

$$\begin{cases} \frac{d}{dt} i_{ds} = -\left( \frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) i_{ds} + \omega_s i_{qs} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{dr} + \frac{1-\sigma}{\sigma L_m} (\omega_s - \omega_r) \varphi_{qr} + \frac{1}{\sigma L_s} v_{ds} \\ \frac{d}{dt} i_{qs} = -\omega_s i_{ds} - \left( \frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) i_{qs} - \frac{1-\sigma}{\sigma L_m} (\omega_s - \omega_r) \varphi_{dr} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{qr} + \frac{1}{\sigma L_s} v_{qs} \\ \frac{d}{dt} \varphi_{dr} = \frac{L_m}{T_r} i_{ds} - \frac{1}{T_r} \varphi_{dr} + \omega_r \varphi_{qr} \\ \frac{d}{dt} \varphi_{qr} = \frac{L_m}{T_r} i_{qs} - \omega_r \varphi_{dr} - \frac{1}{T_r} \varphi_{qr} \\ \frac{d}{dt} \Omega_m = p k_m (i_{qs} \varphi_{qr} - i_{ds} \varphi_{dr}) - m_1 \Omega_m - m_2 C_r \end{cases} \quad (1.64)$$

#### 1.4.6.Choix du référentiel

On a exprimé les équations et les grandeurs de la machine dans un repère  $dq$  faisant un angle  $\theta_s$  avec le stator et un angle  $\theta_r$  avec le rotor

Il existe trois choix possibles pour le référentiel ( $dq$ ) [21, 33]:

- Repère fixe ou lié au stator où :  $\omega_s = 0, \omega_r = \omega_m$ .
- Repère lié au rotor ou :  $\omega_s = \omega_m, \omega_r = 0$ .
- Repère lié au champ tournant qui dépend des deux vitesses :  $\omega_s, \omega_r$ .

Si on choisit le repère lié au stator alors l'équation (1.64) devient :

$$\begin{cases} \frac{d}{dt} i_{ds} = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) i_{ds} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{dr} + \frac{1-\sigma}{\sigma L_m} \omega_m \varphi_{qr} + \frac{1}{\sigma L_s} v_{ds} \\ \frac{d}{dt} i_{qs} = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) i_{qs} - \frac{1-\sigma}{\sigma L_m} \omega_m \varphi_{dr} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{qr} + \frac{1}{\sigma L_s} v_{qs} \\ \frac{d}{dt} \varphi_{dr} = \frac{L_m}{T_r} i_{ds} - \frac{1}{T_r} \varphi_{dr} + \omega_m \varphi_{qr} \\ \frac{d}{dt} \varphi_{qr} = \frac{L_m}{T_r} i_{qs} - \omega_m \varphi_{dr} - \frac{1}{T_r} \varphi_{qr} \\ \frac{d}{dt} \Omega_m = p k_m (i_{qs} \varphi_{qr} - i_{ds} \varphi_{dr}) - m_1 \Omega_m - m_2 C_r \end{cases} \quad (1.65)$$

On le représente sous forme d'équation d'état :

$$\begin{bmatrix} \frac{d}{dt} i_{ds} \\ \frac{d}{dt} i_{qs} \\ \frac{d}{dt} \varphi_{dr} \\ \frac{d}{dt} \varphi_{qr} \\ \frac{d}{dt} \Omega_m \end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) i_{ds} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{dr} + \frac{1-\sigma}{\sigma L_m} p \Omega_m \varphi_{qr} \\ -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) i_{qs} - \frac{1-\sigma}{\sigma L_m} p \Omega_m \varphi_{dr} + \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} \varphi_{qr} \\ \frac{L_m}{T_r} i_{ds} - \frac{1}{T_r} \varphi_{dr} - p \Omega_m \varphi_{qr} \\ \frac{L_m}{T_r} i_{qs} + p \Omega_m \varphi_{dr} - \frac{1}{T_r} \varphi_{qr} \\ p k_m (i_{qs} \varphi_{qr} - i_{ds} \varphi_{dr}) - m_1 \Omega_m - m_2 C_r \end{bmatrix} + \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} \quad (1.66)$$

Si on utilise les constantes simplificatrices (1.67) :

$$\begin{cases} a_1 = \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & a_3 = \frac{1-\sigma}{\sigma} \frac{1}{L_m T_r} & a_4 = p \frac{1-\sigma}{\sigma L_m} & a_5 = \frac{L_m}{T_r} & a_7 = \frac{1}{T_r} & a_8 = p \\ a_9 = p k_m & a_{10} = m_1 & a_{11} = m_2 C_r & b_1 = \frac{1}{\sigma L_s} \end{cases} \quad (1.67)$$

L'équation (1.66) devient :

$$\begin{bmatrix} \frac{d}{dt} i_{ds} \\ \frac{d}{dt} i_{qs} \\ \frac{d}{dt} \varphi_{dr} \\ \frac{d}{dt} \varphi_{qr} \\ \frac{d}{dt} \Omega_m \end{bmatrix} = \begin{bmatrix} -a_1 i_{ds} + a_3 \varphi_{dr} + a_4 \Omega_m \varphi_{qr} \\ -a_1 i_{qs} - a_4 \Omega_m \varphi_{dr} + a_3 \varphi_{qr} \\ a_5 i_{ds} - a_7 \varphi_{dr} - a_8 \Omega_m \varphi_{qr} \\ a_5 i_{qs} + a_8 \Omega_m \varphi_{dr} - a_7 \varphi_{qr} \\ a_9 (i_{qs} \varphi_{dr} - i_{ds} \varphi_{qr}) - a_{10} \Omega_m - a_{11} C_r \end{bmatrix} + \begin{bmatrix} b_1 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} \quad (1.68)$$

### 1.5. Identification des paramètres de la MAS

L'identification consiste à rechercher des modèles mathématiques de systèmes à partir de données expérimentales et de connaissances disponibles à priori. Ces modèles doivent fournir une approximation fidèle du comportement du système physique [26].

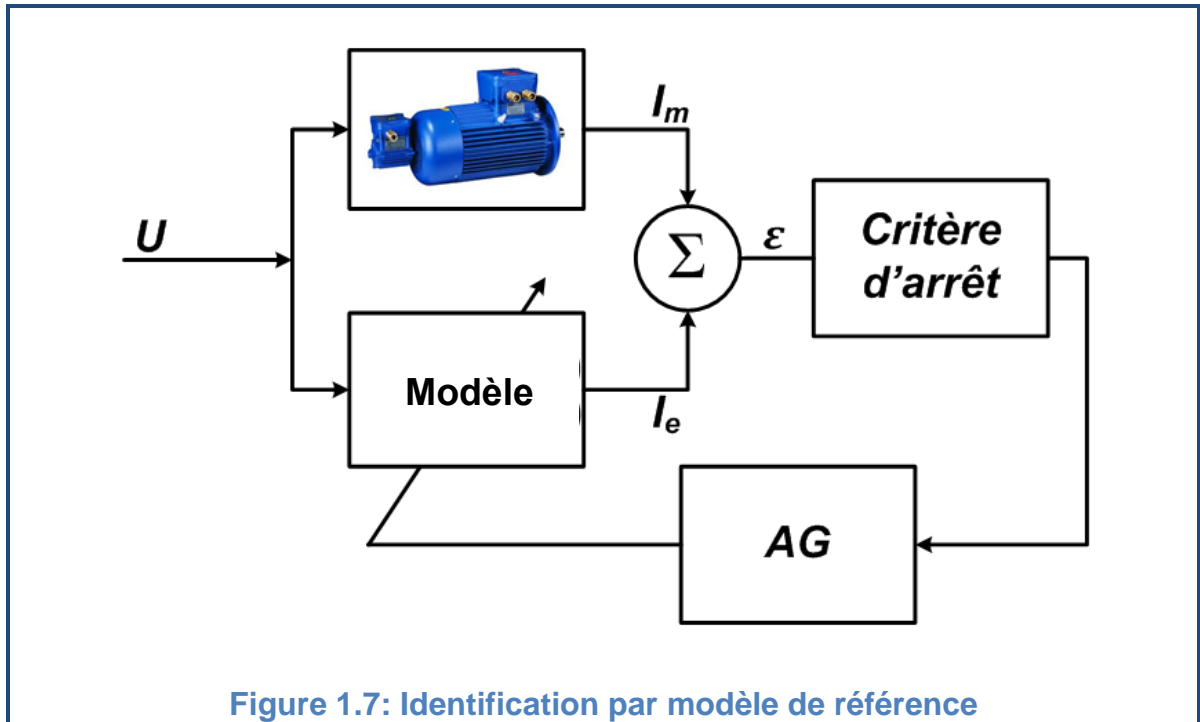
Il existe de nombreuses méthodes d'identification des paramètres de la machine asynchrone, différentes de par leur principe et leur complexité. Les méthodes classiques ont été utilisées depuis très longtemps. Actuellement les méthodes numériques, et spécialement celles qui se basent sur les méta-heuristiques, ont montrés leurs performances [26, 27, 28].

La méthode classique qui est largement employé en électrotechnique est principalement basée sur une série de mesures. Cette méthode est présentée à l'appendice C mais elle n'a pas été utilisée dans ce travail. Une deuxième méthode numérique basée sur les Algorithmes Génétiques a été utilisée et est présentée dans cette partie.

L'identification à l'aide de méthodes numériques repose généralement sur le régime transitoire des variables d'état mesurables (courants, tensions et vitesse) [26].

Les paramètres à déterminer sont des paramètres électriques tels que  $R_s, R_r, L_s, L_r$  et  $L_m$  et d'autres mécaniques tels que  $J$  et  $f$ .

L'approche utilisée permet de déterminer simultanément les paramètres électriques et mécaniques de la machine à l'aide de la méthode du modèle de référence, en utilisant le courant de démarrage ainsi que la tension simple correspondante comme signaux d'entrée-sortie. La figure 1.7 représente le schéma de principe de l'identification par modèle de référence.



### 1.5.1. Système à identifier

Le système à identifier est une MAS alimentée par la source de tension triphasée. Les valeurs mesurées sont le courant d'une phase statorique et la tension correspondante obtenues à l'aide d'un oscilloscope numérique. L'excitation du système est réalisée à l'aide de la tension.

### 1.5.2. Modèle de référence

Le modèle mathématique utilisé pour l'identification est basé sur les courants statoriques, les courants rotoriques et la vitesse mécanique. Il est représenté par l'équation (1.69) [26].

$$\begin{cases}
 \frac{d}{dt} I_{ds} = \frac{-1}{\sigma T_s} I_{ds} + \frac{1-\sigma}{\sigma} \omega I_{qs} + \frac{1-\sigma}{\sigma T_r} I_{qr} \frac{1-\sigma}{\sigma} \omega I_{qr} + \frac{V_{ds}}{\sigma L_s} \\
 \frac{d}{dt} I_{qs} = -\frac{1-\sigma}{\sigma} \omega I_{ds} - \frac{1}{\sigma T_s} I_{qs} - \frac{1-\sigma}{\sigma} \omega I_{dr} + \frac{1-\sigma}{\sigma T_r} I_{qr} - \frac{V_{qs}}{\sigma L_s} \\
 \frac{d}{dt} I_{dr} = \frac{1}{\sigma T_s} I_{ds} - \frac{1}{\sigma} \omega I_{qs} - \frac{1}{\sigma T_r} I_{dr} - \frac{1}{\sigma} I_{qr} - \frac{V_{ds}}{\sigma L_s} \\
 \frac{d}{dt} I_{qr} = \frac{1}{\sigma} \omega I_{ds} + \frac{1}{\sigma T_s} I_{qs} + \frac{1}{\sigma} \omega I_{dr} - \frac{1}{\sigma T_r} I_{qr} - \frac{V_{qs}}{\sigma L_s} \\
 \frac{d}{dt} \omega = \frac{P}{J} L_s (1-\sigma) (I_{qs} I'_{dr} - I_{ds} I'_{qr}) - \frac{f\omega}{J}
 \end{cases} \quad (1.69)$$

Avec :  $I'_{dr} = I_{dr} \frac{L_r}{L_m}$  et  $I'_{qr} = I_{qr} \frac{L_r}{L_m}$

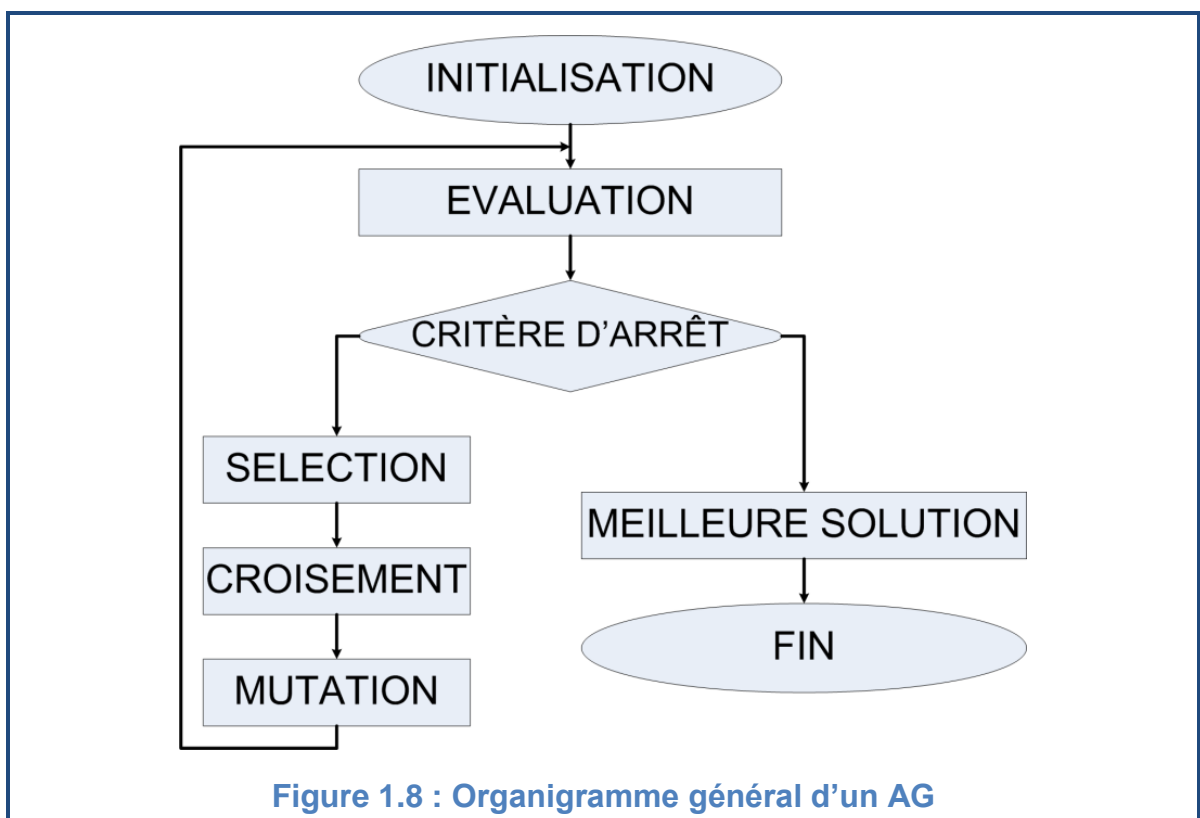
### 1.5.3. Algorithmes génétiques

Les algorithmes génétiques [27] sont des méthodes basées sur les mécanismes biologiques et sur les principes fondamentaux de sélection des individus.

Les principes fondamentaux des algorithmes génétiques sont:

- "Dans chaque environnement, seuls les espèces les mieux adaptées perdurent au cours du temps, les autres étant condamnées à disparaître".
- "Au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce".

Les algorithmes génétiques simulent le processus d'évolution d'une population de N solutions du problème représentant des individus, ils utilisent des opérateurs tels que la sélection, le croisement et la mutation pour reproduire une nouvelle population de solutions de mieux en mieux adaptée au problème. Cette adaptation est évaluée grâce à une fonction coût. Le déroulement général des AG peut être représenté par l'organigramme de la figure 1.8.





### 1.5.3.1. Le codage des chromosomes

Une solution (chromosome) est la suite des paramètres recherchés organisés dans un seul vecteur et chaque paramètre est repérable par sa position.

Dans le cas de la MAS la solution contient les paramètres suivants :

$\sigma$  : Coefficient de fuite.

$T_s, T_r$  : Constantes de temps statorique et rotorique.

$L_s$  : Inductance cyclique statorique.

$J$  : Moment d'inertie de rotor.

$f$  : Coefficient de frottement.

Ces paramètres sont codés en réel en considérant un seul chiffre dans la partie entière et six chiffres pour la partie fractionnaire.

### 1.5.3.2. Initialisation de la population

Une première population a été générée dont la taille et le contenu sont aléatoires. La réservation du dernier chromosome de la population aux différents paramètres de l'AG telle que la taille de la prochaine population et les différentes probabilités (de croisement et de mutation) permet d'optimiser ces paramètres et donc diminuer le temps de calcul.

La pré-connaissance des intervalles de variation des paramètres de la MAS permet d'initialiser le meilleur chromosome afin de garantir la convergence de l'algorithme et de minimiser aussi le temps de calcul. Ce chromosome a été initialisé par les valeurs suivantes :

$[\sigma, T_r, L_s, T_s, J, f] = [0.200000 \quad 0.030000 \quad 0.300000 \quad 0.030000 \quad 0.002000$   
 $0.000200]$

### 1.5.3.3. Evaluation de la population

L'évaluation de chaque individu a été effectuée en calculant l'erreur quadratique entre deux signaux :

- Le premier ( $I_m$ ) est donné par les mesures.
- Et le deuxième ( $I_e$ ) est obtenu à partir du modèle mathématique de la MAS en utilisant les paramètres donnés par cet individu.

$$J_c = \sum(I_m - I_e)^2 \quad (1.70)$$

Cette erreur a été inversée afin de calculer la fonction d'adaptation, ou fitness, associé à chaque individu. Cette valeur a pour but d'évaluer si un individu est mieux adapté qu'un autre à son environnement.

#### 1.5.3.4.Reproduction

La reproduction d'une nouvelle population se fait par les trois étapes suivantes :

**La sélection** sert à choisir dans l'ensemble de la population les individus qui participeront au croisement et à la mutation. La méthode utilisée est celle de la roulette.

Avec cette méthode chaque individu a une chance d'être sélectionné proportionnelle à sa performance, donc plus les individus sont adaptés au problème, plus ils ont de chances d'être sélectionnés.

La probabilité de sélection de chaque individu(i) est :

$$p_s(i) = \frac{f_e(i)}{\sum_{i=1}^N f_e(i)} \quad (1.71)$$

Avec :  $p_s(i)$ : La probabilité de sélection de l'individu (i).

$f_e(i)$  : Fonction d'évaluation (fitness) de l'individu (i).

$N$  : La taille de la population.

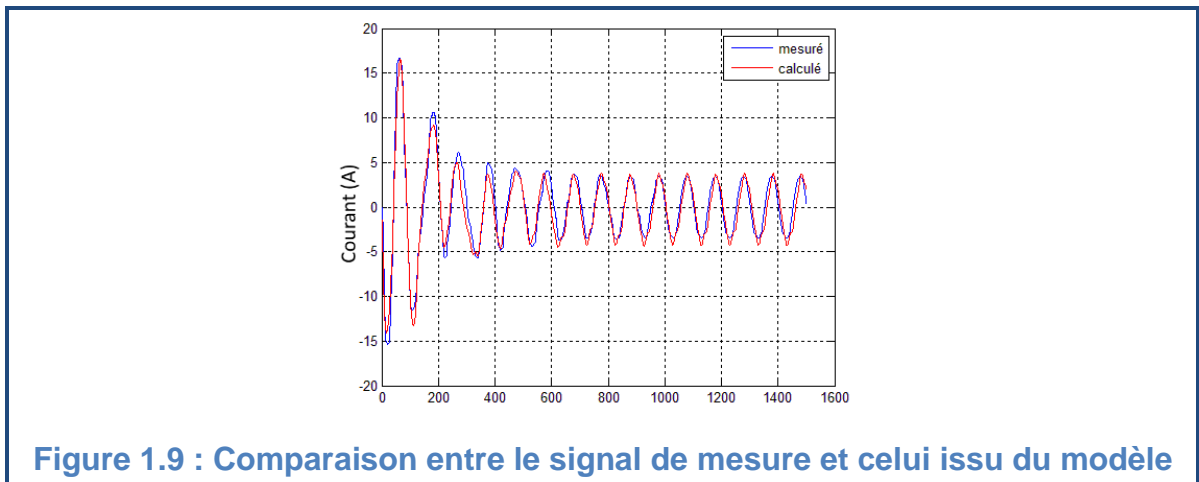
**Le croisement** réalise la reproduction entre les individus de la population avec une probabilité de croisement  $P_c$ , il représente l'étape clef de l'AG.

Si la probabilité de croisement est de 100%, alors toute la population participe au croisement. Par contre, si elle est de 0%, la nouvelle génération au complet est la copie exacte des individus de l'ancienne population. Ceci ne signifie pas que la nouvelle génération est forcément identique. En effet, la nouvelle population n'a pas seulement été modifiée par l'opérateur de croisement, mais également par l'opérateur de mutation et de sélection, ce qui occasionne des changements dans la nouvelle population.

**La mutation** a comme objectif de modifier de façon aléatoire certains individus de la population pour prévenir la convergence prématurée. Elle consiste tout simplement à changer la valeur aléatoire d'un gène avec une probabilité  $P_m$  (ou de plusieurs gènes, vu la probabilité de mutation) se trouvant en un locus bien particulier et qui lui est déterminé de manière aléatoire.

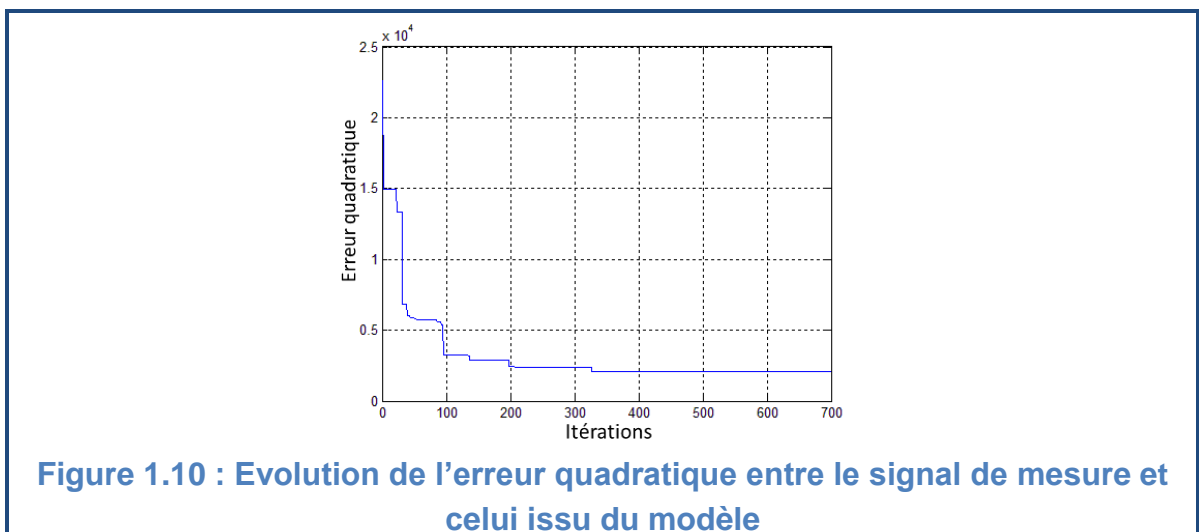
#### 1.5.4. Résultats de l'identification

Les paramètres obtenus sont utilisés pour simuler le modèle de la MAS et le signal du courant résultant a été comparé avec le celui du courant mesuré. Les deux signaux sont représentés dans la figure 1.9.



**Figure 1.9 : Comparaison entre le signal de mesure et celui issu du modèle**

Et l'évolution de l'erreur quadratique est représentée dans la figure 1.10.



**Figure 1.10 : Evolution de l'erreur quadratique entre le signal de mesure et celui issu du modèle**

### 1.6.Observabilité de la MAS

La commandabilité et l'observabilité sont deux principales caractéristiques d'un système dynamique. La condition d'observabilité d'un système assure que les variables d'états sont observables à partir de ses entrées et sorties. La vérification du critère d'observabilité permet d'affirmer qu'on peut calculer l'état initial du système  $X(t_0)$  en fonction des valeurs mesurées de la sortie  $Y(t_0)$  et de l'entrée  $U(t_0)$ . En connaissant l'équation d'état du système et l'état initial  $X(t_0)$ , on définit par la suite le vecteur d'état  $X(t)$ .

Contrairement au cas linéaire, Le problème de l'observabilité des systèmes non linéaires se complique, dans la mesure où l'observabilité dans ce cas peut dépendre de l'entrée appliquée. L'observabilité des systèmes non linéaires est définie à partir de la notion d'indiscernabilité [8, 30]. Nous présentons, ci-après, quelques définitions de base relatives à l'observabilité des systèmes non linéaires

Soit le système non linéaire suivant :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \quad (1.72)$$

**Entrée admissible** : L'ensemble des entrées admissibles  $u(t)$  est celui des fonctions telles que pour toute condition initiale  $x(t_0)$ , il existe une solution unique  $x_u(t)$  définie sur un intervalle maximal  $[0, T_{(u,x)}]$ .

**Entrée universelle** : Une entrée  $u(t)$  est dite universelle sur  $[0, T]$  si elle distingue (discerne) tout couple de points distincts sur  $[0, T]$ . L'entrée  $u(t)$  rend le système observable.

**Indiscernabilité** : Deux états initiaux  $x_1(t_0) = x_{01}$  et  $x_2(t_0) = x_{02}$  sont dits indiscernables pour le système (1.72) si  $\forall t \in [t_0, t_1]$ , les sorties correspondantes  $y_1(t)$  et  $y_2(t)$  sont identiques quelle que soit l'entrée admissible  $u(t)$  du système.

**Observabilité :** Le système non linéaire (1.72) est dit observable s'il n'admet pas de pair de points indiscernables. En d'autres termes, un système non linéaire est observable si toutes les conditions initiales sont départagées par examen de sa sortie.

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

$$L_{f_u} = \sum_{i=1}^n \frac{\partial \varphi}{\partial x_i} f_{u_i}$$

**Espace d'observabilité :** Soit le système (1.72). L'espace d'observabilité, noté  $O$ , est le plus petit sous-espace vectoriel de fonctions de  $R^n$  à valeur dans l'espace de sortie, contenant les sorties  $h_1, h_2, \dots, h_p$  et qui soit fermé sous l'opération de la dérivation de Lie par rapport au champ de vecteur  $f(x, u)$ ,  $u$  étant fixé. On note  $dO$  l'espace des différentielles des éléments de  $O$ .

**Rang d'observabilité :** On dit que le système (1.72) satisfait la condition de rang d'observabilité si  $\forall x \in V, \dim(dO) = n \Leftrightarrow \text{rang}(dO) = n$

Lorsqu'un système satisfait la condition de rang, alors ce système sera localement observable.

Considérons une MAS alimentée en tension par un onduleur de tension. Pour un référentiel lié au stator, les variables d'entrées sont les composantes de la tension statorique  $v_{ds}, v_{qs}$ , les variables de sorties sont les composantes du courant statorique  $i_{ds}, i_{qs}$ . Et le vecteur d'état choisi comprend les deux composantes du courant statorique et celles du flux rotorique.

Dans le cas où la vitesse mécanique soit mesurée, la machine asynchrone peut être considérée comme un système linéaire entre deux instants d'échantillonnage et le système décrit par (1.68) devient :



### 1.7.Conclusion

Dans ce chapitre nous avons d'abord modélisé la machine asynchrone dans le repère triphasé, puis la transformation de PARK a été utilisée pour passer à un repère diphasé équivalent. Cette représentation nous a permis de réduire l'ordre du système en isolant la composante homopolaire.

Ensuite nous nous sommes intéressés au problème d'identification des paramètres de machine et on a présenté deux méthodes différentes, la première est basé sur les essais expérimentaux classiques. La deuxième est une méthode numérique basée sur un algorithme génétique. Ce dernier est utilisé pour minimiser l'erreur quadratique entre le courant mesuré et celui fourni par un modèle mathématique. La mesure du courant statorique et de la tension correspondante a été effectuée à l'aide d'un oscilloscope numérique. Les résultats obtenus par cette méthode sont acceptables.

Finalement nous avons présenté quelques définitions de l'observabilité des systèmes non linéaires ainsi que l'étude de l'observabilité de la Machine asynchrone dans le cas de la vitesse rotorique constante. Ceci va nous servir dans le chapitre suivant pour aborder l'analyse et la synthèse du FK.

## **CHAPITRE 2**

### **CONCEPTION ET REGLAGE**

#### **DU FILTRE DE KALMAN**

##### 2.1.Introduction

La commande et le diagnostic des systèmes dynamiques reposent sur l'hypothèse que tout l'état est connu à chaque instant. Pour des raisons technologiques (de matériel, de réalisabilité, etc.), de fiabilité (panne des éléments de mesure) ou encore économiques (coût des capteurs), dans de nombreuses applications, la mesure de tout l'état n'est pas possible. Il est alors nécessaire, à l'aide des mesures disponibles, de reconstruire les variables d'état non mesurées. C'est le problème bien connu de l'observation [5, 21, 30, 35].

Sous l'hypothèse que la vitesse mécanique est mesurée et qu'elle est constante entre deux instants d'échantillonnage, le filtre de Kalman a été utilisé depuis plusieurs années pour la reconstruction des flux rotoriques de la machine asynchrone [4]. Lorsqu'on veut s'affranchir du capteur de vitesse, la version étendue du filtre de Kalman (FKE) a été utilisée pour l'estimation simultanée des flux rotorique ainsi que de la vitesse mécanique [13, 14]. Une alternative au filtre de Kalman étendu est le filtre de Kalman adaptatif (FKA) [20, 22]. Une loi d'adaptation basée sur la stabilité au sens de Lyapunov est utilisée. Ceci permet d'éviter le calcul du jacobien nécessité par le FKE. Dans des applications de commande, ces derniers (FKE, FKA) ont été utilisé pour suivre les évolutions des résistances statorique ou rotorique [36].

Le filtre de Kalman avec ses différentes versions a aussi été utilisé dans de nombreux domaines (robotique [37] à l'hydrologie [10] à l'économétrie [11]...).



## 2.2.Hypothèses simplificatrices

Le principe du filtrage de Kalman consiste à se placer dans une modélisation probabiliste des incertitudes de modélisation et des bruits de mesure pour calculer le gain d'innovation [4].

Le model d'état proposé par Kalman est le suivant :

$$\dot{x}(t) = Ax(t) + Bu(t) + Mw(t) \quad (2.1)$$

$$y(t) = Cx(t) + Du(t) + v(t) \quad (2.2)$$

Les matrices  $W$  et  $V$  représentent le bruit de d'état et le bruit de mesure de matrice de covariance respective  $Q$  et  $R$ .

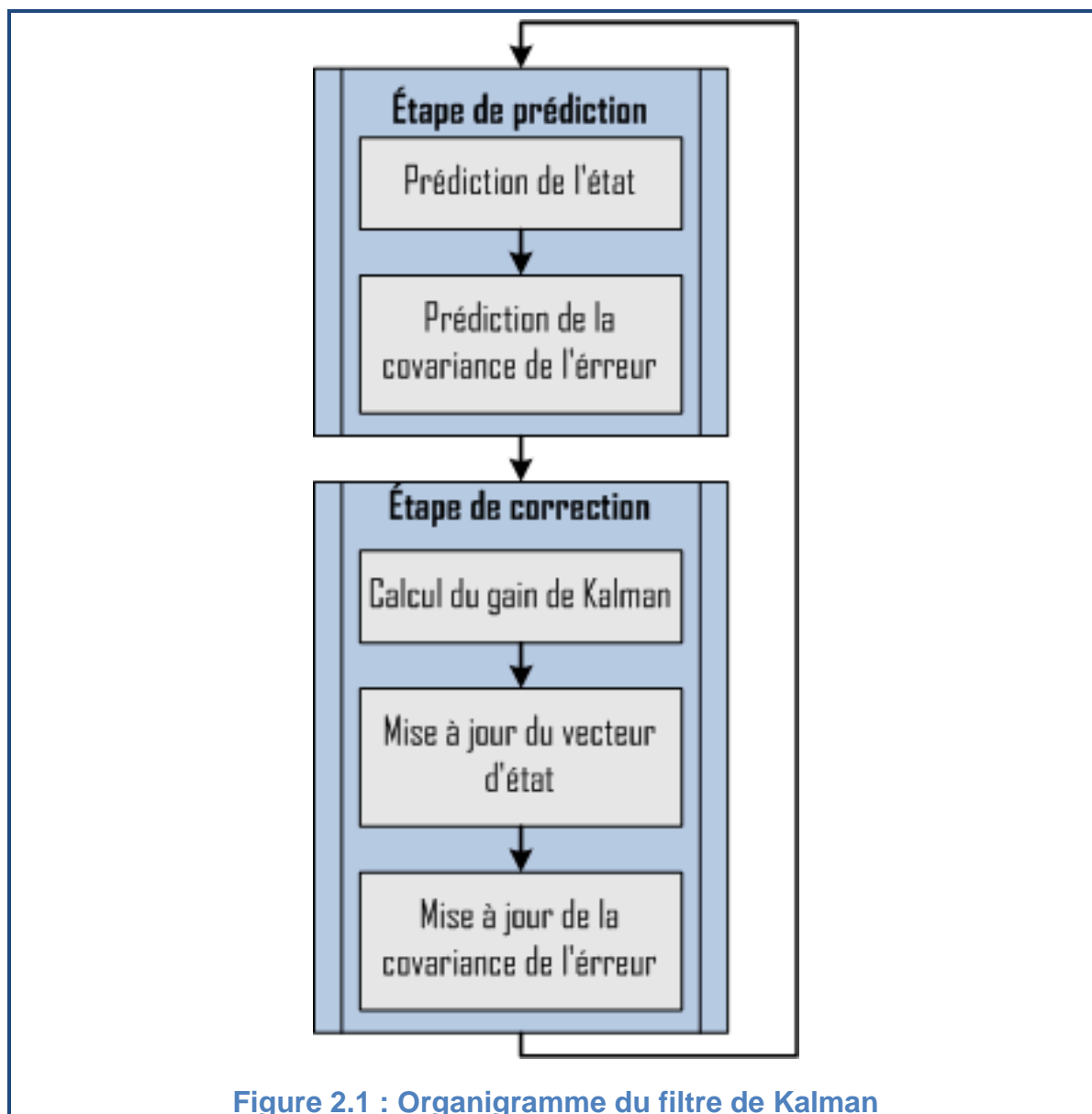
Les hypothèses mathématiques suivantes sont nécessaires pour construire ce filtre [49], [50]:

- La paire (A, C) est détectable, c'est-à-dire qu'il n'y a pas de mode instable et inobservable dans le système.
- Les signaux  $w(t)$  et  $v(t)$  sont des bruits blancs gaussiens centrés de Densité Spectrale de Puissance (DSP)  $Q$  et  $R$  respectivement, c'est-à-dire:
  - $E[w(t)w(t + \tau)^T] = Q \delta(\tau)$
  - $E[v(t)v(t + \tau)^T] = R \delta(\tau)$
  - $E[w(t)v(t + \tau)^T] = 0$
- La dernière relation traduit l'indépendance stochastique des bruits  $w(t)$  et  $v(t)$  : cette hypothèse est introduite pour alléger les calculs qui vont suivre mais n'est pas nécessaire.
- La matrice R est inversible (il y a autant de sources de bruits blancs indépendantes que de mesures dans l'équation de mesure).

## 2.3.Présentation du filtre de Kalman

Le filtre de Kalman se forme de deux systèmes d'équations linéaires. Le premier projette temporellement l'état actuel ce qui permet d'estimer l'état a priori de la prochaine itération. Cela correspond à la phase de **prédiction**. Le second système permet de prendre en compte une nouvelle mesure de l'état estimé a

priori pour obtenir finalement une version ajustée de l'état a posteriori. Cette phase est qualifiée de **correction** [4]. Ce type d'algorithme utilisant une phase de prédiction nous renseignant sur l'état a priori puis de correction améliorant la description de l'état a posteriori peut être caractérisée par la figure 2.1.



### 2.3.1. Equation générale du filtre de Kalman

Le filtre de Kalman est fondé sur un schéma du type prédiction-corrrection. Il utilise l'information d'innovation pour corriger la prédiction de l'état en prenant en compte l'estimation antérieure [4].

L'équation générale du filtre est représentée par (2.3):

$$\dot{\hat{x}}(t) = A_f \hat{x}(t) + B_f u(t) + Ky(t) \quad (2.3)$$

### 2.3.2. Calcul de l'erreur

Soit :  $\varepsilon(t) = x(t) - \hat{x}(t)$  l'erreur d'estimation de l'état du système

Et :  $\varepsilon(t_0) = x(t_0) - \hat{x}(t_0)$  l'erreur d'initialisation.

En retranchant l'équation (2.3) de l'équation (2.1) et en remplaçant l'équation de mesure (2.2), nous pouvons écrire :

$$\begin{aligned} \dot{\varepsilon}(t) &= Ax(t) + Bu(t) + Mw(t) - A_f \hat{x}(t) - B_f u(t) - K(Cx(t) + Du(t) + v(t)) \\ &= (A - KC)x(t) - A_f \hat{x}(t) + (B - B_f - KD)u(t) + Mw(t) - Kv(t) \\ &= (A - KC)\varepsilon(t) + (A - KC - A_f)\hat{x}(t) + (B - KD - B_f)u(t) + Mw(t) - Kv(t) \end{aligned} \quad (2.4)$$

Etant donné que les bruits  $w$  et  $v$  sont gaussiens et le système est linéaire, on peut affirmer que  $\varepsilon(t)$  est une variable aléatoire gaussienne. Nous allons maintenant nous intéresser à l'espérance mathématique (moyenne) de  $\varepsilon(t)$ .

### 2.3.3. Estimateur non biaisé

Avant tout, on souhaite que l'estimateur soit non biaisé, c'est-à-dire que :

- quel que soit le profil de commande  $u(\tau)$  appliqué sur l'horizon  $\tau \in [t_0, t]$
- quel que soit l'initialisation  $\hat{x}(t_0)$

On souhaite que la moyenne de l'erreur d'estimation tende vers zéro lorsque le temps tend vers l'infini [51].

Les bruits  $w$  et  $v$  étant centrés, nous pouvons écrire :

$$\begin{aligned} E[\dot{\varepsilon}(t)] &= E[\dot{\varepsilon}(t)] \\ &= (A - KC)E[\varepsilon(t)] + (A - KC - A_f)E[\hat{x}(t)] + (B - KD - B_f)u(t) = 0 \end{aligned} \quad (2.5)$$

Il faut donc :

$$\begin{aligned}
 (B - KD - B_f)u(t) = 0 & \Rightarrow B_f = (B - KD) \\
 (A - KC - A_f)E[\hat{x}(t)] = 0 & \Rightarrow A_f = (A - KC) \\
 (A - KC)E[\varepsilon(t)] = 0 & \Rightarrow e^{(A-KC)t}\varepsilon(t) \text{ Tend vers zéro,} \\
 & \text{ donc } (A - KC) \text{ doit être stable.}
 \end{aligned}$$

Donc :

$$A_f = (A - KC), \quad B_f = (B - KD) \quad \text{et} \quad (A - KC) \text{ stable} \quad (2.6)$$

Alors l'équation (2.3) devient :

$$\begin{aligned}
 \hat{x}(t) &= (A - KC)\hat{x}(t) + (B - KD)u(t) + Ky(t) \\
 &= A\hat{x}(t) - KC\hat{x}(t) + Bu(t) - KDu(t) + Ky(t) \\
 &= \underbrace{A\hat{x}(t) + Bu(t)}_{\text{Copie du système}} + \underbrace{K(y(t) - C\hat{x}(t) - Du(t))}_{\text{Terme de correction}} \quad (2.7)
 \end{aligned}$$

On voit finalement que le filtre est constitué d'une copie du système qu'on lui ajoute un terme de correction qui correspond au gain de Kalman (K) multiplié par l'erreur entre l'état mesurer et l'état prédit.

#### 2.3.4. Calcul du gain de Kalman

Pour calculer le gain  $K$ , on doit minimiser la variance de l'erreur d'estimation de l'état du système.

On a :  $\varepsilon(t) = x(t) - \hat{x}(t)$  une variable aléatoire vectorielle gaussienne et centrée.

Le caractère gaussien de cette variable permet d'affirmer que si la variance de l'erreur d'estimation est effectivement minimisée, alors  $\hat{x}(t)$  est vraiment le meilleur estimé de  $x(t)$ .

$$\begin{aligned}
 J(t) &= \sum_{i=1}^n E[\varepsilon_i(t)^2] \\
 &= E[\varepsilon^T(t)\varepsilon(t)] \\
 &= \text{Trace } E[\varepsilon(t)\varepsilon^T(t)] \\
 &= \text{Trace } P(t) \quad (2.8)
 \end{aligned}$$

Avec  $P(t)$  est la matrice de covariance de l'erreur d'estimation et:

$$P(t) = E[\varepsilon(t)\varepsilon^T(t)] = E[(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T] \quad (2.9)$$

D'après (2.4) (2.5) :

$$\begin{aligned} \dot{\varepsilon}(t) &= (A - KC)\varepsilon(t) + (A - KC - A_f)\hat{x}(t) + (B - KD - B_f)u(t) + Mw(t) - Kv(t) \\ &= (A - KC)\varepsilon(t) + Mw(t) - Kv(t) \\ &= (A - KC)\varepsilon(t) + [M \quad -K] \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \end{aligned} \quad (2.10)$$

Alors :

$$\begin{aligned} \dot{P}(t) &= E[\dot{\varepsilon}(t)\dot{\varepsilon}^T(t)] \\ &= (A - KC)P(t) + P(t)(A - KC)^T + [M \quad -K]E \left[ \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \begin{bmatrix} w(t) & v(t) \end{bmatrix} \right] \begin{bmatrix} M^T \\ -K^T \end{bmatrix} \\ &= (A - KC)P(t) + P(t)(A - KC)^T + [M \quad -K] \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} M^T \\ -K^T \end{bmatrix} \\ &= (A - KC)P(t) + P(t)(A - KC)^T + MQM^T + KRK^T \end{aligned} \quad (2.11)$$

Puisque le système est stable on tend vers un régime permanent c'est-à-dire  $P(t) = P$  et  $\dot{P} = 0$  donc pour minimiser  $P(t)$  il suffit de minimiser  $\dot{P}(t)$  [51].

$$\frac{\partial (\text{Trace } \dot{P}(t))}{\partial K} = -CP(t) - P(t)C^T + 2KR = -P(t)C^T - P(t)C^T + 2KR = 0 \quad (2.12)$$

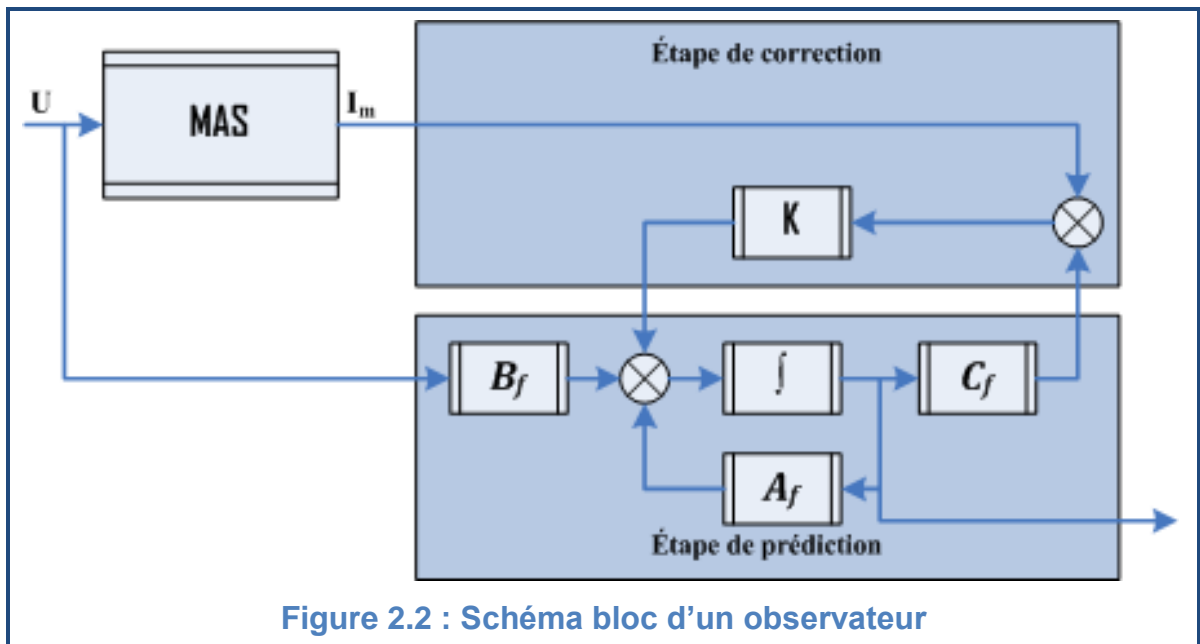
Alors :

$$K(t) = P(t)C^T R^{-1} \quad (2.13)$$

En reportant (2.7) dans (2.8), on obtient :

$$\dot{P}(t) = AP(t) + P(t)A^T - P(t)C^T R^{-1} CP(t) + MQM^T \quad (2.14)$$

Les équations (2.7), (2.13) et (2.14) constituent les équations du filtre de Kalman continu qu'il faut intégrer à partir de l'initialisation  $\hat{x}(t_0)$  et  $P(t_0)$ .



#### 2.4. Discrétisation du filtre de Kalman

En pratique, l'implantation du filtre de Kalman se fera sur un ordinateur numérique et donc en temps discret. On doit alors discrétiser le modèle de Kalman représenté par les équations (2.1) et (2.2):

##### 2.4.1. Equations récurrentes du système discrétisé

On note:  $x(kT) = x(k)$ .

D'après la solution générale, l'intégration de l'équation d'état entre l'instant :  $t_0 = kT$  et  $t = (k+1)T$  s'écrit [52]:

$$x(k+1) = e^{AT}x(k) + \left( \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} B d\tau \right) u(k) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} M w(\tau) d\tau \quad (2.15)$$

On utilise le changement de variable :  $(k+1)T - \tau = \alpha$

$$x(k+1) = e^{AT}x(k) + \left( \int_0^T e^{A\alpha} B d\alpha \right) u(k) + \int_0^T e^{A\alpha} M w((k+1)T - \alpha) d\alpha \quad (2.16)$$

### 2.4.2. Bruits d'état et de mesure discrétisés

$$w_d(k) = \int_0^T e^{A\alpha} M w((k+1)T - \alpha) d\alpha \quad (2.17)$$

$$v_d(k) = v(kT) \quad (2.18)$$

Alors :

$$\begin{aligned} Q_d &= E[w_d(k)w_d^T(k)] \\ &= E \left[ \int_0^T e^{A\alpha} M w((k+1)T - \alpha) d\alpha \int_0^T w^T((k+1)T - \tau) M^T e^{A^T \tau} d\tau \right] \\ &= \iint_0^T e^{A\alpha} M E[w((k+1)T - \alpha)w^T((k+1)T - \tau)] M^T e^{A^T \tau} d\alpha d\tau \end{aligned} \quad (2.19)$$

$$\begin{aligned} &= \iint_0^T e^{A\alpha} M Q \delta(\tau - \alpha) M^T e^{A^T \tau} d\alpha d\tau \\ &= \int_0^T e^{A\alpha} M Q M^T e^{A^T \alpha} d\alpha \end{aligned} \quad (2.20)$$

Si  $T$  est petit par rapport au temps de réponse du système alors :

$$Q_d = T M Q M^T \quad (2.21)$$

$$R_d = R/T \quad (2.22)$$

Les équations (2.16), (2.17) et (2.18) permettent de représenter le modèle d'état discrétisé par les équations suivantes :

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + M_d w_d(k) \end{cases} \quad (2.23)$$

$$\begin{cases} y(k) = C_d x(k) + D_d u(k) + v_d(k) \end{cases} \quad (2.24)$$

Avec :

$$A_d = e^{AT}, \quad B_d = \int_0^T e^{A\alpha} B d\alpha, \quad M_d = I, \quad C_d = C, \quad D_d = D \quad (2.25)$$

### 2.4.3. Equations récurrentes du filtre de Kalman

Soit:  $\hat{x}[k + 1|k]$  est l'état prédit auquel on associe la matrice de covariance de l'erreur de prédiction notée :  $P[k + 1|k]$

Et:  $\hat{x}[k + 1|k + 1]$  est l'état estimé auquel on associe la matrice de covariance de l'erreur d'estimation notée :  $P[k + 1|k + 1]$

Le filtrage de Kalman se fait alors en deux étapes [4]

#### 2.4.3.1. Etape de prédiction

Alors la prédiction à l'instant  $k+1$  se fait à partir de l'état  $\hat{x}[k|k]$  par le modèle déterministe :

$$\hat{x}[k + 1|k] = A_d \hat{x}[k|k] + B_d u(k) \quad (2.26)$$

Et la matrice de covariance de l'erreur de prédiction:

$$P[k + 1|k] = A_d P[k|k] A_d^t + Q \quad (2.27)$$

#### 2.4.3.2. Etape de correction

Le gain de Kalman est calculé par l'équation suivante:

$$K[k + 1] = P[k + 1|k] C^t (C P[k + 1|k] C^t + R)^{-1} \quad (2.28)$$

Et donc la prédiction est ensuite corrigée en tenant compte la nouvelle mesure  $Y[k + 1]$  avec :

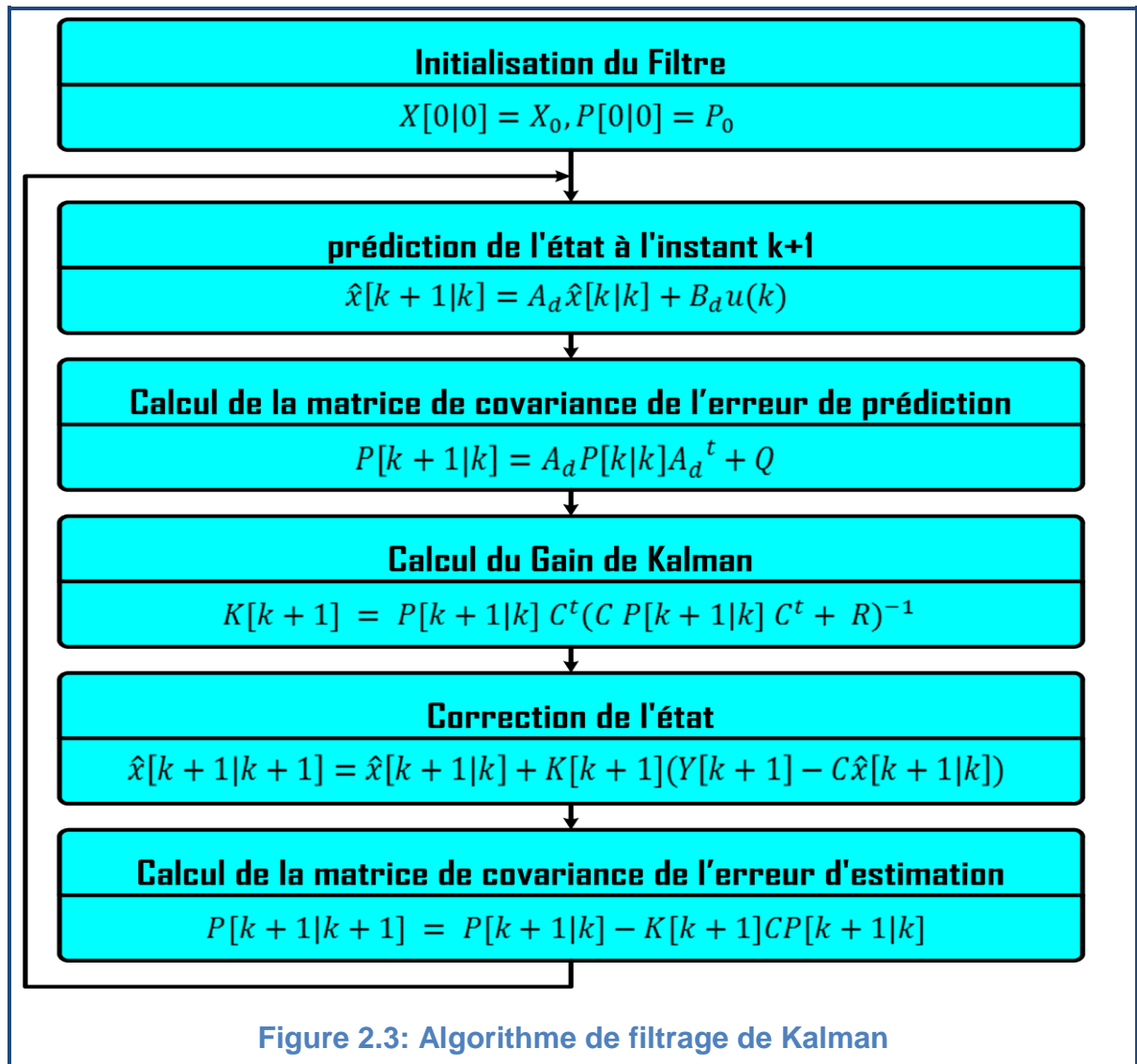
$$\hat{x}[k + 1|k + 1] = \hat{x}[k + 1|k] + K[k + 1](Y[k + 1] - C \hat{x}[k + 1|k]) \quad (2.29)$$

Et finalement la matrice de covariance de l'erreur d'estimation sera :

$$P[k + 1|k + 1] = P[k + 1|k] - K[k + 1] C P[k + 1|k] \quad (2.30)$$



La figure 2.3 représente l'algorithme général du filtrage de Kalman



### 2.5. Extension du filtre de Kalman

Le filtre de Kalman standard décrit par les équations (2.26), (2.27), (2.28), (2.29) et (2.30) permet l'estimation de l'état d'un système linéaire. Si on veut estimer en même temps ses paramètres, une solution consiste à étendre le vecteur d'état estimé aux paramètres souhaités [38, 14]. La modification majeure est que le système devient non linéaire. Pour y remédier, le filtre de Kalman est appliqué au modèle linéarisé à chaque instant.

L'algorithme de filtrage non linéaire porte alors le nom de Filtre de Kalman Étendu.

## 2.6.Application du filtre de Kalman a la mas

Les courants statoriques, les flux rotoriques et la vitesse mécanique sont généralement les paramètres nécessaires à la commande de la machine asynchrone et même au diagnostic.

En se basant sur le modèle représenté dans (1.94), et en mesurant les courants statoriques et la vitesse mécanique, la reconstruction des flux rotoriques qui ne sont pas mesurable peut être effectuée avec le filtre de Kalman. Depuis plusieurs années, un grand nombre de travaux ont permis de reconstruire ces flux [4, 13].

### 2.6.1.Modèle d'état continu

Donc l'équation d'état est de la forme:

$$\dot{x}(t)=Ax(t) + Bu(t) \quad (2.31)$$

Tel que :

$$A = \begin{bmatrix} -a_1 & 0 & a_3 & a_4\Omega_m \\ 0 & -a_1 & -a_4\Omega_m & a_3 \\ a_5 & 0 & -a_7 & -a_8\Omega_m \\ 0 & a_5 & a_8\Omega_m & -a_7 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.32)$$

### 2.6.2.Discrétisation du modèle continu

En se basant sur (2.27) et en utilisant les développements limités jusqu'à l'ordre deux, ce qui donne une précision jugée acceptable, on trouve le système discret suivant :

$$A_d = e^{AT} = I + \frac{AT}{1!} + \frac{(AT)^2}{2} = \begin{bmatrix} -ad_1 & ad_2 & ad_3 & ad_4 \\ -ad_2 & -ad_1 & -ad_4 & ad_3 \\ ad_5 & ad_6 & -ad_7 & -ad_8 \\ -ad_6 & ad_5 & ad_8 & -ad_7 \end{bmatrix} \quad (2.33)$$

$$B_d = \int_0^T e^{At} B dt = A^{-1}(e^{AT} - I)B = T \left( I + \frac{AT}{2} \right) B = \begin{bmatrix} bd_1 & 0 \\ 0 & bd_1 \\ bd_2 & 0 \\ 0 & bd_2 \end{bmatrix} \quad (2.34)$$

Avec:

$$\left\{ \begin{array}{l} ad_1 = 1 - a_1T + (a_1^2 + a_3a_5)\frac{T^2}{2} \\ ad_2 = a_4a_5\frac{T^2}{2} \\ ad_3 = a_3T\left(1 + (a_1 + a_7)\frac{T}{2}\right) + a_4\omega_m\frac{T^2}{2} \\ ad_4 = a_4T\left(1 + (a_1 + a_7)\frac{T}{2}\right) - a_3\omega_m\frac{T^2}{2} \\ ad_5 = a_5T\left(1 + (a_1 + a_7)\frac{T}{2}\right) \\ ad_6 = -a_5\omega_m\frac{T^2}{2} \\ ad_7 = 1 + a_7T + (a_7^2 + a_3a_5)\frac{T^2}{2} - a_8\omega_m\frac{T^2}{2} \\ ad_8 = -a_8T + (a_7a_5 - 2a_7\omega_m)\frac{T^2}{2} \\ bd_1 = b_1T\left(1 + a_1\frac{T}{2}\right) \\ bd_2 = b_1a_5\frac{T^2}{2} \end{array} \right. \quad (2.35)$$

Et le modèle discrétisé devient :

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + Q_d \\ y(k+1) = C_d x(k) + R_d \end{cases} \quad (2.36)$$

Avec :  $Q_d$ ,  $R_d$  sont calculées par les équations (2.21) et (2.22) respectivement.

### 2.6.3. Préréglage du filtre de Kalman

On a vu que la matrice du gain de Kalman  $K$  permet de minimiser l'erreur de covariance a posteriori  $P$ , et elle est calculé en fonction de la confiance que l'on a dans le modèle (caractérisée par la matrice  $Q_d$ ) relativement à la confiance que l'on a dans la mesure (caractérisée par  $R_d$ ). La matrice  $K$  peut être comprise comme un facteur de pondération.

Pour illustrer le rôle de la pondération, si  $R_d$  fixée, diminuer  $Q_d$  signifie que la prédiction est réalisée avec une faible incertitude; l'estimation optimale dépend donc fortement de l'étape de prédiction et non de la mesure. Au contraire, augmenter  $Q_d$  revient à réduire sa confiance dans la prédiction pour l'estimation de l'état, le gain de Kalman augmente donc pour donner plus de poids à la mesure.

Un bon réglage de  $Q_d$  et  $R_d$  garantit donc l'optimalité du filtre (par exemple la robustesse vis-à-vis des incertitudes paramétriques). Ce réglage se fait généralement par des méthodes numériques [21]. Dans cette partie on a programmé le filtre de Kalman sous Matlab et on a utilisé un algorithme génétique encore une fois pour rechercher le couple  $(Q_d, R_d)$  le plus approprié pour minimiser le critère déterministe suivant :

$$J_c = \sqrt{\frac{1}{N} \sum_{k=1}^N (i_{ds}[k] - \hat{i}_{ds}[k])^2} \quad (2.37)$$

#### 2.6.4. Algorithme du filtre de Kalman

L'implémentation du filtre de Kalman peut être faite sur un calculateur à virgule flottante, dans ce cas-là seulement la performance du filtre est importante mais le coût de la réalisation sera aussi important. Ce qui conduit à utiliser des calculateurs à virgule fixe moins performant, L'algorithme peut alors perdre ses performances. Dans le paragraphe suivant on va simplifier cet algorithme pour qu'on puisse l'implémenter dans un DSP à virgule fixe [13, 17] ou dans un FPGA [39, 40].

##### 2.6.4.1. Etape de prédiction

###### 2.6.4.1.1. Prédiction des états

Le développement de l'équation (2.26) permet d'écrire :

$$\begin{cases} \hat{i}_{ds}[k+1|k] = ad_1 \hat{i}_{ds}[k|k] + ad_2 \hat{i}_{qs}[k|k] + ad_3 \hat{\varphi}_{dr}[k|k] + ad_4 \hat{\varphi}_{qr}[k|k] + bd_1 V_{ds} \\ \hat{i}_{qs}[k+1|k] = -ad_2 \hat{i}_{ds}[k|k] + ad_1 \hat{i}_{qs}[k|k] - ad_4 \hat{\varphi}_{dr}[k|k] + ad_3 \hat{\varphi}_{qr}[k|k] + bd_1 V_{qs} \\ \hat{\varphi}_{dr}[k+1|k] = ad_5 \hat{i}_{ds}[k|k] + ad_6 \hat{i}_{qs}[k|k] + ad_7 \hat{\varphi}_{dr}[k|k] + ad_8 \hat{\varphi}_{qr}[k|k] + bd_2 V_{ds} \\ \hat{\varphi}_{qr}[k+1|k] = -ad_6 \hat{i}_{ds}[k|k] + ad_5 \hat{i}_{qs}[k|k] - ad_8 \hat{\varphi}_{dr}[k|k] + ad_7 \hat{\varphi}_{qr}[k|k] + bd_2 V_{qs} \end{cases} \quad (2.38)$$

###### 2.6.4.1.2. Calcul de la covariance de l'erreur de prédiction

La matrice de covariance de l'erreur de prédiction est une matrice symétrique, définie positive et de dimension 4X4, elle prend la forme (2.39) [4] et calculer par l'équation (2.27).

$$P[k + 1|k] = \begin{bmatrix} P_1[k + 1|k] & 0 & P_2[k + 1|k] & P_3[k + 1|k] \\ 0 & P_1[k + 1|k] & -P_3[k + 1|k] & P_2[k + 1|k] \\ P_2[k + 1|k] & -P_3[k + 1|k] & P_4[k + 1|k] & 0 \\ P_3[k + 1|k] & P_2[k + 1|k] & 0 & P_4[k + 1|k] \end{bmatrix} \quad (2.39)$$

Cette structure réduit la complexité algorithmique puisque le nombre de valeurs distinctes est réduit de seize à quatre. Ces paramètres sont calculés par l'équation suivante :

$$\left\{ \begin{array}{l} P_1[k + 1|k] = (ad_1^2 + ad_2^2)P_1[k|k] \\ \quad + 2(ad_1ad_3 + ad_2ad_4)P_2[k|k] \\ \quad + 2(ad_1ad_4 - ad_2ad_3)P_3[k|k] \\ \quad + (ad_3^2 + ad_4^2)P_4[k|k] \\ \quad + q_1 \\ P_2[k + 1|k] = (ad_1ad_5 + ad_2ad_6)P_1[k|k] \\ \quad + (ad_3ad_5 + ad_4ad_6 + ad_1ad_7 + ad_2ad_8)P_2[k|k] \\ \quad + (ad_4ad_5 - ad_3ad_6 + ad_1ad_8 - ad_2ad_7)P_3[k|k] \\ \quad + (ad_3ad_7 + ad_4ad_8)P_4[k|k] \\ P_3[k + 1|k] = (ad_2ad_5 - ad_1ad_6)P_1[k|k] \\ \quad + (ad_4ad_5 - ad_3ad_6 + ad_2ad_7 - ad_1ad_8)P_2[k|k] \\ \quad + (ad_1ad_7 - ad_3ad_5 + ad_2ad_8 - ad_4ad_6)P_3[k|k] \\ \quad + (ad_4ad_7 - ad_3ad_8)P_4[k|k] \\ P_4[k + 1|k] = (ad_5^2 + ad_6^2)P_1[k|k] \\ \quad + 2(ad_5ad_7 + ad_6ad_8)P_2[k|k] \\ \quad + 2(ad_5ad_8 - ad_6ad_7)P_3[k|k] \\ \quad + (ad_7^2 + ad_8^2)P_4[k|k] \\ \quad + q_2 \end{array} \right. \quad (2.40)$$

Avec  $q_1$  et  $q_2$  sont les composantes diagonales de la matrice  $Q_d$ .

#### 2.6.4.2. Etape de correction

##### 2.6.4.2.1. Calcul de la matrice de gain de Kalman

La matrice de gain  $K[k + 1]$  est une matrice de dimension  $4 \times 2$  et de la forme (2.41). Elle est définie précédemment par (2.28).

$$K[k + 1] = \begin{bmatrix} K_1 & 0 \\ 0 & K_1 \\ K_2 & -K_3 \\ K_3 & K_2 \end{bmatrix} \quad (2.41)$$

On reprend l'équation (2.28) :

$$K[k + 1] = P[k + 1|k]C^t(CP[k + 1|k]C^t + R)^{-1}$$

$$\text{Avec : } CP[k + 1|k]C^t = \begin{bmatrix} P_1[k + 1|k] & 0 \\ 0 & P_1[k + 1|k] \end{bmatrix}$$

C'est une matrice diagonale, son inversion se fait donc par des divisions scalaires.

$$\begin{cases} K_1[k+1] = P_1[k+1|k]/(P_1[k+1|k] + r) \\ K_2[k+1] = P_2[k+1|k]/(P_1[k+1|k] + r) \\ K_3[k+1] = P_3[k+1|k]/(P_1[k+1|k] + r) \end{cases} \quad (2.42)$$

Avec  $r$  est la composante diagonale de la matrice  $R_d$ .

#### 2.6.4.2.2. Estimation des états

Cette étape consiste à calculer l'erreur entre les prédictions et les mesures et d'effectuer ensuite une correction adéquate :

A partir de (2.29), l'erreur entre les courants prédit et celles mesurés peut être calculée par l'équation (2.43) :

$$\begin{cases} \varepsilon_d[k+1] = (\hat{i}_{ds}[k+1])_m - i_{ds}[k+1|k] \\ \varepsilon_q[k+1] = (\hat{i}_{qs}[k+1])_m - i_{qs}[k+1|k] \end{cases} \quad (2.43)$$

L'étape de correction se fait par l'équation (2.44) :

$$\begin{cases} \hat{i}_{ds}[k+1|k+1] = \hat{i}_{ds}[k+1|k] + K_1[k+1]\varepsilon_d[k+1] \\ \hat{i}_{qs}[k+1|k+1] = \hat{i}_{qs}[k+1|k] + K_1[k+1]\varepsilon_q[k+1] \\ \hat{\varphi}_{dr}[k+1|k+1] = \hat{\varphi}_{dr}[k+1|k] + K_2[k+1]\varepsilon_d[k+1] + K_3[k+1]\varepsilon_q[k+1] \\ \hat{\varphi}_{qr}[k+1|k+1] = \hat{\varphi}_{qr}[k+1|k] + K_3[k+1]\varepsilon_d[k+1] + K_2[k+1]\varepsilon_q[k+1] \end{cases} \quad (2.44)$$

#### 2.6.4.2.3. Calcul de la covariance de l'erreur d'estimation

La matrice  $P[k+1|k+1]$  est calculée par l'équation (2.30) tel que :

$$P[k+1|k+1] = \begin{bmatrix} P_1[k+1|k+1] & 0 & P_2[k+1|k+1] & P_3[k+1|k+1] \\ 0 & P_1[k+1|k+1] & -P_3[k+1|k+1] & P_2[k+1|k+1] \\ P_2[k+1|k+1] & -P_3[k+1|k+1] & P_4[k+1|k+1] & 0 \\ P_3[k+1|k+1] & P_2[k+1|k+1] & 0 & P_4[k+1|k+1] \end{bmatrix} \quad (2.45)$$

Avec :

$$\begin{cases} P_1[k+1|k+1] = K_1[k+1] \\ P_2[k+1|k+1] = K_2[k+1] \\ P_3[k+1|k+1] = K_3[k+1] \\ P_4[k+1|k+1] = P_4[k+1|k] - (P_2[k+1|k]K_2[k+1] + P_3[k+1|k]K_3[k+1]) \end{cases} \quad (2.46)$$

## 2.7.Conclusion

Ce chapitre a été consacré à la synthèse d'un filtre de Kalman dédié à l'observation de l'état de la machine asynchrone à cage d'écureuil. Il a permis l'estimation des flux rotoriques à partir de la mesure des courants et des tensions statoriques et de la vitesse rotorique. Dans un premier temps, le cas continu a été traité. Ensuite, la discrétisation du modèle de la MAS à l'aide de la méthode de Taylor a permis d'appliquer le cas discret du filtre de Kalman.

Le préréglage du filtre est un point délicat et peu abordé dans la littérature scientifique du génie électrique. Pour résoudre ce problème, une méthode méta-heuristique, basée sur un algorithme génétique, a été utilisée et les résultats obtenus sont satisfaisants. Les calculs théoriques du FK faisant intervenir une inversion matricielle qui ne se prête pas à une implémentation par calculateur numérique, un algorithme efficace permettant de simplifier ses équations a été développé. L'implémentation numérique de cet algorithme dans une cible FPGA fera l'objet du prochain chapitre.

## **CHAPITRE 3**

### **PRESENTATION DES CIRCUITS FPGA**

#### 3.1.Introduction

Dans la réalité industrielle, la complexité des systèmes, ainsi que celle des traitements à réaliser, nécessite souvent le recours à des outils numériques puissants (calculateurs numériques).

Le choix d'un support matériel pour un traitement numérique particulier est évidemment fonction de la complexité du système, de la fonction à réaliser ainsi que des performances exigées (fréquence de traitement, capacité mémoire, consommation d'énergie...).

Les systèmes à microprocesseurs présentent toujours la solution la plus connue, la plus simple à utiliser et la moins chère. Ils sont basés sur un microprocesseur qui est chargé de traiter toutes les informations. Il constitue l'unité centrale de traitement et gère la mémoire centrale (RAM, ROM et EPROM) ainsi que les circuits d'interface qui assurent la communication avec le monde externe.

Les ASICs, qui sont des circuits Intégrés pour Applications Spécifiques, ont réduit considérablement la consommation d'énergie et augmenté la vitesse de fonctionnement et la fiabilité des systèmes. Ils ont également permis la minimisation de taille de la puce ainsi que la protection industrielle. Leur développement nécessite la création d'un ou plusieurs masques. Cette étape se fait en usine par des personnes qualifiées et un matériel spécifique. Cette solution coûte chère et le temps de mise au point d'un circuit particulier est long.

Les industriels et les chercheurs avaient besoin de circuits de même performance que les ASICs mais moins coûteux et plus faciles à utiliser. La solution a été les circuits logiques programmables (PLD) qui ne demandent pas de



fabrication spéciale en usine, ni de systèmes de développement coûteux avec la possibilité de reconfiguration. La performance de ces circuits varie selon les différentes familles (SPLD, CPLD ou FPGA) [41].

FPGA signifie “réseaux de portes programmables sur site”. Ce sont des composants électroniques qui comportent un grand nombre de fonctions logiques de base (ET, OU, etc.) que l'utilisateur peut combiner entre elles en fonction des besoins de son application. Elles ont déjà plus de 25 ans de carrière mais l'intégration de nouvelles fonctions (mémoires, processeurs, etc.) leur permet de sortir des applications classiques de traitement d'opérations de logique booléenne et d'accéder aux applications plus complexes de traitement du signal et d'images [42].

Dans ce chapitre on va présenter les deux architectures des systèmes à microprocesseurs ; l'architecture de Von Neumann et celle de Harvard ainsi que les différents types de circuits logiques programmables. Ensuite on va présenter plus de détails sur les circuits FPGA qui ont été utilisés dans ce projet pour l'implémentation du FK.

### 3.2.Les systèmes à microprocesseurs

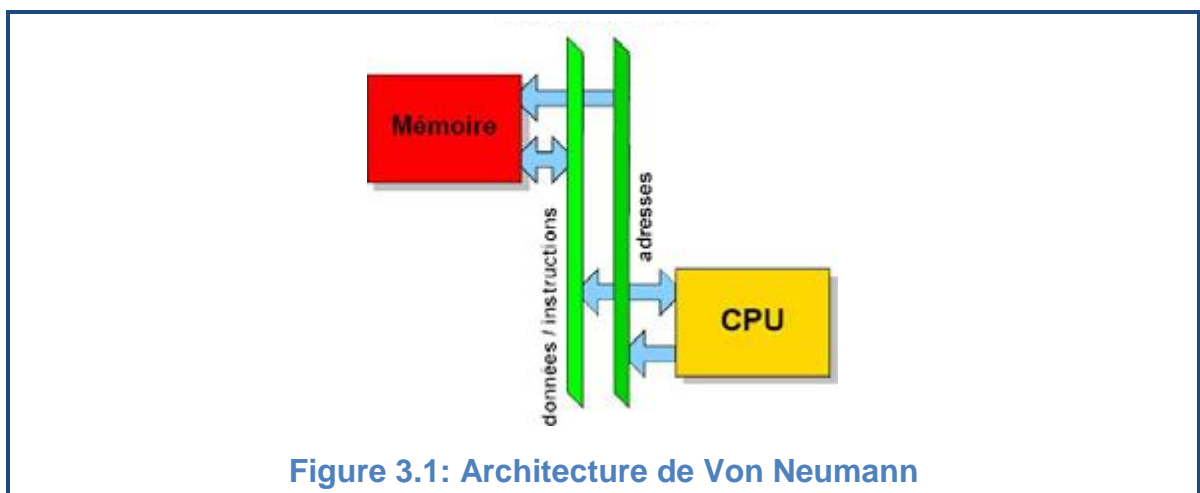
Un processeur est un composant intégré ou une partie de composant susceptible d'exécuter des instructions selon un programme d'instructions préétabli sur des données numériques. Le programme, et parfois les données, sont enregistrés dans des mémoires numériques.

L'architecture est ce qui détermine d'emblée et de manière définitive les principales caractéristiques du processeur en particulier la rapidité d'exécution des instructions. Elles sont liées au nombre de bus internes, qui sont des liaisons parallèles à N bits, à la valeur même de N, aux opérations possibles sur les données transitant sur les bus et enfin à la puissance de calcul de l'unité centrale.

Il existe deux architectures différentes de systèmes à microprocesseur, l'architecture de Von Neumann et celle de Harvard [43].

### 3.2.1. Architecture de Von Neumann

Dans l'architecture de la machine de Von Neumann, le programme et les données sont enregistrés sur la même mémoire. Chaque instruction contient la commande de l'opération à effectuer et l'adresse de la donnée à utiliser, il faut donc souvent plusieurs cycles d'horloge pour exécuter une instruction. La Figure 3.1 indique une architecture simple de Von Neumann, constituée d'un bus de données et de programme et d'un bus d'adresses.

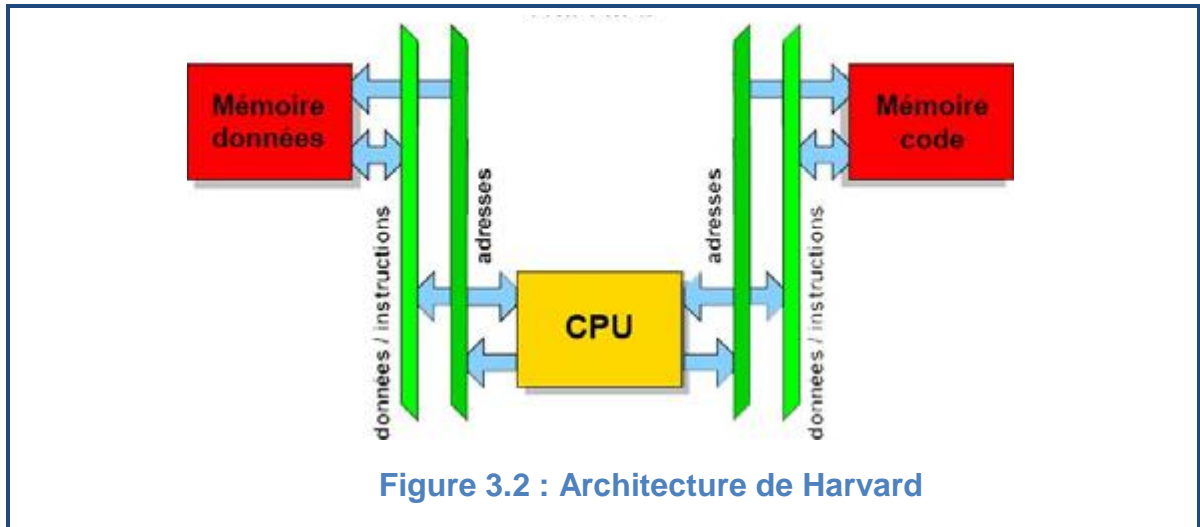


On voit que les échanges s'effectuent de manière simple entre l'unité arithmétique et logique, c'est-à-dire l'unité centrale et la mémoire unique, par un bus transitant les codes de programme et les données. On a ainsi des données «collées» aux instructions.

Les microprocesseurs et beaucoup de microcontrôleurs (exemples : la famille Motorola 68XXX, la famille Intel 80X86) utilisent cette architecture car elle est très souple pour la programmation.

### 3.2.2. Architecture de Harvard

Dans l'architecture dite de Harvard (car mise au point dans cette université américaine en 1930), on sépare systématiquement la mémoire de programme de la mémoire des données : l'adressage de ces mémoires est indépendant. La Figure 3.2 indique une architecture simple de Harvard, constituée d'un bus de données, d'un bus de programme et de deux bus d'adresse.

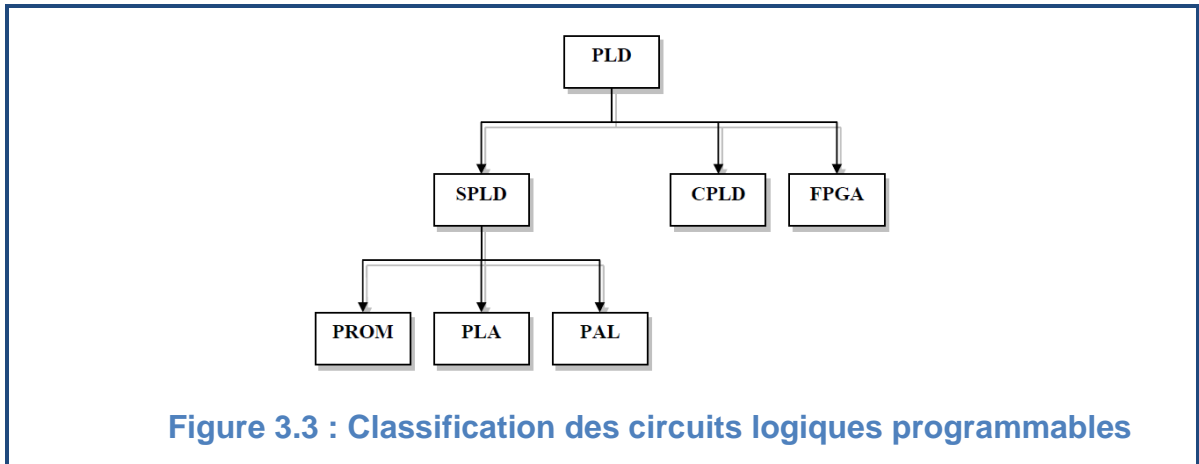


On voit que les échanges s'effectuent de manière double entre l'unité centrale et les deux mémoires, ce qui permet une grande souplesse pour l'enregistrement et l'utilisation des données. D'ailleurs, la mémoire de programme est également utilisée en partie comme mémoire de données pour obtenir encore plus de possibilités de traitement avec des algorithmes complexes.

L'architecture Harvard est utilisée généralement dans des microprocesseurs spécialisés pour des applications temps réels (DSP). Cette architecture requiert deux fois plus de bus de données, d'adresses, et donc de broches sur la puce. Or un des éléments augmentant le coût de productions des puces est précisément le nombre de broches à implanter.

### 3.3.Circuits logiques programmables

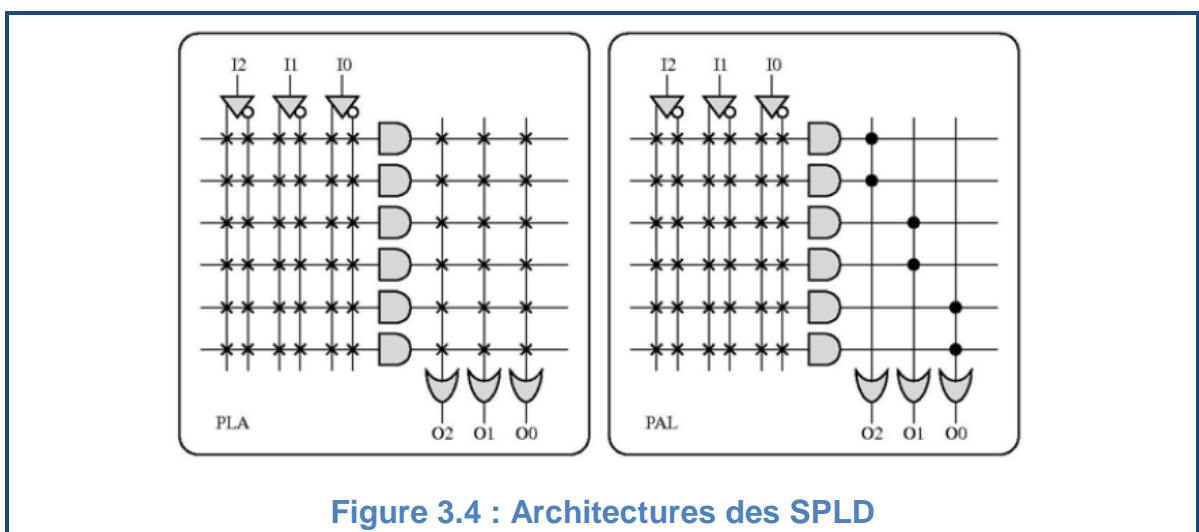
Les circuits programmables sont apparus en 1970, et depuis ils ont évolué d'une manière étonnante et plusieurs familles se sont distinguées. Il existe dans la littérature plusieurs manières de les classer. Nous choisirons l'approche illustrée à la figure 3.3 consistant à séparer les PLDs (Programmable Logic Device) en trois sous-classes, à savoir les SPLDs (Simple Programmable Logic Device), les CPLDs (Complex Programmable Logic Device) et les FPGAs (Field Programmable Gate Array).



Nous allons décrire les différents types de circuits programmables, dans l'ordre chronologique de leur apparition, correspondant également à leur complexité, en commençant par les SPLD, les CPLDs, puis les FPGAs.

### 3.3.1.SPLD

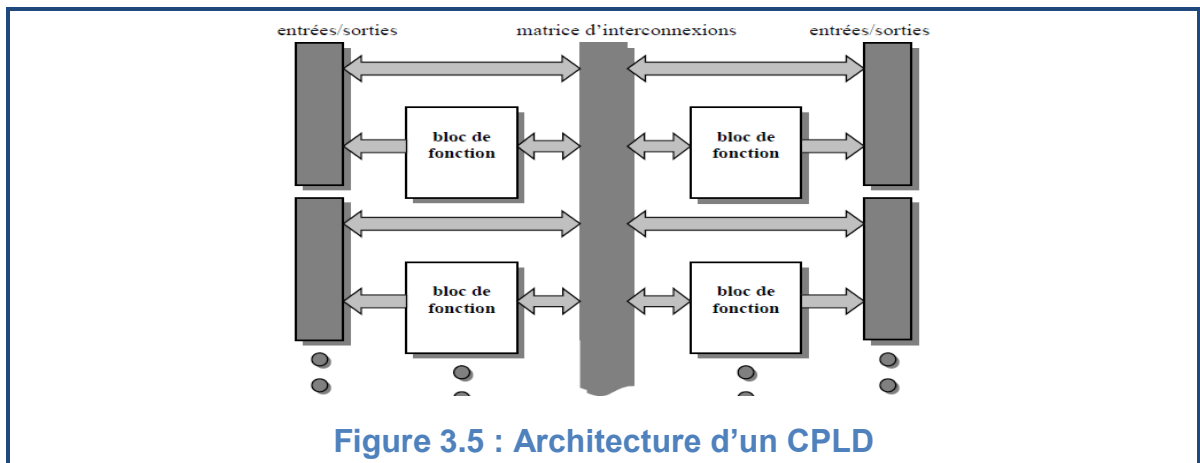
Les SPLDs (Simple Programmable Logic Device), dans une description haut niveau, sont composés d'une grille de portes ET et d'une grille de portes OU, les deux étant reliées. Les entrées du système peuvent être connectées aux portes ET, et le résultat des portes OU correspond à la sortie du système. Dans ces circuits, les connexions sont préexistantes, les différentes lignes étant reliées par des fusibles ou des transistors. En brûlant certains de ces fusibles, ou en programmant les transistors, il est alors possible de réaliser différentes fonctions logiques. Il existe dans cette famille les PROMs, les PLAs ainsi que les PALs [41].



Les SPLDs présentent deux limitations majeures, à savoir l'impossibilité de réaliser des fonctions à plusieurs niveaux et celle de ne pouvoir partager les produits de différentes fonctions.

### 3.3.2.CPLD

Les CPLDs (Complex Programmable Logic Device), apparus au début des années 80, sont le résultat de l'évolution des PLDs. Ils permettent l'implémentation de systèmes nettement plus complexes, et sont composés d'éléments de base programmables, connectés entre eux par un réseau d'interconnexions relativement simple. Ces éléments de base sont du type SPLD [41]. La technologie de programmation des CPLDs dépend évidemment du constructeur, et peut être de type EPROM, EEPROM, FLASH ou SRAM (figure 3.5).



### 3.3.3.FPGAs

Nous présentons dans cette partie les circuits PLD de type FPGA (Field Programmable Gate Array). Au début des années 80, les développeurs disposaient des circuits de type PLD, facilement configurables, mais ne pouvant contenir des architectures considérablement complexes. Les ASICs, quant à eux, supportaient des systèmes de grande complexité, mais n'avaient pas les propriétés de configuration des PLDs ni celles de la reconfiguration. Il manquait donc un type de circuits permettant la réalisation de systèmes complexes, tout en offrant une reconfiguration rapide et peu coûteuse. Ces circuits ont été complexifiés par la suite en intégrant de plus en plus d'autres composants tels que RAM, multiplieurs, processeurs...

Il existe plusieurs fabricants des circuits FPGA. Mais ceux de la société Xilinx occupent à peu près la moitié du marché à cause de leur diversité ainsi que leurs performances.

### 3.4.Familles des FPGA de Xilinx

La tendance des dernières générations est de cibler certains créneaux porteurs du marché comme la taille, le type et le nombre de cellules qui varient suivant les familles de composants [44].

Les FPGA du constructeur Xilinx sont divisés en deux gammes :

- FPGA hautes performances : gamme **Virtex**: VirtexE, VirtexEM, Virtex2, Virtex2Pro, Virtex4, Virtex5...
- FPGA pour la fabrication en grande série: gamme **Spartan**: Spartan2, Spartan2E, Spartan3, Spartan3A, Spartan3AN, Spartan3DSP, Spartan3E...

### 3.5.Langage de description matériel

Afin de pouvoir finaliser un FPGA, il est nécessaire d'utiliser un langage de description du matériel **HDL (Hardware Description Language)** ou bien un outil de saisie graphique. Après compilation de cette description, on obtient un fichier de configuration pour le FPGA choisi.

En 1987, une première standardisation des langages de descriptions du hardware est créée. Cette standardisation permet d'étendre la conception de circuit aux particuliers. En effet, avant cette standardisation, les entreprises utilisaient leur propre code propriétaire. Cela donna naissance à deux langages :

- Le VHDL, IEEE.1076, facile à appréhender mais peu compact, principalement utilisé en Europe.
- le Verilog, proche du C. Il a eu l'acceptation IEEE.1364 en 1995, principalement utilisé aux Etats-Unis depuis les années 1995.

### 3.6. Etapes de conception d'un projet sur FPGA

#### 3.6.1. Description du circuit

La description des circuits logiques se fait la plupart de temps à l'aide des langages de description de matériel tels que VHDL. Souvent, lorsque l'utilisateur souhaite rester proche du matériel, le circuit se décrit de façon schématique.

Il existe d'autres formalismes permettant de décrire un circuit logique, souvent proposés par un producteur de FPGA particulier, comme par exemple l'éditeur graphique FSM chez Xilinx.

La description d'un circuit peut être faite exclusivement à l'aide d'un formalisme bien particulier, ou combiner des formalismes différents.

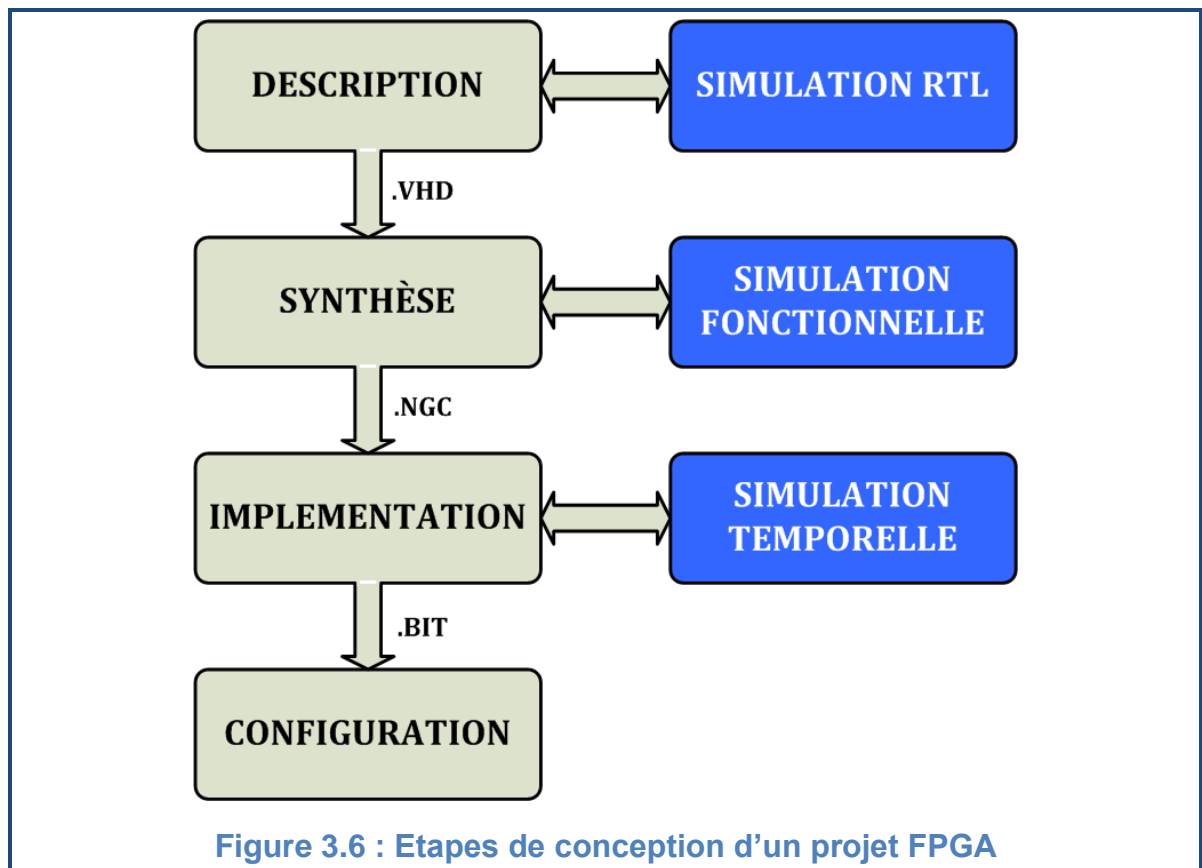
Les différentes parties d'un même circuit logique complexe peuvent donc être décrites de façons différentes avant d'être traduites en une configuration particulière pour un (ou plusieurs) FPGA cible.

Le développement d'un projet en VHDL nécessite l'utilisation de trois outils principaux:

- Le simulateur permet de simuler la description VHDL avec un fichier de simulation appelé «**test-bench**»; cet outil interprète directement le langage VHDL.
- L'objectif du synthétiseur est très différent; il doit traduire le comportement décrit en VHDL en fonctions logiques de base, celles-ci dépendent de la technologie choisie.
- L'intégration finale dans le circuit cible est réalisée par l'outil de placement et routage. Celui-ci est fourni par le fabricant de la technologie choisie.

Le **ModelSim** de **MentorGraphics** est parmi les outils les plus utilisés pour simuler une description VHDL. Il est particulièrement intéressant lorsqu'on veut visualiser des signaux analogiques. Alors que l'outil **ISE** de **Xilinx** peut être utilisé pendant toutes les étapes de la procédure de développement.

La figure 3.6 donne les différentes étapes nécessaires au développement d'un projet sur circuit FPGA.



### 3.6.2.Simulation comportementale

La simulation permet de vérifier le comportement d'une description avant ou après implémentation dans le composant cible. Elle représente une étape essentielle qui nous fera gagner du temps lors de la mise au point sur la carte. Il faut juste noter qu'un projet peut être simulé même s'il n'est pas synthétisable.

Pour simuler une description VHDL, il faut lui associer les différents stimuli, décrits également en même langage. La description du circuit et le fichier de stimulus sont ensuite passés au logiciel de simulation, permettant d'effectuer une vérification de la description.

Lors de l'étape de simulation comportementale, on valide l'application indépendamment de l'architecture et des temps de propagation du futur circuit cible.



### 3.6.3.Synthèse

L'étape de synthèse permet de passer d'un niveau d'abstraction élevé « VHDL » vers un niveau d'abstraction plus bas qui est le niveau **RTL (*Registre Translate Level*)** adapté au langage machine. Elle fournit au processus d'implémentation un fichier contenant la description physique de circuit, sous forme d'un ensemble de primitives (briques de base) des circuits FPGAs (il s'agit d'un fichier portant une extension **NGC**).

### 3.6.4.Simulation fonctionnelle

La phase de simulation après synthèse (simulation fonctionnelle) valide l'application sur l'architecture du circuit cible. Elle ne tient pas compte des capacités de liaison dues au routage entre les différentes cellules. Elle permet donc de vérifier uniquement la validité du circuit par rapport au cahier des charges d'un point de vue fonctionnel et non d'un point de vue temporel.

### 3.6.5.Implémentation

Mis à part la description du circuit, le processus d'implémentation passe par les sous-étapes suivantes:

- Le **mapping** consiste à traduire le circuit logique obtenu par le processus de synthèse en mémoires des cellules logiques élémentaires d'un circuit FPGA particulier.
- Lors de la sous-étape de **placement**, une cellule logique existante, avec sa position unique au sein du FPGA cible, est attribuée à chaque cellule logique nécessaire pour la réalisation du circuit. Ceci respecte la contrainte des longueurs minimales des connexions entre les différentes cellules.
- Le **routage** permet d'établir les différentes connexions physiques entre les cellules élémentaires placées dans le circuit afin de réaliser le fonctionnement attendu.

### 3.6.6.Simulation temporelle

Il s'agit de vérifier la fonctionnalité du circuit en prenant en compte par un calcul estimatif les longueurs d'interconnexion et les retards apportés par les capacités parasites liées au routage. Elle vérifie donc que la fonctionnalité n'a pas

été modifiée par l'introduction des délais de propagation et reste conforme au cahier des charges.

### 3.6.7. Configuration

La dernière étape consiste à générer un fichier binaire de configuration appelé **Bitstream** téléchargeable dans une PROM. Ce fichier contient les informations fournies au composant FPGA Xilinx afin qu'il prenne la configuration souhaitée.

### 3.7. Applications des FPGA

Depuis quelque temps, un certain nombre de chercheurs intéressés par les techniques d'optimisation méta heuristiques ont utilisé des puces FPGA dans leurs expériences. Elles sont pour eux intéressantes à plus d'un titre. Elles permettent de faire évoluer des algorithmes génétiques par exemple à des temps d'exécution très faibles.

Les industriels considèrent les FPGA comme un moyen de prototypage rapide et peu cher. En effet, les puces FPGA peuvent être reprogrammées sans nécessiter de matériel lourd. Cela leur permet de tester un nouveau circuit sans avoir à racheter des composants ou de les reconcevoir à chaque amélioration.

La technologie FPGA intéressa aussi l'armée de par la possibilité de programmation des puces en cours de fonctionnement. Globalement, l'armée est intéressée par ces puces FPGA pour les instruments de vols et les systèmes de communications militaires nécessitant la possibilité d'être reconfigurés et de fonctionner dans une large gamme de températures.

Une puce FPGA de marque Xilinx fut également choisie par la Nasa pour l'exploration de Mars par les **Rovers Spirit** et **Opportunity**. Ces puces appartiennent à la famille Virtex et sont spécialement étudiées pour résister aux radiations solaires qui frappent Mars continuellement (la faible atmosphère qui entoure Mars ne peut stopper ces radiations et une puce classique serait détruite en peu de temps). Elle se charge de contrôler le bras articulé du Rover et les divers appareils de mesure qui le composent.

### 3.8. Apport des FPGA aux problèmes de contrôle

Malgré tout, les FPGA ne s'imposent pas si facilement dans les applications de traitement du signal. La seule raison à cela c'est que pour programmer efficacement un FPGA, il faut repenser les algorithmes fondamentaux du traitement du signal. De plus, il ne suffit pas d'avoir des compétences en logiciel, il faut également des compétences en électronique numérique (au niveau "matériel"), et rares sont les ingénieurs à avoir cette double compétence "matériel/logiciel". Les concepteurs essaient toujours de résoudre ce problème ce qui donne naissance à des nouveaux outils permettant de passer aisément du langage C, celui de la communauté des développeurs logiciels, au VHDL des concepteurs de matériels tels :

- **DSPBuilder** chez **Altera**.
- **SystemGenerator** chez **Xilinx**.
- **DSPSynthesis** chez **AccelChip**.
- **Catapult** chez **MentorGraphics**.

L'introduction des FPGA dans le domaine de contrôle est un remède à plusieurs problèmes algorithmiques et surtout économiques, ceci peut être résumé dans les points suivants :

- Eviter l'inconvénient majeur des solutions analogiques classiques qui réside dans l'influence des variations paramétriques engendré par la sensibilité aux perturbations externes comme la chaleur.
- Pas d'entretien qui nécessite du temps et des pertes d'ordre économiques à l'inverse des solutions analogiques
- C'est une technologie embarquée hautement intégrée, elle réduit donc l'espace utilisé ainsi que la consommation d'énergie.
- L'intégration sur une seule puce de plusieurs systèmes grâce à la configuration dynamique avec une grande flexibilité pour un changement de la structure.
- La possibilité de réduire fortement les délais de développement et de commercialisation.

### 3.9.Conclusion

Dans ce chapitre nous avons présenté en premier temps les différents types de calculateurs numériques ainsi que leurs différentes architectures. Ensuite une description générale des circuits FPGA de Xilinx a été présentée. Ceci nous a permis de conclure que la technologie FPGA s'inscrit au sommet de l'évolution des composants logiques et le besoin croissant de composants plus performants, plus économiques et disponibles en grandes quantités est le grand axe du progrès qui est disponible dans ces circuits. On a donné également un court rappel sur l'historique des langages de description matériel (VHDL et Verilog) ainsi que leurs standardisations.

Finalement, nous avons présenté les différents modes de simulation d'une description VHDL, les étapes principales de l'implémentation du circuit dans une cible FPGA ainsi que l'apport des FPGA aux problèmes de contrôle.

Le prochain chapitre contient la description hardware du filtre de Kalman, les différentes techniques utilisées pour implémenter ce filtre ainsi que les résultats obtenus.

## CHAPITRE 4

### ARCHITECTURE DU FK ET TECHNIQUES D'IMPLEMENTATION

#### 4.1.Introduction

La mise en œuvre de l'algorithme du filtre de Kalman nécessite d'effectuer des calculs intensifs impliquant des additions, des multiplications et des divisions. Il implique aussi la nécessité de choisir un format convenable pour les données. Ces opérations peuvent être effectuées à l'aide de différents outils numériques. De nombreux chercheurs ont utilisé le DSP pour parvenir à cette fin, avec des performances satisfaisantes.

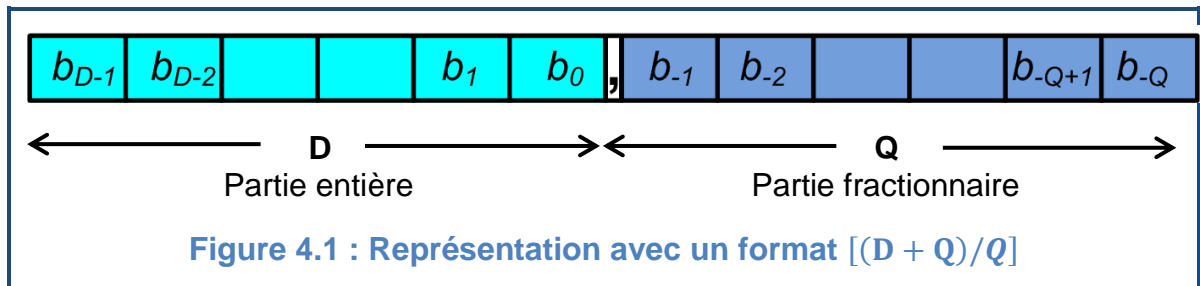
L'utilisation des FPGA pour implémenter le filtre de Kalman peut être considérée comme une solution très attrayante. Cette solution implique la possibilité de décrire l'architecture de calcul avec un niveau de parallélisme très élevé. Ce qui peut réduire considérablement le temps de calcul et donc augmenter la fréquence de traitement. Certains travaux publiés ont été consacrés à l'exploration de cette voie. Par contre, l'implémentation sur circuit FPGA du FK pour la machine asynchrone a été très peu traitée par les chercheurs.

Dans ce chapitre, l'implémentation de filtre de Kalman pour la reconstruction des variables d'états d'une machine asynchrone est présentée. L'analyse des résultats obtenus et des ressources consommées du circuit FPGA cible sont données.

#### 4.2.Représentation des données

Une représentation avec un format à virgule fixe a été utilisée. Cette représentation est caractérisée par l'association d'un nombre fini de bits. Elle contient une partie entière de  $D$  bits et une partie fractionnaire de  $Q$  bits.

Ce type de représentation est référencé par le format  $s[(D + Q)/Q]$  lorsqu'il s'agit d'une représentation signée et par le format  $u[(D + Q)/Q]$  lorsqu'il s'agit d'une représentation non signée. Ce type de format indique que le nombre de bits total est égal à  $(D + Q)$ , où les  $D$  bits les plus significatifs représentent la partie entière tandis que les  $Q$  bits restants représentent la partie fractionnaire. Ce format est représenté dans la figure 4.1.



Deux types de données sont utilisées dans ce projet; les constantes et les variables.

#### 4.2.1. Les constantes

Les constantes utilisées dans le présent travail sont les éléments des matrices  $A_d, B_d, Q_d, R_d$  (équations 2.33, 2.34, 2.21, 2.22) ainsi que la matrice de transformation de Park (équation 1.12). Ces constantes peuvent être représentées en virgule fixe par le format représenté par la figure 4.1.

La plus petite constante utilisée est égale à :  $6 \cdot 10^{-8}$ . Tandis que la plus grande est égale à : **0.987**. Donc, ces grandeurs peuvent être codé sur **26 Bits**; un bit pour la partie entière, un bit de signe et **24 bits** pour la partie fractionnaire. Ainsi, la représentation proposée est :  $s[26|24]$ .

#### 4.2.2. Les variables

Les variables sont des grandeurs signées et représentent les variations des différents signaux telles que : les tensions, les courants, les flux et la vitesse ainsi que les variations des éléments des matrices intermédiaires telles que : la matrice des erreurs, les matrices de covariances de l'erreur et la matrice des gains de Kalman.

Ces variables peuvent atteindre une valeur maximale égale à **381**. Donc leurs parties entières peuvent être codées sur **9 bits** plus un bit de signe. La partie fractionnaire prend le même format que celle des constantes. Et la représentation proposée pour les signaux est:  $s[34|24]$ .

#### 4.3. Troncature des résultats de multiplications

Les éléments des différentes matrices sont codés sur **26 bits** avec une partie entière qui contient un seul bit significatif qui est égal à zéro et un bit de signe ; et une partie fractionnaire de **24 bits** . Par contre, les différents signaux sont codés sur **34 bits** avec une partie entière qui contient **9 bits** significatifs et un bit de signe et une partie fractionnaire de **24 bits**.

La taille d'un résultat de multiplication de deux nombres binaires est la somme des tailles de ces deux nombres. Dans ce projet, deux types de multiplication ont été utilisés.

- Pour le premier type, la multiplication est effectuée entre deux coefficients de la matrice d'état représentés dans le format  $s[26|24]$  et le résultat, après troncature, doit être dans ce même format.
- Pour le deuxième, elle est effectuée entre un coefficient de format  $s[26|24]$  et un échantillon de signal de format  $s[34|24]$  et le résultat, après troncature, aura le deuxième format.

##### 4.3.1. Premier type de troncature

Dans le premier cas, le résultat est de **52 bits** avec une partie fractionnaire de **48 bits**. Puisque tous les coefficients sont inférieurs à un, le résultat de multiplication sera aussi inférieur à un et sa partie entière contient **4 bits** avec :

- 1 *bit* significatif.
- 1 *bit* de signe.
- Les bits restants seront une extension du signe.

La troncature des deux bits les plus significatifs de la partie entière (qui sont toujours une extension de signe) et des **24 bits** les moins significatifs de la partie fractionnaire (diminuer la précision) permet de représenter le résultat final de la multiplication avec le même format que celui des entrées, c'est le format  $s[26|24]$ .

#### 4.3.2. Deuxième type de troncature

Dans le deuxième cas, le résultat est de 60 *bits* avec une partie fractionnaire de 48 *bits*.

Puisque tous les coefficients sont inférieurs à un, le résultat de multiplication sera toujours inférieur à la valeur de l'échantillon du signal et la taille de la partie entière de ce résultat ne dépasse jamais neuf bits. Cette partie sera donc codé sur 12 *bits* avec :

- 9 *bits* significatifs.
- 1 *bit* de signe.
- Les bits restants seront une extension du signe.

La troncature des deux bits les plus significatifs de la partie entière (qui sont toujours une extension de signe) et des 24 *bits* les moins significatifs de la partie fractionnaire (diminuer la précision) permet de représenter le résultat final de la multiplication avec le même format que celui du signal d'entrée, c'est le format  $s[34|24]$ .

L'une ou l'autre de ces deux méthodes de troncatures doit être effectuée après chaque multiplication afin de récupérer le format des différents signaux pour simplifier l'architecture et pour diminuer la consommation des ressources.

#### 4.4. Synchronisation des signaux internes

Afin de simplifier la gestion d'horloge, les architectures proposées pour toutes les équations du filtre sont des architectures asynchrones. Comme chaque bloc du filtre a un temps de propagation de l'information différent de ceux des autres blocs, une synchronisation de tous les signaux intermédiaires devient alors nécessaire pour pouvoir les relier.

Des registres synchrones de 34 *bits* à chargement parallèle ont été utilisés à la sortie de chaque bloc pour maintenir les valeurs des différents signaux constantes jusqu'à la prochaine impulsion d'horloge.

L'horloge de synchronisation prend comme période, la période la plus longue entre les temps de propagation de l'information dans les différents blocs.

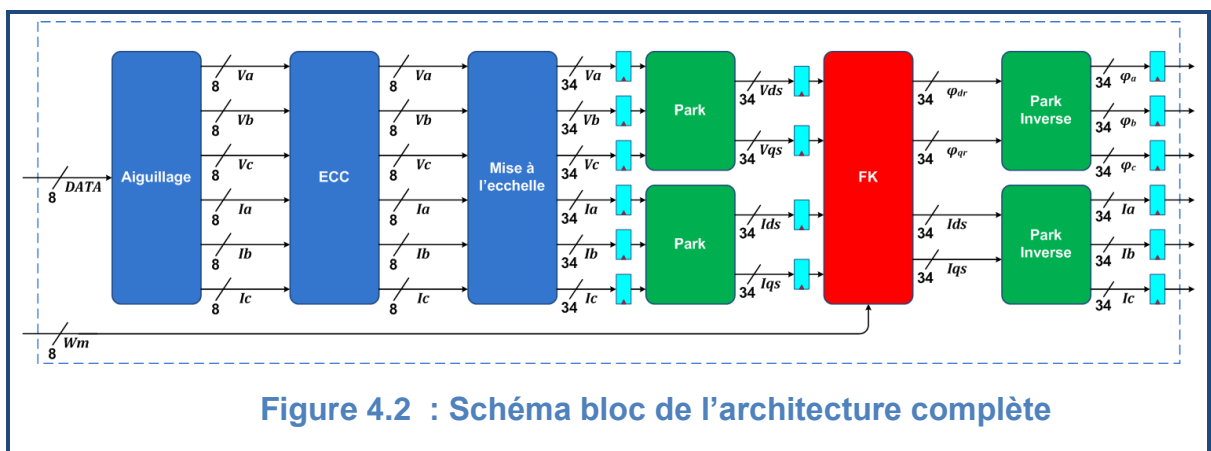


#### 4.5. Présentation de l'architecture

L'architecture proposée se divise en cinq étapes principales :

- Prétraitement des signaux d'entrée.
- Transformation de Park.
- Calcul de la matrice d'état.
- Filtrage de Kalman.
- Transformation de Park Inverse.

La figure 4.2 représente le schéma bloc de l'architecture proposée.



##### 4.5.1. Prétraitement des signaux d'entrée

Les signaux d'entrée sont proviennent d'un convertisseur analogique/numérique. Ce dernier donne un signal numérique à valeurs entières positives codées sur  $N$  bits.

Un aiguillage de signaux doit être effectué afin de minimiser le nombre d'Entrées/Sorties utilisés dans le circuit FPGA. Cette étape peut être effectuée par le même circuit de conversation analogique/numérique.

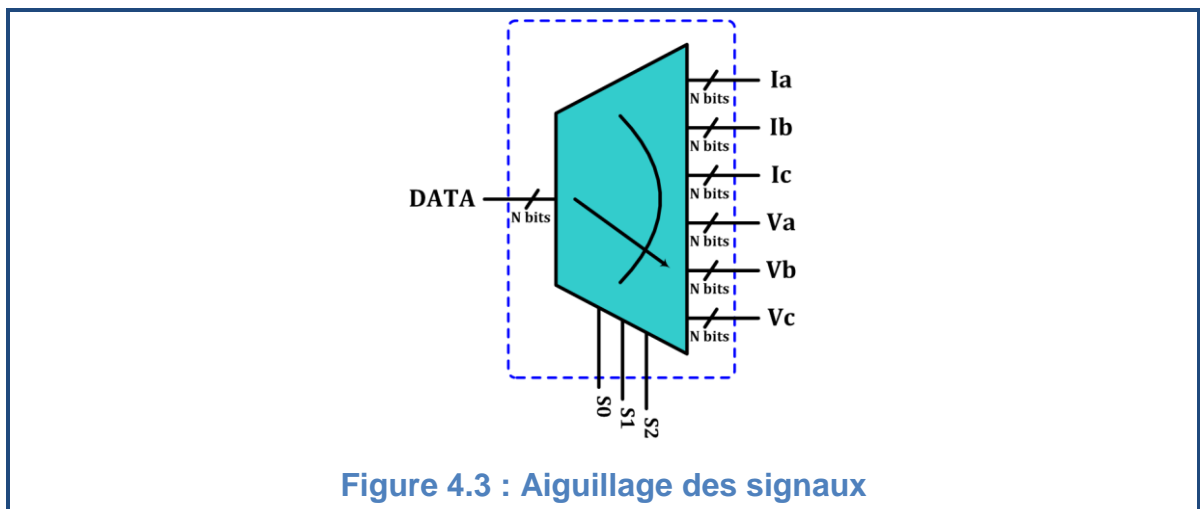
Avant l'utilisation de ces signaux trois étapes de prétraitement sont alors indispensables :

- Aiguillage des signaux
- Elimination de la composante continue.
- Mise à l'échelle.

#### 4.5.1.1. Aiguillage des signaux

L'information reçue par le FPGA contient donc plusieurs signaux numériques (trois courants et trois tensions). Alors on doit tout d'abord aiguiser ces signaux.

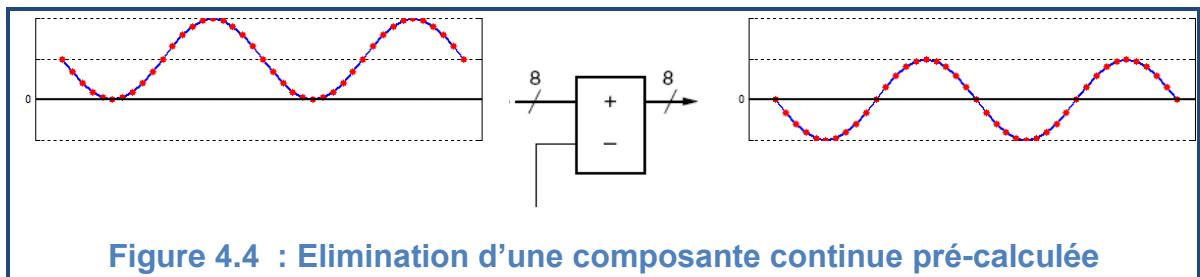
La figure 4.3 représente le schéma de l'aiguilleur des signaux d'entrée utilisé dans cette partie.



#### 4.5.1.2. Elimination de la composante continue

Le signal reçu par le FPGA est un signal à valeurs positives, on doit alors éliminer la composante continue pour retrouver l'alternance négative.

Puisque le niveau de la composante continue est connu, elle peut être éliminée en soustrayant sa valeur de chaque échantillon du signal numérique comme le montre la figure 4.3.



#### 4.5.1.3. Mise à l'échelle

Les signaux reçus à partir du CAN sont des signaux numériques de  $2^N$  *niveaux*. Tous les échantillons du signal numérique doivent être alors multipliés par un certain gain (constante) pour retrouver les valeurs réelles du signal mesuré.

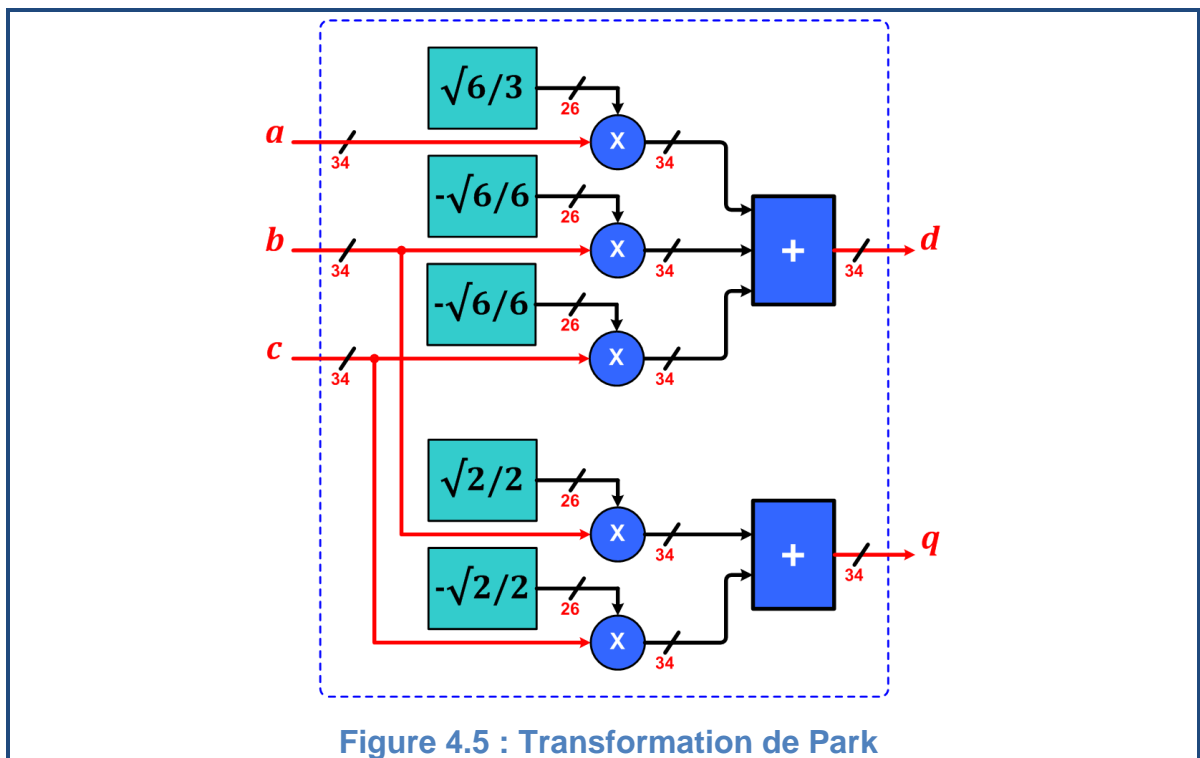
$$x_{reel} = x * \frac{x_{max}}{2^N} \quad (3.1)$$

#### 4.5.2. Transformation de Park

La matrice de transformation de Park donnée par l'équation (1. 15) se réécrit sous la forme suivante en prenant  $\xi = 0$  :

$$T_{32} = \begin{bmatrix} \frac{\sqrt{6}}{3} & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{6} \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \quad (3.2)$$

Les éléments de cette matrice doivent être déclarés comme constantes et la structure logique de la transformation se résume dans La figure 4.4.



### 4.5.3.calcul de la matrice d'état

L'équation (2.35) montre que certains coefficients de la matrice d'état discrétisé sont fonction de la vitesse mécanique. Ils doivent être calculés à chaque impulsion d'horloge. Les coefficients concernés sont :  $ad_3, ad_4, ad_6, ad_7$  et  $ad_8$

En simplifiant leurs équations on trouve l'équation (3.3) :

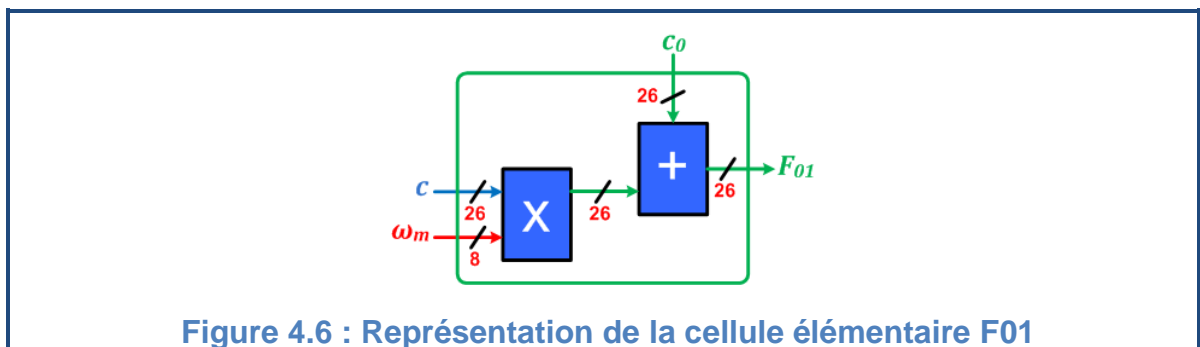
$$\begin{cases} ad_3 = c_{03} + c_3\omega_m \\ ad_4 = c_{04} - c_4\omega_m \\ ad_6 = c_6\omega_m \\ ad_7 = c_{07} - c_7\omega_m \\ ad_8 = c_{08} - c_8\omega_m \end{cases} \quad (3.3)$$

Avec :

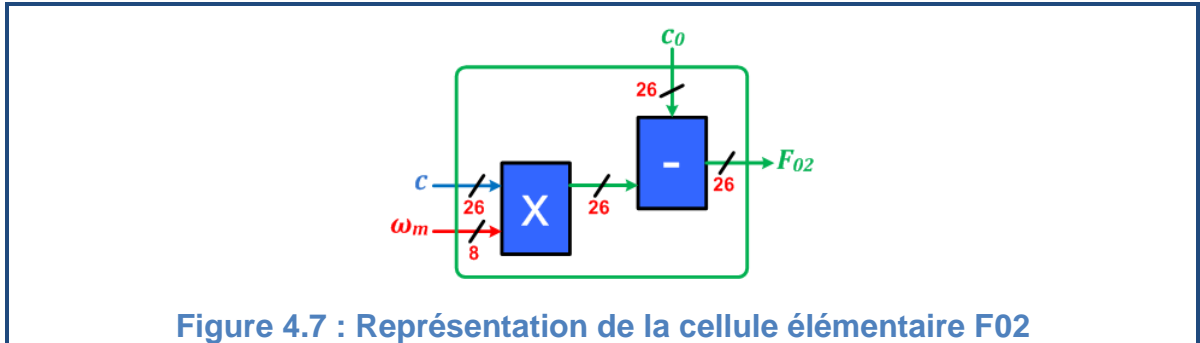
$$\begin{cases} c_{03} = a_3T \left(1 + (a_1 + a_7)\frac{T}{2}\right) & , & c_3 = a_4\frac{T^2}{2} \\ c_{04} = a_4T \left(1 + (a_1 + a_7)\frac{T}{2}\right) & , & c_4 = a_3\frac{T^2}{2} \\ c_6 = -a_5\frac{T^2}{2} \\ c_{07} = 1 + a_7T + (a_7^2 + a_3a_5)\frac{T^2}{2} & , & c_7 = a_8\frac{T^2}{2} \\ c_{08} = -a_8T + a_5a_7\frac{T^2}{2} & , & c_8 = 2a_7\frac{T^2}{2} \end{cases} \quad (3.4)$$

On remarque que le système d'équation (3.3) est basé sur deux fonctions élémentaires :

- La première fait une addition d'un produit de deux entrées (une constante  $c$  de format  $s[26|24]$  et une variable  $\omega_m$  de format  $s[34|24]$ ) avec une troisième entrée  $c_0$  (constante de format  $s[26|24]$ ). Cette fonction est représentée dans la figure 4.6.

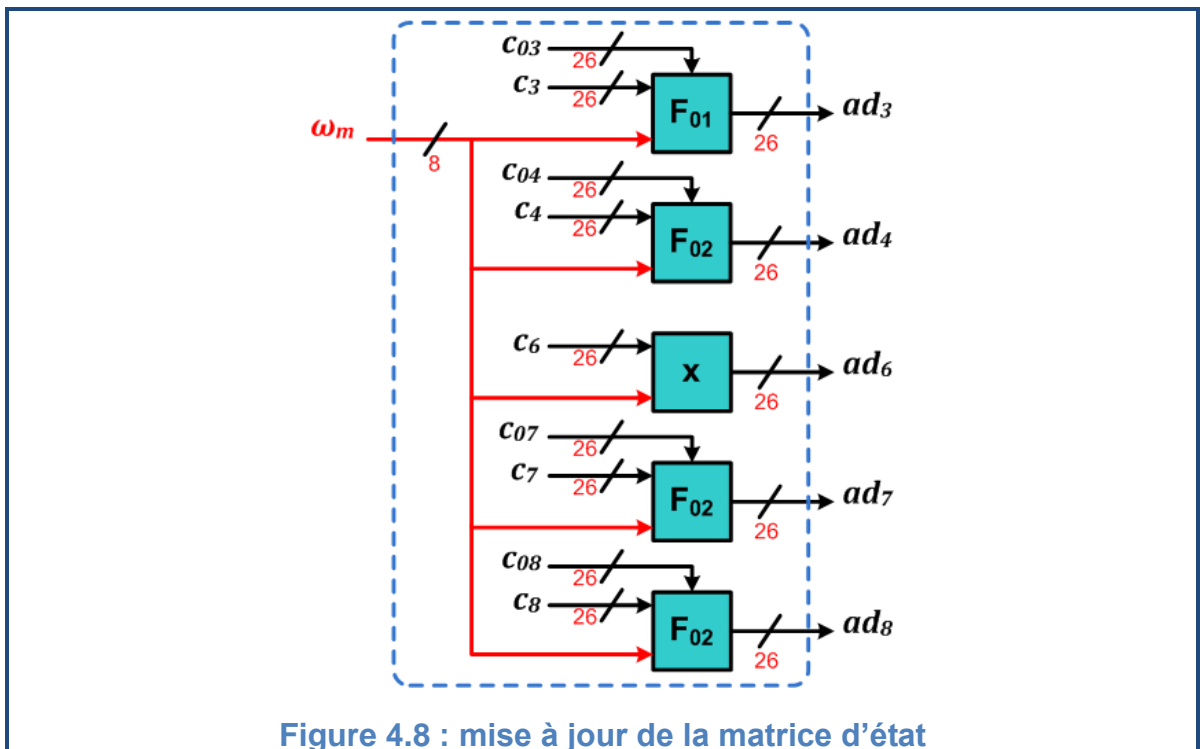


- La deuxième fait soustraire le produit de la constante  $c$  avec la variable  $\omega_m$  de la troisième entrée  $c_0$ . Elle est représentée dans la figure 4.7.



Dans les deux cas, le résultat de multiplication des deux entrées  $c$  (de format  $s[26|24]$ ) et  $\omega_m$  (de format  $s[34|24]$ ) est de format  $s[60|48]$ . Le résultat est toujours inférieur à un, donc il peut être représenté par le format  $s[26|24]$  en utilisant la troncature représentée dans le paragraphe (4.3.1)

La structure logique proposée pour l'équation (3.3) est représentée dans la figure 4.8.



#### 4.5.4. Filtrage de Kalman

Le filtrage de Kalman se fait en trois sous étapes :

**La première sous étape** fait fonctionner trois blocs indépendants en parallèle :

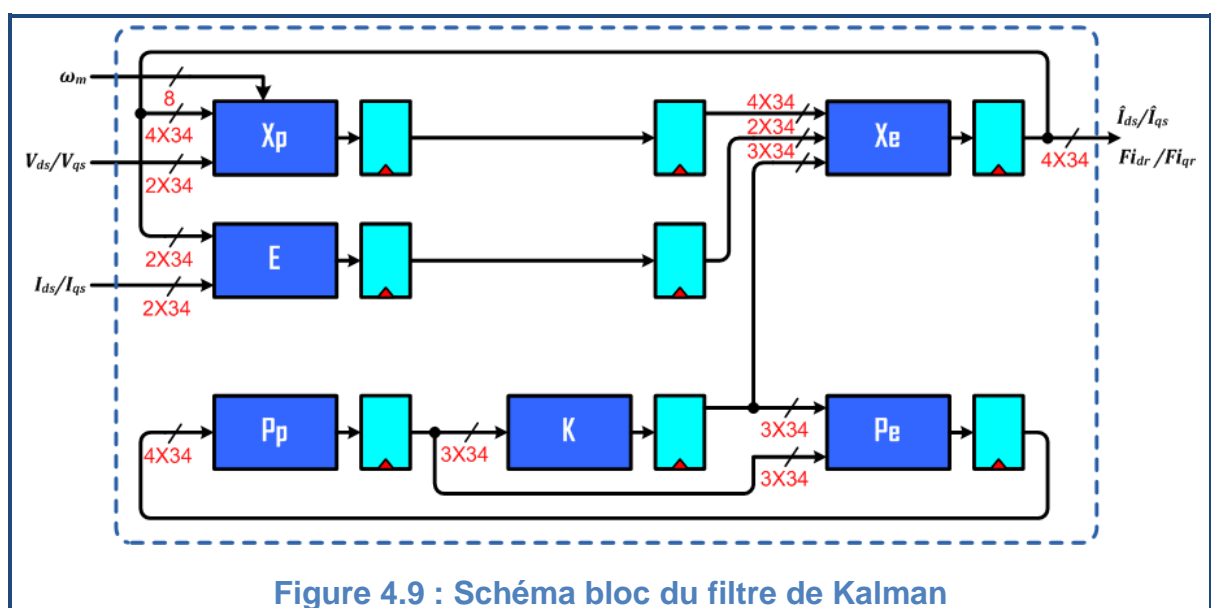
- Le premier nommé ( $X_p$ ) et fait la prédiction des états du système.
- Le deuxième nommé ( $E$ ) et calcule les erreurs entre les états estimés et les états mesurés.
- Le troisième nommé ( $P_p$ ) et calcule la matrice de covariance de l'erreur de prédiction.

**La deuxième sous étape** fait fonctionner le bloc nommé ( $K$ ) qui calcule la matrice des gains de Kalman.

**La troisième sous étape** fait fonctionner deux blocs indépendants :

- Le premier nommé ( $X_e$ ) met à jours le vecteur des états estimés.
- Le deuxième nommé ( $P_e$ ) met à jours la matrice de covariance de l'erreur.

La figure 4.9 représente le schéma bloc du filtre de Kalman.



#### 4.5.4.1. Prédiction

##### 4.5.4.1.1. Calcul des états prédits

On reprend l'équation d'état simplifiée du système représentée dans (2. 38)

$$\begin{cases} \hat{i}_{ds}[k+1|k] = ad_1 \hat{i}_{ds}[k|k] + ad_2 \hat{i}_{qs}[k|k] + ad_3 \hat{\varphi}_{dr}[k|k] + ad_4 \hat{\varphi}_{qr}[k|k] + bd_1 V_{ds} \\ \hat{i}_{qs}[k+1|k] = -ad_2 \hat{i}_{ds}[k|k] + ad_1 \hat{i}_{qs}[k|k] - ad_4 \hat{\varphi}_{dr}[k|k] + ad_3 \hat{\varphi}_{qr}[k|k] + bd_1 V_{qs} \\ \hat{\varphi}_{dr}[k+1|k] = ad_5 \hat{i}_{ds}[k|k] + ad_6 \hat{i}_{qs}[k|k] + ad_7 \hat{\varphi}_{dr}[k|k] + ad_8 \hat{\varphi}_{qr}[k|k] + bd_2 V_{ds} \\ \hat{\varphi}_{qr}[k+1|k] = -ad_6 \hat{i}_{ds}[k|k] + ad_5 \hat{i}_{qs}[k|k] - ad_8 \hat{\varphi}_{dr}[k|k] + ad_7 \hat{\varphi}_{qr}[k|k] + bd_2 V_{qs} \end{cases}$$

Ce système d'équation est basé sur deux autres fonctions élémentaires  $F_1$  et  $F_2$  qui ont exactement les même structure de  $F_{01}$  et  $F_{02}$  respectivement, sauf que leurs sortie sont de format  $s[34|24]$ :

- La première fait une addition du produit de deux entrées (une constante  $ad$  de format  $s[26|24]$  et une variable  $x$  de format  $s[34|24]$ ) avec le résultat d'une opération précédente (entrée  $x_0$  de format  $s[34|24]$ ). Cette fonction est représentée dans la figure 4.10.

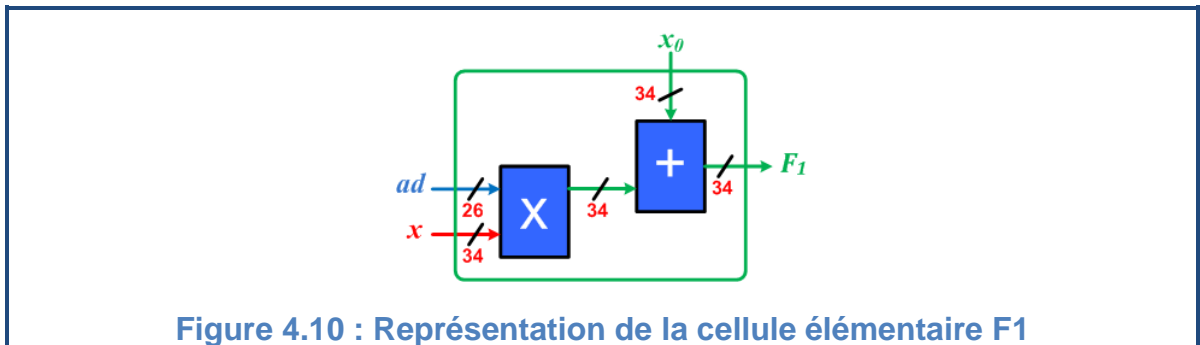


Figure 4.10 : Représentation de la cellule élémentaire F1

- La deuxième fait soustraire le produit (des deux entrées  $ad$  et  $x$ ) du résultat de l'opération précédente (l'entrée  $x_0$ ). Elle est représentée dans la figure 4.11.

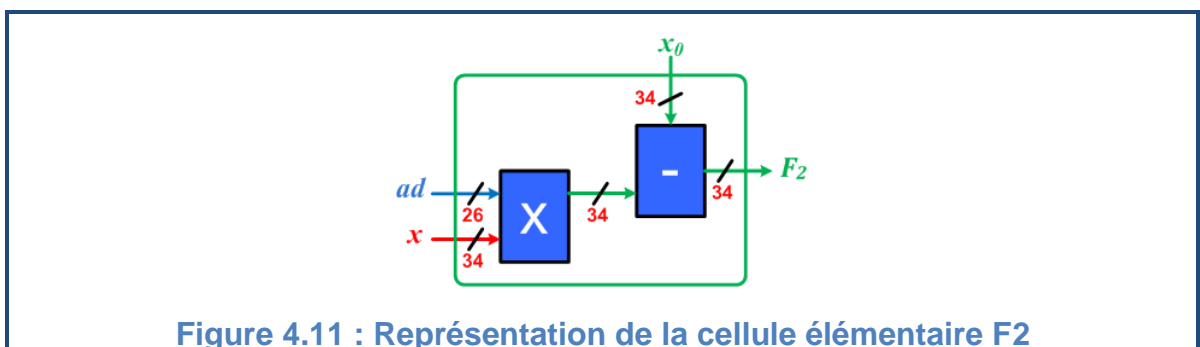


Figure 4.11 : Représentation de la cellule élémentaire F2

Dans ces deux cas, le résultat de multiplication des deux entrées  $ad$  (de format  $s[26|24]$ ) et  $x$  (de format  $s[34|24]$ ) est de format  $s[60|48]$ . Il a été tronqué de la manière représentée dans le paragraphe (4.3.2).

La structure de la première équation est représentée dans la figure 4.12.

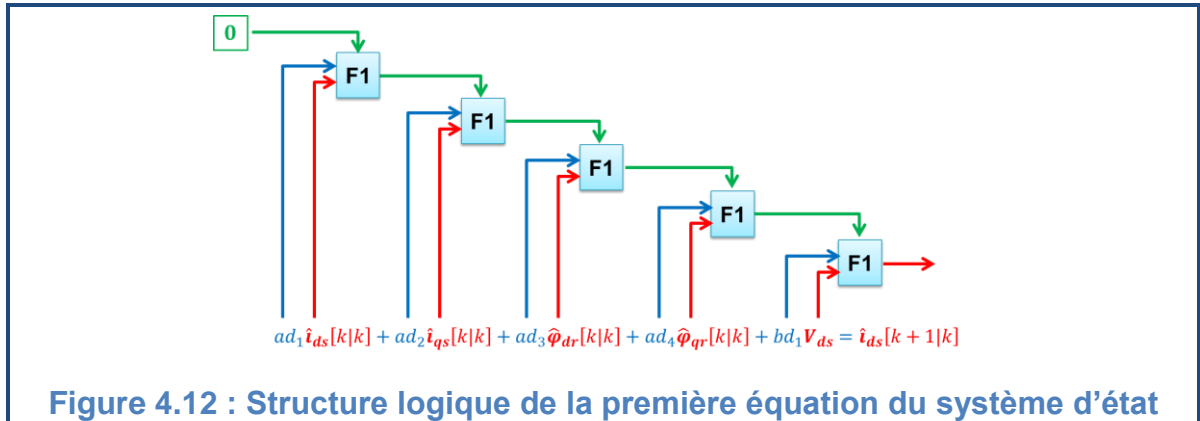


Figure 4.12 : Structure logique de la première équation du système d'état

La structure de la deuxième équation est représentée dans la figure 4.13.

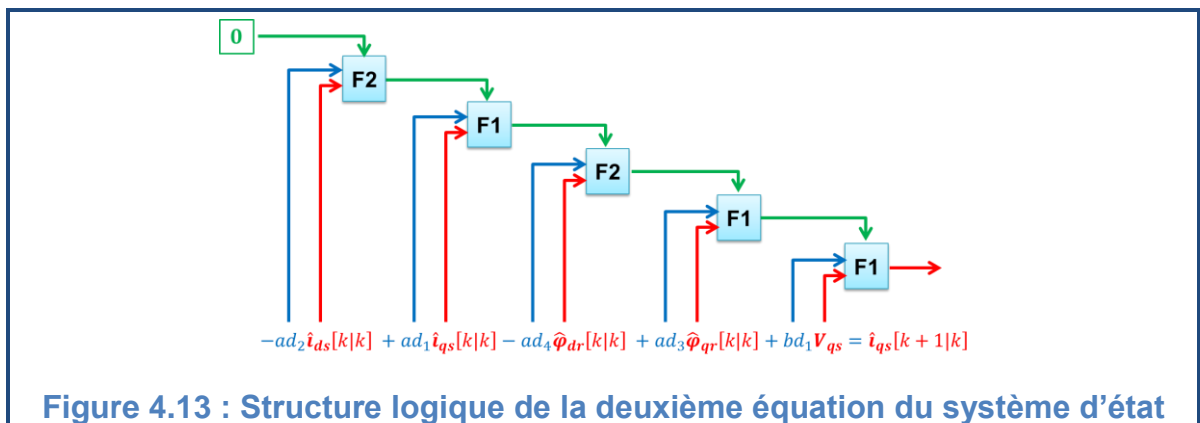


Figure 4.13 : Structure logique de la deuxième équation du système d'état

La structure de la troisième équation est représentée dans la figure 4.14.

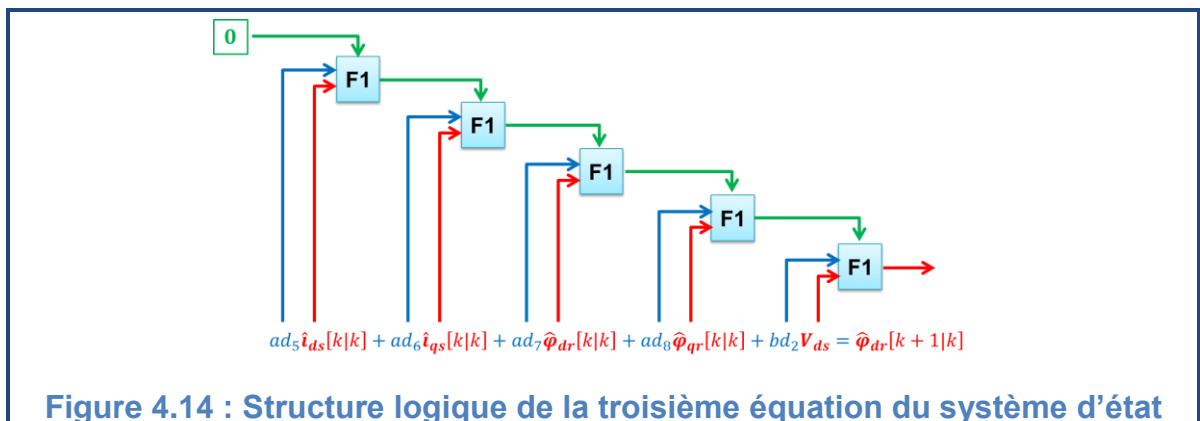
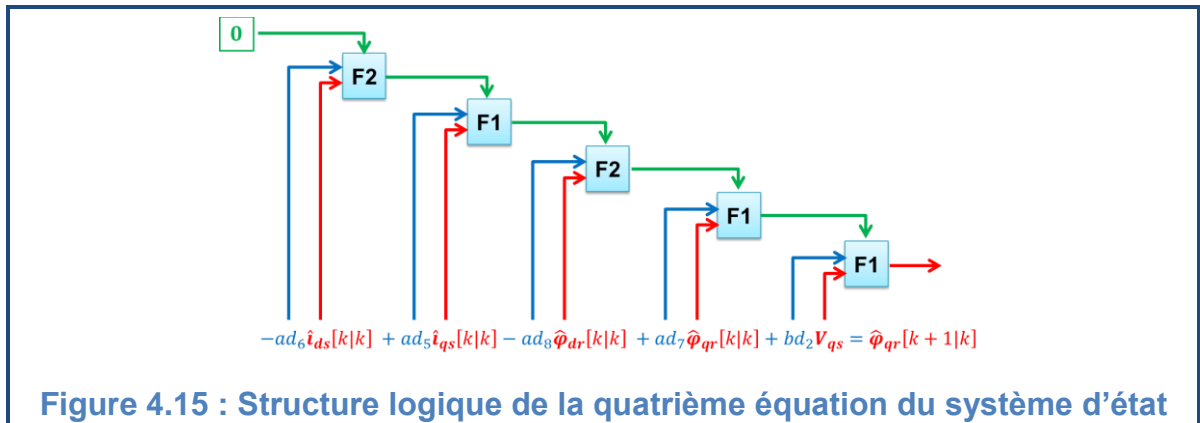


Figure 4.14 : Structure logique de la troisième équation du système d'état

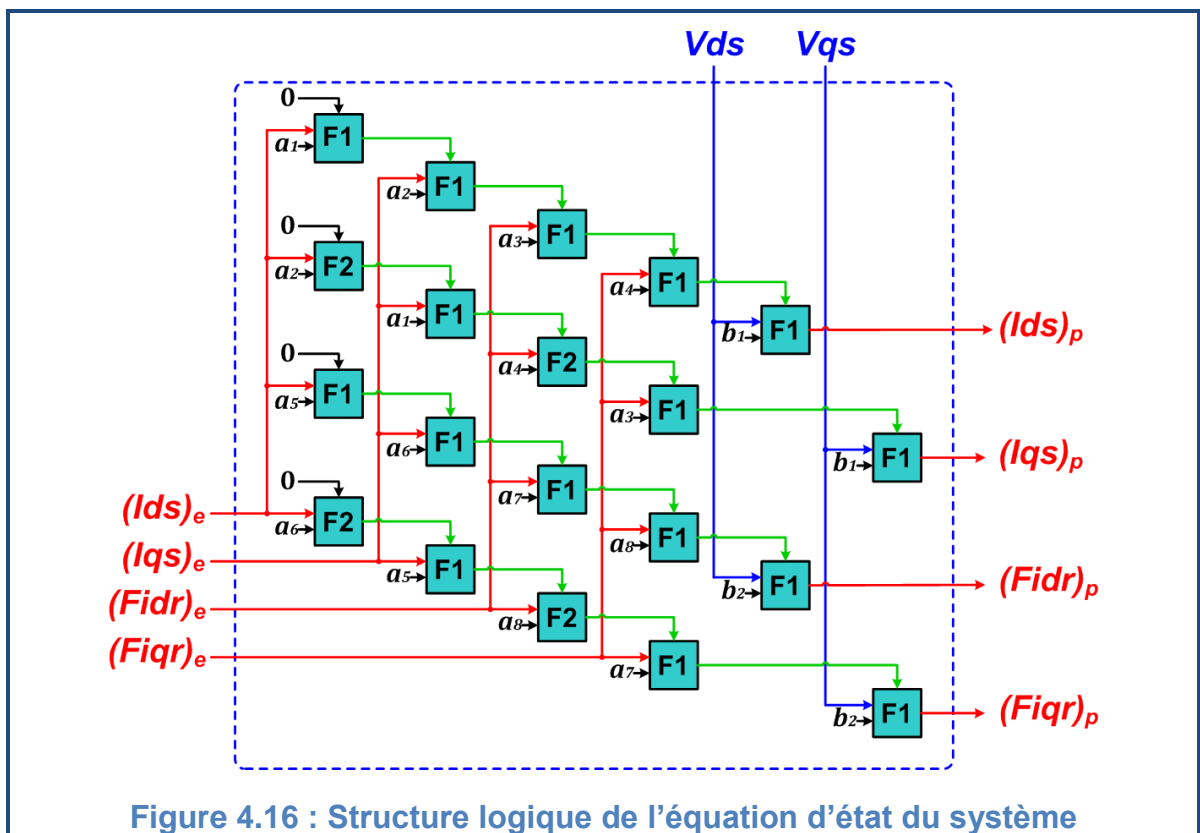


La structure de la quatrième équation est représentée dans la figure 4.15.



A la fin de chaque équation, un registre synchrone de 34 bits à chargement parallèle a été utilisé pour maintenir la valeur du signal stable jusqu'à la prochaine impulsion d'horloge.

La structure complète du bloc de prédiction de l'état nommé ( $X_p$ ) est représentée dans la figure 4.16.



Avec les  $a_i$  sont les éléments de la matrice  $A_d$ .

#### 4.5.4.1.2. Calcul de la matrice de covariance de l'erreur de prédiction

Pour le calcul de la matrice de covariance de l'erreur de prédiction, l'équation (2. 40) est utilisée.

On reprend donc l'équation (2.40) :

$$\left\{ \begin{array}{l}
 P_1[k + 1|k] = (ad_1^2 + ad_2^2)P_1[k|k] \\
 \quad + 2(ad_1ad_3 + ad_2ad_4)P_2[k|k] \\
 \quad + 2(ad_1ad_4 - ad_2ad_3)P_3[k|k] \\
 \quad + (ad_3^2 + ad_4^2)P_4[k|k] + q_1 \\
 \\
 P_2[k + 1|k] = (ad_1ad_5 + ad_2ad_6)P_1[k|k] \\
 \quad + (ad_3ad_5 + ad_4ad_6 + ad_1ad_7 + ad_2ad_8)P_2[k|k] \\
 \quad + (ad_4ad_5 - ad_3ad_6 + ad_1ad_8 - ad_2ad_7)P_3[k|k] \\
 \quad + (ad_3ad_7 + ad_4ad_8)P_4[k|k] \\
 \\
 P_3[k + 1|k] = (ad_2ad_5 - ad_1ad_6)P_1[k|k] \\
 \quad + (ad_4ad_5 - ad_3ad_6 + ad_2ad_7 - ad_1ad_8)P_2[k|k] \\
 \quad + (ad_1ad_7 - ad_3ad_5 + ad_2ad_8 - ad_4ad_6)P_3[k|k] \\
 \quad + (ad_4ad_7 - ad_3ad_8)P_4[k|k] \\
 \\
 P_4[k + 1|k] = (ad_5^2 + ad_6^2)P_1[k|k] \\
 \quad + 2(ad_5ad_7 + ad_6ad_8)P_2[k|k] \\
 \quad + 2(ad_5ad_8 - ad_6ad_7)P_3[k|k] \\
 \quad + (ad_7^2 + ad_8^2)P_4[k|k] + q_2
 \end{array} \right.$$

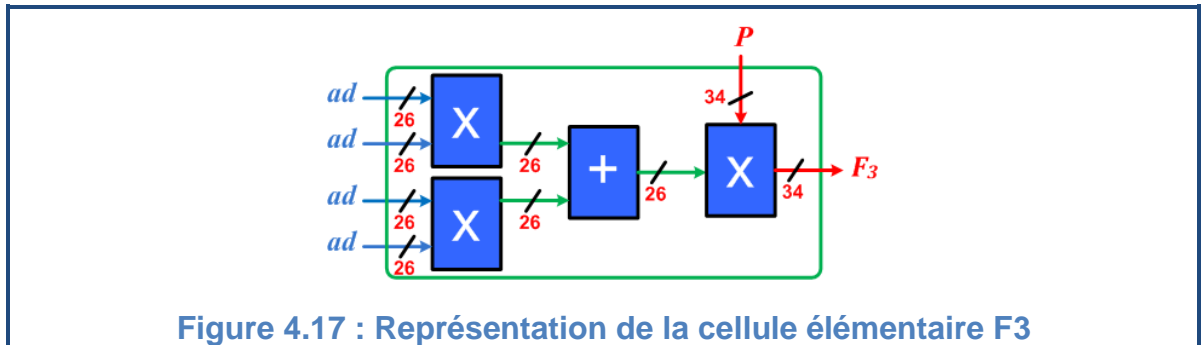
Avec  $q_1$  et  $q_2$  sont des constantes de format  $s[34|24]$ .

Cette équation est basée sur les trois opérateurs arithmétiques de base : l'addition, la soustraction et la multiplication.

L'implémentation de ce système d'équation est basée donc sur deux autres fonctions élémentaires :

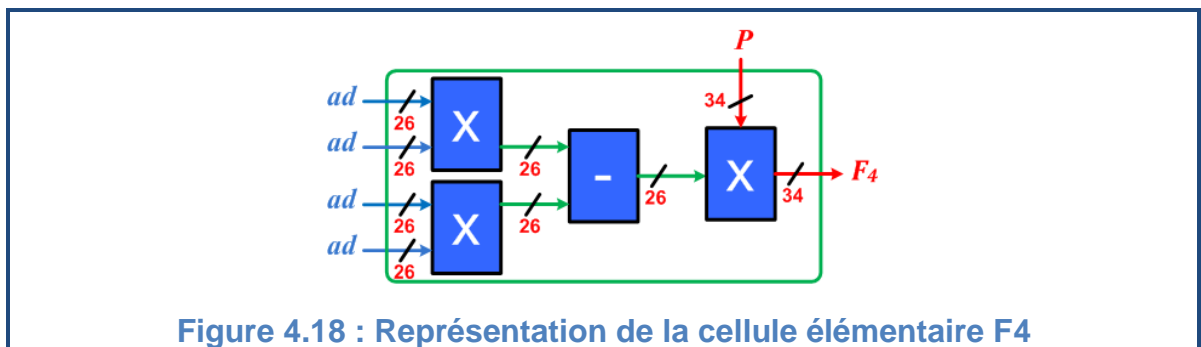
- La première fait une multiplication de quatre coefficients de format  $s[26|24]$  deux à deux. La somme des deux résultats est multipliée par une variable de format  $s[34|24]$ .

Cette fonction est représentée dans la figure 4.17.



- La deuxième multiplie le résultat de soustraction des deux produits avec la cinquième entrée.

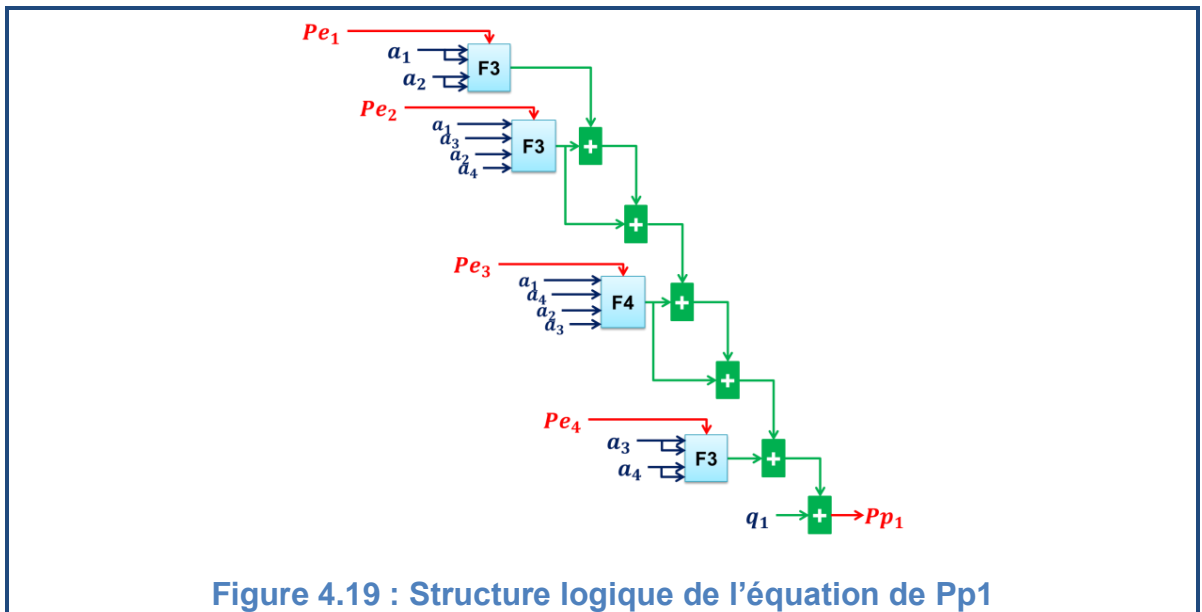
Cette fonction est représentée dans la figure 4.18.



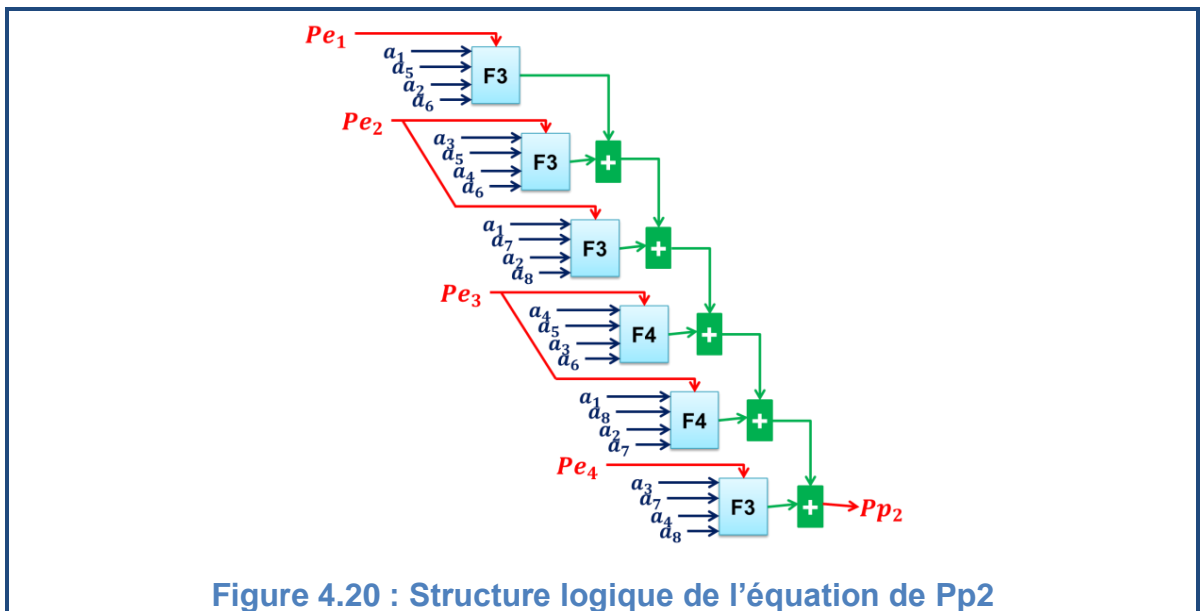
Le résultat de multiplication de deux entrées qui sont de format  $s[26|24]$  et ont des valeurs réelles inférieures à un est de format  $s[52|48]$  avec un seul bit significatif dans la partie entière plus un bit de signe et deux bits d'extension de signe. La troncature proposée dans le paragraphe (4.3.1) permet de représenter le résultat sous le même format que celui des entrées.

La multiplication du résultat de soustraction (de format  $s[26|24]$ ) avec l'entrée  $P$  (de format  $s[34|24]$ ) donne un résultat de format  $s[60|48]$  qui peut être tronqué de la façon représentée dans le paragraphe (4.3.2).

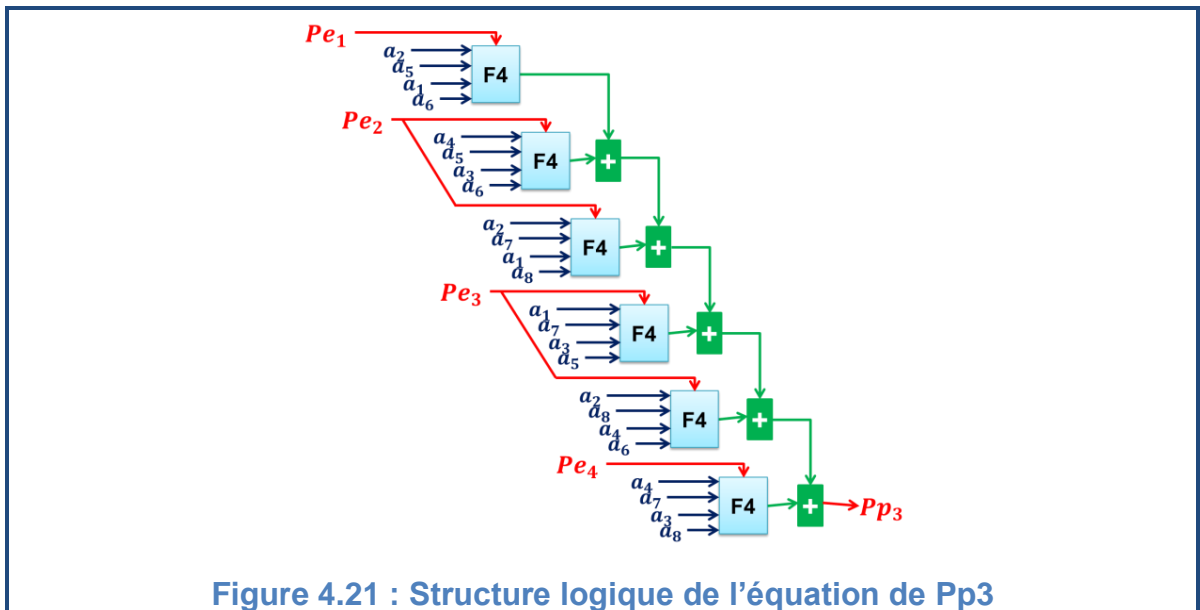
La structure logique de l'équation de prédiction du premier élément de la matrice de covariance de l'erreur de prédiction est représentée dans la figure 4.19.



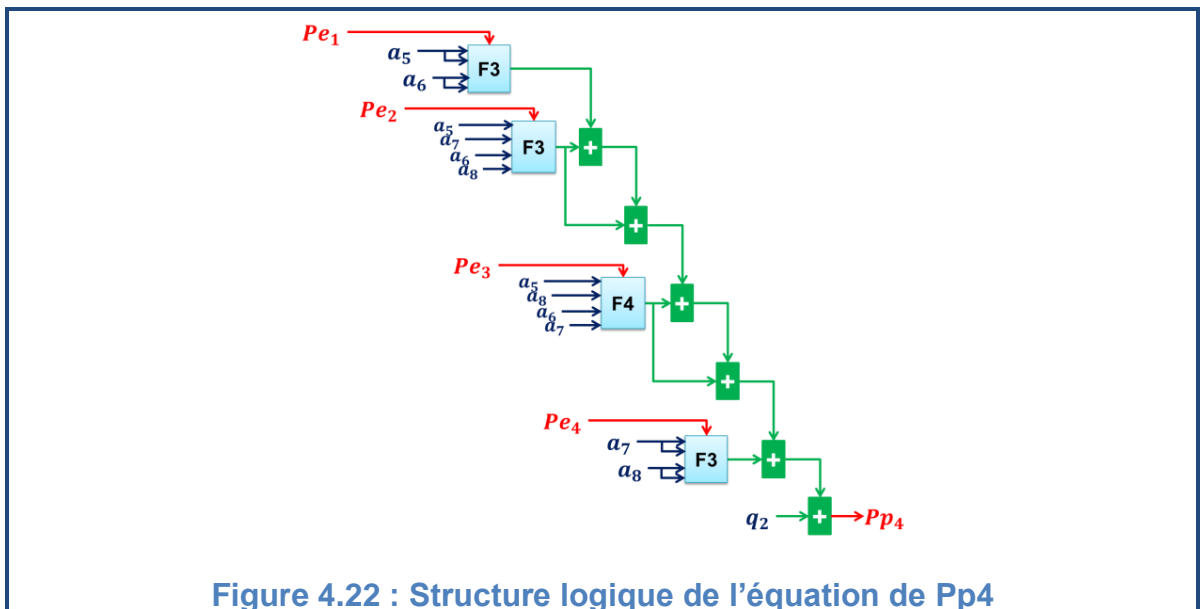
La structure logique de l'équation de prédiction du deuxième élément de la matrice de covariance de l'erreur de prédiction est représentée dans la figure 4.20.



La structure logique de l'équation de prédiction du troisième élément de la matrice de covariance de l'erreur de prédiction est représentée dans la figure 4.21.

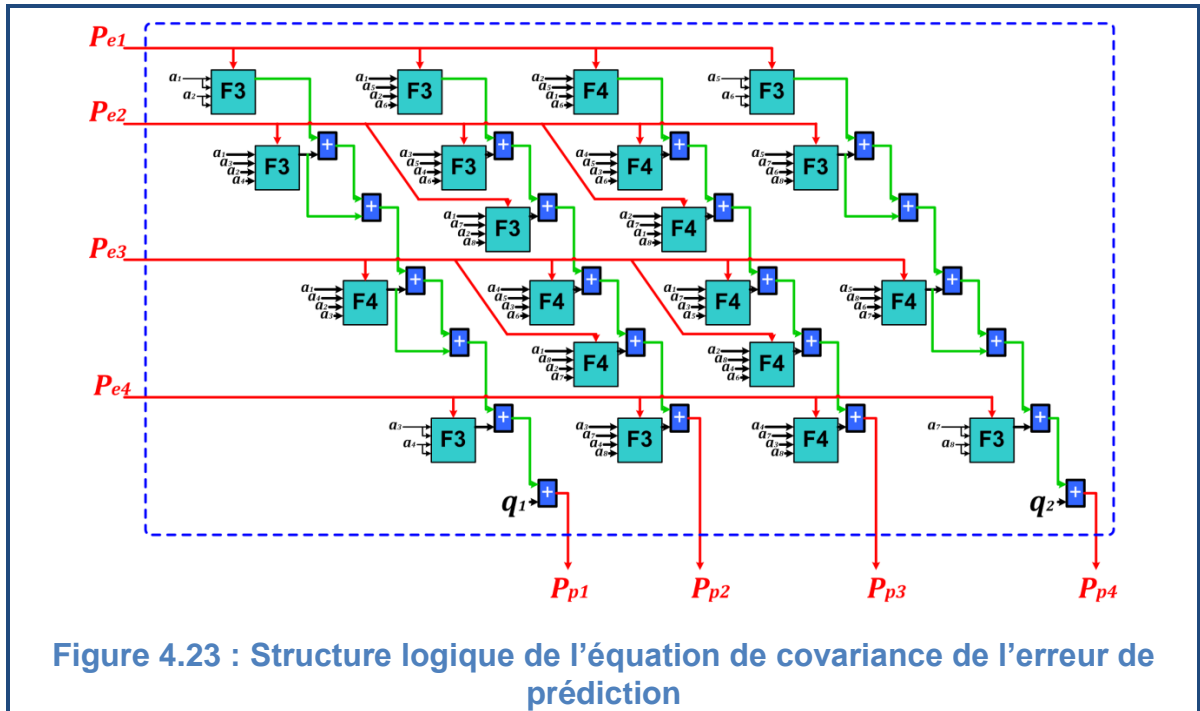


La structure logique de l'équation de prédiction du quatrième élément de la matrice de covariance de l'erreur de prédiction est représentée dans la figure 4.22.



Dans ce bloc aussi on a utilisé des registres de 34 bits à chargement parallèle pour maintenir les sorties stables entre deux impulsions d'horloge.

La structure complète du bloc de prédiction de la matrice de l'erreur nommé ( $P_p$ ) est représentée dans la figure 4.23.



Avec :

- Les  $P_e$  sont les éléments de la matrice de covariance de l'erreur d'estimation de l'itération précédente.
- Les  $P_p$  sont les éléments de la matrice de covariance de l'erreur de prédiction.
- $q_1, q_2$  sont les éléments de la matrice de covariance du bruit d'état  $Q$ .

#### 4.5.4.1.3. Calcul des erreurs

Le vecteur de l'erreur contient les deux composantes données par l'équation (2.43):

- La première est entre la composante  $d$  du courant mesuré et celle du courant prédit.
- La deuxième est entre la composante  $q$  du courant mesuré et celle du courant prédit.

L'implémentation de ces équations se fait par des soustractions de nombres signés de format  $s[34|24]$  et les résultats seront du même format.

La figure 4.24 représente la structure logique des deux équations qui font le calcul du vecteur de l'erreur.

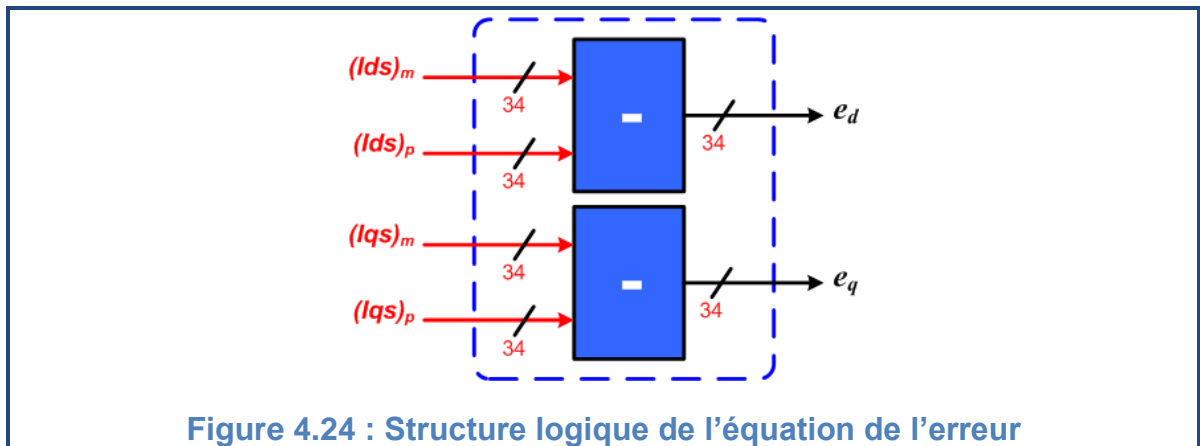


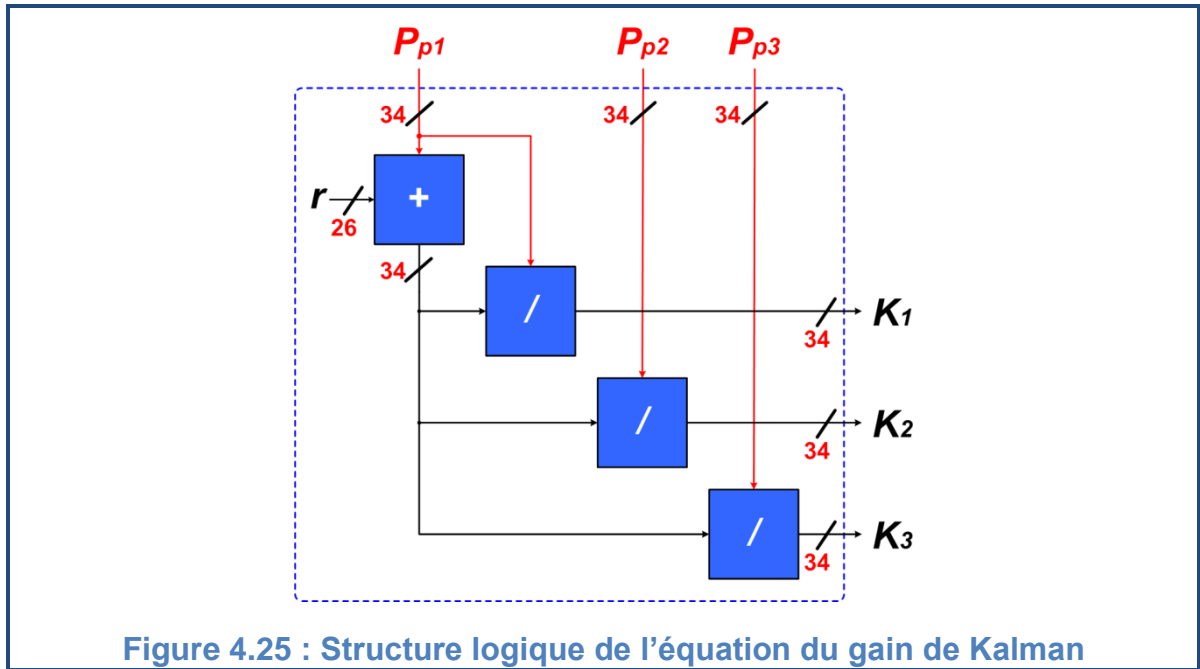
Figure 4.24 : Structure logique de l'équation de l'erreur

#### 4.5.4.2. Calcul du gain de Kalman

On reprend l'équation (2.42) pour calculer les éléments de la matrice de gain de Kalman  $K$ :

$$\begin{cases} K_1[k+1] = P_1[k+1|k]/(P_1[k+1|k] + r) \\ K_2[k+1] = P_2[k+1|k]/(P_1[k+1|k] + r) \\ K_3[k+1] = P_3[k+1|k]/(P_1[k+1|k] + r) \end{cases}$$

Cette étape est basée sur l'opérateur de division mais les trois équations ont le même dénominateur qui se calcule par une sommation de la variable  $P_1$  (de format  $s[34|24]$ ) avec la variance-covariance de la mesure du courant  $r$  (de format  $s[34|24]$ ) et le résultat est du même format  $s[34|24]$ . Les différents algorithmes de division sont coûteux de point de vue hardware. Et c'est pour cette raison qu'on a évité cette opération en approximant le dénominateur à  $2^N$  et la division devient donc un décalage. Les différents signaux de sortie du bloc  $K$  ont le format  $s[34|24]$ . Et la figure 4.25 représente la structure logique de ce bloc.



#### 4.5.4.3. Estimation

##### 4.5.4.3.1. Calcul des états estimés

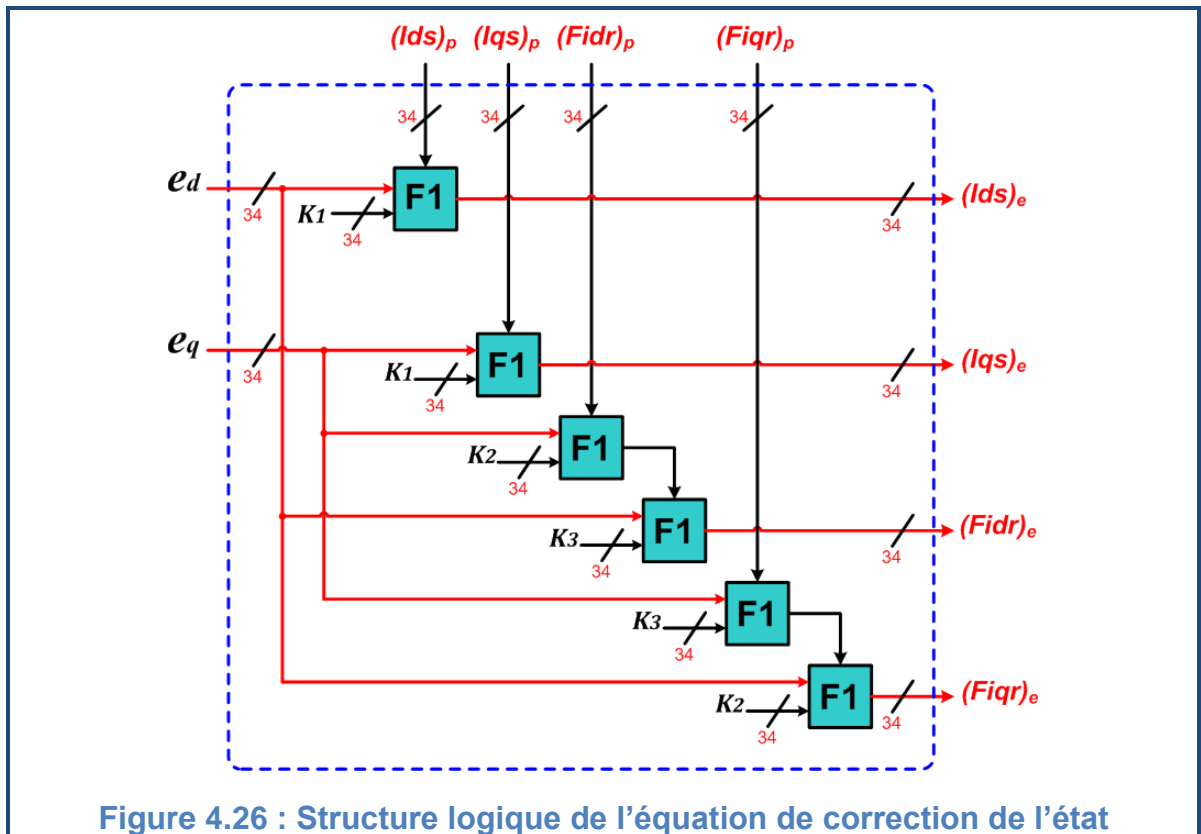
L'étape d'estimation utilise les éléments de la matrice de gain pour corriger les états prédits en prenant en compte les deux composantes de l'erreur comme le présente l'équation (2.44).

$$\begin{cases} \hat{i}_{ds}[k+1|k+1] = \hat{i}_{ds}[k+1|k] + K_1[k+1]\varepsilon_d[k+1] \\ \hat{i}_{qs}[k+1|k+1] = \hat{i}_{qs}[k+1|k] + K_1[k+1]\varepsilon_q[k+1] \\ \hat{\phi}_{dr}[k+1|k+1] = \hat{\phi}_{dr}[k+1|k] + K_2[k+1]\varepsilon_d[k+1] + K_3[k+1]\varepsilon_q[k+1] \\ \hat{\phi}_{qr}[k+1|k+1] = \hat{\phi}_{qr}[k+1|k] + K_3[k+1]\varepsilon_d[k+1] + K_2[k+1]\varepsilon_q[k+1] \end{cases}$$

Ces quatre équations sont basées sur le principe de la multiplication addition, on utilise donc la fonction  $F_1$  représenté auparavant.

La structure logique de ce bloc est représentée dans la figure 4.26.





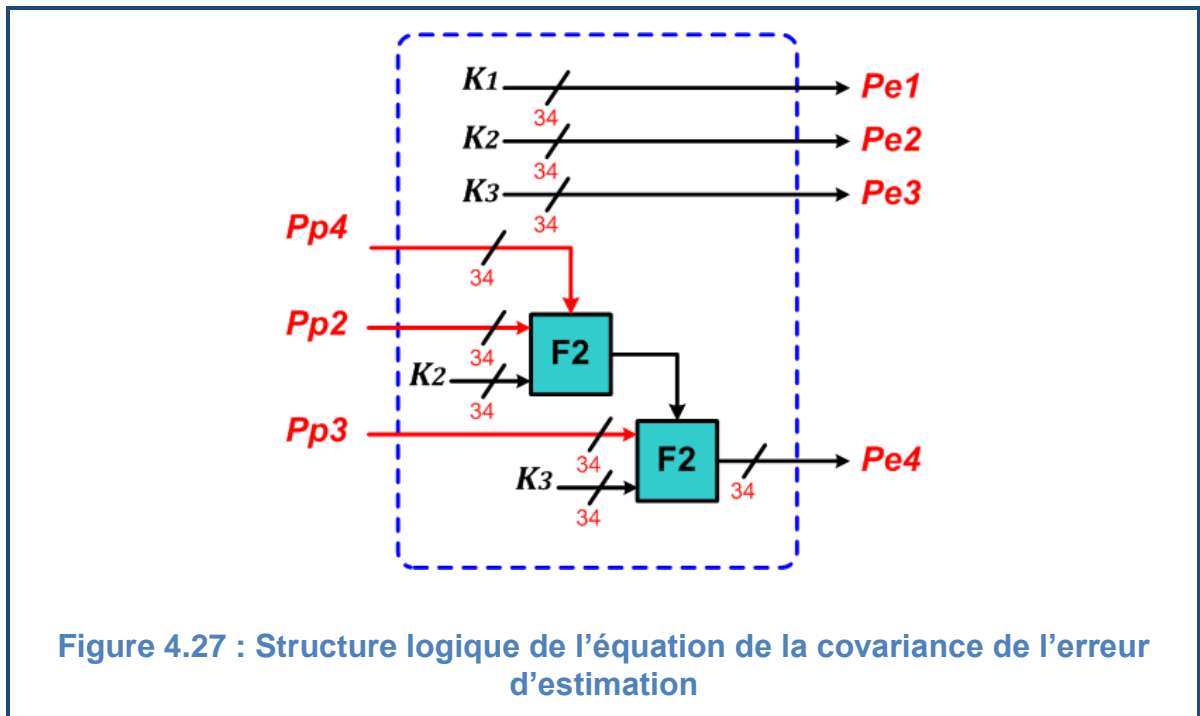
#### 4.5.4.3.2. Calcul de la matrice de covariance de l'erreur d'estimation

La dernière étape consiste à déterminer la quatrième composante de la matrice de covariance de l'erreur d'estimation, les autres composantes sont les mêmes avec les trois premières composantes de la matrice de gain.

On reprend l'équation (2.46) :

$$\begin{cases} P_1[k+1|k+1] = K_1[k+1] \\ P_2[k+1|k+1] = K_2[k+1] \\ P_3[k+1|k+1] = K_3[k+1] \\ P_4[k+1|k+1] = P_4[k+1|k] - P_2[k+1|k]K_2[k+1] - P_3[k+1|k]K_3[k+1] \end{cases}$$

La structure logique du bloc qui calcule la matrice de covariance de l'erreur d'estimation est représentée dans la figure 4.27.



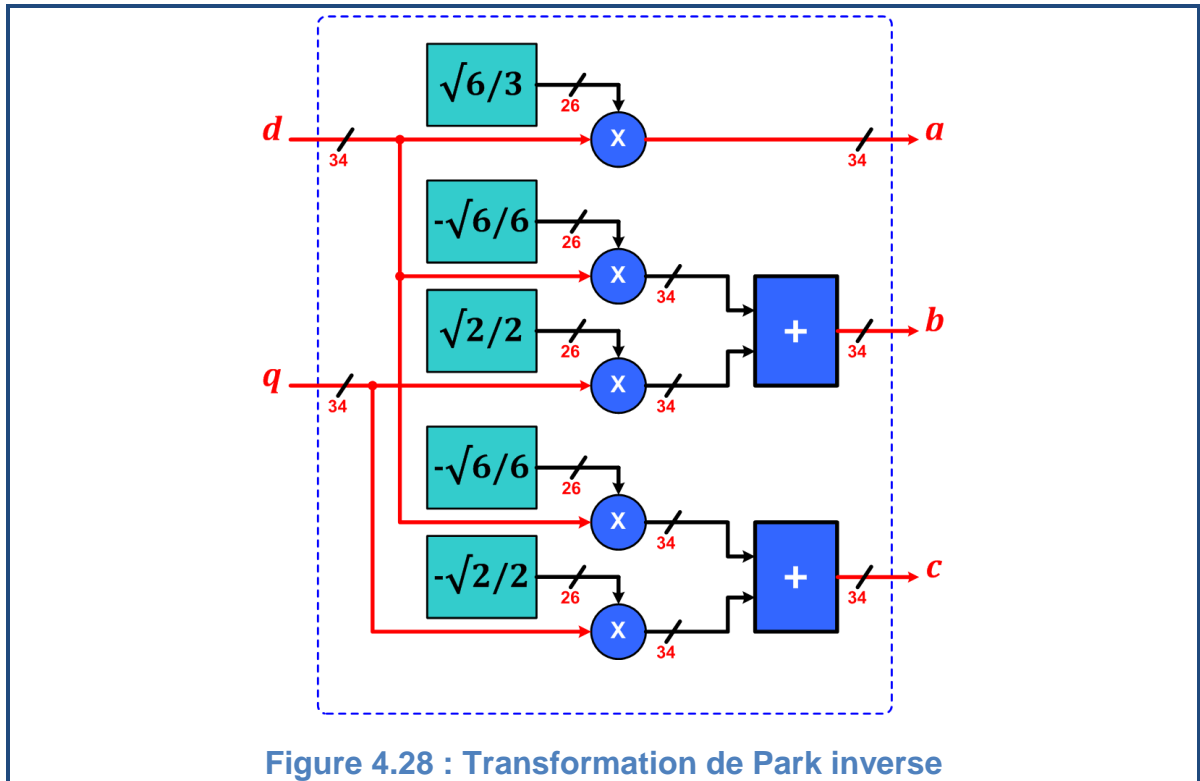
#### 4.5.5. Transformation de Park inverse

La transformation de Park inverse est utilisée pour retrouver les différentes composantes des courants et des flux estimés dans le repère triphasé. Cette opération est basée sur la matrice représentée par l'équation (1.11).

En prenant  $\xi = 0$  et en négligeant le régime homopolaire, cette matrice peut être représentée par l'équation (3.5):

$$T_{23} = \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 \\ -\frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \end{bmatrix} \quad (3.5)$$

Les éléments de cette matrice sont déjà déclarés comme constantes dans la partie de la transformation (triphase-biphase) et la structure logique de cette fonction est représentée dans La figure 4.28.



#### 4.6. Présentation des résultats

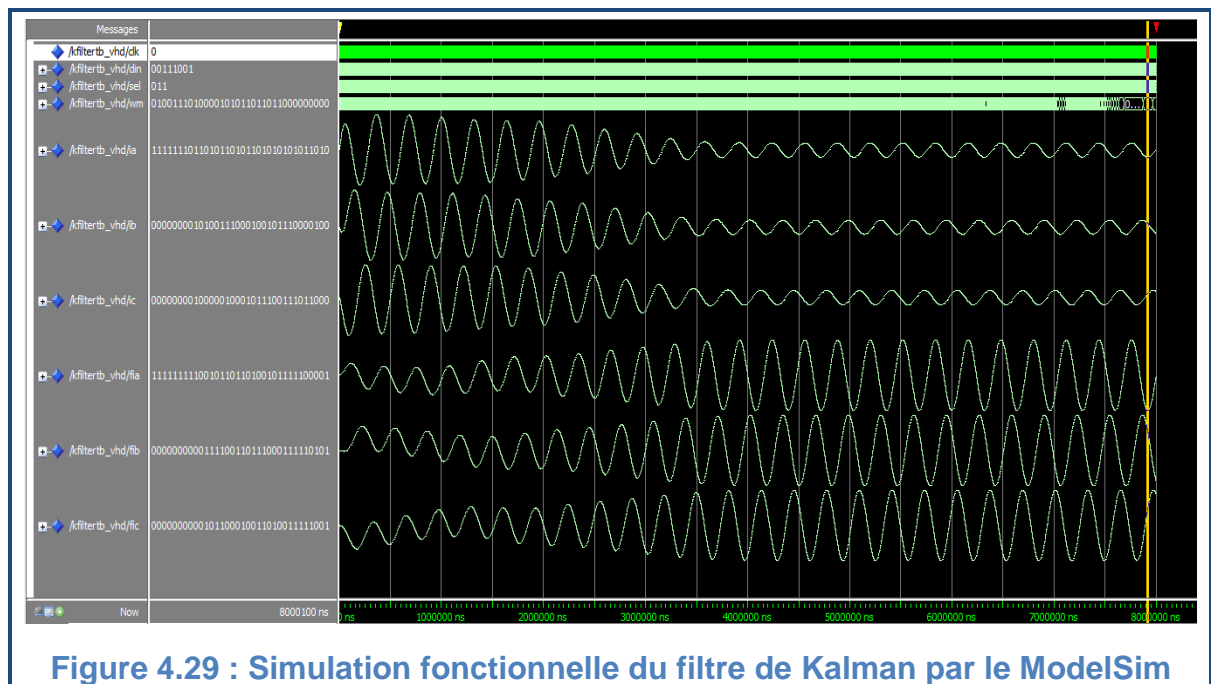
La validation des résultats peut être effectuée en plusieurs étapes :

- La simulation de l'architecture proposée avec un simulateur HDL (ModelSim) est utilisée pour vérifier la fonctionnalité du circuit en tenant en compte de toutes les contraintes temporelles.
- L'utilisation des outils du SystemGénérateur permet de comparer les résultats obtenus par l'architecture proposée avec ceux qui proviennent d'une simulation du même modèle avec Simulink afin de vérifier la validité des résultats obtenus.
- Les résultats de synthèse sont utilisés pour vérifier la synthétisabilité de l'architecture proposée, le pourcentage des ressources consommées ainsi que les temps de propagation de l'information dans le circuit et donc la fréquence maximale du fonctionnement.

#### 4.6.1. Résultats de Simulation

Les simulations comportementale, fonctionnelle et temporelle de la description ont été effectuées à l'aide du logiciel « **ModelSim Simulator** » de **MentorGraphics**.

La figure 4.29 présente les résultats obtenus après une simulation fonctionnelle de l'architecture proposée pour le filtre de Kalman.



**Figure 4.29 : Simulation fonctionnelle du filtre de Kalman par le ModelSim**

Les trois premières courbes représentent les courants des trois phases statoriques. Et les autres courbes représentent les trois flux rotoriques.

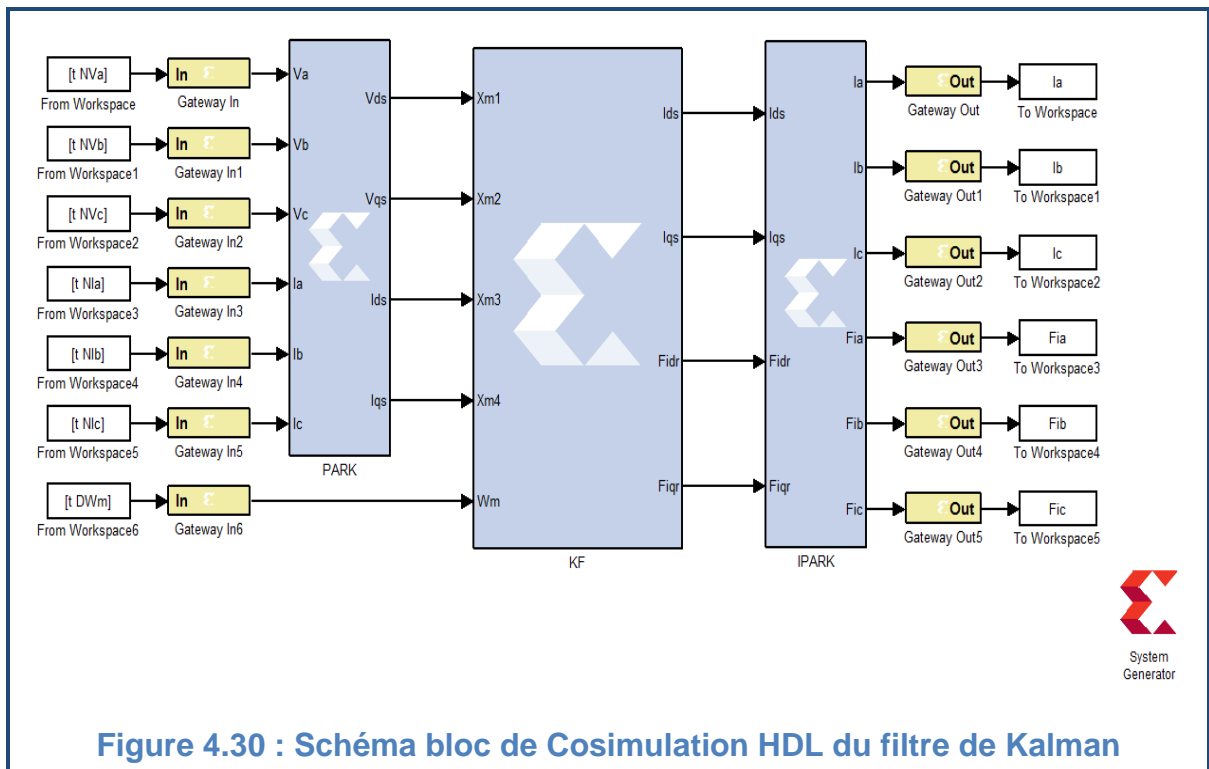
Les signaux obtenus par cette simulation ont des formes similaires aux signaux réels de la MAS. Cela vaut dire que les temps de propagation de l'information dans le circuit FPGA n'ont pas d'influence sur la fonctionnalité de l'architecture proposée pour le FK.

#### 4.6.2. Résultats de Cosimulation HDL

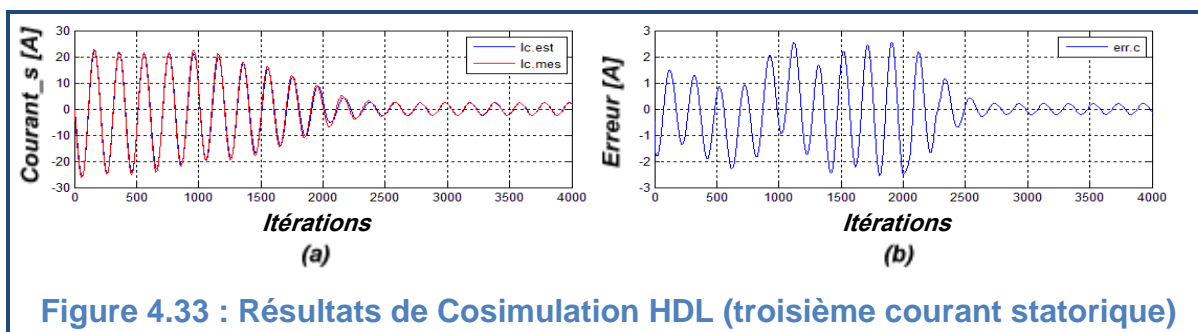
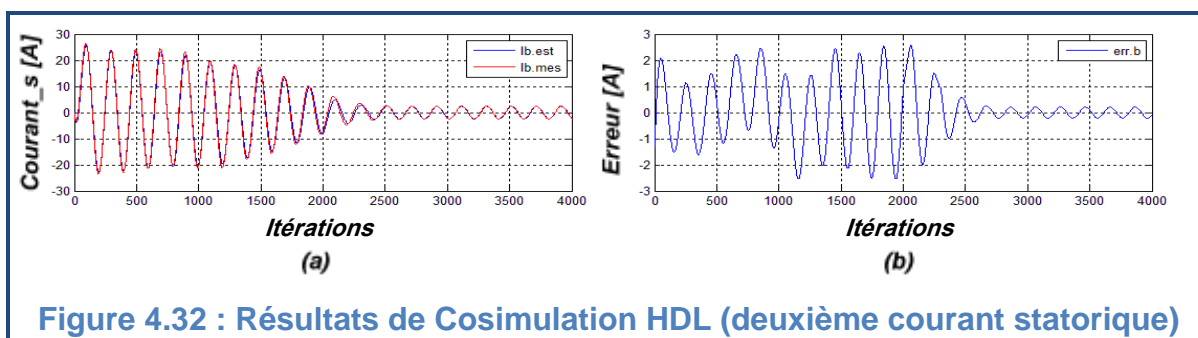
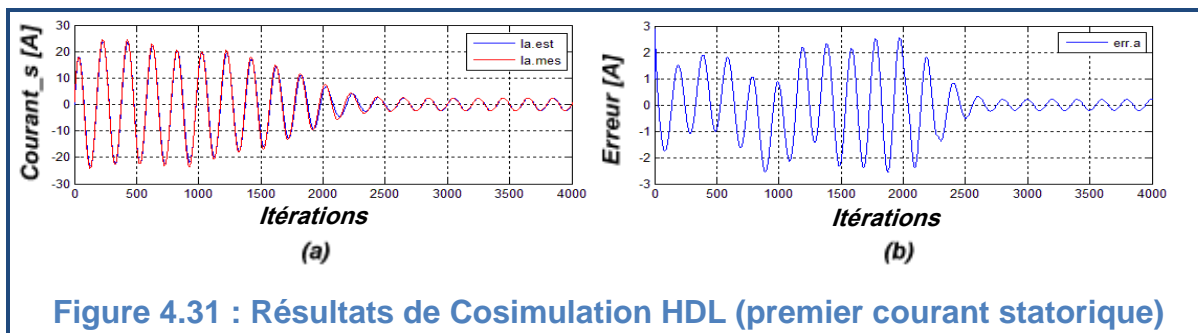
Avec la Cosimulation HDL, on peut implanter une description HDL d'un bloc hardware dans un environnement Matlab ou Simulink pour qu'on puisse lui appliquer des signaux d'entrée et lire ou visualiser ses signaux de sortie. Cette technique permet de comparer les résultats obtenus par ce bloc (en précision finie) avec ceux provenant d'une simulation Matlab ou Simulink (en précision infinie) pour vérifier la bonne fonctionnalité de la description.

L'utilisation des blocs du **SystemGénérateur** développé par Xilinx avec des blocs ordinaires du Simulink facilite l'établissement d'une relation simple entre le Simulateur HDL (**ModelSim** ou **ISE Simulator**) et le Simulink.

La figure 4.30 représente l'implantation de la description VHDL du filtre de Kalman dans un model Simulink en vue de sa Cosimulation.

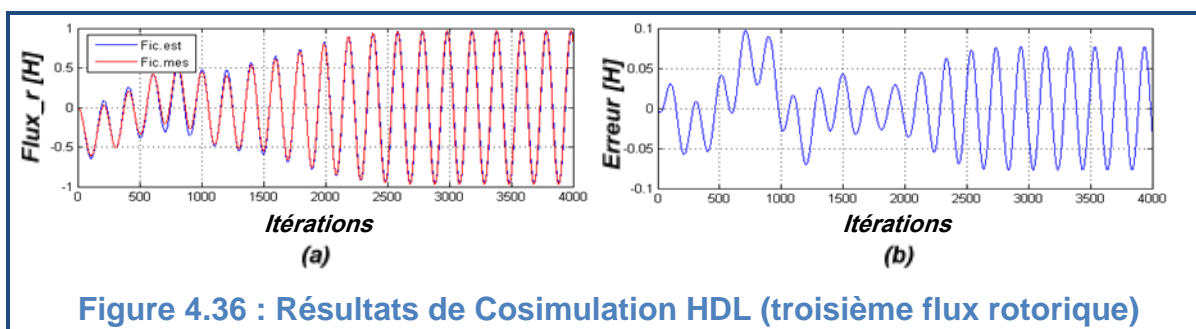
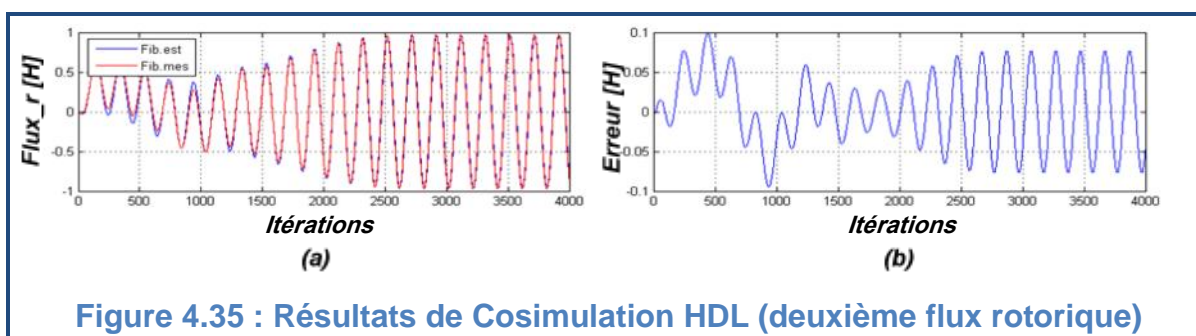
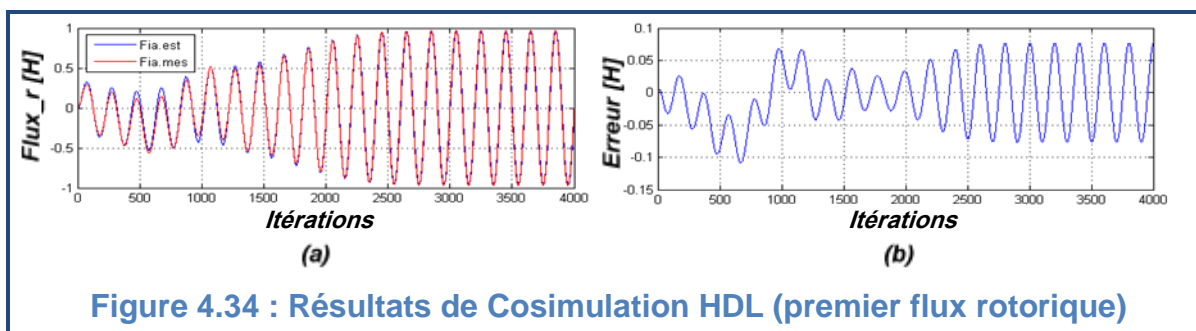


Les figures 4.31, 4.32, 4.33, 4.34, 4.35, 4.36 représentent les résultats obtenus par la Cosimulation.



Les figures 4.31(a), 4.32(a), 4.33(a) représentent la superposition des courants estimés et mesurés pour chacune des phases statoriques. Et les figures 4.31(b), 4.32(b), 4.33(b) représentent les erreurs entre les courants estimé et mesuré de la même phase. Ces erreurs tendent vers zéro après un régime transitoire.

L'architecture proposée pour le filtre de Kalman arrive donc à reconstituer les courants statoriques de façon satisfaisante.



Les figures 4.34(a), 4.35(a), 4.36(a) représentent la superposition des flux estimés et mesurés pour chacune des phases rotoriques. Et les figures 4.34(b), 4.35(b), 4.36(b) représentent les erreurs entre les flux estimé et mesuré de la même phase. Ces erreurs tendent vers zéro après un régime transitoire.

L'architecture proposée pour le filtre de Kalman arrive donc à reconstituer les flux rotoriques de façon satisfaisante.

### 4.6.3. Résultats de Synthèse

L'outil **ISE** de **Xilinx** peut être utilisé pendant toutes les étapes de la procédure de développement d'un projet FPGA. Son interface est représentée par la figure 4.37 :

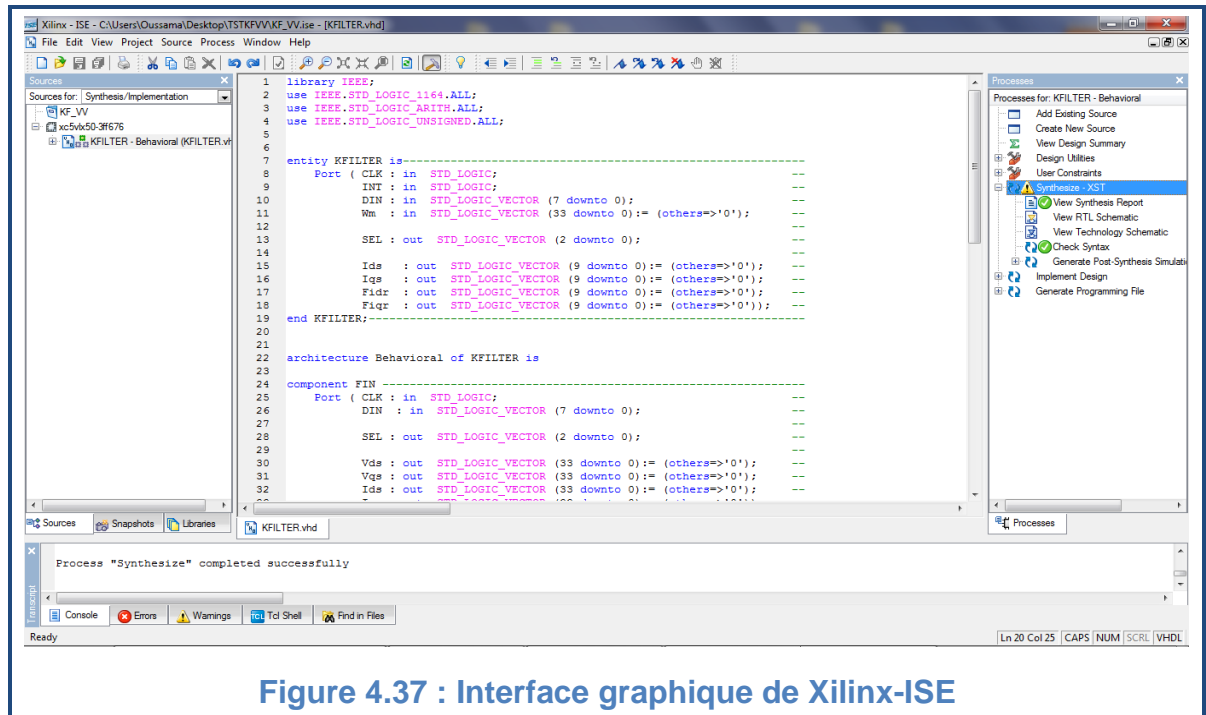
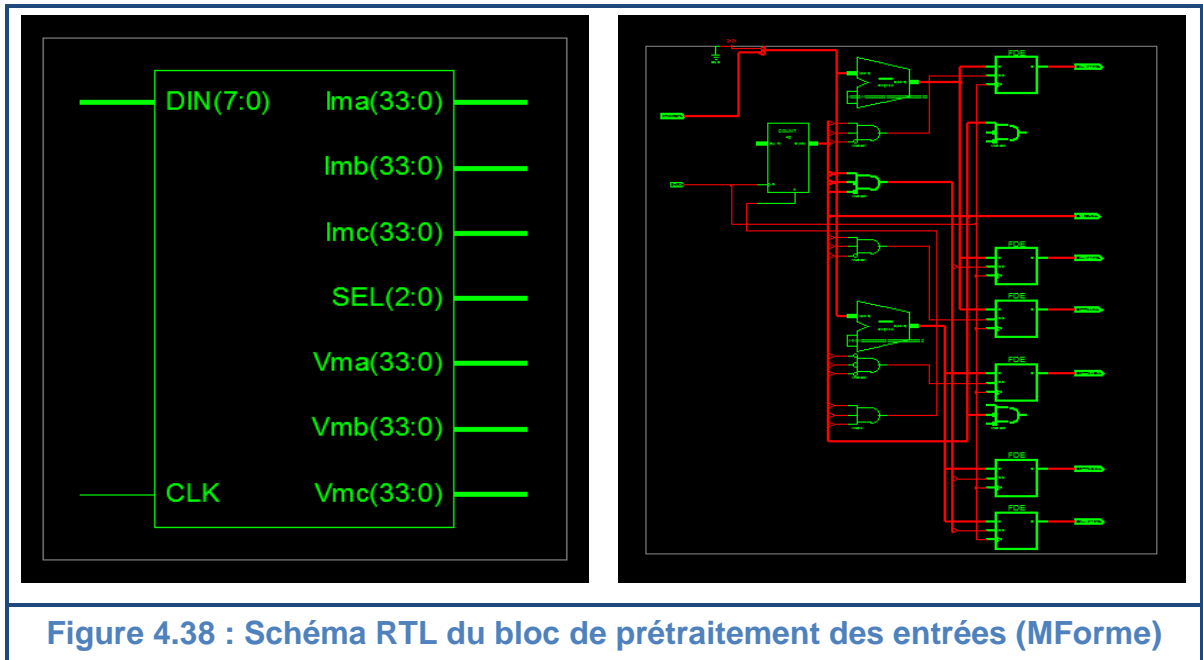


Figure 4.37 : Interface graphique de Xilinx-ISE

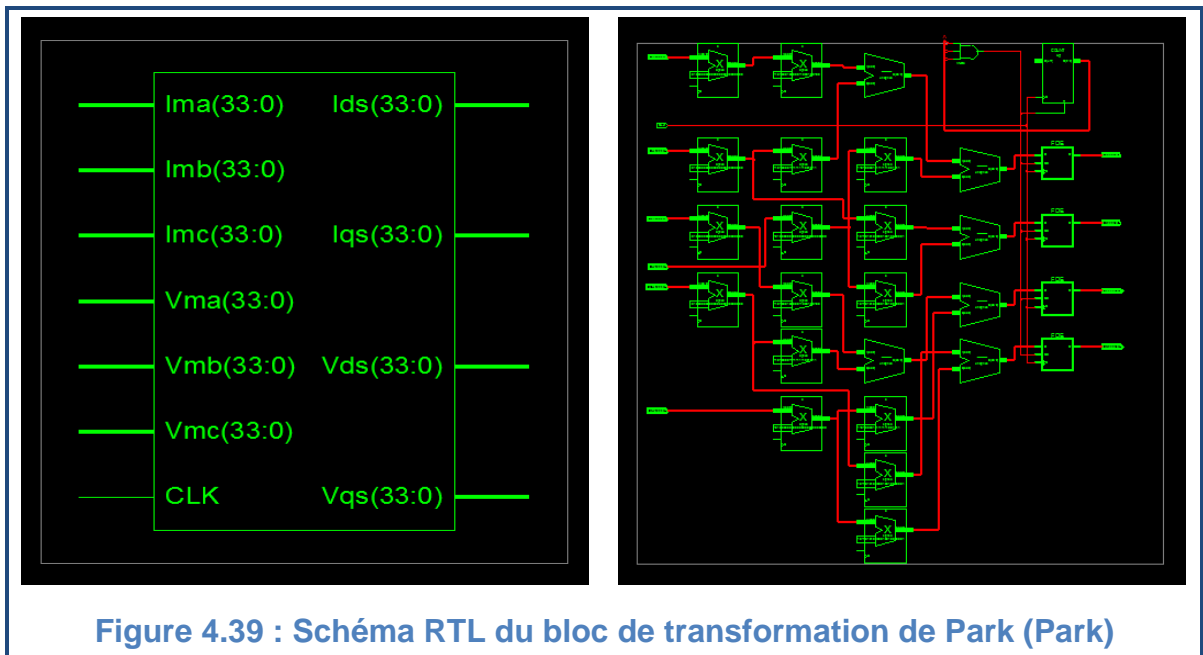
Après la synthèse, on peut visualiser les schémas RTL des différents blocs de notre architecture.

Le premier bloc est le bloc d'entrée (**MForme**), il fait l'aiguillage, l'élimination de la composante continue ainsi que la mise à l'échelle des signaux d'entrée. Ce bloc est représenté par la figure 4.38 :

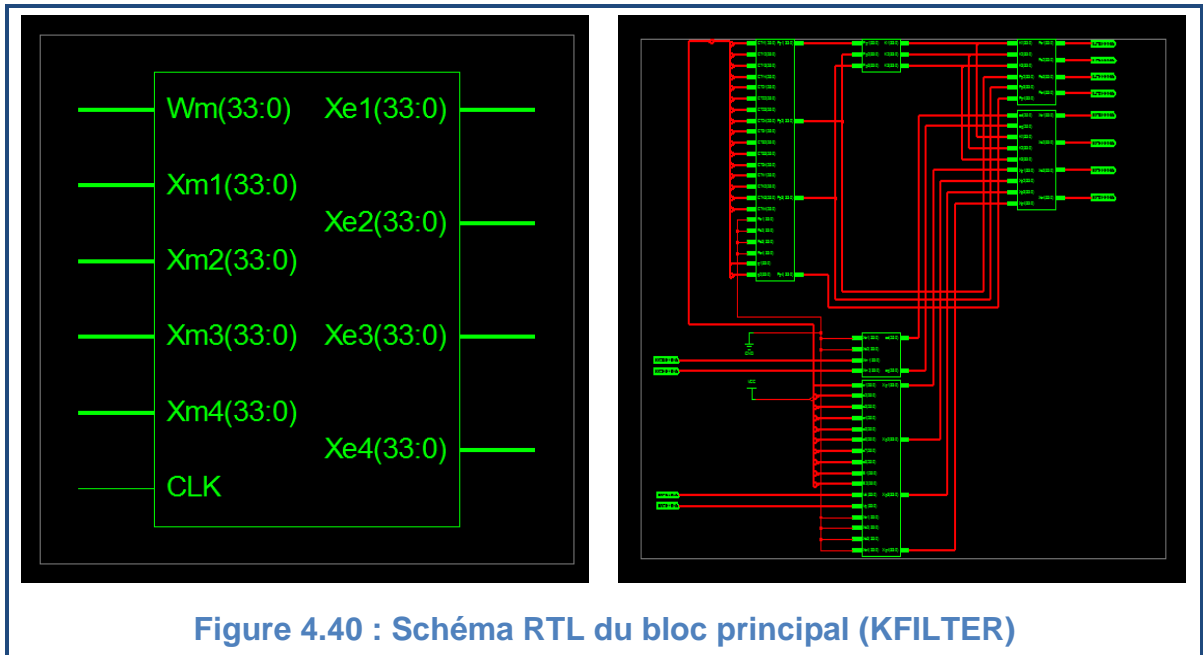




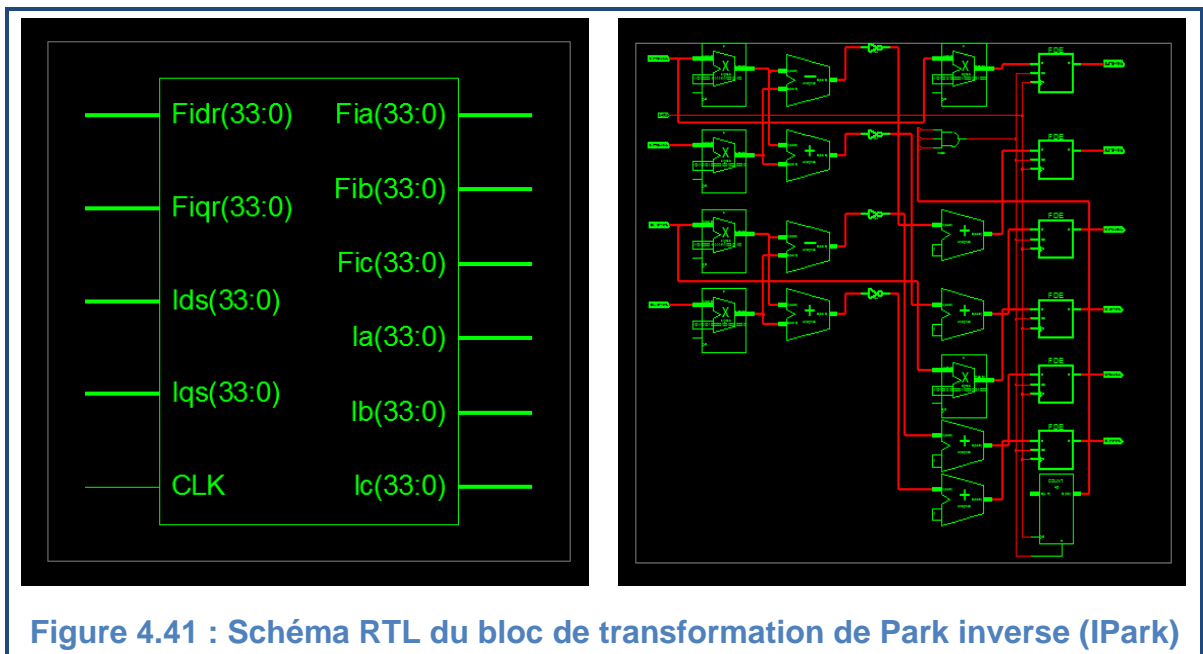
Le bloc (*Park*) applique la transformation de Park sur les signaux d'entrée. Ce bloc est représenté par la figure 4.39



Le filtre de Kalman (*KFILTER*) est le bloc principal dans cette architecture. Il a comme entrées : les deux tensions et les deux courants statoriques ainsi que la vitesse rotorique ( $V_{ds}$ ,  $V_{qs}$ ,  $I_{ds}$ ,  $I_{qs}$ ,  $\omega_m$ ) et comme sorties les deux courants statoriques et les deux flux rotoriques ( $I_{ds}$ ,  $I_{qs}$ ,  $\Phi_{dr}$ ,  $\Phi_{qr}$ ). Ce bloc est représenté par la figure 4.40 :



Le bloc de sortie (**IPark**) fait la transformation de Park inverse. Ce bloc est représenté par la figure 4.41.



L'outil **ISE** nous a donné aussi un rapport de synthèse détaillé. Ce rapport peut être résumé par les deux figures 4.42 et 4.43:

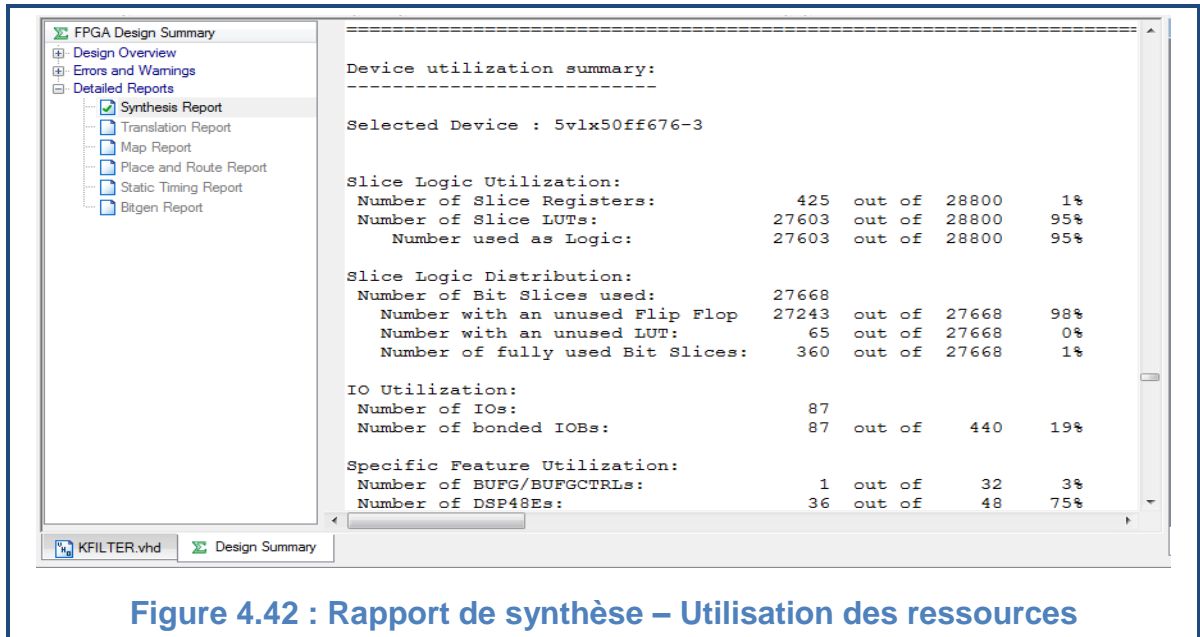


Figure 4.42 : Rapport de synthèse – Utilisation des ressources

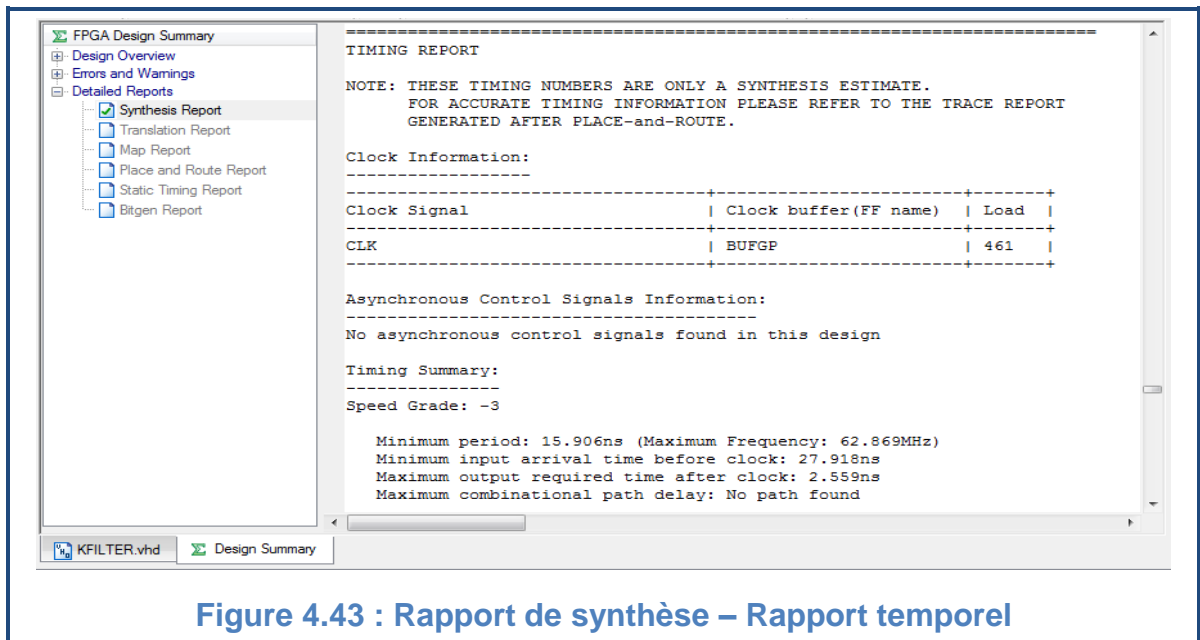


Figure 4.43 : Rapport de synthèse – Rapport temporel

A partir du rapport de synthèse (figure 4.42), on remarque qu'on a utilisé 27668 slices parmi 28800 ce qui représente 98% des ressources du composant FPGA. 360 slices ont été complètement exploitées, 65 slices ont été utilisées comme des registres et 27243 ont été utilisés comme des LUT. On a utilisé également 19% des blocs d'entrée/sortie, 3% des Buffers et 75% des slices DSP.

La figure 4.43 montre que la fréquence maximale du fonctionnement de notre architecture est de 62,869 MHz, ce qui est largement suffisant pour les applications de contrôle.

#### 4.7.Conclusion

Dans ce chapitre, une description détaillée des équations du filtre de Kalman a été proposée. Ces dernières font intervenir des termes constants et des termes qui sont fonction de la vitesse rotorique. Deux formats ont été utilisés pour représenter chacun de ces deux types. Une troncature est nécessaire pour garder, à l'issue des opérations arithmétiques, le même format. La présence des blocs de traitement de signal (DSP48E) a permis de réaliser des multiplications de 43x48 bits, des multiplications de 18x25 bit avec accumulation, des multiplications de 18x25 bit avec addition ou soustraction de 48 bits ainsi que des additions et des soustractions de 43 avec 48 bits.

Les différents choix effectués se trouvent être un très bon compromis entre la précision des résultats obtenus et l'optimisation nécessaire des ressources du FPGA.

## CONCLUSION

Depuis quelques années le filtre de Kalman est l'observateur le plus utilisé pour l'estimation des grandeurs physiques des machines électriques. Il permet de réduire le nombre de capteurs de mesure et donc le coût de revient et le coût de maintenance. Notre travail a porté sur l'application du filtre de Kalman à la reconstruction des flux rotoriques et des courants statoriques de la machine asynchrone triphasé. Celle-ci nécessite un modèle mathématique approprié.

Une première modélisation a été développée dans le repère triphasé. A cause de l'inobservabilité de ce modèle, on a lui appliqué une transformation mathématique (transformation de Park) pour isoler la partie non observable. Cette dernière est éliminée du modèle. La transformation de Park nous permet de passer du repère triphasé à un repère biphasé équivalent.

Pour effectuer les mesures nécessaires à l'identification une machine de 1.1kw a été utilisée. Ses paramètres ont été identifiés à l'aide d'un algorithme génétique qui minimise l'erreur quadratique entre le courant mesuré et celui qui provient du modèle mathématique.

Après l'analyse de l'observabilité de la machine, la synthèse de l'observateur, qui est dans notre cas le filtre de Kalman, a été entamée. Ce dernier a été utilisé dans le cas discret. Pour cette raison, le modèle de Park a été discrétisé à l'aide d'un développement de Taylor à l'ordre deux. Les deux modèles, continu et discret, ont été présentés dans le deuxième chapitre. Le pré réglage du filtre de Kalman a été effectué à l'aide d'un algorithme génétique ; permettant de minimiser un critère d'optimisation basé sur l'erreur quadratique entre l'état estimé et celui fourni par le modèle de la machine. L'hypothèse de la linéarité du modèle de la MAS entre deux instants d'échantillonnage lorsque la vitesse mécanique est mesurée nous a permis d'utiliser un filtre de Kalman pour la reconstruction du vecteur d'état.

Pour parvenir à l'implémentation matérielle du FK, il a été nécessaire de transformer les équations matricielles en équations scalaires ; ensuite ces dernières ont été mises sous une forme adéquate, permettant la mise en évidence

des similitudes existantes entre les termes. Ainsi, quatre fonctions de base ont été développées en langage VHDL, et ont été utilisées pour implémenter, sur une cible FPGA, les différentes équations du FK. Ces quatre fonctions sont formées par:

- ✓ Une multiplication suivie d'une addition,
- ✓ Une multiplication suivie d'une soustraction,
- ✓ Deux multiplications suivies d'une addition, elle-même suivie d'une autre multiplication.
- ✓ Deux multiplications suivies d'une soustraction, elle-même suivie d'une autre multiplication.

Cette implémentation massivement parallèle a été effectuée en ayant pour objectif la minimisation du temps de calcul. Cet objectif a été atteint. Le temps nécessaire pour le calcul du FK est de 15 ns. En contrepartie, les ressources du FPGA ont été consommées en grande partie. L'utilisation d'un circuit FPGA pour implémenter l'architecture proposée pour le FK nous a donné des résultats globalement satisfaisants, car suffisamment proches de la réalité. Par ailleurs, la précision peut être augmentée en modifiant le nombre de bits utilisés dans les différents blocs.

Le temps nécessaire pour le calcul du FK avec l'approche utilisée est très court. Ceci laisse largement le temps à l'exécution d'algorithmes plus complexe, destinés à la commande et au diagnostic de la MAS. Ceux-ci nécessitent une fréquence de traitement élevée. Cependant, la consommation des ressources du FPGA est aussi un point crucial. L'amélioration du rendement global est obtenue en faisant une répartition judicieuse entre le temps de calcul (parallélisme) et la consommation des ressources matérielles. L'avantage de l'architecture proposée réside dans sa modularité. L'introduction de fonctions de commande permettra d'utiliser les blocs proposés de façon séquentielle, afin de réduire la consommation des ressources matérielles. Le temps de calcul sera plus long, mais devra rester dans les limites acceptables. On pourra ajuster le niveau de parallélisme pour trouver le bon compromis entre le temps de calcul, de l'algorithme global (observation-commande ou observation-diagnostic) et l'utilisation de ressources matérielles du FPGA. Ceci constitue une perspective très intéressante pour la suite de ce travail.

**APPENDICE A**  
**LISTE DES SYMBOLES**

MAS	Machine Asynchrone
FK	Filtre de Kalman
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Arrays
ASIC	Application Specific Integer Circuit
$P$	Nombre de paires de pôles
$g$	Glissement
$\theta_m, \omega_m, \Omega_m$	Angle, Pulsation et vitesse mécaniques
$\theta_r, \omega_r, \Omega_r$	Angle, Pulsation et vitesse Rotoriques
$\theta_s, \omega_s, \Omega_s$	Angle, Pulsation et vitesse Statoriques
$C_{em}, C_r$	Couple électromagnétique et couple de charge
$v_{sa, sb, sc}, v_{ra, rb, rc}$	Tensions Triphasés Statoriques et Rotoriques
$i_{sa, sb, sc}, i_{ra, rb, rc}$	Courants Triphasés Statoriques et Rotoriques
$\varphi_{sa, sb, sc}, \varphi_{ra, rb, rc}$	Flux Triphasés Statoriques et Rotoriques
$v_{ds, qs}, v_{dr, qr}$	Tensions de Park
$i_{ds, qs}, i_{dr, qr}$	Courants de Park
$\varphi_{ds, qs}, \varphi_{dr, qr}$	Flux de Park
$l_s$ et $l_r$	Inductances propres statorique et rotorique.
$l_{ss}$ et $l_{rr}$	Inductances mutuelles entre deux phases statoriques ou rotoriques
$l_{sr}$	Inductance mutuelle maximale entre une phase statorique et une phase rotorique
$L_s, L_r$	Inductances Cycliques Statorique et Rotorique
$L_m$	Inductance Mutuelle Cyclique entre Stator et Rotor
[T]	Matrice de transformation de Park
$R_s, R_r$	Résistances statorique et rotorique.

$J$	Moment d'inertie de la MAS
$f$	Frottement Totale de la MAS et de la charge
$AG$	Algorithme Génétique
$P_s P_c P_m$	Probabilités de Sélection, de Croisement et de Mutation
$f_e$	Fonction d'évaluation (AG)
$W et V$	Bruits d'état et Bruits de mesure
$Q et R$	Matrices de Variance-Covariance des bruits d'état et des bruits de mesure
$x(t), \hat{x}(t)$	Etats Prédit et Estimé
$\varepsilon(t)$	Erreur de prédiction
$K$	Matrice de Gain de Kalman
$X_p$	Vecteur d'état prédit
$P_p$	Matrice de Variance-Covariance d'erreur de prédiction
$X_e$	Vecteur d'état estimé
$P_e$	Matrice de Variance-Covariance d'erreur d'estimation



## APPENDICE B

### ARCHITECTURE INTERNE DE Virtex5 DE Xilinx

Le circuit FPGA est un réseau de blocs logiques programmables (**CLB**) organisées dans une structure matricielle et de blocs d'entrée/sortie (**IOB**) placés sur la périphérie du circuit FPGA, le tout est interconnectées par un réseau d'interconnexions programmables. La figure suivante simplifie la représentation de l'architecture interne d'un FPGA.

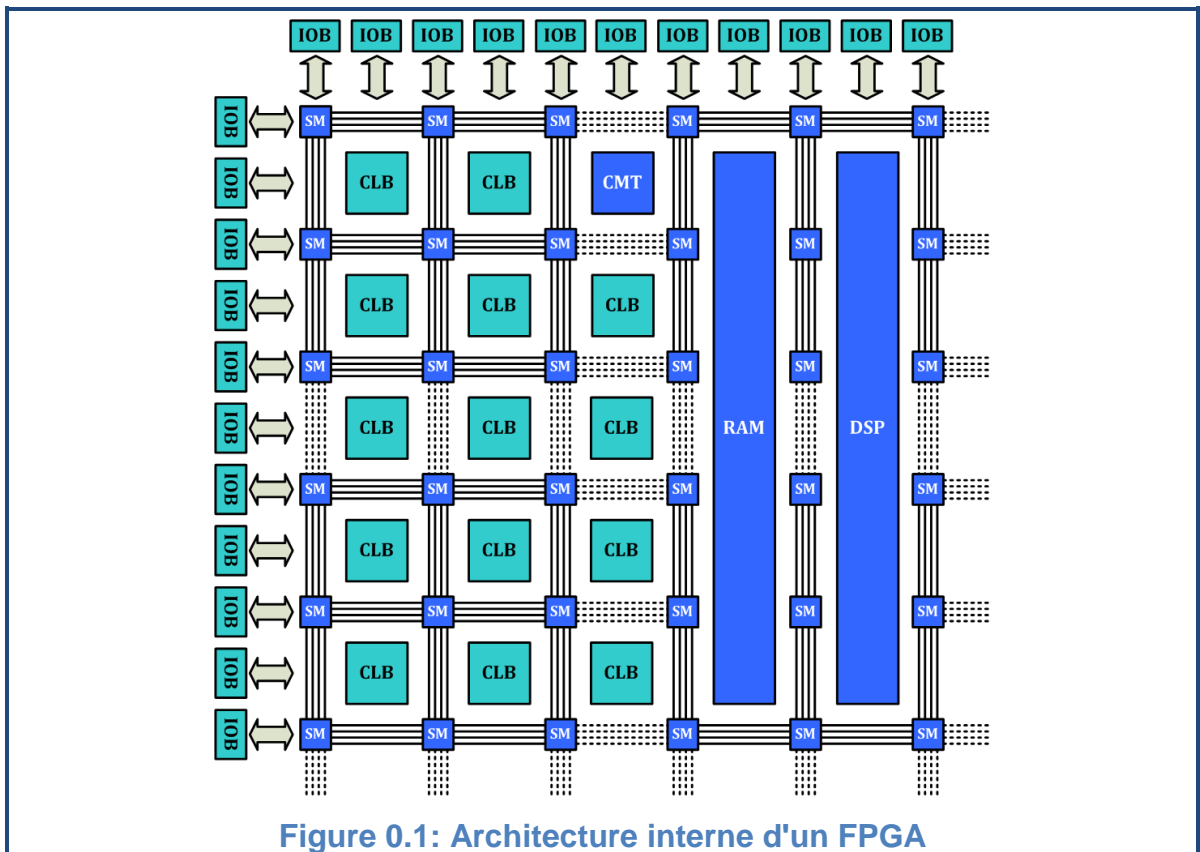


Figure 0.1: Architecture interne d'un FPGA

Le composant Virtex-5 **XC5VLX50 -3 FFG 676 C** contient une matrice de 120X30 **CLB** et 560 **IOB**. Ce FPGA contient aussi plusieurs blocs dédiés pour différentes utilisations, on site : 48 blocs **RAM** de 36kb de type FIFO, 06 blocs de gestion d'horloge (**CMT**), et 48 blocs de traitement du signal (**DSP48E**) [55], [36].

## 1. Bloc d'Entrées/Sorties (IOB)

Ces blocs entrée/sortie (**Input/Output Blocks**) permettent l'interface entre les broches du composant FPGA et la logique interne développée à l'intérieur du composant. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé (haute impédance).

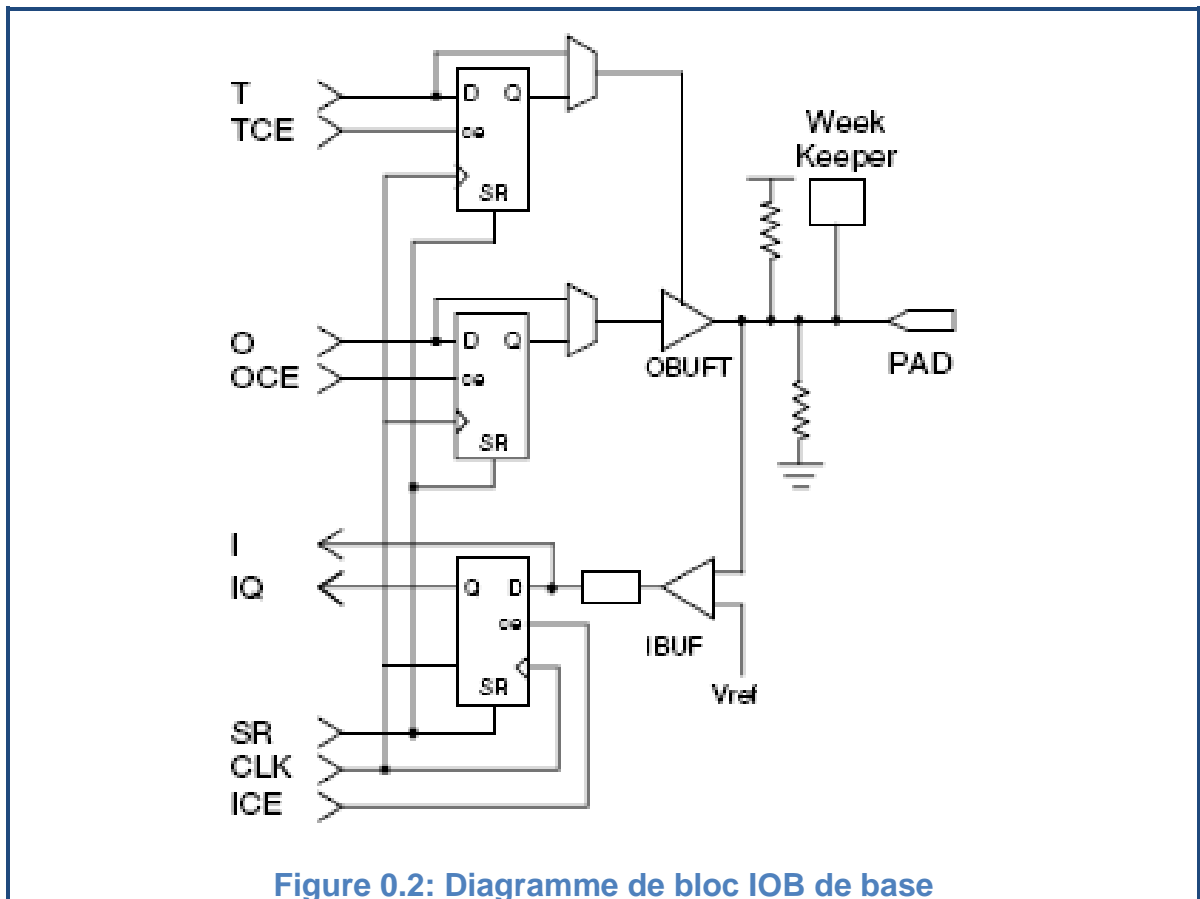
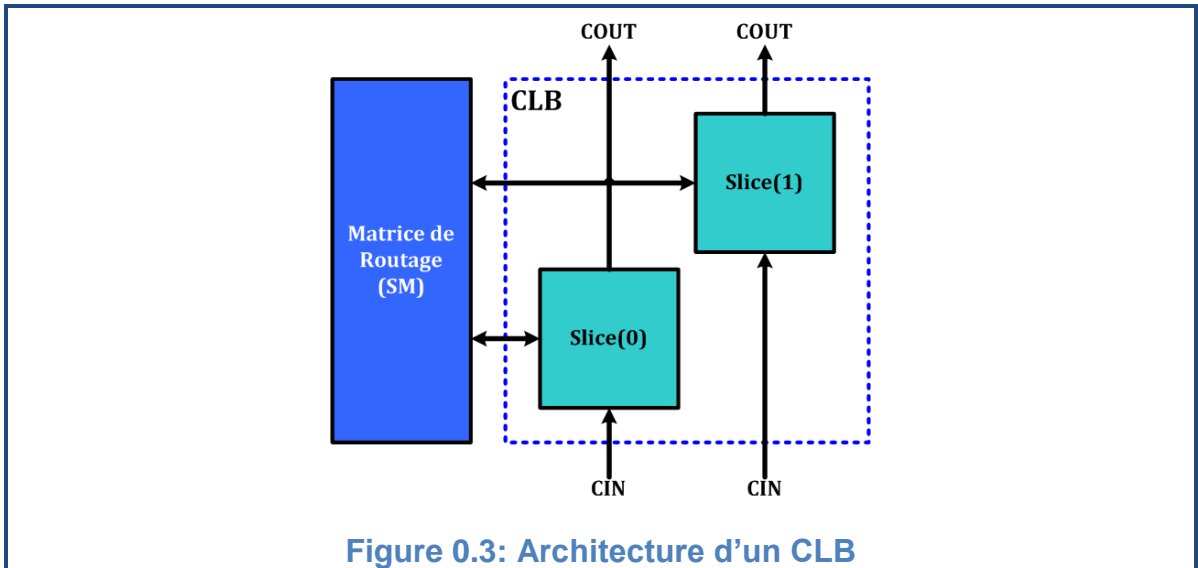


Figure 0.2: Diagramme de bloc IOB de base

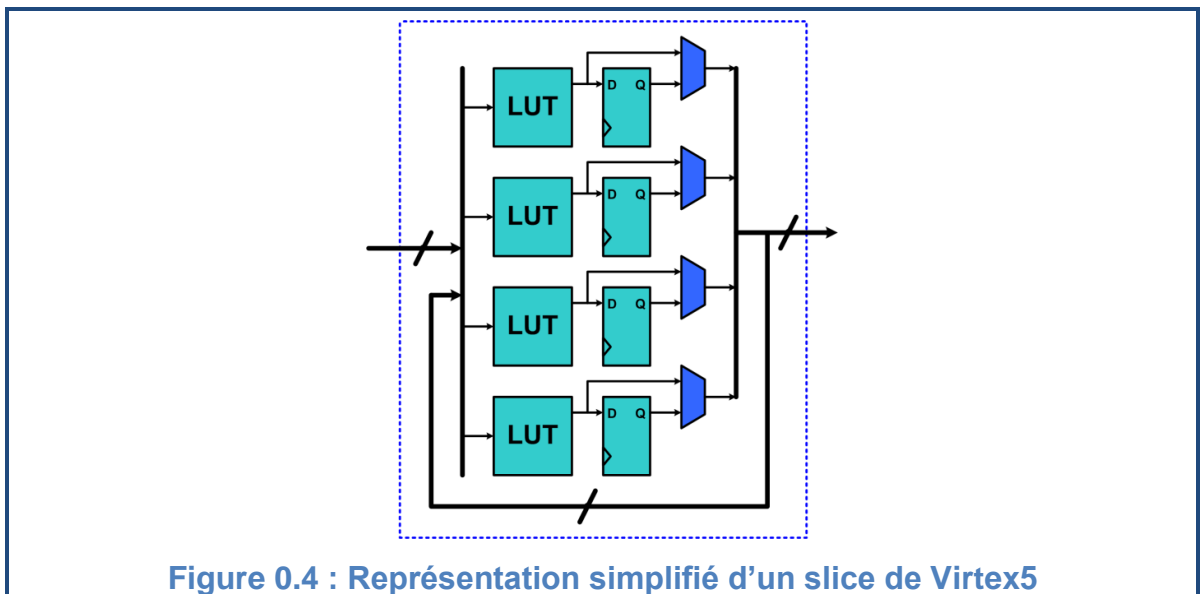
## 2. Bloc Logique Configurable (CLB)

Le bloc logique configurable (**Configurable Logic Blocks**) est l'élément de base d'un composant FPGA, il est constitué de deux **slices** sans connections directes entre eux.



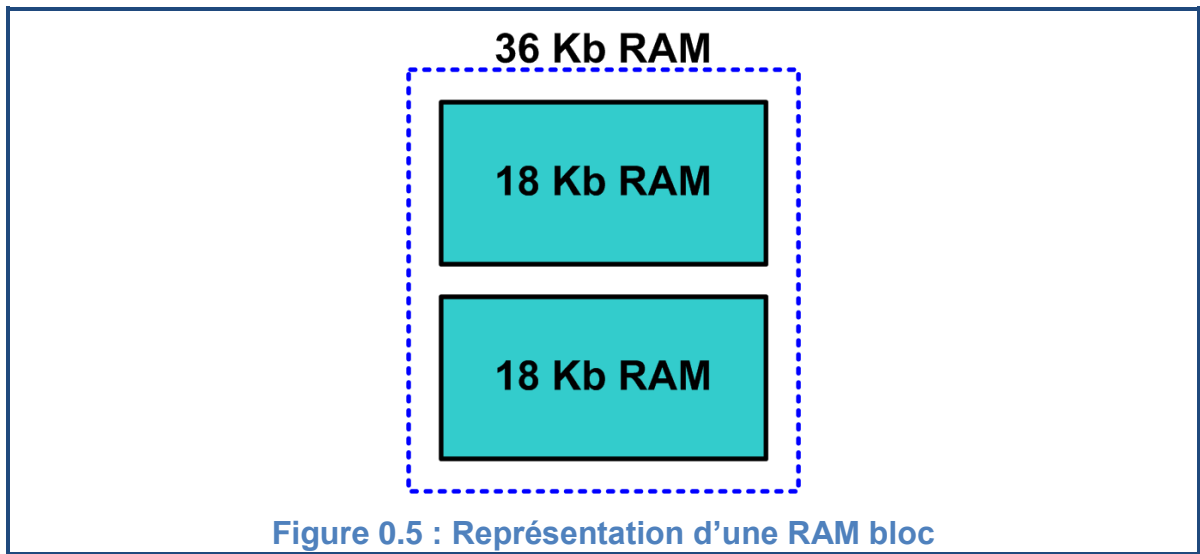
Un *slices* est de manière générale constitué de quatre tables de correspondance **LUT (Look-Up-Table)**, de quatre bascules D et de quelques multiplexeurs. Elles sont utilisées pour réaliser des fonctions logiques, arithmétiques, décalages et ROM. Elles peuvent être considérées aussi comme une RAM distribuée.

La LUT est une petite mémoire sert à implémenter des équations logiques de 6 entrées et deux sorties. Elle peut toutefois être considérée comme, un multiplexeur ou un registre à décalage. Le registre permet de mémoriser un état (machine séquentielle) ou de synchroniser un signal (pipeline) [57].



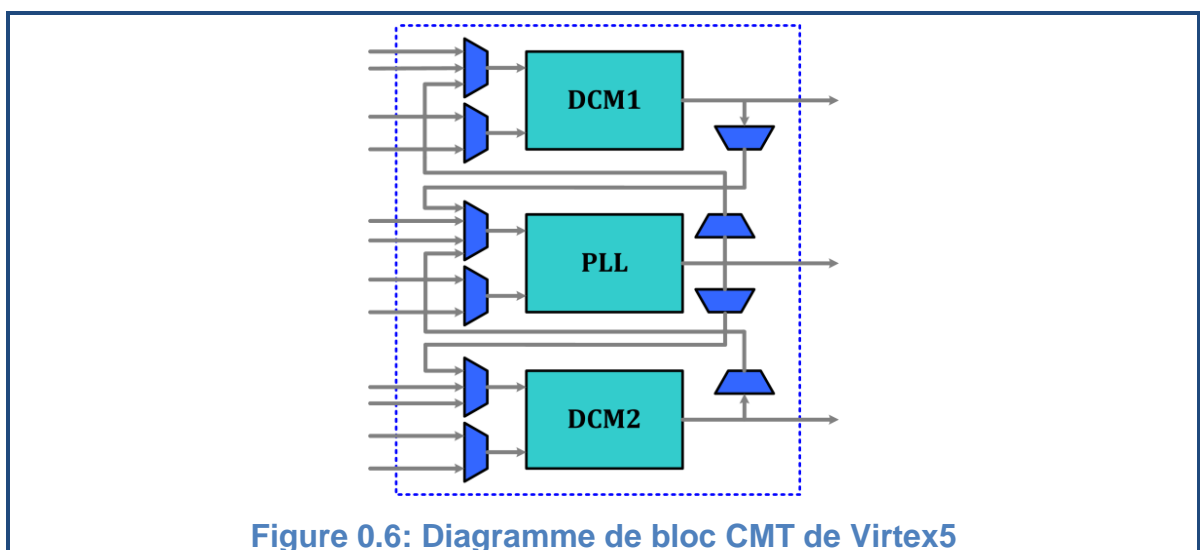
### 3. Bloc de mémoire vive (RAM Bloc)

Un FPGA de famille Virtex5 contient 48 **RAM** bloc (*Random Access Memory*) de 36kb organisées en colonnes. Chacune de ces RAM est constituée de deux blocs de 18 kb contrôlé indépendamment. Ces blocs peuvent être utilisés séparément, comme ils peuvent être utilisés en cascade pour former des RAM de plus grande taille, des ROM ou des mémoires FIFO synchrones ou asynchrones.



### 4. Cellule de gestion d'Horloges (CMT)

Les blocs **CMT** (*Clock Management Tile*) dans la famille Virtex5 assurent une gestion souple et très performante des horloges dans un circuit FPGA. Chaque CMT contient deux blocs **DCM** et un bloc **PLL**.



#### 4.1. Gestionnaire d'Horloge Digitale (DCM)

Les blocs DCM (**Digital Clock Manager**) des FPGA sont utilisés pour générer des signaux d'horloge internes au FPGA parfaitement en phase avec les horloges sources venant de l'extérieur ou bien déphasé de 90°, 180°, ou 270° de la version originale.

Les DCM contiennent des unités **DLL (delay-locked loop)** qui éliminent les délais de distribution du signal d'horloge dans les endroits éloignés de la source.

Ils sont également capables de générer un grand nombre de fréquences d'horloge différentes, à travers un ensemble de diviseurs/multiplicateurs de fréquence.

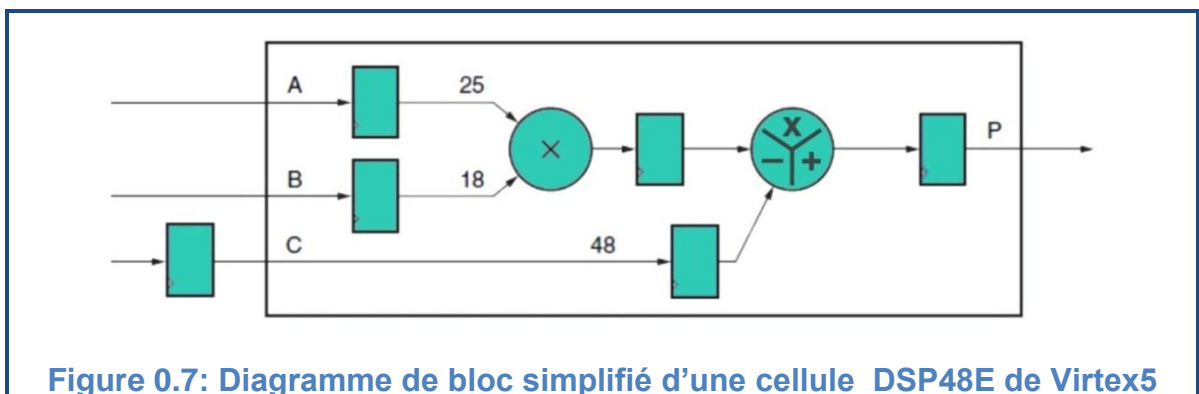
#### 4.2. Boucle à Verrouillage de Phase (PLL)

Le **PLL (Phase-Locked Loop)** est utilisé principalement comme un synthétiseur pour une large bande de fréquences, il se sert aussi avec le DCM pour filtrer tous les signaux d'horloge internes et externes [44].

#### 5. Bloc de traitement de signal (DSP48E)

La Virtex5 contient une deuxième génération de blocs DSP c'est la **DSP48E**. Ces blocs peuvent réaliser plusieurs fonctions mathématiques indépendantes pour le traitement des signaux comme la multiplication, multiplication addition, multiplication accumulation ...

L'architecture simplifiée d'un bloc DSP48E peut être représentée par la figure suivante :



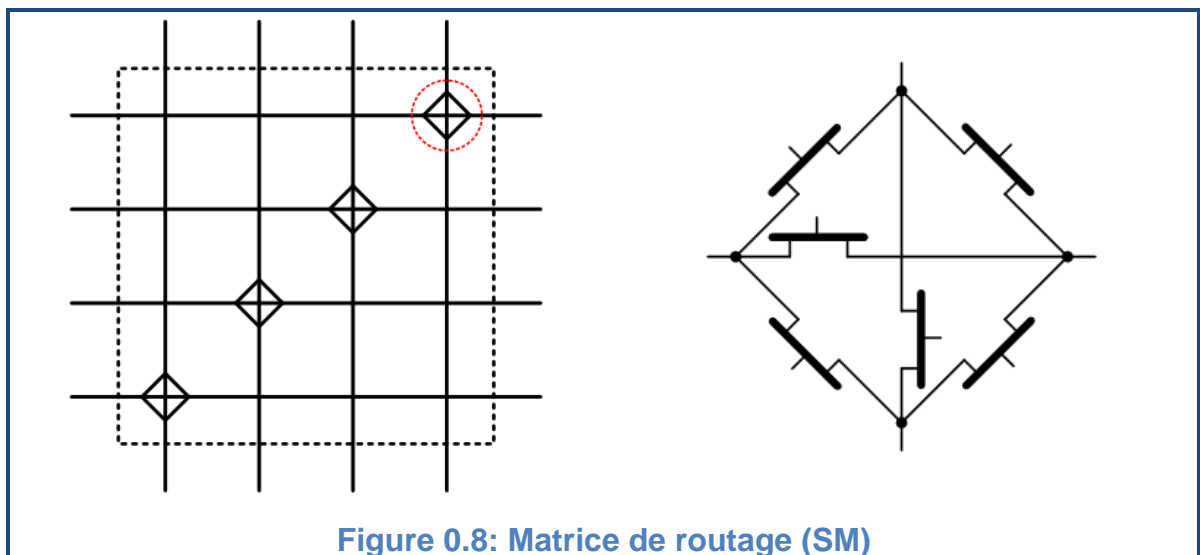
## 6. Réseau d'interconnexions

Le réseau d'interconnexions permet la réalisation des connexions (le routage) entre les différentes tranches au sein d'un même CLB, entre les différents CLBs, les CMTs, les multiplieurs, les DSP48E, la mémoire RAM et les IOBs utilisés pour la construction d'un circuit donné.

Au sein d'un même CLB, les différentes tranches peuvent communiquer via des connexions rapides. Une matrice de routage (**SM**) est associée à chaque CLB, IOB, CMT, à la mémoire RAM, au multiplieur et au blocs DSP48E.

Le rôle de cette matrice est de réaliser une connexion programmée entre les conducteurs "entrants" et "sortants". La programmation de la connexion est établie à l'aide d'un mot de mémoire défini par la configuration. Grâce aux **SMs** toute ressource peut donc accéder à toute autre ressource, quel que soit l'endroit où elle se trouve dans le circuit FPGA.

Les conducteurs "entrants" et "sortants" d'une matrice SM sont disposés entre les lignes et les colonnes des CLBs.



**Figure 0.8: Matrice de routage (SM)**

## APPENDICE C

### IDENTIFICATION PAR METHODE CLASSIQUE

L'identification classique détermine les paramètres en utilisant le schéma équivalent de la machine asynchrone à l'aide des trois essais classiques suivants [29]:

- L'essai à courant continu,
- l'essai à vide,
- l'essai à rotor bloqué ou bien en fonctionnement à charge et vitesse nominales.

La figure suivante représente le schéma monophasé équivalent de la MAS tel que  $R_f$  modélise les pertes ferromagnétiques et  $N_r$  est l'inductance de fuite totale.

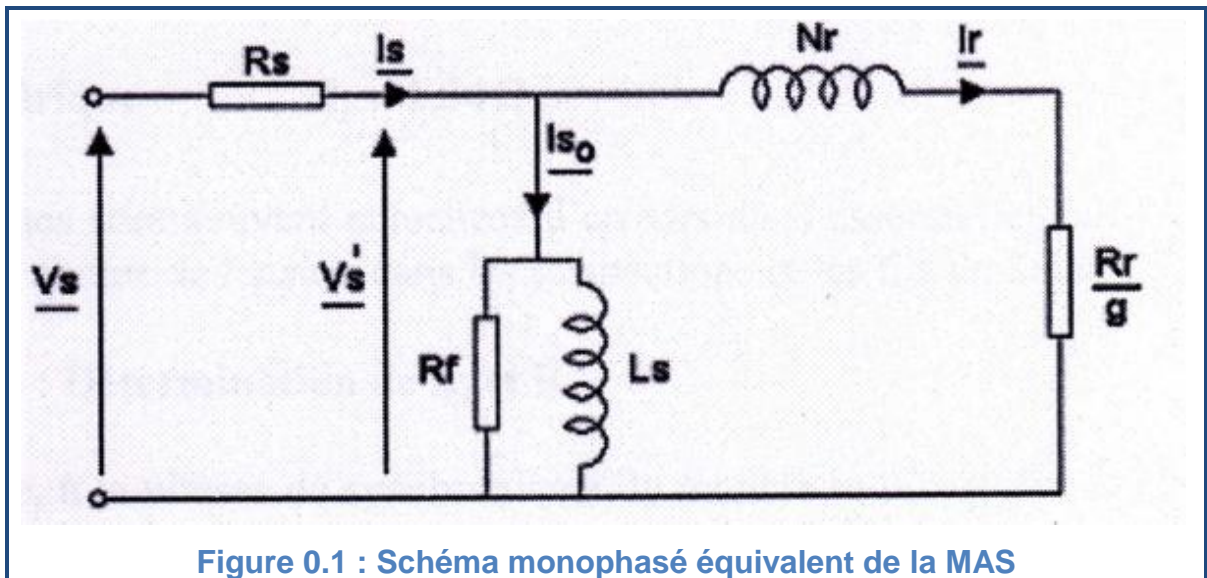
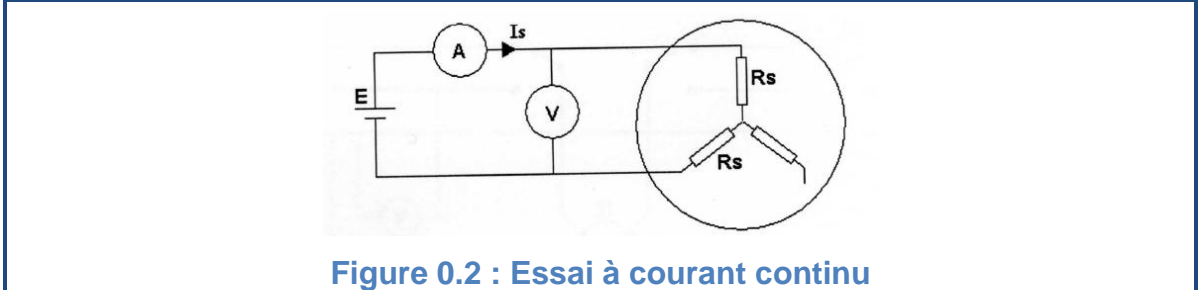


Figure 0.1 : Schéma monophasé équivalent de la MAS

## 1. Détermination de $R_S$

Pour qu'on puisse déterminer la résistance statorique  $R_S$ , On doit alimenter la machine en étoile avec de faibles tensions continue pour des raisons de sécurité. Cet essai s'appelle : **essai à courant continu**.



La valeur de la résistance statorique est obtenue en appliquant la loi d'Ohm, par la relation :

$$R_s = \frac{E}{2I_s} \quad (\text{C.1})$$

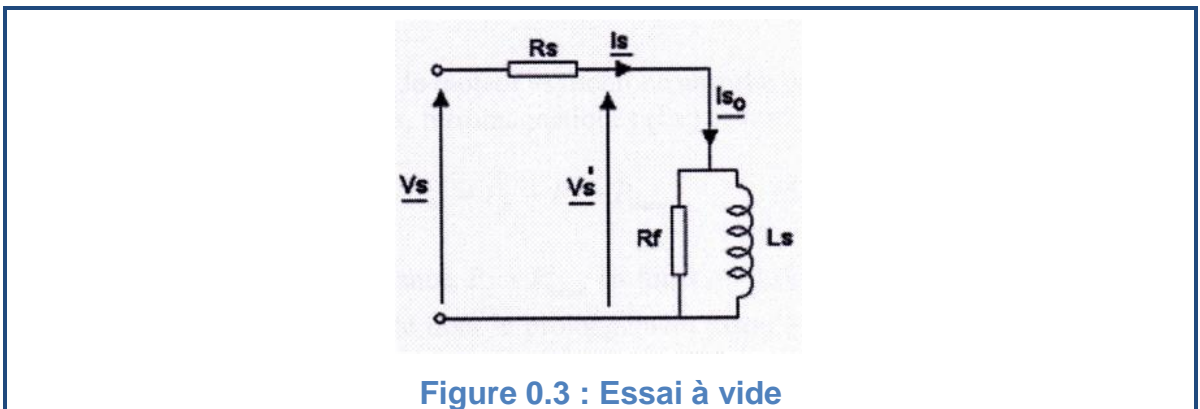
## 2. Détermination de $R_f, L_s, J, f$

Cette étape s'appelle aussi **essai à vide** (sans charge mécanique). Pratiquement, elle contient deux essais :

- Le premier est à la vitesse de synchronisme et permet la détermination de:  $L_s, R_f$ .
- Le deuxième est l'essai au ralentissement et détermine  $J$  et  $f$ .

### 2.1. Essai à vitesse constante

Dans l'essai à vide, le rotor tourne à la vitesse de synchronisme, donc le glissement  $g$  tend vers zéro et le modèle se réduit aux trois éléments  $R_s, R_f$  et  $L_s$ .





Dans cette étape on fait une série de mesures des paramètres statoriques (la tension ( $V$ ), le courant ( $I$ ) et les puissances ( $P$  et  $Q$ )):

La puissance absorbée  $P_s$  correspond à la somme des pertes mécaniques, magnétiques et joules au stator :

$$P_s = P_{j_s} + P_{f_s} + P_m \quad (\text{C.2})$$

Où :

- Les pertes mécaniques sont la somme de pertes par frottements et des pertes par ventilation.
- Les pertes fer sont les pertes par courant de Foucault et par Hystérésis.
- Les pertes Joules sont liées aux résistances des trois phases statoriques. Elles peuvent être calculées par l'équation suivante :

$$P_{j_s} = 3R_s I_s^2 \quad (\text{C.3})$$

En traçant la courbe  $(P_{f_s} + P_m)$  en fonction de  $(U_s^2)$  avec  $U_s = V_s \sqrt{3}$ . On obtient quasiment une droite. En la linéarisant et en la prolongeant jusqu'à la tension nulle on trouve la valeur de la somme des pertes mécaniques ( $P_m$ ). On peut maintenant déduire la valeur de  $(P_{f_s})$ .

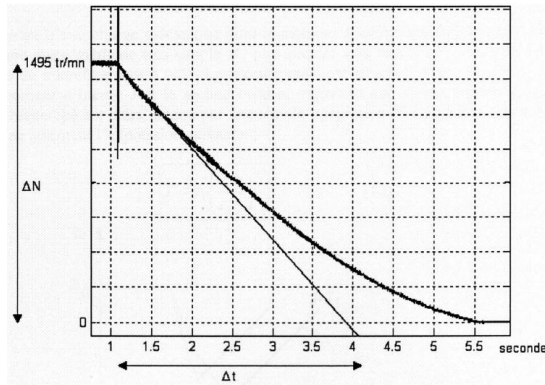
Et on peut finalement calculer les deux paramètres  $L_s, R_f$  à partir des deux équations suivantes :

$$R_f = \frac{U_s^2}{P_{f_s}} \quad (\text{C.4})$$

$$L_s = \frac{U_s^2}{Q_s \omega_s} \quad (\text{C.5})$$

## 2.2. Essai au ralentissement

Cet essai est basé sur le relevé de la courbe de ralentissement de la machine à partir de la vitesse nominale



**Figure 0.4 : Courbe de ralentissement de la machine à partir de la vitesse nominale**

Le moment d'inertie  $J$  peut être calculée par :

$$J = \frac{P_m}{\Omega_n \left( \frac{d\Omega}{dt} \right)_{\Omega=\Omega_n}} \quad (\text{C.6})$$

Avec:

$$\Omega_n \cong 2\pi \frac{N_n}{60} \quad \text{et} \quad \frac{d\Omega}{dt} \cong \frac{2\pi \Delta N}{60 \Delta t} \quad (\text{C.7})$$

Pour calculer le coefficient de frottement visqueux  $f$ , on doit mesurer le couple résistant  $C_r$  lorsque la machine tourne à la vitesse nominale, et puisque ce couple en régime permanent à vide n'est dû qu'aux frottements visqueux, on peut donc déduire le coefficient  $f$  à partir de la relation suivante :

$$C_r = f \Omega_m \quad (\text{C.8})$$

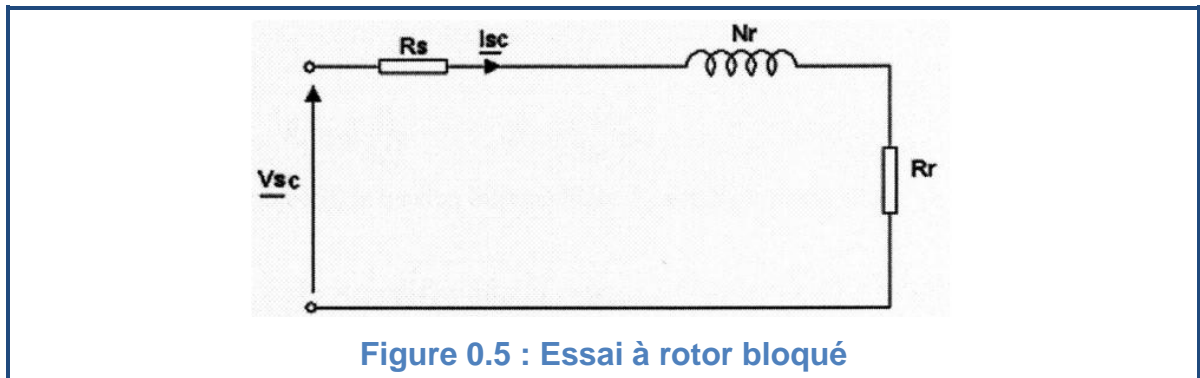
### 3. Détermination de $R_r$ et $M$

Pour déterminer la résistance rotorique  $R_r$  et l'inductance mutuelle  $M$  on peut citer deux méthodes

#### 3.1. Essai à rotor bloqué

Lorsque le rotor est bloqué, le glissement est maximal ( $g = 1$ ), la tension statorique doit être ajustée de telle sorte que le courant statorique soit proche du courant nominal. Dans ces conditions, la tension appliquée devient nettement

faible devant la tension nominale, ce qui permet de négliger le courant  $I_s$  et de réduire le schéma de la **Figure C.1** à celui de la figure qui suit :



Donc on peut écrire :

$$P_s = 3(R_s + R_r)I_s^2 \quad (\text{C.9})$$

$$Q_s = 3(N_r \omega_s)I_s^2 \quad (\text{C.10})$$

On déduit :

$$R_r = \frac{P_s}{3I_s^2} - R_s \quad (\text{C.11})$$

$$N_r = \frac{Q_s}{3\omega_s I_s^2} \quad (\text{C.12})$$

### 3.2. Essai à la vitesse nominale

Cette méthode repose sur le modèle la **figure C.1** Une tension intermédiaire  $V_{si}$  est d'abord calculée tel que :

$$V_{si} = \frac{\sqrt{(P_s - 3R_s I_s^2)^2 + Q_s^2}}{I_s} \quad (\text{C.13})$$

Et puis on déduit  $P_i$  et  $Q_i$  :

$$P_i = P_s - 3R_s I_s^2 - 3\frac{V_{si}^2}{R_f} \quad (\text{C.14})$$

$$Q_i = Q_s - 3\frac{V_{si}^2}{L_s \omega_s} \quad (\text{C.15})$$

Avec  $R_f$  est calculé dans (C.4).

La valeur de  $I_r$  est alors :

$$I_r = \frac{\sqrt{P_i^2 + Q_i^2}}{V_{si}} \quad (\text{C.16})$$

Et on déduit maintenant  $R_r$  et  $N_r$  :

$$R_r = g \frac{P_i}{3I_r^2} \quad (\text{C.17})$$

$$N_r = \frac{Q_i}{3\omega_s I_r^2} \quad (\text{C.18})$$

Finalement et à partir de (C.11) et (C.12) ou bien (C.18) et (C.19) on peut déterminer  $M$ :

$$M = \frac{-N_r + \sqrt{N_r^2 + 4L_s^2}}{2} \quad (\text{C.19})$$

Les essais de cette méthode nécessitent un matériel spécifique avec des compétences pratiques considérables parce qu'ils portent le risque de :

- Casser l'arbre du tachymètre à cause d'une mauvaise installation.
- Casser l'arbre de la machine à cause de l'application d'une grande charge au rotor.
- Détériorer le bobinage de la machine à cause de l'application d'un grand courant continu au stator.

Ce qui nous a conduit à utiliser des techniques numériques basées sur le modèle mathématique de la machine [26, 47].

## REFERENCES

1. Kalage A.A., « Modeling and Simulation of FPGA based direct Torque Control of Induction Motor Drive », International Journal of Recent Trends in Engineering, V.1, N° 4, (May 2009),
2. Mailloux J.G. Simard S. Beguenane R., « FPGA implementation of Induction Motor Vector Control using Xilinx System Generator », 6th WSEAS International Conference on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING, Cairo, Egypt, (Dec 2007),
3. Rezgui S.D., « Commande de machine électrique en environnement Matlab/Simulink et Temps Réel », Thèse de Magister, Université Mentouri de Constantine, (2009).
4. Hilairet M., Auger F., Darengosse C., « Two efficient Kalman filters for flux and velocity estimation of induction motors », IEEE2000.
5. Soto G.G., Mendes E. and Razek A., « Reduced-order observers for rotor flux, rotor resistance and speed estimation for vector controlled induction motor drives using the extended Kalman filter technique », IEE Proc -Electr Power Appl , Vol 146, No 3, May 1999
6. Djemai M., Barbot J.P., Glumineau A. et Boisliveau R., « Observateur à Modes Glissants pour la Machine Asynchrone : Etude Expérimentale », Journal Européen des Systèmes Automatisés, Vol. X2002, (2002),
7. Idir A.H., Belmehdi A. and Chikouche D., « Recherche de Signatures de Défaut de la Machine à Induction en Vue de Diagnostic », 4th International Conference on Computer Integrated Manufacturing, (2007),

8. EL-merraoui K., « Détection de défauts du moteur a induction à l'aide d'observateurs », Thèse de Magister, université de Blida, (2008),
9. Ouachani R., Bargaoui Z. and Taha O., « Intégration d'un filtre de Kalman dans le modèle hydrologique HBV pour la prévision des débits », Hydrological Sciences Journal, avril 2007.
10. Racicot F.E. et Théorêt R., « Quelques applications du filtre de Kalman en finance estimation et prévision de la volatilité stochastique et du rapport couts-bénéfices », Cahier de recherche 15, ESG Université de Québec à Montreal, (2005),
11. Martin M.M., « Filtrage de Kalman d'une série temporelle saisonnière. Application à la prévision de consommation d'électricité », Revue de Statistique Appliquée V.2, N°4, (1999),
12. Habbani L et El Ahoual K, « Estimation conjointe des variables d'état et des paramètres du moteur », Thèse d'ingénieur, université de Blida, (2008),
13. Gunabalan R., Subbiah V. and Reddy B.R., « Sensorless Control of Induction Motor with Extended Kalman Filter on TMS320F2812 Processor », International Journal of Recent Trends in Engineering, V 2, N° 5, November 2009
14. Barut M., Bogosyan S. and Gokasan M., « Speed-Sensorless Estimation for Induction Motors Using Extended Kalman Filters », IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 1, FEBRUARY 2007
15. Samuelsen D. and Sulkowski W., « Asynchronous machine stator resistance estimation using integrated PWM modulator and sampler unit as FPGA application ». IEEE 2008.
16. Aggounitane I et Belhadia S, « application du filtre de Kalman étendu pour l'estimation conjointe des états et des paramètres d'une machine asynchrone », Thèse de d'ingénieur, université de Blida, (2009),

17. Yeh H.G., « Processing Performance of Two Kalman Filter Algorithms with a DSP32C by Using Assembly and C Languages », IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 38, NO. 4, AUGUST 1991
18. Naouar M.W., « Commande numérique à base de composants FPGA d'une Machine Synchronique », Thèse de doctorat, Université de Cergy Pontoise, France, (2007).
19. Idkhajine L, Monmasson E, Maalouf A., « FPGA-Based Sensorless Controller for Synchronous Machine using an Extended Kalman Filter », EPE 2009 – Barcelona.
20. Chabbi S et M'barki A, « Application du filtre de Kalman Adaptatif à la MAS », Thèse d'ingénieur, université de Blida, (2010),
21. Doumbia L, Roy G, Rajagopalan V, « An Integrated Solution for Simulating Electrical Drive Systems with Matlab/Simulink », International Symposium on Industrial Electronics, 1997.
22. Boumaiza F et Debbi W, « Estimation des paramètres de la MAS à l'aide du filtre de Kalman adaptatif », Thèse d'ingénieur, université de Blida, (2011),
23. Kifouche A., « L'estimation conjointe des variables d'état et des paramètres d'un moteur asynchrone », Thèse de magister, université de Blida, (2011),
24. Meroufel A., Massoum M., Belabbes B., Bentaallah A., « Linéarisation entrée sortie et réglage flou d'une machine asynchrone alimentée en courant », ACTA ELECTROTEHNICA, V 48, N° 1 (2007),
25. Eltabach M., Antoni J. and Najjar M., « Quantitative analysis of noninvasive diagnostic procedures for induction motor drives », Mechanical Systems and Signal Processing 21, (2007),

26. Oueld aissa A., Bounekhla M., Taieb A., Elmerraoui K. et Boudissa E., « Proposition d'une structure PSO. Application à l'identification paramétrique de la machine asynchrone », CGE'06, EMP, ALGER, (Avril 2009),
27. Baghli L. and Rezzoug A. « Particle Swarm and Genetic Algorithms applied to the identification of Induction Machine Parameters », EPE 2003 - Toulouse
28. Toliyat H.A., Levi E. and Raina M., « A Review of RFO Induction Motor Parameter Estimation Techniques », IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 18, NO. 2, JUNE 2003
29. Reama A, Cela A, Benmansour K, « Méthode d'identification des machines asynchrones à cage d'écureuil », Rapport, Laboratoire COSI, 2007.
30. Ghanes M. « Observation et commande de la machine asynchrone sans capteur mécanique », Thèse de doctorat, université de Nantes, France, (2005),
31. Chaouch S., « Commande vectorielle robuste d'une machine à induction sans capteur de vitesse », Thèse de Doctorat, université de Batna, (2005),
32. Hilairet M., « Application des outils du traitement du signal à la commande des machines tournantes », Thèse de doctorat, Université de Nantes, France, (2001),
33. Malrait F., « Problèmes d'identification et d'observabilité du moteur à induction pour la variation de vitesse industrielle sans capteur », Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris, France, (2001),
34. Martinez L.Z. and Martinez A.Z., « Identification of induction machines using artificial neural networks », ISIE'97 - Guimarges, Portugal.
35. Morand F., « Techniques d'observation sans capteur de vitesse en vue de la commande des machines asynchrones », Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, (2005),



36. Sasao T, Kubota H « Stability Analysis of Sensor-less Induction Motor Drives with Stator Resistance Adaptation using Estimation Error Index », Proceeding of International Conference on Electrical Machines and Systems 2007.
37. Opderbecke J. et Durieu C., « Une application de Filtrage de Kalman à la fusion de données pour la localisation d'un robot mobile », Quatorzieme colloque GRETSI - JUAN-LES-PINS, (Septembre 1993),
38. Chenchen W. and Yongdong L., « A Novel Speed Sensorless Field-Oriented Control Scheme of IM Using Extended Kalman Filter with Load Torque Observer », IEEE 2008
39. Chen G. and Guo L., « The FPGA Implementation Of Kalman Filter », Proceedings of the 5th WSEAS Int, Malta, (September 2005),
40. Pasricha R. and Sharma S., « An FPGA-Based Design of Fixed-Point Kalman Filter », DSP Journal, Volume 9, Issue 1, June, 2009
41. Maamoun M, « Nouvelles architectures d'interfaçage graphique dans les systèmes à microprocesseur », Thèse de Doctorat, ENP, Alger.
42. Ghazzi F., « Optimisation d'une bibliothèque de modules matériels de traitement d'images. Conception et test VHDL, implémentation sous forme fpga », Thèse de Doctorat, université de Bordeaux I, (2003),
43. Mark B, « Complete Digital Design», McGraw-Hill Companies, Inc., USA, (2003), 460p
44. Chekired F., « Etude et implémentation d'une commande MPPT neurofloue sur FPGA », Thèse de Magister, ENP, Alger, (2008),
45. Merabet A., « Commande non linéaire à modèle prédictif pour une machine asynchrone », Thèse de doctorat, université du Québec, Canada, (2007),

46. Bacha K., Henao H., Gossa M. and Capolino G.A., « Induction machine fault detection using stray flux EMF measurement and neural network-based decision ». Electric Power Systems Research, (2007),
47. Abdelhadi B., Benoudjit A., and Nait-Said N., « Application of Genetic Algorithm With a Novel Adaptive Scheme for the Identification of Induction Machine Parameters», IEEE Transactions on energy conversion, VOL. 20, NO. 2, (JUNE 2005),
48. Leonard A. McGee and Schmidt S.F., « Discovery of the Kalman Filter as a practical tool for aerospace and industry », NASA Technical Memorandum 86847, (1985),
49. Labarrère M., Krief J. P. et Gimonet B., « Le filtrage et ses applications », Cépaduès Editions,
50. Han S., « A closed-form solution to the discrete-time Kalman filter and its applications », Systems & Control Letters 59, (2010),
51. Alazard D., « Introduction au filtre de Kalman », Notes de cours, SUPAERO, France, (Janvier 2005),
52. Burns R.S., « Advanced Control Engineering », Butterworth-Heinemann, Oxford, (2001), 450p,
53. Ait Ouali Z., « Application des FPGA a la commande d'un moteur asynchrone », Thèse de Magister, université de Tizi-Ouzou, (2009),
54. « Virtex-5 Family Overview », Data Sheet (DS100, v5.0), by Xilinx, (2009),
55. « Virtex-5 FPGA », User Gide (UG190, v5.3), by Xilinx, (2010),
56. « Virtex-5 FPGA, XtremeDSP Design Considerations », User Gide (UG193, v5.3), by Xilinx, (2010),

57. Ahmed T., Kundarewich P.D., Anderson J.H., Taylor B.L., and Aggarwal R., « Architecture-Specific Packing for Virtex-5 FPGAs », FPGA'08, Monterey, California, USA, ( 2008)
58. Chapman K., « Digitally Removing a DC Offset: DSP Without Mathematics », White Paper: Xilinx FPGAs, (WP279, v1.0), (2008),