

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département : Génie électrique Et Informatique Industrielle

Mémoire de projet de fin d'études  
pour l'obtention du diplôme d'ingénieur d'état en Électronique  
Spécialité  
Systèmes Embarqués

**Conception d'un Actionneur électromécanique(EMA) pour la  
commande de vol des manœuvres d'avion**

Réalisé par

***BOUIFROU Souhila***  
***ABDELMALEK Dounia***

Présentée et soutenue publiquement le (04/07/2023)

**Composition du Jury :**

|              |                      |       |      |
|--------------|----------------------|-------|------|
| Promoteur    | M .Reda DERMOUCHE    | M.C.B | ENST |
| Co-Promoteur | M .Toufik BOUDIAR    | M.C.A | ENST |
| Président    | Mme Khadidja ZELLAT  | M.C.B | ENST |
| Examineur    | Mme.Djamila BENDOUDA | M.C.B | ENST |
| Examineur    | M .Samir KHELOUAT    | M.C.B | ENST |

---

# DÉDICACE

À tous ceux qui m'ont soutenu et encouragé tout au long de ce parcours académique, à tous ceux qui ont été présents à mes côtés, je dédie ce mémoire de fin d'étude. Je tiens également à exprimer ma gratitude envers mes parents, mes frères et sœurs, ainsi que mes amis, pour leur amour et leur soutien.

- ABDELMALEK Dounia

---

# DÉDICACE

Je dédie ce travail à :

A mes chers parents.  
A mes frères et soeurs.  
A toute ma famille.  
A tout mes amis.

Pour leurs affection et leurs sacrifices afin que rien n'entrave le déroulement de mes études.

**- BOUIFROU SOUHILA**

“Etudie, non pour savoir plus, mais pour savoir mieux”.

**De Alain, Les idées et les âges.**

---

# REMERCIEMENT

Nous tenons tout d'abord à remercier Dieu le tout-puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Nos remerciements vont aux membres de jury, nous sommes honorés d'avoir eu l'occasion de présenter ce travail devant vous.

Nous souhaitons également exprimer notre reconnaissance envers nos encadrants, **M. DERMOUCHE Reda.** et **M. BOUDIAR Toufik.**, qui nous ont guidés avec bienveillance tout au long de ce projet. Leur expertise, leurs conseils avisés et leur disponibilité ont été essentiels pour orienter nos travaux et les mener à bien.

Enfin, nous sommes reconnaissants envers nos professeurs pour les enseignements que nous avons pu en tirer, et envers notre institution d'enseignement qui nous a offert les ressources nécessaires, un environnement propice à l'apprentissage et des opportunités de croissance intellectuelle.

Veillez accepter nos remerciements les plus sincères et notre reconnaissance éternelle.

Cordialement.



---

# Table des matières

## Introduction Générale

|           |   |           |
|-----------|---|-----------|
| <b>I</b>  | <b>État de l’art sur la commande de vol</b>                             | <b>3</b>  |
| I.1       | Généralités . . . . .   | 4         |
| I.2       | Avion plus électrique . . . . .   | 6         |
| I.3       | L’électronique de puissance dans les réseaux de bord d’avion . . . . .  | 6         |
| I.3.1     | Génération et distribution électriques . . . . .                        | 7         |
| I.4       | Les Commandes de vol . . . . .  | 7         |
| I.5       | Famille de commande de vol . . . . .                                    | 9         |
| I.5.1     | Les commandes à puissance hydrauliques . . . . .                        | 9         |
| I.5.2     | Les commandes à puissance électrique . . . . .                          | 9         |
| I.5.2.1   | Les actionneurs électromécaniques(EMA) . . . . .                        | 10        |
| I.6       | La différence entre la commande hydraulique et électrique . . . . .     | 11        |
| I.7       | Chaîne de variateur de vitesse . . . . .                                | 12        |
| I.7.1     | Schéma fonctionnel d’un EMA . . . . .                                   | 12        |
| I.7.1.1   | Les convertisseurs dans le variateur de vitesse . . . . .               | 13        |
| I.7.1.2   | Moteur MSAP . . . . .   | 13        |
| I.8       | Fiabilité des commandes de vol . . . . .                                | 13        |
| <b>II</b> | <b>Modélisation de moteur MSAP et l’onduleur</b>                        | <b>15</b> |
| II.1      | Machine synchrone à aimant permanent . . . . .                          | 16        |
| II.1.1    | Description . . . . .   | 16        |
| II.1.2    | Principe et mode de fonctionnement . . . . .                            | 17        |
| II.1.3    | Avantages de la MSAP[1] . . . . .                                       | 17        |
| II.1.4    | Domaine d’application[2][3] . . . . .                                   | 18        |
| II.2      | Modélisation de la MSAP . . . . .                                       | 18        |
| II.2.1    | Transformation de Park . . . . .  | 20        |
| II.3      | Autopilotage et la commande vectorielle . . . . .                       | 22        |
| II.3.1    | Autopilotage de la machine synchrone à aimants permanents [4] . . . . . | 22        |
| II.3.1.1  | Régulation de courant . . . . .   | 23        |

|            |   |           |
|------------|---|-----------|
| II.3.1.2   | Régulation de vitesse . . . . .   | 24        |
| II.3.2     | la commande vectorielle . . . . .                                       | 24        |
| II.3.3     | principe de fonctionnement . . . . .                                    | 25        |
| II.3.4     | Avantages de la commande vectorielle : . . . . .                        | 25        |
| II.3.5     | Application de la commande vectorielle dans l'avion . . . . .           | 26        |
| II.4       | Synoptique de la chaîne de traction . . . . .                           | 26        |
| II.5       | Onduleur . . . . .  | 27        |
| II.5.1     | Présentation . . . . .  | 27        |
| II.5.2     | Alimentation à partir d'un onduleur de tension . . . . .                | 27        |
| II.5.3     | Modélisation de l'onduleur triphasé de tension à deux niveaux . . . . . | 28        |
| II.5.4     | Différents types de commandes de commutation . . . . .                  | 29        |
| II.5.4.1   | La modulation de largeur d'impulsion (PWM) . . . . .                    | 29        |
| <b>III</b> | <b>Etude de la carte DSP TMS320F28335</b>                               | <b>31</b> |
| III.1      | Présentation de DSP . . . . .   | 32        |
| III.2      | Architecture de DSP . . . . .   | 32        |
| III.3      | Avantage des DSPs . . . . .   | 33        |
| III.4      | Les Caractéristiques de TMS320F28335 . . . . .                          | 33        |
| III.5      | Architecture de TMS320F28335 . . . . .                                  | 34        |
| III.5.1    | Diagramme fonctionnel . . . . .   | 34        |
| III.5.2    | Diagramme des broches . . . . .   | 36        |
| III.5.3    | L'unité Centrale de Traitement (CPU) . . . . .                          | 37        |
| III.5.4    | Les mémoires de DSP . . . . .   | 38        |
| III.5.5    | Les périphériques de DSP . . . . .                                      | 38        |
| III.5.5.1  | Module convertisseur analogique digital (ADC) . . . . .                 | 39        |
| III.5.5.2  | Modulation de largeur d'impulsion PWM . . . . .                         | 39        |
| III.5.5.3  | Les Entrées/Sorties à usage général (GPIO) . . . . .                    | 40        |
| III.5.5.4  | Module d'interface série asynchrone (SCI) . . . . .                     | 40        |
| III.5.5.5  | Le Périphérique de capture amélioré (eCAP) . . . . .                    | 41        |
| III.5.5.6  | Codeur à quadrature d'impulsion (QEP) . . . . .                         | 41        |
| III.5.5.7  | Temporisateurs CPU . . . . .  | 41        |
| III.5.5.8  | Temps mort Dead-band(DB) . . . . .                                      | 41        |
| III.5.5.9  | Watchdog . . . . .  | 41        |
| III.5.6    | Trames périphériques 0, 1, 2, 3 (PFn) . . . . .                         | 42        |
| III.6      | Arithmétique de TMS320F28335 . . . . .                                  | 42        |
| III.6.1    | Les DSP à virgule fixe . . . . .  | 43        |
| III.6.2    | Les DSP à virgule flottante . . . . .                                   | 43        |
| <b>IV</b>  | <b>Étude des principaux modules pour la commande</b>                    | <b>44</b> |
| IV.1       | Module ePWM . . . . .   | 45        |
| IV.1.1     | Généralités [5] . . . . .   | 45        |
| IV.1.2     | les modules de EPWM . . . . .   | 45        |
| IV.1.3     | Time-base (TB) . . . . .  | 46        |
| IV.1.4     | Counter-compare (CC) . . . . .  | 48        |

|          |   |           |
|----------|---|-----------|
| IV.1.5   | The Action-Qualifier (AQ)   | 49        |
| IV.1.6   | Dead-Band Generator (DB)  | 51        |
| IV.1.7   | PWM Chopper Control (PCCTL)   | 54        |
| IV.1.8   | trip zone   | 56        |
| IV.1.8.1 | TZ1 to TZ6 (Trip Zones 1 to 6)  | 56        |
| IV.1.8.2 | EPWMxTZINT (ePWMx Trip Zone Interrupt)                                      | 57        |
| IV.1.8.3 | EPWMxBO (PWMxB Output)  | 57        |
| IV.1.8.4 | Event Trigger   | 57        |
| IV.1.8.5 | HRPWM (High Resolution PWM)   | 58        |
| IV.2     | Exemple de code EPWM sur ccs  | 58        |
| IV.3     | Module eQEP   | 60        |
| IV.3.1   | Généralités   | 60        |
| IV.3.2   | Définition d'eQEP (Enhanced Quadrature Encoder Pulse)                       | 62        |
| IV.3.3   | Description   | 62        |
| IV.3.3.1 | Les Entrées de eQEP   | 62        |
| IV.3.3.2 | Description fonctionnelle   | 63        |
| IV.3.4   | Quadrature Decoder Unit (QDU)   | 64        |
| IV.3.4.1 | Modes d'entrée du compteur de position                                      | 65        |
| IV.3.4.2 | Sortie Synchronisation Position-Compare                                     | 67        |
| IV.3.5   | Compteur de position/Unité de contrôle et de comptage de la position (PCCU) | 67        |
| IV.3.5.1 | Modes de fonctionnement du compteur de position                             | 68        |
| IV.3.5.2 | Unité de comparaison de positions eQEP                                      | 68        |
| IV.3.6   | Unité de capture des bords eQEP   | 71        |
| IV.3.7   | eQEP Watchdog   | 72        |
| IV.3.8   | Unité Base de la minuterie  | 73        |
| IV.3.9   | Structure d'interruption eQEP   | 74        |
| IV.4     | Code CCS pour la configuration de module QEP                                | 75        |
| IV.5     | ADC   | 76        |
| IV.6     | Généralités   | 76        |
| IV.7     | Les registres de l'ADC  | 78        |
| IV.8     | code ccs pour la configuration de ADC                                       | 80        |
| <b>V</b> | <b>implémentations software et hardware</b>                                 | <b>83</b> |
| V.1      | Méthodes de validation et de test   | 83        |
| V.1.1    | Software in the loop  | 84        |
| V.1.2    | Process in the loop   | 84        |
| V.1.3    | Hardware in the loop  | 84        |
| V.2      | Hardware in the loop et Matlab  | 85        |
| V.3      | Implémentation software   | 87        |
| V.3.1    | epwm simulink   | 87        |
| V.3.2    | Programme Matlab de Projet  | 88        |
| V.3.2.1  | Bloc serial receive   | 90        |
| V.3.2.2  | Bloc de contrôle de vitesse   | 90        |

|                            |  |            |
|----------------------------|--|------------|
| V.3.2.3                    | Bloc de contrôle de courant . . . . .                | 91         |
| V.3.2.4                    | Le bloc de Système de contrôle . . . . .             | 97         |
| V.3.2.5                    | Le bloc de data-logging . . . . .                    | 98         |
| V.3.2.6                    | Output Scaling . . . . .                             | 98         |
| V.4                        | Bloc de l'onduleur . . . . .                         | 98         |
| V.5                        | Bloc de simulation d'onduleur et de moteur . . . . . | 99         |
| V.6                        | Résultats dans data inspector . . . . .              | 100        |
| <b>Conclusion Générale</b> |  | <b>104</b> |
| V.7                        | EPWM . . . . .                                       | 108        |
| V.7.1                      | Active et shadow register . . . . .                  | 108        |
| V.7.2                      | Les registres de epwm . . . . .                      | 108        |
| V.7.3                      | les signaux . . . . .                                | 110        |
| V.8                        | EQEP . . . . .                                       | 111        |
| V.8.1                      | Les registres de module QEP . . . . .                | 111        |
| V.8.2                      | les registres utilises dans le code ccs . . . . .    | 113        |

---

# TABLE DES FIGURES

|       |   |    |
|-------|---|----|
| I.1   | Composantes d'un avion [6]  | 5  |
| I.2   | Les gouvernes de vol [7]  | 8  |
| I.3   | les axes de l'avion [7]   | 8  |
| I.4   | schéma de principe d'une commande de vol électrique [8]                         | 9  |
| I.5   | Schéma de principe d'un EMA [9]   | 10 |
| I.6   | schéma fonctionnel de variateur de vitesse[4]                                   | 12 |
| I.7   | Onduleur triphasé[4]  | 13 |
| II.1  | Machine à aimants permanents[10]  | 16 |
| II.2  | composition d'un MSAP   | 17 |
| II.3  | Représentation d'une machine synchrone à aimants permanents sur ses 3 axes [11] | 19 |
| II.4  | Paramètres réglables du modèle de MSAP[4]                                       | 20 |
| II.5  | Présentation de MSAP sur les axes fictives[12]                                  | 21 |
| II.6  | Schéma fonctionnel du modèle de Park  | 22 |
| II.7  | schéma bloc de la boucle de régulation de courant [4]                           | 23 |
| II.8  | schéma bloc de la boucle de régulation de vitesse [4]                           | 24 |
| II.9  | commande vectorielle d'un moteur triphasé[13]                                   | 25 |
| II.10 | Schéma Global de L'association MSAP- Convertisseur                              | 26 |
| II.11 | Principe de fonctionnement d'un onduleur  | 27 |
| II.12 | Schéma de l'association MSAP et l'onduleur triphasé de tension                  | 28 |
| II.13 | Schéma synoptique de la M.L.I[14]   | 30 |
| III.1 | L'architecture de Harvard modifié[15]   | 32 |
| III.2 | Image réelle d'une carte DSP F28335   | 34 |
| III.3 | Le schéma fonctionnel du DSP TM320F28335 [16]                                   | 35 |
| III.4 | Diagramme des broches de la carte DSP[16]                                       | 37 |
| III.5 | Les interconnexions des signaux avec l'ePWM[16]                                 | 40 |
| IV.1  | les modules de EPWM[5]  | 45 |
| IV.2  | time base TB [5]  | 46 |

---

|       |   |    |
|-------|---|----|
| IV.3  | Architecture interne de TB[5]   | 47 |
| IV.4  | chronogramme de Time-Base Up-Count Mode [5]   | 47 |
| IV.5  | counter compare[5]  | 48 |
| IV.6  | Architecture interne de CC[5]   | 48 |
| IV.7  | Counter-compare Up-Count Mode Waveforms[5]  | 49 |
| IV.8  | AQ[5]   | 50 |
| IV.9  | Possible AQ Actions for EPWMxA and EPWMxB Outputs [5]   | 50 |
| IV.10 | dead band (DB)[5]   | 51 |
| IV.11 | Configuration Options for the Dead-Band Submodule [5]   | 52 |
| IV.12 | Dead-Band Waveforms for Typical Cases (0% < Duty < 100%) [5]                                    | 53 |
| IV.13 | PWM chopper(PC)[5]  | 54 |
| IV.14 | chronogramme du PWM-Chopper uniquement l'action de hachage [5]                                  | 54 |
| IV.15 | chronogramme du pwm-chopper [5]   | 55 |
| IV.16 | le module trip zone[5]  | 56 |
| IV.17 | Event Trigger[5]  | 57 |
| IV.18 | Event Inputs and Prescaled Outputs [5]  | 58 |
| IV.19 | Le Disque optique d'encodeur[17]  | 60 |
| IV.20 | Le signal de sortie du codeur QEP pour le mouvement avant/arrière[17]                           | 61 |
| IV.21 | Exemple d'impulsion d'index[17]   | 61 |
| IV.22 | Schéma fonctionnel du module eQEP   | 64 |
| IV.23 | Diagramme fonctionnel de l'unité décodeur[16]   | 65 |
| IV.24 | Décodage de l'horloge en quadrature et de la direction[17]                                      | 66 |
| IV.25 | Unité de comparaison de positions eQEP[17]  | 69 |
| IV.26 | eQEP Position-compare Event Generation Points[17]   | 70 |
| IV.27 | eQEP Position-compare Sync Output Pulse Stretcher[17]   | 70 |
| IV.28 | Unité de capture des bords eQEP[17]   | 71 |
| IV.29 | Événement de position unitaire pour la mesure de la vitesse lente (QCAPCTL[UPPS]<br>= 0010)[17] | 72 |
| IV.30 | Minuterie du chien de garde de l'eQEP[17]   | 73 |
| IV.31 | Base de temps de l'unité eQEP[17]   | 74 |
| IV.32 | Génération d'interruption EQEP[17]  | 75 |
| IV.33 | Schéma fonctionnel du module ADC  | 77 |
| IV.34 | Les Registres du module ADC   | 79 |
| V.1   | blocs de c28335   | 86 |
| V.2   | blocs des outils électrique   | 86 |
| V.3   | exemple epwm  | 87 |
| V.4   | paramètres epwm   | 88 |
| V.5   | epwm oscilloscope   | 88 |
| V.6   | Schéma Global de L'association MSAP-Convertisseur   | 89 |
| V.7   | programme Matlab  | 89 |
| V.8   | Serial Receive  | 90 |
| V.9   | speed control   | 91 |
| V.10  | current control   | 92 |

|   |     |
|---|-----|
| V.11 input scaling . . . . .                                      | 92  |
| V.12 bloc de calculate speed . . . . .                            | 93  |
| V.13 Paramétrage de Quadrature . . . . .                          | 94  |
| V.14 paramétrage de speed measurement . . . . .                   | 95  |
| V.15 paramétrage de conversion mécanique à l'électrique . . . . . | 96  |
| V.16 Bloc de mesure de courant de phase . . . . .                 | 97  |
| V.17 Bloc de système de contrôle . . . . .                        | 97  |
| V.18 Bloc de Output Scaling . . . . .                             | 98  |
| V.19 Bloc de l'onduleur . . . . .                                 | 99  |
| V.20 simulation d'un onduleur et d'un moteur . . . . .            | 99  |
| V.21 les courbes . . . . .  | 100 |
| V.22 iq_ref et id_fb . . . . .                                    | 101 |
| V.23 IQ_FB . . . . .  | 101 |
| V.24 speed_ref_pu :1 . . . . .                                    | 102 |
| V.25 speed_ref . . . . .  | 102 |
| V.26 Les registres de QEP . . . . .                               | 113 |
| V.27 Registre de contrôle du décodeur QEP (QDECCTL) . . . . .     | 113 |
| V.28 Registre de contrôle eQEP (QEPCTL) . . . . .                 | 114 |

---

# LISTE DES TABLEAUX

|     |   |    |
|-----|---|----|
| I.1 | La différence entre la commande hydraulique et électrique . . . . . | 11 |
|-----|---|----|



---

# LISTE DES ABRÉVIATIONS

- DSP :Digital signal process.
- MSAP : Moteur synchrone à aimant permanent.
- ADC :Convertisseur analogique numérique.
- GPIO :Entée/Sortie à usage général.
- QEP :Périphérique de positionnement.
- CAP :Périphérique de capture amélioré.
- SCI :Interface de communication série synchrone.
- CPU :Unité centrale de traitement.
- EMA :Actionneur électromécanique.
- AC :Courant alternatif.
- DC :Courant continu.
- FOC :Commande de champ orienté.
- MLI :Modulation par largeur d'impulsion.
- PI :Correcteur Proportionnel Intégral
- f.é.m : Force électromotrice.
- d,q :Composantes de Park directes et quadrature.
- a,b,c :Indices correspondants aux trois phases du rotor.
- $V_s$  :La matrice des tensions par phase au stator.
- $I_s$  : La matrice des courants au stator.
- $\Phi_s$  :Vecteur de flux statorique.
- $R_s$  :Matrice résistance du stator.
- $L_s$  : Inductance cyclique propre statorique.
- $\theta$  :Angle entre l'axe d et l'axe de référence dans le système triphasé.
- $\omega$  :La pulsation électrique.
- P :Le nombre de pair de pôles de la machine.
- $W_r$  :La vitesse rotorique de rotation de la machine.
- $L_p$  :Inductance propre d'une phase statorique.
- $f_p$  :Inductance de fuite.
- J : Le moment d'inertie du moteur.
- F :Le coefficient de frottement visqueux.
- $C_{em}$  :Le couple électromagnétique.

- $C_r$  : Le couple résistant ou de charge.
- $f$  : Vecteur flux créé par à travers l'enroulement.
- $I_{dq0}$  : Les composants directs en quadrature et homopolaire du courant.
- $V_{dq0}$  : Les composants directs en quadrature et homopolaire de la tension.
- $\Phi_{dq0}$  : Les composants directs en quadrature et homopolaire du flux.
- $V_d, V_q$  : Tension directe et quadratique.
- $\Phi_d, \Phi_q$  : Flux direct et quadratique.
- FTBO : Fonction de transfert en boucle ouverte.
- FTBF : Fonction de transfert en boucle fermée.
- $I_d, I_q$  : Courant direct et quadratique.
- $K_p, K_i$  : Le gain proportionnel et intégral

**Les autres symboles sont explicités dans le texte.**

---

# INTRODUCTION GÉNÉRALE

L'émergence des nouvelles générations d'avions civils et militaires a conduit à l'électrification des fonctions de commande de vol, englobant les interfaces homme-machine et les actionneurs de puissance. Ainsi, l'utilisation d'actionneurs purement électromécaniques (EMA) est devenue essentielle pour se débarrasser des inconvénients de l'énergie hydraulique. Parallèlement, l'électronique de puissance a connu des avancées notables, améliorant les systèmes de distribution électrique, les moteurs électriques, les générateurs, etc. Les progrès dans les calculateurs numériques ont également permis des systèmes plus performants en termes de puissance de calcul, de fiabilité et de sécurité. Ces développements dans l'électronique de puissance et les calculateurs numériques ont contribué à des avancées significatives dans l'aéronef.

L'évolution des technologies aéronautiques a conduit au remplacement progressif des actionneurs hydrauliques traditionnels par des EMA plus avancés. Ces derniers offrent des avantages considérables en termes de fiabilité, de poids, d'efficacité énergétique et de réactivité, tout en réduisant la complexité des systèmes et les risques de fuites hydrauliques. En outre, les EMA permettent une intégration plus étroite avec les systèmes de commande numériques, favorisant ainsi une meilleure optimisation des performances de vol.

Dans cette perspective, cette étude propose de concevoir et de commander un actionneur électromécanique pour la commande d'un aile d'avion. Nous aborderons les principaux aspects de cette conception, y compris la sélection des composants, la modélisation et la simulation. L'objectif ultime est de commander un EMA fiable, précis et efficace, capable de répondre aux exigences rigoureuses des systèmes de commande de vol modernes.

Notre objectif consiste à utiliser une carte de commande basée sur un DSP (Digital Signal Processor) pour contrôler un EMA (Electromechanical Actuator) dans le but de gérer les manœuvres d'un avion. L'EMA est un système électromécanique utilisé pour activer les surfaces de contrôle de l'avion, telles que les ailerons, les gouvernes de profondeur et les gouvernes de direction.

Le premier chapitre de cette étude propose un état de l'art sur le domaine aéronautique et les commandes de vol en général. Nous explorerons les principaux concepts et technologies utilisés dans les avions modernes, tels que les commandes de vol électriques (ECS) et les systèmes de commande numérique (Fly-by-Wire). Nous mettrons également en évidence les défis et les exigences auxquels sont confrontés les concepteurs d'actionneurs électromécaniques pour les aéronefs.

Le deuxième chapitre se concentrera sur le moteur synchrone à aimants permanents et sa modélisation avec l'alimentation de ce dernier avec l'onduleur triphasé. Nous examinerons le moteur électrique, son comportement électromécanique et son modèle mathématique pour prédire et optimiser les performances de l'actionneur. De plus, nous aborderons l'onduleur, et la commande vectorielle utilisée pour la commande de moteur.

Dans le troisième chapitre, nous nous pencherons sur la conception de la carte de commande du moteur numérique signal-process (DSP). Cette carte est responsable de la gestion et de la coordination des signaux électriques pour contrôler précisément l'actionneur. Nous explorerons les algorithmes de commande et les différents blocs utilisés pour garantir un fonctionnement efficace et sécurisé de l'EMA.

Le dernier chapitre de cette étude se concentre sur l'implémentation software et hardware de l'actionneur électromécanique (EMA) pour la commande de vol d'un avion. Nous aborderons l'utilisation du langage de programmation MATLAB et le programme spécifique utilisé pour développer et contrôler l'EMA.

---

---

# CHAPITRE I

---

## ÉTAT DE L'ART SUR LA COMMANDE DE VOL

### Introduction

La commande de vol, également connue sous le nom de pilotage, est un aspect essentiel de l'aviation moderne. Elle fait référence aux divers moyens par lesquels un pilote contrôle un aéronef en vol. Que ce soit un avion, un hélicoptère ou même un drone, la commande de vol joue un rôle important dans la manipulation et la manœuvre de l'aéronef pour assurer un vol sûr et efficace.

Dans ce premier chapitre, nous introduirons les concepts fondamentaux de l'aéronautique, y compris l'histoire de l'aviation, les principes de vol, les types d'aéronefs et la commande de vol.

Ensuite nous concentrons sur la tendance croissante vers des avions plus électriques. Nous discuterons des avantages de l'électrification dans l'aviation, notamment en termes de réduction de poids, d'efficacité énergétique et de réduction des émissions.

Après nous nous pencherons sur le rôle de l'électronique de puissance dans le réseau de bord des avions. La partie qui suit sera consacré aux commandes de vol, qui sont essentielles pour le contrôle de l'aéronef. Nous examinerons les différentes familles de commandes de vol, en mettant l'accent sur les commandes à puissance hydraulique et les commandes à puissance électrique. Nous discuterons des avantages, des inconvénients et des caractéristiques de chaque système.

Nous étudierons également les actionneurs électromécaniques, ainsi que les actionneurs électro-hydrostatiques qui combinent l'énergie électrique et hydraulique pour générer le mouvement des surfaces de contrôle.

Nous aborderons en général la chaîne de variateur de vitesse avec ses différents blocs de

conversion, et l'actionneur EMA qui seront exploités dans le reste de projet.

Enfin, nous aborderons l'importance de la sûreté, de la sécurité de fonctionnement et de la fiabilité des commandes de vol.

## I.1 Généralités

L'aéronautique est un domaine qui englobe l'étude, la conception, la fabrication et l'exploitation des aéronefs, y compris les avions, qui est un moyen de transport aérien plus lourd que l'air, qui utilise des forces aérodynamiques pour se maintenir en vol.

L'aéronautique englobe différents types d'aéronefs, tels que les avions, les hélicoptères, les planeurs et les drones. Chaque type d'aéronef présente des caractéristiques spécifiques et est conçu pour des missions particulières, comme le transport de passagers, le fret, la recherche, la surveillance ou les opérations militaires.

L'histoire de l'aviation remonte aux premiers pionniers tels que les frères Wright et leur premier vol motorisé en 1903. Depuis lors, l'aviation a connu une évolution rapide, avec des avancées technologiques significatives dans les domaines des moteurs, des matériaux, de la navigation et des systèmes de contrôle.

Les principes de vol expliquent les forces qui permettent à un avion de décoller, de rester en vol et de se mouvoir dans différentes directions. Ces principes incluent la portance, la traînée, le poids et la poussée, ainsi que les différentes forces qui agissent sur un aéronef en mouvement.

La commande de vol est un aspect important de l'aviation, qui concerne les systèmes et les dispositifs utilisés pour contrôler les mouvements d'un avion. Elle permet de manipuler les surfaces de contrôle, telles que les gouvernes (ailerons, gouvernail de direction, et gouvernail de profondeur), les volets et les bords d'attaque, afin d'assurer la stabilité, la manœuvrabilité et la sécurité en vol.

## COMPOSANTES D'UN AVION

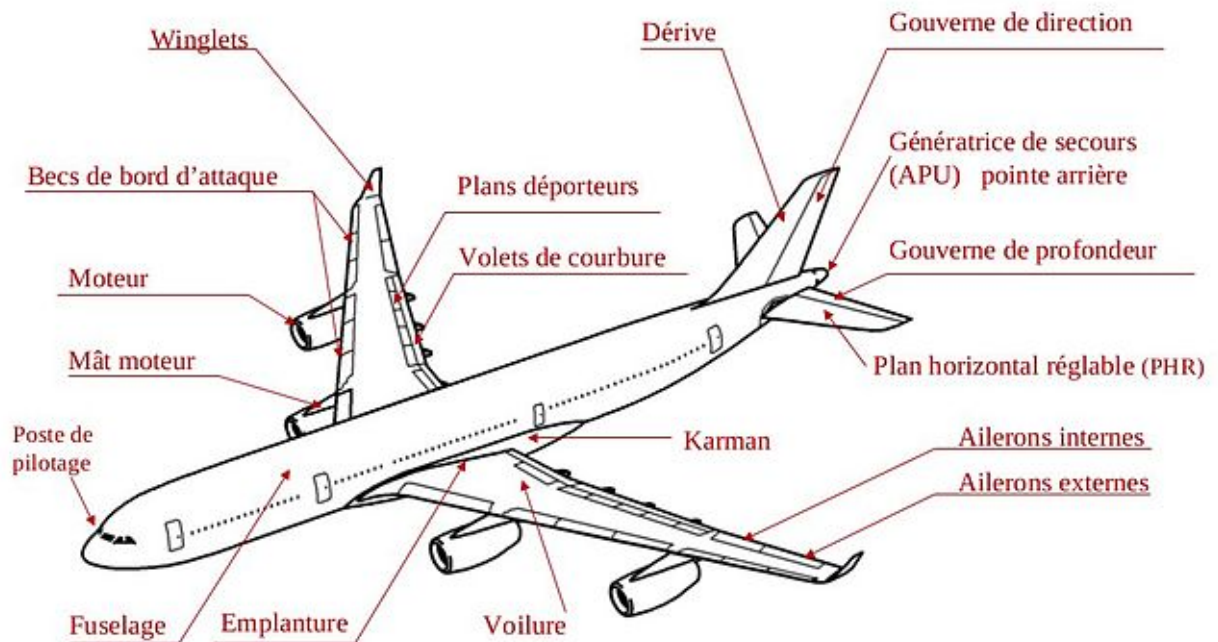


FIGURE I.1 – Composantes d'un avion [6]

Les systèmes de commande de vol sont généralement constitués de capteurs, de calculateurs et d'actionneurs. Les capteurs mesurent les paramètres essentiels tels que l'attitude, la vitesse, l'accélération et les mouvements de l'avion. Les calculateurs traitent ces données pour déterminer les commandes nécessaires afin d'obtenir les mouvements souhaités. Les actionneurs, tels que les servomoteurs ou les vérins hydrauliques, convertissent les signaux de commande en mouvements physiques des surfaces de contrôle.

Les avancées technologiques ont conduit au développement de systèmes de commande de vol de plus en plus sophistiqués, tels que les systèmes de commande de vol électriques (EFCS - Electric Flight Control Systems) qui utilisent des actionneurs électriques plutôt que hydrauliques. Ces systèmes offrent de nombreux avantages, notamment une meilleure efficacité, une réduction du poids et des coûts de maintenance, ainsi qu'une intégration plus facile avec les autres systèmes de l'avion.

La commande de vol joue un rôle essentiel dans la sécurité et les performances des aéronefs, et elle fait l'objet de normes et de réglementations strictes pour garantir le bon fonctionnement des avions. Les pilotes sont formés pour comprendre et utiliser les systèmes de commande de vol afin de maintenir le contrôle de l'avion dans différentes phases de vol, des décollages et atterrissages aux manœuvres en vol.

## I.2 Avion plus électrique

L'avion plus électrique est une évolution technologique visant à augmenter l'utilisation de l'énergie électrique dans les systèmes d'un avion. Cette transition permet d'optimiser les performances, de réduire les émissions et le bruit, tout en offrant une flexibilité accrue dans la conception des systèmes. L'électrification des fonctions de commande de vol, des systèmes de refroidissement et d'autres sous-systèmes contribue à cette évolution vers des solutions durables dans l'industrie aéronautique.

Le développement d'avions "plus électriques" offre une plus grande flexibilité dans l'installation des systèmes, grâce à la souplesse du câblage électrique par rapport aux systèmes hydrauliques et pneumatiques. Cela permet aux constructeurs et aux clients d'aménager l'avion selon leurs besoins spécifiques. De plus, cette transition pourrait conduire à un gain de masse global en remplaçant les éléments lourds et volumineux des systèmes hydrauliques et pneumatiques, tout en améliorant la compacité grâce à l'intégration des convertisseurs électriques.[4]

En particulier, la forte émergence de l'électronique de puissance est une conséquence de cette évolution technologique et constitue un point majeur pour l'obtention de solutions performantes, compétitives et fiables.[4]

## I.3 L'électronique de puissance dans les réseaux de bord d'avion

Dans les dernières générations d'avions commerciaux, la présence de convertisseurs statiques au sein des réseaux de bord était limitée à une quantité réduite d'applications : alimentation de charges commerciales (galleys), redresseurs (115V-400Hz/28V) et onduleurs de secours (28V/115V-400Hz) du réseau de distribution électrique.

Les réseaux de bord d'avion sont des systèmes électriques complexes qui fournissent l'alimentation électrique à divers équipements et systèmes à bord de l'avion.

L'électronique de puissance est utilisée pour convertir la tension et la fréquence de l'alimentation électrique provenant des générateurs d'avion en niveaux appropriés pour les différents systèmes embarqués. Cela peut inclure des convertisseurs AC/DC, DC/DC et DC/AC pour adapter l'énergie électrique aux besoins spécifiques des équipements.

Les réseaux de bord d'avion utilisent des systèmes de distribution d'énergie basse tension (généralement 28 VDC) pour alimenter les systèmes essentiels tels que l'éclairage, les commandes de vol, les communications, les systèmes de divertissement des passagers...etc.

L'électronique de puissance joue un rôle important dans les avions, permettant la commutation automatique vers des sources d'alimentation de secours en cas de défaillance du système principal et assurant une gestion efficace de la charge électrique pour répondre aux besoins changeants des systèmes embarqués. De plus, l'utilisation croissante de systèmes de



stockage d'énergie tels que les batteries lithium-ion permet des applications telles que le démarrage des moteurs et la réduction de la consommation de carburant, avec l'électronique de puissance assurant la gestion et la conversion bidirectionnelle de l'énergie entre ces systèmes de stockage et les autres systèmes à bord de l'avion.

### I.3.1 Génération et distribution électriques

**ATA 24** : La section ATA 24 concerne les systèmes de commande de vol, y compris les commandes manuelles, les actionneurs, les surfaces de contrôle, les indicateurs de commande de vol, les systèmes de retour d'effort, les commandes de trim...etc. Elle englobe tous les éléments nécessaires à la commande et à la manipulation directe de l'avion par les pilotes.

**ATA 27** : La section ATA 27 se concentre sur les systèmes de commande de vol automatiques. Elle englobe les commandes de vol et ses calculateurs automatiques, les actionneurs, les capteurs et les systèmes de gestion de l'assiette du roulis et de l'assiette en lacet. Elle concerne les systèmes qui aident les pilotes à maintenir une trajectoire précise, à stabiliser l'avion et à effectuer des manœuvres complexes.

## I.4 Les Commandes de vol

Initialement (dans les premiers avions), certains services comme la commande de vol par exemple étaient réalisés mécaniquement par le pilote qui manœuvrait au moyen de câble, tringles et bielles [9, 18]. Ce type de technologie, qualifiée de "technologie mécanique" convenait aux aéronefs de petite taille. [19]

Actuellement, pour faciliter les manœuvres dans certains gros avions, l'entrée mécanique de la commande est maintenue mais la source de puissance est maintenant hydraulique. Les actionneurs ainsi constitués sont dits « hydromécaniques ». [19]

Pour réduire sa masse, et faciliter la commande, l'entrée mécanique d'un actionneur hydromécanique peut être remplacée par son équivalent électrique. Dans le cas de la commande de vol, cette technologie est dite Fly By Wire (FBW). [19]

La source de puissance peut aussi être électrique; la technologie correspondante est le Power By Wire (PBW). Ici, l'hydraulique est soit réduite (cas des actionneurs ElectroHydrostatiques ou électro Hydrostatic Actuator ou encore EHA), soit supprimée comme c'est le cas dans les actionneurs électromécaniques (électromécanique actuator ou EMA). [19]

Avec des actionneurs comme ceux de la technologie FBW ou les EHA, les avions sont "plus électriques" et si tous les actionneurs étaient du type EMA, les avions sont alors dits "tout électriques". La maintenance des avions est alors facilitée et de plus courte durée. [20, 21, 19]

Sur un avion, les commandes de vols représentent l'ensemble des équipements permettant au pilote de positionner les gouvernes et donc de contrôler la trajectoire de l'avion. [22] On distingue : [8]

- **Les commandes de vol primaires** qui sont des dispositifs destinés à faire évoluer l'avion autour des axes de référence.

● **Les commandes de vol secondaires** qui permettent un changement de la configuration aérodynamique de l'avion. Ce sont les hypersustentateurs, les aérofreins et les destructeurs de portance.

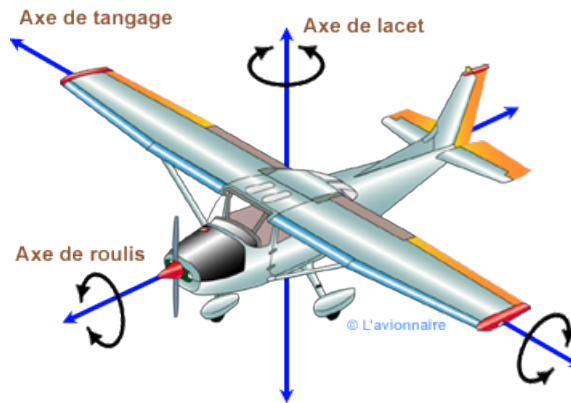


FIGURE I.2 – Les gouvernes de vol [7]



FIGURE I.3 – les axes de l'avion [7]

Pour piloter un avion, il est nécessaire de pouvoir contrôler en permanence ses évolutions dans l'espace suivant trois directions ou axes :

- Axe de roulis est l'axe longitudinal du plan reliant le bec et la queue.
- Axe de tangage. L'axe horizontal du plan qui relie les ailes.
- Axe de lacet, l'axe perpendiculaire au plan (roulis-tangage).

Pour changer la position de l'avion selon ces trois axes, le pilote peut jouer avec différents axes, appelées commandes de vol primaires.

- Ailerons et spoilers qui fonctionnent en roulis.
- Les gouvernes de profondeur qui agissent en tangage.
- La gouverne de direction qui agit en lacet.

Lors du décollage et à l'atterrissage, un autre niveau de vol est utilisé appelé commande de vol secondaire. Elle sert à modifier l'aérodynamique de l'avion.

- Les hypersustentateurs (becs et volets) agissant sur la portance.
- Spoiler agissant sur la traînée.

Le positionnement de ses différentes surfaces est assuré par des actionneurs mécaniques, hydrauliques ou électriques et à commande mécanique ou électrique appelés servocommandes. [9]

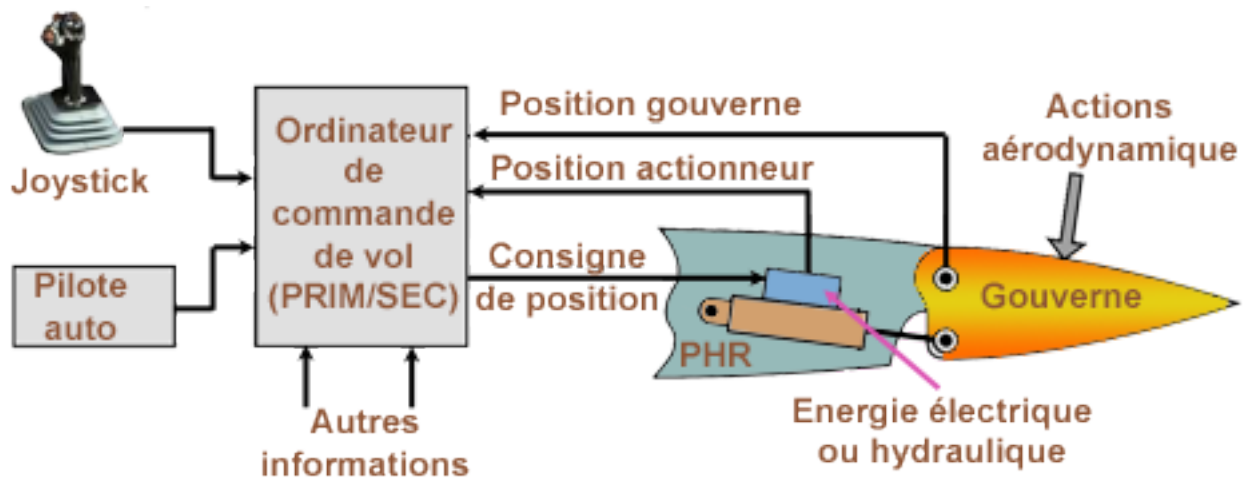


FIGURE I.4 – schéma de principe d'une commande de vol électrique [8]

## I.5 Famille de commande de vol

En se limitant à la gamme des moyens et gros porteurs, on peut différencier deux grandes familles de commandes de vol : les commandes à "puissance hydraulique" et à "puissance électrique".

### I.5.1 Les commandes à puissance hydrauliques

Certains avions utilisent la commande de vol à puissance hydraulique, où les surfaces de contrôle sont actionnées par des actionneurs hydrauliques en réponse aux commandes du pilote. Cela permet une manipulation aisée à des vitesses élevées et une meilleure perception des mouvements. Cependant, d'autres avions optent pour des commandes de vol mécaniques ou électriques en fonction de leurs besoins et de leur taille.

En résumé, la commande de vol hydraulique offre une puissance élevée et une réponse rapide, tandis que la commande de vol électrique offre une précision de contrôle accrue et une meilleure flexibilité. Le choix entre les deux dépend des exigences spécifiques de l'avion, de sa taille, de son utilisation et des considérations de poids, de fiabilité et de maintenance. De nombreux avions modernes combinent les deux systèmes pour bénéficier des avantages des deux technologies.

### I.5.2 Les commandes à puissance électrique

Les commandes de vol à puissance électrique, également connues sous le nom de fly-by-wire électrique, sont un système de commande de vol qui remplace les connexions mécaniques traditionnelles par des signaux électriques.

Dans ce système, les mouvements des commandes de vol sont convertis en signaux électriques qui sont ensuite transmis aux actionneurs électriques pour effectuer les mouvements correspondants des surfaces de contrôle de l'aéronef.

Les mouvements des commandes de vol (volant,manche à balai,palonniers) sont détectés par des capteurs et convertis en signaux électriques.Les signaux sont ensuite transmis au système de commande. Ensuite ce système de commande les traite pour générer les signaux appropriés pour les actionneurs électriques qui peuvent être des moteurs électriques,vérins électriques ou d'autres dispositifs qui convertissent l'énergie électrique en mouvement mécanique.Il peut également inclure des calculateurs de commande et des logiciels de traitement pour des fonctionnalités avancées telles que la protection contre les décrochages,la stabilisation automatique,etc.

Les commandes de vol à puissance électrique nécessitent un système électrique robuste pour fournir l'énergie nécessaire au fonctionnement des commandes,des actionneurs et des calculateurs.Ce réseau électrique peut être alimenté par des générateurs auxiliaires,des batteries,des sources d'énergie solaire ou d'autres moyens.

Cette technologie est représentée principalement par deux types d'actionneurs :les actionneurs électro-hydrauliques(EHA),les actionneurs électromécaniques (EMA) qui sont la base de notre projet.[9]

### I.5.2.1 Les actionneurs électromécaniques(EMA)

Les actionneurs électromécaniques ne possèdent pas de circuit hydraulique.La transmission de puissance se fait de manière mécanique à l'aide de réducteurs ou de vis sans fin.

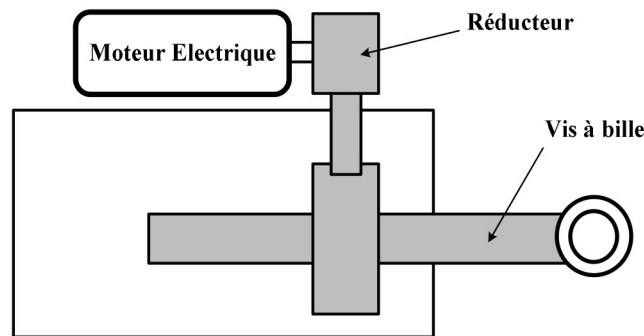


FIGURE I.5 – Schéma de principe d'un EMA [9]

Suivant les grandeurs de puissances ou de vitesses mises en jeu,les EMA peuvent être soit à attaque directe et dans ce cas le rotor du moteur électrique est lié directement au système à entraîner,soit déporté et muni d'un étage réducteur.[19]

Un moteur électrique entraîne une chaîne de transmission mécanique (boîte de vitesse, réducteur, vis à bille. . .) qui actionne à son tour la gouverne située en bout de chaîne.Dans ce principe,différentes configurations d'EMA peuvent être identifiées :[9]

- le moteur électrique est déporté,il s'agit de la technologie Gear drive.Dans ce cas,le moteur électrique est accouplé à un réducteur lui même orienté.
- le moteur entraîne directement le vérin, c'est la technologie direct drive,qui diminue les interfaces entre le moteur et le vérin et limite ainsi le risque de grippage.

En combinant les avantages d'un système de commande de vol à puissance électrique, tel que la réduction du poids, la flexibilité, le contrôle avancé et la facilité de maintenance. Ce système offre une plus grande efficacité, une meilleure manœuvrabilité et une sécurité accrue.

D'un point de vue global, les EMAs qu'ils soient déportés ou direct drive sont attractifs pour les raisons suivantes : [9]

- la suppression totale de la puissance hydraulique.
- un gain de masse au niveau de la plateforme.
- de la valeur ajoutée en termes de performance, précision et vitesse (dû à la présence de l'actionneur électrique).
- une réduction du coût de la maintenance.

## I.6 La différence entre la commande hydraulique et électrique

Le tableau suivant représente une comparaison entre les familles de commandes de vol en terme de performances, fiabilité et fonctionnement.

TABLE I.1 – La différence entre la commande hydraulique et électrique

| type           | commande hydraulique  | commande électrique   |
|----------------|---|---|
| Fonctionnement | Utilise l'énergie hydraulique pour déplacer les surfaces de contrôle à l'aide d'actionneurs hydrauliques. L'huile sous pression est utilisée pour générer des forces élevées et déplacer les surfaces de contrôle de manière rapide et précise.                     | Utilise l'énergie électrique pour déplacer les surfaces de contrôle à l'aide d'actionneurs électriques. Les moteurs électriques fournissent la puissance nécessaire pour déplacer les surfaces de contrôle avec précision.  |
| Performances   | Offre une puissance élevée, permettant de générer des forces importantes sur les surfaces de contrôle. Elle est souvent utilisée dans les avions de grande taille et les avions militaires, offrant une réponse rapide et une capacité à gérer des charges élevées. | Permet un contrôle précis des surfaces de contrôle grâce à la technologie des moteurs électriques. Elle offre une meilleure gestion de la charge, une meilleure rétroaction du pilote et une plus grande flexibilité en termes de programmation des lois de commande. Elle est souvent utilisée dans les avions plus petits et les avions commerciaux modernes. |

| type                    | commande hydraulique  | commande électrique   |
|-------------------------|---|---|
| Fiabilité et redondance | Les systèmes hydrauliques sont généralement fiables et peuvent fonctionner dans des conditions extrêmes. Cependant, ils peuvent être sujets à des fuites hydrauliques et nécessitent un entretien régulier. | Les systèmes électriques peuvent être plus légers et plus compacts que les systèmes hydrauliques, ce qui peut entraîner des économies de poids et de carburant. Cependant, l'ajout de câblage électrique peut augmenter la complexité du système. |

## I.7 Chaîne de variateur de vitesse

Une chaîne de variateur de vitesse est un système utilisé pour réguler la vitesse de rotation de moteur. Elle se compose d'un variateur de vitesse, qui permet de contrôler la vitesse de rotation de moteur MSAP d'un aile d'avion, qui fournit la puissance mécanique.

### I.7.1 Schéma fonctionnel d'un EMA

Dans le cas de notre projet on utiliserons la commande de vol à puissance électrique avec un actionneur électromécanique (EMA), donc la chaîne fonctionnelle d'un(EMA) est composée de plusieurs blocs :

- Une source d'alimentation (batterie).
- Un convertisseur AC/AC.
- Redresseur (AC/DC).
- Un onduleur (DC/AC).
- L'actionneur électromécanique : le moteur MSAP.

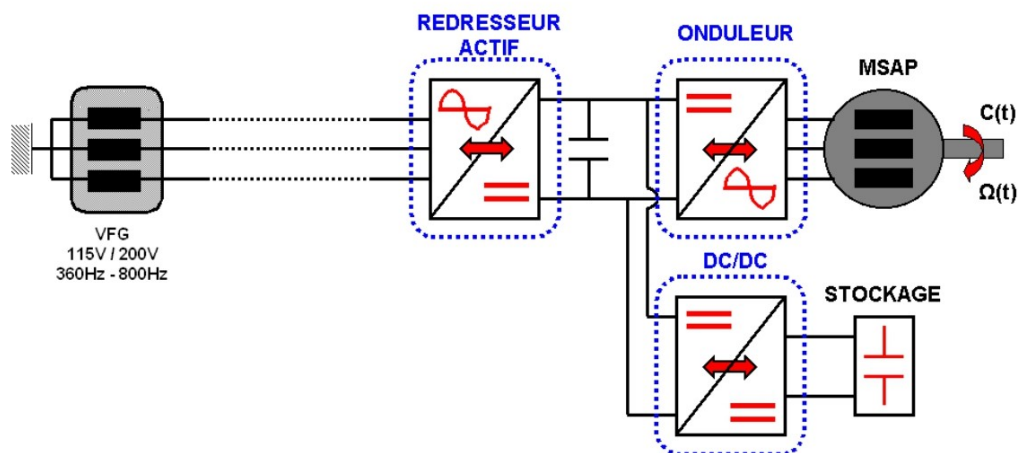


FIGURE I.6 – schéma fonctionnel de variateur de vitesse[4]

### I.7.1.1 Les convertisseurs dans le variateur de vitesse

Pour faire fonctionner notre moteur, l'énergie dérivée d'un réseau alternatif doit être ajustée par des convertisseurs de signal, Ce qui implique trois étapes de conversion distinctes :

- Conversion alternatif-alternatif.
- Conversion alternatif-continu.
- Conversion continu-alternatif.

Dans le chapitre suivant nous avons détaillé le convertisseur le plus répandu utilisé dans les réseaux d'avions "plus électriques" contemporains qui est la fonction d'onduleur triphasé, spécialement conçue pour la gestion des machines tournantes.

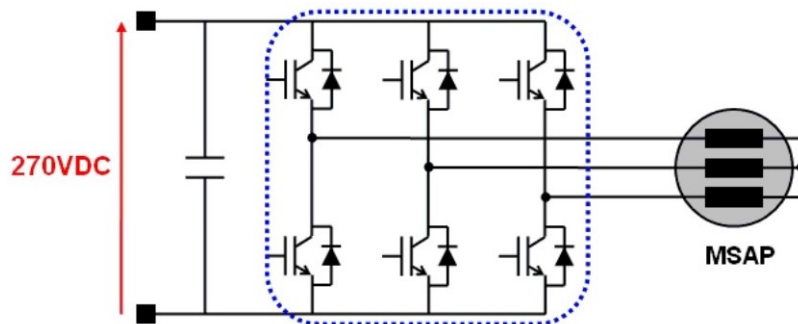


FIGURE I.7 – Onduleur triphasé[4]

### I.7.1.2 Moteur MSAP

Les machines synchrones à aimants permanents (MSAP) sont des machines sans balais, à champs tournants et dont les flux inducteurs sont produits par des aimants permanents. Les rotors des MSAP tournent de façon synchrone avec le champ tournant créé par les courants qui circulent dans leurs bobinages statoriques. Du fait de son fonctionnement synchrone, la MSAP pourrait bien être une alternative pour des applications de positionnement. [19]

Le moteur synchrone (MSAP) est l'actionneur électromécanique utilisé dans notre chaîne pour la commande de vol d'avion.

Le moteur est alimentée par un onduleur triphasé, dont la modélisation de ces deux et la commande vectorielle utilisée est décrite dans le chapitre qui suit.

## I.8 Fiabilité des commandes de vol

- La fiabilité des commandes de vol est primordiale pour garantir la sécurité des vols.
- Les systèmes de commandes de vol sont conçus pour être extrêmement robustes et subissent des tests rigoureux pour minimiser le risque de défaillance.
- La conception robuste permet au système de fonctionner dans diverses conditions.
- La redondance du système est mise en place pour maintenir le contrôle de l'avion en cas de panne.

- La maintenance préventive est effectuée régulièrement pour détecter et résoudre les problèmes potentiels.
- Des tests approfondis sont réalisés lors de la conception et de la fabrication de l'avion pour anticiper les problèmes éventuels.
- Les pilotes sont formés pour comprendre les systèmes de commande de vol et savoir comment réagir en cas d'erreur.
- La surveillance continue du système et la vigilance du pilote sont essentielles pour assurer la sécurité des vols malgré le faible risque de panne.

## Conclusion

Ce chapitre introductif a visé à placer l'étude dans le contexte général du domaine aéronautique. Dans un premier temps un état de l'art sur les concepts fondamentaux de l'aéronautique en détaillant les principes de vol, les types d'aéronefs et la commande de vol.

Deuxièmement l'évènement de l'avion plus électrique pousse le développement de la technologie tout électrique pour laquelle le retour d'expérience reste cependant encore faible. Cette technologie permet en effet de s'affranchir des contraintes de maintenance et environnementales liées à l'utilisation de l'hydraulique. Les développements concernant les futurs actionneurs s'orientent donc d'avantage vers des solutions toutes électriques représentées par les EMAs. C'est dans ce contexte que s'inscrit l'étude présentée sur les commandes de vol à puissance électrique à base des actionneurs électromécanique EMAs.

La seconde partie de ce chapitre s'est donc focalisée sur la technologie des commandes de vol et ses différentes familles qui sont à puissance hydrauliques et électriques.

Enfin, dans un dernier temps, le type de spécifications concernant les actionneurs EHA et EMA de la technologie des commandes de vol à puissance électrique est représentée, en se basant par la suite sur les différents blocs d'une chaîne fonctionnelle d'un actionneur électromécanique qui nous intéresse le plus. Un dernier point consiste sur l'importance de la fiabilité des commandes de vol pour assurer un vol sur et minimiser le risque de défaillance.

La modélisation de moteur (MSAP) utilisé pour la commande de mouvement d'avion est présentée dans le chapitre suivant.



---

---

## CHAPITRE II

---

# MODÉLISATION DE MOTEUR MSAP ET L'ONDULEUR

### Introduction

Le moteur synchrone à aimants permanents est une innovation majeure dans le domaine de la propulsion aéronautique. Ce type de moteur électrique présente des caractéristiques uniques qui le rendent extrêmement adapté à la commande de vol d'un avion, offrant ainsi des avantages significatifs en termes de performances, d'efficacité énergétique et de fiabilité.

Le moteur MSAP et la commande de vol d'un avion est étroitement liée à la façon dont ce type de moteur électrique est intégré dans le système de propulsion global de l'aéronef. Traditionnellement, les avions utilisaient des moteurs à combustion interne pour la propulsion, tandis que les actionneurs hydrauliques contrôlaient les surfaces de contrôle. Cependant, l'évolution technologique a conduit à l'adoption de systèmes de propulsion électrique et à l'intégration des moteurs MSAP pour la commande de vol.

Cette relation entre eux réside dans l'intégration harmonieuse de ces deux éléments pour assurer une propulsion et un contrôle efficaces de l'aéronef. Les systèmes de commande numérique modernes, tels que les commandes de vol électriques (ECS) et les systèmes fly-by-wire, permettent une gestion précise et coordonnée des moteurs MSAP, garantissant ainsi une stabilité, une maniabilité et une sécurité optimales en vol.

Dans ce chapitre, nous aborderons la modélisation du moteur synchrone à aimants permanents (MSAP) ainsi que l'onduleur associé. La modélisation de la MSAP est essentielle pour comprendre son comportement électromécanique et permettre une commande précise du moteur. L'onduleur, quant à lui, est responsable de la conversion de l'énergie électrique et de la fourniture du courant nécessaire au fonctionnement du moteur.

Nous commencerons par présentation des différentes familles de machines électromagnétiques, mettant en évidence les machines à aimants permanents et en particulier la machine synchrone à aimants permanents. Ensuite, nous nous concentrerons sur la modélisation de la MSAP, en expliquant les principes de fonctionnement, les avantages et le domaine d'application de ce type de moteur. Enfin, nous aborderons la modélisation de l'onduleur et son rôle dans la commande du moteur MSAP.

Ce chapitre permettra de jeter les bases nécessaires à la compréhension et à l'analyse du fonctionnement des moteurs MSAP et de leur intégration dans les systèmes de propulsion aéronautique.

## II.1 Machine synchrone à aimant permanent

Les machines synchrones à aimant permanent sont généralement plus légères et plus compactes par rapport à d'autres types de machines électriques, ce qui est un avantage important dans le domaine de l'aviation où la réduction du poids est importante pour améliorer les performances et l'efficacité de système.

### II.1.1 Description

Le terme machine synchrone regroupe toutes les machines dont la vitesse de rotation de l'arbre de sortie est égale à la vitesse de rotation du champ tournant.[10]  
Le moteur synchrone à aimants permanents(MSAP) est une machine électrique qui convertit l'énergie électrique en énergie mécanique.Il se distingue des moteurs synchrones traditionnels par l'utilisation d'aimants permanents intégrés dans le rotor pour créer un champ magnétique constant.Contrairement aux bobines d'excitation des moteurs synchrones conventionnels,les aimants permanents permettent un fonctionnement plus efficace et une meilleure performance.Les MSAP sont utilisés dans diverses applications où une haute efficacité et une précision de contrôle sont nécessaires,comme dans les commandes de vol.



FIGURE II.1 – Machine à aimants permanents[10]

Donc ce type de machines se caractérise par une vitesse de rotation constante, indépendante de la charge, mais liée à la fréquence du réseau par la relation suivante :

$$\Omega = \frac{60 \cdot f}{P} (\text{tr}/\text{min}) \quad (\text{II.1})$$

$\Omega$  : Vitesse de rotation de la machine en (tr/mn).

$f$  : fréquence des courants d'alimentation en (Hz).

$P$  : nombre de paire de pôles.

La MSAP est comme toutes les machines électriques, tournantes, constituée de deux parties essentielles, l'une fixe appelée stator et l'autre mobile dite rotor. [10]

- **Le stator** : nommé aussi induit, il est identique à celui d'une machine asynchrone, constitué de tôles magnétiques empilées, muni de trois enroulements décalés de  $120^\circ$  parcourus par des courants triphasés équilibrés qui créent un champ magnétique tournant.
- **Le rotor** : nommé aussi inducteur, il représente la pièce tournante de la machine, constitué d'un aimant naturel qui a pour fonction principale la création du flux inducteur.

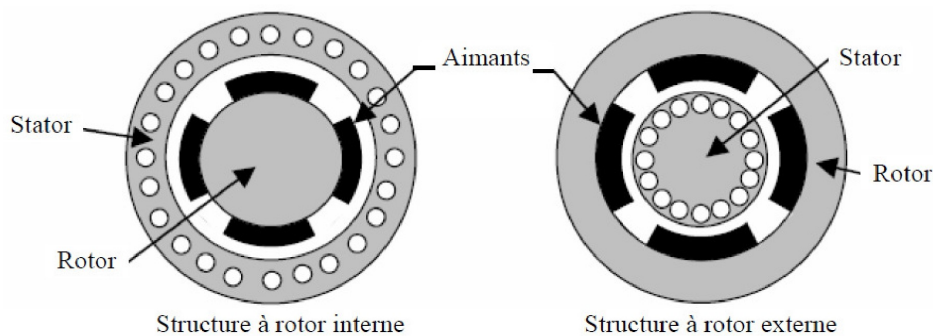


FIGURE II.2 – composition d'un MSAP

## II.1.2 Principe et mode de fonctionnement

un MSAP fonctionne en exploitant l'interaction entre le champ magnétique fixe créé par les aimants permanents du rotor et le champ magnétique tournant produit par les bobines d'inducteurs du stator. Cette interaction génère un couple électromagnétique qui entraîne la rotation du rotor, permettant ainsi la conversion d'énergie électrique en énergie mécanique.

## II.1.3 Avantages de la MSAP[1]

Les machines synchrones à aimants permanents (MSAP) offrent plusieurs avantages par rapport à d'autres types de machines électriques, parmi eux on cite les suivants :

- Puissances massives importantes et élevées.
- Absence de contacts glissants.
- Pas des pertes résistives au rotor ; ce qui facilite l'évaluation de la chaleur due aux pertes dans la machine. Ainsi, il y a suppression d'équipement de refroidissement au rotor.

- Suppression des bagues et des ballait,ce qui réduit les problèmes de maintenance.
- Possibilité de supporter des surcharges transitoires importantes et un bon comportement dynamique en accélération et en freinage.
- Grande fiabilité.
- Fonctionnement en sur vitesse.

### II.1.4 Domaine d'application[2][3]

Grâce au développement de l'électronique de puissance,l'association machine à aimants convertisseur de puissance a trouvé de nombreuses applications dans des domaines très divers tels que :

- La propulsion des véhicules électriques et la propulsion des sous marins.
  - Les équipements de technologie de l'information.
  - Les équipements de soins médicaux et de santé.
  - Les applications robotiques.
  - Les machines outils.
  - Dans le contexte des commandes de vol d'avion,les MSAPs peuvent être utilisés pour actionner les surfaces de contrôle,comme les gouvernes des ailes (ailerons) et les gouvernes de profondeur (élevateurs) d'un avion.
- Ils offrent un contrôle précis, une réponse rapide et une haute fiabilité,contribuant ainsi aux performances et à la sécurité de l'aéronef.

## II.2 Modélisation de la MSAP

Afin de bien commander un moteur MSAP nous devons faire une modélisation de ce dernier,donec la modélisation d'un moteur synchrone à aimants permanents (MSAP) est le processus de création d'un modèle mathématique qui représente le comportement électromécanique du moteur.Ce modèle permet de prédire et d'analyser le fonctionnement du moteur dans différentes conditions de fonctionnement.

Dans cette étude,la machine comporte un stator et un rotor de constitution symétrique avec (P) paires de pôles.Les enroulements statoriques sont le plus souvent connectés en étoile à neutre isolé.L'excitation rotorique est créée par des aimants permanents au rotor.[23]

Les hypothèses de modélisation sont les suivantes : [4]

- le phénomène d'amortissement au rotor est négligeable.
- la machine fonctionne en régime linéaire (circuit magnétique non saturé).
- les inductances sont constantes et ne dépendent pas de la position du rotor (machine magnétiquement lisse au rotor).
- la répartition spatiale du champ magnétique est considérée sinusoïdale (entrefer constant).
- il n'y a pas de courants induits au rotor (machine sans frettes conductrices).

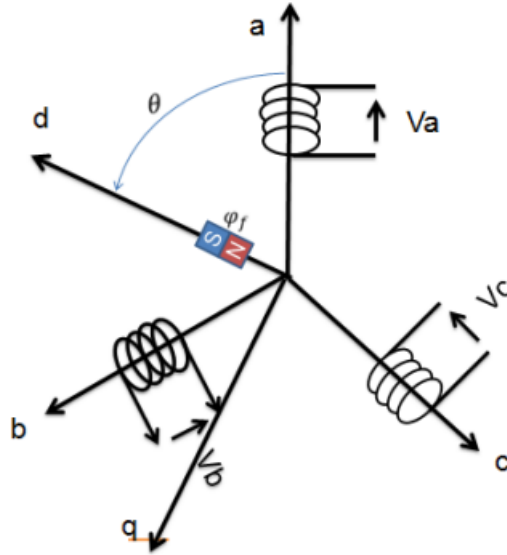


FIGURE II.3 – Représentation d'une machine synchrone à aimants permanents sur ses 3 axes [11]

Le fonctionnement de la machine est régi par les équations présentées ci dessous. La vitesse de rotation  $\Omega$ , proportionnelle à la pulsation électrique, est déduite de la position angulaire mécanique  $\theta_m$  du rotor.

$$\Omega = \frac{d\theta_m}{dt} = \frac{\omega}{P} \quad (\text{II.2})$$

L'équation mathématique de flux magnétique généré par les aimants est :

$$\begin{cases} \phi_A = \phi_M \cdot \cos(p\theta_m) \\ \phi_B = \phi_M \cdot \cos(p\theta_m - \frac{2\pi}{3}) \\ \phi_C = -\phi_A - \phi_B \end{cases} \quad (\text{II.3})$$

Dans les phases, les forces électromotrices résultant de la rotation du champ rotorique sont donnés par les équations suivantes :

$$\begin{cases} e_A = p \cdot \Omega \cdot \phi_M \cdot \sin(p\theta_m) \\ e_B = p \cdot \Omega \cdot \phi_M \cdot \sin(p\theta_m - \frac{2\pi}{3}) \\ e_C = -e_A - e_B \end{cases} \quad (\text{II.4})$$

Les tensions différentielles aux bornes des enroulements s'écrivent :

$$\begin{cases} V_A = R_s \cdot i_{phA} + \frac{d(L \cdot i_{phA} + M \cdot i_{phB} + M \cdot i_{phC})}{dt} + e_A \\ V_B = R_s \cdot i_{phB} + \frac{d(L \cdot i_{phB} + M \cdot i_{phA} + M \cdot i_{phC})}{dt} + e_B \\ V_C = R_s \cdot i_{phC} + \frac{d(L \cdot i_{phC} + M \cdot i_{phB} + M \cdot i_{phA})}{dt} + e_C \end{cases} \quad (\text{II.5})$$

L'inductance  $L$  est la somme de l'inductance propre  $L_p$  et de l'inductance de fuite  $L_f$ .

$$L = L_p + L_f \quad (\text{II.6})$$

Le couple électromagnétique  $C_{em}$  produit par la machine est fonction des courants circulant dans les phases :

$$C_{em} = p \cdot \phi_M \cdot [i_{phA} \cdot \sin(p\theta_m) + i_{phB} \cdot \sin(p\theta_m - \frac{2\pi}{3}) + i_{phC} \cdot \sin(p\theta_m + \frac{2\pi}{3})] \quad (II.7)$$

Le couple utile instantané  $C_u$  appliqué à l'arbre moteur tient compte des frottements visqueux  $f_v$  de l'ensemble machine-pompe.

$$C_r = C_{em} - f_v \cdot \Omega - j \cdot \frac{d\Omega}{dt} \quad (II.8)$$

Avec :

$J$  : Le couple d'inertie des masses tournantes.

$C_r$  : Couple résistant (ou statique) imposé par la charge mécanique.

$C_{em}$  : Couple électromagnétique.

$\Omega$  : vitesse mécanique de rotation.

$f$  : Coefficients des frottements visqueux.

$\phi_f$  : Flux des aimants permanents.

Enfin, les paramètres à régler du modèle de MSAP sont récapitulés dans le tableau suivant.

|                         |   |
|-------------------------|---|
| $R_S$ ( $\Omega$ )      | Résistance de bobine statorique                                     |
| $L$ (H)                 | Inductance d'une phase statorique                                   |
| $M$ (H)                 | Inductance mutuelle entre deux phases                               |
| $\Phi_m$ (Wb)           | Valeur maximale du flux total des aimants dans une phase statorique |
| $p$                     | Nombre de paires de pôles   |
| $J$ ( $\text{kg.m}^2$ ) | Inertie   |
| $f_v$ (N.m.s/rad)       | Coefficient de frottements visqueux                                 |

FIGURE II.4 – Paramètres réglables du modèle de MSAP[4]

Le système sur les trois axes est fortement non linéaire. L'étude de ce dernier est très compliquée le cas de saturation, la perméabilité devient variable, alors les inductances deviennent aussi variables, donc pour cela faut simplifier ce modèle en utilisant la transformation de Park qui consiste à transformer les enroulements immobiles (a, b, c) par des enroulements (d, q) tourne avec le rotor.

## II.2.1 Transformation de Park

A l'aide de la transformation de Park, on passe des grandeurs statoriques réelles (tension, flux courant) à leurs composantes fictives appelés les composantes dq.[24]

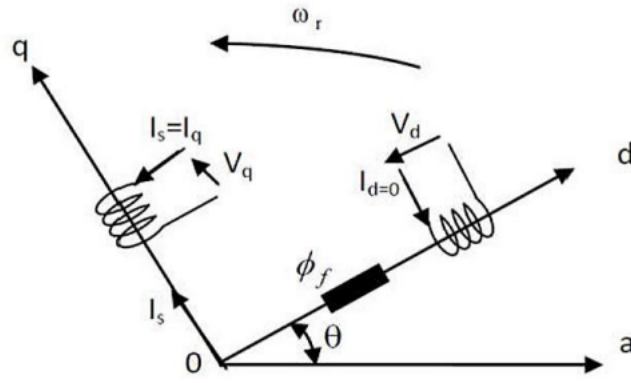


FIGURE II.5 – Présentation de MSAP sur les axes fictifs[12]

En termes physiques, cette transformation est interprétée comme étant une substitution des enroulements immobiles (abc) par des enroulements fictifs (dq) en quadrature tournant avec le rotor et reprenant les mêmes considérations ou aspects en termes de f.m.m, de flux, de couple ou du moins une image qui leur sera parfaitement proportionnelle :

$$X_{dq0} = P(\theta) \cdot X_{abc} \quad , \quad \begin{bmatrix} X_d \\ X_q \\ X_0 \end{bmatrix} = P(\theta) \cdot \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (\text{II.9})$$

$P(\theta)$  est la matrice de park défini par :

$$[P(\theta)] = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (\text{II.10})$$

En appliquant la transformation de Park au système d'équations du moteur synchrone, on obtient :

$$V_{dq0} = P(\theta) \cdot V_{abc} \quad (\text{II.11})$$

ce qui implique :

$$V_{dq0} = P(\theta) \left( [R] \cdot [I_{abc}] + \frac{d}{dt} \cdot [L \cdot I_{abc} + \phi_{abc}] \right) \quad (\text{II.12})$$

sachant que :

$$I_{abc} = [P(\theta)]^{-1} \cdot I_{dq} \quad \text{et} \quad \phi_{abc} = [P(\theta)]^{-1} \cdot \phi_{dq} \quad (\text{II.13})$$

Alors :

$$V_{dq0} = P(\theta) \cdot R \cdot [P(\theta)]^{-1} \cdot I_{dq} + P(\theta) \cdot [P(\theta)]^{-1} \cdot \frac{d}{dt} \cdot \phi_{dq} + P(\theta) \cdot \frac{d}{dt} \cdot [P(\theta)]^{-1} \cdot \phi_{dq} \quad (\text{II.14})$$

On déduit l'équation de Park :

$$V_{dq0} = R \cdot I_{dq} + \frac{d}{dt} \cdot \phi_{dq} + P(\theta) \cdot \frac{d}{dt} \cdot [P(\theta)]^{-1} \cdot \phi_{dq} \quad (\text{II.15})$$

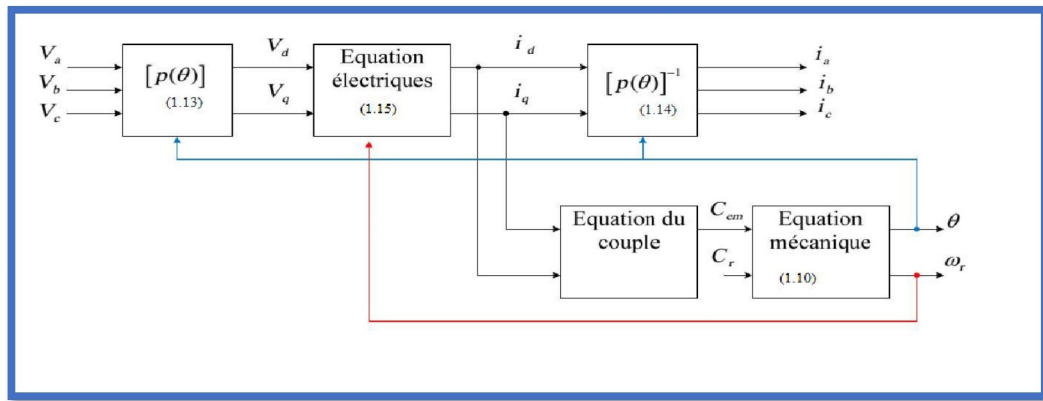


FIGURE II.6 – Schéma fonctionnel du modèle de Park

## II.3 Autopilotage et la commande vectorielle

L'autopilotage et la commande vectorielle sont deux technologies interconnectées qui jouent un rôle essentiel dans le domaine de l'aviation moderne. L'autopilotage permet d'automatiser certaines tâches de vol et de maintenir l'avion sur une trajectoire prédéfinie, tandis que la commande vectorielle offre un contrôle précis des moteurs électriques utilisés dans les systèmes de propulsion électrique de l'avion. En combinant ces deux approches, l'autopilotage peut tirer parti de la commande vectorielle pour gérer avec précision la poussée, la vitesse et d'autres paramètres du moteur électrique, facilitant ainsi les opérations de vol automatiques et contribuant à une meilleure performance, une réduction de la charge de travail du pilote et une amélioration de la sécurité dans le domaine de l'aviation.

### II.3.1 Autopilotage de la machine synchrone à aimants permanents [4]

L'autopilotage d'une machine synchrone à aimants permanents (MSAP) se réfère à la capacité d'un système de contrôle à maintenir et réguler automatiquement le fonctionnement de la MSAP. Cela implique la régulation de la vitesse, du couple et de la position de la MSAP en utilisant des boucles de rétroaction et des algorithmes de contrôle avancés. L'autopilotage permet d'optimiser les performances de la MSAP tout en assurant une régulation précise et une efficacité énergétique élevée. Il inclut également des fonctionnalités de protection et de diagnostic pour garantir un fonctionnement sûr de la MSAP.

Ce système de contrôle comprend généralement un microcontrôleur ou un processeur numérique de signal (DSP) qui traite les signaux de commande et de rétroaction provenant des capteurs et des actionneurs de la MSAP.



### II.3.1.1 Régulation de courant

Le flux magnétique généré par les aimants s'exprime en fonction de la position mécanique  $\theta_m$  du rotor.

$$\begin{cases} \phi_A = \phi_M \cdot \cos(p\theta_m) \\ \phi_B = \phi_M \cdot \cos(p\theta_m - \frac{2\pi}{3}) \\ \phi_C = \phi_M \cdot \cos(p\theta_m + \frac{2\pi}{3}) \end{cases} \quad (\text{II.16})$$

La rotation de la machine induit dans les phases des forces électromotrices proportionnelles à la pulsation électrique  $\omega = p\Omega$ .

$$e = \frac{d\phi}{dt} = \frac{\partial\phi}{\partial\phi_m} \cdot \frac{d\phi_m}{dt} \quad (\text{II.17})$$

$$\begin{cases} e_A = p \cdot \Omega \cdot \phi_M \cdot \sin(p\theta_m) \\ e_B = p \cdot \Omega \cdot \phi_M \cdot \sin(p\theta_m - \frac{2\pi}{3}) \\ e_C = p \cdot \Omega \cdot \phi_M \cdot \sin(p\theta_m + \frac{2\pi}{3}) \end{cases} \quad (\text{II.18})$$

Pour une commande à couple maximal, les courants de référence générés sont en phase avec les forces électromotrices.

$$\begin{cases} i_{phA} = -I_{ref} \cdot \sin(p\theta_m) \\ i_{phB} = -I_{ref} \cdot \sin(p\theta_m - \frac{2\pi}{3}) \end{cases} \quad (\text{II.19})$$

La régulation est alors réalisée avec un correcteur proportionnel-intégral . La fonction de transfert du système en boucle fermée est la suivante.

$$T_{BFi} = \frac{1 + (\frac{k_{p-i}}{k_{i-i}}) \cdot p}{1 + (\frac{k_{p-i}}{k_{i-i}}) \cdot p + (\frac{L_c}{k_{i-i} \cdot V_{DC}}) \cdot p^2} \quad (\text{II.20})$$

Le correcteur PI est numérisé simplement par la méthode des trapèzes. Les mesures de courant issues des modules de conversion sont numérisées au niveau de la carte de commande. Enfin, les rapports cycliques calculés sont transmis au FPGA, qui s'acquitte de la modulation de largeur d'impulsion et transmet les ordres de commande vers le module de conversion.

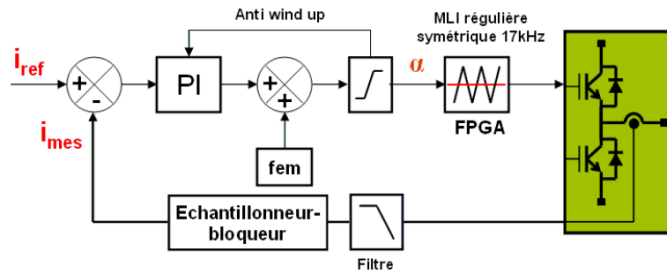


FIGURE II.7 – schéma bloc de la boucle de régulation de courant [4]

### II.3.1.2 Régulation de vitesse

L'utilisateur transmet les consignes de vitesse de rotation de la MSAP par le PC. La régulation de vitesse est réalisée sur le même principe que la régulation de courant. Le couple de charge  $C_{ch}$ , qui intervient comme une perturbation, est également imposé par l'utilisateur via le variateur de la machine asynchrone. Les fonctions de transfert du système en boucle ouverte TBO et en boucle fermée TBF sont établies en assimilant l'ensemble mécanique à un système du premier ordre. Il est considéré que la boucle de courant est à gain constant.

$$T_{BO\Omega} = K_c \cdot \frac{k_{p\Omega} \cdot p + k_{i\Omega}}{p} \cdot \frac{1}{J \cdot p + f} \quad (\text{II.21})$$

$$T_{BF\Omega} = \frac{1 + \frac{k_{p\Omega}}{k_{i\Omega}}}{1 + \frac{f + K_C \cdot k_{p\Omega}}{K_C \cdot k_{i\Omega}} \cdot p + \frac{J}{K_C \cdot k_{i\Omega}} \cdot p^2} \quad (\text{II.22})$$

Les paramètres du correcteur sont déduits par assimilation avec un système du deuxième ordre de pulsation propre  $\omega_0\Omega$  et de coefficient d'amortissement  $\varepsilon\Omega$ .

$$\begin{cases} k_{p\Omega} = \frac{2 \cdot \zeta \Omega \cdot J \cdot \omega_0 \Omega - f}{K_C} \\ k_{i\Omega} = \frac{J \cdot \omega_0^2 \Omega}{K_C} \end{cases} \quad (\text{II.23})$$

Les caractéristiques mécaniques ont dû être identifiées : l'inertie de l'ensemble mécanique est déterminée par des essais de lâcher ( $J=21 \cdot 10^{-3} \text{kg} \cdot \text{m}^2$ ) et les frottements sont mesurés en régime statique ( $f=4 \cdot 10^{-3} \text{N} \cdot \text{m} \cdot \text{s}/\text{rad}$ ).

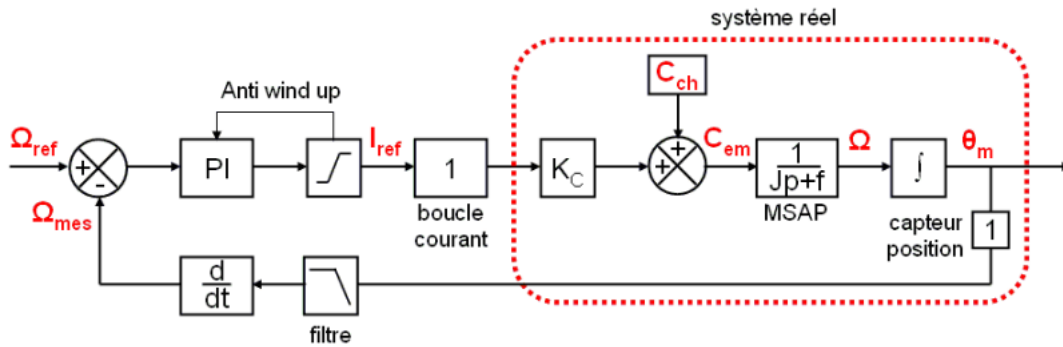


FIGURE II.8 – schéma bloc de la boucle de régulation de vitesse [4]

## II.3.2 la commande vectorielle

La commande vectorielle, également connue sous le nom de commande de champ orienté (FOC), est une méthode utilisée pour contrôler les moteurs électriques, en particulier les moteurs à courant alternatif (AC). Elle permet un contrôle précis du couple et de la vitesse du moteur, ainsi qu'une réponse dynamique améliorée.

La commande vectorielle repose sur une représentation mathématique du moteur dans un espace à deux dimensions, composé d'un axe direct (flux magnétique) et d'un axe en

quadrature (couple électromagnétique). En contrôlant indépendamment ces deux composantes, la commande vectorielle permet de générer le couple et la vitesse souhaités.

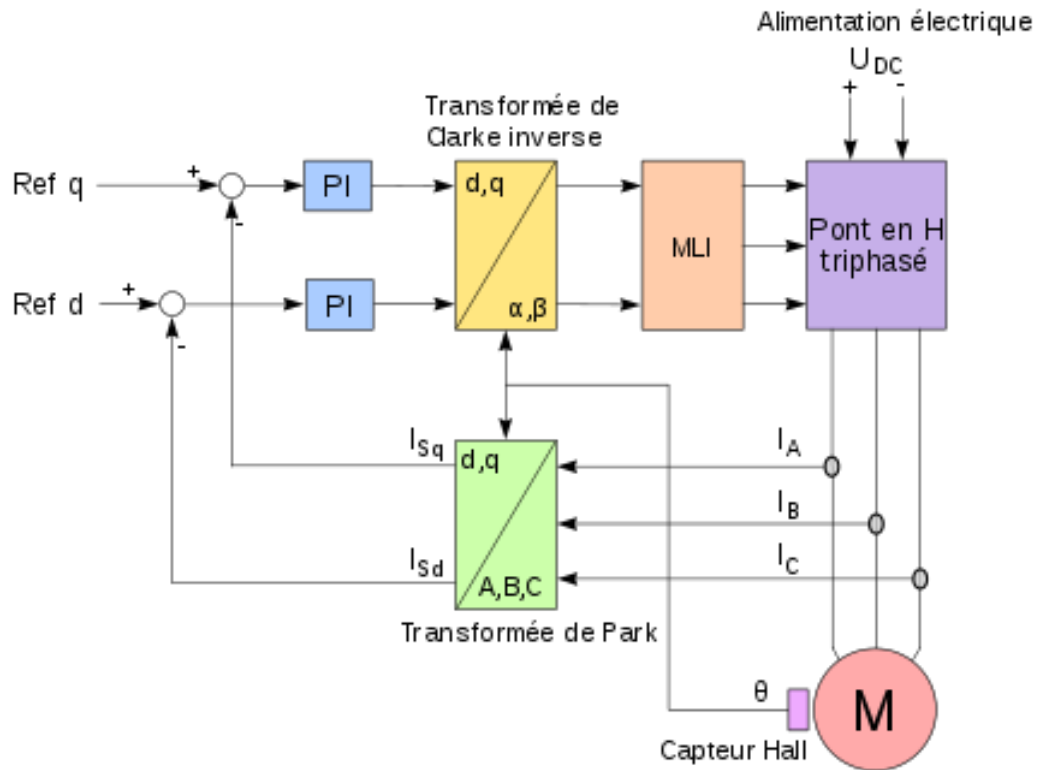


FIGURE II.9 – commande vectorielle d'un moteur triphasé[13]

### II.3.3 principe de fonctionnement

La commande vectorielle avec moteur synchrone à aimants permanents (PMSM) permet de réguler de manière indépendante le flux magnétique et le couple du moteur. Elle offre une réponse dynamique rapide, un contrôle précis de la vitesse et une haute efficacité énergétique. La technique repose sur la transformation du système de coordonnées triphasé en coordonnées fixes liées au rotor, la régulation du flux magnétique par ajustement du courant dans l'axe d, la régulation du couple par ajustement du courant dans l'axe q, et éventuellement le contrôle de la vitesse. Des boucles de rétroaction mesurent les grandeurs électriques et mécaniques pour calculer les signaux de commande en temps réel.

La commande vectorielle est couramment utilisée dans les moteurs PMSM pour diverses applications, y compris les véhicules électriques, les machines industrielles et les systèmes de propulsion électrique des avions.

### II.3.4 Avantages de la commande vectorielle :

- **Contrôle précis** : La commande vectorielle permet un contrôle précis du couple et de la vitesse du moteur, offrant une réponse dynamique élevée.
- **Fonctionnement à basse vitesse** : Elle permet un fonctionnement stable et précis à des vitesses très faibles, y compris à l'arrêt complet.

- **Réduction des pertes d'énergie** : La commande vectorielle contribue à réduire les pertes d'énergie en optimisant l'efficacité globale du système.
- **Contrôle indépendant du couple et de la vitesse** : Elle permet un contrôle indépendant du couple et de la vitesse, offrant une flexibilité accrue pour répondre aux besoins spécifiques de l'application.

### II.3.5 Application de la commande vectorielle dans l'avion

La commande vectorielle trouve plusieurs applications dans le domaine aéronautique, notamment dans les avions. Voici quelques domaines d'application de la commande vectorielle dans l'aviation :

- Commande des moteurs électriques des systèmes auxiliaires.
- Commande des servomoteurs des gouvernes.
- Commande des moteurs électriques des trains d'atterrissage.
- Commande des moteurs électriques des actionneurs de portes.

## II.4 Synoptique de la chaîne de traction

Dans notre cas l'alimentation est un convertisseur constitué de trois étages, un redresseur connecté au réseau, un filtre qui permet de réduire les ondulations du courant et de la tension, et un onduleur de tension type modulation de largeur d'impulsion à onde sinusoïdale (PWM).

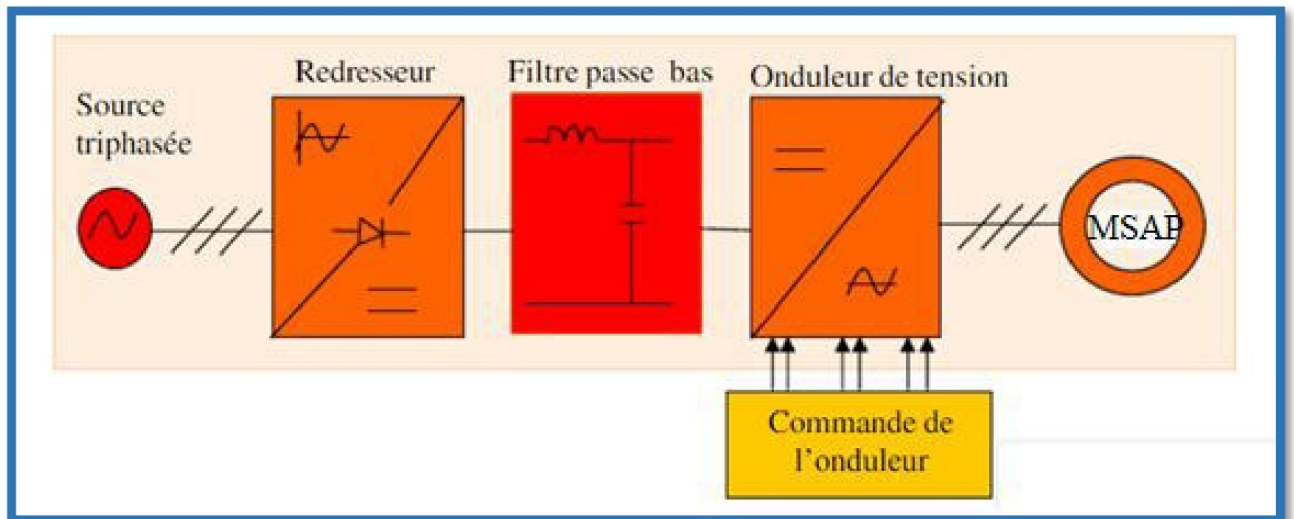


FIGURE II.10 – Schéma Global de L'association MSAP- Convertisseur

- **Redresseur** : pont de diodes montées permet de faire la conversion de la tension alternative en une tension continue de forme d'onde.
- **Filtrage** : Bloc insérer entre le redresseur et l'onduleur de tension un filtre passe bas (LC) pour éliminer les hautes fréquences.
- **Onduleur** : un pont redresseur monté par des thyristors qui convertit la tension continue en tension alternative avec une amplitude et une fréquence variables, tout en gardant le rapport  $U/f$  constant.

## II.5 Onduleur

### II.5.1 Présentation

Un onduleur est un dispositif d'électronique de puissance permettant de générer des tensions et des courants alternatifs à partir d'une source d'énergie électrique continue. Son fonctionnement est à dissocier des autres convertisseurs comme les convertisseurs AC/AC, les redresseurs (AC/DC) ou encore les convertisseurs DC/DC. Cependant un onduleur peut être associé à d'autres convertisseurs pour en changer la fonction.

L'onduleur (DC/AC) est un convertisseur statique permettant la conversion de l'énergie électrique sous sa forme continue (DC) vers la forme alternative (AC). Grâce à des composants semi conducteurs commandés, il permet d'obtenir aux bornes du récepteur une tension alternative réglable en fréquence et en valeur efficace.

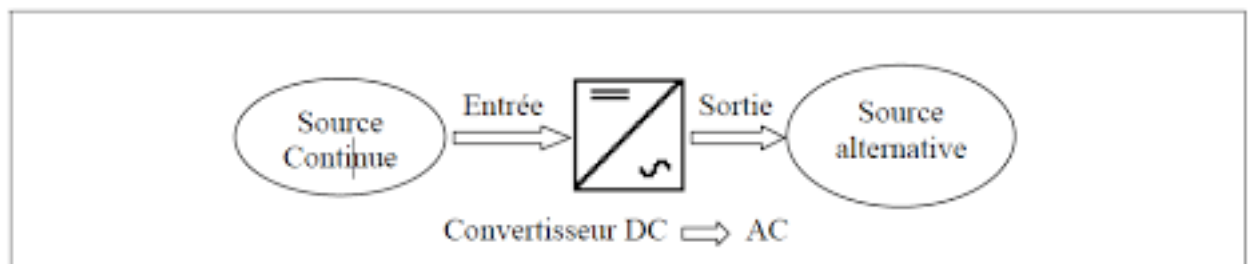


FIGURE II.11 – Principe de fonctionnement d'un onduleur

### II.5.2 Alimentation à partir d'un onduleur de tension

Pour commander un moteur synchrone, un onduleur est utilisé pour convertir une tension continue constante en une tension alternative ajustable, appliquée aux enroulements statoriques de la machine. Le contrôle précis de la tension de sortie de l'onduleur influence la vitesse, le couple et d'autres paramètres du moteur.

Des techniques de modulation de largeur d'impulsion (MLI) ou de modulation de largeur d'impulsion à onde sinusoïdale (PWM) sont utilisées pour ajuster la durée des impulsions de tension et générer des formes d'ondes sinusoïdales ou approximativement sinusoïdales. Ces stratégies de commande reposent sur les performances en fréquence de découpage autorisées par les composants électroniques. Le système d'entraînement complet associe la machine synchrone à l'onduleur de tension triphasé.

La figure(12) représente le schéma d'association de moteur avec l'onduleur triphasé de tension :

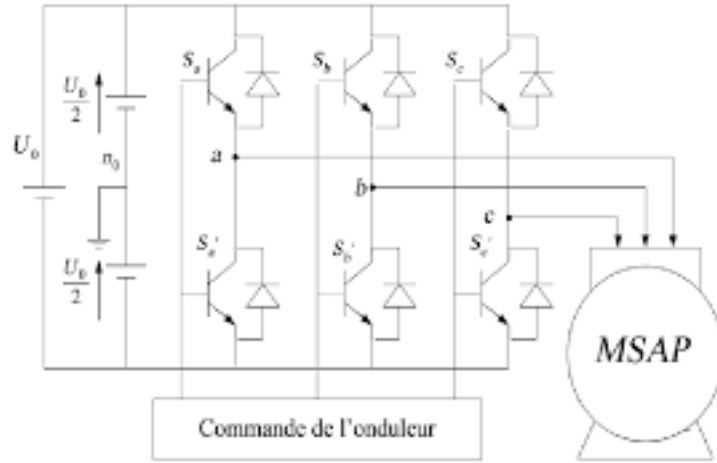


FIGURE II.12 – Schéma de l'association MSAP et l'onduleur triphasé de tension

### II.5.3 Modélisation de l'onduleur triphasé de tension à deux niveaux

La modélisation de l'onduleur triphasé de tension à deux niveaux consiste à représenter le fonctionnement et les caractéristiques de cet onduleur de manière mathématique. Ce type d'onduleur est couramment utilisé dans les systèmes d'entraînement électrique pour contrôler la tension appliquée aux moteurs électriques (MSAP).

Les tensions aux bornes de l'onduleur :

$$\begin{cases} V_{ab} = V_{a0} - V_{b0} \\ V_{bc} = V_{b0} - V_{c0} \\ V_{ca} = V_{c0} - V_{a0} \end{cases} \quad (\text{II.24})$$

Si N est le point neutre des phases de la machine, alors :

Les tensions  $V_{an}$ ,  $V_{bn}$  et  $V_{cn}$  forment un système de tension triphasées équilibrées, Donc :

$$V_{an} + V_{bn} + V_{cn} = 0 \quad (\text{II.25})$$

$$\begin{cases} V_{an} = V_{a0} + V_{0n} \\ V_{bn} = V_{b0} + V_{0n} \\ V_{cn} = V_{c0} + V_{0n} \end{cases} \quad (\text{II.26})$$

$$V_{an} + V_{bn} + V_{cn} = 3.V_{0n} + V_{a0} + V_{b0} + V_{c0} \quad (\text{II.27})$$

On déduit :

$$V_{aN} = \frac{1}{3} \cdot (V_{a0} + V_{b0} + V_{c0}) \quad (\text{II.28})$$

A partir des deux équations précédente :

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} V_{a0} \\ V_{b0} \\ V_{c0} \end{bmatrix} \quad (\text{II.29})$$

Dans une branche d'onduleur, un interrupteur statique peut prendre la valeur +1 ou -1 selon les conditions suivantes :

$$V_{aref} \geq V_p \implies \begin{cases} S_a = 1 \\ S_a = -1 \quad \text{sinon} \end{cases} \quad (\text{II.30})$$

$$V_{bref} \geq V_p \implies \begin{cases} S_b = 1 \\ S_b = -1 \quad \text{sinon} \end{cases} \quad (\text{II.31})$$

$$V_{cref} \geq V_p \implies \begin{cases} S_c = 1 \\ S_c = -1 \quad \text{sinon} \end{cases} \quad (\text{II.32})$$

$V_{ref}$  : amplitude de référence.

$V_p$  : amplitude de porteuse.

Les tensions simples à la sortie de l'onduleur peuvent s'exprimer par la relation matricielle suivante :

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{U_c}{3} \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \quad (\text{II.33})$$

L'objectif principal de l'onduleur est de produire des tensions à sa sortie, aussi proches que possible d'une forme sinusoïdale idéale. Ainsi, plusieurs stratégies de commande ont été développées pour déterminer les trois fonctions logiques  $S_i(a, b, c)$ .

## II.5.4 Différents types de commandes de commutation

Ils existent plusieurs types de commandes de commutation incluent :

- Contrôle des ondes carrées.
- Commandes avec fentes de largeur variable.
- Commande de pas (marche d'escalier).
- Contrôle de la modulation de largeur d'impulsion (PWM).

Dans notre travail, nous exécuterons les commandes PWM.

### II.5.4.1 La modulation de largeur d'impulsion (PWM)

#### a) définition

La modulation de largeur d'impulsion (PWM) est une technique couramment utilisée dans le contrôle de la quantité de puissance fournie à un dispositif ou à un système. Elle consiste à varier la largeur relative des impulsions électriques périodiques tout en maintenant une fréquence constante. Le signal PWM est composé d'impulsions de durée variable mais de largeur constante, où la durée de chaque impulsion détermine la quantité d'énergie électrique transférée pendant cette période. Cette technique est largement appliquée dans

le contrôle des moteurs à stator actif à aimants permanents (MSAP),notamment dans le domaine de l'aéronautique. En utilisant la modulation PWM,il est possible de réguler avec précision la vitesse et le couple du moteur MSAP,offrant ainsi un contrôle efficace pour la commande de vol des avions.

### b) Principe de fonctionnement

En modulant la largeur d'impulsion,on découpe une onde rectangulaire pour former une succession de créneaux d'amplitude égale à la tension d'alimentation continue et de largeur variable,qui composent la tension de sortie de l'onduleur.Pour générer un signal MLI,on compare généralement une porteuse de haute fréquence,représentée par un signal triangulaire,avec un signal de référence modulatrice,qui constitue l'énergie du signal recueilli à la sortie de l'onduleur.

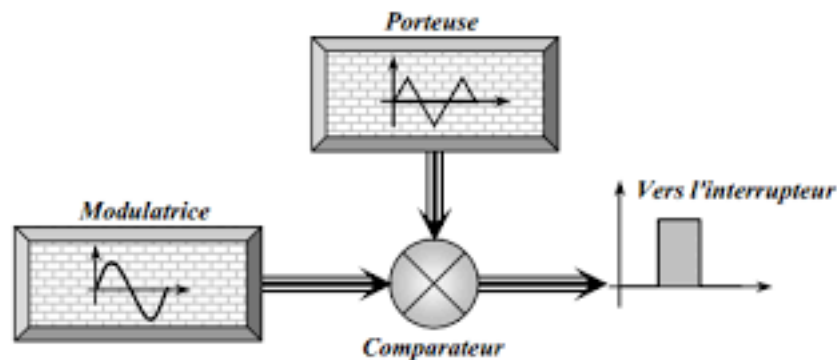


FIGURE II.13 – Schéma synoptique de la M.L.I.[14]

## Conclusion

Le chapitre se concentrera sur la modélisation du moteur MSAP et de l'onduleur associé.En comprenant le fonctionnement électromécanique de la MSAP,nous pourrions développer une commande précise du moteur, tandis que l'onduleur jouera un rôle important dans la conversion de l'énergie électrique et la fourniture du courant nécessaire au moteur.

La modélisation de ces éléments clés jettera les bases nécessaires pour comprendre le fonctionnement des moteurs MSAPs et leur intégration dans les systèmes de propulsion aéronautique.Cela permettra une analyse approfondie du comportement des moteurs MSAPs, de leurs performances et de leur interaction avec les systèmes de commande de vol.

En somme,ce chapitre fournira les connaissances essentielles pour explorer plus en détail les aspects de la modélisation et de la commande des moteurs MSAPs dans le contexte de la propulsion aéronautique.Il s'agit d'une étape nécessaire pour exploiter pleinement les avantages de ces moteurs innovants et pour développer des systèmes de commande de vol avancés et efficaces.

Grâce à sa puissance de calcul élevée et à sa capacité à traiter des signaux en temps réel,la carte DSP constitue une solution idéale pour la commande de moteur MSAP.L'étude de cette carte est présentée dans le chapitre suivant.



---

---

# CHAPITRE III

---

## ETUDE DE LA CARTE DSP TMS320F28335

### Introduction

Spécialement conçus pour les applications de contrôle en temps réel telles que les entraînements de moteurs industriels, les onduleurs solaires et l'alimentation numérique, les microcontrôleurs C2000™ offrent des capacités optimales de traitement, de détection et d'actionnement pour améliorer les performances en boucle fermée. Ces microcontrôleurs sont également idéaux pour la détection et le traitement des signaux dans les véhicules électriques et les transports, ainsi que pour les systèmes de commande de moteurs.

La gamme C2000 comprend des microcontrôleurs hautes performances ainsi que des microcontrôleurs performants d'entrée de gamme. Dans le domaine des applications de contrôle complexes, les familles des TMS320F2833x se distinguent comme des solutions exceptionnellement performantes et entièrement intégrées.

L'utilisation d'une carte DSP pour les commandes de vol d'avion revêt une importance majeure. Elle permet un traitement rapide des signaux provenant des capteurs et des commandes du pilote, assurant ainsi des réponses précises et stables. Grâce à sa flexibilité et à sa programmabilité, elle peut être adaptée aux différentes conditions de vol et garantir la sécurité de l'aéronef. De plus, l'utilisation d'une carte DSP redondante améliore la fiabilité du système de commande de vol en assurant une continuité des opérations en cas de défaillance.

Ce chapitre est consacré à une présentation générale au DSP en premier lieu puis une description architecturale détaillée de DSP TMS320F28335 (on se réfère à F28335 dans la suite du mémoire) de Texas Instrument que nous avons utilisé dans notre projet, issu de la génération 28x et appartenant à la famille TMS320 conçu pour le contrôle numérique des signaux ainsi qu'à d'autres applications de commande (robotique) et de traitement de signaux.

### III.1 Présentation de DSP

Un DSP (processeur de signal numérique) est un type spécialisé de microprocesseur conçu pour le traitement rapide et efficace des signaux numériques en temps réel. Contrairement à un microprocesseur standard, un DSP est équipé de fonctions spéciales qui lui confèrent des capacités de traitement de signal supérieures.

Il est généralement utilisé de manière autonome et est souvent intégré à un microcontrôleur avec des périphériques tels que des mémoires, des ports d'entrée/sortie, des timers et des ports série synchrones rapides.

Il est doté de la première génération de timers ePWM. Le 12-bit 12.5 MSPS CAN est toujours le meilleur de sa catégorie pour une intégration analogique-numérique, convertisseur analogique-numérique. Le F2833x dispose d'un processeur de 150 MHz et jusqu'à 512KB de Flash sur la puce. Il est disponible avec un boîtier QF de 176 broches ou un boîtier BGA de 179 boules. Le F28335 fournit 88 ports GPIO configurables comme entrées ou sorties numériques. [16]

Le DSP utilise un jeu d'instructions et une architecture optimisés pour un traitement de signal efficace en temps réel. De plus, chaque nouvelle génération de DSP est plus performante que la précédente tout en étant moins coûteuse. [16]

### III.2 Architecture de DSP

Une architecture de DSP est une structure interne spécifique conçue pour le traitement numérique des signaux. Elle est optimisée pour exécuter des opérations mathématiques et des algorithmes de traitement de signal de manière efficace et rapide.

A priori, pour augmenter le rapport performance/prix, les concepteurs de DSP ont utilisé une structure de Harvard dite 'structure de Harvard modifiée'.

À l'extérieur, le DSP ne propose qu'un bus de données et un bus d'adresse, comme la structure Von Neuman. Toutefois, à l'intérieur, la puce DSP dispose de deux bus distincts de données et de deux bus distincts d'adresses. [25]

Le transfert des données entre les bus externes et internes est effectué par multiplexage temporel, c'est le cas du TMS320F28335. [26]

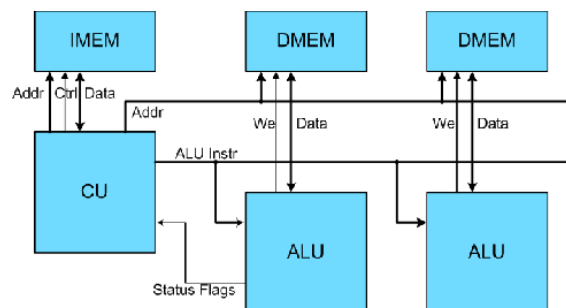


FIGURE III.1 – L'architecture de Harvard modifiée [15]

L'un des principaux avantages de l'utilisation d'un processeur de signaux numériques (DSP) est sa capacité à traiter plusieurs opérandes par le biais d'une seule instruction, ce qui se traduit par une nette augmentation des performances en matière de traitement des signaux numériques en temps réel. Toutefois, pour y parvenir, un DSP doit effectuer deux accès à la mémoire en un cycle d'horloge. Pour ce faire, plusieurs ensembles de mémoire vive (RAM) et de mémoire morte (ROM) doivent être utilisés pour stocker les données et les instructions.

### III.3 Avantage des DSPs

Dans la conception de systèmes de traitement numérique, différent du processeur Traditionnellement, tout système basé sur DSP a bénéficié des avantages de Spécificités architecturales et de programmation, dont :

- Traitement en temps réel pour des réponses rapides et précises.
- Puissance de calcul élevée pour gérer des algorithmes complexes.
- Flexibilité de programmation et de configuration.
- Optimisation de l'efficacité énergétique.
- Intégration avec d'autres systèmes embarqués.
- Réduction des coûts et de l'encombrement.

Grâce à ces avantages, les DSP sont largement utilisés dans les systèmes de commande de vol d'avion pour assurer une précision, une réactivité et une fiabilité élevées dans le pilotage de l'aéronef.

### III.4 Les Caractéristiques de TMS320F28335

Principales caractéristiques de TMS320F28335 : [16]

- Alimentation 5V.
- DSP TMS320F28335 fonctionnant à 150MHz avec un temps de cycle de 6.67 ns en 32 bits.
- Une mémoire Flash (EEPROM) de  $16 \times 256$  Kbits.
- Une mémoire RAM (SRAM) d'une capacité totale de  $16 \times 34$  Kbits.
- Une ROM OTP de  $1K \times 16$  et une ROM d'amorçage de  $8K \times 16$ .
- Un bus mémoire pour relier la mémoire, le CPU et les périphériques.
- Une interface JTAG IEEE 1149.1, permettant dans ce cas de contrôler et d'intervenir en temps réel via un PC et un kit de développement sur le fonctionnement du DSP.
- Il est également équipé d'une interface externe 16 ou 32 bits (XINTF) avec plus de 2 millions d'adresses.

## III.5 Architecture de TMS320F28335

### III.5.1 Diagramme fonctionnel

La figure III.2 correspond à une image réelle.

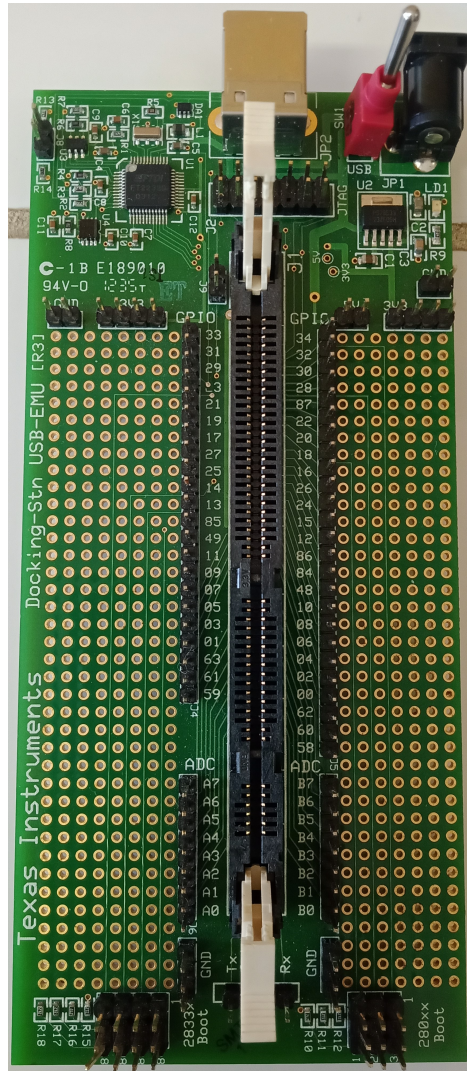


FIGURE III.2 – Image réelle d'une carte DSP F28335

La figure(3) présente le schéma fonctionnel du F28335 et le calculateur numérique. Le diagramme est extrait de la fiche technique élaborée par le concepteur (Texas Instruments). Ce diagramme montre (en blocs) les différents modules constituant le processeur (CPU, ADC, module MLI, SARAM...) et les liaisons et canaux de communication (bus, broches...).

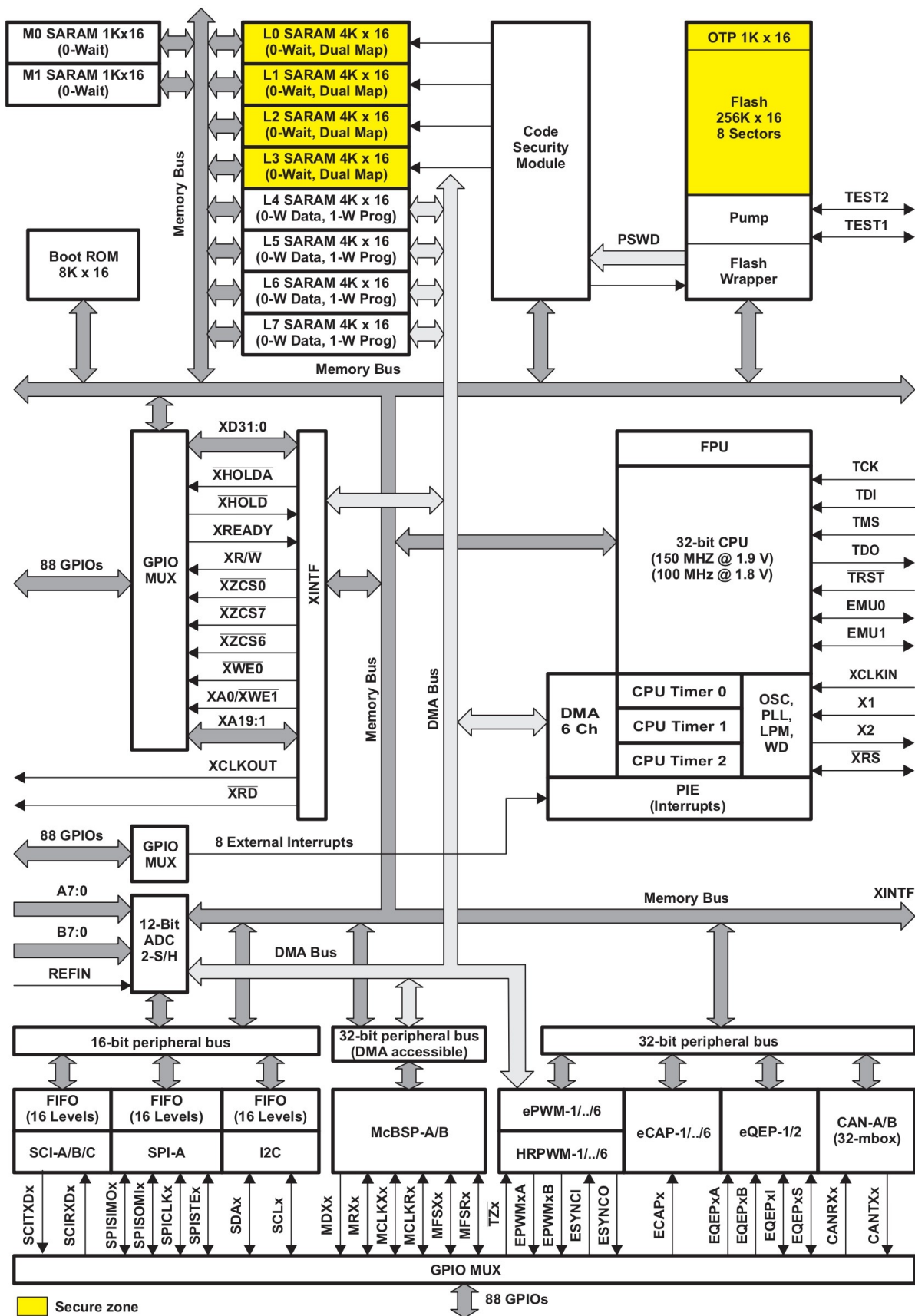


FIGURE III.3 – Le schéma fonctionnel du DSP TM320F28335 [16]

### III.5.2 Diagramme des broches

Le diagramme des broches du DSPF28335 est un schéma qui montre les différentes broches du microcontrôleur et leurs fonctions.

La carte DSPF28335 dispose de 176 broches réparties sur deux rangées de broches de chaque côté de la carte.[16]

Voici la liste des différentes fonctions associées aux broches de la carte :

- GPIO (Entrées/Sorties Générales) : broches 0 à 63
- EMIF (Interface de mémoire externe) : broches 64 à 79
- CAN (Contrôleur d'Accès Réseau) : broches 80 à 87
- SCI (Interface de communication série synchrone) : broches 88 à 91
- SPI (Interface de communication série) : broches 92 à 95
- I2C (Bus de communication série) : broches 96 à 97
- McBSP (Port série multi-canal) : broches 98 à 111
- PWM (Modulation de largeur d'impulsion) : broches 112 à 123
- QEP (Périphérique de positionnement) : broches 124 à 127
- EQEP (Périphérique d'encodeur de quadrature amélioré) : broches 128 à 131
- XINTF (Interface externe de bus de données) : broches 132 à 147
- USB (Bus série universel) : broches 148 à 151
- CAN (Contrôleur d'Accès Réseau) : broches 152 à 159
- ADC (Convertisseur analogique-numérique) : broches 160 à 167
- SCI (Interface de communication série synchrone) : broches 168 à 171
- SPI (Interface de communication série) : broches 172 à 175

Il est important de noter que certaines broches peuvent avoir des fonctions alternatives en fonction des besoins de l'utilisateur, mais ces fonctions alternatives doivent être programmées via le logiciel de développement approprié pour la carte DSPF28335.

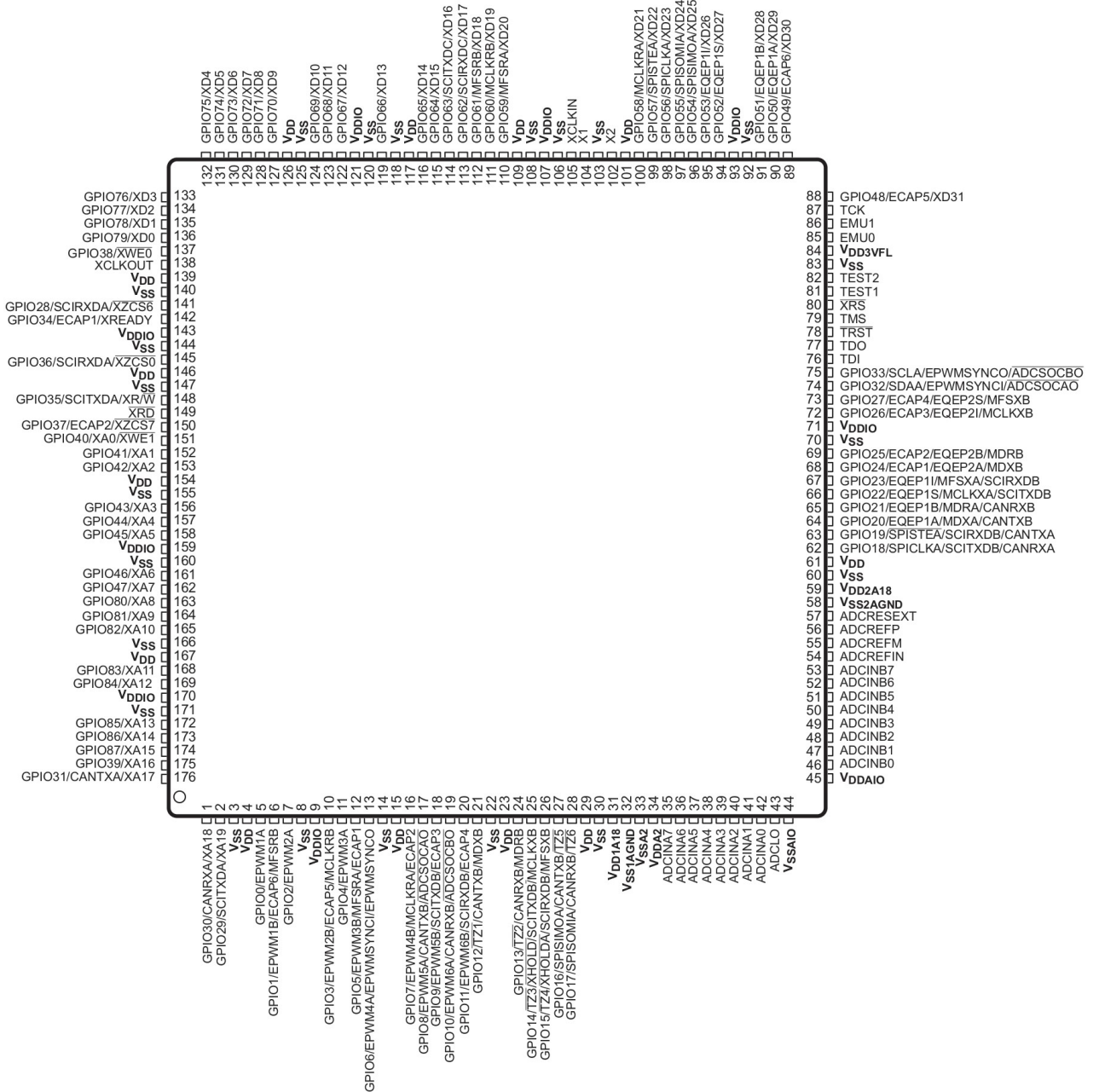


FIGURE III.4 – Diagramme des broches de la carte DSP[16]

### III.5.3 L'unité Centrale de Traitement (CPU)

Il est basé sur l'architecture TMS320C28x. Celle-ci est destinée plus particulièrement aux DSP et apporte des fonctionnalités telles que des instructions de traitement de signal optimisées, des registres de données spécialisés et une gestion efficace de la mémoire. Le CPU de DSP F28335 fonctionne à une fréquence maximale de 150 MHz et dispose d'un pipeline à cinq étages pour améliorer les performances. Il dispose également d'une mémoire cache de 16 Ko pour améliorer l'efficacité du traitement des données. Le processeur DSP F28335 prend en charge une grande variété d'instructions de traitement du signal numérique, incluant les opérations mathématiques, les filtres numériques, les transformations de Fourier, les algorithmes de modulation/dem, et beaucoup plus.



Le CPU Offre de hautes performances et de fonctionnalités avancées pour satisfaire les besoins d'une variété d'applications.

### III.5.4 Les mémoires de DSP

Plusieurs bus servent à l'échange de données entre la mémoire, les périphériques et CPU. L'architecture de bus C28x contient :[16]

- un bus de lecture de programme : qui comprend 22 lignes d'adresse et 32 lignes de données.
- un bus de lecture de données et un bus d'écriture de données : Les bus de données de 32 bits permettent des opérations sur 32 bits à un seul cycle.

Le DSP F28335 possède une combinaison de mémoires programme, RAM, ROM ,des mémoires cache et externes, qui sert au stockage des instructions et des données nécessaires au fonctionnement du DSP,et donc voici la liste des mémoires principales de ce dernier :

1. Mémoire Flash :Cette mémoire est destinée à stocker le code du programme que doit exécuté le DSP. Le DSP F28335 dispose de 512 KB de mémoire Flash.
2. Mémoire RAM interne :Cette mémoire est destinée au stockage des données et des variables temporaires employées par le DSP au lors de son exécution.
3. Mémoire ROM : Cette mémoire est destinée au stockage de données en lecture seule, comme les tableaux des constantes.
4. Mémoire Cache :Le DSP F28335 est doté d'une partie de la mémoire cache intégrée à son processeur, ce qui lui permet un accès plus rapide à la RAM interne.
5. Mémoire externe :Le DSP F28335 comporte une interface de bus de données externe (XINTF), qui autorise la connexion du processeur à des dispositifs de mémoire externes.

### III.5.5 Les périphériques de DSP

Le DSP F28335 dispose de plusieurs périphériques, qui sont les suivants :[16]

- Accès direct à la mémoire (DMA) à 6 canaux.
- Trois temporisateurs CPU 32 bits.
- Jusqu'à six modules PWM améliorés (ePWM1, ePWM2, ePWM3, ePWM4, ePWM5, ePWM6).
- Jusqu'à six modules de capture améliorés (eCAP1, eCAP2, eCAP3, eCAP4, eCAP5, eCAP6).
- Jusqu'à deux modules QEP améliorés (eQEP1, eQEP2).
- Module convertisseur analogique-numérique (ADC) amélioré.
- Jusqu'à deux modules de réseau de contrôleurs amélioré (eCAN) (eCAN-A, eCAN-B).
- Jusqu'à trois modules d'interface de communication série (SCI-A, SCI-B, SCI-C).
- Un module d'interface périphérique série (SPI) (SPI-A).
- Un module de circuit intégré (I2C).



- Jusqu'à deux modules de port série tampon multicanal (McBSP-A, McBSP-B).
- E/S numériques et fonctions de broches partagées.
- Interface externe (XINTF).

### III.5.5.1 Module convertisseur analogique digital (ADC)

Le bloc ADC est un convertisseur de 12 bits, à terminaison unique, à 16 canaux. Il contient deux unités d'échantillonnage et de maintien pour l'échantillonnage simultané. Les registres ADC sont pris en charge par le DMA afin de réduire la charge de travail liée à l'entretien de ce périphérique.

L'une des unités périphériques les plus significatives d'un contrôleur embarqué, car elle constitue une interface importante entre le contrôleur et le monde réel.

### III.5.5.2 Modulation de largeur d'impulsion PWM

Le périphérique PWM amélioré d'un DSP F28335 permet de générer des signaux PWM indépendants ou complémentaires. Il offre également la possibilité de régler une bande morte pour les fronts d'attaque et de fuite, ainsi qu'un mécanisme de déclenchement latché et cycle par cycle. Certaines broches PWM prennent également en charge les fonctions HRPWM. De plus, les registres ePWM sont pris en charge par le DMA pour alléger la charge de travail liée à ce périphérique.

La fonction de modulateur de largeur d'impulsion (PWM) est très courante dans les systèmes embarqués. Il peut générer une forme d'onde périodique pour le contrôle d'un moteur ou peut agir en tant que convertisseur numérique-analogique avec quelques composants externes. Ce périphérique PWM est essentiellement un temporisateur avec un compteur de période et un comparateur de durée de première phase où la largeur de la période et la durée de la première phase sont toutes deux programmables.

Le PWM présente les caractéristiques suivantes :[16]

- Compteur de période 32 bits.
- Compteur de durée de première phase 32 bits.
- Configurable pour fonctionner soit en mode one-shot, qui génère une seule interruption à la fin de l'opération, soit en mode continu, qui génère une interruption par période.
- Compteur de répétition 8 bits pour le fonctionnement en mode one-shot. Le mode one-shot produit  $N + 1$  périodes de la forme d'onde, où  $N$  est le nombre de répétitions.
- État inactif de la broche de sortie PWM configurable.
- Interruption et événements de synchronisation EDMA.
- Prise en charge de l'émulation pour les opérations d'arrêt ou de marche libre.

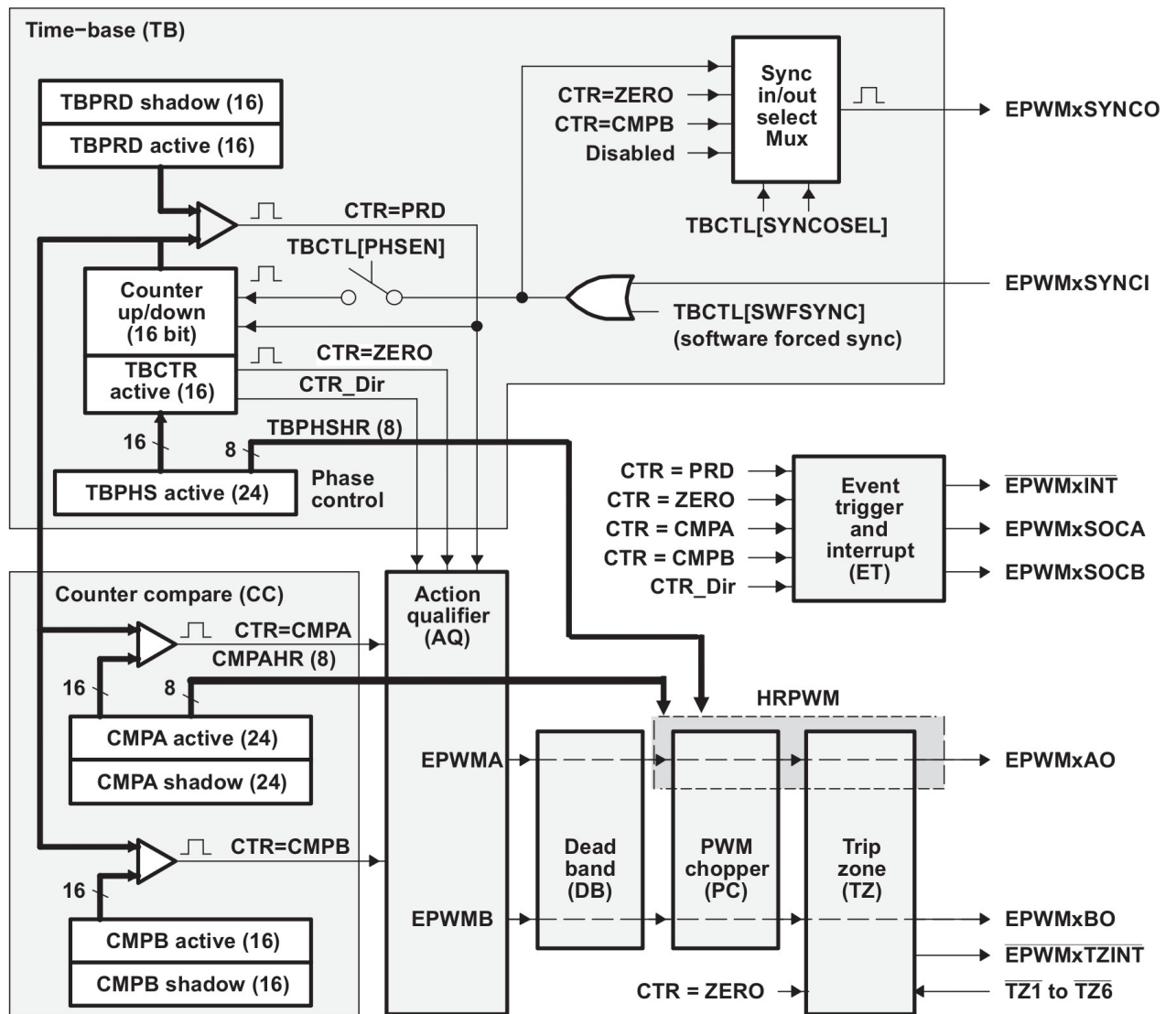


FIGURE III.5 – Les interconnexions des signaux avec l'ePWM[16]

### III.5.5.3 Les Entrées/Sorties à usage général (GPIO)

Le F28335 possède plusieurs broches d'entrée/sortie à usage général (GPIO), qui peuvent être programmées pour différentes fonctions selon les besoins de l'application. 88 ports GPIO sont configurables en tant qu'entrées ou sorties numériques.[27]

### III.5.5.4 Module d'interface série asynchrone (SCI)

L'interface de communication série est un port série asynchrone à 2 fils, généralement appelé UART. Elle comporte une FIFO de réception et d'émission à 16 niveaux permettant de réduire la charge de travail associée à la gestion des interruptions. Cette interface garantit les communications numériques entre l'unité centrale (CPU) et les autres périphériques asynchrones qui utilisent la norme NRZ (non-return-to-zero).[16]

### III.5.5.5 Le Périphérique de capture amélioré (eCAP)

Le périphérique de capture amélioré d'un DSP est un module matériel avancé qui permet la capture de signaux d'entrée en temps réel avec une grande précision temporelle. Ce module est très souvent utilisé pour la mesure et le contrôle des signaux dans les applications de commande de moteur, de conversion de puissance et de traitement des signaux en temps réel.

Il peut avoir plusieurs canaux d'entrée, ce qui permet la capture simultanée de plusieurs signaux.

Il utilise un compteur à haute résolution qui permet de mesurer le temps avec une grande précision, plus une base de temps de 32 bits et enregistre jusqu'à quatre événements programmables en mode continu ou unique. Il peut également être configuré pour générer un signal PWM auxiliaire.[16]

### III.5.5.6 Codeur à quadrature d'impulsion (QEP)

Le QEP est un module périphérique matériel qui fournit une mesure précise de la position, de la vitesse et de l'accélération d'un moteur CC, d'un codeur, d'un capteur de position ou d'un autre dispositif de rétroaction rotatif.

Le module QEP du DSP F28335 peut prendre en charge jusqu'à 4 codeurs quadrature haute résolution et offre une interface matérielle dédiée pour la lecture des signaux des codeurs à 2 canaux (A et B) et d'un canal d'index. Le module QEP comprend également un compteur de position, un compteur de vitesse et un compteur d'accélération interne pour le traitement des signaux des codeurs.

Il prend en charge la détection du sens de rotation et l'émulation du codeur.[17]

### III.5.5.7 Temporisateurs CPU

Les temporisateurs CPU-Timer 0, 1 et 2 sont des temporisateurs 32 bits qui ont les mêmes spécifications, telles que des périodes pré réglables et une mise à l'échelle de l'horloge 16 bits.[16] Le registre de comptage 32 bits des temporisateurs produit une interruption une fois que le compteur atteint zéro.

Le CPU-Timer 1 est destiné à un usage général et peut être connecté à l'INT13 du CPU. Le CPU-Timer 2 est réservé aux applications Real-Time OS (RTOS)/BIOS. Elle est connectée à l'INT 14 de l'unité centrale.

Le CPU-Timer 0 est également destiné à un usage général et est connecté au bloc PIE.

### III.5.5.8 Temps mort Dead-band(DB)

Le circuit de ce registre permet de programmer un temps mort à chaque période du signal PWM, afin d'éviter les courts-circuits transitoires des interrupteurs dans le même bras de l'onduleur.[16]

### III.5.5.9 Watchdog

Le bloc chien de garde est couramment utilisé dans les systèmes critiques de sécurité, où il est important de s'assurer que le système fonctionne dans des limites sûres et qu'il

peut détecter et répondre aux pannes en temps voulu. Le bloc chien de garde est également utile pour surveiller les performances du système et pour diagnostiquer les problèmes dans les systèmes complexes.

Le rôle principal du chien de garde (WDT) est de surveiller en permanence l'exécution du programme et de prendre des mesures appropriées en cas de défaillance ou de blocage du système.

### III.5.6 Trames périphériques 0, 1, 2, 3 (PFn)

Le dispositif divise les appareils en quatre sections. Le mapping des appareils est le suivant : [16]

**PF0**  $\implies$

**PIE** : Registres de contrôle et d'activation des interruptions PIE plus table vectorielle PIE.

**Flash** : Registres d'état d'attente de la mémoire flash.

**XINTF** : Registres d'interface externe.

**DMA** : Registres DMA.

**Timers** : Registres CPU-Timers 0, 1, 2.

**CSM** : Module de sécurité du code Registres KEY.

**ADC** : Registres de résultats ADC (double mappage).

**PF1**  $\implies$

**eCAN** : Registres de boîte aux lettres et de contrôle eCAN.

**GPIO** : Registres de configuration et de contrôle du MUX GPIO.

**ePWM** : Module et registres de modulation de largeur d'impulsion améliorée (double correspondance).

**eCAP** : Module et registres de capture améliorés.

**eQEP** : Module et registres d'impulsion de codeur en quadrature amélioré.

**PF2**  $\implies$

**SYS** : Registres de contrôle du système.

**SCI** : Registres de contrôle et RX/TX de l'interface de communication série (SCI).

**SPI** : Interface de port série (SPI) Registres de contrôle et RX/TX.

**ADC** : Registre d'état, de contrôle et de résultat de l'ADC.

**I2C** : Module et registres de circuits intégrés.

**XINT** : Registres d'interruption externe.

**PF3**  $\implies$

**McBSP** : Registres du port série tampon multicanal McBSP.

**ePWM** : Modulateur de largeur d'impulsion amélioré (double mappage).

## III.6 Arithmétique de TMS320F28335

On distingue deux types de DSP : ceux à virgule fixe et ceux à virgule flottante.

### III.6.1 Les DSP à virgule fixe

Les données sont représentées sous forme de nombres décimaux à virgule fixe (exemple :- 1.0 à +1.0) , ou d'entiers réguliers. La représentation de ces nombres fractionnaires est basée sur la méthode du complément à deux. L'avantage de cette représentation est qu'elle permet facilement l'addition binaire de nombres positifs et négatifs.[28]

### III.6.2 Les DSP à virgule flottante

les données sont représentée sen utilisant une mantisse et un exposant. La représentation de ces nombres s'effectue selon la formule suivante :  $n = \text{mantisse} \times 2^{\text{exposant}}$ . Généralement, la mantisse est un nombre fractionnaire (-1.0 à +1.0), et l'exposant est un entier indiquant la place de la virgule en base 2 (c'est le même mécanise qu'en base 10).[28]

## Conclusion

Le contrôle du régime d'un moteur par un DSP exige l'adaptation et le conditionnement des signaux d'entrée et de sortie. Pour ce faire, il est nécessaire de concevoir et d'implémenter une interface matérielle spécifique à l'aide de composants appropriés. Cependant, le bon fonctionnement du DSP dépend également de paramètres directement liés au moteur, tels que les informations sur les valeurs de courant de phase qui sont traitées par les convertisseurs analogique-numérique (ADC), ainsi que les informations instantanées sur la vitesse du moteur qui sont capturées par les détecteurs de position QEP. De plus, la génération de signaux PWM (modulation de largeur d'impulsion) doit être assurée en temps réel par le DSP pour un contrôle efficace de la vitesse du moteur.

En bref, le contrôle de la vitesse des moteurs par un DSP exige une attention particulière à de nombreux paramètres pour assurer un fonctionnement optimal.

L'utilisation de la carte DSP dans la commande de moteur MSAP représente une approche puissante et efficace pour obtenir un contrôle précis, réactif et adaptatif du moteur. Sa capacité de traitement numérique avancée, sa programmabilité et sa flexibilité en font un outil essentiel pour le développement de systèmes de commande de moteur MSAP performants et fiables.

Ce chapitre a montré que ce DSP est une bonne solution pour les applications de commande de moteur numérique et est parfait pour notre projet.

---

---

# CHAPITRE IV

---

## ÉTUDE DES PRINCIPAUX MODULES POUR LA COMMANDE

### Introduction

Ce chapitre constitue une étape importante vu qu'il vise à concrétiser les concepts et les designs théoriques en une réalité fonctionnelle. Dans ce chapitre, nous aborderons les aspects pratiques tant au niveau matériel que logiciel.

L'implémentation nécessite une compréhension approfondie de tous les aspects théoriques donc nous explorerons les principaux aspects du module ePWM, ses fonctionnalités générales et ses différentes configurations. Nous examinerons également ces modules un par un en détaillant chaque module et ses fonctionnalités spécifiques, en mettant l'accent sur les aspects pratiques de leur implémentation.

la deuxième partie se concentre sur le module eQEP (Enhanced Quadrature Encoder Pulse) qui est un module très important dans notre système. tout d'abord nous commencerons par donner un aperçu général des encodeurs incrémentaux, Nous aborderons également l'architecture de ce module, Nous expliquerons différentes formes d'impulsion finale présentes dans les codeurs en quadrature, ces différentes entrées, y compris les broches pour le mode d'horloge en quadrature ou le mode de comptage de direction, l'entrée d'index et l'entrée stroboscopique. Nous expliquerons leur rôle et leur utilisation dans le contexte de l'eQEP.

nous présenterons également la description fonctionnelle du module eQEP, en mettant en évidence les principales unités fonctionnelles qui le composent, telles que la qualification d'entrée, l'unité de décodage en quadrature, le compteur de position, l'unité de capture des fronts en quadrature, la base de temps et la minuterie chien de garde.

On dernier temps, nous aborderons une vision globale sur les ADC en détaillant ces différents registres et la configuration de l'ADC dans CCS pour le contrôle des MSAP.

## IV.1 Module ePWM

### IV.1.1 Généralités [5]

Le module ePWM représente un canal PWM complet composé de deux sorties PWM : **EPWMxA** et **EPWMxB**. Plusieurs modules ePWM sont instanciés dans un appareil. Chaque instance ePWM est identique à une exception près. Certaines instances comprennent une extension matérielle qui permet un contrôle plus précis des sorties PWM.

### IV.1.2 les modules de EPWM

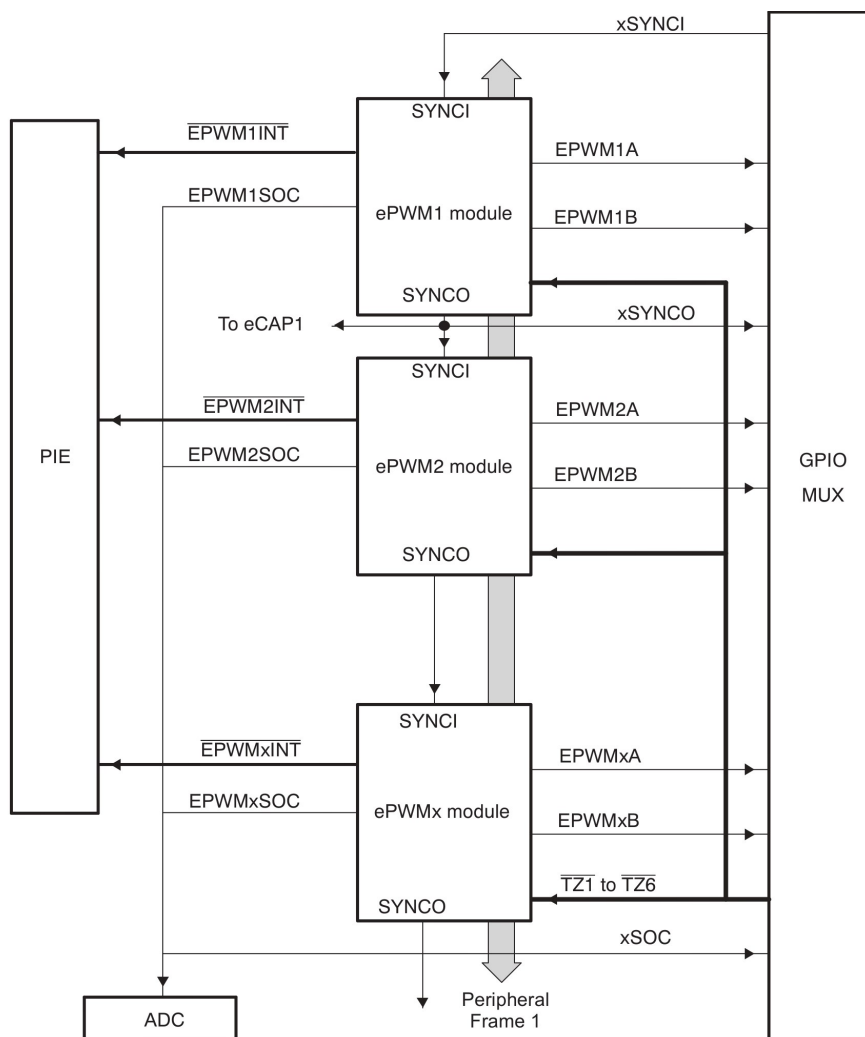


FIGURE IV.1 – les modules de EPWM[5]

Le microcontrôleur TMS320F28335 intègre 6 modules EPWM qui sont interconnectés pour faciliter la synchronisation et la gestion des opérations. Chaque module EPWM est synchronisé par un signal d'entrée appelé  $xSYNCl$ , et la sortie  $SYNCO$  du module précédent est utilisée comme l'entrée du module suivant.

Chaque module est synchronisé par un signal de synchronisation d'entrée  $xSYNCl$ , et génère 4 signaux de sorties (ces signaux sont tous interconnectés et permettent de contrôler,

synchroniser et gérer les opérations du module ePWM) :

**L'EPWMA (ePWMxA)** : permet un contrôle précis de la synchronisation et du rapport cyclique du signal, généralement utilisé pour contrôler la vitesse d'un moteur.

**L'EPWMB (ePWMxB)** est déphasé de 180 degrés par rapport à l'EPWMA et génère un signal complémentaire, ce qui est utile dans les applications nécessitant des signaux déphasés.

**EPWMxINT** : Ce signal est une demande d'interruption générée par le module ePWM. Il peut être utilisé pour déclencher une interruption de l'unité centrale lorsqu'une certaine condition est remplie, par exemple lorsqu'un compteur de temps atteint une certaine valeur ou qu'un événement de comparaison se produit. L'interruption peut être utilisée pour effectuer des actions ou exécuter du code en réponse à l'événement ePWM.

Les signaux **EPWMxSOCA** et **EPWMxSOCB** jouent un rôle important dans la synchronisation du module ePWM avec des dispositifs ou des signaux externes. En configurant ces signaux, vous pouvez générer des impulsions ou déclencher des événements à des points spécifiques de la forme d'onde ePWM, tels que le début d'une période ou une phase spécifique. Cette capacité de synchronisation est particulièrement utile pour coordonner la synchronisation de plusieurs signaux PWM ou pour déclencher d'autres appareils en réponse à la sortie ePWM.

Outre leur fonction de synchronisation, EPWMxSOCA et EPWMxSOCB peuvent également servir de déclencheurs pour les conversions ADC ou d'autres événements nécessitant une synchronisation avec la forme d'onde PWM. Cette synchronisation garantit un échantillonnage précis des données ou permet une coordination précise entre le PWM et d'autres opérations du système.

### IV.1.3 Time-base (TB)

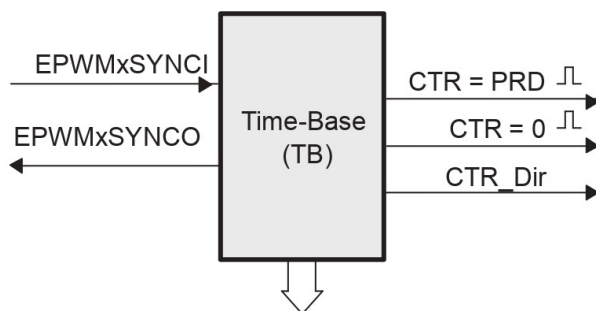


FIGURE IV.2 – time base TB [5]

Le bloc TB (Time-Base) du module ePWM du TMS320F28335 est responsable de la génération de la fréquence de base pour les signaux PWM. Il offre plusieurs fonctionnalités configurables, telles que la mise à l'échelle de l'horloge de base, la configuration de la fréquence ou de la période du compteur de base de temps, et le réglage du mode de comptage. Il permet également la synchronisation avec d'autres modules ePWM, la configuration de la phase de base de temps, la spécification de

la source de sortie de synchronisation et la définition du comportement du compteur lors



de l'arrêt de l'appareil. Ces fonctionnalités offrent une grande flexibilité pour configurer et contrôler le compteur de base de temps et le comportement de synchronisation du module ePWM.

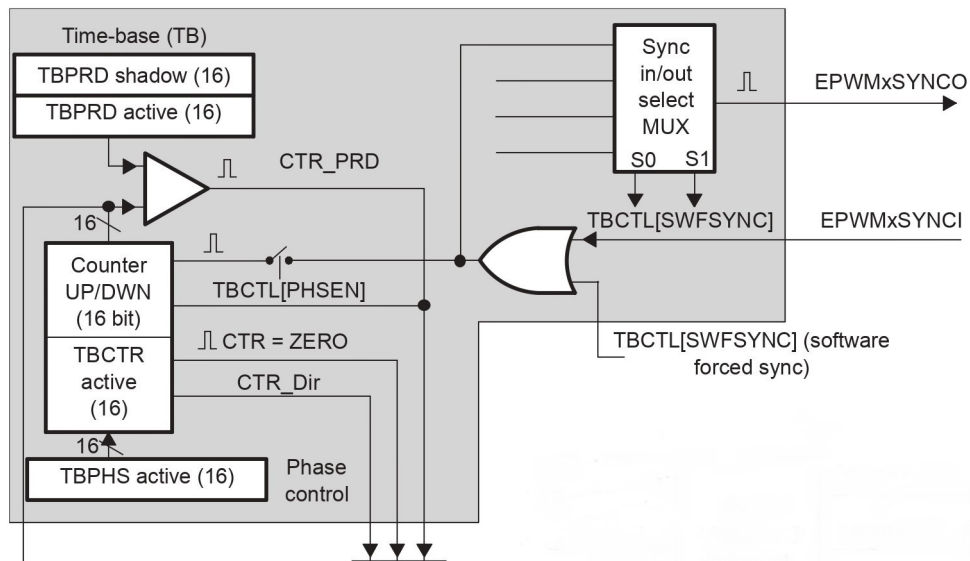


FIGURE IV.3 – Architecture interne de TB[5]

Ce bloc pour définir la période et le déphasage et la direction de comptage de epwm qu'ils vont transférer au bloc **Counter-compare (CC)** IV.1.4

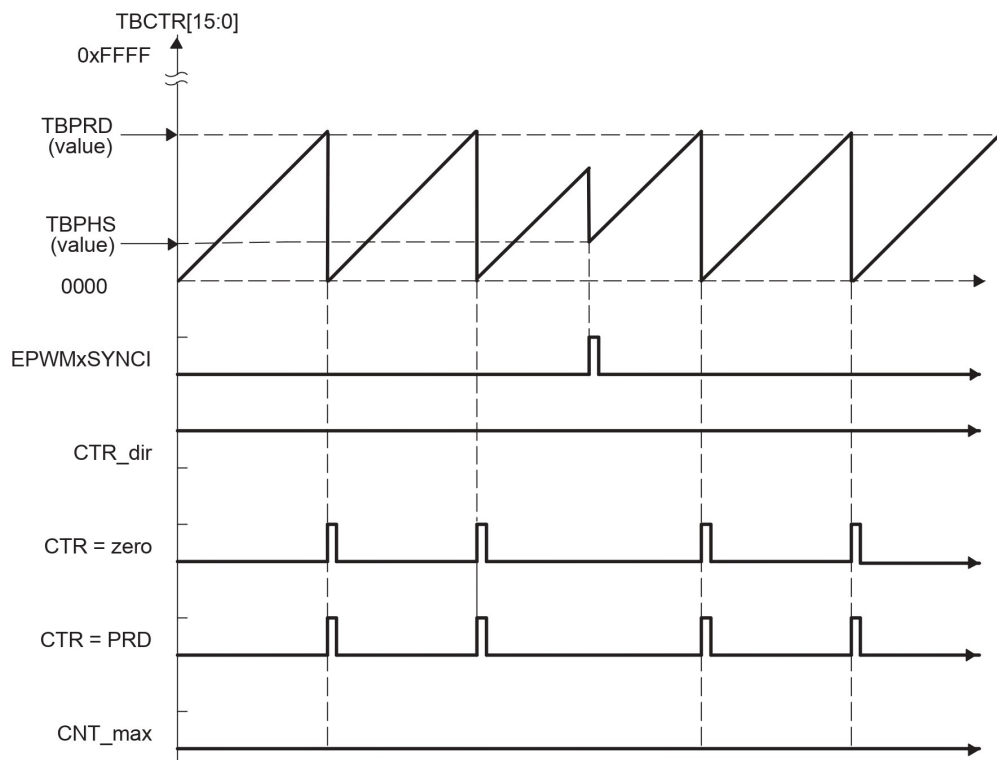


FIGURE IV.4 – chronogramme de Time-Base Up-Count Mode [5]

Dans le mode Time-Base Up-Count du module ePWM, un compteur incrémente à

partir de 0 jusqu'à la valeur de la période de base, puis revient à 0 pour commencer un nouveau cycle. Lorsque le compteur atteint la valeur de la période de base, une impulsion est générée, signalant le début d'une nouvelle période. Cette impulsion peut être utilisée pour synchroniser d'autres périphériques ou modules du système. Lorsqu'un signal de synchronisation externe est reçu, le compteur s'arrête et prend la valeur d'un registre spécifique. Cela permet de synchroniser le module ePWM avec d'autres modules ou signaux, en démarrant une nouvelle période avec un décalage de phase précis.

#### IV.1.4 Counter-compare (CC)

consiste à comparer la valeur actuelle du compteur avec une valeur de comparaison programmée.

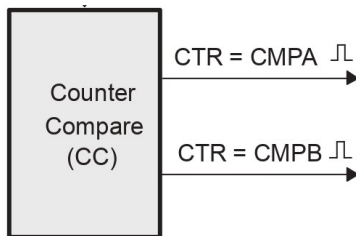


FIGURE IV.5 – counter compare[5]

- Spécifier le rapport cyclique PWM pour la sortie EPWMxA et/ou la sortie EPWMxB.

- Spécifier la synchronisation des événements de commutation sur les sorties EPWMxA ou EPWMxB.

Comparateur de sortie (CMP) : ce bloc fonctionnel est chargé de comparer la valeur de référence avec la valeur du compteur de la minuterie de base. Il est configuré à l'aide du registre CMPCTL et peut être utilisé pour générer des signaux PWM en mode haut, bas ou haut-bas.

**PWM Outputs (PWM A and PWM B) :** Ces blocs fonctionnels sont chargés de générer des signaux PWM sur la base des événements de comparaison détectés par le comparateur de sortie. Les durées des impulsions PWM sont définies à l'aide des registres CMPAHR, CMPALR, CMPBHR et CMPBLR.

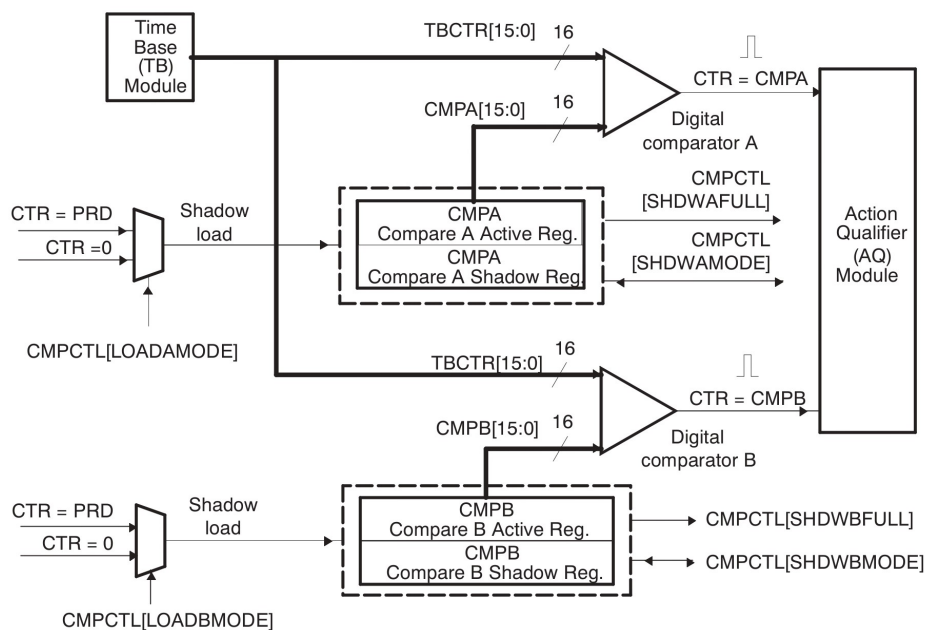


FIGURE IV.6 – Architecture interne de CC[5]

Les registres de counter-compare CMPA et CMPB ont chacun un registre shadow associé. Le registre shadow permet de synchroniser les mises à jour des registres avec le matériel. Lorsque le registre shadow est utilisé, les mises à jour des registres actifs ne se produisent qu'à des points stratégiques. Cela permet d'éviter la corruption ou les opérations parasites dues à la modification asynchrone du registre par le logiciel. L'adresse mémoire du registre actif et du registre shadow est identique. Les bits CMPCTL[SHDWAMODE] et CMPCTL[SHDWBMODE] déterminent quel registre est écrit ou lu. Ces bits activent et désactivent respectivement le registre shadow CMPA et le registre shadow CMPB.

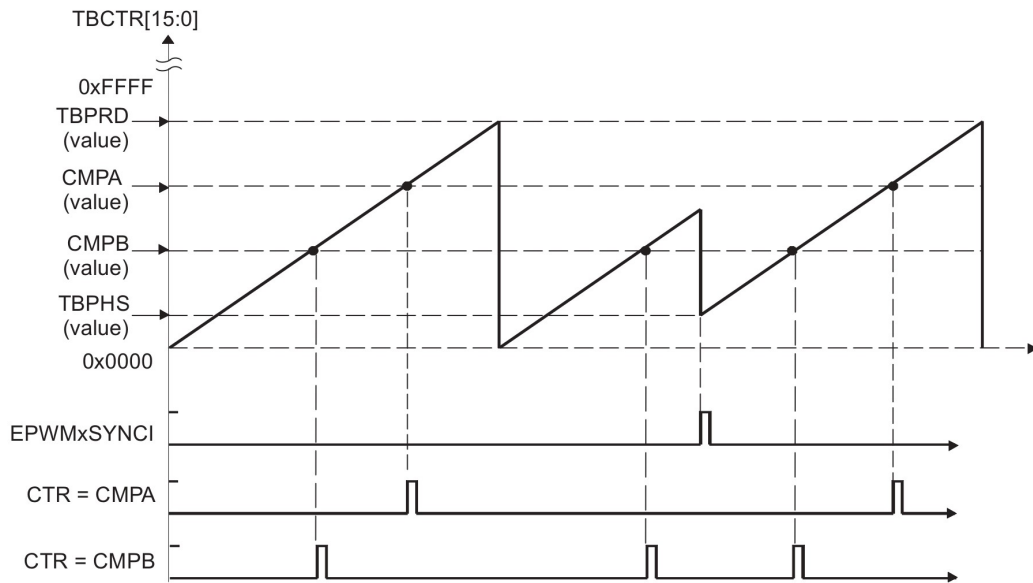


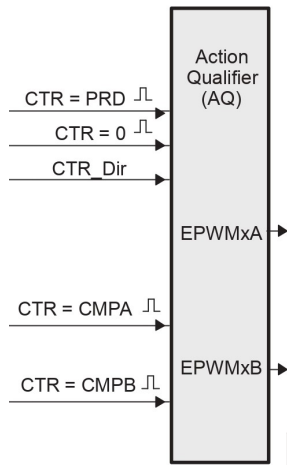
FIGURE IV.7 – Counter-compare Up-Count Mode Waveforms[5]

Dans le mode de comptage Up-Count du module ePWM, une forme d'onde est générée par l'incréméntation d'un compteur à partir de 0 jusqu'à la valeur de la période de base, puis il se remet à 0 pour commencer un nouveau cycle. Lorsque le compteur atteint les valeurs de comparaison CMPA et CMPB, des impulsions sont générées sur les sorties correspondantes. Ces impulsions peuvent être utilisées pour déclencher des actions ou des événements dans d'autres parties du système. En ajustant les valeurs de CMPA et CMPB, il est possible de contrôler précisément le moment où les impulsions sont générées, offrant ainsi une grande flexibilité dans la configuration du module ePWM et les actions qui en découlent.

### IV.1.5 The Action-Qualifier (AQ)

les actions à prendre lorsque des événements spécifiques se produisent dans le sous-module de la base de temps ou de la contre-comparaison.

Le qualificateur d'action (AQ) permet de définir Voici les actions disponibles qui peuvent être configurées :



- Aucune action n'est prise : La sortie PWM reste inchangée.
- La sortie EPWMxA et/ou EPWMxB est activée : La (les) sortie(s) PWM correspondante(s) est (sont) réglée(s) à un niveau logique élevé.
- Sortie EPWMxA et/ou EPWMxB commutée à l'état bas : La (les) sortie(s) PWM correspondante(s) est (sont) réglée(s) sur un niveau logique bas.
- Sortie EPWMxA et/ou EPWMxB commutée : La ou les sorties PWM correspondantes changent d'état, soit de haut en bas, soit de bas en haut.
- Forcer l'état de la sortie PWM par commande logicielle : Le logiciel contrôle directement l'état de la sortie PWM.
- Configurer et contrôler la dead-band du PWM par logiciel :

FIGURE IV.8 – AQ[5] La bande morte, qui introduit un délai entre les fronts montant et descendant des signaux PWM, peut être programmée et contrôlée par logiciel.

Le sous-module "Action-Qualifier" prend en entrée les événements liés au zéro, à la comparaison A et à la comparaison B. En fonction de ces événements, il génère des signaux de sortie selon les actions définies, telles que le réglage, l'effacement, le basculement ou le forçage de la sortie PWM. Ce sous-module offre une grande souplesse dans le contrôle du comportement des sorties PWM et permet un contrôle précis de la génération de la bande morte PWM.

| S/W force | TB Counter equals: |         |         |        | Actions    |
|-----------|--------------------|---------|---------|--------|------------|
|           | Zero               | Comp A  | Comp B  | Period |            |
| SW<br>X   | Z<br>X             | CA<br>X | CB<br>X | P<br>X | Do Nothing |
| SW<br>↓   | Z<br>↓             | CA<br>↓ | CB<br>↓ | P<br>↓ | Clear Low  |
| SW<br>↑   | Z<br>↑             | CA<br>↑ | CB<br>↑ | P<br>↑ | Set High   |
| SW<br>T   | Z<br>T             | CA<br>T | CB<br>T | P<br>T | Toggle     |

FIGURE IV.9 – Possible AQ Actions for EPWMxA and EPWMxB Outputs [5]

Dans le module Action Qualifier (AQ) de l'ePWM du TMS320F28335, les différentes actions (zero, comp a, comp b, period) sont des conditions qui déterminent les actions à prendre sur les signaux de sortie PWM (EPWMxA et EPWMxB) en fonction des comparaisons effectuées avec les registres de comparaison (CMPA et CMPB) et le registre de période (PRD).

Voici le principe de fonctionnement des différentes actions :

Action "zero" (Z) : Lorsque le compteur de comparaison (CTR) atteint la valeur zéro (0), cette action est déclenchée. Elle permet de prendre une action spécifique sur les signaux de sortie PWM, telle que la mise à un niveau logique bas (LOW) ou la génération d'une impulsion.

Action "comp a" (CA)/"comp b" (CB) : Lorsque le CTR est égal au registre de comparaison CMPA/comparaison CMPB, cette action est déclenchée. Elle permet de prendre une action spécifique sur les signaux de sortie PWM, telle que la mise à un niveau logique bas (LOW) ou la génération d'une impulsion.

Action "period" (P) : Lorsque le CTR atteint la valeur du registre de période PRD, cette action est déclenchée. Elle permet de prendre une action spécifique sur les signaux de sortie PWM, telle que la mise à un niveau logique bas (LOW) ou la génération d'une impulsion.

Le principe de fonctionnement de l'AQ est d'évaluer en permanence les comparaisons entre le CTR et les registres CMPA, CMPB et PRD. Lorsque l'une de ces conditions est remplie, l'action correspondante est activée et les signaux de sortie PWM sont modifiés en conséquence.

Cela permet de contrôler précisément la synchronisation et le rapport cyclique des signaux PWM, offrant ainsi un contrôle flexible et précis de la modulation PWM dans les applications telles que le contrôle de vitesse des moteurs.

#### IV.1.6 Dead-Band Generator (DB)

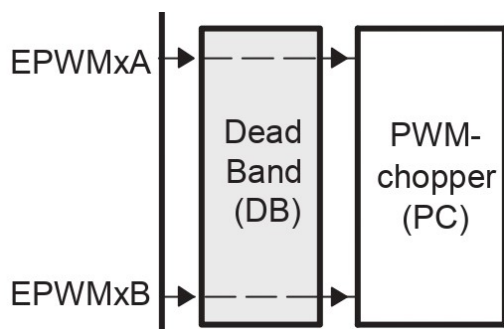


FIGURE IV.10 – dead band (DB)[5]

Le générateur de dead-band (DB) est responsable de l'élimination des transitions non désirées sur les sorties PWM. Il est configuré à l'aide du registre DBCTL et peut être utilisé pour éviter les courts-circuits ou les interférences électromagnétiques. Les entrées de ce bloc comprennent le mode dead-band, qui détermine le comportement des retards dead-band, ainsi que les retards des fronts montants et descendants. Le bloc DB génère les retards appropriés pour les sorties PWM, en veillant à ce qu'il y ait une séparation suffisante entre les fronts montants et descendants pour éviter tout chevauchement ou interférence indésirable.

**Dead-band (DB) :**

- Contrôle de la relation traditionnelle de dead-band complémentaire entre les interrupteurs supérieurs et inférieurs.

- Spécifier la valeur du délai du front montant de la sortie
- Spécifier la valeur de la temporisation du front descendant de la sortie
- Contourner entièrement le module dead-band. Dans ce cas, la forme d'onde PWM est transmise sans modification.

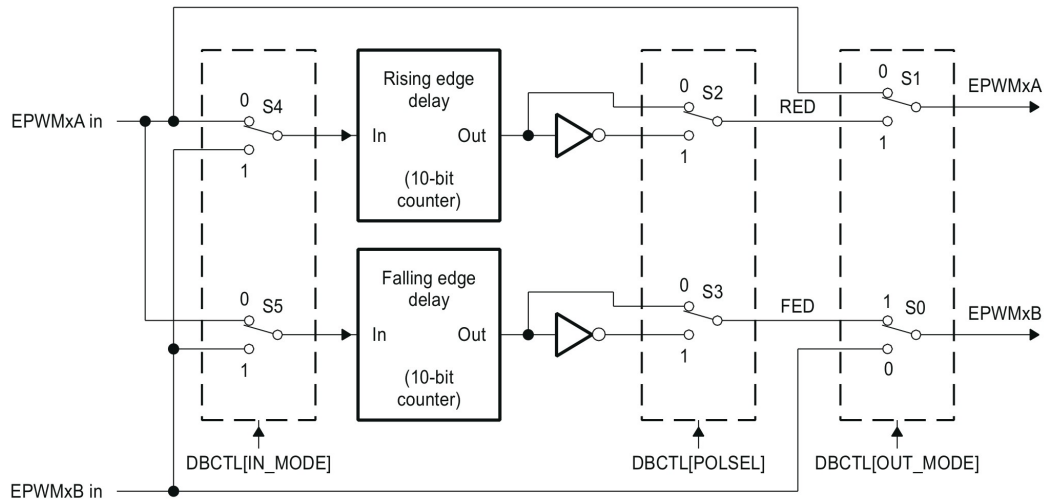


FIGURE IV.11 – Configuration Options for the Dead-Band Submodule [5]

**Génération de paires de signaux :** Le module dead-band prend une seule entrée EPWMxA et génère des paires de signaux appropriées, à savoir EPWMxA et EPWMxB, avec une relation dead-band. Ces paires de signaux sont utilisées pour contrôler des sorties complémentaires ou des signaux synchronisés.

**Programmation des paires de signaux :** Le module dead-band permet de programmer des paires de signaux pour différentes configurations, notamment active high (AH), active low (AL), active high complementary (AHC) et active low complementary (ALC). Ces configurations définissent la relation entre les signaux EPWMxA et EPWMxB.

**Adding delay to rising edges RED :** Le module dead-band peut introduire un retard programmable sur les fronts montants des signaux EPWM. Ce retard permet d'ajuster la relation temporelle entre les deux signaux de la paire de signaux.

**Adding delay to falling edges FED :** De même, le module dead-band peut également ajouter un retard programmable aux fronts descendants des signaux EPWM. Ce retard permet d'affiner la relation temporelle entre EPWMxA et EPWMxB.

**Contournement du module dead-band :** Le module dead-band peut être complètement contourné, ce qui permet au signal EPWMxA de passer directement à la sortie sans aucun effet de dead-band. Ce mode de contournement est représenté par les lignes pointillées dans le diagramme.

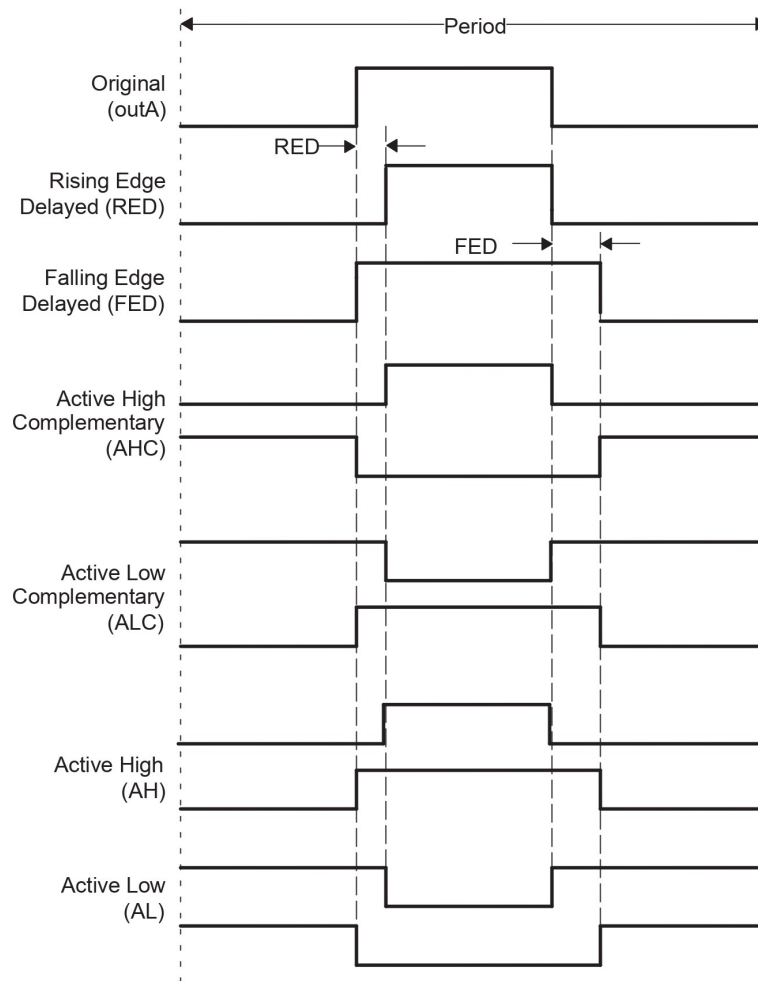


FIGURE IV.12 – Dead-Band Waveforms for Typical Cases ( $0\% < \text{Duty} < 100\%$ ) [5]

**1. Rising Edge Delayed (RED)** : Cela fait référence à l'ajout d'un retard programmable aux fronts montants des signaux EPWMxA et EPWMxB. Ce retard est configuré pour retarder le front montant d'un des signaux par rapport à l'autre. Cela permet de contrôler le décalage temporel entre les deux signaux dans une paire de signaux, ce qui est utile pour les applications nécessitant une synchronisation précise.

**2. Falling Edge Delayed (FED)** : Cela fait référence à l'ajout d'un retard programmable aux fronts descendants des signaux EPWMxA et EPWMxB. Ce retard est configuré pour retarder le front descendant d'un des signaux par rapport à l'autre. Cela permet également de contrôler le décalage temporel entre les deux signaux dans une paire de signaux.

**3. Active High Complementary (AHC)** : Dans ce mode, le signal EPWMxA est actif à l'état haut (niveau logique élevé), tandis que le signal EPWMxB est son complément actif, c'est-à-dire qu'il est actif à l'état bas (niveau logique bas). Cela crée une paire de signaux complémentaires où l'un est actif lorsque l'autre est inactif.

**4. Active Low Complementary (ALC)** : C'est l'inverse de AHC. Dans ce mode, le signal EPWMxA est actif à l'état bas (niveau logique bas), tandis que le signal EPWMxB est son complément actif, c'est-à-dire qu'il est actif à l'état haut (niveau logique élevé).

**5. Active High (AH)** : Dans ce mode, le signal EPWMxA est actif à l'état haut

(niveau logique élevé) sans être complémentaire à un autre signal. Il n'y a pas de signal complémentaire associé à EPWMxA.

**6. Active Low (AL) :** Dans ce mode, le signal EPWMxA est actif à l'état bas (niveau logique bas) sans être complémentaire à un autre signal. Il n'y a pas de signal complémentaire associé à EPWMxA.

#### IV.1.7 PWM Chopper Control (PCCTL)

le sous-module PWM Chopper Control (PCCTL) est responsable de la réduction des pertes de commutation sur les sorties PWM. Il y parvient en ajustant le mode, la fréquence et le cycle de travail du hacheur afin d'optimiser l'efficacité énergétique. Le sous-module PCCTL reçoit des entrées et génère des signaux de commande qui régulent activement le fonctionnement du hacheur, en minimisant les transitoires de tension et en améliorant les performances du système.

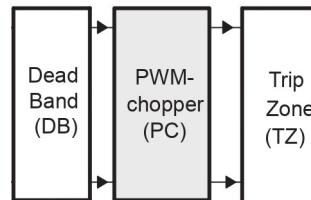


FIGURE IV.13 – PWM chopper(PC)[5]

La forme d'onde **PSCLK** du module EPWM sert de signal de synchronisation pour coordonner le fonctionnement de plusieurs modules et dispositifs EPWM dans un système. Il assure l'alignement temporel et la synchronisation des phases entre les modules, ce qui permet un contrôle précis et des actions coordonnées. Les fronts montants et descendants de la forme d'onde déclenchent des événements au sein du système. Le sous-module EPWM Chopper Control améliore encore les performances en optimisant l'efficacité énergétique et en réduisant les transitoires de tension sur les sorties PWM. Ensemble, ces fonctionnalités permettent un contrôle précis et synchronisé des signaux PWM dans des applications telles que la commande de moteurs et la conversion d'énergie.

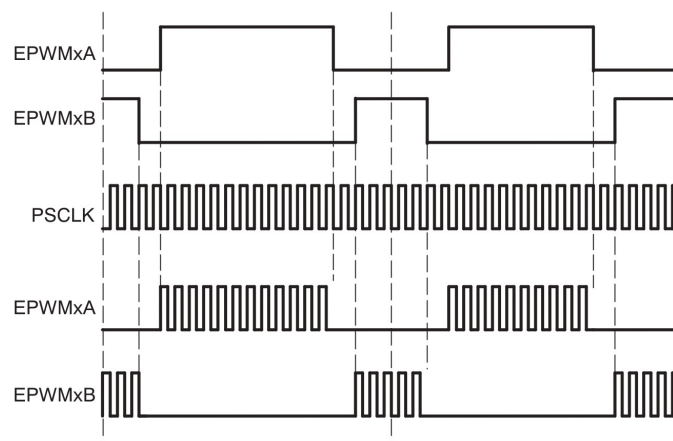


FIGURE IV.14 – chronogramme du PWM-Chopper uniquement l'action de hachage [5]



Les signaux EPWMxA et EPWMxB sont synchronisés avec le signal PSCLK pour assurer une opération coordonnée.

Le signal PSCLK est utilisé pour diviser la fréquence de base du module EPWM. Cela permet d'obtenir des signaux EPWMxA et EPWMxB avec des fréquences réduites par rapport à la fréquence de base.

En utilisant le signal PSCLK synchronisé et la fréquence divisée, le module de contrôle du hacheur génère les signaux EPWMxA et EPWMxB. Ces signaux sont généralement déphasés de 180 degrés l'un par rapport à l'autre pour un contrôle précis de la modulation en largeur d'impulsion (PWM).

Ainsi, en utilisant le signal PSCLK et en configurant correctement le module de contrôle du hacheur, on peut obtenir les signaux EPWMxA et EPWMxB dans le TMS320F28335. Ces signaux sont essentiels pour contrôler et moduler avec précision le rapport cyclique des signaux PWM dans différentes applications.

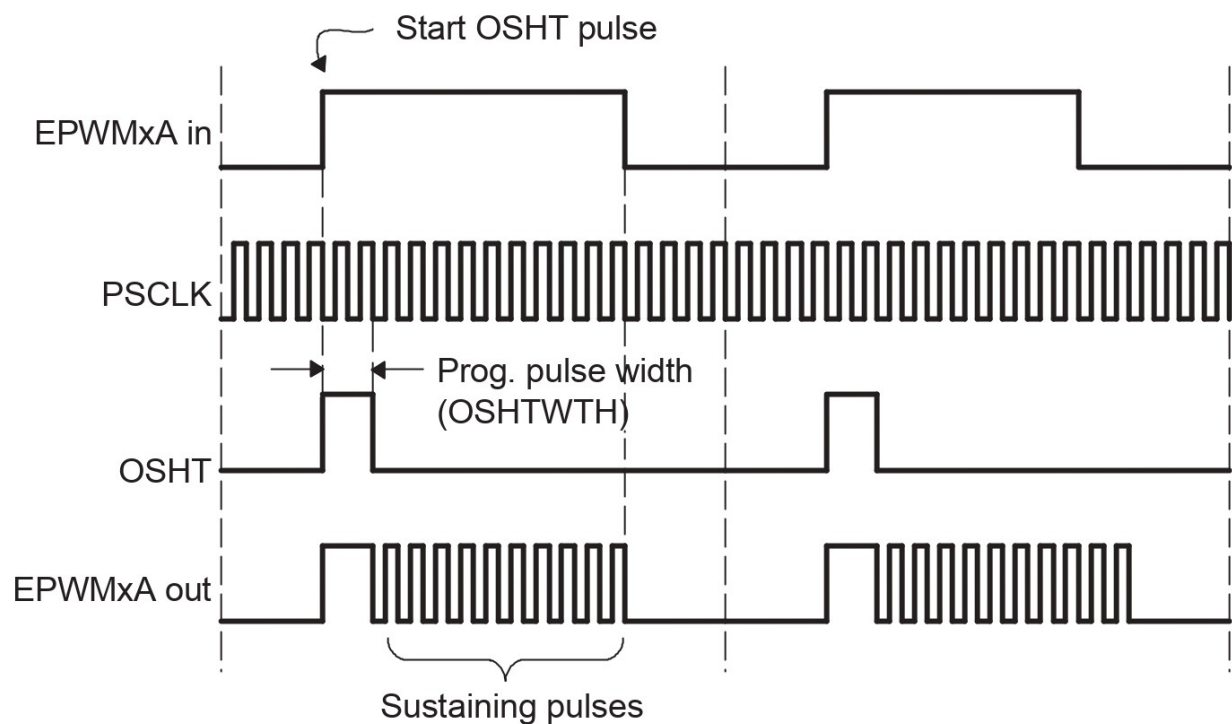


FIGURE IV.15 – chronogramme du pwm-chopper [5]

L'OSHT permet de spécifier la durée pendant laquelle l'impulsion haute sera active. Une fois que la durée de l'OSHT est écoulée, les interrupteurs du hacheur sont désactivés pour une certaine période avant de passer à la prochaine impulsion.

on peut réduire les pertes de commutation, telles que les pertes de conduction et les pertes de commutation, dans le hacheur. Cela permet d'améliorer l'efficacité énergétique globale du système.

### IV.1.8 trip zone

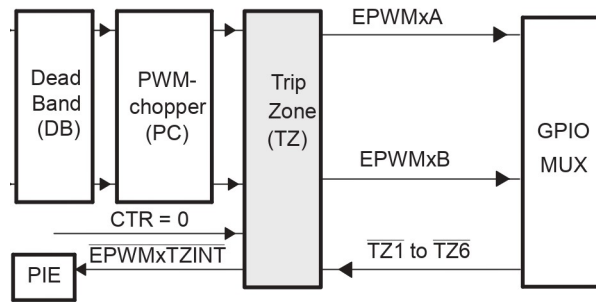


FIGURE IV.16 – le module trip zone[5]

Ces blocs fonctionnels sont responsables de la détection des événements qui peuvent générer une interruption ou une action sur les sorties PWM. Ils sont configurés à l'aide des registres TZSEL, TZCTL et TZEINT.

Configurez le module ePWM pour qu'il réagisse à une, à toutes ou à aucune des broches de zone de déclenchement.

Spécifiez l'action de déclenchement prise lorsqu'un défaut se produit :

- Forcer EPWMxA et/ou EPWMxB à l'état haut
- Forcer EPWMxA et/ou EPWMxB à l'état bas
- Forcer EPWMxA et/ou EPWMxB à un état de haute impédance
- Configurer l'EPWMxA et/ou l'EPWMxB pour qu'ils ignorent toute condition de déclenchement.

Configurer la fréquence à laquelle l'ePWM réagira à chaque broche de la zone de déclenchement :

- Une seule fois
- Cycle par cycle

Activer la zone de déclenchement pour déclencher une interruption.

Contourner entièrement le module de zone de déclenchement.

#### IV.1.8.1 TZ1 to TZ6 (Trip Zones 1 to 6)

Les zones de déclenchement du module EPWM du TMS320F28335 sont utilisées pour détecter les défauts ou les conditions anormales dans le système et déclencher des routines de protection contre les défauts ou des séquences d'arrêt. Chacune des six zones de déclenchement (TZ1 à TZ6) peut être configurée pour surveiller divers paramètres tels que la tension, le courant, la température, etc. Le registre de contrôle de chaque zone de déclenchement permet de configurer la source, l'état et l'action à prendre lorsqu'un événement de déclenchement se produit, avec des options comprenant PWMxA, PWMxB, CMPA, CMPB, DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2, ou des combinaisons de ces sources.

L'état de la zone de déclenchement peut être haut ou bas. L'action à entreprendre lorsqu'un événement de déclenchement se produit peut être l'une des suivantes :

- Ne rien faire
- Mettre la broche de sortie correspondante dans un état spécifié (bas ou haut)
- Déclencher une impulsion unique sur la broche de sortie correspondante.
- Générer une interruption vers l'unité centrale (TZINT)
- Déclencher une interruption logicielle vers l'unité centrale (TZSEL)

#### IV.1.8.2 EPWMxTZINT (ePWMx Trip Zone Interrupt)

EPWMxTZINT est un signal généré lorsqu'un événement de zone de déclenchement se produit sur le module ePWMx. L'interruption de la zone de déclenchement peut être utilisée pour générer une interruption vers le CPU, qui peut être utilisée pour déclencher une routine de protection contre les défauts ou pour initier une séquence d'arrêt.

#### IV.1.8.3 EPWMxBO (PWMxB Output)

EPWMxBO est un signal qui représente la sortie de la broche PWMxB du module ePWM. Ce signal est utilisé pour générer un signal PWM complémentaire déphasé de 180 degrés par rapport au signal PWMxA. La sortie PWMxB peut être utilisée pour piloter le côté bas d'un inverseur en demi-pont ou en pont complet.

#### IV.1.8.4 Event Trigger

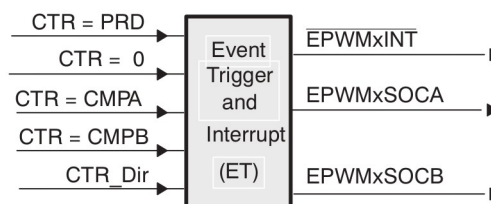


FIGURE IV.17 – Event Trigger[5]

Déclencheurs d'événement sont chargés de détecter les événements susceptibles de déclencher des actions sur les sorties PWM. Ils sont configurés à l'aide des registres ETSEL et ETPS. En configurant les registres ETSEL et ETPS, vous pouvez contrôler le comportement des déclencheurs d'événements et leur impact sur les sorties PWM. Les signaux de déclenchement d'événement générés peuvent être utilisés à des fins de synchronisation ou comme déclencheurs pour d'autres modules du système.

Déclenchement d'événement (ET) :

- Active les événements ePWM qui déclencheront une interruption.
- Activer les événements ePWM qui déclencheront un événement de début de conversion ADC.
- Spécifier la fréquence à laquelle les événements provoquent des déclenchements (chaque occurrence ou toutes les deux ou trois occurrences).
- Interroger, définir ou effacer les drapeaux d'événements

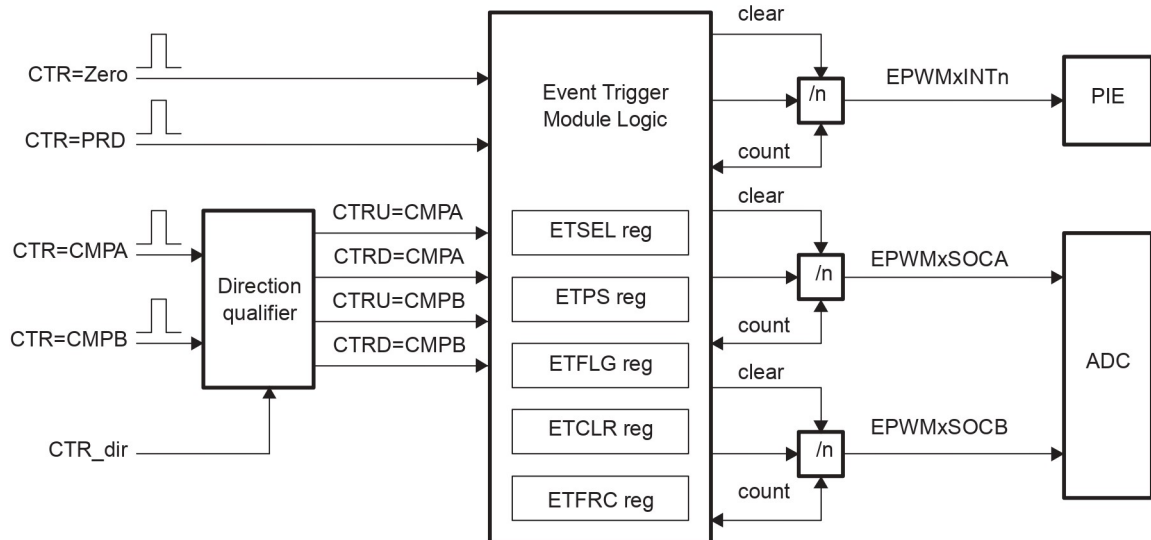


FIGURE IV.18 – Event Inputs and Prescaled Outputs [5]

#### IV.1.8.5 HRPWM (High Resolution PWM)

HRPWM (High Resolution PWM) Module : Ce bloc fonctionnel est chargé d'augmenter la résolution des signaux PWM. Il est configuré à l'aide des registres HRCNFG, HRCTL, HRMSTEP, HRMCTRL, HRMEVENT, HRMLOAD et HRMDBG.

## IV.2 Exemple de code EPWM sur ccs

```
[backgroundcolor=lightgray !20]
Tout d'abord, il faut inclure les fichiers header nécessaires :
#include "DSP2833x_Device.h"
#include "DSP2833x_Examples.h"
définir la fréquence d'horloge du module ePWM :
#define EPWM_CLK_FREQ 15000000L
on peut définir les périodes et les temps de montée et descente des signaux PWM :
#define EPWM_PERIOD 1500 // Période du signal PWM en ticks
#define EPWM_ON_TIME 750 // Temps d'allumage du signal PWM en ticks
#define EPWM_RISE_TIME 50 // Temps de montée du signal PWM en ticks
#define EPWM_FALL_TIME 50 // Temps de descente du signal PWM en ticks
// Configure les pins GPIO pour le module ePWM
GpioCtrlRegs.GPAMUX2.bit.GPIO18 = 1; // Configure GPIO18 pour ePWM1A
GpioCtrlRegs.GPAMUX2.bit.GPIO19 = 1; // Configure GPIO19 pour ePWM1B
On peut ensuite initialiser le module ePWM1 :
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Compte dans les
deux directions.
// Configure le module ePWM1 pour utiliser le signal interne de l'horloge de référence.
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // Diviseur de l'horloge haute vitesse
par 1
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1; // Diviseur de l'horloge par 1
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Désactive le déphasage
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE; // Désactive la synchronisation
```

```

EPwm1Regs.TBPRD = EPWM_PERIOD - 1; // Configure la période du signal PWM
EPwm1Regs.TBPHS.bit.TBPHS = 0x0000; // Initialise le décalage de phase
EPwm1Regs.TBCTR = 0x0000; // Initialise le compteur du module ePWM
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW; // Active le mode d'ombre
pour le registre CMPA
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW; // Active le mode d'ombre
pour le registre CMPB
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // Charge le registre CMPA
lors d'un comptage à zéro
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // Charge le registre CMPB
lors d'un comptage à zéro
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; // Configure le comportement lors d'un comp-
tage à zéro du signal PWM actif (EPWMxA)
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR; // Configure le comportement lors d'un si-
gnal de comparaison A en augmentation du signal PWM actif (EPWMxA)
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET; // Configure le comportement lors d'un comp-
tage à zéro du signal PWM complémentaire (EPWMxB)
EPwm1Regs.AQCTLB.bit.CAU = AQ_CLEAR; // Configure le comportement lors d'un
signal de comparaison B en augmentation du signal PWM complémentaire (EPWMxB)
EPwm1Regs.CMPA.bit.CMPA = EPWM_ON_TIME - 1; // Configure la valeur de com-
paraison pour le signal PWM actif
EPwm1Regs.CMPB.bit.CMPB = EPWM_ON_TIME - 1; // Configure la valeur de com-
paraison pour le signal PWM complémentaire
EPwm1Regs.AQSFRC.bit.RLDCSF = 1; // Force un rechargement du registre de comparaison
lors de chaque comptage à zéro
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW; // Charge le registre de période à
partir du registre TBPRD au prochain comptage à zéro
// Active les sorties du module ePWM1
EPwm1Regs.TZCTL.bit.TZA = TZ_FORCE_HI;
EPwm1Regs.TZCTL.bit.TZB = TZ_FORCE_HI;
EPwm1Regs.TZEINT.bit.OST = 1; // Active l'interruption de déclenchement de sortie
EPwm1Regs.ETSEL.bit.INTSEL = ET_CTR_ZERO; // Sélectionne le comptage à zéro
comme source d'interruption
EPwm1Regs.ETSEL.bit.INTEN = 1; // Active l'interruption du module ePWM1
// Configure la priorité de l'interruption ePWM1
PieCtrlRegs.PIEIER3.bit.INTx1 = 1; // Active l'interruption de l'ePWM1
PieCtrlRegs.PIEACK.bit.ACK3 = 1; // Réinitialise l'indicateur d'interruption de l'ePWM1
// Active les interruptions globales EINT;
// Boucle infinie while (1) { // le code }

```

## IV.3 Module eQEP

### IV.3.1 Généralités

La périphérie d'un codeur incrémental est constituée d'une seule piste comportant des fentes qui créent un motif alternant de lignes sombres et claires, comme illustré dans la figure 1.

Le nombre de disques fait référence au nombre de paires de lignes sombres/claires qui se produisent lors d'une révolution complète du codeur (lignes par tour). Dans certains cas, une deuxième piste est ajoutée pour générer un signal qui se produit une fois par tour, appelé signal d'index (QEPI). Ce signal peut être utilisé pour indiquer une position absolue. Différents termes tels que index, marqueur, position d'origine ou référence zéro sont utilisés par les fabricants de codeurs pour identifier cette impulsion d'index.[17]

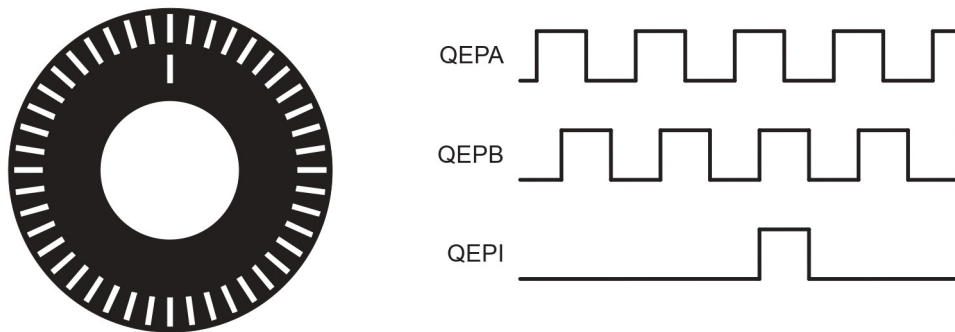


FIGURE IV.19 – Le Disque optique d'encodeur[17]

Pour récupérer les données directionnelles d'un disque, deux éléments photo distincts sont utilisés pour balayer les lignes gravées sur sa surface. Ces éléments respectent le motif du disque avec une séparation mécanique d'un quart de la hauteur des lignes. Un masque d'objet est utilisé pour limiter la vue de l'élément photo à la partie souhaitée des lignes du disque. Lorsque le disque tourne, les deux éléments photo génèrent des signaux en quadrature, QEPA et QEPB, qui sont précisément déphasés de 90 degrés l'un par rapport à l'autre. Le sens de rotation de la plupart des encodeurs est déterminé par le signal positif du canal QEPA arrivant avant le canal QEPB, et vice versa.[17]

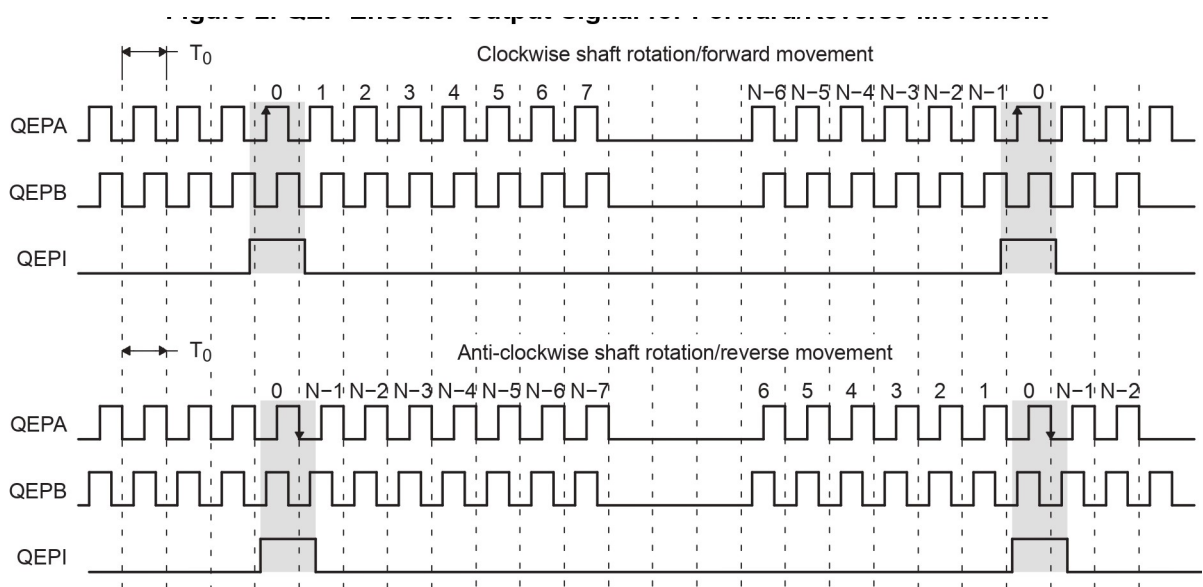


FIGURE IV.20 – Le signal de sortie du codeur QEP pour le mouvement avant/arrière[17]

Dans la plupart des roues codeuses, un tour complet est effectué à chaque tour du moteur, bien que le rapport de rotation entre la roue et le moteur puisse différer. Cela entraîne un changement proportionnel de la fréquence du signal numérique émis par les sorties QEPA et QEPB en fonction de la vitesse du moteur. Les codeurs en quadrature de différents fabricants proposent deux types d'impulsions finales : une impulsion d'index ou une impulsion d'index non saturé. Une impulsion finale standard est court-circuitée. L'impulsion d'indexation est alignée sur l'un des quatre fronts de quadrature et correspond à la largeur de l'impulsion d'entrée, pouvant représenter un quart, une moitié ou la totalité de la période du signal en quadrature.

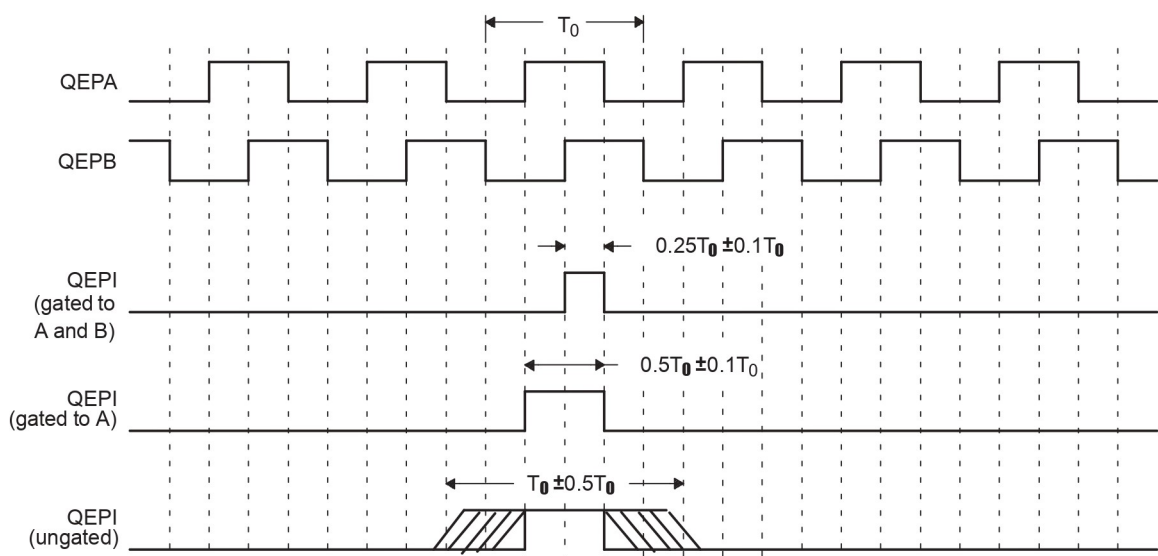


FIGURE IV.21 – Exemple d'impulsion d'index[17]

## IV.3.2 Définition d'eQEP (Enhanced Quadrature Encoder Pulse)

l'eQEP (Enhanced Quadrature Encoder Pulse) est un module périphérique spécifique qui permet de capturer et de traiter les signaux provenant d'un codeur incrémental (quadrature encoder). L'eQEP est conçu pour prendre en charge la lecture précise des mouvements et des positions angulaires mesurés par un codeur incrémental. Il offre des fonctionnalités avancées pour capturer les impulsions du codeur, calculer la vitesse et la position, et détecter les événements de position spécifiques. Les principales fonctionnalités de l'eQEP d'un DSP peuvent inclure :

- Capture des impulsions : L'eQEP peut détecter les impulsions provenant des signaux en quadrature générés par le codeur incrémental. Il peut compter le nombre d'impulsions capturées pour déterminer la position relative ou absolue.
- Détection d'événements : L'eQEP peut détecter des événements spécifiques tels que la position unitaire (UP) ou l'index. Ces événements peuvent être utilisés pour effectuer des actions spéciales ou déclencher des interruptions dans le programme.
- Calcul de la vitesse : L'eQEP peut mesurer la vitesse angulaire du codeur en utilisant les variations de position sur une période donnée. Il peut fournir des informations précises sur la vitesse de rotation du codeur.
- Modes de fonctionnement : L'eQEP peut offrir différents modes de fonctionnement, tels que le mode de comptage, le mode de suivi de vitesse, le mode de capture d'événements, etc. Ces modes permettent de configurer le comportement de l'eQEP en fonction des besoins spécifiques de l'application.

## IV.3.3 Description

### IV.3.3.1 Les Entrées de eQEP

Les entrées eQEP comportent deux broches pour le mode d'horloge en quadrature ou le mode de comptage de direction, un index (ou marqueur 0) et une entrée stroboscopique.[17]

- **QEPA/XCLK et QEPB/XDIR :**  
La broche QEPA/XCLK peut être utilisée en mode horloge en quadrature et QEPB/XDIR en mode comptage de direction.
- **Mode horloge en quadrature :**

Les codeurs eQEP délivrent deux signaux rectangulaires (A et B) qui sont électriquement déphasés de 90 degrés et dont la relation de phase est utilisée pour déterminer le sens de rotation de l'arbre d'entrée et le nombre d'impulsions eQEP à partir de la position d'index pour obtenir l'information sur la position relative.



Dans le cas d'une rotation vers l'avant ou dans le sens des aiguilles d'une montre, le signal QEPA précède le signal QEPB et vice versa. Le décodeur en quadrature utilise ces deux entrées pour générer des signaux d'horloge et de direction en quadrature.

- **Mode comptage de direction :**

Les signaux de direction et d'horloge sont apportés directement par la source externe.

La broche QEPA délivre l'entrée d'horloge et la broche QEPB délivre l'entrée de direction.

- **eQEPI-Index ou marqueur de zéro :**

Le codeur eQEP utilise un signal d'index pour établir une position de départ absolue. Ce signal est connecté à une broche de sortie du codeur et peut être utilisé pour réinitialiser le compteur de position à chaque tour si nécessaire. La présence du signal d'index permet d'initialiser ou de verrouiller le compteur de position du codeur lorsque l'événement souhaité se produit sur la broche d'index. Cela permet d'obtenir une référence de position absolue à partir de laquelle les mesures de position ultérieures peuvent être effectuées de manière incrémentielle.

- **QEPS-Entrée stroboscopique :**

L'entrée stroboscopique QEPS est utilisée pour effectuer une action précise ou une capture de position à un moment spécifique du signal du codeur incrémental. Elle est utilisée en conjonction avec les signaux d'impulsions quadrature (A et B) pour déterminer la position exacte du codeur.

Ce signal est généralement connecté à un capteur ou à un interrupteur de fin de course pour signaler que le moteur a atteint une position définie.

#### IV.3.3.2 Description fonctionnelle

Le périphérique eQEP contient les principales unités fonctionnelles suivantes :

- Qualification d'entrée programmable pour chaque broche (partie du MUX GPIO).
- Unité de décodage en quadrature (QDU).
- Compteur de position et unité de contrôle pour la mesure de la position (PCCU).
- Unité de capture des fronts en quadrature pour les mesures à basse vitesse (QCAP).
- Base de temps unitaire pour la mesure de la vitesse/fréquence (UTIME).
- Minuterie chien de garde pour la détection des décrochages (QWDOG).



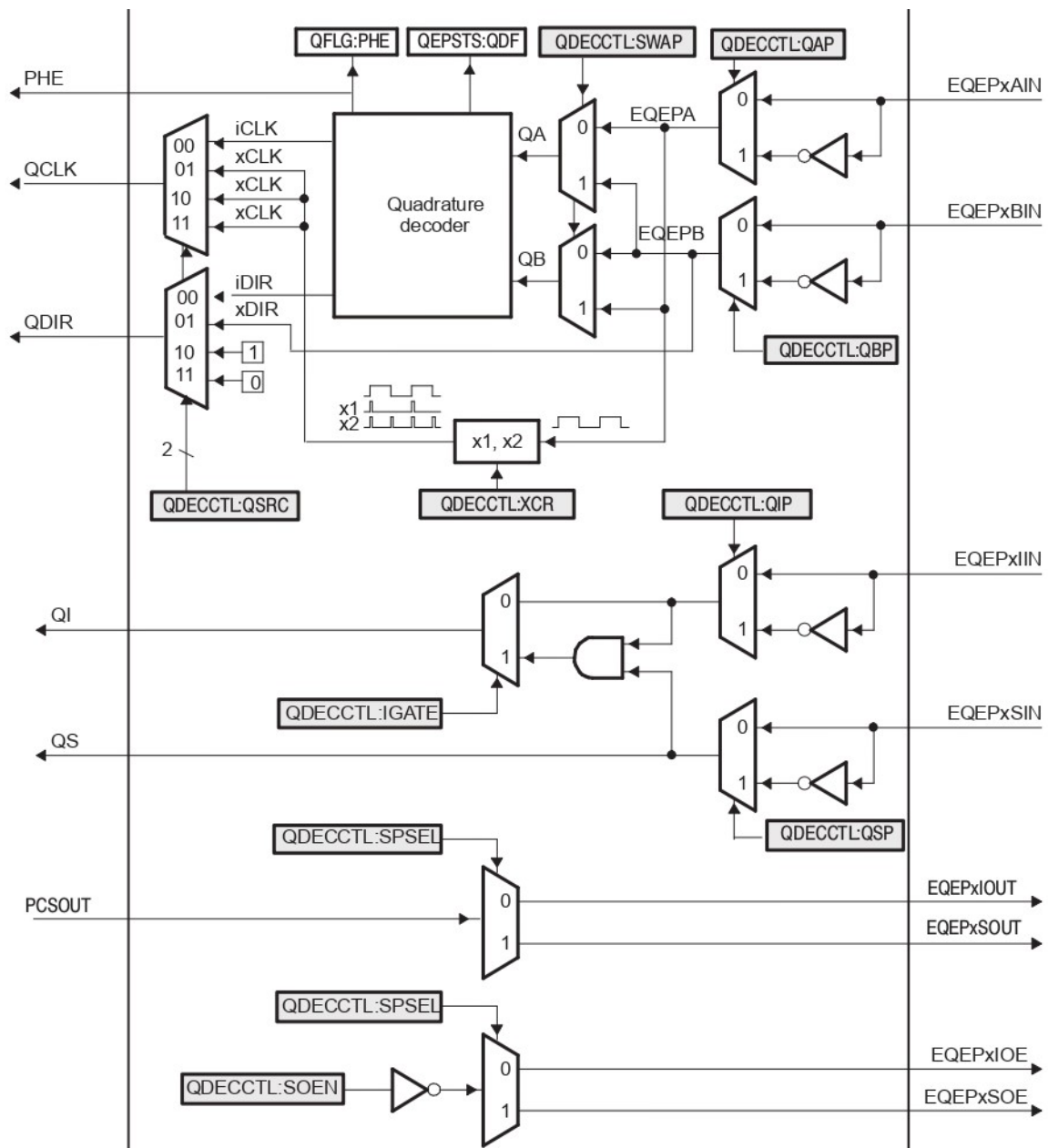


FIGURE IV.23 – Diagramme fonctionnel de l'unité décodeur[16]

#### IV.3.4.1 Modes d'entrée du compteur de position

L'horloge et la direction de l'entrée du compteur de position sont sélectionnées à l'aide des bits **QDECCTL[QSRC]**, en fonction de l'exigence d'entrée de l'interface, comme suit :[17]

- Mode comptage en quadrature.
- Mode comptage directionnel.
- Mode comptage UP.
- Mode comptage DOWN.

##### Mode de comptage en quadrature

##### Drapeau d'erreur de phase :

Dans des conditions de fonctionnement normales, les entrées en quadrature **QEPA** et

QEPB sont déphasées de 90 degrés de déphasage.

Le drapeau d'erreur de phase (PHE) est généralement utilisé pour indiquer la présence d'une erreur de phase entre les signaux en quadrature, qui peut survenir en raison de diverses raisons, telles que des distorsions du signal, des interférences ou des problèmes de synchronisation. Lorsque le drapeau d'erreur de phase est activé, cela indique qu'une erreur de phase a été détectée.

**La multiplication du comptage :**

La multiplication du comptage consiste à multiplier le nombre d'impulsions capturées par un facteur d'échelle spécifié pour obtenir une valeur plus élevée. Par exemple, si le QEP capture 100 impulsions et qu'un facteur d'échelle de 4 est appliqué, le comptage sera multiplié par 4, ce qui donne une valeur de position plus élevée (400).

**Le comptage inversé :**

Le comptage inversé dans un QEP peut être utilisé pour mesurer des mouvements dans la direction opposée à celle définie par les impulsions du codeur incrémental.

Le comptage inversé est activé en réglant le bit SWAP dans le registre QDECCTL.

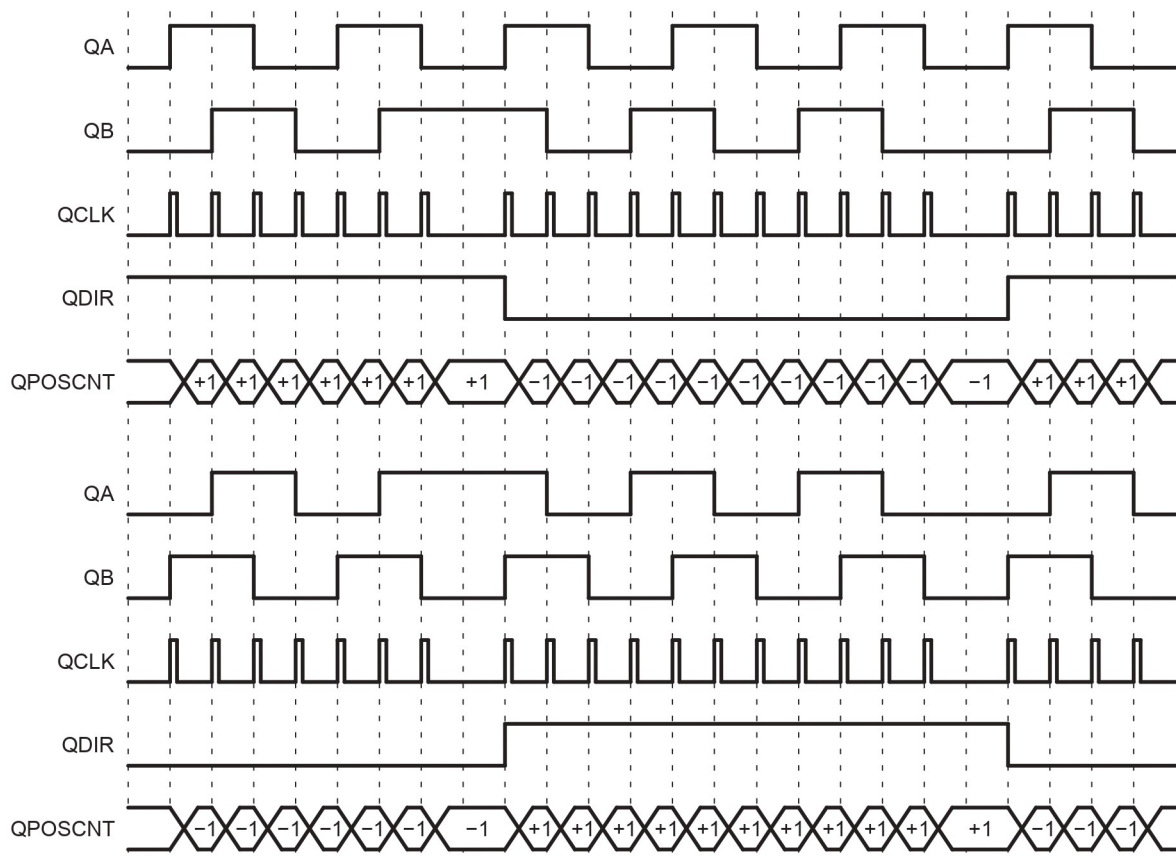


FIGURE IV.24 – Décodage de l'horloge en quadrature et de la direction[17]

**Mode de comptage de direction :**

L'entrée QEPA fournira l'horloge pour le compteur de position et l'entrée QEPB fournira

l'information de direction. Le compteur de position est incrémenté à chaque front montant d'une entrée QEPA lorsque l'entrée de direction est activée, et décrémente lorsque l'entrée direction est basse.

#### **Mode de comptage ascendant/descendant :**

mode de comptage ascendant fait référence à un mode de fonctionnement où le compteur du QEP est incrémenté/décrémenté à chaque impulsion du codeur incrémental.

Lorsque le mode de comptage ascendant est activé, chaque impulsion du codeur incrémental est détectée et utilisée pour augmenter/diminuer la valeur du compteur du QEP. Cela permet de mesurer et de suivre le mouvement dans une direction spécifique, généralement définie par les impulsions du codeur incrémental.

Le réglage du bit QDECCTL[XCR] permet de générer une horloge pour le compteur de position sur les deux fronts de l'entrée QEPA, ce qui augmente la résolution de la mesure par un facteur de 2.

#### **IV.3.4.2 Sortie Synchronisation Position-Compare**

La sortie de synchronisation Position-Compare est une sortie spécifique qui fournit une synchronisation ou une indication d'événement lorsqu'une condition de comparaison de position spécifique est atteinte.

Le QEP peut être configuré pour comparer la position actuelle du codeur incrémental avec une valeur de position cible préalablement définie. Lorsque la position actuelle du codeur atteint ou dépasse cette valeur de position cible, la sortie de synchronisation Position-Compare est activée.

La comparaison se fait entre le registre du compteur de positions (QPOSCNT) et le registre de comparaison de positions (QPOSCMP).

Le réglage du bit QDECCTL[SOEN] active la sortie de synchronisation de la comparaison de position et le bit QDECCTL[SPSEL] sélectionne une broche d'index eQEP ou une broche stroboscopique eQEP.

#### **IV.3.5 Compteur de position/Unité de contrôle et de comptage de la position (PCCU)**

compteur de position ou l'unité de contrôle et de comptage de la position est chargée de suivre et de maintenir la valeur de position du codeur incrémental.

Le compteur de position ou l'unité PCCU est utilisé pour compter les impulsions du codeur incrémental et fournir une mesure de la position du codeur. Il peut être configuré pour fonctionner en mode ascendant (où le compteur est incrémenté à chaque impulsion) ou en mode descendant (où le compteur est décrémente à chaque impulsion).

La valeur du compteur de position représente la position actuelle du codeur et peut être utilisée pour déterminer la distance parcourue, la position angulaire, ou d'autres mesures de position en fonction de la résolution et de l'encodage du codeur.

Le compteur de position et l'unité de contrôle fournissent deux registres de configuration (QEPCTL et QPOSCTL) pour la configuration des modes opérationnels du compteur de position, des modes d'initialisation/de verrouillage du compteur de position et de la

logique de comparaison de position pour la génération du signal de synchronisation

La résolution du compteur de position dépend de l'encodeur incrémental utilisé. Elle est généralement spécifiée en termes de nombre d'impulsions par tour de l'encodeur. Par exemple, si un encodeur a une résolution de 1000 impulsions par tour, le compteur de position augmentera ou diminuera de 1000 pour chaque tour complet de l'encodeur.

#### **IV.3.5.1 Modes de fonctionnement du compteur de position**

La façon dont les données du compteur de position sont traitées peut varier selon le système. Certains systèmes accumulent les données du compteur de position sur plusieurs tours, fournissant ainsi des informations de position par rapport à un point de référence connu. Dans d'autres systèmes, le compteur de position est remis à zéro à chaque tour grâce à une impulsion d'index, ce qui permet de mesurer l'angle du rotor par rapport à cette impulsion d'index.

Le compteur de position peut être configuré selon quatre modes :

- Remise à zéro du compteur de position lors de l'événement d'index.
- Remise à zéro du compteur de position à la position maximale.
- Remise à zéro du compteur de position lors du premier événement d'index.
- Réinitialisation du compteur de position en cas de dépassement du temps de l'unité.

En cas de débordement, le compteur de position est remis à zéro, tandis qu'en cas de dépassement, le compteur est réglé sur la valeur spécifiée dans le registre QPOS MAX. Lorsque le compteur de position dépasse cette valeur, un dépassement se produit. De plus, lorsque le compteur de position atteint "0" puis décompte, cela entraîne un débordement, qui est signalé par le drapeau d'interruption dans le registre QFLG.

#### **IV.3.5.2 Unité de comparaison de positions eQEP**

Le périphérique eQEP comprend une unité de comparaison de position pour générer une sortie de synchronisation et/ou une interruption en cas de comparaison de position. Le registre de comparaison de position (QPOSCMP) est occulté et le mode occultant peut être activé ou désactivé à l'aide du bit QPOSCTL[PSSHDW]. Si le mode fantôme n'est pas activé, la CPU écrit directement dans le registre de comparaison de position actif.

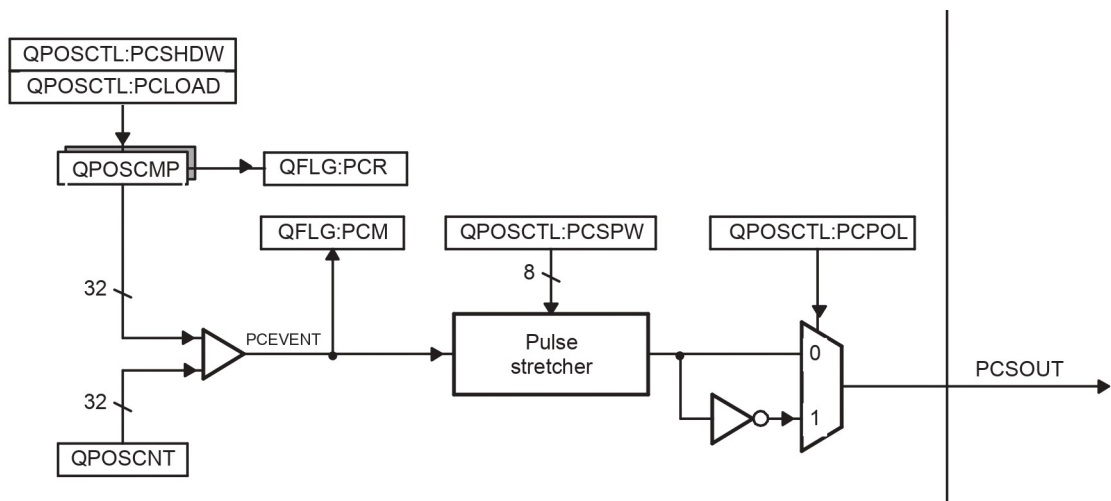


FIGURE IV.25 – Unité de comparaison de positions eQEP[17]

En mode fantôme, l'unité de comparaison de position (QPOSCTL[PCLOAD]) peut être configurée pour charger la valeur du registre fantôme dans le registre actif à un moment ultérieur et générer une interruption de comparaison de position après le chargement. Position prête (QFLG[PCR]).

- Chargement si la comparaison correspond
- Chargement en cas d'événement de compteur de position zéro.

La comparaison de position (QFLG[PCM]) est activée lorsque la valeur du compteur de position (QPOSCNT) correspond au registre de comparaison de position actif (QPOSCMP), et la sortie de temporisation de comparaison de position de la largeur d'impulsion programmable est générée en cas de comparaison pour déclencher un dispositif externe.

Par exemple, si  $QPOSCMP = 2$ , l'unité de comparaison de positions génère un événement de comparaison de positions sur 1 à 2 du compteur de position eQEP pour le sens de comptage avant et sur 3 à 2 transitions du compteur de position eQEP pour le sens de comptage arrière.

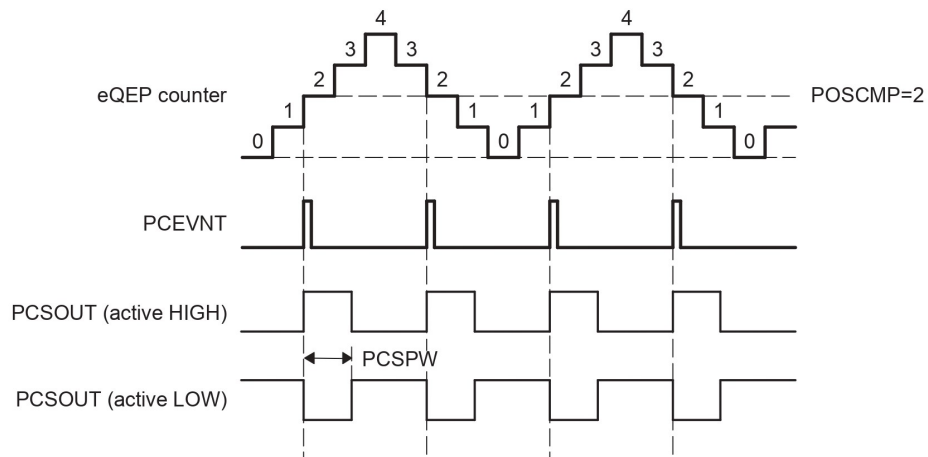


FIGURE IV.26 – eQEP Position-compare Event Generation Points[17]

La logique de l'étireur d'impulsions dans l'unité de comparaison de positions génère une sortie d'impulsion de synchronisation de comparaison de positions programmable sur la correspondance de la comparaison de positions. Dans le cas d'une nouvelle correspondance de position alors qu'une impulsion précédente est encore active, l'étireur d'impulsions génère une impulsion d'une durée spécifiée à partir du nouvel événement de comparaison de position, comme indiqué ci-dessous.

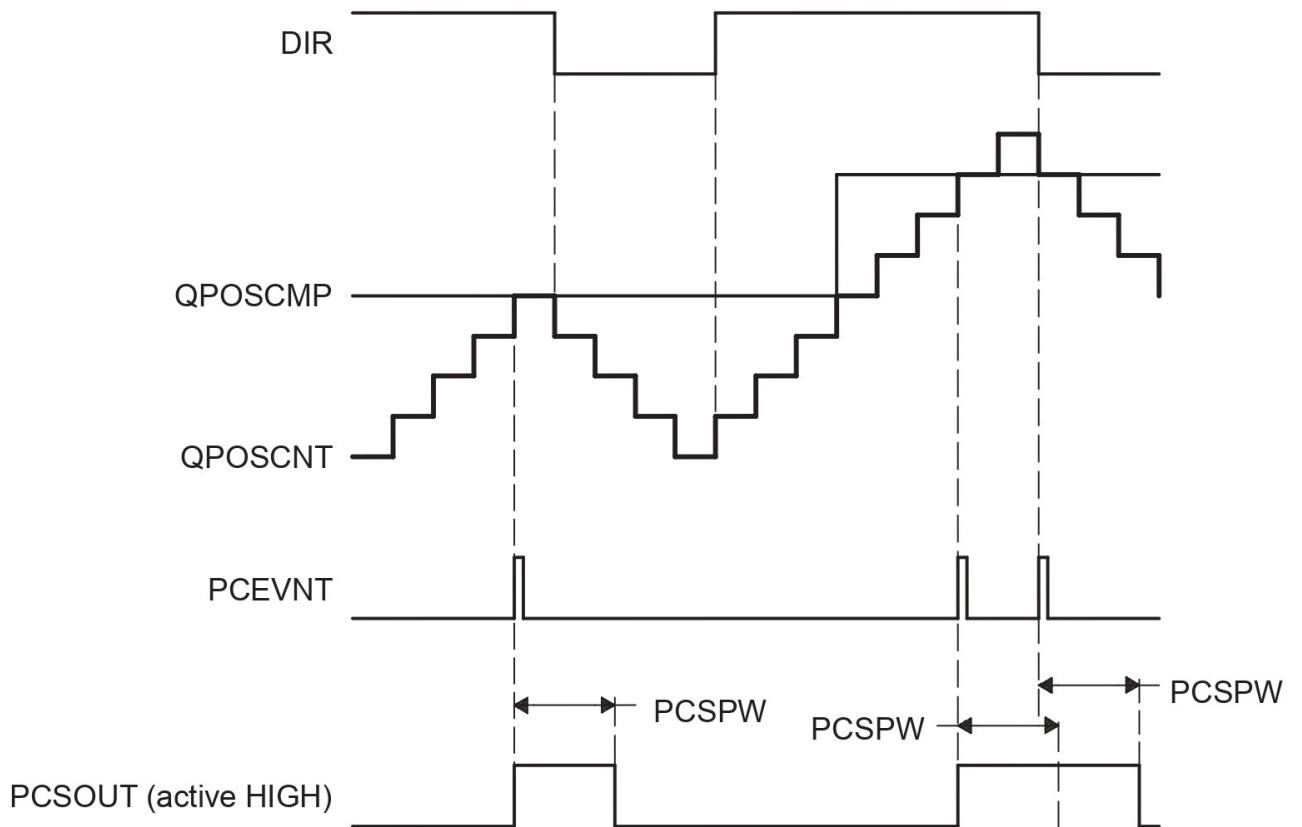


FIGURE IV.27 – eQEP Position-compare Sync Output Pulse Stretcher[17]



### IV.3.6 Unité de capture des bords eQEP

L'unité de capture des bords (Edge Capture Unit) permet de capturer et d'enregistrer les transitions ou les changements d'état des signaux d'impulsions quadrature (A et B) du codeur incrémental.

L'unité de capture des bords est capable de détecter les fronts montants et descendants des signaux d'impulsions quadrature et de capturer le moment précis où ces transitions se produisent. Cela permet d'obtenir des informations temporelles précises sur les changements de position du codeur.

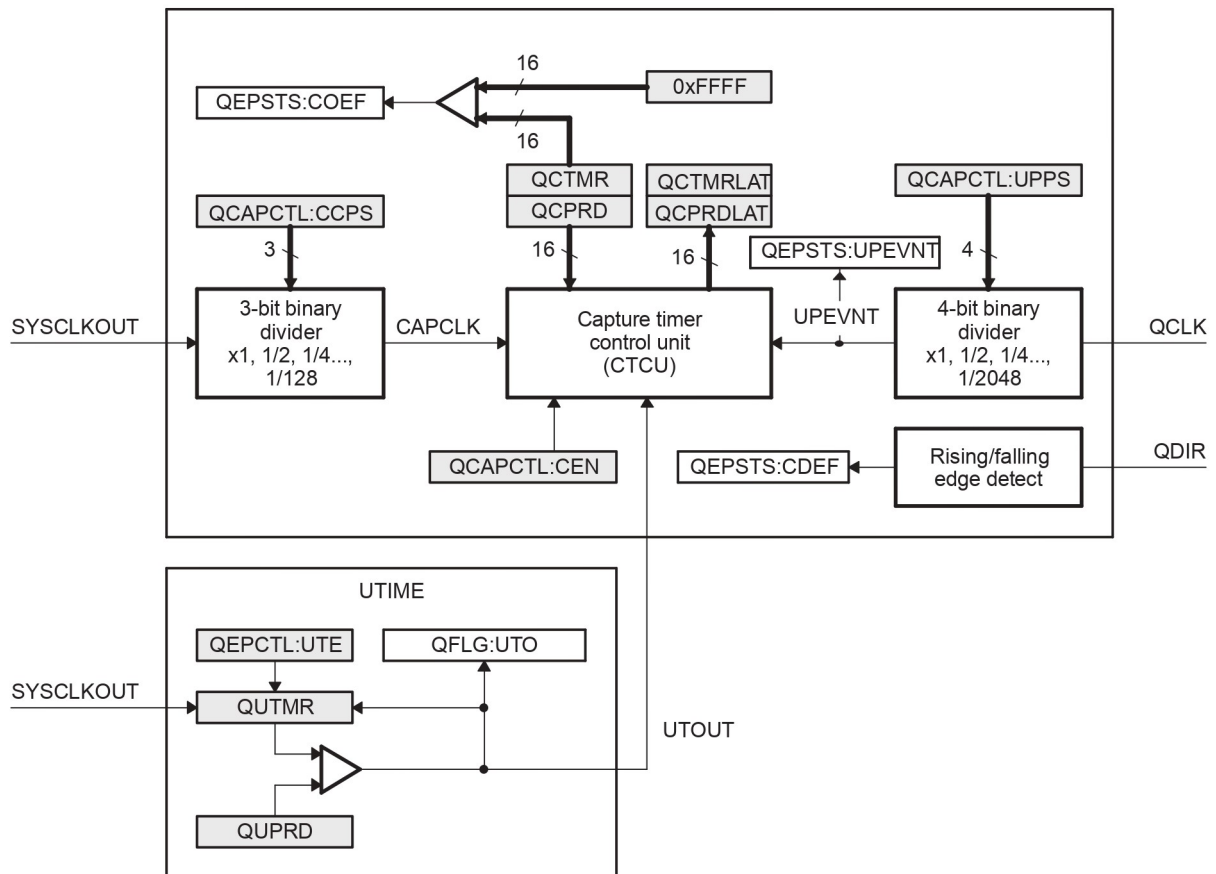


FIGURE IV.28 – Unité de capture des bords eQEP[17]

Le temporisateur de capture eQEP (QCTMR) fonctionne à partir du SYSCLKOUT présalé et le présalé est programmé par les bits QCAPCTL[CCPS].

La valeur du temporisateur de capture (QCTMR) est mise en latence dans le registre de période de capture (QCPRD) à chaque événement de position d'unité, puis le temporisateur de capture est réinitialisé, un drapeau est placé dans QEPSTS :UPEVNT pour indiquer que la nouvelle valeur est enregistrée dans le registre QCPRD.

Le logiciel peut vérifier ce drapeau d'état avant de lire le registre de période pour les mesures à faible vitesse et effacer le drapeau en écrivant 1.

L'unité de capture active le drapeau d'erreur de dépassement eQEP (QEPSTS[COEF]) en cas de dépassement du temporisateur de capture entre les événements de position de

l'unité.

Si un changement de direction se produit entre les événements de position de l'unité, alors un drapeau d'erreur est placé dans le registre d'état (QEPSTS[CDEF]).

Si le bit QEPCTL[QCLM] est désactivé, les valeurs du temporisateur de capture et de la période de capture sont verrouillées dans les registres QCTMRLAT et QCPRDLAT, respectivement, lorsque l'unité centrale lit le compteur de position (QPOSCNT).

Si le bit QEPCTL[QCLM] est activé, les valeurs du compteur de position, de la temporisation de capture et de la période de capture sont verrouillées dans les registres QPOSLAT, QCTMRLAT et QCPRDLAT, respectivement, lors de la sortie de l'unité.

Le registre QCAPCTL[UPPS] permet de définir la valeur du prédiviseur pour la capture de position unitaire dans le QEP. Le prédiviseur détermine le rapport entre la fréquence du signal d'entrée du QEP et la fréquence à laquelle les positions unitaires sont capturées. La valeur du registre QCAPCTL[UPPS] est généralement une puissance de 2, qui peut être configurée pour diviser la fréquence d'entrée du QEP. Par exemple, si le registre QCAPCTL[UPPS] est réglé sur 2, la fréquence du signal d'entrée sera divisée par 2 avant que les positions unitaires soient capturées.

En ajustant la valeur du registre QCAPCTL[UPPS], il est possible de contrôler la précision et la résolution des positions capturées par le QEP. Un prédiviseur plus élevé permet d'obtenir une meilleure résolution, mais cela peut limiter la fréquence maximale à laquelle les positions peuvent être capturées.

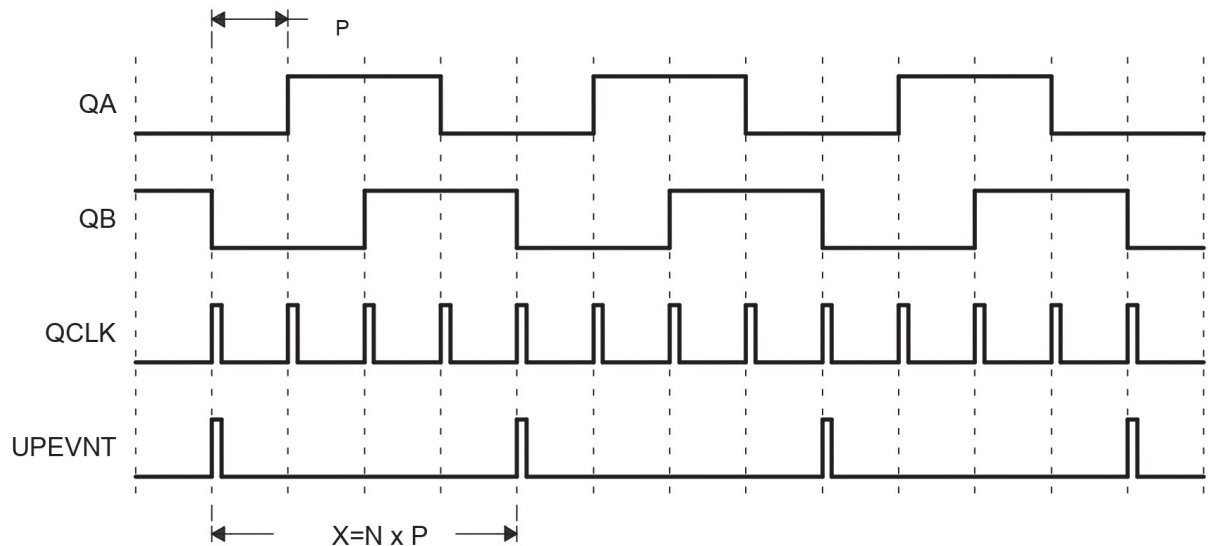


FIGURE IV.29 – Événement de position unitaire pour la mesure de la vitesse lente (QCAPCTL[UPPS] = 0010)[17]

### IV.3.7 eQEP Watchdog

Le dispositif eQEP comprend une horloge de surveillance 16 bits qui surveille l'horloge en quadrature pour indiquer le bon fonctionnement du système de contrôle de mouvement. Le temporisateur de chien de garde eQEP est cadencé à partir de SYSCLKOUT/64, et

un événement d'horloge en quadrature QCLK (impulsion) réinitialise le temporisateur de chien de garde.

Si aucun événement d'horloge en quadrature n'est détecté avant une correspondance de période ( $QWDPRD = QWDTMR$ ), le temporisateur de chien de garde est arrêté et l'indicateur d'interruption de chien de garde (QFLG [WTO]) est activé. La valeur de temporisation est programmable dans le registre de période de surveillance (QWDPRD).

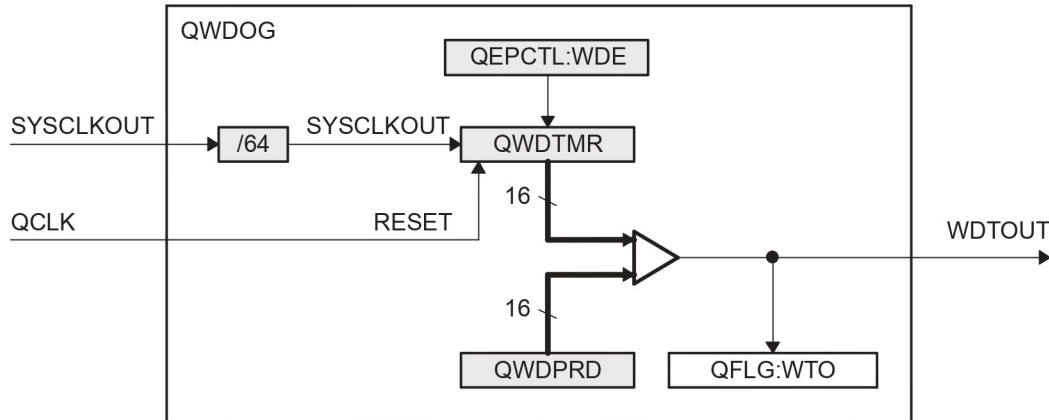


FIGURE IV.30 – Minuterie du chien de garde de l'eQEP[17]

### IV.3.8 Unité Base de la minuterie

Le périphérique eQEP comprend une minuterie de 32 bits (QUTMR) qui est cadencée par SYSCLKOUT afin de générer des interruptions périodiques pour les calculs de vitesse. L'interruption de sortie du temps de l'unité est activée (QFLG[UTO]) lorsque la minuterie de l'unité (QUTMR) correspond au registre de la période de l'unité (QUPRD).

Le périphérique eQEP peut être configuré pour verrouiller les valeurs du compteur de position, de la minuterie de capture et de la période de capture lors d'un événement de dépassement du temps de l'unité, de sorte que les valeurs verrouillées soient utilisées pour le calcul de la vitesse.

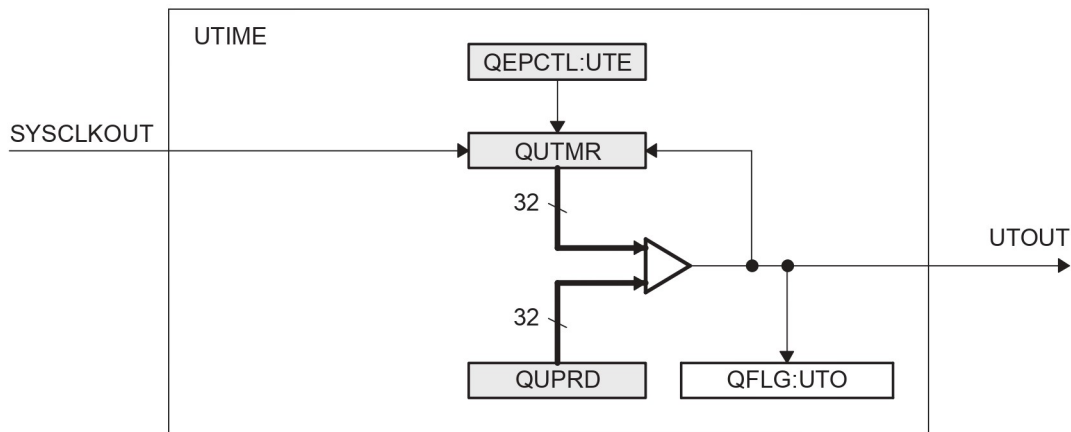


FIGURE IV.31 – Base de temps de l'unité eQEP[17]

### IV.3.9 Structure d'interruption eQEP

Le module eQEP peut générer des interruptions ou des événements en fonction de conditions spécifiques, telles que des changements de position ou des seuils de vitesse. Onze événements d'interruption (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL et UTO) peuvent être générés.

Le registre de contrôle des interruptions (QEINT) est utilisé pour activer/désactiver les sources d'interruption individuelles.

Le registre des drapeaux d'interruption (QFLG) indique si un événement d'interruption a été enregistré et contient le bit de drapeau d'interruption global (INT). Une impulsion d'interruption est générée uniquement vers le PIE si l'un des événements d'interruption est activé, le bit d'indicateur est à 1 et le bit d'indicateur INT est à 0.

La routine de service d'interruption devra effacer le bit d'indicateur d'interruption global et l'événement géré, via le registre d'annulation d'interruption (QCLR), avant que d'autres impulsions d'interruption ne soient générées. Vous pouvez forcer un événement d'interruption au moyen du registre de force d'interruption (QFRC), ce qui est utile à des fins de test.

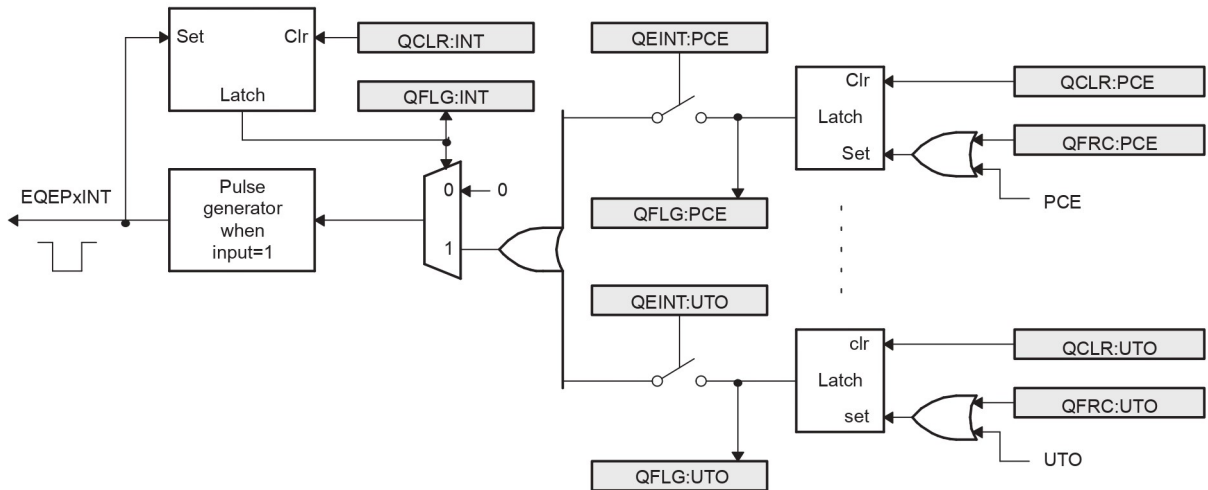


FIGURE IV.32 – Génération d'interruption EQEP[17]

## IV.4 Code CCS pour la configuration de module QEP

Ce code ci dessus est un exemple de configuration des registres de QEP pour un contrôle de moteur PMSM, sachant bien qu'il faut les configurer bit par bit . [background-color=lightgray !20] #include "DSP2833x\_Device.h" // Inclure les définitions de périphérique pour le microcontrôleur.

```

void configureQEP(void)
{ // Configuration des registres du module QEP
  EQep1Regs.QDECCTL.bit.QSRC = 0; // Utiliser le signal de quadrature pour le comptage
  EQep1Regs.QDECCTL.bit.SOEN = 1; // Activer la sortie de synchronisation
  EQep1Regs.QDECCTL.bit.SWAP = 0; // Ne pas inverser les signaux de quadrature
  EQep1Regs.QDECCTL.bit.XCR = 0; // Pas de division du signal de quadrature
  EQep1Regs.QDECCTL.bit.SPSEL = 0; // Pas de division de la période de l'horloge de syn-
  chronisation
  EQep1Regs.QEPCTL.bit.FREE_SOFT = 2; // Le module est libre de se réinitialiser à tout
  moment
  EQep1Regs.QEPCTL.bit.PCRM = 0; // Réinitialisation du compteur de position sur l'évène-
  ment d'index
  EQep1Regs.QEPCTL.bit.IEI = 1; // Activer les interruptions sur les événements d'index
  EQep1Regs.QEPCTL.bit.QPEN = 1; // Activer le module QEP
  EQep1Regs.QPOSMAX = 1000; // Définir la valeur maximale du compteur de position
  // Configuration des broches d'entrée pour le module QEP
  GpioCtrlRegs.GPAMUX2.bit.GPIO20 = 1; // Broche GPIO20 configurée comme entrée de signal
  de quadrature A
  GpioCtrlRegs.GPAMUX2.bit.GPIO21 = 1; // Broche GPIO21 configurée comme entrée de signal
  de quadrature B GpioCtrlRegs.GPAMUX2.bit.GPIO22 = 1; // Broche GPIO22 configurée comme
  entrée de signal d'index
}
void controlPMSM(void)
{ // Code de contrôle du moteur PMSM }
int main(void)
{ // Configuration du module QEP

```

```
    configureQEP() ;  
// Boucle principale  
while (1)  
{ // Contrôle du moteur PMSM  
controlPMSM() ;  
}}
```

## IV.5 ADC

## IV.6 Généralités

Le module ADC des dispositifs 2833x/2823x a été amélioré afin de fournir une interface flexible aux périphériques ePWM. Il est basé sur un module ADC rapide de 12 bits offrant un taux de conversion rapide allant jusqu'à 80 ns avec une horloge ADC de 25 MHz. Ce module ADC comporte 16 canaux, pouvant être configurés en deux modules indépendants de 8 canaux. Ces deux modules indépendants peuvent également être montés en cascade pour former un module à 16 canaux.

Il convient de noter que bien qu'il existe plusieurs canaux d'entrée et deux séquenceurs, il n'y a qu'un seul convertisseur dans le module ADC. La figure 8-8 présente le schéma fonctionnel détaillé du module ADC.

Les deux modules à 8 canaux ont la capacité de réaliser l'auto-séquençage des conversions, permettant à chaque module de choisir entre les convertisseurs respectifs. Ils peuvent sélectionner n'importe lequel des huit canaux disponibles via un MUX analogique. Lorsqu'ils fonctionnent en mode cascade, les séquenceurs automatiques agissent comme un seul séquenceur à 16 canaux. Une fois la conversion terminée sur chaque séquenceur, la valeur du canal sélectionné est stockée dans son registre RESULTAT respectif.

L'auto-séquençage offre la possibilité de convertir plusieurs fois le même canal, permettant ainsi à l'utilisateur d'effectuer des algorithmes de suréchantillonnage. Cette fonctionnalité permet d'améliorer la résolution par rapport aux résultats de la conversion traditionnelle à échantillonnage unique.

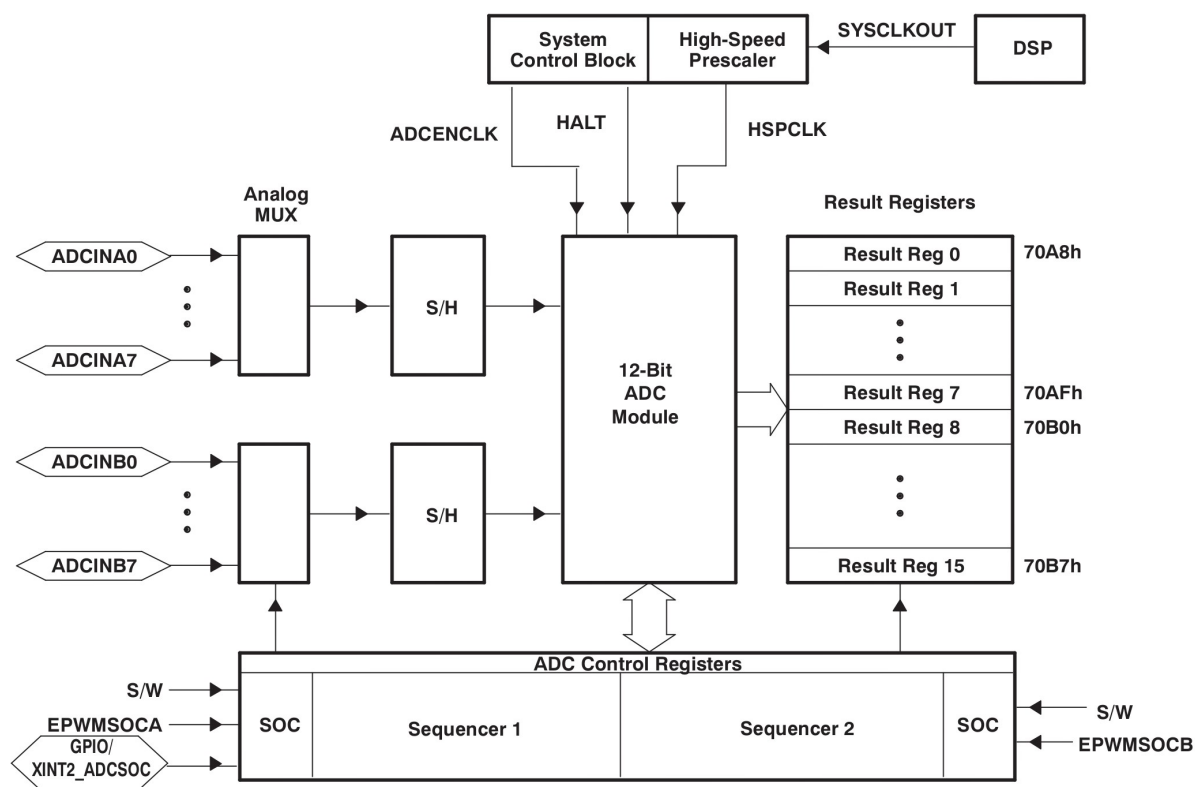


FIGURE IV.33 – Schéma fonctionnel du module ADC

L'ensemble de ces composantes interagit pour réaliser la conversion précise du signal analogique en signal numérique à l'aide de l'ADC. Les signaux de déclenchement, les horloges et les registres de contrôle permettent de configurer et de synchroniser les différentes étapes de conversion, garantissant ainsi une acquisition précise des données analogiques. Voici une explication détaillée du schéma fonctionnel de l'ADC en tenant compte des différentes composantes mentionnées :

**ADCINA0-7 et ADCINB0-7 :** Ce sont les voies d'entrée analogiques de l'ADC. Elles sont utilisées pour connecter les signaux analogiques à convertir en signaux numériques. Les ADCINA0-7 représentent les voies A et les ADCINB0-7 représentent les voies B.

**Analog Mux :** Le multiplexeur analogique (Analog Mux) permet de sélectionner la voie d'entrée appropriée à convertir. Il permet de commuter entre les différentes voies d'entrée pour que le signal analogique souhaité soit acheminé vers le convertisseur ADC.

**S/H (Sample and Hold) :** Le module S/H conserve la valeur instantanée du signal analogique pendant la durée de conversion. Il fige le signal à un instant précis afin de garantir une conversion précise.

**12-bit ADC Module :** Il s'agit du convertisseur Analogique-Numérique lui-même. Ce module effectue la conversion du signal analogique en une représentation numérique sur 12 bits. Les 12 bits permettent une résolution plus élevée et une meilleure précision de conversion.

**ADCENCLK :** Il s'agit de l'horloge d'échantillonnage du convertisseur ADC. Cette horloge définit la fréquence d'échantillonnage du signal analogique à convertir en signal numérique.

**HALT** : C'est un signal de contrôle qui permet de mettre en pause ou de reprendre le fonctionnement de l'ADC. Lorsque le signal HALT est actif, l'ADC arrête de fonctionner et conserve son état actuel.

**System Control Block** : Ce bloc est responsable de la configuration et du contrôle général du système. Il permet de configurer les différents paramètres de l'ADC, tels que la sélection des voies d'entrée, la fréquence d'horloge, etc.

**HSPCLK (High-Speed Prescaler)** : Il s'agit d'un diviseur d'horloge haute vitesse. Le HSPCLK divise la fréquence de l'horloge principale pour obtenir une horloge haute vitesse adaptée aux besoins de l'ADC.

**SysClkOut** : C'est le signal de sortie de l'horloge système. Il peut être utilisé pour synchroniser d'autres parties du système en fonction de l'horloge système.

**S/W** : Il s'agit d'un interrupteur logiciel qui permet de contrôler le démarrage ou l'arrêt du processus de conversion ADC.

**EPWMSOCA** : C'est un signal de sortie du module ePWM (modulation de largeur d'impulsion) qui peut être utilisé comme déclencheur pour lancer la conversion ADC. Lorsque le signal EPWMSOCA est actif, il déclenche le démarrage de la conversion ADC.

**XINT2\_ADCSOC** : C'est une broche d'interruption externe qui peut être configurée pour déclencher la conversion ADC. Lorsque la broche XINT2\_ADCSOC est activée, elle déclenche le démarrage de la conversion ADC.

**SOC (Start of Conversion)** : C'est le signal de démarrage de la conversion ADC. Lorsque le signal SOC est activé, la conversion ADC commence à convertir le signal analogique en signal numérique.

**ADC Control Register** : Ce registre de contrôle permet de configurer les paramètres spécifiques de l'ADC, tels que la résolution, le mode de fonctionnement, les voies d'entrée, etc.

L'ensemble de ces composantes interagit pour réaliser la conversion précise du signal analogique en signal numérique à l'aide de l'ADC. Les signaux de déclenchement, les horloges et les registres de contrôle permettent de configurer et de synchroniser les différentes étapes de conversion, garantissant ainsi une acquisition précise des données analogiques.

## IV.7 Les registres de l'ADC

les registres d'un ADC (Analog-to-Digital Converter) jouent un rôle essentiel dans la conversion des signaux analogiques provenant des capteurs en données numériques utilisables par le contrôleur.

Le fonctionnement des registres d'un ADC dans ce contexte peut être décrit en plusieurs étapes :

Avant de commencer la conversion, les registres de l'ADC doivent être configurés selon les besoins du système. Cela inclut la sélection des canaux d'entrée appropriés pour les capteurs (tels que les capteurs de courant ou de tension), le choix de la résolution (nombre de bits) de la conversion, la fréquence d'échantillonnage, etc. Ces configurations sont généralement effectuées en écrivant les valeurs appropriées dans les registres de configuration de l'ADC.



Une fois les registres de configuration correctement paramétrés, la conversion peut être démarrée. Cela peut être réalisé en activant un bit de démarrage de conversion dans le registre de contrôle de l'ADC. Le contrôleur ou le microcontrôleur envoie un signal pour déclencher la conversion.

Pendant la conversion, l'ADC lit les signaux analogiques provenant des capteurs et les convertit en valeurs numériques. Ces valeurs numériques sont stockées dans des registres spécifiques de l'ADC, généralement appelés registres de résultat. Chaque canal d'entrée a un registre de résultat associé où la valeur numérique de l'échantillon est stockée.

Une fois que tous les échantillons requis ont été acquis, l'ADC met fin à la conversion. Cela peut être détecté en surveillant un bit d'indicateur de fin de conversion dans le registre de statut de l'ADC. Lorsque ce bit est activé, cela indique que les conversions sont terminées et que les résultats sont disponibles dans les registres de résultat.

Les valeurs numériques converties des échantillons sont ensuite lues à partir des registres de résultat de l'ADC par le contrôleur du système. Ces valeurs numériques peuvent être utilisées pour calculer les grandeurs électriques et mécaniques du moteur, telles que le courant, la vitesse ou la position, nécessaires pour le contrôle FOC.

| NAME         | ADDRESS <sup>(1)</sup> | ADDRESS <sup>(2)</sup> | SIZE (x16) | DESCRIPTION                                      |
|--------------|------------------------|------------------------|------------|--|
| ADCTRL1      | 0x7100                 |                        | 1          | ADC Control Register 1                           |
| ADCTRL2      | 0x7101                 |                        | 1          | ADC Control Register 2                           |
| ADCMAXCONV   | 0x7102                 |                        | 1          | ADC Maximum Conversion Channels Register         |
| ADCCHSELSEQ1 | 0x7103                 |                        | 1          | ADC Channel Select Sequencing Control Register 1 |
| ADCCHSELSEQ2 | 0x7104                 |                        | 1          | ADC Channel Select Sequencing Control Register 2 |
| ADCCHSELSEQ3 | 0x7105                 |                        | 1          | ADC Channel Select Sequencing Control Register 3 |
| ADCCHSELSEQ4 | 0x7106                 |                        | 1          | ADC Channel Select Sequencing Control Register 4 |
| ADCASEQSR    | 0x7107                 |                        | 1          | ADC Auto-Sequence Status Register                |
| ADCRESULT0   | 0x7108                 | 0x0B00                 | 1          | ADC Conversion Result Buffer Register 0          |
| ADCRESULT1   | 0x7109                 | 0x0B01                 | 1          | ADC Conversion Result Buffer Register 1          |
| ADCRESULT2   | 0x710A                 | 0x0B02                 | 1          | ADC Conversion Result Buffer Register 2          |
| ADCRESULT3   | 0x710B                 | 0x0B03                 | 1          | ADC Conversion Result Buffer Register 3          |
| ADCRESULT4   | 0x710C                 | 0x0B04                 | 1          | ADC Conversion Result Buffer Register 4          |
| ADCRESULT5   | 0x710D                 | 0x0B05                 | 1          | ADC Conversion Result Buffer Register 5          |
| ADCRESULT6   | 0x710E                 | 0x0B06                 | 1          | ADC Conversion Result Buffer Register 6          |
| ADCRESULT7   | 0x710F                 | 0x0B07                 | 1          | ADC Conversion Result Buffer Register 7          |
| ADCRESULT8   | 0x7110                 | 0x0B08                 | 1          | ADC Conversion Result Buffer Register 8          |
| ADCRESULT9   | 0x7111                 | 0x0B09                 | 1          | ADC Conversion Result Buffer Register 9          |
| ADCRESULT10  | 0x7112                 | 0x0B0A                 | 1          | ADC Conversion Result Buffer Register 10         |
| ADCRESULT11  | 0x7113                 | 0x0B0B                 | 1          | ADC Conversion Result Buffer Register 11         |
| ADCRESULT12  | 0x7114                 | 0x0B0C                 | 1          | ADC Conversion Result Buffer Register 12         |
| ADCRESULT13  | 0x7115                 | 0x0B0D                 | 1          | ADC Conversion Result Buffer Register 13         |
| ADCRESULT14  | 0x7116                 | 0x0B0E                 | 1          | ADC Conversion Result Buffer Register 14         |
| ADCRESULT15  | 0x7117                 | 0x0B0F                 | 1          | ADC Conversion Result Buffer Register 15         |
| ADCTRL3      | 0x7118                 |                        | 1          | ADC Control Register 3                           |
| ADCST        | 0x7119                 |                        | 1          | ADC Status Register                              |
| Reserved     | 0x711A –<br>0x711B     |                        | 2          |  |
| ADCREFSEL    | 0x711C                 |                        | 1          | ADC Reference Select Register                    |
| ADCOFFTRIM   | 0x711D                 |                        | 1          | ADC Offset Trim Register                         |
| Reserved     | 0x711E –<br>0x711F     |                        | 2          |  |

FIGURE IV.34 – Les Registres du module ADC

Les ADC fonctionnent en échantillonnant le signal analogique à des intervalles réguliers et en convertissant chaque échantillon en une valeur numérique correspondante. La résolution d'un ADC, exprimée en bits, détermine le nombre de niveaux de quantification possibles et donc la précision de la conversion. Par exemple, un ADC de 12 bits peut représenter le signal analogique en 4096 niveaux distincts.

$$2^{12} = 4096V \quad \Longrightarrow \quad 0 : [000000000000]$$

.

.

.

$$3V \quad \Longrightarrow \quad 4096 : [111111111111]$$

L'ADC peut être configuré avec différentes options pour s'adapter aux besoins spécifiques de l'application. Cela inclut la sélection de la plage de tension d'entrée, la vitesse de conversion, le mode de fonctionnement (simple ou différentiel), la gestion des canaux d'entrée multiples, la compensation des erreurs, etc.

## IV.8 code ccs pour la configuration de ADC

```
[colback=blue !10 !white,colframe=lightgray]
#include "DSP2833x_Device.h"
#include "DSP2833x_Examples.h"
#define ADC_CLK_FREQ 15000000L // Fréquence d'horloge de l'ADC
#define ADC_SAMPLE_FREQ 1000 // Fréquence d'échantillonnage désirée
#define ADC_SAMPLE_PERIOD (ADC_CLK_FREQ / ADC_SAMPLE_FREQ)
void ConfigureADC()
{ // Configuration de l'horloge de l'ADC
  AdcRegs.ADCCTRL3.bit.ADCCLKPS = 0; // Diviseur de l'horloge de l'ADC = 1 (ADCCLK = SYSCLK)
  AdcRegs.ADCCTRL1.bit.SEQ_CASC = 0; // Mode de fonctionnement indépendant (non-cascade)
  // Configuration des paramètres globaux de l'ADC
  AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 0; // 1 conversion par séquence (un seul canal utilisé)
  AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // Séquence de conversion 1 : premier canal utilisé
  // Configuration des paramètres spécifiques du canal ADC
  AdcRegs.ADCSOC0CTL.bit.CHSEL = 0; // Convertit le canal ADC0
  AdcRegs.ADCSOC0CTL.bit.TRIGSEL = 1; // Déclenché par EPWM1 SOCA
  AdcRegs.ADCSOC0CTL.bit.ACQPS = 6; // Temps d'acquisition = 7 * Période de l'horloge de l'ADC (7 cycles)
  // Configuration des interruptions de l'ADC (optionnel)
  PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Active l'interruption de l'ADC
  // Active l'ADC
  AdcRegs.ADCCTRL1.bit.CONT_RUN = 1; // Mode de fonctionnement continu
  AdcRegs.ADCCTRL1.bit.SEQ_OVRD = 1; // Remplace les valeurs précédentes lors de chaque nouvelle conversion
  AdcRegs.ADCCTRL2.bit.EPWM_SOCA_SEQ1 = 1; // Active la conversion de séquence 1 avec le signal EPWM1 SOCA }
void InitADC()
{ // Configuration des registres pour les GPIO utilisés par les canaux ADC
  // Par exemple, pour utiliser GPIO0 (pin 1) comme entrée ADC0 :
  GpioCtrlRegs.AIOMUX1.bit.AIO1 = 0; // Configure GPIO0 comme entrée AIO (ADC0)
  // Configuration des paramètres de l'ADC
  AdcRegs.ADCCTRL1.bit.RESET = 1; // Réinitialise l'ADC avant la configuration
  AdcRegs.ADCCTRL1.bit.SUSMOD = 2; // Met l'ADC en mode de suspension entre les conversions
  // Configure et active l'ADC
  ConfigureADC(); }
void main(void)
{ // Initialisation du système
  InitSysCtrl(); DINT; // Désactive les interruptions globales
  InitPieCtrl(); InitPieVectTable(); EALLOW;
  // Configuration et initialisation de l'ADC
  InitADC();
```

```
// Configuration et initialisation des périphériques supplémentaires (par ex. PWM)
// Activation des interruptions globales
EINT; ERTM;
// Boucle principale
while (1)
{ // Lecture de la conversion ADC
Uint16 adcValue = AdcResult.ADCRESULT0; // Récupère la valeur de la conversion du canal ADC0
    // Utilisation de la valeur du convertisseur ADC (par ex. traitement, contrôle, etc.)
    // Attente pour la prochaine conversion DELAY_US(ADC_SAMPLE_PERIOD); } }
```

## Conclusion

En conclusion, les modules EQEP (Enhanced Quadrature Encoder Pulse), ePWM (Enhanced Pulse Width Modulation) et ADC (Analog-to-Digital Converter) jouent un rôle essentielle dans le contrôle des machines synchrones à aimants permanents (MSAP). Ces modules offrent des fonctionnalités avancées qui permettent d'atteindre un contrôle précis et efficace de ces machines.

Le module EQEP permet la mesure précise de la position, de la vitesse et de l'accélération de l'arbre moteur grâce à des encodeurs quadrature. Il fournit des informations essentielles pour le contrôle de la MSAP, notamment pour le positionnement précis de l'arbre, la détection de défauts ou le suivi de trajectoire. La configuration du module EQEP dans CCS peut être relativement complexe en raison des différentes options de configuration, mais elle permet un contrôle avancé de la machine.

Le module ePWM est responsable de la génération de signaux de commande pour les MSAP, en particulier pour le contrôle du rapport cyclique du signal PWM. Il est largement utilisé pour le contrôle de la vitesse des moteurs, en ajustant la durée du signal PWM en fonction de la commande de vitesse désirée. La configuration détaillée des registres dans CCS permet de contrôler précisément la période, la fréquence, les temps de montée et de descente, ainsi que les comportements lors des événements spécifiques.

L'ADC est un composant essentiel pour convertir les signaux analogiques provenant des capteurs en valeurs numériques exploitables par le microcontrôleur. Dans le cas du contrôle des MSAP, l'ADC est souvent utilisé pour mesurer les courants, les tensions ou d'autres grandeurs physiques nécessaires au contrôle de la machine. La configuration de l'ADC dans CCS implique la sélection des canaux, la fréquence d'échantillonnage, les temps d'acquisition, les déclencheurs de conversion, etc.

Bien que la programmation de ces modules dans CCS puisse être compliqué en raison du nombre de registres et des options de configuration, il existe des alternatives pour simplifier le développement, notamment en utilisant Simulink de MATLAB, qui offre une interface graphique conviviale qui permet de modéliser, simuler et générer du code automatiquement, facilitant ainsi la mise en œuvre des algorithmes de contrôle pour les MSAP. Cela permet de réduire considérablement le temps de développement et de faciliter la maintenance du code.

---

---

# CHAPITRE V

---

## IMPLÉMENTATIONS SOFTWARE ET HARDWARE

### Introduction

En raison de la complexité d'utilisation de Code Composer Studio (CCS) et de la longueur du code impliqué comme nous avons cité dans le chapitre précédent, on choisit d'utiliser les toolbox de DSP dans l'environnement Simulink de MATLAB pour un développement et une simulation de systèmes de contrôle meilleur d'un MSAP.

Dans Simulink, la programmation se fait à l'aide de blocs fonctionnels plutôt qu'en écrivant du code texte. Cela permet de réduire considérablement la complexité de la programmation, notamment pour les utilisateurs moins expérimentés en programmation bas niveau. Dans ce chapitre nous introduirons dans un premier temps les différentes méthodes de validation et de test on ce basant sur la partie qui nous intéresse, l'approche Hardware in the Loop.

Deuxièmement nous aborderons une généralité sur la méthode d'implémentation et des différents blocs adaptés pour notre carte tms320f28335 dans le contexte de l'approche HIL. La suite de chapitre est consacré à l'implémentation software, nous abordons le programme matlab utilisé pour la commande de moteur, ses différents blocs qui le constitue avec le fonctionnement de chacun d'eux.

### V.1 Méthodes de validation et de test

Les méthodes de validation et de test sont essentielles pour garantir la fiabilité, la sécurité et les performances des systèmes électroniques de puissance et des calculateurs numériques utilisés dans l'aéronautique. Il existe trois méthodes de validation et de test présentés ci-dessus :

### V.1.1 Software in the loop

Software-in-the-Loop (SIL) est une technique utilisée dans le développement et la validation des systèmes embarqués pour tester et valider le logiciel qui contrôle ces systèmes. Contrairement au Hardware-in-the-Loop (HIL), qui implique l'intégration de composants matériels réels, le SIL se concentre sur l'exécution du logiciel dans un environnement de simulation.

### V.1.2 Process in the loop

Le Process-in-the-Loop (PIL) est une technique utilisée dans le développement et la validation des systèmes embarqués pour tester et valider le logiciel et son intégration avec le matériel en utilisant des protocoles de communication en boucle fermée.

Le PIL est une approche intermédiaire entre le Software-in-the-Loop (SIL) et le Hardware-in-the-Loop (HIL). Il vise à évaluer le comportement du logiciel embarqué dans un environnement de simulation tout en interagissant avec les composants matériels réels.

Les avantages du PIL comprennent :

- Validation du logiciel embarqué dans des conditions réelles en intégrant les composants matériels réels.
- Détection précoce des erreurs de communication et des problèmes d'intégration entre le logiciel et le matériel.
- Réduction des risques associés au déploiement du logiciel embarqué sur le matériel réel.
- Optimisation des performances et de la synchronisation entre le logiciel et le matériel.

### V.1.3 Hardware in the loop

Hardware-in-the-Loop (HIL) est une technique de simulation et de test utilisée dans le développement et la validation des systèmes embarqués en général, tels que notre systèmes de contrôle de moteurs PMSM avec FOC. Le HIL permet de vérifier le fonctionnement du système en intégrant des composants matériels réels dans un environnement de simulation contrôlé.

Un modèle de simulation est créé pour représenter le système réel, y compris le moteur PMSM, les composants électroniques et le système de contrôle. Le modèle est développé à l'aide d'outils de simulation MATLAB/Simulink.

Par la suite Une interface HIL est utilisée pour connecter le modèle de simulation aux composants matériels réels du système, dans notre cas les composants matériels sont :la carte DSP et moteur PMSM.

Le modèle de simulation et les composants matériels sont exécutés en temps réel. Il génère des signaux de commande en fonction des entrées du système, tandis que les composants matériels réels répondent aux signaux de commande et produisent des sorties réelles.

Le HIL permet de tester le système dans des conditions réalistes et de vérifier son bon fonctionnement. Le processus HIL est généralement itératif, permettant de faire évoluer

et d'améliorer le système en fonction des résultats des tests. Les ajustements et les modifications peuvent être apportés au modèle de simulation et au système matériel pour optimiser les performances.

Le Hardware-in-the-Loop (HIL) présente plusieurs avantages significatifs on citant :

- Il permet une réduction des coûts en remplaçant les composants physiques coûteux par des modèles de simulation, ce qui évite les dépenses liées aux tests sur des équipements réels.
- Il offre une grande flexibilité en permettant des tests dans des conditions de fonctionnement variées, y compris des scénarios difficiles ou dangereux à reproduire en environnement réel.
- La flexibilité permet de couvrir un large éventail de cas d'utilisation et de valider les performances du système dans des situations réalistes.
- Il accélère le développement en réalisant des tests rapides et répétables, ce qui permet de détecter les problèmes plus tôt dans le processus de développement.

## V.2 Hardware in the loop et Matlab

Avec MATLAB et Simulink, il est possible de développer des modèles de simulation de systèmes électriques, tels que des moteurs PMSM, et de les connecter à du matériel réel via des interfaces matérielles appropriées. Cela permet de tester et de valider les algorithmes de contrôle en utilisant des données réelles, ce qui offre une approche plus réaliste pour l'analyse, la conception et l'optimisation des systèmes de contrôle.

MATLAB et HIL (Hardware-in-the-Loop) sont deux concepts et outils souvent utilisés ensemble dans le domaine du contrôle et de la simulation de systèmes embarqués.

MATLAB est utilisé pour la modélisation, la simulation, l'analyse et le développement des algorithmes de contrôle, on combinant avec le HIL pour la validation, le test et l'intégration des systèmes de contrôle en utilisant des composants matériels réels.

MATLAB offre une large gamme de bibliothèques et de fonctions mathématiques avancées, ce qui facilite l'implémentation d'algorithmes de contrôle sophistiqués.

L'utilisation de blocs Simulink dans MATLAB pour le contrôle des machines synchrones à aimants permanents (MSAP) avec DSP offre une approche pratique et efficace. Les blocs Simulink permettent de modéliser, simuler et générer automatiquement le code nécessaire pour le contrôle DSP. Cela facilite le développement de systèmes de contrôle avancés et permet d'implémenter des algorithmes sophistiqués avec une configuration intuitive. L'intégration avec le DSP permet de générer du code optimisé et de télécharger facilement le modèle sur le DSP pour une exécution en temps réel. Cette approche réduit le temps de développement, offre une validation préalable grâce à la simulation et permet d'obtenir des performances de contrôle optimales pour les MSAP.

Voici les différents blocs constitue la carte DSP sous matlab ,y compris les blocs utilisée dans notre programme matlab pour la commande de moteur :PWM,QEP,ADC,SCI,...etc,et qui sont présentée en détail dans le chapitre précédent.

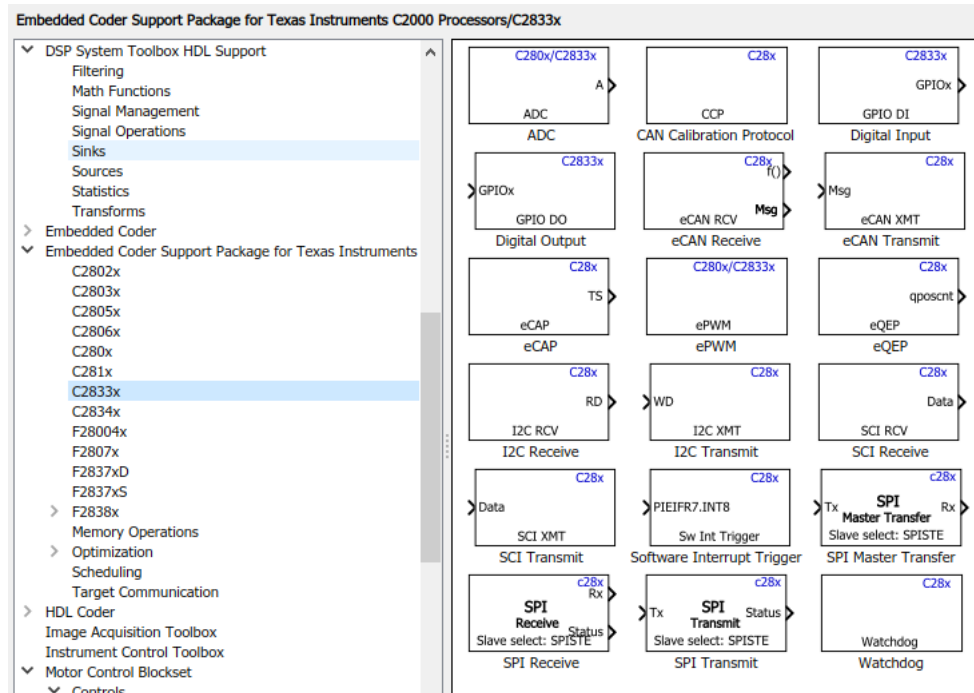


FIGURE V.1 – blocs de c28335

La figure suivante montre la librairie constituant les blocs des outils électrique, ou on trouve tous les blocs constituant notre chaîne y compris le moteur MSAP,Onduleur,les blocs de mesure de vitesse, décodeur de quadrature, mechanical to electrical position,les blocs de communication,Les contrôleurs PI...etc.

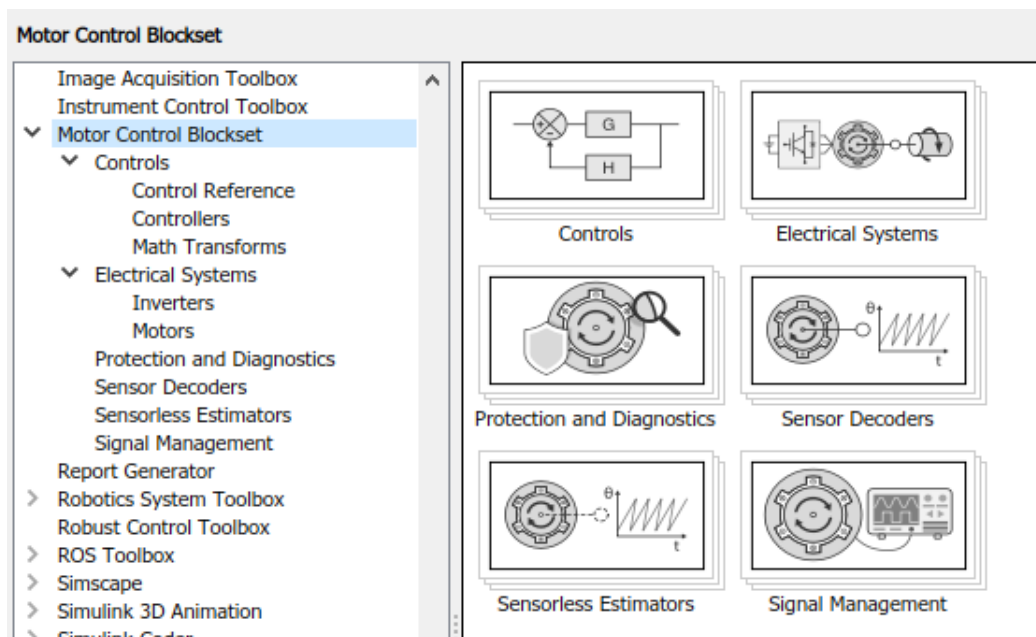


FIGURE V.2 – blocs des outils électrique



## V.3 Implémentation software

le but de notre projet est d'intégrer les composants matériels réels (moteur, onduleur, convertisseur, DSP...etc) dans un environnement de simulation contrôle(Matlab), c'est ce qu'on appelle la technique Hardware in the Loop.

Nous utiliserons les différents signaux et blocs de MATLAB Simulink pour réaliser le contrôle du moteur PMSM sur la carte DSP C2000/C28335, qui permet de générer les signaux de commande appropriés, de mesurer les grandeurs physiques du moteur, de réaliser les boucles de contrôle et d'établir des interfaces de communication pour un contrôle précis et efficace du moteur.

### V.3.1 epwm simulink

Voici le code matlab simulink Pour générer les signaux PWMs :

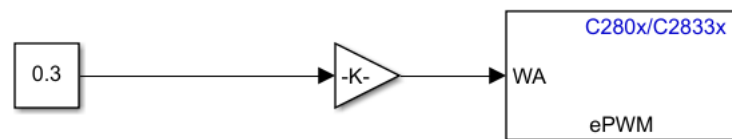


FIGURE V.3 – exemple epwm

La configuration de ce bloc se fait comme suit :

La fréquence de PWM se calcule comme suit :  $f_{pwm} = \frac{f_{clk}}{TimePRD * factor}$

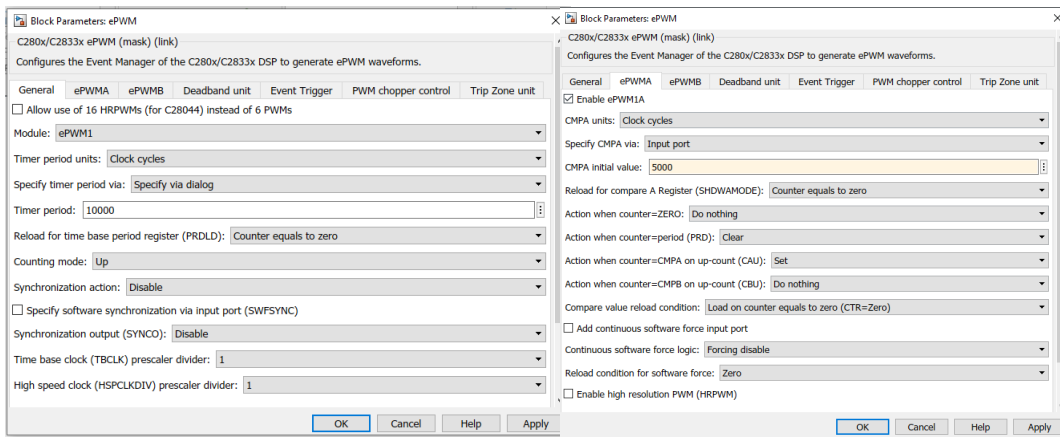
$$factor = \begin{cases} 1 & \text{si up or down mode} \\ 2 & \text{si up-down mode} \end{cases}$$

comme nous avons travaillé avec le (up) mode donc le factor=1 , le  $f_{clk} = 150MHz$  et le time period=10000

⇒

$$f_{pwm} = \frac{150.10^6}{10^4} = 15KHz$$

$f_{pwm}$  mesuré est égale presque à la valeur affiché dans l'oscilloscope (14.99 KHz).



(a) paramètre générale

(b) paramètre epwmA

FIGURE V.4 – paramètres epwm

Les signaux PWMs générés sur l'oscilloscope :

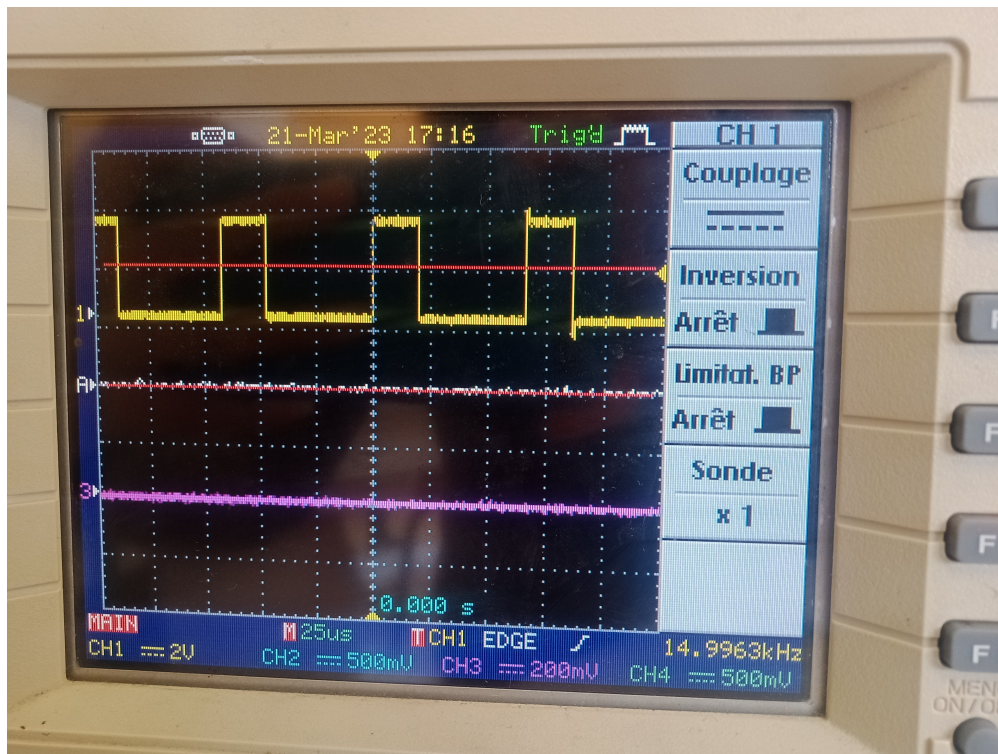


FIGURE V.5 – epwm oscilloscope

### V.3.2 Programme Matlab de Projet

La figure ci dessus représente la chaîne global de notre projet pour la commande de moteur MSAP alimentée par un onduleur avec les différents blocs de conversion et commande utilisé.

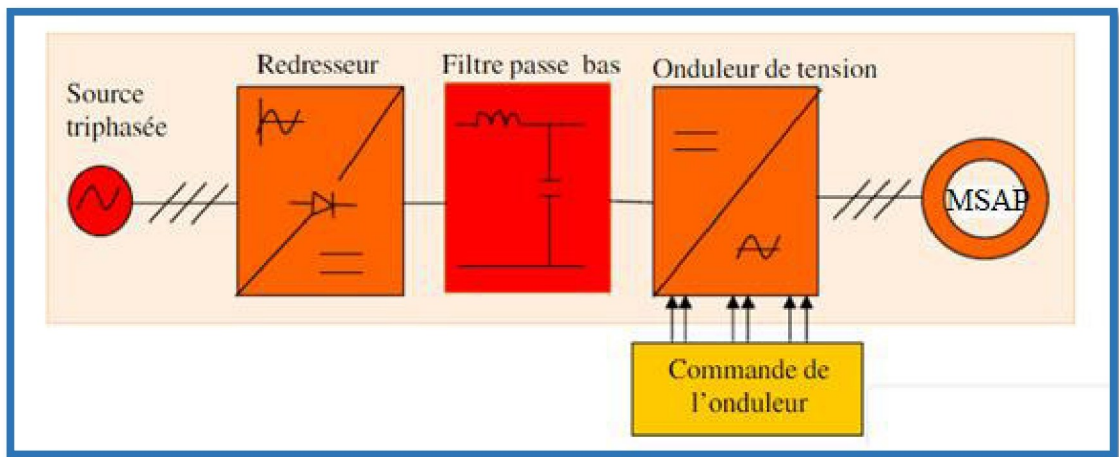


FIGURE V.6 – Schéma Global de L'association MSAP-Convertisseur

Les simulations présentées dans cette section sont réalisées sur un MSAP alimentée par un onduleur de tension à deux niveaux commandé par la commande vectorielle.

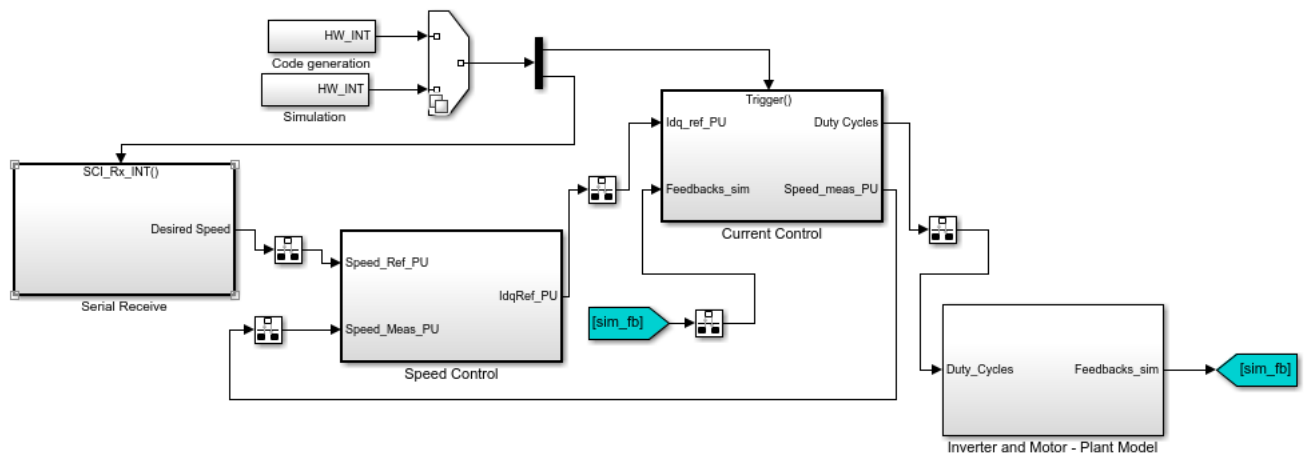


FIGURE V.7 – programme Matlab

La technique de contrôle avancée utilisée dans notre cas pour un contrôle précis du vitesse et du couple de moteur synchrone à aimant permanent(PMSM) est le contrôle orienté champ (FOC).

Le programme de commande orientée champ (FOC) d'un moteur synchrone à aimant permanent (PMSM) se compose généralement de plusieurs blocs, chacun étant responsable d'une tâche spécifique.

les deux blocs de contrôle de courant et contrôle de vitesse représente l'autopilotage de moteur ,l'un représente la régulation de courant et l'autre la régulation de vitesse comme on a détaillé dans cette partie II.3.1

Voici une brève description des principaux blocs d'un programme FOC typique :

### V.3.2.1 Bloc serial receive

Le bloc de réception SCI prend en charge la communication numérique série asynchrone entre le processeur et d'autres périphériques asynchrones. Ce bloc reçoit des données scalaires ou vectorielles en utilisant le module matériel SCI spécifié.

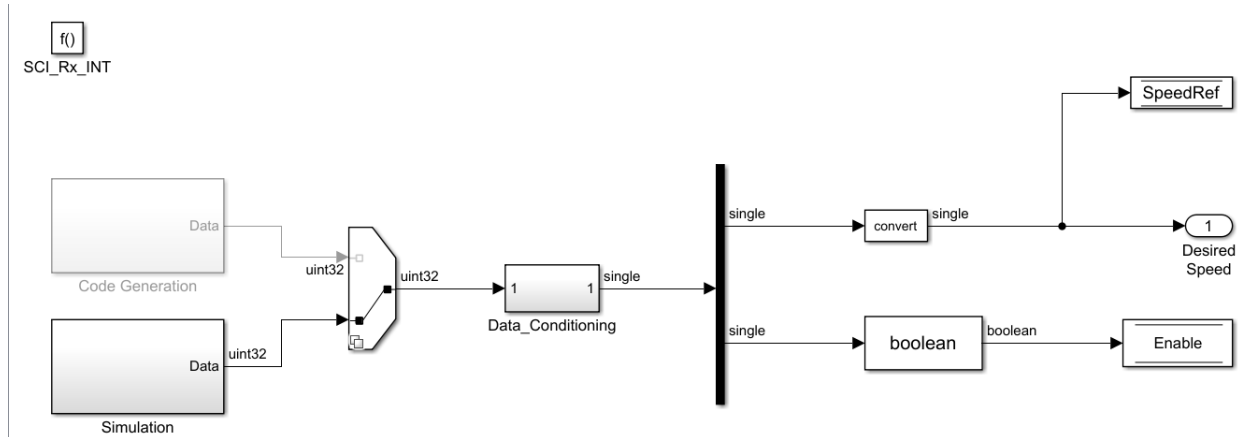


FIGURE V.8 – Serial Receive

Le bloc intitulé "Serial Receive" dans le modèle Simulink est responsable de la réception de données série à partir d'une source externe. Il est généralement utilisé pour communiquer avec un dispositif externe ou une interface qui envoie des données au modèle Simulink via une connexion série.

Il reçoit le flux de données série entrant et émet les données reçues sur son port de sortie. Il émet également l'état de l'opération de réception, indiquant si la réception des données a réussi ou si des erreurs se sont produites pendant la réception.

Les paramètres du bloc "Réception série" peuvent être configurés pour spécifier la vitesse de transmission, le format des données, la parité et d'autres paramètres de communication afin de répondre aux exigences de l'appareil ou de l'interface externe.

### V.3.2.2 Bloc de contrôle de vitesse

Le contrôle de vitesse vise à réguler et maintenir la vitesse de rotation du moteur PMSM à une valeur désirée.

Il utilise les mesures de vitesse réelle du moteur et les compare à la référence de vitesse souhaitée afin de générer une commande appropriée pour le système.

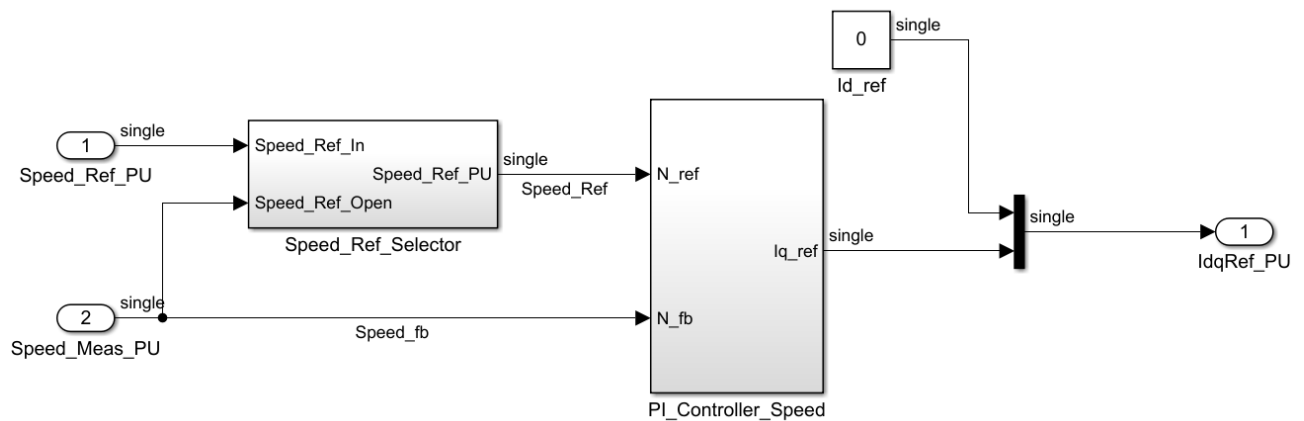


FIGURE V.9 – speed control

Il reçoit une référence de vitesse souhaitée (généralement exprimée en tours par minute, ou radian par seconde), qui peut être fournie par l'utilisateur ou être issue d'un contrôleur supérieur, tel qu'un régulateur PID (Proportional-Integral-Derivative). Elle est utilisée pour la comparer avec la vitesse mesurée et calculer l'erreur .

Après la comparaison des deux vitesses, un régulateur PI qui calcule la commande de vitesse nécessaire pour réduire l'erreur qui est utilisée pour générer une commande de courant appropriée pour le bloc de contrôle de courant suivant.

En ajustant la commande de courant en fonction de l'erreur de vitesse, le bloc de contrôle de vitesse permet d'ajuster la vitesse de rotation du moteur PMSM pour qu'elle se rapproche de la référence de vitesse souhaitée. Ce processus est itératif et se produit en boucle fermée, permettant ainsi d'atteindre et de maintenir la vitesse cible du moteur.

Ce bloc assure une régulation efficace de la vitesse, permettant au moteur de répondre aux exigences de l'application, telles que le suivi de la vitesse, la précision du positionnement ou les contraintes de couple.

### V.3.2.3 Bloc de contrôle de courant

Ce bloc est chargé de réguler le courant du stator pour obtenir le couple souhaité. Il joue un rôle essentiel vu que il est responsable de générer les signaux de commande appropriés pour le convertisseur de puissance afin de réguler le courant dans les phases du moteur.

Il effectue une transformation de coordonnées pour passer du système triphasé du moteur (généralement en coordonnées abc) à un système de référence plus adapté, qui est le système d'axes d-q, ce qui est présenté par la transformation de Park dans le deuxième chapitre. Cela permet de simplifier la régulation du courant et facilite le contrôle indépendant du courant de flux (d) et du courant de couple (q).

Ce courant de flux est perpendiculaire au flux magnétique et génère le couple nécessaire pour faire tourner le moteur. En réglant le courant de couple, le bloc de contrôle de courant peut ajuster la vitesse et le couple de sortie du moteur selon les exigences.

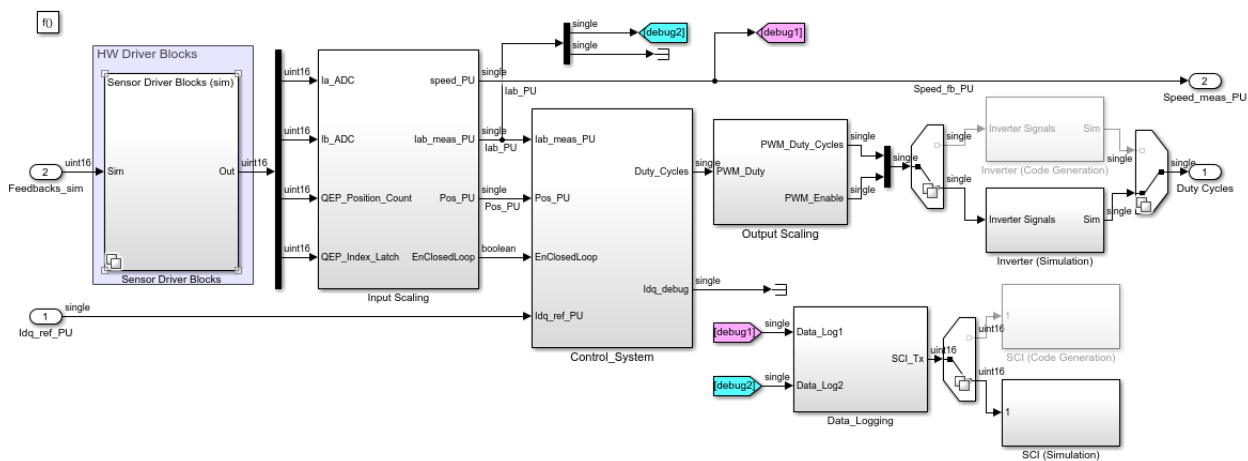


FIGURE V.10 – current control

Voici une descriptions des différentes blocs internes du bloc de contrôle de courant :

**Le bloc Sensor drivers :**

Le bloc "Sensor drivers" dans le modèle de contrôle d'un moteur PMSM est chargé de générer les signaux de commande nécessaires pour piloter le capteur utilisé afin de détecter la position du rotor. Dans ce cas, le capteur utilisé est un codeur incrémental, conçu spécifiquement pour mesurer la position précise du rotor.

Le bloc "Sensor drivers" est responsable de générer les signaux de commande essentiels pour alimenter le capteur, déclencher les mesures, synchroniser les lectures, et assurer le bon fonctionnement du capteur.

Ces signaux de commande sont importants pour contrôler le capteur de manière précise et obtenir des mesures fiables de la position du rotor du moteur PMSM. La génération précise des signaux de commande par le bloc "Pilot Sensor" garantit une détection précise de la position du rotor, ce qui est essentiel pour la mise en œuvre réussie de la commande FOC et l'optimisation des performances du moteur PMSM.

**Mise à l'échelle des entrées (Input scaling) :**

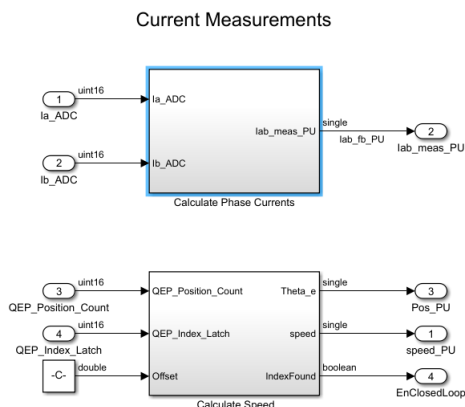


FIGURE V.11 – input scaling

Le bloc est composé de deux sous-blocs qui calculent la position, vitesse et le courant de phase de moteur.

Nous commençons par le bloc de calcul de vitesse (calculate speed) :

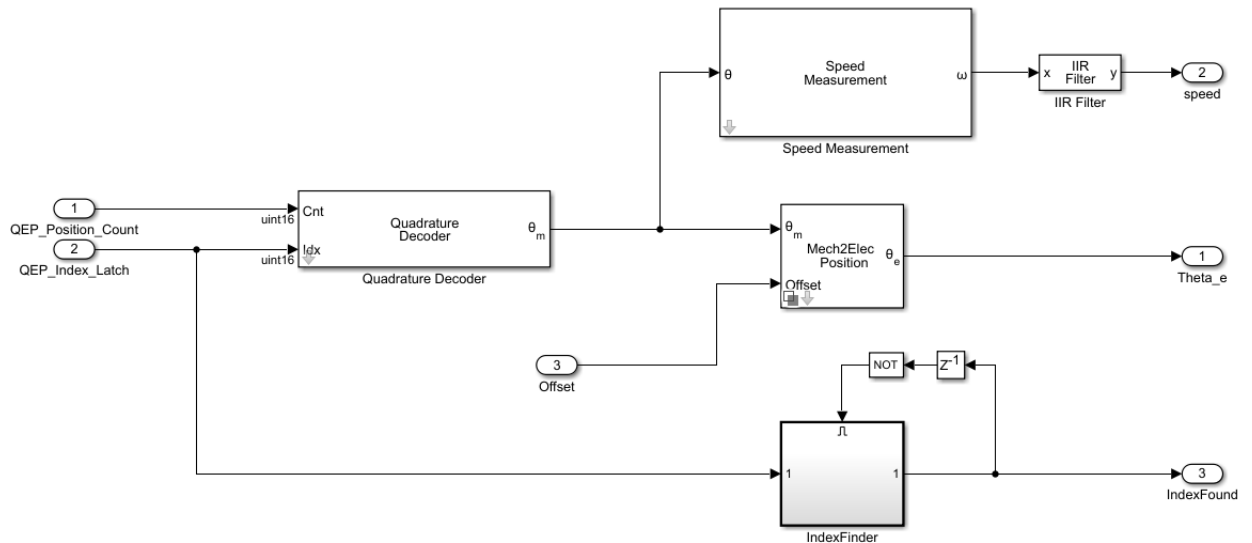


FIGURE V.12 – bloc de calculate speed

Dans ce bloc nous aurons comme sortie la vitesse ( $\omega$ ) et la position électrique ( $\theta$ ) de moteur.

Le bloc Décodeur en quadrature calcule la position du codeur en quadrature. Il utilise deux entrées de décodeur en quadrature ; position-count (Cnt) et une entrée d'index (Idx) sur 16 bits pour calculer la position angulaire du codeur en quadrature (et du rotor) en degrés, en radians ou par unité.

Les entrées de ce bloc sont :

**Cnt** : Valeur du compteur du codeur en quadrature.

**Idx** : Valeur du compteur du codeur en quadrature à la dernière impulsion de l'index.

La sortie de ce bloc est : La Position angulaire du codeur en quadrature que le bloc calcule sur la base des entrées Cnt et Idx.

**Paramètre :**

— Encoder slits : Nombre de fentes par phase 1000 (par défaut) |.

Le nombre de fentes disponibles dans chaque phase du codeur en quadrature.

— encoder counts per slit : Nombre de comptes générés pour chaque fente.

4 (par défaut) | 1 | 2 |

4 pour que le codeur génère quatre comptes correspondant aux positions ou valeurs des fentes 00, 10, 11 et 01.

— Taille du compteur : Taille du compteur du codeur en quadrature.

16 bits (par défaut) | 8 bits | 32 bits

Taille du compteur du codeur en quadrature.

- Comptage d'index externe :Activation du port d'entrée Idx. on (défaut) | off  
Le bloc s'attend à ce que la valeur du port d'entrée Cnt soit réinitialisée au moment de l'impulsion Idx.
- Unité de position :Unité de sortie de la position angulaire.  
Radians (par défaut) | Degrés | Par unité  
Unité de la sortie de position angulaire.
- Type de données de position :Type de données de sortie de position angulaire.  
Type de données pour la sortie de la position angulaire.

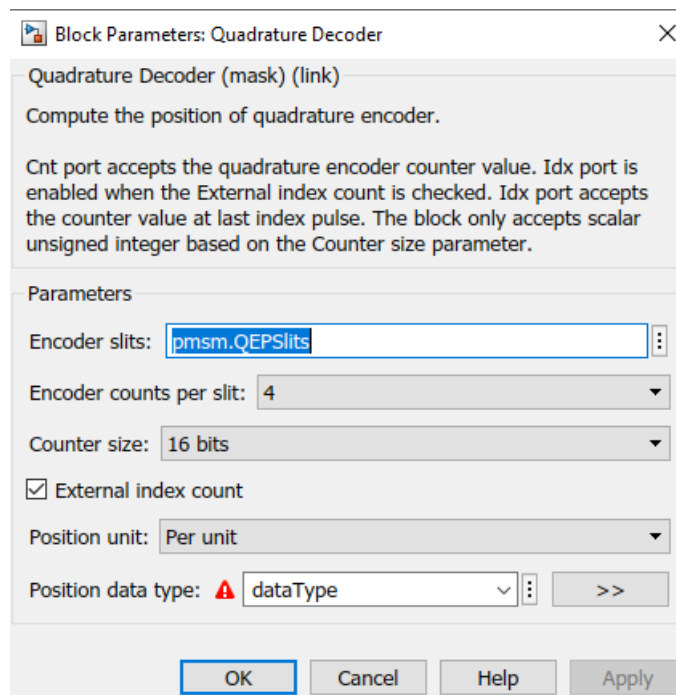


FIGURE V.13 – Paramétrage de Quadrature

### Mesure de vitesse :

Ce bloc calcule la vitesse angulaire à partir de la position angulaire du rotor en calculant sa variation en fonction du temps.

Le paramétrage de ce bloc :

- Critères de calcul de la vitesse :Méthode de calcul de la vitesse.  
Vitesse maximale de l'application (par défaut) | Résolution de la vitesse | Intervalle de temps pour le calcul de la vitesse.  
La méthode sélectionnée détermine la plage de vitesse du rotor que le bloc peut mesurer.
- Temps d'échantillonnage du bloc (S) : Temps d'échantillonnage après lequel le bloc s'exécute à nouveau.  
100e-6 (par défaut) | scalaire C'est l'intervalle de temps fixe (en secondes) entre



- deux exécutions consécutives du bloc.
- Vitesse maximale de l'application (RPM) : Vitesse maximale mesurable du rotor. 1000 (par défaut) | scalaire Vitesse maximale du rotor (en rotations par minute) que le bloc peut mesurer.
  - Résolution de la vitesse (RPM) : Valeur minimale détectable du changement de position angulaire  $5e-6$  (par défaut) | scalaire Valeur minimale de l'entrée que le bloc peut détecter.
  - Délais de calcul de la vitesse (nombre d'échantillons) : Nombre d'échantillons de position angulaire mesurés. 299 ou 28 (par défaut) | scalaire Le nombre d'échantillons de l'entrée de position angulaire que le bloc mesure pour calculer la valeur moyenne de la position.
  - Unité de vitesse : Unité de sortie de la vitesse angulaire. RPM (par défaut) | Degrés/Sec | Radians/Sec | Par unité basée sur la vitesse maximale mesurable | Par unité basée sur le dialogue. Unité de la sortie de vitesse angulaire.
  - Type de données de vitesse : Type de données de la sortie de vitesse angulaire. simple (par défaut) | double | point fixe Le type de données de la sortie de vitesse angulaire .

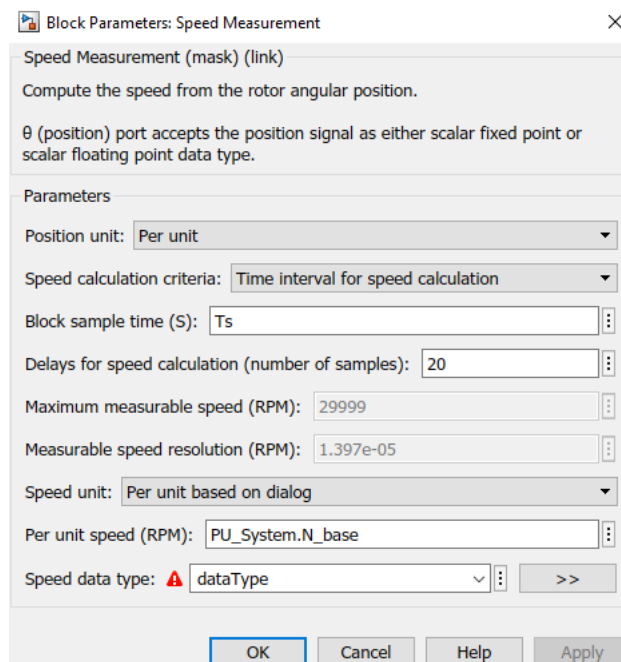


FIGURE V.14 – paramétrage de speed measurement

Nous appliquons un filtre passe-bas discret à réponse impulsionnelle infinie (IIR) sur le

signal d'entrée(vitesse angulaire de rotor),ce dernier prend en charge les types de données en virgule fixe et en virgule flottante.

Pour avoir en finale le signal de sortie filtré renvoyé par le bloc Filtre IIR dans le domaine temporel qui est la vitesse de moteur.

### **Mechanical to Electrical Position :**

Le bloc de conversion de position mécanique en position électrique calcule la position électrique du rotor en utilisant sa position et son décalage mécanique.

Nous avons la position mécanique du rotor comme entrée (telle que fournie par le capteur de position du rotor) en radians (0 à 2), en degrés (0 à 360) ou par unité (0 à 1),et le Décalage de la position mécanique(Offset) comme étant l'écart entre le zéro électrique du rotor et la position zéro mécanique.

L'unité de décalage est identique à l'unité de l'entrée de position mécanique.

Pour activer le port de décalage,on Spécifier le décalage via le port d'entrée.

Nous aurons comme sortie ( $\theta$ ) électrique :la Position électrique du rotor avec une plage identique et de même type de donnée à celle de l'entrée de la position mécanique.

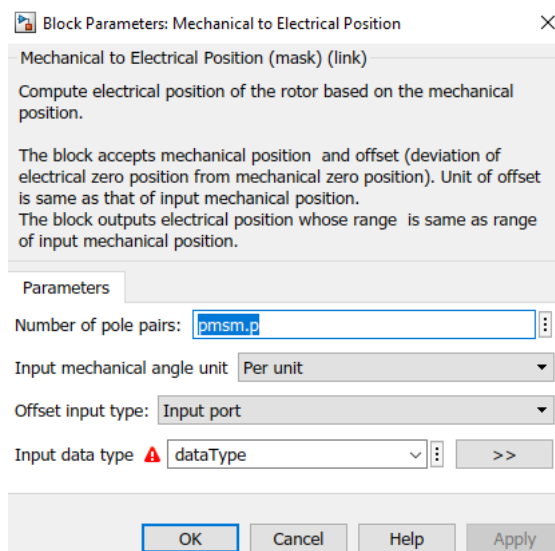


FIGURE V.15 – paramétrage de conversion mécanique à l'électrique

Nombre de paires de pôles :Nombre de paires de pôles disponibles dans le moteur.  
4 (par défaut) | scalaire.

### **Le bloc "Calculate A-Phase Currents" :**

Le bloc est responsable du calcul des courants de phase du moteur PMSM à partir des mesures de courant de l'onduleur.Il prend en entrée les mesures de courant de l'onduleur, qui sont mesurés à partir des capteurs de courant intégrés dedans.

Ces mesures de courant fournies sont les courants dans les phases A, B et C du moteur PMSM.

Le bloc utilise des techniques de démodulation pour extraire les courants de phase individuels à partir des mesures de courant.

Les mesures de courant vont être converties à l'aide de la transformations de Park pour

obtenir les courants de phase spécifiques.

Les courants de phase calculés sont ensuite disponibles en sortie du bloc et peuvent être utilisés pour la régulation du courant ou pour d'autres opérations de contrôle.

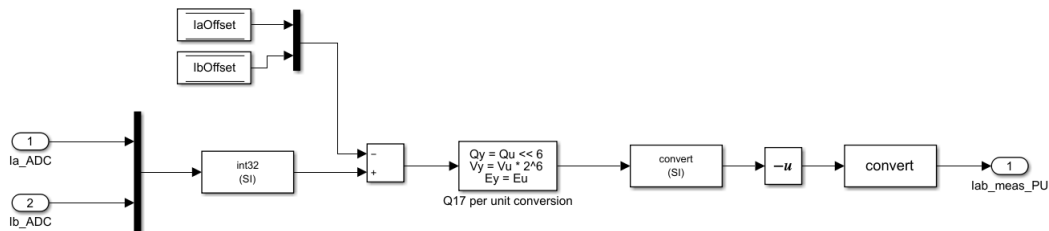


FIGURE V.16 – Bloc de mesure de courant de phase

### V.3.2.4 Le bloc de Système de contrôle

Ce bloc est responsable de la mise en œuvre du contrôle global du moteur. Il intègre et coordonne les différents sous-systèmes et blocs de contrôle pour réguler la vitesse, la position et/ou le courant du moteur PMSM.

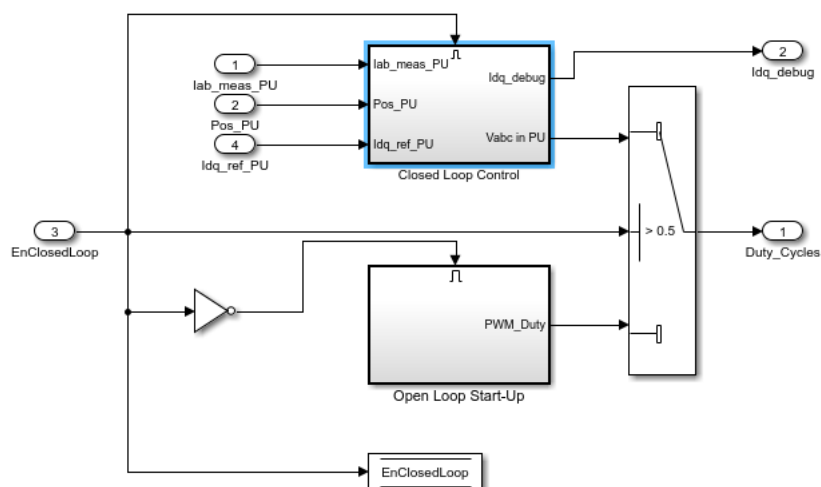


FIGURE V.17 – Bloc de système de contrôle

Le bloc prend en entrée la position du rotor et le courant du moteur pour générer le rapport cyclique, qui sera ensuite utilisé pour générer les signaux de modulation de largeur d'impulsion (PWM) pour le convertisseur de puissance.

Le bloc utilise ces informations pour calculer les commandes appropriées pour le moteur. Il effectue des calculs complexes basés sur l'algorithme FOC pour déterminer les valeurs souhaitées des courants de flux et de couple.

Une fois que les valeurs souhaitées des courants de flux et de couple sont déterminées par

le bloc de contrôle, le rapport cyclique est calculé. Ce dernier représente la durée pendant laquelle le signal PWM est maintenu à l'état haut par rapport à la période totale du signal. Il est calculé en fonction des valeurs des courants de flux et de couple.

Le rapport cyclique calculé est ensuite utilisé pour générer les signaux de modulation de largeur d'impulsion (PWM) dans le bloc de mise à l'échelle de sortie (output scaling).

Ces signaux PWM sont envoyés au convertisseur de puissance, qui alimente les bobines du stator du moteur PMSM et contrôlent la commutation des transistors du convertisseur de puissance, permettant de réguler le courant et la tension appliqués aux phases du moteur.

### V.3.2.5 Le bloc de data-logging

le bloc de data logging permet de collecter et d'enregistrer des données relatives aux courants de phase et aux performances du moteur, afin de faciliter l'analyse, l'optimisation, la vérification et la maintenance du système.

### V.3.2.6 Output Scaling

Le bloc suivant active les signaux PWM et génère les rapport cyclique de PWM.

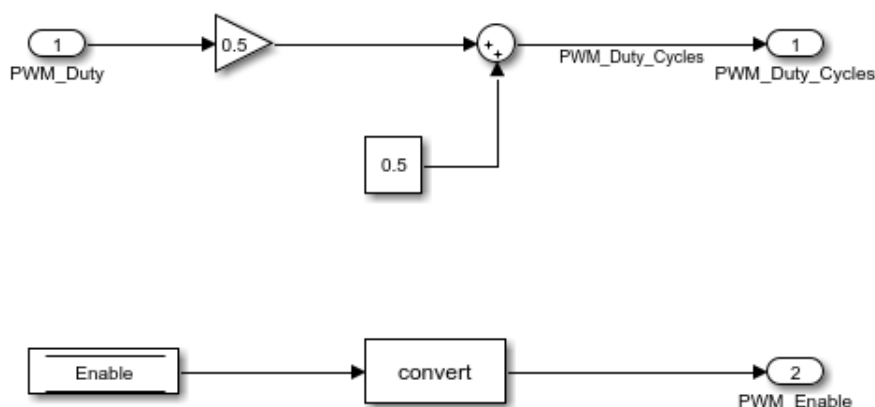


FIGURE V.18 – Bloc de Output Scaling

## V.4 Bloc de l'onduleur

Ce bloc génère les signaux PWM à partir des signaux d'onduleur et le rapport cyclique de PWM.

Il Utilise des techniques de modulation de largeur d'impulsion (PWM) pour générer les signaux de commande des interrupteurs de l'onduleur.

Bloc de l'onduleur utilise comme entrées les tensions de sortie directe (Vds) et en quadrature (Vqs) pour avoir en sortie les tensions de phase de l'onduleur.

Les paramètres de l'onduleur (Vdc, fsw) sont utilisés pour configurer les tensions de sortie et la fréquence de commutation.

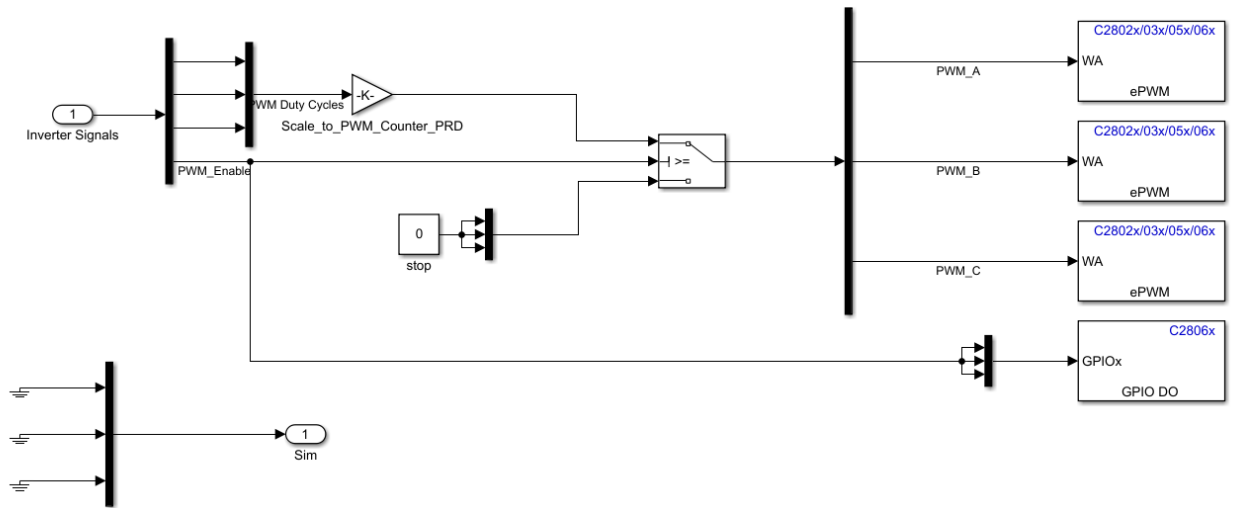


FIGURE V.19 – Bloc de l'onduleur

## V.5 Bloc de simulation d'onduleur et de moteur

Bloc du moteur PMSM contient comme entrées les tensions de sortie directe ( $V_{ds}$ ) et en quadrature ( $V_{qs}$ ) et la vitesse angulaire ( $\omega$ ) qui sont détaillé dans le deuxième chapitre. Il génère comme sortie la Vitesse angulaire ( $\omega$ ), Couple électromagnétique (torque), Courant direct ( $I_d$ ) et le Courant en quadrature ( $I_q$ ).

Il utilise les équations électriques et mécaniques du moteur PMSM pour calculer la dynamique du système en fonction des entrées et des états actuels.

Les équations comprennent le calcul des tensions de phase, le calcul des courants, le calcul de la vitesse angulaire et le calcul du couple électromagnétique.

Les paramètres du moteur PMSM ( $R$ ,  $L_d$ ,  $L_q$ ,  $J$ ,  $B$ ,  $P$ ,  $K_e$ ,  $K_t$ ) sont utilisés dans les équations pour modéliser le comportement du moteur.

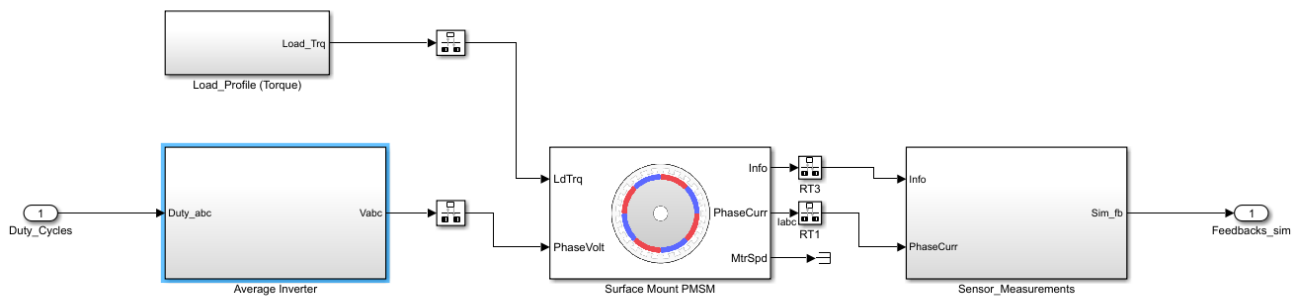


FIGURE V.20 – simulation d'un onduleur et d'un moteur

## V.6 Résultats dans data inspector

Le Data Inspector permet de visualiser, suivre et comprendre l'évolution des variables et des signaux dans le Simulink. voici les résultats de différents signaux dans le programme :

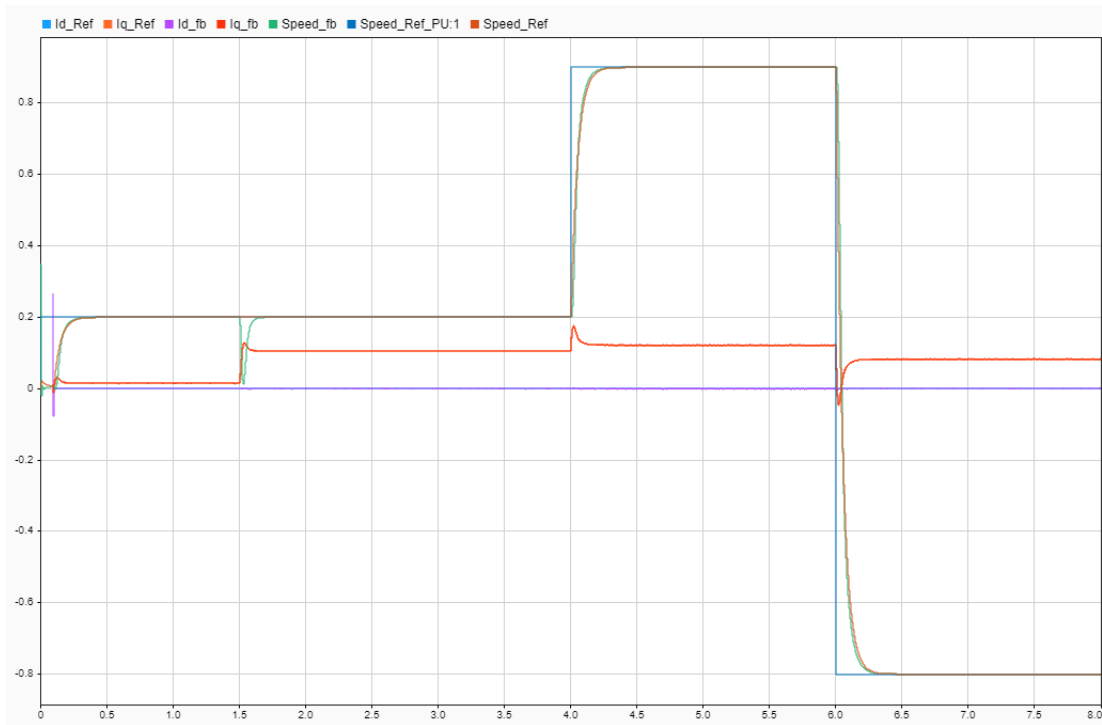


FIGURE V.21 – les courbes

$i_{q\_ref}$  (Current Reference in Quadrature Axis) : Ce signal représente la référence de courant en quadrature souhaitée pour le contrôle du moteur PMSM. La courbe devrait montrer une valeur constante ou des changements selon le profil de courant désiré.

$i_{d\_fb}$  (Feedback Current in Direct Axis) : Ce signal représente le courant de rétroaction en axe direct mesuré dans le système de coordonnées du stator. Il représente le courant réel mesuré dans l'axe direct du moteur PMSM. La courbe devrait suivre les variations du courant réel mesuré dans l'axe direct du moteur.

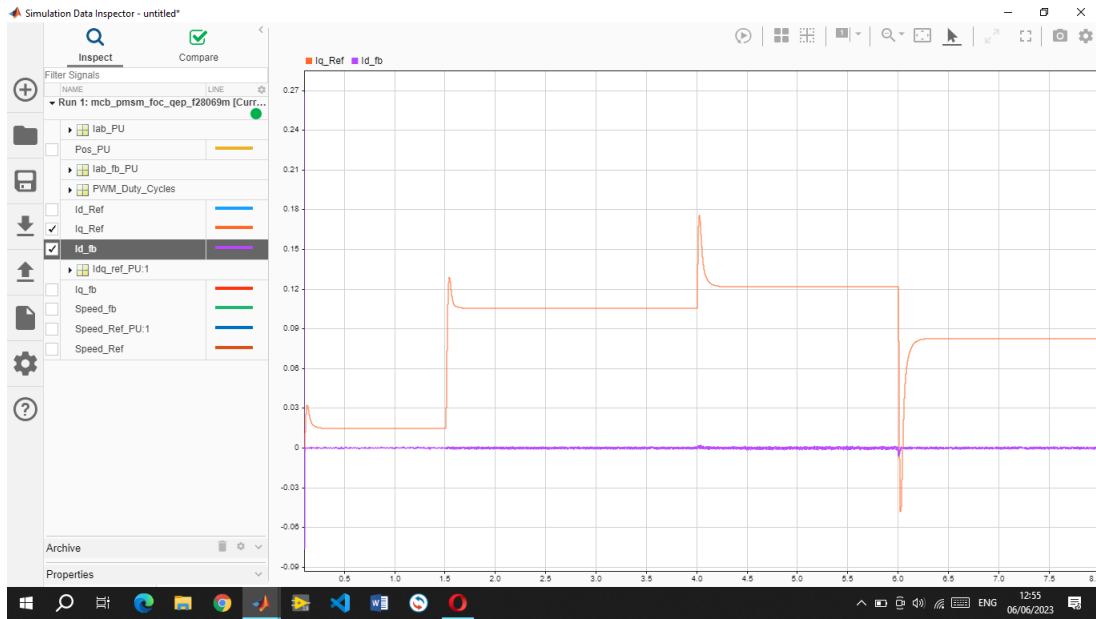


FIGURE V.22 – iq\_ref et id\_fb

iq\_fb (Feedback Current in Quadrature Axis) : Ce signal représente le courant de rétroaction en quadrature mesuré dans le système de coordonnées du rotor. Il représente le courant réel mesuré dans l'axe en quadrature du moteur PMSM.

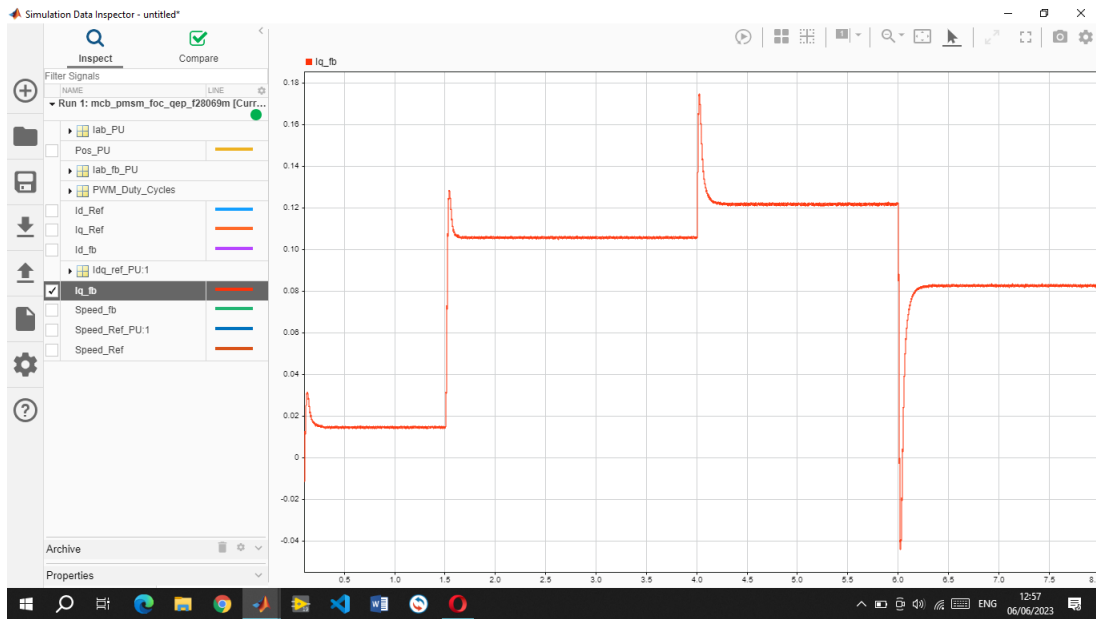


FIGURE V.23 – IQ\_FB

Ce signal représente la valeur de référence de la vitesse du moteur PMSM exprimée en unités relatives (per unit). Les unités relatives sont souvent utilisées pour normaliser les grandeurs électriques et mécaniques afin de faciliter le contrôle.

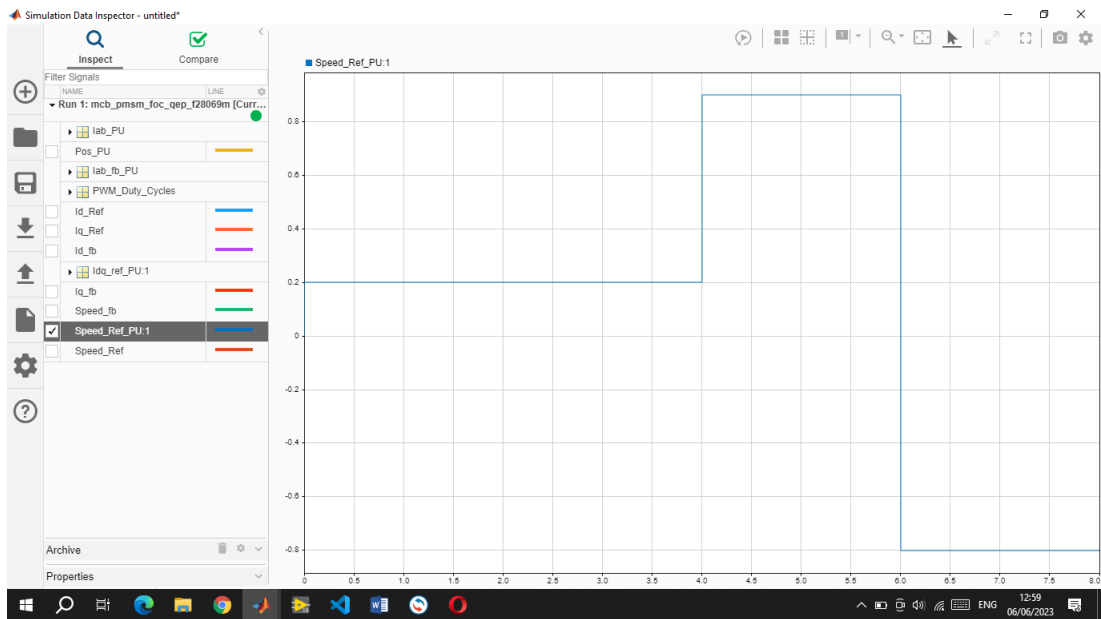


FIGURE V.24 – speed\_ref\_pu :1

Ce signal représente la valeur de référence de la vitesse du moteur PMSM en unités physiques, telles que les tours par minute (RPM) ou les radian par seconde (rad/s). Il s'agit de la valeur de vitesse souhaitée pour le contrôle du moteur PMSM.

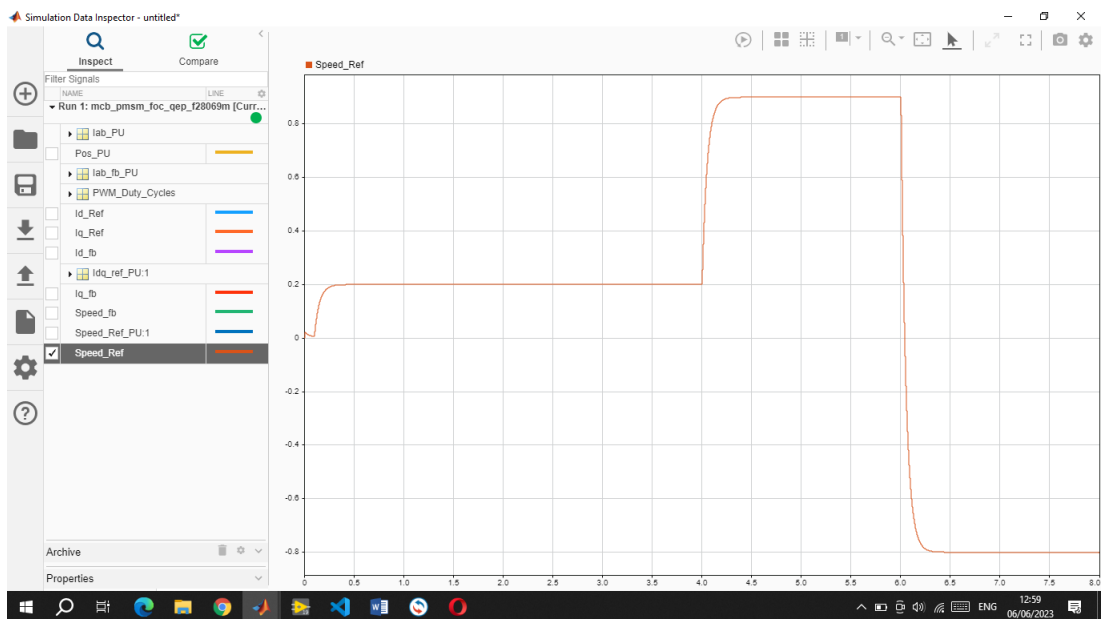


FIGURE V.25 – speed\_ref

## Conclusion

En conclusion, l'utilisation des toolbox DSP dans l'environnement Simulink de MATLAB présente des avantages significatifs pour le développement et la simulation de systèmes de commande d'un MSAP. En utilisant des blocs fonctionnels plutôt que du code texte, la complexité de la programmation est réduite, ce qui facilite la tâche des utilisateurs, même ceux ayant moins d'expérience en programmation bas niveau. L'approche



Hardware in the Loop (HIL) est également introduite comme une méthode de validation et de test,offrant une approche pratique et efficace pour vérifier le fonctionnement du système de commande.Ce chapitre c'est concentré sur l'implémentation software,en décrivant le programme MATLAB utilisé pour la commande du moteur,ainsi que les différents blocs qui le composent et leur fonctionnement respectif.

---

## CONCLUSION GÉNÉRALE

En conclusion, l'électrification des fonctions de commande de vol dans l'aviation a évolué de manière significative avec l'émergence des nouvelles générations d'avions civils et militaires. L'utilisation d'actionneurs électromécaniques (EMA) a progressivement remplacé les systèmes hydrauliques traditionnels, offrant des avantages tels que la fiabilité, la légèreté, l'efficacité énergétique et la réactivité accrue.

Les avancées dans l'électronique de puissance et les calculateurs numériques ont joué un rôle clé dans cette transition. L'électronique de puissance a permis des améliorations dans la conversion, le contrôle et la gestion de l'énergie électrique à des niveaux de puissance élevés, tandis que les calculateurs numériques ont fourni une puissance de calcul accrue, une meilleure fiabilité et une plus grande sécurité pour les systèmes de commande.

La conception et la commande des EMA pour la commande de vol d'un avion nécessitent une compréhension approfondie des systèmes de commande de vol, une modélisation précise des EMA et la mise en place d'une carte de commande efficace. Cela implique la sélection des composants appropriés, la modélisation mathématique des moteurs synchrones à aimants permanents, l'utilisation d'onduleurs triphasés et la mise en œuvre d'algorithmes de commande adaptés.

L'étude de l'EMA pour la commande de vol d'un avion est essentielle pour répondre aux exigences rigoureuses des systèmes de commande de vol modernes. Les progrès dans l'électronique de puissance, les calculateurs numériques et la modélisation des moteurs électriques ont permis de développer des actionneurs plus performants, fiables et efficaces. L'avenir de l'aviation repose sur l'électrification des fonctions de commande de vol, et les EMA jouent un rôle important dans cette évolution technologique.

---

# BIBLIOGRAPHIE

- [1] Ssemanda Denis. "contribution à l'étude thermique d'un moteur synchrone à aimant permanent ", 2015-2016.
- [2] A. Ameer. "commande sans capteur de vitesse par dtc d'une machine synchrone à aimants permanents dotée d'un observateur d'ordre complet à mode glissants ", 2003.
- [3] D. Talah S. Oubellil. "control directe du couple d'une machine synchrone à aimants permanents ", 2008.
- [4] Jérôme MAVIER. *Convertisseurs génériques à tolérance de panne Applications pour le domaine aéronautique*. PhD thesis, DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE, Thèse préparée au Laboratoire Plasma et Conversion d'Energie (LAPLACE) – site ENSEEIHT Unité Mixte de Recherche INPT – UPS – CNRS N°5213, TOULOUSE, 22 mars 2007.
- [5] Texas instrument. *Delfino Enhanced Pulse Width Modulator (ePWM) Module*. Texas Instruments, Post Office Box 655303, Dallas, Texas 75265, <https://engineering.purdue.edu/dionysis/EE452/Lab5/Lab5supportingmaterials/ePWMsprug9e.pdf>, 2009.
- [6] Configuration générale d'un avion. [https://fr.wikipedia.org/wiki/Configuration\\_gnrale\\_d%27un\\_avion](https://fr.wikipedia.org/wiki/Configuration_gnrale_d%27un_avion)
- [7] l'avionnaire. <https://www.lavionnaire.fr/CelluleGouvernes.php>.
- [8] P. BERRARD P. BONNEAU. Commandes de vol ata27. <https://studylibfr.com/doc/8665348/commandes-de-vol-ata-27-partie-66-cat.-a-11.1>.
- [9] Delphine Mami. *Définition, conception et expérimentation de structures d'actionneurs électromécaniques innovants incluant par conception des fonctionnalités de sûreté et de sécurité de fonctionnement*. PhD thesis, DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE, Unité de recherche : LABORATOIRE PLASMA ET CONVERSION D'ENERGIE UMR5213, TOULOUSE, 22 janvier 2010.
- [10] M.BOUHERAOUA. "contribution à l'étude thermique d'un moteur synchrone à aimant permanent ", 2010-2011.
- [11] Sonia MOUZAOUI. Etude d'une machine synchrone à aimants permanents en présence de défauts de courts-circuits et

- de désaimantations par la méthode des éléments finis, 2015.  
<https://www.ummtto.dz/dspace/bitstream/handle/ummtto/8610/MouzaouiSonia.pdf?sequence=1>.
- [12] Betka Achour. Commande vectorielle du moteur synchrone a aimants permanents (msap). [http://elearning.univ-biskra.dz/moodle/pluginfile.php/330879/mod\\_resource/content/1/Commande%20FOC-MSAP.pdf](http://elearning.univ-biskra.dz/moodle/pluginfile.php/330879/mod_resource/content/1/Commande%20FOC-MSAP.pdf).
- [13] Commande vectorielle. [https://www.wikiwand.com/fr/Commande\\_vectorielle](https://www.wikiwand.com/fr/Commande_vectorielle).
- [14] [https://www.researchgate.net/figure/Schema-synoptique-de-la-MLI\\_fig12\\_33323601](https://www.researchgate.net/figure/Schema-synoptique-de-la-MLI_fig12_33323601).
- [15] Modified harvard architecture. <https://www.semanticscholar.org/topic/Modified-Harvard-architecture/1338123>.
- [16] Texas instrument. *TMS320F2833x, TMS320F2823x Real-Time Microcontrollers, SPRS439Q – JUNE 2007 – REVISED AUGUST 2022*.
- [17] Texas instrument. *Delfino Enhanced Quadrature Encoder Pulse (eQEP) Module*. Texas Instruments, Post Office Box 655303, Dallas, Texas 75265, 2009. Literature Number : SPRUG74.
- [18] Servocommandes française (1945-1982), 10 août 1983. extrait d'un ouvrage du GI-FAS.
- [19] Éric Duckler Kenmoe Fankem. *Étude des différentes structures d'actionneurs de positionnement pour l'aéronautique*. PhD thesis, Université de Lorraine, <https://hal.univ-lorraine.fr/tel-01749368/document>, 26 Novembre 2012.
- [20] Dr. VAN DEN BOSSCHE. "more electric control surface actuation ; a standard for the next generation of transport aircraft". *EPE, Toulouse*, 2003.
- [21] P. Wheeler, Jon Clare, Lee Empringham, M. Apap, K. Bradley, Chris Whitley, and Graham Towers. A matrix converter based permanent magnet motor drive for an aircraft actuation system. pages 1295 – 1300 vol.2, 07 2003.
- [22] Matthias Mitter. Bossard, "l'hydraulique dans les commandes de vol de l'a320, aérospatiale", juin 1986. <https://www.bossard.com/global-en/product-solutions/brands/welding-technology/mm-welding/>.
- [23] BELAHBIB Abdelhakim. "commande dtc-svm de la machine synchrone à aimants permanents (msap) par l'utilisation des régulateurs pi du flux et du couple ", 2018.
- [24] BARDADI BENDAHA Mansour. "identification des paramètres d'une machine synchrone à aimant permanent en vue d'une intégration dans des simulateurs en temps réel ", 2015.
- [25] BARIS Bagci. "programming and use of tms320f2812 dsp to control and regulate power electronic converters", October 2003.
- [26] HAMAMI Latifa. "dsp et famille c6000". Ecole Nationale polytechnique d'Alger.
- [27] AMAN Nair Harish Gopala Pillai. "design and development of embedded dsp controllers for power electronic applications", Mai 2006.

- [28] BOUDJIT Kamel. "introduction aux processeurs de signaux dsp". cours à l'Université Saad Dahlab de Blida.

---

## V.7 EPWM

### V.7.1 Active et shadow register

- **Active Register** Le registre actif contrôle le matériel et est responsable des actions que le matériel provoque ou invoque.

- **Shadow Register** Le registre shadow met en mémoire tampon ou fournit un emplacement temporaire pour le registre actif. Il n'a aucun effet direct sur le matériel de contrôle. À un moment stratégique, le contenu du registre shadow est transféré au registre actif. Cela permet d'éviter toute corruption ou toute erreur de fonctionnement due à une modification asynchrone du registre par le logiciel.

### V.7.2 Les registres de epwm

Voici une explication détaillée de chaque registre du module EPWM du TMS320F28335 :

- **TBCTL** : Timer Base Control Register Ce registre permet de configurer le timer de base (TB) qui est utilisé pour générer les signaux PWM. On peut notamment définir la fréquence de base du timer, le mode de comptage, le mode de synchronisation, etc.

- **TBSTS** : Timer Base Status Register Ce registre permet de lire l'état du timer de base (TB). On peut notamment savoir si le timer est en train de compter, si une interruption a été générée, etc.

- **TBPHS** : Timer Base Phase Register Ce registre permet de définir la phase du timer de base (TB). On peut ainsi synchroniser plusieurs signaux PWM en définissant des phases différentes pour chaque signal.

- **TBCTR** : Timer Base Counter Register Ce registre permet de lire ou d'écrire la valeur du compteur du timer de base (TB). Cette valeur est utilisée pour générer les signaux PWM en fonction de la fréquence de base définie dans le registre TBCTL.

- **CMPCTL** : Compare Control Register Ce registre permet de configurer le comparateur de sortie (CMP) qui est utilisé pour générer les signaux PWM. On peut notamment définir le mode de comparaison (up, down, up-down), le mode de charge/décharge des sorties, etc.

- **CMPAHR** : Compare A High Resolution Register Ce registre permet de définir la

partie haute de la valeur de comparaison pour la sortie PWM A. Cette valeur est utilisée pour définir la durée de l'impulsion PWM.

- **CMPALR** : Compare A Low Resolution Register Ce registre permet de définir la partie basse de la valeur de comparaison pour la sortie PWM A. Cette valeur est utilisée pour définir la durée de l'impulsion PWM avec une résolution plus faible que celle du registre CMPAHR.

- **CMPBHR** : Compare B High Resolution Register Ce registre permet de définir la partie haute de la valeur de comparaison pour la sortie PWM B. Cette valeur est utilisée pour définir la durée de l'impulsion PWM.

- **CMPBLR** : Compare B Low Resolution Register Ce registre permet de définir la partie basse de la valeur de comparaison pour la sortie PWM B. Cette valeur est utilisée pour définir la durée de l'impulsion PWM avec une résolution plus faible que celle du registre CMPBHR.

- **AQCTL** : Action Qualifier Control Register Ce registre permet de configurer les actions à effectuer sur les sorties PWM en fonction des événements de comparaison. On peut notamment définir les actions à effectuer lorsqu'une comparaison est égale ou inférieure/supérieure à la valeur de référence.

- **AQSFRC** : Action Qualifier Software Force Register Ce registre permet de forcer manuellement une action sur les sorties PWM en utilisant un signal logiciel.

- **AQCSFRC** : Action Qualifier Continuous Software Force Register Ce registre permet de forcer continuellement une action sur les sorties PWM en utilisant un signal logiciel.

- **DBCTL** : Dead-Band Control Register Ce registre permet de configurer le module de blanking (DB) qui permet de supprimer les transitions indésirables sur les sorties PWM. On peut notamment définir la durée de la période de blanking, le mode de fonctionnement, etc.

- **DBFED** : Dead-Band Falling Edge Delay Count Register Ce registre permet de définir le délai de la transition de la sortie PWM A à la sortie PWM B lorsqu'une transition descendante est détectée.

- **DBRED** : Dead-Band Rising Edge Delay Count Register Ce registre permet de définir le délai de la transition de la sortie PWM B à la sortie PWM A lorsqu'une transition montante est détectée.

- **TZSEL** : Trip Zone Select Register Ce registre permet de sélectionner les zones de déclenchement (trip zones) qui peuvent générer une interruption ou une action sur les sorties PWM.

- **TZCTL** : Trip Zone Control Register Ce registre permet de configurer les zones de déclenchement (trip zones) qui peuvent générer une interruption ou une action sur les sorties PWM.

- **TZEINT** : Trip Zone Enable Interrupt Register Ce registre permet d'activer ou de désactiver les interruptions générées par les zones de déclenchement (trip zones).

- **TZFLG** : Trip Zone Flag Register Ce registre permet de lire l'état des zones de déclenchement (trip zones) qui ont généré une interruption ou une action sur les sorties PWM.

- **ETSEL** : Event Trigger Select Register Ce registre permet de sélectionner les événements qui peuvent déclencher une action sur les sorties PWM.
- **ETPS** : Event Trigger Pre-Scale Register Ce registre permet de définir le pré-diviseur pour les événements qui peuvent déclencher une action sur les sorties PWM.
- **ETFLG** : Event Trigger Flag Register Ce registre permet de lire l'état des événements qui ont déclenché une action sur les sorties PWM.
- **ETFRC** : Event Trigger Force Register Ce registre permet de forcer manuellement un événement qui peut déclencher une action sur les sorties PWM.
- **PCCTL** : PWM Chopper Control Register Ce registre permet de configurer le module de hachage (chopper) qui permet de réduire les pertes de commutation sur les sorties PWM.
- **HRCNFG** : HRPWM Configuration Register Ce registre permet de configurer le module HRPWM (High Resolution PWM) qui permet d'augmenter la résolution des signaux PWM.
- **HRCTL** : HRPWM Control Register Ce registre permet de contrôler le module HRPWM (High Resolution PWM) qui permet d'augmenter la résolution des signaux PWM.
- **HRMSTEP** : HRPWM MEP Step Register Ce registre permet de définir le pas de modulation (MEP) pour le module HRPWM (High Resolution PWM).
- **HRMCTRL** : HRPWM MEP Control Register Ce registre permet de contrôler le module MEP (Modulation Edge Position) pour le module HRPWM (High Resolution PWM).
- **HRMEVENT** : HRPWM MEP Event Register Ce registre permet de définir les événements qui peuvent déclencher une action sur le module MEP (Modulation Edge Position) pour le module HRPWM (High Resolution PWM).
- **HRMLOAD** : HRPWM MEP Load Register Ce registre permet de charger une valeur dans le module MEP (Modulation Edge Position) pour le module HRPWM (High Resolution PWM).
- **HRMDBG** : HRPWM MEP Debug Register Ce registre permet de lire l'état du module MEP (Modulation Edge Position) pour le module HRPWM (High Resolution PWM) à des fins de débogage.

Ces registres permettent de configurer et de contrôler le fonctionnement du module EPWM du TMS320F28335. Chaque registre a une fonction spécifique et permet de définir des paramètres tels que la fréquence de base du timer, la durée des impulsions PWM, les actions à effectuer sur les sorties PWM, etc.

### V.7.3 les signaux

Ces signaux sont utilisés dans le module ePWM pour le contrôle de la synchronisation :

- **CTR (Counter)** : Il représente le comptage actuel du module ePWM.
- **PRD (Period)** : Il détermine la période de la forme d'onde ePWM et correspond au nombre maximal de compteurs avant qu'ils ne soient réinitialisés.
- **ZERO** : C'est la valeur à laquelle le compteur est remis à zéro.



- **CMPB (Compare B)** : Il détermine le point auquel la sortie de l'ePWM passe de l'état haut à l'état bas.
- **CPMA (Compare A)** : Il détermine le point auquel la sortie de l'ePWM passe d'un niveau bas à un niveau élevé.
- **CTR\_Dir (Counter Direction)** : Il détermine la direction du compteur, soit vers le haut, soit vers le bas.

La relation entre ces signaux est utilisée pour créer diverses formes d'ondes de modulation de largeur d'impulsion, qui sont largement utilisées dans les applications d'électronique de puissance et de commande de moteur.

- **EPWMxSYNCI** : Ce signal d'entrée est utilisé pour synchroniser plusieurs modules ePWM. Lorsque ce signal est activé, le compteur de base de temps est remis à zéro.
- **EPWMxSYNCO** : Ce signal de sortie est utilisé pour synchroniser d'autres modules ePWM. Ce signal est activé lorsque le compteur de base de temps atteint zéro.
- **TBCTL[SWFSYNC]** : Ce bit est utilisé pour forcer une synchronisation logicielle du module ePWM. Lorsque ce bit est activé, le compteur de base de temps est remis à zéro.
- **TBCTL[SYNCOSEL]** : Ce bit est utilisé pour sélectionner la source du signal SYNCO. Lorsque ce bit est activé, le signal SYNCO est généré en fonction de la valeur de TBCTL [SYNCI\_SEL]. Lorsque ce bit est désactivé, le signal SYNCO est généré en fonction de la valeur de TBCTL[PRDL].
- **TBCTL[PHSEN]** : Ce bit est utilisé pour activer/désactiver la synchronisation de phase. Lorsque ce bit est activé, la phase du compteur de base de temps est synchronisée avec la phase du signal SYNCI.
- **TBPRD shadow (16)** : Ce registre est utilisé pour définir la période du compteur de base de temps. Le registre shadow est chargé avec la valeur du registre actif lorsque le TBCTR atteint zéro.
- **TBPRD active (16)** : Ce registre est utilisé pour définir la période du compteur de base de temps. Le registre actif est utilisé pendant le fonctionnement normal du module ePWM.
- **TBPHSHR (8)** : Ce registre est utilisé pour définir la phase du compteur de base de temps. La valeur de ce registre est décalée de 8 bits vers la gauche et ajoutée à la valeur de TBCTR.
- **TBCTR active (16)** : Ce registre est utilisé comme compteur de base de temps. La valeur de ce registre est incrémentée à chaque cycle d'horloge du module ePWM. Lorsque la valeur de ce registre correspond à la valeur du registre TBPRD, le compteur est remis à zéro.

## V.8 EQEP

### V.8.1 Les registres de module QEP

Le module eQEP comporte généralement un ensemble de registres de configuration et de contrôle qui permettent de programmer et de contrôler la fonctionnalité eQEP. Voici

quelques registres couramment trouvés dans le module eQEP :

**QDECCTL** : Quadrature Decoder Control Register (registre de contrôle du décodeur en quadrature) - Ce registre est utilisé pour configurer le mode de fonctionnement, le mode de capture des fronts et d'autres paramètres liés à la fonctionnalité du décodeur en quadrature.

**QEPCTL** : registre de contrôle QEP - Ce registre contrôle le comportement général du module eQEP, comme l'activation ou la désactivation du module, la configuration des options de réinitialisation du compteur de positions et la sélection du mode de temporisation de l'unité.

**QPOSCTL** : registre de contrôle de la position - Ce registre permet de configurer le comportement du compteur de position, notamment les options de réinitialisation du compteur de position, les modes de comparaison de position et la direction du compteur de position.

**QCAPCTL** : registre de contrôle de capture - Ce registre est utilisé pour configurer l'unité de capture, qui capture l'horodatage d'événements tels que les impulsions d'index ou d'autres événements externes.

**QPOSINIT** : registre d'initialisation du compteur de positions - Ce registre définit la valeur initiale du compteur de positions, ce qui permet d'initialiser la position du codeur à une valeur connue.

**QPOSMAX** : Position Counter Maximum Register (registre du maximum du compteur de positions) - Ce registre définit la valeur maximale que le compteur de positions peut atteindre avant de s'arrêter ou de déclencher une interruption.

**QEINT** : eQEP Interrupt Enable Register - Ce registre active ou désactive des interruptions ou des événements spécifiques générés par le module eQEP, ce qui permet un fonctionnement par interruption ou le déclenchement de routines en fonction d'un événement.

| NAME     | eQEP1 ADDRESS   | eQEP2 ADDRESS   | eQEP1 SIZE(x16)/ #SHADOW | REGISTER DESCRIPTION                   |
|----------|-----------------|-----------------|--------------------------|--|
| QPOSCNT  | 0x6B00          | 0x6B40          | 2/0                      | eQEP Position Counter                  |
| QPOSINIT | 0x6B02          | 0x6B42          | 2/0                      | eQEP Initialization Position Count     |
| QPOSMAX  | 0x6B04          | 0x6B44          | 2/0                      | eQEP Maximum Position Count            |
| QPOSCMP  | 0x6B06          | 0x6B46          | 2/1                      | eQEP Position-compare                  |
| QPOSILAT | 0x6B08          | 0x6B48          | 2/0                      | eQEP Index Position Latch              |
| QPOSSLAT | 0x6B0A          | 0x6B4A          | 2/0                      | eQEP Strobe Position Latch             |
| QPOSLAT  | 0x6B0C          | 0x6B4C          | 2/0                      | eQEP Position Latch                    |
| QUTMR    | 0x6B0E          | 0x6B4E          | 2/0                      | eQEP Unit Timer                        |
| QUPRD    | 0x6B10          | 0x6B50          | 2/0                      | eQEP Unit Period Register              |
| QWDTMR   | 0x6B12          | 0x6B52          | 1/0                      | eQEP Watchdog Timer                    |
| QWDPRD   | 0x6B13          | 0x6B53          | 1/0                      | eQEP Watchdog Period Register          |
| QDECCTL  | 0x6B14          | 0x6B54          | 1/0                      | eQEP Decoder Control Register          |
| QEPCTL   | 0x6B15          | 0x6B55          | 1/0                      | eQEP Control Register                  |
| QCAPCTL  | 0x6B16          | 0x6B56          | 1/0                      | eQEP Capture Control Register          |
| QPOSCTL  | 0x6B17          | 0x6B57          | 1/0                      | eQEP Position-compare Control Register |
| QEINT    | 0x6B18          | 0x6B58          | 1/0                      | eQEP Interrupt Enable Register         |
| QFLG     | 0x6B19          | 0x6B59          | 1/0                      | eQEP Interrupt Flag Register           |
| QCLR     | 0x6B1A          | 0x6B5A          | 1/0                      | eQEP Interrupt Clear Register          |
| QFRC     | 0x6B1B          | 0x6B5B          | 1/0                      | eQEP Interrupt Force Register          |
| QEPSTS   | 0x6B1C          | 0x6B5C          | 1/0                      | eQEP Status Register                   |
| QCTMR    | 0x6B1D          | 0x6B5D          | 1/0                      | eQEP Capture Timer                     |
| QCPRD    | 0x6B1E          | 0x6B5E          | 1/0                      | eQEP Capture Period Register           |
| QCTMRLAT | 0x6B1F          | 0x6B5F          | 1/0                      | eQEP Capture Timer Latch               |
| QCPRLAT  | 0x6B20          | 0x6B60          | 1/0                      | eQEP Capture Period Latch              |
| Reserved | 0x6B21 – 0x6B3F | 0x6B61 – 0x6B7F | 31/0                     |  |

FIGURE V.26 – Les registres de QEP

## V.8.2 les registres utilisés dans le code ccs

Les deux registres utilisées dans ce code sont :le registre de contrôle décodeur QDECCTL et le registre de contrôle de QEP :QEPCTL.

**QDECCTL :**

|       |       |       |       |          |       |       |       |
|-------|-------|-------|-------|----------|-------|-------|-------|
| 15    | 14    | 13    | 12    | 11       | 10    | 9     | 8     |
| QSRC  | SOEN  | SPSEL | XCR   | SWAP     | IGATE | QAP   |       |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0    | R/W-0 | R/W-0 | R/W-0 |
| 7     | 6     | 5     | 4     | Reserved |       |       | 0     |
| QBP   | QIP   | QSP   |       | Reserved |       |       |       |
| R/W-0 | R/W-0 | R/W-0 |       | R-0      |       |       |       |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

FIGURE V.27 – Registre de contrôle du décodeur QEP (QDECCTL)

- Le bit QSRC a pour but de Sélectionner la source du compteur de position, dans notre exemple le choix effectuer est pour le Mode de comptage en quadrature

(QCLK = iCLK, QDIR = iDIR).

- Le bit SOEN est pour l'activation de la sortie de synchronisation : le 0 bit pour la Désactivation et le 1 bit pour Activation.
- Le bit SPSEL Sélectionne la broche de sortie de synchronisation : soit La broche Index (0), soit La broche stroboscopique(1) qui est utilisée pour la sortie de synchronisation.
- Le bit XCR détermine la Fréquence d'horloge externe : Comptage du front montant/descendant ou Comptage du front montant uniquement.
- Le bit SWAP Intervertir les entrées d'horloge en quadrature. Cette opération permet d'intervertir l'entrée du décodeur en quadrature, en inversant le sens de comptage, et pour cela on distingue deux cas :l'inversion des entrées de l'horloge en quadrature ou la non inversion de ces derniers.

### QEPCTL :

|            |    |       |    |       |    |       |   |       |       |       |   |       |       |       |       |
|------------|----|-------|----|-------|----|-------|---|-------|-------|-------|---|-------|-------|-------|-------|
| 15         | 14 | 13    | 12 | 11    | 10 | 9     | 8 | 7     | 6     | 5     | 4 | 3     | 2     | 1     | 0     |
| FREE, SOFT |    | PCRM  |    | SEI   |    | IEI   |   | SWI   | SEL   | IEL   |   | QPEN  | QCLM  | UTE   | WDE   |
| R/W-0      |    | R/W-0 |    | R/W-0 |    | R/W-0 |   | R/W-0 | R/W-0 | R/W-0 |   | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

FIGURE V.28 – Registre de contrôle eQEP (QEPCTL)

- Le bit PCRM :  
Mode de réinitialisation du compteur de position dans notre cas il est réinitialiser à l'événement d'index donc remise à zéro.
- Le bit IEI :  
Index événement initialisation du compteur de position Le registre active les interruptions sur les évènement d'index.
- Le bit SWI :  
il utiliser pour l'initialisation logicielle du compteur de position, le 1 vu dire que le compteur de position est initialiser à (QPOSCNT=QPOSINIT).
- Le bit OPEN :  
Pour l'activation du compteur de position en quadrature ou la réinitialisation logicielle.  
1 Le compteur de position eQEP est activé.

- Le bit QCLM : Mode de verrouillage de la capture eQEP Le bit 0 pour le Verrouillage du compteur de position lu par l'unité centrale. Les valeurs du temporisateur de capture et de la période de capture sont verrouillées dans les registres QCTMRLAT et QCPRDLAT lorsque l'UC lit le registre QPOSCNT. Le bit 1 pour le Verrouillage en cas de dépassement du temps de l'unité. Les valeurs du compteur de position, de la minuterie de capture et de la période de capture sont verrouillées dans les registres QPOSLAT, QCTMRLAT et QCPRDLAT à la sortie de l'unité.

**Registre de compteur de position QPOSCNT :**

Ce registre de compteur de position 32 bits compte vers le haut/bas à chaque impulsion eQEP en fonction de l'entrée de direction. Ce compteur agit comme un intégrateur de position dont la valeur de comptage est proportionnelle à la position par rapport à un point de référence donné.

**Registre d'initialisation du compteur de position eQEP (QPOSINIT) :**

Ce registre contient la valeur de position utilisée pour initialiser le compteur de position en fonction d'un événement stroboscopique externe ou d'un événement d'index. Le compteur de position peut être initialisé par logiciel.

**Le registre QPOSCMP :**

La valeur de comparaison de position dans ce registre est comparée au compteur de position (QPOSCNT) afin de générer une sortie de synchronisation et/ou une interruption en cas de comparaison.

**Le registre Index Position Latch (QPOSILAT) :**

. La valeur du compteur de position est mise en latence dans ce registre lors d'un événement d'indexation défini par les bits QEPCTL[IEL].

**Le registre Strobe Position Latch(QPOSSLAT) :**

La valeur du compteur de position est verrouillée dans ce registre lors d'un événement stroboscopique défini par les bits QEPCTL[SEL].

**Le registre Position Counter Latch(QPOSLAT) :**

La valeur du compteur de position est verrouillée dans ce registre en cas de dépassement du temps de l'unité.

**Le registre eQEP Période d'unité (QUPRD) :**

Ce registre contient le comptage de la période pour la minuterie de l'unité afin de générer des événements périodiques de temps unitaire pour verrouiller les informations de position eQEP, et éventuellement, pour générer une interruption.

## ملخص

يتعلق هذا الموجز بالتحكم في مشغل كهروميكانيكي (EMA) للتحكم في طيران مناورات الطائرات. المشغل مهم في التحكم في الطائرة، مما يسمح بحركات دقيقة ومتفاعلة لأسطح التحكم الديناميكية الهوائية. تتطلب مراقبة EMA نمذجة تشغيلها، ودراسة على لوحة التحكم وهذه الكتل المختلفة التي يتم عرضها جميعاً في هذا العمل، وبالتالي فإن الهدف النهائي هو التحكم في EMA القادر على توفير أسطح تحكم دقيقة وأمنة لأجنحة الطائرة، مما يساهم في السلامة وأداء الطيران.

الكلمات الرئيسية: محرك PMSM المترام، FOC، لوحة EQEP، ADC، PWM، DSP، عاكس الطور الثلاثة.

## Abstract

This thesis deals with the control of an electromechanical actuator (EMA) for the flight control of aircraft manoeuvres. The actuator is important in the piloting of the aircraft, allowing precise and reactive movements of the aerodynamic control surfaces. Control of the EMA requires modelling of its operation, a study of the control board and its various blocks, all of which are presented in this work. The ultimate objective is to control an EMA capable of providing precise and safe control of the control surfaces of aircraft wings, thereby contributing to flight safety and performance.

**Key words:** PMSM synchronous motor, FOC, DSP board, PWM, ADC, EQEP, Three-phase inverter.

## Résumé

Ce mémoire porte sur le contrôle d'un actionneur électromécanique (EMA) pour la commande de vol des manœuvres d'un avion. L'actionneur est important dans le pilotage de l'avion, permettant des mouvements précis et réactifs des gouvernes aérodynamiques. La commande de l'EMA nécessite une modélisation de son fonctionnement, une étude sur la carte de commande et ces différents blocs qui sont tous présentés dans ce travail, donc l'objectif ultime est de contrôler un EMA capable de fournir un contrôle précis et sûr des surfaces de commande des ailes d'avion, contribuant ainsi à la sécurité et aux performances de vol.

**Mots clés:** Moteur synchrone PMSM, FOC, Carte DSP, PWM, ADC, EQEP, Onduleur triphasé.