AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

Ecole doctorale IAEM Lorraine

Simulation temps réel de dispositifs électrotechniques

THÈSE

présentée et soutenue publiquement le 15 Mai 2014

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Électrotechnique)

par

RAKOTOZAFY Andriamaharavo Mamianja

Composition du jury

*P*résident :	Philippe Lataire, Professeur, VUB, Bruxelles
*R*apporteurs :	Fouad Benkhoris, Professeur, IREENA, Saint-Nazaire Philippe Ladoux, Professeur, LAPLACE, Toulouse
*E*xaminateurs :	Abdollah Mirzaian, Engineering Manager, GE, Belfort Sami Siala, Senior Manager R&D, GE, Massy Philippe Poure, Maître de conférence, Hdr, IJL, UL Shahrokh Saadate, Professeur, GREEN, UL

Groupe de Recherche en Électrotechnique et Électronique de Nancy
Institut Jean Lamour

Remerciements

A mes parents Michou sy Vero,
A ma soeur Balisoa et sa famille : Leonce, Hir'iela
A ma famille Nalini, Adrian, Mitantsiky, Mailalatia,
A toute la grande famille de Madagascar, à Mama.

Je remercie vivement mes encadreurs universitaires Monsieur Shahrokh SAADATE et Monsieur Philippe POURE pour leur confiance et leurs investissements.
Je remercie également Monsieur Sami SIALA pour sa confiance , son support et ses propositions.

Merci au président du Jury Monsieur Philippe LATAIRE, ainsi qu'aux rapporteurs Monsieur Fouad BENKHORIS et Monsieur Philippe LADOUX d'avoir amené un regard critique externe à ce travail.

Merci à Monsieur Loïc LECLERE et Monsieur Cédric BORDAS pour avoir initier ces travaux de thèse.

Aux collègues et amis de Belfort : Jérôme, Mathieu, Pascal, Stéphane,

Aux collègues et amis de Massy : Thufail, Guillaume, Wissam, Yohan, Abdel, Jean-Luc, Jean-Pierre, Daniel, Lionel, Philippe

Aux collègues d'EDC : Prakash, Kartik, Sandeep, Jayakumar, Kotte

Ma thèse fut aussi bien une expérience humaine qu'une expérience technique.

Table des matières

Table des matières iii

Glossaire vii

Acronymes ix

Liste des symboles xi

Table des figures xii

Liste des tableaux xv

Introduction Générale 1

1 Simulation des Systèmes Électrotechniques 7
1.1 Introduction . . . 7
1.2 Modèles à temps continu pour la simulation . . . 9
1.2.1 Représentation par les équations différentielles . . . 9
1.2.2 Représentation d'état . . . 10
1.2.2.1 Principe . . . 11
1.2.2.2 Exemple . . . 11
1.2.2.3 Fonction de transfert . . . 11
1.2.2.4 Remarque . . . 12
1.2.3 Représentation modale . . . 12
1.2.4 Représentation nodale . . . 13
1.2.4.1 Principes . . . 13
1.2.4.2 Exemple . . . 14
1.2.4.3 Remarques . . . 15
1.3 Discrétisation . . . 15
1.3.1 Méthodes d'intégration numérique et stabilité . . . 15
1.3.1.1 Méthodes d'intégration du premier ordre . . . 16
1.3.1.1.1 Méthode d'Euler avant . . . 16
1.3.1.1.2 Méthode d'Euler arrière . . . 16
1.3.1.1.3 Méthode trapézoïdale . . . 17
1.3.1.2 Méthode d'intégration d'ordre 2 . . . 17
1.3.1.2.1 Méthode de Gear d'ordre 2 . . . 18
1.3.1.3 Méthodes d'intégration explicites - Définition . . . 18
1.3.1.4 Méthodes d'intégration implicites - Définition . . . 19
1.3.1.5 Amortissement numérique . . . 19
1.3.1.6 Stabilité des méthodes d'intégration . . . 19
1.3.1.6.1 Stabilité de la méthode d'Euler avant . . . 19

1.3.1.6.2 Stabilité de la méthode d'Euler arrière 20
1.3.1.6.3 Stabilité de la méthode trapézoïdale 20
1.3.1.6.4 Méthodes d'intégration A-stables 21
1.3.1.7 Exemple 21
1.3.1.8 Discussions 22
1.3.2 Méthodes de discrétisation analytiques 22
1.3.2.1 Transformée inverse de Laplace 23
1.3.2.1.1 Décomposition en éléments simples 23
1.3.2.1.2 Méthode des résidus I 24
1.3.2.1.3 Méthode des résidus II 24
1.3.2.1.4 Inversion numérique de la transformée de Laplace . . . 25
1.3.2.2 Résolution temporelle exacte 25
1.3.2.2.1 Principe 25
1.3.2.2.2 Calcul de l'exponentielle 26
1.4 Modélisations directes à temps discret 30
1.4.1 Méthode de *Dommel* - Méthode du Circuit Compagnon 30
1.4.1.1 Principe 30
1.4.1.2 Calcul de la résistance R_d et de la fonction linéaire f 31
1.4.2 Latency Insertion Method 32
1.4.3 Lignes de transmission 33
1.5 Cas particulier des convertisseurs statiques 34
1.5.1 Solutions pour les changements de topologie 35
1.5.1.1 Modélisation instantanée 35
1.5.1.2 Modélisation par interrupteurs impédants 36
1.5.1.2.1 Interrupteur à résistance variable 36
1.5.1.2.2 Interrupteur L/C 36
1.5.1.2.3 Interrupteur L/RC 39
1.5.2 Solutions apportées au problème de résolution 40
1.5.2.1 Modèle moyen 40
1.5.2.2 Interface de pré-traitement 40
1.6 Bases matérielles pour la simulation temps réel 41
1.6.1 Simulateurs à base de cœurs 41
1.6.2 Simulateurs sur FPGA 42
1.6.3 Comparaison des deux types de bases matérielles 42
1.6.4 Approches et méthodes selon les bases matérielles 43
1.7 Optimisation et réduction de coût 44
1.7.1 Techniques de partitionnement des modèles 44
1.7.1.1 Techniques de partitionnement 44
1.7.1.2 Effet du partitionnement 44
1.7.1.3 Sur-échantillonnage 46
1.7.1.4 Sous-échantillonnage 46
1.7.2 Traitement des matrices creuses 46
1.7.3 Algorithmes de réduction du nombre d'opération de calcul 46
1.7.4 Pré-enregistrement des matrices de transfert discrètes à durée quantifiée . 47
1.8 Conclusion 48

2 Validation temps réel par interface logicielle 53
2.1 Introduction 53
2.2 Principe de la VIL 54
2.2.1 Cas où les instants de commutation sont calculés *a priori* par l'Unité Logicielle (UL) 54
2.2.2 Cas où les instants de commutation sont traités sur la UGES 58

2.2.3 Avantages et limitations de la VIL . . . 58
2.3 Modélisation du système de puissance . . . 58
2.3.1 Principe de l'interpolation évènementielle . . . 59
2.3.2 Modèle discrétisé pour la VIL . . . 61
2.3.2.1 Modèle discret exact . . . 61
2.3.2.2 Modèle discret approximatif . . . 61
2.4 Mise en œuvre de la validation . . . 63
2.5 Étude de cas : VIL d'un ensemble convertisseur/machine . . . 65
2.5.1 Modèle du convertisseur . . . 66
2.5.2 Modèle de la machine asynchrone . . . 69
2.5.2.1 Discrétisation de la partie électrique de la machine asynchrone . 69
2.5.2.1.1 Discrétisation exacte du modèle de la partie électrique de la machine asynchrone . . . 69
2.5.2.1.2 Discrétisation approximative de la partie électrique de la machine asynchrone . . . 78
2.5.2.2 Comparaison des deux modèles discrets de la partie électrique . 79
2.5.3 Résultats . . . 80
2.6 Conclusion . . . 86

3 Validation par Interface Matérielle 91
3.1 Introduction . . . 91
3.2 Simulateurs 'Tout FPGA' . . . 91
3.2.1 Modélisation de l'onduleur . . . 94
3.2.1.1 Mode de fonctionnement normal . . . 95
3.2.1.2 Mode de fonctionnement haute impédance . . . 97
3.2.1.3 Mode de fonctionnement redresseur . . . 97
3.2.1.4 Fonction séquentielle . . . 98
3.2.2 Modélisation de la machine asynchrone . . . 99
3.2.3 Implémentation . . . 100
3.2.4 Résultats . . . 103
3.3 Simulateur FPGA/Processeur . . . 104
3.3.1 Simulateurs FPGA/Processeur rapide . . . 104
3.3.1.1 Modélisation de l'onduleur trois niveaux . . . 107
3.3.1.1.1 Méthode basée sur la tension moyenne . . . 107
3.3.1.1.2 Méthode par les impulsions moyennées . . . 107
3.3.1.2 Validation du modèle du convertisseur Neutral Point Clamped converter (NPC) sur une charge *RLE* . . . 112
3.3.1.3 Cas d'une machine asynchrone alimentée par deux onduleurs . . 117
3.3.1.3.1 Modèle de la charge dans le cas où un seul onduleur est actif . . . 117
3.3.1.3.2 Modèle de la charge dans le cas de deux onduleurs actifs 119
3.3.1.3.3 Courants de ligne de chaque convertisseur . . . 120
3.3.1.3.4 Courants de circulation . . . 121
3.3.1.3.5 Procédure de validation . . . 121
3.3.1.3.6 Résultats . . . 121
3.3.2 Simulateur sur FPGA/Processeur lent . . . 123
3.3.2.1 Horodatage de la tension . . . 125
3.3.2.2 Validation de l'horodatage . . . 126
3.4 Conclusion . . . 128

Conclusion Générale 133

Appendices 139

A Rappel sur le calcul des résidus **141**

B Exemples pour la deuxième méthode des résidus **143**

C Exemples pour la méthode temporelle exacte **145**

D Modèle continu de la machine asynchrone **149**
D.1 Hypothèses 149
D.2 Modélisation 150
D.2.1 Tensions statoriques et rotoriques 150
D.2.2 Flux statoriques et rotoriques 151
D.2.3 Diagonalisation des matrices inductances L_S et L_R 151
D.2.3.1 Matrices de transformation P pour diagonaliser 152
D.2.4 Représentation d'état de la partie électrique de la machine asynchrone . . 154
D.2.4.1 Flux statoriques et rotoriques dans le repère $(DQ0)$ 154
D.2.4.2 Equations des tensions statoriques et rotoriques dans le repère $(DQ0)$ 155
D.2.4.3 Équations d'état dans le repère (DQ) en utilisant les courants statoriques et rotoriques 158
D.2.5 Modélisation de la partie mécanique 160
D.2.6 Différentes variantes du modèle d'état de la machine asynchrone 161
D.2.6.1 Variantes en fonction des variables d'état 161
D.2.6.2 Variante en fonction du positionnement du repère (DQ) 162
D.2.6.3 Variante en fonction de la tension rotorique 163
D.2.6.4 Variante machine asynchrone à neutre non relié ou fortement impédant 163

Bibliographie **165**

Glossaire

A-stable : Caractéristique de certaines méthodes d'intégration. v, 21, 38, 80

Explicite : se dit des méthodes d'intégration numériques utilisant les valeurs précédentes pour calculer les valeurs actuelles. v, 18, 22, 46

Implicite : se dit des méthodes d'intégration numériques utilisant les valeurs précédentes et des valeurs actuelles pour calculer les valeurs actuelles. Par exemple, pour calculer le courant passant par une inductance à l'instant actuel, une méthode d'intégration numérique implicite utiliserait les valeurs du courant aux instants précédents et la valeur actuelle de la tension aux bornes de cette inductance (et éventuellement des valeurs précédentes de la tension). v, 19, 22

Temps différé : se dit des systèmes numériques calculant avec une base de temps plus grande ce qui se serait produit dans la réalité. Ceci permet de retracer des phénomènes dans le temps sans pour autant donner les résultats dans les délais réels. Ces systèmes sont dédiés à des fins purement de simulation sans contrainte de temps de calcul. Une simulation d'une heure d'un parc éolien peut par exemple prendre deux jours pour l'obtention des résultats. La validation d'un contrôleur physique par exemple nécessite de lui fournir avec une résolution supérieure à la sienne ses signaux de retour. Le contrôleur pourra ainsi mettre à jour ses signaux de commande et leurs effets à chaque fois qu'il fait des acquisitions. v, 9

Temps réel : se dit des systèmes numériques ayant des temps de calcul inférieurs au temps d'établissement des signaux. Ceci permet de générer des signaux de dans le temps avec une base de temps réel (contrairement au temps différé). v, 7–9, 15, 19, 22, 34–36, 38, 40–43, 47–49, 147

Acronymes

3PC Triple Processor Card. v, 41

ABB ASEA - Brown - Boveri & Cie. v, 39

ACSL Advanced Continuous Simulation Language. v, 10

BT Basse Tension. v, 3

CAO Conception Assistée par Ordinateur. v, 15

CIFRE Convention Industrielle de Formation par la Recherche. v, 1

DITS Digital Input Times Stamp. v, 41

DSP Digital Signal Processor. v, 42, 43, 53, 91

EMTP ElectroMagnetic Transient Program. v, 13, 39

FEM Force ElectroMotrice. v, 94, 97

FPGA Field Programmable Gate Array. v, vii–ix, 4, 38, 41–43, 48, 49, 53, 81, 91–95, 100–108, 112, 123–126, 128, 129, 133, 134

FPSO Floating Production, Storage and Offloading. v, 1

GE-PC General Electric - Power Conversion ©. v, vi, 1, 40, 66, 75, 84, 92, 107

GIC Graphe Informationnel Causal. v, 35

GPC Giga Processor Card. v, 41

GREEN Groupe de Recherche en Électrotechnique et Électronique de Nancy. v, 1

HIL Hardware In the Loop. v, 7, 91

IGBT Insulated Gate Bipolar Transistor. v, 42, 59, 66, 97, 101

IJL Institut Jean Lamour de Nancy. v, 1

LIEN Laboratoire d'Instrumentation Électronique de Nancy. v, 1

LIM Latency Insertion Method. v, vi, 32, 33

MLI Modulation de Largeur d'Impulsion. v, vi, 2, 53–55, 57, 58, 61, 83, 84, 86, 87, 122, 128

N2EV Nanomatériaux, Electronique Et Vivant. v, 1

NPC Neutral Point Clamped converter. iii, v–viii, 66–69, 104, 106, 107, 109, 112, 113, 117, 125

NPP Neutral Point Piloted converter. v–vii, ix, 66–69, 93, 94, 96, 97, 123

PHIL Power Hardware In the Loop. v, 7, 8, 44

PIBe Power Interface Board EtherCAT. v, 81, 83, 91, 92, 106

PIC Peripheral Interface Controller. v, 41

PLECS Piece-wise Linear Electrical Circuit Simulation. v, viii, 10, 65, 83, 112, 113, 122

PSB Power System Blockset. v, 10

RCM Resistive Companion Method. v, 30, 31, 33, 36, 133

RISC Reduced Instruction Set Computer. v

RPC RISC Processor Card. v, 41

RT-LAB Real-Time Laboratory. v, 41, 42, 106, 107, 121, 122

RTDS Real-Time Digital Simulator. v, 39–41, 46

RTW Real Time Workshop. v, 64

SIL Software In the Loop. v, 7

SLIT Système Linéaire Invariant dans le Temps. v, 9

SPICE Simulation Program with Integrated Circuit Emphasis. v, 13

SPS SimPowerSystem. v, 65

TBTS Très Basse Tension de Sécurité. v, 3

TNA Transient Network Analyser. v, 3, 7

TSB Time Stamped Bridge. v, 41

UGES Unité de Gestion des Entrées/Sorties. ii, v, 53, 57–59, 81, 83, 123, 125, 128

UL Unité Logicielle. ii, v, 53, 54, 57–59, 81, 83, 91, 103, 123, 125, 128

VHDL Very High Speed Integrated Circuits Hardware Description Langage. v, 42

VIL Validation par Interface Logicielle. v–viii, 54, 57–65, 82–87, 91, 92, 95, 102–104, 122–124, 126–128, 133–135

VIM Validation par Interface Matérielle. v, vii, viii, 4, 91–93, 95, 103, 104, 106, 123, 124, 127, 128, 134

VPS Virtual Power system Simulator. v

VTB-RT Real-Time extension of the Virtual Test Bed. v, 30

XSG Xilinx System Generator. v, 107

Liste des symboles

I_{nod} : Vecteur d'injection de courant de la méthode nodale. v, 14

T_s : Pas de temps de calcul. v, 16, 17, 21, 22

T_{EM} : Couple électromagnétique de la machine. v

V_{nod} : Vecteur tension de la méthode nodale. v, 14

Y_{nod} : Matrice admittance de la méthode nodale. v, 14

$\Im(z)$: Partie imaginaire de la variable complexe z. v

$\Re(z)$: Partie réelle de la variable complexe z. v

λ : Valeur propre. v, 21

$\overline{X}$: Vecteur des variables d'état équivalents. v, 12

$\phi_{\overline{A}}(\lambda)$: Equation caractéristique de la matrice dynamique équivalente $\overline{A}$. v, 12

$\phi_A(\lambda)$: Equation caractéristique de la matrice dynamique A. v, 12

$\overline{A}$: Matrice de dynamique équivalente. v, 12

$\overline{B}$: Matrice de commande équivalente. v, 12

$\overline{C}$: Matrice de mesure équivalente. v, 12

$\overline{D}$: Matrice d'action directe équivalente. v, 12

z^{-1} : Opérateur de décalage. v, 16–18, 21, 31, 38

A : Matrice de dynamique. v, 11, 12

B : Matrice de commande ou d'entrée. v, 11

C : Matrice d'observation ou de mesure. v, 11

D : Matrice d'action directe. v, 11, 12

H(s) : Matrice de fonctions de transfert. v, 12

I : Matrice identité. v, 11, 12

P : Matrice de transformation. v, 12

s : Variable de Laplace. v, 11, 16–18, 21, 31, 38

X : Vecteur des variables d'état. v, 11

X(0) : Vecteur d'état initial. v, 12

Y : Vecteur de sorties. v, 11

Table des figures

0.1 Domaine d'activités de General Electric - Power Conversion © (GE-PC). 1
0.2 Ordre de grandeur des fréquences d'alimentation des machines selon l'application. . 2

1.1 Représentation d'état d'un système. 10
1.2 Circuit avec des sources de courant Vlach et Singhal [1983] 14
1.3 Discrétisation du modèle d'un système 15
1.4 Régions de stabilité de la méthode d'Euleur avant ($\mathcal{C}_1$), d'Euleur arrière ($\mathcal{C}_2$) et trapézoïdale ($\mathcal{P}_{Ox'}$). 21
1.5 Illustration de l'utilité de l'analyse de stabilité numérique. 22
1.6 Représentation discrète des éléments réactifs. 30
1.7 Branche (a) et noeud (b) dans la méthode d'insertion de délai ou Latency Insertion Method (LIM). 32
1.8 Interrupteur de type L/C. 37
1.9 Etude du comportement d'un interrupteur L/C selon la méthode trapézoïdale . . . 38
1.10 Partitionnement par une inductance série : Circuit avec inductance (a) – Détection des sous circuits en série (b) – Circuits partitionnés équivalents (c). 45
1.11 Partitionnement par une capacité parallèle : Circuit avec capacité (a) – Détection des sous circuits en parallèle (b) – Circuits partitionnés équivalents (c). 45

2.1 Système électrotechnique avec son contrôleur industriel. 53
2.2 Validation par Interface Logicielle 54
2.3 Forme d'onde type d'une tension trois niveaux $v_{pu}(\theta)$ pour une Modulation de Largeur d'Impulsion (MLI) précalculée. 54
2.4 Angles de commutation dans le cas d'une MLI précalculée à 5 angles. 55
2.5 Spectre harmonique pour différents indices de modulation. 56
2.6 Formes d'ondes des tensions simples pour une MLI à 5 angles. 56
2.7 Détermination temps réel des instants de commutation T_{Ck}. 57
2.8 Détermination temps réel des instants de commutation T_{Ck} dans le cas d'une MLI à porteuse triangulaire. 57
2.9 Interpolation évènementielle pour la VIL. 60
2.10 Détection de l'annulation d'un courant (a) - Pas intermédiaire jusqu'au temps d'annulation calculé (b) - Correction des conditions initiales et poursuite du calcul jusqu'à la fin du pas de simulation (c) 60
2.11 Limitation des pas internes à T_{smax}. 62
2.12 Mise en œuvre de la VIL. 63
2.13 Utilisation d'une fonction Matlab embarquée. 65
2.14 Système de puissance simulé dans le cadre d'une Validation par Interface Logicielle (VIL). 65
2.15 Bras de la phase U d'un convertisseur de type Neutral Point Piloted converter (NPP) - i(t) : Courant de la charge. 66
2.16 Bras de la phase U d'un convertisseur de type NPC. 66

2.17 Détermination de l'annulation du courant par linéarisation. 68
2.18 Composition de variables d'état complexes. 70
2.19 Composition d'une entrée tension complexe V_{DQS}. 72
2.20 Parties réelle et imaginaire de λ_0. 75
2.21 Parties réelle et imaginaire de λ_1. 75
2.22 Comparaison des éléments de la matrice discrète pour un pas de calcul $T_s = 20ms$. . 79
2.23 Comparaison des éléments de la matrice discrète pour un pas de calcul $T_s = 0.2ms$. 80
2.24 Contrôleur testé en VIL. 81
2.25 Mise en oeuvre de la VIL d'un contrôleur industriel pour l'entraînement à vitesse variable d'une machine asynchrone. 82
2.26 De haut en bas : Courants statoriques en simulation à pas variable (continu) i_{USC}, i_{VSC}, i_{WSC} – Courants statoriques du modèle construit avec une fonction matlab (vu du contrôleur en simulation) – i_{USE}, i_{VSE}, i_{WSE} – Tensions de référence en simulation v_{USE}, v_{VSE}, v_{WSE} – Conditions de fonctionnement : Vitesse de référence = 0.35 pu - Flux rotorique de référence = 0.65 pu 84
2.27 Courants statoriques en VIL i_{US-SIL}, i_{VS-SIL}, i_{WS-SIL} – Tensions de référence en VIL v_{US-SIL}, v_{VS-SIL}, v_{WS-SIL} – Conditions de fonctionnement : Vitesse de référence = 0.35 pu - Flux rotorique de référence = 0.65 pu 85
2.28 En haut : spectre du courant i_{USE} de la phase U vu du contrôleur en simulation – spectre du courant i_{US-SIL} de la phase U vu du contrôleur en VIL
En bas : spectre de la tension statorique v_{USE} de la phase U en simulation – spectre de la tension statorique v_{US-SIL} de la phase U en en VIL 85

3.1 Validation par Interface Matérielle (VIM) réalisée sur une unique cible Field Programmable Gate Array (FPGA). 93
3.2 Système de puissance simulé dans le cadre d'une VIM. 93
3.3 Bras d'un convertisseur de type NPP - $[G_1G_2G_3G_4]$: Combinaison des Signaux de Gâchette (CSG) - i(t) : Courant de la charge. 94
3.4 Illustration fonctionnelle du modèle d'un bras du convertisseur implémenté sur FPGA $v_{RO}(t)$: Tension du bras - i_{pmn} : [i_p i_m i_n] Courants du bus continu - $V_{DC} = [V_{PO}, V_{MO}]$. 95
3.5 Redresseur avec neutre non connecté. 98
3.6 Redresseur avec neutre impédant connecté. 98
3.7 Machine d'états modélisant le fonctionnement d'une diode. 98
3.8 Machine d'états pour la gestion des différents modes de fonctionnement. 99
3.9 Discrétisation du modèle de la machine asynchrone pour une implémentation sur FPGA. 100
3.10 Implémentation sur FPGA de l'ensemble convertisseur/machine asynchrone. 101
3.11 Exemple de demi-circuit pour le calcul de i_{DS} et i_{QS}. 102
3.12 Validation du modèle développé sur FPGA en VIM par comparaison avec le modèle de l'ensemble convertisseur/machine simulé en VIL (référence). 103
3.13 Courants des phases U et V du moteur en VIL et en VIM calculés sur FPGA. 104
3.14 Cas d'étude pour la validation temps réel sur FPGA/processeur rapide. 105
3.15 Plateforme de validation de contrôleurs industriels incluant deux tours contrôle configurables et le simulateur temps réel. 105
3.16 Bras d'un convertisseur de type NPC. 106
3.17 Principe de la méthode utilisant les pulses moyennées. 108
3.18 Séquence de commande de l'onduleur trois niveaux de type NPC avec les configurations autorisées (table 2.1 , chapitre 2). 109
3.19 Groupe de fenêtres I. 109
3.20 Groupe de fenêtres II. 110
3.21 Groupe de fenêtres III. 110

3.22 Groupe de fenêtres IV. 111
3.23 Groupe de fenêtres V. 111
3.24 Groupe de fenêtres VI. 111
3.25 Système de puissance utilisé pour valider le modèle du convertisseur trois niveaux NPC à impulsion de commande moyennée. 113
3.26 Tensions entre phases du convertisseur en mode normal. 114
3.27 Courants de ligne en mode normal. 114
3.28 Tensions entre phases du convertisseur en mode haute impédance. 115
3.29 Courants de ligne en mode haute impédance. 115
3.30 Tensions entre phases du convertisseur en mode redresseur. 116
3.31 Courants de ligne en mode redresseur. 116
3.32 Différence entre le modèle proposé et le modèle continu sous PLECS. 117
3.33 Machine asynchrone alimentée par un seul onduleur. 118
3.34 Machine asynchrone alimentée par deux onduleurs. 119
3.35 Schéma de base pour le calcul des courants de ligne des convertisseurs. 119
3.36 Comparaison du modèle proposé pour la machine asynchrone alimentée par un ou deux onduleurs avec le modèle de référence (Piece-wise Linear Electrical Circuit Simulation (PLECS)). 122
3.37 Système de puissance simulé dans le cadre d'une VIL repris pour la VIM sur FPGA/Processeur lent. 123
3.38 VIM par horodatage de la tension sur FPGA. 124
3.39 Circuit d'horodatage des tensions d'un convertisseur trois niveaux. 125
3.40 Validation de l'horodatage par comparaison avec le modèle développé pour la VIL. . 127
3.41 Courants statoriques de la machine en VIL et VIM. 128

D.1 Disposition des enroulements de la machine asynchrone triphasée. 149
D.2 Circuits électriques équivalents de la partie homopolaire 158

Liste des tableaux

1.1 Substitution de l'opérateur de Laplace s en fonction de l'opérateur de décalage z^{-1}. 18
1.2 Représentation discrète des éléments réactifs par une résistance discrète et un courant d'injection. . . . 31
1.3 Expression de la conductance et du courant d'injection d'un interrupteur L/C selon le type d'intégration numérique. . . . 37
1.4 Expression de la conductance et du courant d"injection d'un interrupteur L/RC selon le type d'intégration numérique. . . . 39

2.1 Niveaux de tension de la phase U en fonction du courant et des signaux de gâchette 67
2.2 Exemple de paramètres machine 75
2.3 Paramètres de la machine asynchrone. . . . 81

3.1 Comparaison des composants FPGAs EP2C20F484 et EP1S80B956C7. . . . 92
3.2 Table de vérité du signal v_p. . . . 96
3.3 Expressions logiques des signaux v_p, v_m, v_o et S_c dans le cas d'un onduleur trois niveaux NPP. . . . 96
3.4 Taux d'utilisation des ressources logiques. . . . 104
3.5 Expressions des tensions moyennes pour chaque groupe. . . . 112
3.6 Paramètres du système étudié. . . . 121
3.7 Composition des mots de 32 bits d'horodatage. . . . 125
3.8 Taux d'utilisation des ressources logiques. . . . 126

Introduction Générale

Ce travail a été réalisé dans le cadre d'une Convention Industrielle de Formation par la Recherche (CIFRE) impliquant le Groupe de Recherche en Électrotechnique et Électronique de Nancy (GREEN) et le département Nanomatériaux, Electronique Et Vivant (N2EV) de l'Institut Jean Lamour de Nancy (IJL) (anciennement appelé Laboratoire d'Instrumentation Électronique de Nancy (LIEN)) avec l'entreprise General Electric - Power Conversion © (GE-PC). GE-PC travaille essentiellement dans le domaine de l'entraînement à vitesse variable (marine, production des hydrocarbures, métallurgie), dans les énergies renouvelables (éoliennes, panneaux photovoltaïques) et dans la qualité de l'énergie électrique (compensation statique d'énergie réactive, filtrage actif ...). Ces différents thèmes constituent également des centres d'intérêts des deux laboratoires associés à ce travail de recherche.

Marine

Secteur pétro-gazier

Énergies renouvelables

Métallurgie

FIGURE 0.1 – Domaine d'activités de GE-PC.

Le matériel d'exploitation est énormément coûteux et un arrêt de la production engendre des pertes importantes. Par exemple, un Floating Production, Storage and Offloading (FPSO)

produit des centaines de milliers de barils par jour. Les laminoirs utilisés dans le secteur de la métallurgie produisent plusieurs centaines de tonnes de métal par jour. Il est donc impératif d'avoir un système de validation des contrôleurs industriels impliqués dans ces applications pour éviter d'endommager le matériel à la mise en service et diminuer la durée des interventions et donc des arrêts de production.

Si l'on prend le cas de l'entraînement à vitesse variable, selon le type d'application, le système d'entraînement peut se trouver dans différents types d'environnement et soumis à différentes contraintes avec différentes spécifications.

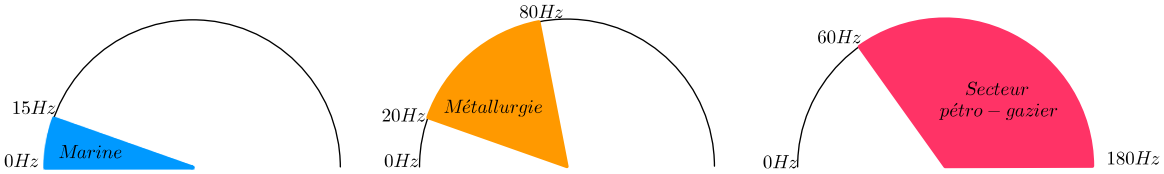


FIGURE 0.2 – Ordre de grandeur des fréquences d'alimentation des machines selon l'application.

Par exemple, les réseaux électriques des bateaux sont des réseaux isolés à faible puissance de court circuit et donc plus vulnérables. En outre, les systèmes de propulsion de bateau sont à faible fréquence (autour d'une quinzaine de hertz) pour éviter les phénomènes de cavitation sur les hélices.

Par contre, les réseaux des secteurs de la métallurgie sont des réseaux plus puissants et sont même soutenus par des systèmes de compensation de réactif, voire même de filtrage harmonique pour supporter des applications types fours à arc. Les fréquences statoriques mises en jeu dans des applications de la métallurgie sont généralement de l'ordre d'une cinquantaine de hertz.

Les systèmes d'entraînement dans la production de gaz et d'hydrocarbure tiennent les records en vitesse. Les machines peuvent avoir des fréquences de rotation avoisinant les $200Hz$, raison pour laquelle les rotors sont supportés par des paliers magnétiques.

Pour répondre à ces différents types d'application, plusieurs architectures du système de puissance existent. Par exemple, la partie redressement peut-être faite par des redresseurs propres (commandés) ou par plusieurs ponts redresseurs à diodes. Par ailleurs, le type de filtre et les stratégies de Modulation de Largeur d'Impulsion (MLI) sont définis en fonction de la gamme de vitesse.

Tous les jours, des améliorations et des changements sont proposés par des ingénieurs et des chercheurs. Les technologies utilisées évoluent en permanence et les solutions vendues ne doivent pas se baser sur des technologies obsolètes. Au delà de ces différences entre types d'applications et de ces perpétuelles évolutions se rajoutent les spécifications propres à chaque projet.
Le système d'entraînement est accouplé à un système mécanique. Tout système mécanique a ses fréquences propres de résonance qu'il ne faut pas entretenir par le contrôle. Il arrive que ces résonances tombent en plein milieu de la bande passante du contrôleur. Par ailleurs les rampes d'accélération et de décélération sont spécifiques au système mécanique. Ces rampes imposent donc la disponibilité d'un couple d'accélération sur l'arbre, fonction de l'inertie totale du système.
Certains clients spécifient des points de fonctionnement assez particuliers. On peut citer les fonctionnements avec des creux de tension allant jusqu'à 70% de la tension nominale du réseau disponible temporairement ou sur une longue durée pour des réseaux peu étendus. Ceci implique l'implémentation de fonctions spécifiques dans le contrôleur qui limiteraient les contraintes sur le système de puissance pour pouvoir tenir ces points de fonctionnement particuliers. Ces fonctions spécifiques sont critiques et sont très difficiles voire impossible à tester sur site.

Un contrôleur industriel doit couvrir une large gamme d'application ainsi que des spécificités propres à chaque projet. En absence de système de prévalidation, chaque modification (matérielle ou logicielle) est susceptible d'endommager le système de puissance. Une modification du contrôleur peut également interrompre la production pour une durée inacceptable dans le cas de travaux de réfection.

Les simulateurs analogiques connus sous le nom de Transient Network Analysers (TNAs) ont été largement utilisés depuis les années trente Thompson [1992] pour vérifier la stabilité du système ainsi que le comportement du contrôleur. Les simulateurs analogiques offrent certes une très bonne résolution mais sont difficilement reconfigurable et plus complexes à paramétrer par rapport à des simulateurs numériques.

Une autre solution pour valider en amont un contrôleur industriel est le recours à une maquette homothétique.
Les modèles homothétiques ne conservent pas toujours les propriétés du système original. Par exemple, sur des petites machines électriques, le rapport entre l'entrefer et le diamètre est beaucoup plus important que sur des grosses machines. Cela se manifeste par des fuites plus importantes.
Par ailleurs, les tests récurrents sur maquette accélèrent le vieillissement et l'usure de cette dernière.
Cependant, une expérimentation sur maquette valide l'aspect fonctionnel du contrôleur. Néanmoins, le paramétrage du contrôleur ne peut se faire que sur le système réel ou sur une plateforme pouvant reproduire le comportement du système réel. De nos jours, une plateforme de simulation temps réel digitale peut permettre de réaliser ce challenge.

La validation des contrôleurs industriels sur simulateur temps réel permet :
- d'éviter ou de limiter les surprises sur site ;
- de faire des tests qui peuvent arriver dans la réalité mais qui restent difficiles à reproduire volontairement ;
- de faire des tests intermédiaires impossibles à réaliser dans la réalité (exemple onduleur avec bus continu parfait) avant les tests d'ensemble ;
- de faire des tests d'acceptation avec le client. Dans certains cas, le client vient avec une réplique portative de son matériel dédiée à l'interface avec le système d'entraînement. Il vérifie donc comme sur un système réel les ordres de démarrage et d'arrêt, donne les consignes de référence et vérifie les échelles des signaux physiques qui lui auraient été remis sur site en temps réel. Des tests à des points de fonctionnement établis durant la phase de dimensionnement se mettent en place et on vérifie en temps réel si le contrôleur et le dimensionnement sont corrects ;
- de revérifier le dimensionnement. La phase de dimensionnement est souvent faite sur des simulations à fenêtre de temps réduite. La simulation temps réel fournit des résultats instantanés et permet de voir rapidement et globalement [1] les contraintes sur chaque élément du système de puissance.

En plus de ces avantages par rapport à des maquettes, une plateforme numérique :
- est flexible et peut simuler un système comme un autre par changement de paramètres et de modèles ;
- occupe moins d'espace et dissipe moins de chaleur qu'une maquette ;
- réduit les risques et accidents comme les signaux échangés sont à majorité en Basse Tension (BT) voire à Très Basse Tension de Sécurité (TBTS)

Les simulateurs numériques peuvent donc servir de plateforme de prévalidation de contrôleurs avec les paramètres du système réel mais également de formation pour les metteurs en

1. Les contraintes sur les composants de commutation ne sont pas toujours simulées car elles nécessitent une modélisation fine et un pas de calcul infiniment petit. On peut cependant trouver dans la littérature certains auteurs qui s'intéressent à ces phénomènes pour des applications de simulation temps réel

service. Un passage par une Validation par Interface Matérielle (VIM) permet de réduire la durée des mises en service et donc des coûts. Notons que les coûts de mise en service peuvent être très importants par exemple pour les plateformes de forages offshore ou marine.

Les premiers articles parlant de simulation temps réel ont été publiés dans les années cinquante Everett [1951] Gray *et al.* [1953] Rubinoff [1955] Thiberville [1958] Bauer [1958]. Ces travaux touchent surtout les domaines de l'aviation et de la mécanique. Ils concernent des domaines plus ou moins complexes allant jusqu'au système de guidage de missile. La simulation temps réel des dispositifs d'électrotechniques est apparue beaucoup plus tard, à cause de la puissance de calcul requise pour prendre en compte les commutations d'interrupteurs commandés.

Pour bien comprendre la notion de simulation temps réel, nous reprenons quelques définitions de la simulation temps réel :

Crosbie *et al.* [2004] : "*Real-time simulation is a familiar technique for testing hardware and software in the loop and for operator training. An important parameter of these simulations is the frame-time necessary to capture the dynamics of the system being simulated. Modern power electronic systems, using higher frequency pulse-width modulation (PWM) converter control demand frame times that are significantly shorter than those found in most real-time simulators.*"

Kim *et al.* [1996] : "*Real-time simulation is an advanced mode of simulation in which the simulation objects are designed to show the same timing behavior that the simulation targets do.*"

Uchida et McPhee [2011] : "*Real-time simulation is an essential component of hardware and operator-in-the-loop applications, such as driving simulators, and can greatly facilitate the design, implementation, and testing of dynamic controllers.*"

La simulation temps réel est donc une technique permettant d'effectuer une validation du matériel et/ou du logiciel sur un modèle numérique exécuté par un calculateur. Dans notre cas, nous nous intéressons à la validation du contrôleur physique et de la partie logicielle. Le temps de calcul et le temps de mise à disposition des signaux doivent être strictement inférieurs au pas de calcul utilisé pour que le simulateur puisse refléter le système original.

Les simulateurs numériques offrent beaucoup d'avantages mais leur mise en place requiert un grand effort de modélisation d'où la problématique de cette thèse qui s'articule autour des méthodologies de modélisation de systèmes électrotechniques et d'implémentation de modèles temps réel sur des calculateurs numériques. Les techniques utilisées diffèrent selon que l'on implémente sur des cibles type processeur ou sur des cibles Field Programmable Gate Arrays (FPGAs), mais également selon le niveau de validation demandé.

La première partie de cette thèse expose d'une façon générale les bases nécessaires pour le développement de modèles numériques des parties linéaires ainsi que des parties incluant des interrupteurs commandés.

La deuxième partie de cette thèse présente une méthode de validation de l'algorithme de contrôle avec les paramètres du système réel et sans ressource matérielle supplémentaire (hormis les ressources matérielles dédiées au contrôle). Dans ce cas, le niveau de validation algorithmique ne prend pas en compte l'interface d'entrée/sortie du contrôleur.

La troisième partie présente une validation complète incluant l'interface d'entrée/sortie. Elle propose deux types de simulateurs : le simulateur à base de processeur et de FPGA et le simulateur uniquement implanté sur cible FPGA.

Chapitre 1

Simulation des Systèmes Électrotechniques

1.1 Introduction

Les premiers simulateurs permettant de valider des contrôleurs étaient des maquettes analogiques (en anglais les TNAs). Ensuite sont venus les simulateurs hybrides (Analogique/Numérique) pour lesquels des fonctions spéciales étaient réalisées en analogique à base d'amplificateurs opérationnels.

D'autres solutions de validation de contrôleurs se basaient sur le développement de modèles homothétiques. Le développement de maquettes homothétiques, dans le cas d'une machine par exemple, nécessite des attentions particulières, Allegre *et al.* [2010]. Les limitations liées à la mise à l'échelle justifient le développement des simulateurs complètement numériques, Tao *et al.* [2008].

L'avènement des énergies renouvelables a en quelques sortes encouragé la validation de contrôleur Temps réel sur simulateurs numériques. La difficulté d'accès à des parcs éoliens ou des parcs photovoltaïques dans les premières phase de développement est évidente. La solution fournie par les simulateurs numériques permet de faire en laboratoire des tests de contrôleurs à l'aide de modèles Temps réel de tels systèmes.

Dans ce chapitre, nous décrivons brièvement les différentes notions utiles et indispensables à la simulation numérique d'un système électrotechnique comportant des convertisseurs statiques.

On définit par système de puissance électrotechnique la partie matérielle dédiée à la conversion d'énergie sans le contrôleur. Ce système peut contenir des convertisseurs statiques, des machines tournantes, un ensemble d'éléments passifs (un réseau électrique, des filtres passifs ...).

La finalité de cette étude est de donner une solution pour pouvoir valider des contrôleurs industriels sans avoir à disposition [1] le système de puissance. Le contrôleur peut être validé en deux étapes avant d'être livré au client :

- une *validation temps réel par interfaçage logiciel* (en anglais Software In the Loop (SIL)) ;
- une *validation temps réel par interfaçage matériel* (en anglais Hardware In the Loop (HIL)).

Dans les travaux présentés dans cette thèse, nous ne nous intéressons pas à la validation du système de puissance lui même. La validation du système de puissance est une étude faite dans ce que nous appelons *validation par interfaçage mixte* (en anglais Power Hardware In the Loop (PHIL)) . Ce type de validation est composé :

1. pour des raisons de coût, d'encombrement et de limitation des modèles réduits homothétiques

- d'une partie logicielle qui calcule et donne les contraintes (courants, tensions, couples de charge ...) à appliquer sur le système de puissance ;
- d'une partie matérielle qui amplifie les références des contraintes. La partie matérielle applique ces contraintes sur le système de puissance.

Les applications PHIL, par exemple dans le secteur de l'automobile, sont utilisées pour tester la partie mécanique avec un profil de couple ou de charge donné.

Les difficultés à surmonter dans le cadre d'une validation Temps réel par interface logicielle ou matérielle sont alors :

- la présence de convertisseurs statiques de part les phénomènes non linéaires qu'ils introduisent lors des commutations des interrupteurs ;
- la complexité du modèle qui augmente avec la taille du système ;
- les limitations en puissance de calcul et en coût.

Une issue aux problèmes liés aux convertisseurs statiques est d'insérer physiquement ces derniers dans la boucle de simulation Temps réel. Hormis les convertisseurs, les autres parties du système de puissance sont simulées. Par exemple, Ding *et al.* [2008] utilisent un filtre actif réel dans la boucle de simulation Temps réel.
Les problèmes rencontrés par la présence d'un convertisseur physique dans la simulation sont les mêmes que les problèmes d'interfaçage d'un simulateur numérique avec un simulateur analogique. En effet, l'interfaçage insère un délai supplémentaire dans la boucle. Ce délai additionnel est susceptible de conduire à des résultats erronés voire même à des phénomènes d'instabilité factices du contrôleur. Le système hybride (analogique/numérique) nécessite une bonne amplification et une synchronisation.
Dans des applications où il est seulement nécessaire de valider l'algorithme de contrôle sans l'interface de génération des signaux de gâchette, un étage d'amplification est utilisé à la place du convertisseur. Par exemple dans les travaux de Courtecuisse *et al.* [2008], un amplificateur reçoit les références de courant et injecte ces courants dans le réseau. Il est clair que ce type d'approche ne valide pas complètement le contrôleur final qui sera implanté sur site. Les différentes sections de ce chapitre décrivent les étapes à suivre pour développer un simulateur complètement numérique, allant de la modélisation du système de puissance incluant des convertisseurs statiques jusqu'à l'implémentation numérique sur une base matérielle donnée. Les bases de modélisation sont les mêmes que celles des simulateurs hors Temps réel.

La première section traite des différentes méthodes de modélisations du système dans le domaine continu.
La section suivante décrit les méthodes de discrétisation. Ces méthodes de discrétisation sont utiles dans le cas d'une "pré-modélisation" continue et permettent de déterminer, à partir de modèles continus, les suites numériques récurrentes à implémenter sur les calculateurs numériques.
La troisième section décrit brièvement les méthodes de modélisation discrètes.
La quatrième section traite ensuite des alternatives pour la simulation Temps réel des convertisseurs statiques.
La section cinq présente et compare les deux grandes familles de bases matérielles (FPGA/Processeurs) utilisées dans la simulation Temps réel tout en justifiant le choix des approches adéquates pour chaque famille.
La dernière section traite des différents axes d'optimisation et de réduction de coût possibles dans le cadre de la simulation Temps réel.

1.2 Modèles à temps continu pour la simulation des systèmes électrotechniques

La dynamique d'un système quelconque est régie par des équations linéaires ou non linéaires. Nous nous intéressons dans cette partie au cas d'un Système Linéaire Invariant dans le Temps (SLIT) [2].

Dans le cadre de la validation d'un contrôleur sur simulateur Temps réel par interface logicielle ou matérielle, le modèle du système doit être :

- suffisamment représentatif du système à simuler (capable de représenter les phénomènes liés au contrôleur. A travers les signaux de retour, ce dernier doit être informé en temps réel de l'impact de ses signaux de commande sur le système électrotechnique.) ;
- aussi simple que possible (pour la rapidité d'exécution en Temps réel [3], et pour la détermination des paramètres [4]) ;
- à la portée des outils d'analyses mathématiques.

Les principaux types de représentations continues présentés dans la littérature sont :

- la représentation par équation différentielle ;
- la représentation d'état ;
- la représentation modale ;
- la représentation nodale.

1.2.1 Représentation par les équations différentielles

Une des formes appréciées des mathématiciens pour représenter un système est la forme dite "en équation différentielle". Prenons l'exemple d'un système monovariable [5] pour illustrer ce propos. La dynamique peut être décrite par une équation différentielle du type :

$$\begin{aligned} \frac{d^n y(t)}{dt^n} + a_{n-1}.\frac{d^{n-1} y(t)}{dt^{n-1}} + ... + a_1.\frac{dy(t)}{dt} + a_0.y(t) \\ = b_{n-1}.\frac{d^{n-1} u(t)}{dt^{n-1}} + ... + b_1.\frac{du(t)}{dt} + b_0.u(t) \end{aligned} \tag{1.1}$$

où :
a_i et $b_j(\forall(i,j) \mid 0 \leq i \leq n-1 \; et \; 0 \leq j \leq n-1)$ sont des coefficients constants, fonctions des caractéristiques du système,
avec $(j_{max} \leq n-1)$ pour des raisons de causalité [6].

Ce type d'équation est une équation différentielle d'ordre n et se résout en deux étapes :

---

2. Un système est dit :
- linéaire si il répond aux critères d'additivité ($f(x_1+x_2) = f(x_1) + f(x_2)$) et d'homogénéité($f(\alpha.x) = \alpha.f(x)$)
- invariant dans le temps si il répond à tout instant de la même façon pour les mêmes conditions initiales et les mêmes entrées. Ces systèmes ont des caractéristiques (résistances, inductances, capacités, poids, inerties...) indépendantes du temps.

3. Par exemple, dans le cas d'une validation pure du contrôleur, les transitoires de commutation ne sont pas simulés car ils nécessitent une très haute résolution mais apportent peu en ce qui concerne la validation du contrôleur. Dans le cas de nos systèmes, ces phénomènes ont une durée de l'ordre de la microseconde alors que le pas de calcul du contrôleur dépasse les 300µs. Les phénomènes transitoires de commutation ont néanmoins une grande importance dans l'étude des contraintes et régimes homopolaires.

4. Les tests d'identification et de détermination de paramètres peuvent être coûteux.

5. Une seule entrée et une seule sortie.

6. Un système est causal si la sortie à un instant donné ne dépend pas de l'entrée future. Les systèmes simulés dans le cadre de validation de contrôleurs sont causals. Des fonctions non causales existent cependant dans des cas de traitement en Temps différé.

1. La recherche de la solution sans second membre (dite en régime libre) en utilisant le polynôme caractéristique. Le problème se décline alors en une recherche des racines du polynôme caractéristique d'ordre n ;
2. La recherche de la solution dite en régime forcé qui a la même forme [7] que la contrainte $u(t)$.

La solution générale est alors la somme des deux solutions précédemment décrites.

Cependant, il n'existe pas de solution analytique générale pour les polynômes de degré supérieur ou égal à 5. Les racines d'un polynôme d'ordre 4 se calculent par la méthode de *Ferrari* qui à son tour fait appel à la méthode de *Cadran* (résolution d'un polynôme d'ordre 3).
Notons que l'équation différentielle est en général assez complexe et le sera encore plus dans le cas d'un système multivariable [8]. La représentation d'état offre alors une écriture plus simple que la représentation par équation différentielle.

1.2.2 Représentation d'état

La représentation d'état est une des formes les plus connues pour modéliser un système (Figure 1.1). La représentation d'état présente sous forme matricielle les équations différentielles régissant la dynamique du système.
Cette forme de représentation a été largement utilisée depuis les années soixante grâce à la disponibilité des codes résolvant des équations différentielles d'ordre 1 Vlach et Singhal [1983] . Elle est utilisée par des logiciels comme Piece-wise Linear Electrical Circuit Simulation (PLECS) Alimeling et Hammer [1999], Advanced Continuous Simulation Language (ACSL) Pekarek *et al.* [2004] , et par la toolbox Power System Blockset (PSB) de l'environnement Matlab Le-Huy [2001].

"Nice artwork kiddo. I have a feeling that a great many people will make a living off that third line someday!" (*Out of Control*, IEEE Control Systems Magazine)

FIGURE 1.1 – Représentation d'état d'un système.

7. Si la contrainte est de type échelon, la solution est de type échelon. Si la contrainte est de type harmonique, la solution est de type harmonique, ...
8. à plusieurs entrées et/ou plusieurs sorties

1.2.2.1 Principe

Chaque ligne de la représentation d'état est formée par une équation différentielle élémentaire d'ordre 1. Le principe est de mettre les équations régissant la dynamique du système sous la forme :

$$\dot{X} = \frac{dX}{dt} = A.X + B.U \tag{1.2}$$

$$Y = C.X + D.U \tag{1.3}$$

où
X est le vecteur de variables d'état, de dimension $(n \times 1)$;
Y est le vecteur de sorties, de dimension $(q \times 1)$;
A est la matrice de dynamique, de dimension $(n \times n)$;
B est la matrice de commande ou d'entrée, de dimension $(n \times p)$;
C est la matrice d'observation ou de mesure, de dimension $(q \times n)$;
D est la matrice d'action directe, de dimension $(q \times p)$.

1.2.2.2 Exemple

L'équation (1.1) peut s'écrire sous forme d'état comme suit :

$$\underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix}}_{\dot{X}_{(1.1)}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \vdots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix}}_{A_{(1.1)}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}}_{X_{(1.1)}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}}_{B_{(1.1)}} \underbrace{u}_{U_{(1.1)}} \tag{1.4}$$

$$y = \underbrace{\begin{pmatrix} b_0 & b_1 & b_2 & \dots & b_{n-1} \end{pmatrix}}_{C_{(1.1)}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}}_{X_{(1.1)}} \tag{1.5}$$

et $D_{(1.1)} = 0$

1.2.2.3 Fonction de transfert

En utilisant la transformée de Laplace, les équations d'état (1.2) et (1.3) s'écrivent :

$$s.X(s) - X(0) = A.X(s) + B.U(s) \tag{1.6}$$

$$Y(s)) = C.X(s) + D.U(s) \tag{1.7}$$

d'où :

$$Y(s) = \left(C\left(sI - A\right)^{-1} B + D\right) U(s) + \left(C\left(sI - A\right)^{-1}\right) X(0) \tag{1.8}$$

avec :
I la matrice identité, de même dimension que A ;
s la variable de Laplace ;

X(0) le vecteur d'état initial ;
Pour des conditions initiales nulles ($X(0) = 0$), on a la relation suivante définissant la matrice de fonction de transfert $H(s)$ de dimension (q,n) :

$$Y(s) = H(s)U(s) = \left(C\left(sI - A\right)^{-1} B + D\right) U(s) \tag{1.9}$$

1.2.2.4 Remarque

Le modèle d'état d'un système donné n'est pas unique. La différence entre deux variantes réside dans le choix des variables d'état. Le choix des variables d'état le plus pratique pour représenter un système est celui de la représentation modale détaillée ci-après car elle est faite de plusieurs états indépendants ou, dans certains cas quasi-indépendants.

1.2.3 Représentation modale

Soit P une matrice non singulière (inversible ou régulière).
On définit le vecteur de variable d'état équivalent $\overline{X}$ lié au vecteur d'état X par la relation :

$$\overline{X} = P\ X \tag{1.10}$$

alors :

$$X = P^{-1}\ \overline{X} \tag{1.11}$$

A partir des équations (1.2) et (1.3) , on peut établir :

$$\overline{X} = \underbrace{P\ A\ P^{-1}}_{\overline{A}} \overline{X}\ +\ \underbrace{P\ B}_{\overline{B}}\ U \tag{1.12}$$

$$Y = \underbrace{C P^{-1}}_{\overline{C}} .X + \underbrace{D}_{\overline{D}} .U \tag{1.13}$$

$\overline{X}$ est le vecteur de variables d'état équivalent, de dimension $(n \times 1)$;
$\overline{A}$ est la matrice de dynamique équivalente, de dimension $(n \times n)$;
$\overline{B}$ est la matrice de commande ou d'entrée équivalente, de dimension $(n \times p)$;
$\overline{C}$ est la matrice d'observation ou de mesure équivalente, de dimension $(q \times n)$;
$\overline{D}$=D est la matrice d'action directe, de dimension $(q \times p)$.

Les représentations d'état définies par les relations (1.2) et (1.3) et par les équations (1.12) et (1.13) sont équivalentes. D'ailleurs, elles ont les mêmes équations caractéristiques.
Si I est la matrice identité de même dimension que A et $\overline{A}$, on a :

$$\begin{aligned} \phi_{\overline{A}}(\lambda) = \left|\lambda\ I\ - \overline{A}\right| &= \left|\lambda\ P\ P^{-1}\ - P\ A\ P^{-1}\right| = \left|P\ (\lambda\ I -\ A)\ P^{-1}\right| \\ &= |P|\ |(\lambda\ I -\ A)|\ \left|P^{-1}\right| = |(\lambda\ I -\ A)| = \phi_A(\lambda) \end{aligned} \tag{1.14}$$

$\phi_A(\lambda)$ et $\phi_{\overline{A}}(\lambda)$ sont respectivement l'équation caractéristique de A et de $\overline{A}$.

Les valeurs propres de A et $\overline{A}$ sont les racines de $\phi_A(\lambda)$. La méthode la plus simple pour obtenir ces valeurs propres est de calculer les coefficients de $\phi_{\overline{A}}$ et de chercher ensuite ses racines. Cette méthode n'est valable que pour des matrices de petite taille. Pour des matrices à plus de 10 lignes, elle ne convient plus et donne des résultats erronés. En effet, les formats de nombres en virgule flottante ou en double précision ne suffisent pas pour avoir des résultats pertinents. Pour des grandes matrices, il est nécessaire d'utiliser des algorithmes numériques plus complexes.

Dans le cas où les valeurs propres sont deux à deux distinctes (de multiplicité égale à un), la matrice de dynamique (A) est diagonalisable. Les éléments diagonaux correspondent alors aux valeurs propres et les colonnes de la matrice de transformation P correspondent aux vecteurs propres associés. Ce cas particulier de représentation est appelé représentation modale ou représentation diagonale du fait de l'obtention d'une matrice de dynamique équivalente diagonale.

Dans le cas où au moins une des valeurs propres a une multiplicité supérieure ou égale à deux, on obtient une matrice équivalente quasi-diagonale. La matrice de dynamique équivalente est alors une matrice dont les éléments diagonaux sont constitués des chaînes de Jordan. La détermination des vecteurs propres et la détermination du nombre et de la taille maximale des chaînes de Jordan dépendent alors de la multiplicité algébrique et de la multiplicité géométrique de chaque valeur propre multiple.

La liste de référence suivante répertorie des documents pertinents sur le thème de la forme de Jordan : Kailath [1980] Borne *et al.* [1992] Brogan [1991] Horn et Johnson [1990] Antsaklis et Michel [2005]. La représentation diagonale ou quasi-diagonale est intéressante du point de vue simulation car elle allège énormément les calculs. La manipulation de matrice diagonale est moins coûteuse. Par contre, la détermination des valeurs propres et de la matrice de transformation se complexifie avec la taille de la matrice dynamique.

1.2.4 Représentation nodale

La représentation nodale est utilisée par plusieurs logiciels d'analyse de circuits comme par exemple ElectroMagnetic Transient Program (EMTP) et Simulation Program with Integrated Circuit Emphasis (SPICE). Elle est utilisée pour générer des modèles à partir d'une description topologique (type netlist) du circuit à simuler. La méthode des mailles qui lui est apparentée est moins utilisée en analyse de circuit du fait de la complexité d'automatisation et de sa limitation au circuit planaire.

Pour simplifier l'écriture des équations, on définit l'admittance Y_C d'une capacité C et l'admittance Y_L d'une inductance L comme suit :

$$Y_C = s.C \tag{1.15}$$

$$Y_L = \frac{1}{s.L} \tag{1.16}$$

1.2.4.1 Principes

La méthode détaillée de la représentation nodale, incluant la modélisation des interrupteurs commandés, a été traitée en détails par Vlach et Singhal [1983]. Nous reprenons dans cette partie les grandes lignes de la méthode nodale pour montrer au lecteur la simplicité de cette formulation.

Par convention, les courants entrant auront un signe positif. Pour la mise en équation, il est préférable, mais non obligatoire, de convertir les sources de tension en sources de courant équivalentes. Pour un réseau d'éléments passifs à n nœuds, la représentation nodale est de la forme :

$$\underbrace{\begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{pmatrix}}_{V_{nod}} = \underbrace{\begin{pmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{pmatrix}^{-1}}_{Y_{nod}^{-1}} \underbrace{\begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{pmatrix}}_{I_{nod}} \tag{1.17}$$

V_{nod} est le vecteur tension.
I_{nod} est le vecteur d'injection de courant.
Y_{nod} est la matrice admittance.

L'obtention de la représentation nodale consiste alors à déterminer la matrice admittance et le vecteur d'injection de courant. La mise en équation est basée sur la loi de $Kirchhoff$. La matrice admittance et le vecteur d'injection de courant se construisent méthodiquement comme suit :

1. Définir un nœud de référence et numéroter les nœuds.
2. Les éléments diagonaux de Y_{nod} sont positifs et ses termes diagonaux $Y_{nod,jj}$ sont égaux à la somme des admittances connectées au nœud j .
3. Les éléments extra diagonaux sont négatifs et $Y_{nod,jk}(j \neq k) = -\sum$ admittances connectées entre les nœuds j et k.
4. Chaque élément du vecteur d'injection de courant $I_{nod,j} = \sum$ algébrique des courants indépendants entrants (positifs) et sortant (négatifs) du nœud j.

1.2.4.2 Exemple

Prenons l'exemple de la Figure 1.2.

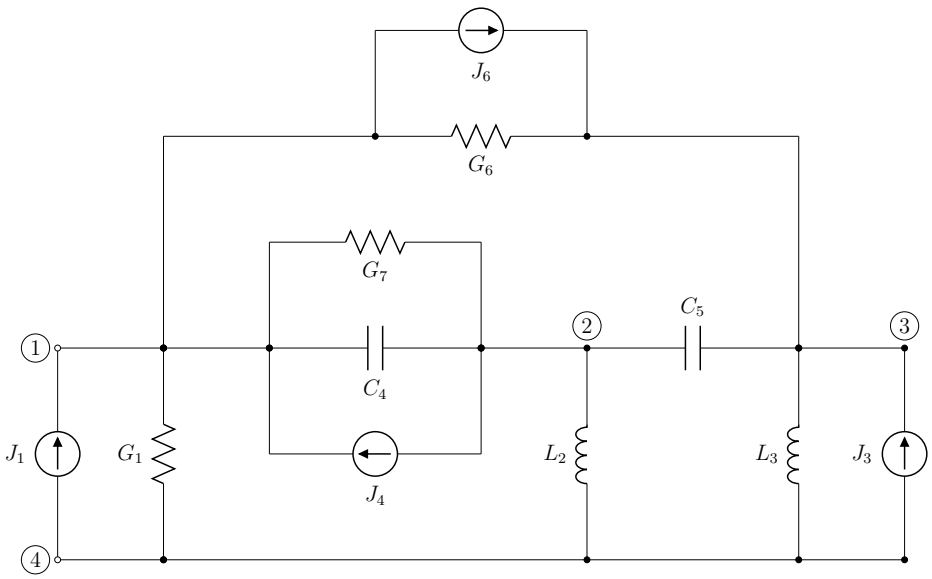


FIGURE 1.2 – Circuit avec des sources de courant Vlach et Singhal [1983]

La représentation nodale correspondante est :

$$\underbrace{\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix}}_{V_{nod}} = \underbrace{\begin{pmatrix} G_1 + s.C_4 + G_6 + G_7 & -s.C_4 - G_7 & -G_6 & -G_1 \\ -s.C_4 - G_7 & \frac{1}{s.L_2} + s.C_4 + s.C_5 + G_7 & -s.C_5 & -\frac{1}{s.L_2} \\ -G_6 & -s.C_5 & \frac{1}{s.L_3} + s.C_5 + G_6 & -\frac{1}{s.L_3} \\ -G_1 & -\frac{1}{s.L_2} & -\frac{1}{s.L_3} & G_1 + \frac{1}{s.L_2} + \frac{1}{s.L_3} \end{pmatrix}^{-1}}_{Y_{nod}^{-1}} \cdot \underbrace{\begin{pmatrix} J_1 + J_4 - J_6 \\ -J_4 \\ J_3 \\ J_3 + J_6 \end{pmatrix}}_{I_{nod}} \tag{1.18}$$

Robbins et College [2003] (Chapitre 8 – Section 6) présentent également cette méthode pour le cas d'un circuit contenant des sources de tension, des sources de courant et des résistances. Un exemple incluant des éléments réactifs (inductance et capacité) a été également traité par Le-Huy [2006].

1.2.4.3 Remarques

Vlach et Opal [1997] qualifient la méthode nodale comme la meilleure méthode de formulation pour les outils de Conception Assistée par Ordinateur (CAO). Elle est adéquate pour les systèmes fortement non linéaires contenant des interrupteurs. Par ailleurs, la formulation nodale fournit un degré correct de parallélisme de calcul.
Pour une application Temps réel, il est préférable d'utiliser directement les versions discrètes des éléments du système de puissance et de résoudre ensuite avec la méthode nodale.
La génération des équations du réseau par la méthode nodale peut être faite automatiquement par un calculateur en utilisant la théorie des graphes Vlach et Singhal [1983] (Chapitre 3).

1.3 Discrétisation

Une fois le système modélisé, il est nécessaire de trouver les expressions numériques à implanter pour simuler le système sur un calculateur numérique. La Figure 1.3 résume symboliquement la procédure de discrétisation d'un système à partir de son modèle continu.

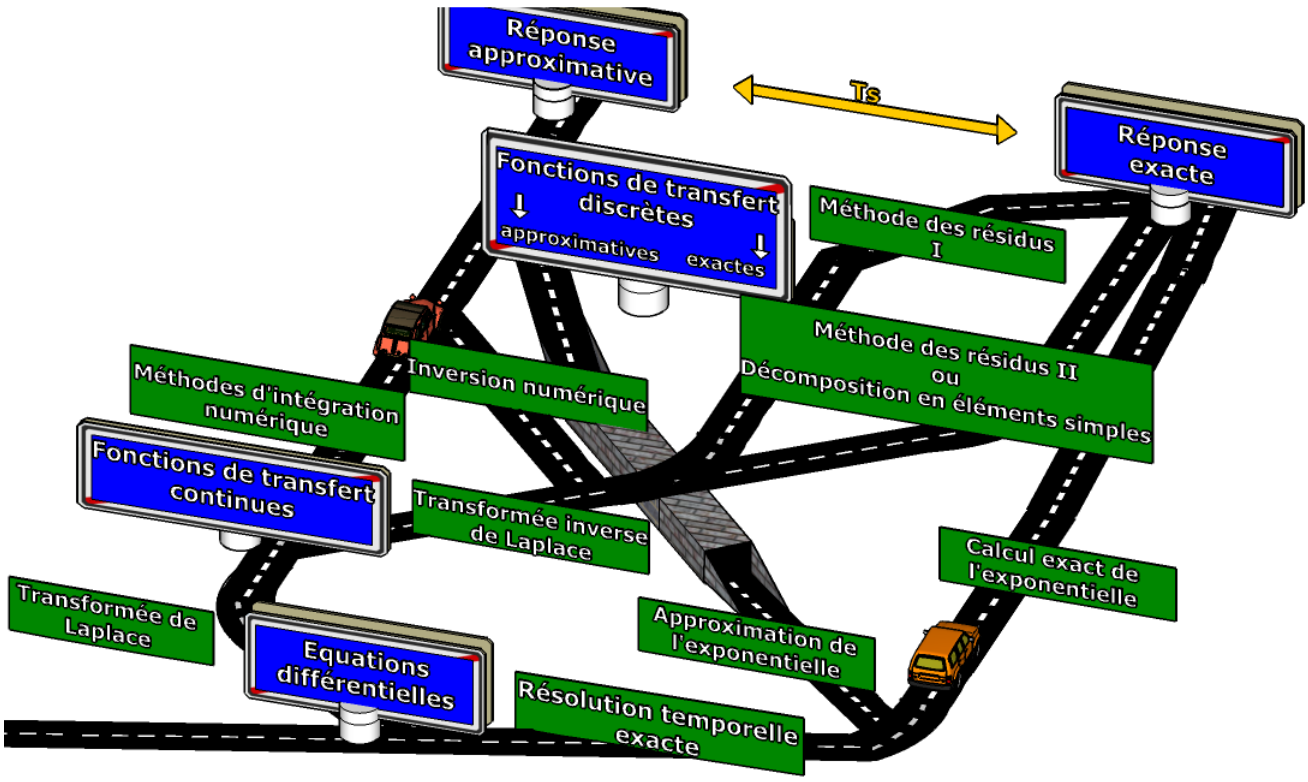


FIGURE 1.3 – Discrétisation du modèle d'un système

1.3.1 Méthodes d'intégration numérique et stabilité

Les méthodes d'intégration numériques sont nécessaires quand il est difficile ou coûteux d'établir l'inverse de la transformée de Laplace pour avoir une expression temporelle exacte (par

la méthode des résidus ou par une décomposition en polynôme simple).

Si la fonction de transfert continue est de type polynomiale, il est possible de calculer directement la forme discrète de cette fonction de transfert par une substitution.

Pour cela, on utilisera la transformée en Z dans laquelle le retard pur est représenté par z^{-1}. Dans la suite de cette section, nous présentons les différentes méthodes d'intégration numérique de base.

1.3.1.1 Méthodes d'intégration du premier ordre

Les méthodes d'intégration d'ordre 1 supposent qu'une fonction $y(t)$ évolue entre deux instants de mesure discrets selon l'équation :

$$\frac{dy}{dt} = k_1 \Leftrightarrow y(t) = k_1 t + k_0 \tag{1.19}$$

A $t = t_n = 0s$ (on prend t_n comme origine de temps) :

$$y(t_n) = y_n \tag{1.20}$$

A l'instant $t = t_n + T_s$, on a :

$$y(t_n + T_s) = y_{n+1} = k_1 T_s + y_n \tag{1.21}$$

où T_s représente le pas de calcul.

1.3.1.1.1 Méthode d'Euler avant

La méthode d'Euler avant utilise une approximation linéaire d'ordre 1. Elle approxime en estimant la dérivée à l'instant précédant, donc :

$$\frac{dy}{dt}|_{(t=t_n)} = \dot{y}_n = k_1 \tag{1.22}$$

Des équations (1.21) et (1.22) on déduit :

$$\frac{dy}{dt}|_{(t=t_n)} = \dot{y}_n = \frac{y_{n+1} - y_n}{T_s} \tag{1.23}$$

$$z^{-1}.s.Y = \frac{Y - z^{-1}.Y}{T_s} \tag{1.24}$$

D'où la relation de s fonction de z^{-1}, issue de la méthode d'Euler arrière :

$$s = \frac{1 - z^{-1}}{T_s z^{-1}} \tag{1.25}$$

Comme la méthode d'Euler avant approxime en estimant la dérivée à l'instant précédent, elle introduit un retard égal à un pas de calcul T_s.

1.3.1.1.2 Méthode d'Euler arrière

La méthode d'Euler arrière est une méthode d'intégration du premier ordre qui approxime en estimant la dérivée à l'instant du calcul (contrairement à la méthode d'Euler avant qui la calcule sur l'instant précédent), donc :

$$\frac{dy}{dt}|_{(t=t_n+T_s)} = \dot{y}_{n+1} = k_1 \tag{1.26}$$

Des équations (1.21) et (1.26) on déduit :

$$\frac{dy}{dt}|_{(t=t_n+T_s)} = \dot{y}_{n+1} = \frac{y_{n+1} - y_n}{T_s} \tag{1.27}$$

$$s.Y = \frac{Y - z^{-1}.Y}{T_s} \tag{1.28}$$

D'où la relation de s fonction de z^{-1} par la méthode d'Euler arrière :

$$s = \frac{1 - z^{-1}}{T_s} \tag{1.29}$$

Contrairement à la méthode d'Euler avant, la méthode d'Euler arrière introduit une avance égale au pas de calcul T_s.

1.3.1.1.3 Méthode trapézoïdale

La méthode trapézoïdale approxime en estimant la dérivée par la moyenne des valeurs de l'instant du calcul, et de l'instant précédent, soit :

$$\frac{dy}{dt}|_{(t=t_n+T_s)} = \frac{\dot{y}_{n+1} + \dot{y}_n}{2} = k_1 \tag{1.30}$$

Des équations (1.21) et (1.30) ont déduit :

$$\dot{y}_{n+1} + \dot{y}_n = \frac{2}{T_s}(y_{n+1} - y_n) \tag{1.31}$$

$$s.Y + z^{-1}.s.Y = 2.\frac{Y - z^{-1}.Y}{T_s} \tag{1.32}$$

D'où la relation de s fonction de z^{-1} par la méthode trapézoïdale :

$$s = \frac{2}{T_s}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{1.33}$$

La méthode trapézoïdale étant une moyenne de la méthode d'Euler avant et la méthode d'Euler arrière, elle n'introduit pas de délai ni d'avance.

1.3.1.2 Méthode d'intégration d'ordre 2

Les méthodes d'intégration d'ordre 2 utilisent une approximation parabolique. Cela signifie qu'une fonction $y(t)$ évolue entre deux instants de mesure discrets selon l'équation :

$$\frac{d^2y}{dt^2} = k_2 \Leftrightarrow y(t) = \frac{1}{2}k_2t^2 + k_1t + k_0 \tag{1.34}$$

A $t = t_n$:

$$y(t_n) = y_n \tag{1.35}$$

A $t = (t_n + T_s)$:

$$y(t_n + T_s) = y_{n+1} = \frac{1}{2}k_2T_s^2 + k_1T_s + y_n \tag{1.36}$$

A $t = (t_n + 2T_s)$:

$$y(t_n + 2T_s) = y_{n+2} = 2k_2T_s^2 + 2k_1T_s + y_n \tag{1.37}$$

1.3.1.2.1 Méthode de Gear d'ordre 2

La méthode de Gear approxime en estimant la dérivée à l'instant de calcul, donc :

$$\frac{dy}{dt}|_{(t=t_n+2T_s)} = \dot{y}_{n+2} = k_2(2T_s) + k_1 \tag{1.38}$$

Pour trouver les coefficients de pondérations, on écrit les équations (1.36) (1.37) et (1.38) sous la forme matricielle :

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{n+2} \\ y_{n+1} \\ y_n \end{pmatrix} = \begin{pmatrix} 2T_s^2 & 2T_s & 0 \\ \frac{T_s^2}{2} & T_s & 0 \\ 2T_s & 1 & -1 \end{pmatrix} \begin{pmatrix} k_2 \\ k_1 \\ \dot{y}_{n+2} \end{pmatrix} \tag{1.39}$$

On en déduit alors que :

$$\begin{pmatrix} k_2 \\ k_1 \\ \dot{y}_{n+2} \end{pmatrix} = \begin{pmatrix} \frac{1}{T_s^2} & \frac{-2}{T_s^2} & \frac{1}{T_s^2} \\ \frac{-1}{2T_s} & \frac{4}{2T_s} & \frac{-3}{2T_s} \\ \frac{3}{2T_s} & \frac{-4}{2T_s} & \frac{1}{2T_s} \end{pmatrix} \begin{pmatrix} y_{n+2} \\ y_{n+1} \\ y_n \end{pmatrix} \tag{1.40}$$

De la troisième ligne de l'équation (1.40), on déduit :

$$\frac{dy}{dt}|_{(t=t_n+2T_s)} = \dot{y}_{n+2} = \frac{3y_{n+2} - 4y_{n+1} + y_n}{2T_s} \tag{1.41}$$

D'où la relation de s en fonction de z^{-1} par la méthode de Gear de second ordre :

$$s = \frac{3 - 4z^{-1} + z^{-2}}{2T_s} \tag{1.42}$$

Le tableau 1.1 résume les correspondances s / z^{-1} pour ces différents méthodes d'intégration.

TABLE 1.1 – Substitution de l'opérateur de Laplace s en fonction de l'opérateur de décalage z^{-1}.

Méthodes d'intégration	**Approximations**
Euler avant [9]	$s = \frac{1-z^{-1}}{T_s.z^{-1}}$
Euler arrière	$s = \frac{1-z^{-1}}{T_s}$
Trapézoïdale	$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}$
Gear du deuxième ordre	$s = \frac{3-4z^{-1}+z^{-2}}{2T_s}$

Lorsque la fonction de transfert continue contient des fonctions non linéaires (comme les fonctions hyperboliques ou exponentielles par exemple), des méthodes numériques d'inversion de la transformée de Laplace Vlach et Singhal [1983](chapitre 10 - page 282) sont utilisées. Il existe beaucoup de méthodes d'intégration numérique. Plus l'ordre est élevé, plus la méthode est précise et plus on peut s'autoriser à utiliser de grands pas de calcul. Cependant, une méthode d'intégration d'ordre élevé nécessite un plus grand nombre de conditions initiales.

1.3.1.3 Méthodes d'intégration explicites - Définition

Une méthode d'intégration est dite Explicite si elle nécessite uniquement les données des pas de calcul précédant le point recherché (Euler avant par exemple). Ce genre d'intégration est utilisé pour prédire les valeurs initiales Le-Huy [2006](page 15) Vlach et Singhal [1983](page 274).

9. même résultat qu'avec le bloqueur d'ordre zéro

1.3.1.4 Méthodes d'intégration implicites - Définition

Une méthode d'intégration est dite Implicite si elle nécessite l'information au point recherché (Méthodes d'Euler arrière et trapézoïdale par exemple). Ce genre d'intégration est utilisé pour les correcteurs, mais également pour la simulation Temps réel Le-Huy [2006](page 15) Vlach et Singhal [1983](page 274).

1.3.1.5 Amortissement numérique

Certaines méthodes d'intégration introduisent artificiellement dans le modèle discret un amortissement qui n'existe pas dans le modèle continu d'origine. L'amortissement numérique est un des inconvénients des méthodes d'intégration très stables telles que la méthode de *Gear*. Cet amortissement numérique tend à amortir un circuit qui, dans la réalité, oscille Schilders et Maten [2005](section 10 - page 567).
Dans certaines applications, la notion d'amortissement numérique est utilisée pour amortir des modes de résonance dont les fréquences ne peuvent pas être prises en compte par le pas de calcul numérique. La contribution de ces modes (bien que ces modes existent dans le système) dans ces cas là est peu importante car on y considère qu'ils ne sont pas excités.

1.3.1.6 Stabilité des méthodes d'intégration

Les erreurs de troncature et l'ordre de la méthode d'intégration ne suffisent pas pour évaluer le comportement d'un système discrétisé par une méthode d'intégration donnée. Il est nécessaire d'étudier la stabilité de la méthode d'intégration pour le système concerné Vlach et Singhal [1983](Chapitre 9 - page 270).

Nous présentons ici à titre d'exemple la stabilité des méthodes d'intégration, précédemment examinées, en étudiant le cas d'un système régi par l'équation différentielle suivante :

$$\dot{x} = \lambda x \tag{1.43}$$

avec $\lambda \in \mathbb{C}$

La solution temporelle exacte est alors :

$$x(t) = x_0 e^{-\lambda t} \tag{1.44}$$

1.3.1.6.1 Stabilité de la méthode d'Euler avant

Pour un pas d'intégration T_s, la solution de l'équation (1.43) par la méthode d'Euler avant s'écrit :

$$x_n = x(n.T_s) = x_0(1 + T_s\lambda)^n \tag{1.45}$$

x_n est une suite géométrique de raison $r = 1 + T_s\lambda$ qui converge si :

$$|1 + T_s\lambda| < 1 \tag{1.46}$$

On pose $q = T_s\lambda$ avec $q \in \mathbb{C}$.
Pour satisfaire l'inégalité (1.46), on opte pour une écriture complexe $\lambda = \Re(\lambda) + j\Im(\lambda)$. Le résultat discrétisé avec la méthode d'Euler avant converge pour tout point q à l'intérieur du cercle $(\mathcal{C}_1)$ de centre(-1,0) et de rayon unitaire.

Pour avoir une version discrète stable par la méthode d'Euler avant d'un système continu d'ordre n stable [10]*, il faut que tous les points* $q_i = \lambda_i T_s$ $(\forall\ 1 \leq i \leq n)$ *soient à l'intérieur du cercle* $(\mathcal{C}_1)$.

10. Un système continu d'ordre n est stable si les parties réelles de toutes ses valeurs propres λ_i sont négatives $(\Re(\lambda_{i=1..n}) < 0)$

1.3.1.6.2 Stabilité de la méthode d'Euler arrière

Pour un intervalle de temps T_s suffisamment petit, la solution de l'équation (1.43) par la méthode d'Euler arrière s'écrit :

$$x_n = x(n.T_s) = x_0 \left(\frac{1}{1 - T_s\lambda}\right)^n \tag{1.47}$$

x_n est une suite géométrique de raison $r = \left(\frac{1}{1-T_s\lambda}\right)$ qui converge si :

$$\left|\left(\frac{1}{1 - T_s\lambda}\right)\right| < 1 \Leftrightarrow |1 - T_s\lambda| > 1 \tag{1.48}$$

On pose $q = T_s\lambda$ avec $q \in \mathbb{C}$.
Pour satisfaire l'inégalité (1.48), comme précédemment on choisit une écriture complexe $\lambda = \Re(\lambda) + j\Im(\lambda)$. Le résultat discrétisé avec la méthode d'Euler arrière converge pour tout point q à l'extérieur du cercle $(\mathcal{C}_2)$ de centre(1,0) et de rayon unitaire.

Pour avoir une version discrète stable par la méthode d'Euler arrière d'un système continu d'ordre n stable[10]*, il faut que tout les points* $q_i = \lambda_i T_s$ $(\forall 1 \leq i \leq n)$ *soient à l'extérieur du cercle* $(\mathcal{C}_2)$.

1.3.1.6.3 Stabilité de la méthode trapézoïdale

Pour un intervalle de temps T_s suffisamment petit, la solution de l'équation (1.43) par la méthode trapézoïdale s'écrit :

$$x_n = x(n.T_s) = x_0 \left(\frac{1 + \frac{T_s\lambda}{2}}{1 - \frac{T_s\lambda}{2}}\right)^n = x_0 \left(\frac{2 + T_s\lambda}{2 - T_s\lambda}\right)^n \tag{1.49}$$

x_n est une suite géométrique de raison $r = \left(\frac{2+T_s\lambda}{2-T_s\lambda}\right)$ et converge si :

$$\left|\frac{2 + T_s\lambda}{2 - T_s\lambda}\right| < 1 \tag{1.50}$$

On pose $q = T_s\lambda$ avec $q \in \mathbb{C}$
Pour satisfaire l'inégalité (1.50), soit :
$\lambda = \Re(\lambda) + j\Im(\lambda)$
$q = \lambda.Ts = Ts.\Re(\lambda) + j.Ts.\Im(\lambda) = \Re(q) + j.\Im(q)$
L'inégalité (1.50) s'écrit alors :

$$0 \leq \frac{\sqrt{(2 + \Re(q))^2 + \Im(q)^2}}{\sqrt{(2 - \Re(q))^2 + \Im(q)^2}} < 1 \tag{1.51}$$

Soit :

$$0 \leq \sqrt{(2 + \Re(q))^2 + \Im(q)^2} < \sqrt{(2 - \Re(q))^2 + \Im(q)^2} \tag{1.52}$$

D'où :

$$(2 + \Re(q))^2 < (2 - \Re(q))^2 \tag{1.53}$$

Soit :

$$\Re(q) < 0 \tag{1.54}$$

Le résultat discrétisé avec la méthode trapézoïdale converge pour tout point q du demi plan gauche $(\mathcal{P}_{Ox'})$.

Un système continu stable[10] *a systématiquement une version discrète stable par la méthode trapézoïdale. En effet,* $\Re(q) < 0$ *tant que* $\Re(\lambda) < 0$ *car* $q = \lambda Ts$ *et* $0 < Ts$.

La Figure 1.4 illustre les régions de stabilité pour les trois cas de méthodes d'intégration présentées.

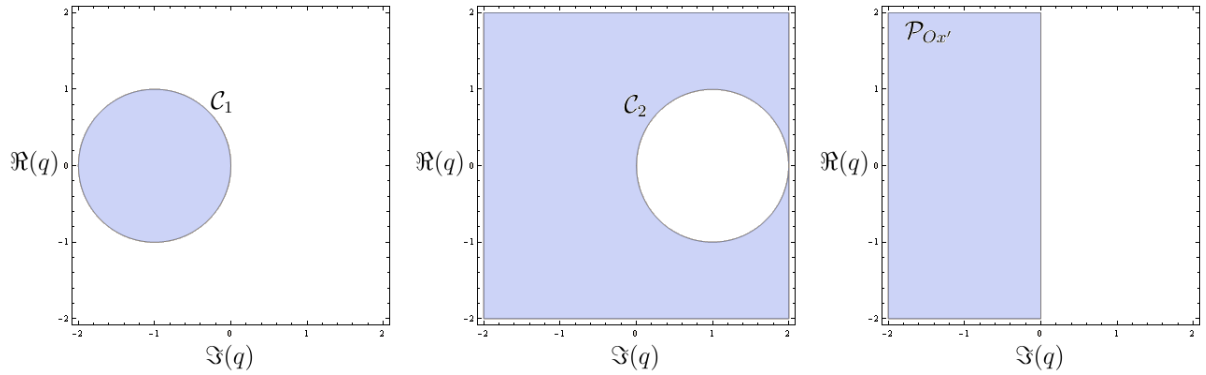


FIGURE 1.4 – Régions de stabilité de la méthode d'Euleur avant ($\mathcal{C}_1$), d'Euleur arrière ($\mathcal{C}_2$) et trapézoïdale ($\mathcal{P}_{Ox'}$).

1.3.1.6.4 Méthodes d'intégration A-stables

Une méthode d'intégration numérique est dite A-stable si elle donne une solution bornée de l'équation (1.43) pour tout $\lambda \mid \Re(\lambda)<0$, pour tout pas de calcul T_s et pour tout nombre d'itérations n donné. La méthode trapézoïdale est une méthode A-stable.

1.3.1.7 Exemple

Prenons l'exemple de la fonction de transfert continue suivante :

$$h(s) = \frac{1}{s-3} \tag{1.55}$$

$h(s)$ possède un pôle $s_0 = 3$ (qui n'est autre que la valeur propre λ=3) à partie réelle positive. La réponse exacte est donc du type exponentiel et diverge pour une contrainte de type échelon. Soit $h(z^{-1})$ la version discrète de h(s) par la méthode d'Euler arrière.

$$h\left(z^{-1}\right) = \frac{T_s}{1-3T_s-z^{-1}} \tag{1.56}$$

La discrétisation avec la méthode d'Euler arrière conduit à une mauvaise réponse indicielle pour T_s=1s (Figure 1.5). La version discrète $h(z^{-1})$ converge car q=T_s.λ=3. En effet q=3 se trouve en dehors du cercle ($\mathcal{C}_2$) de rayon unitaire et de centre (1,0) et la réponse discrète est donc stable.

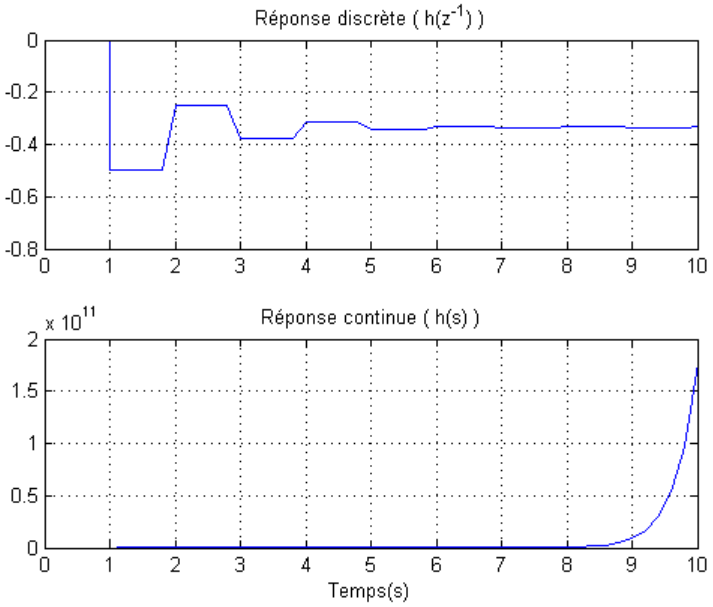


FIGURE 1.5 – Illustration de l'utilité de l'analyse de stabilité numérique.

En choisissant un pas de calcul T_s tel que q se trouve à l'intérieur de $(\mathcal{C}_2)$ ($Ts = 0.25s$ par exemple), on obtient alors une réponse divergente.

1.3.1.8 Discussions

Les méthodes d'intégration d'ordre supérieur à deux ont une région de stabilité restreinte. Elles nécessitent plus de performances de calcul et peuvent avoir des problèmes de conditions initiales.

Par ailleurs, les méthodes Implicites sont beaucoup plus stables que les méthodes Explicites. Selon les deux derniers points précédemment discutés, les méthodes d'intégration adéquates à la simulation Temps réel sont donc la méthode trapézoïdale, la méthode d'*Euler* arrière et la méthode de *Gear* de deuxième ordre Dinavahi [2000].

Faruque et Dinavahi [2010] proposent de choisir une méthode d'intégration numérique en fonction de la nature des entrées (échelon, rampe ...) et de basculer en Temps réel d'une méthode à une autre selon la nature de la variation de l'entrée du système. Dans ce cas, l'erreur est minimisée tant que la méthode d'intégration choisie correspond à chaque instant à la nature de l'entrée.

1.3.2 Méthodes de discrétisation analytiques

Comme décrit précédemment, la discrétisation du modèle d'un système peut se faire assez facilement par des méthodes d'intégration numérique. Cependant, la stabilité numérique n'est pas assurée pour tout pas de calcul donné si l'on utilise des méthodes d'intégration numérique. Les résultats peuvent amener à une version discrète stable alors que le modèle continu original est instable (et inversement).

Dans la partie qui suit, nous décrivions des démarches, qui, si elles aboutissent peuvent garantir un comportement du modèle discret tout à fait identique au modèle continu, indépendamment du pas de calcul utilisé.

Ces méthodes requièrent un effort important de calcul mathématique, et nécessitent donc

davantage de temps. L'aide apportée par des logiciels de calcul symbolique tels que *Mapple* ou *Mathematica* est bénéfique. Toutefois, si les calculs sont trop complexes et que ces méthodes *exactes* n'aboutissent pas, nous proposerons des alternatives pour les finaliser par approximations.

Les méthodes d'analyse mathématiques existantes permettent de rechercher la solution exacte :

- en inversant la transformée de Laplace ;
- en résolvant les équations différentielles préalablement formulées sous la forme d'état.

1.3.2.1 Transformée inverse de Laplace

On peut calculer l'expression temporelle des variables qui nous intéressent en passant par la transformée inverse de Laplace.
On commence alors par les modèles continus donnés par les équations (1.17) ou (1.8). Ces modèles définissent une relation directe entre les sorties et les entrées sans forcément donner des informations sur les états du système. On les met sous la forme générale :

$$Y(s) = H(s)U(s) \tag{1.57}$$

$Y(s)$ est le vecteur des transformées de Laplace des n sorties $y_i(t)$:

$$Y_i(s) = \sum_{j=1}^{k} \mathcal{L}(y_{ij}(t)) = \sum_{j=1}^{k} \int_0^{\infty} y_{ij}(t)e^{-st}dt \tag{1.58}$$

$H(s)$ est la matrice (n, k) des transformées de Laplace des réponses impulsionnelles $h_{ij}(t)$ qui lient les k entrées aux n sorties :

$$H_{ij}(s) = \mathcal{L}(h_{ij}(t)) = \int_0^{\infty} h_{ij}(t)e^{-st}dt \tag{1.59}$$

$U(s)$ est le vecteur des transformées de Laplace des k entrées $u_i(t)$: $U_i(s) = \mathcal{L}(u_i(t)) = \int_0^{\infty} u_i(t)e^{-st}dt$

Les fonctions $Y_i(s)$ sont donc connues une fois que le système est modélisé et que les entrées de commande sont connues. On se propose de retrouver chaque élément $y_i(t)$ pour $Y_i(s)$ connu.

La transformation qui permet de retrouver la fonction originale est la transformation de Mellin-Fourier définie comme suit :

$$y_i(t) = \mathcal{L}^{-1}(Y_i(s)) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} Y_i(s)e^{st}ds \tag{1.60}$$

σ est l'abscisse de convergence. La transformation de Mellin-Fourier est utilisée par les mathématiciens pour définir les tables des transformées inverses de Laplace. Pratiquement, dans la majorité des cas, on procède par les méthodes qui seront décrites dans les sections suivantes.

1.3.2.1.1 Décomposition en éléments simples

La méthode la plus simple d'inversion de la transformée de Laplace est de décomposer la fonction de transfert en éléments simples. La fonction de transfert est alors mise sous forme de somme de fonctions de transfert élémentaires dont les transformées inverses sont connues. On se réfère ensuite à la table des transformées de Laplace et ses inverses pour retrouver la fonction temporelle originale.

1.3.2.1.2 Méthode des résidus I

Une méthode efficace pour calculer la fonction originale est l'utilisation du théorème des résidus qui conduit à l'expression suivante :

$$y_i(t) = \mathcal{L}^{-1}(Y_i(s)) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} Y_i(s)e^{st}ds = \sum \text{Résidus}\left(Y_i(s)e^{st}\right) \tag{1.61}$$

Un rappel illustré par des exemples sur le calcul des résidus est présenté en annexe A.

1.3.2.1.3 Méthode des résidus II

La méthode des résidus I passe directement de la fonction de transfert à la fonction temporelle associée. La méthode ici dite de "résidus II" permet de retrouver la réponse exacte en passant par une version discrète de la fonction de transfert.
Soient :
$Y_{ij}(s)$ la transformée de Laplace de la réponse $y_{ij}(t)$ à l'entrée $u_j(t)$
$H_{ij}(s)$ la transformée de Laplace de la réponse impulsionnelle $h_{ij}(t)$
$U_j(s)$ la transformée de Laplace de l'entrée $u_j(t)$
$Y_{ij}(z)$ la transformée en z de la réponse $y_{ij}(t)$ à l'entrée $u_j(t)$
$H_{ij}(z)$ la transformée en z de la réponse impulsionnelle $h_{ij}(t)$
$U_j(z)$ la transformée en z de l'entrée $u_j(t)$
On écrit :

$$\begin{array}{ccccc} Y_{ij}(z) = H_{ij}(z)U_j(z) & & \text{et} & & Y_{ij}(s) = H_{ij}(s)U_j(s) \\ y_{ij}(t) = \mathcal{Z}^{-1}\left(H_{ij}(z)U_j(z)\right) & & \text{et} & & y_{ij}(t) = \mathcal{L}^{-1}\left(H_{ij}(s)U_j(s)\right) \end{array} \tag{1.62}$$

On se propose alors de déterminer les fonctions de transfert discrètes $H_{ij}(z)$ équivalentes aux fonctions de transfert continues. Pour cela, on suppose que la commande évolue selon une fonction mathématique connue entre deux instants de mesures discrets. La connaissance de la nature de la commande entre deux instants donnés permet de mettre en équation sa transformée de Laplace et sa transformée en z. Nous allons nous limiter à trois types d'entrée en ce qui concerne la méthode des résidus II :

- une entrée impulsion ;
- une entrée échelon ;
- et une entrée rampe.

Réponse impulsionnelle
Dans le cas où l'entrée est de type impulsion, on écrit : $U_j(s) = 1$ et $U_j(z) = 1$
La deuxième ligne de (1.62) nous donne :

$$y_{ij}(t) = \mathcal{Z}^{-1}\left(H_{ij}(z)\right) \text{ et } y_{ij}(t) = \mathcal{L}^{-1}\left(H_{ij}(s)\right) \tag{1.63}$$

D'où la fonction de transfert discrète obtenue par invariance impulsionnelle :

$$H_{ij}(z) = \mathcal{Z}\left(\mathcal{L}^{-1}\left(H_{ij}(s)\right)\right) \tag{1.64}$$

Réponse indicielle
Dans le cas où l'entrée est de type échelon : $U_j(s) = \frac{1}{s}$ et $U_j(z) = \frac{1}{1-z^{-1}}$
La deuxième ligne de (1.62) nous donne :

$$y_{ij}(t) = \mathcal{Z}^{-1}\left(H_{ij}(z)\frac{1}{1-z^{-1}}\right) \text{ et } y_{ij}(t) = \mathcal{L}^{-1}\left(H_{ij}(s)\frac{1}{s}\right) \tag{1.65}$$

D'où la fonction de transfert discrète obtenue par invariance indicielle :

$$H_{ij}(z) = (1-z^{-1})\mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{H_{ij}(s)}{s}\right)\right) \tag{1.66}$$

Réponse à une rampe
Dans le cas où l'entrée est de type rampe : $U_j(s) = \frac{1}{s^2}$ et $U_j(z) = \frac{T_s z^{-1}}{(1-z^{-1})^2}$
La deuxième ligne de (1.62) nous donne :

$$y_{ij}(t) = \mathcal{Z}^{-1}\left(H_{ij}(z)\frac{T_s z^{-1}}{(1-z^{-1})^2}\right) \text{ et } y_{ij}(t) = \mathcal{L}^{-1}\left(H_{ij}(s)\frac{1}{s^2}\right) \tag{1.67}$$

D'où la fonction de transfert discrète obtenue par invariance à une rampe :

$$H_{ij}(z) = \frac{(1-z^{-1})^2}{T_s z^{-1}}\mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{H_{ij}(s)}{s^2}\right)\right) \tag{1.68}$$

Un exemple utilisant cette deuxième méthode est traité dans l'annexe B.

1.3.2.1.4 Inversion numérique de la transformée de Laplace

La décomposition en éléments simples et les méthodes des résidus I et II nécessitent la connaissance des racines du dénominateur de la fonction de transfert à inverser. Quand la détermination des racines du dénominateur est difficile, ou quand le dénominateur n'est pas un polynôme en s et que l'inverse de la fonction de transfert ne peut se calculer analytiquement, on peut procéder à une inversion numérique pour l'inversion de la transformée de Laplace. Cette inversion fait intervenir une approximation de Padé de e^{st} Singhal *et al.* [1975] Vlach et Singhal [1983].

1.3.2.2 Résolution temporelle exacte

1.3.2.2.1 Principe

On se propose de déterminer la solution temporelle exacte de la représentation d'état (équations (1.2) et (1.3)) . Comme dans le cas d'une équation différentielle à variable scalaire, on *suppose* que la solution est de la forme :

$$X(t) = e^{At}V(t) \tag{1.69}$$

donc

$$\frac{dX}{dt} = \dot{X} = Ae^{At}V(t) + e^{At}\frac{dV}{dt} = AX(t) + BU(t) = Ae^{At}V(t) + BU(t)$$

$$Ae^{At}V(t) + e^{At}\frac{dV}{dt} = Ae^{At}V(t) + BU(t)$$

$$e^{At}\frac{dV}{dt} = BU(t)$$

En utilisant le théorème de Moler et Van Loan [11] ((-At) A = A (-A t) car t est un scalaire) :

$$e^{-At}e^{At}\frac{dV}{dt} = \frac{dV}{dt} = e^{-At}BU(t) \tag{1.70}$$

Et par intégration on a :

$$V(t) - V(0) = \int_0^t e^{-At}BU(t)dt \tag{1.71}$$

De (1.69) on déduit que $V(0) = X(0)$ et $V(t) = e^{-At}X(t)$.
(1.71) s'écrit alors :

$$e^{-At}X(t) - X(0) = \int_0^t e^{-At}BU(t)dt \tag{1.72}$$

11. $e^{At}e^{Bt} = e^{(A+B)t}\ \forall t$ si et seulement si $AB = BA$

D'où :

$$X(t) = e^{At}X(0) + e^{At}\int_0^t e^{-At}BU(t)dt \tag{1.73}$$

Des cas d'applications dans le cas d'entrées de type impulsionnelle, indicielle et rampe sont présentés dans l'annexe C

1.3.2.2.2 Calcul de l'exponentielle

Nous avions vu que la résolution temporelle exacte en tenant compte de la nature de l'entrée utilise la fonction exponentielle d'une matrice. L'exponentielle de la matrice peut se calculer en utilisant le théorème de Cayley-Hamilton ou par des approximations itératives.

Calcul de l'exponentielle par le théorème de Cayley-Hamilton Si A est une matrice carrée non singulière de dimension (n,n) et $f(\lambda) = det(\lambda I - A)$ son polynôme caractéristique, alors A annule f.
$f(\lambda)$ est de la forme :

$$f(\lambda) = \lambda^n + \sum_{k=0}^{n-1} c_k \lambda^k$$

D'après le théorème de Cayley-Hamilton,

$$f(A) = A^n + \sum_{k=0}^{n-1} c_k A^k = 0 \tag{1.74}$$

Il s'ensuit que :

$$A^n = -\sum_{k=0}^{n-1} c_k A^k \tag{1.75}$$

$$A^{n+1} = -A\sum_{k=0}^{n-1} c_k A^k = -\sum_{k=0}^{n-1} c_k A^{k+1} = -c_{n-1}A^n - \sum_{k=0}^{n-2} c_k A^{k+1}$$

$$A^{n+1} = \alpha_{n-1}\sum_{k=0}^{n-1} c_k A^k - \sum_{k=0}^{n-2} c_k A^{k+1} \tag{1.76}$$

Les équations (1.75) et (1.76) montrent que :
Toutes les puissances A^k $\forall k \geq n$ de la matrice A s'écrivent en une combinaison linéaire de puissance A^m avec $m \leq n-1$
Par définition :

$$e^{At} = \sum_{k=0}^{\infty} \frac{1}{k!} A^k t^k \tag{1.77}$$

$$e^{At} = \sum_{k=0}^{n-1} \frac{1}{k!} A^k t^k + \sum_{k=n}^{\infty} \frac{1}{k!} A^k t^k$$

$$e^{At} = \sum_{k=0}^{n-1} \frac{1}{k!} A^k t^k + \sum_{k=0}^{n-1} \beta_k A^k t^k$$

e^{At} peut également s'écrire en une combinaison linéaire de puissance A^k avec $k \leq n-1$

e^{At} peut s'écrire sous la forme :

$$e^{At} = \sum_{k=0}^{n-1} \alpha_k \prime A^k t^k = \sum_{k=0}^{n-1} \alpha_k(t) A^k \tag{1.78}$$

Calcul des coefficients $\alpha_k(t)$:

Les coefficients $\alpha_k(t)$ se calculent différemment selon que les valeurs propres sont deux à deux distinctes ou multiples.

Cas où toutes les valeurs propres sont deux à deux distinctes :
Si les valeurs propres sont deux à deux distinctes, on a n équations de la forme :

$$\begin{pmatrix} e^{\lambda_0 t} \\ e^{\lambda_1 t} \\ \vdots \\ e^{\lambda_{n-2} t} \\ e^{\lambda_{n-1} t} \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{n-1} \alpha_k \lambda_0^k \\ \sum_{k=0}^{n-1} \alpha_k \lambda_1^k \\ \vdots \\ \sum_{k=0}^{n-1} \alpha_k \lambda_{n-2}^k \\ \sum_{k=0}^{n-1} \alpha_k \lambda_{n-1}^k \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \lambda_0 & \dots & \lambda_0^{n-2} & \lambda_0^{n-1} \\ 1 & \lambda_1 & \dots & \lambda_1^{n-2} & \lambda_1^{n-1} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & \lambda_{n-2} & \dots & \lambda_{n-2}^{n-2} & \lambda_{n-2}^{n-1} \\ 1 & \lambda_{n-1} & \dots & \lambda_{n-1}^{n-2} & \lambda_{n-1}^{n-1} \end{pmatrix}}_{M_{Vandermonde}} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-2} \\ \alpha_{n-1} \end{pmatrix} \tag{1.79}$$

$M_{Vandermonde}$ est la matrice de $Vandermonde$ et elle est inversible tant que les valeurs propres sont deux à deux distinctes ($\lambda_i \neq \lambda_j$ tant que $i \neq j$) car son déterminant est :

$$det(M_{Vandermonde}) = \prod_{0 \leq i < j \leq n-1} (\lambda_j - \lambda_i) \tag{1.80}$$

Cas où au moins une des valeurs propres a une multiplicité supérieure ou égale à deux :
Dans le cas où au moins une des valeurs propres λ_i est multiple, la matrice de $Vandermonde$ n'est pas inversible. Une ou plusieurs colonnes sont dupliquées et le déterminant s'annule.
Pour avoir n équations indépendantes à n inconnues, on utilise les dérivées de la fonction exponentielle et de la fonction caractéristique en fonction de λ.
Prenons le cas où la dernière valeur propre λ_{n-1} est de multiplicité $m \geq 2$ et toutes les autres valeurs propres sont simples. La procédure à faire pour chaque valeur propre multiple sera la même.
A la place de l'équation (1.79), on a :

$$\begin{pmatrix} \begin{matrix} (n-m) \text{ équations des} \\ (n-m) \text{ valeurs propres} \\ \text{simples} \end{matrix} & \left\{ \begin{matrix} e^{\lambda_0 t} \\ e^{\lambda_1 t} \\ \vdots \\ e^{\lambda_{n-m-2} t} \\ e^{\lambda_{n-m-1} t} \end{matrix} \right. \\ \begin{matrix} m \text{ équations de la} \\ \text{valeur propre} \\ \text{de multiplicité } m \end{matrix} & \left\{ \begin{matrix} e^{\lambda_{n-m} t} \\ \left.\frac{de^{\lambda t}}{d\lambda}\right|_{\lambda=\lambda_{n-m}} = te^{\lambda_{n-m} t} \\ \left.\frac{d^2 e^{\lambda t}}{d\lambda^2}\right|_{\lambda=\lambda_{n-m}} = t^2 e^{\lambda_{n-m} t} \\ \vdots \\ \left.\frac{d^{m-2} e^{\lambda t}}{d\lambda^{m-2}}\right|_{\lambda=\lambda_{n-m}} = t^{m-2} e^{\lambda_{n-m} t} \\ \left.\frac{d^{m-1} e^{\lambda t}}{d\lambda^{m-1}}\right|_{\lambda=\lambda_{n-m}} = t^{m-1} e^{\lambda_{n-m} t} \end{matrix} \right. \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{n-1} \alpha_k \lambda_0^k \\ \sum_{k=0}^{n-1} \alpha_k \lambda_1^k \\ \vdots \\ \sum_{k=0}^{n-1} \alpha_k \lambda_{n-m-2}^k \\ \sum_{k=0}^{n-1} \alpha_k \lambda_{n-m-1}^k \\ \sum_{k=0}^{n-1} \alpha_k \lambda_{n-m}^k \\ \left.\frac{d \sum_{k=0}^{n-1} \alpha_k \lambda^k}{d\lambda}\right|_{\lambda=\lambda_m} = \sum_{k=1}^{n-1} \alpha_k k \lambda^{k-1}\Big|_{\lambda=\lambda_m} = \sum_{k=1}^{n-1} \alpha_k k \lambda_m^{k-1} \\ \left.\frac{d^2 \sum_{k=0}^{n-1} \alpha_k \lambda^k}{d\lambda^2}\right|_{\lambda=\lambda_m} = \sum_{k=2}^{n-1} \alpha_k k(k-1) \lambda^{k-2}\Big|_{\lambda=\lambda_m} = \sum_{k=2}^{n-1} \alpha_k k(k-1) \lambda_m^{k-2} \\ \vdots \\ \left.\frac{d^{m-2} \sum_{k=0}^{n-1} \alpha_k \lambda^k}{d\lambda^{m-2}}\right|_{\lambda=\lambda_m} = \sum_{k=m-2}^{n-1} \alpha_k \frac{k!}{(k-(m-2))!} \lambda_m^{k-(m-2)} \\ \left.\frac{d^{m-1} \sum_{k=0}^{n-1} \alpha_k \lambda^k}{d\lambda^{m-1}}\right|_{\lambda=\lambda_m} = \sum_{k=m-1}^{n-1} \alpha_k \frac{k!}{(k-(m-1))!} \lambda_m^{k-(m-1)} \end{pmatrix} \quad (1.81)$$

$$\begin{pmatrix} e^{\lambda_0 t} \\ e^{\lambda_1 t} \\ \vdots \\ e^{\lambda_{n-m-2} t} \\ e^{\lambda_{n-m-1} t} \\ e^{\lambda_{n-m} t} \\ te^{\lambda_{n-m} t} \\ t^2 e^{\lambda_{n-m} t} \\ \vdots \\ t^{m-1} e^{\lambda_{n-m} t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \lambda_0 & \dots & \lambda_0^{m-2} & \lambda_0^{m-1} & \lambda_0^m & \lambda_0^{m+1} & \lambda_0^{m+2} & \dots & \lambda_0^{n-2} & \lambda_0^{n-1} \\ 1 & \lambda_1 & \dots & \lambda_1^{m-2} & \lambda_1^{m-1} & \lambda_1^m & \lambda_1^{m+1} & \lambda_1^{m+2} & \dots & \lambda_1^{n-2} & \lambda_1^{n-1} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & \lambda_{n-m-2} & \dots & \lambda_{n-m-2}^{m-2} & \lambda_{n-m-2}^{m-1} & \lambda_{n-m-2}^m & \lambda_{n-m-2}^{m+1} & \lambda_{n-m-2}^{m+2} & \dots & \lambda_{n-m-2}^{n-2} & \lambda_{n-m-2}^{n-1} \\ 1 & \lambda_{n-m-1} & \dots & \lambda_{n-m-1}^{m-2} & \lambda_{n-m-1}^{m-1} & \lambda_{n-m-1}^m & \lambda_{n-m-1}^{m+1} & \lambda_{n-m-1}^{m+2} & \dots & \lambda_{n-m-1}^{n-2} & \lambda_{n-m-1}^{n-1} \\ 1 & \lambda_{n-m} & \dots & \lambda_{n-m}^{m-2} & \lambda_{n-m}^{m-1} & \lambda_{n-m}^m & \lambda_{n-m}^{m+1} & \lambda_{n-m}^{m+2} & \dots & \lambda_{n-m}^{n-2} & \lambda_{n-m}^{n-1} \\ 0 & 1 & \dots & \frac{(m-2)!}{(m-3)!}\lambda_{n-m}^{m-3} & \frac{(m-1)!}{(m-2)!}\lambda_{n-m}^{m-2} & \frac{m!}{(m-1)!}\lambda_{n-m}^{m-1} & \frac{(m+1)!}{(m)!}\lambda_{n-m}^{m} & \frac{(m+2)!}{(m+1)!}\lambda_{n-m}^{m+1} & \dots & \frac{(n-2)!}{(n-3)!}\lambda_{n-m}^{n-3} & \frac{(n-1)!}{(n-2)!}\lambda_{n-m}^{n-2} \\ 0 & 0 & \ddots & \frac{(m-2)!}{(m-4)!}\lambda_{n-m}^{m-4} & \frac{(m-1)!}{(m-3)!}\lambda_{n-m}^{m-3} & \frac{(m)!}{(m-2)!}\lambda_{n-m}^{m-2} & \frac{(m+1)!}{(m-1)!}\lambda_{n-m}^{m-1} & \frac{(m+2)!}{m!}\lambda_{n-m}^{m} & \dots & \frac{(n-2)!}{(n-4)!}\lambda_{n-m}^{n-4} & \frac{(n-1)!}{(n-3)!}\lambda_{n-m}^{n-3} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & (m-1)! & m!\lambda_{n-m} & \frac{(m+1)!}{2!}\lambda_{n-m}^2 & \frac{(m+2)!}{3!}\lambda_{n-m}^3 & \dots & \frac{(n-2)!}{(n-m-1)!}\lambda_{n-m}^{n-m-1} & \frac{(n-1)!}{(n-m)!}\lambda_{n-m}^{n-m} \end{pmatrix}}_{M} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-2} \\ \alpha_{m-1} \\ \alpha_m \\ \alpha_{m+1} \\ \alpha_{m+2} \\ \vdots \\ \alpha_{n-2} \\ \alpha_{n-1} \end{pmatrix} \tag{1.82}$$

La procédure est la même dans le cas où plusieurs valeurs propres sont multiples :
- écrire les équations pour toutes les valeurs propres comme si elles étaient toutes simples ;
- écrire les équations des dérivées pour chaque valeur multiple ;
- calculer les coefficients $\alpha_k(t)$ en inversant la matrice obtenue.

Il suffit alors d'inverser la matrice M pour retrouver les coefficients $\alpha_k(t)$ et utiliser l'équation (1.78) de la page 26 pour calculer e^{At}.

Calcul de l'exponentiel par approximation Le calcul des valeurs propres d'une matrice peut se faire analytiquement si son polynôme caractéristique est d'ordre inférieur ou égal à quatre. Dans le cas de polynômes d'ordre supérieur à quatre, l'exponentielle de la matrice peut alors se calculer en utilisant des méthodes numériques pour avoir les valeurs propres et ensuite en utilisant le théorème de Cayley Hamilton, où directement par approximation.
L'exponentielle de la matrice peut donc se faire par approximation ou par des algorithmes récursifs. Nous en citons quelques uns mais la liste n'est pas exhaustive :

- par un développement en série de Taylor $e^{AT_s} \approx \sum_{i=0}^{k} \frac{AT_s}{i!}$. Dans ce cas l'erreur diminue avec k et avec le pas de calcul T_s ;
- en utilisant les approximations de Paddé ;
- par l'interpolation de Lagrange $e^{At} = \sum_{j=0}^{n-1} \prod_{k=1,k\neq j}^{n} \frac{A-\lambda_k I}{\lambda_j-\lambda_k}$.

Nous avions vu jusqu'ici comment modéliser un système électrotechnique en continu et ensuite trouver les versions discrètes finales qui seront implantées sur un calculateur numérique. Dans ce qui suit, nous proposons des méthodes qui utilisent directement une modélisation discrète des éléments du système électrotechnique.

1.4 Modélisations directes à temps discret des systèmes électrotechniques et électriques

1.4.1 Méthode de *Dommel* - Méthode du Circuit Compagnon

La version initiale de cette méthode est une représentation discrète des éléments réactifs par la méthode trapézoïdale Dommel [1969] huei Lu et Meyer [1987]. Les travaux de Dommel arrivent au moment où un grand débat confrontant l'analogique et le numérique prend place au sein de la communauté scientifique. Les travaux de **?** reprennent les bases de Dommel et proposent d'autres représentations discrètes utilisant d'autres méthodes d'intégration numérique pour l'approche dite Resistive Companion Method (RCM). Le simulateur Real-Time extension of the Virtual Test Bed (VTB-RT) de l'Université de la Caroline du Sud utilise cette méthode RCM Lu *et al.* [2007].

1.4.1.1 Principe

La méthode de *Dommel* et la méthode du circuit compagnon dite RCM, prédéfinissent avant toute formulation un équivalent discret des éléments réactifs.

Composants réactifs linéaires

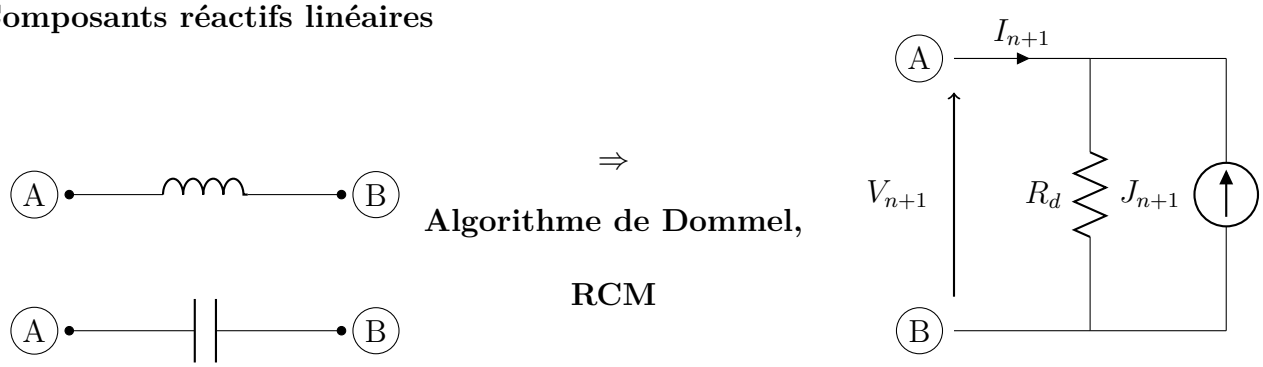


FIGURE 1.6 – Représentation discrète des éléments réactifs.

Chaque élément est représenté par une résistance R_d et un courant d'injection J_{n+1} (Figure 1.6). J_{n+1} est mis à jour à chaque pas de calcul en fonction de l'historique passé de V_{n+1} et de l'historique passé de J_{n+1} lui même. On écrit :

$$I_{n+1} = \frac{1}{R_d} V_{n+1} - J_{n+1} \tag{1.83}$$

$$J_{n+1} = f\left(V_n, V_{n-1}, V_{n-2}, \ldots, J_n, J_{n-1}, \ldots\right) \tag{1.84}$$

R_d est la résistance discrète.
J_k est la valeur du courant d'injection à l'instant $T_k = kT_s$.
V_k est la tension à l'instant $T_k = kT_s$.
f est une fonction linéaire qui dépend du pas de calcul, de la caractéristique de l'élément réactif (inductance ou capacité) et de la méthode d'intégration numérique.

La résolution du système entier peut alors se faire à l'aide de méthodes bien connues et utilisées pour résoudre les circuits statiques [12] (telles que la méthode de *Norton*, la méthode de *Thévenin*, la méthode des mailles, la méthode de superposition). La méthode nodale (1.2.3) utilise ces représentations discrètes RCM pour aboutir à la méthode nodale discrète.

Grâce à ces travaux, les logiciels jusqu'alors utilisés pour analyser des circuits ne comportant que des résistances, des sources de tensions et des sources de courant sont candidats pour se mettre à jour et pouvoir, par la suite, simuler les transitoires de circuits dynamiques.

1.4.1.2 Calcul de la résistance R_d et de la fonction linéaire f

La résistance R_d et la fonction linéaire f dans le cas d'une capacité (ou d'une inductance) pour les méthodes d'intégrations d'Euler arrière, trapézoïdale , de Gear du deuxième ordre se calculent à partir de l'équation (1.85) (ou (1.86)) :

$$I_C = Y_C\, V_C = s\, C\, V_C \tag{1.85}$$

$$I_L = Y_L\, V_L = \frac{1}{s\, L}\, V_L \tag{1.86}$$

L'opérateur de Laplace s est ensuite substitué par son équivalent discret en z^{-1} en utilisant les résultats du tableau 1.1.
On met ensuite le résultat sous la forme des équations (1.83) et (1.84). Le coefficient de V_{n+1} donne l'inverse de R_d. Les termes restants forment les coefficients de la fonction linéaire f. La table 1.2 résume les résultats pour ces trois méthodes d'intégration. D'autres représentations discrètes vues dans la littérature y ont été également rajoutées.

TABLE 1.2 – Représentation discrète des éléments réactifs par une résistance discrète et un courant d'injection.

$\mathbf{R_d}$ et $\mathbf{J_{n+1}}$	**L**	**C**	**Méthode d'intégration**
R_d	$\frac{L}{T_s}$	$\frac{T_s}{C}$	Euler arrière
	$2\frac{L}{T_s}$	$\frac{1}{2}\frac{T_s}{C}$	Trapézoïdale
	$\frac{3}{2}\frac{L}{T_s}$	$\frac{2}{3}\frac{T_s}{C}$	Gear du deuxième ordre
J_{n+1}	$-I_n$	$\frac{1}{R_d}V_n$	Euler arrière
	$-I_n - \frac{1}{R_d}V_n$	$I_n + \frac{1}{R_d}V_n$	Trapézoïdale
	$-\frac{4}{3}I_n + \frac{1}{3}I_{n-1}$	$\frac{4}{3}\frac{V_n}{R_d} - \frac{1}{3}\frac{V_n}{R_d}$	Gear du deuxième ordre

12. On définit par circuit statique un circuit ne comportant que des résistances, des sources de tension continue et des sources de courant continu.

1.4.2 Latency Insertion Method

La méthode dite "Latency Insertion Method (LIM) " proposée par Schutt-Aine [2001] suppose que :

- toutes les branches comportent une inductance (Figure 1.7-a) ;
- chaque nœud est lié à la masse via une capacité (Figure 1.7-b).

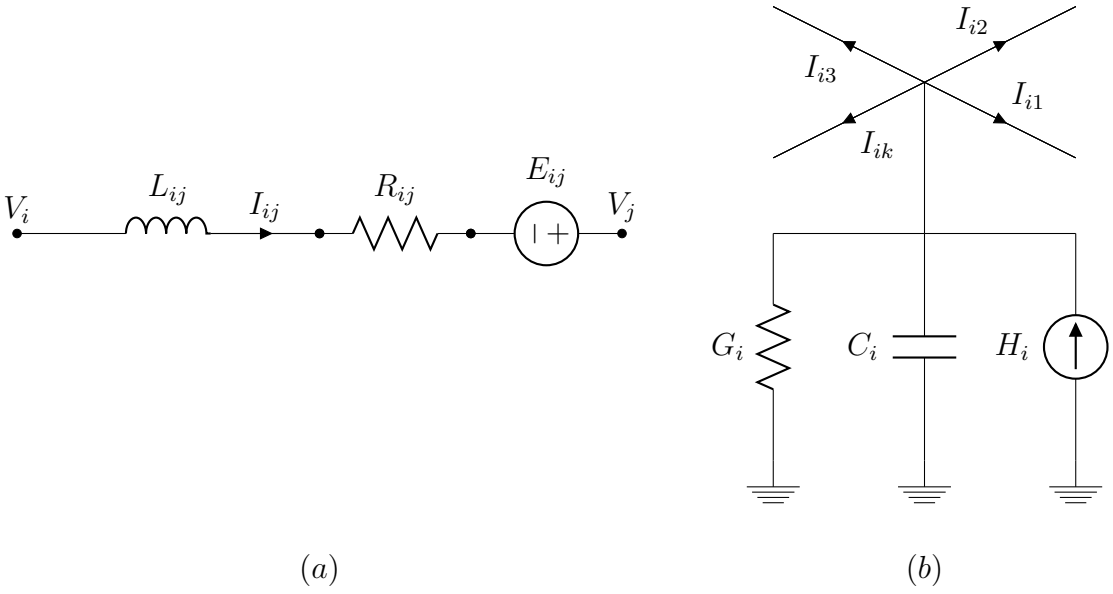


FIGURE 1.7 – Branche (a) et noeud (b) dans la méthode d'insertion de délai ou LIM.

Une petite inductance (respectivement une grande capacité) est rajoutée dans une branche (respectivement entre un nœud et le nœud de référence) si elle n'existe pas.

La méthode consiste à calculer les tensions au niveau de chaque nœud et ensuite les courants de chaque branche. Les séquences de calcul des tensions et des courants sont décalées d'une demi-période de calcul.
On désigne par :
M_i le nombre de branches connectées au nœud i ;
V_i^k la tension au nœud i à l'instant $T = k.T_s$;
I_{ij}^k le courant de la branche (i, j) à l'instant $T = k.T_s$;
G_i l'admittance de liaison au de référence au nœud i ;
C_i la capacité de liaison du nœud i au nœud de référence ;
H_i la source de courant nœud i au nœud de référence.
Les tensions de nœuds se calculent comme suit :

$$V_i^{n+1/2} = \frac{\frac{C_i V_i^{n-1/2}}{T_s} + H_i^n - \sum_{k=1}^{M_i} I_{ik}^n}{\frac{C_i}{T_s} + G_i} \tag{1.87}$$

Les courants de branche se calculent comme suit :

$$I_{ij}^{n+1} = I_{ij}^n + \frac{T_s}{L_{ij}} \left(V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^n + E_{ij}^{n+1/2} \right) \tag{1.88}$$

Les équations (1.87) et (1.88) présentent un fort potentiel de calcul parallèle car :

- la tension d'un nœud peut être calculée indépendamment de celles des autres nœuds ;
- le courant d'une branche peut être calculé indépendamment de celui des autres branches.

Des traitement spéciaux sont appliqués pour les inductances mutuellement couplées, les capacités dans les branches et les sources de tension/courant contrôlées.
Par exemple, comme la branche définie par la méthode LIM ne comporte pas de capacité, on fait appel à la représentation RCM (voir 1.4.1) de la capacité (avec la source de courant du RCM remplacée par l'équivalent en source de tension) qui se situe dans une branche pour avoir la forme de branche de la Figure 1.7.

Cette méthode est bien adaptée à la modélisation d'un réseau de distribution d'énergie complexe étant donné que l'inductance dans la branche et la capacité reliant un nœud à la terre existe naturellement dans les modèles de lignes de transmission d'énergie électrique.
Par contre, dans le cas où l'on n'a pas par exemple une inductance dans une branche, on y insère une petite inductance. Le fait d'ajouter une petite inductance rajoute un pôle rapide (venant de l'inductance rajoutée et de la capacité de liaison au nœud de référence). Pour conserver la stabilité numérique (voir 1.3.1.6), il est alors nécessaire d'utiliser un pas de calcul suffisamment petit. Ceci est pénalisant car le pas de calcul maximal autorisé doit alors être réduit. La réduction du pas de calcul augmente les performances requises pour le matériel dédié à la simulation temps réel.

1.4.3 Lignes de transmission

La simulation numérique des circuits électriques est une science plus jeune que l'analyse du comportement des lignes de transmission. Le comportement des lignes de transmission a été étudié bien avant, grâce aux théories de propagation des ondes. Une autre façon de modéliser le comportement des éléments réactifs est d'assimiler leur comportement à celui des lignes de transmission.

Nous reprenons l'exemple d'une capacité C décrit par Hui et Christopoulos [1990]. On assimile cette capacité à une ligne de capacité linéique C_l :

$$C_l = \frac{C}{l} \tag{1.89}$$

Une ligne est caractérisée par :
- sa vitesse de propagation d'onde :

$$V_p = \sqrt{L_l C_l} \tag{1.90}$$

- et son impédance caractéristique :

$$Z_c = \sqrt{\frac{L_l}{C_l}} \tag{1.91}$$

L_l est l'inductance linéïque.
C_l est la capacité linéïque.
l est la longueur de la ligne.

Le délai de propagation T_p est alors le temps nécessaire pour une onde pour faire l'aller-retour de la ligne de longueur l. On fait correspondre ce délai de propagation au temps de calcul utilisé pour la simulation :

$$T_p = T_s = 2lV_p \tag{1.92}$$

Des équations (1.90) et (1.92), par substitution de l'inductance linéïque, nous pouvons déduire l'expression de l'impédance caractéristique Z_c de la ligne qui représente la capacité C :

$$Z_c = \frac{T_s}{2C} \tag{1.93}$$

L'assimilation de la capacité C par une ligne de transmission de capacité linéïque $C_l = \frac{C}{l}$ introduit alors une inductance parasite L_p. On déduit L_p de l'équation (1.92) :

$$L_p = \frac{T_s^2}{4C} \tag{1.94}$$

Plus le pas de calcul est petit, plus l'inductance parasite diminue et plus notre ligne de transmission se comporte comme une capacité. L'inductance parasite introduit donc une erreur qui diminue avec le pas de calcul.

Une démarche similaire peut être menée le cas d'une inductance et aboutit aux mêmes conclusions. La représentation d'une inductance par une ligne de transmission d'énergie introduit une capacité parasite qui sera une source d'erreur.

A partir des études sur les lignes de transmission Dommel [1969] Marti [1982] Hui et Christopoulos [1990] Hui et Christopoulos [1990] Matar et Iravani [2010b], les logiciels précédemment utilisés pour l'analyse comportementale des lignes de transmission sont capables de donner des résultats dynamiques pour n'importe quel circuit électrique.

Nous avons vu jusqu'ici les différentes façons de modéliser la partie sans commutation des systèmes de puissance électrotechnique sans pour autant traiter la présence d'élément de commutation. La section qui suit traite le cas de convertisseurs statiques présents au niveau du système de puissance.

1.5 Cas particulier des convertisseurs statiques

Les convertisseurs statiques engendrent des changements de configuration rapides du système. Ces changements modifient les équations qui régissent la dynamique du système. Un changement de configuration peut arriver à tout moment. Dans un environnement de simulation hors Temps réel, il est possible d'altérer le temps de simulation et de refaire un calcul en retournant en arrière. Cela n'est pas possible dans un environnement Temps réel.

Les changements de configuration découlent d'un évènement qui est :
- soit un évènement sur les signaux de gâchette (donc un évènement venant des signaux physiques du contrôleur, et donc externe au simulateur Temps réel) ;
- soit une annulation du courant ou un changement de signe du courant (cas d'ouverture d'une diode ou d'un thyristor : évènement interne au simulateur Temps réel) ;
- soit une tension qui atteint une valeur seuil (cas de déclenchement d'une diode : évènement interne au simulateur Temps réel).

La détection de ces évènements pose problème. Ces évènements peuvent arriver à n'importe quel moment pendant que le support de calcul effectue des calculs :
- les signaux de gâchette sont des signaux rapides et ne peuvent être pris en compte qu'au pas suivant dans le cas de la simulation Temps réel ;
- le courant peut avoir des dynamiques rapides selon le type de charge. Il est fréquent qu'en numérique, la valeur du courant ne soit pas absolument nulle, ou que l'on manque le passage du courant par zéro puisque le courant peut changer de signe en un seul pas de calcul.

Bedrosian et Vlach [1992] ainsi que Alimeling et Hammer [1999] soulignent le problème du passage par zéro du courant qui engendre une impulsion de tension (impulsion de Dirac) dans le cas d'une diode en série avec une inductance par exemple. En effet, au moment du passage par zéro, la variable d'état (courant de l'inductance) forcée à zéro, n'est plus une variable d'état du système. Similairement, lorsqu'une capacité est court-circuitée, sa tension n'est plus une

variable d'état. Un changement de topologie qui garde ou augmente le nombre de variable d'état (courant d'une inductance ou tension d'une capacité par exemple) n'induit pas de discontinuité ni de quantité impulsive. Par contre, une topologie diminuant le nombre de variables d'état engendre une impulsion Massarini *et al.* [1997] si l'on garde la même représentation de la charge du convertisseur sans prendre des dispositions particulières.

L'existence de convertisseurs statiques dans le système de puissance engendre deux grands problèmes pour la simulation numérique :
→ problème de modélisation : dû au changement de topologie du système de puissance.
→ problème de résolution : dû à l'introduction de phénomènes dont la fréquence est élevée comparé au pas de calcul Temps réel admissible.

Les solutions pour traiter ces deux problèmes sont décrites dans les sections suivantes. Les mesures particulières relatives à l'annulation du courant seront discutées dans le chapitre 2, section 2.3.1.

1.5.1 Solutions pour les changements de topologie

1.5.1.1 Modélisation instantanée

La modélisation instantanée des convertisseurs à interrupteurs commandés consiste à représenter ce convertisseur par des fonctions de commutation. On commence par trouver le niveau de tension (respectivement de courant dans le cas de convertisseur source de courant) à un instant t donné en fonction du courant (respectivement de la tension pour les convertisseurs sources de courant) et des combinaisons de signaux de gâchette à cet instant par inspection du circuit.
Dans le cas d'un onduleur de tension par exemple, l'interrupteur complet (incluant la diode) est considéré comme un interrupteur bidirectionnel en courant et commandé par une fonction de gâchette donnée. Cette fonction de gâchette est déterminée par le courant, la tension et la présence ou l'absence d'un signal de commande (impulsion, train d'impulsion ou tension de gâchette selon le type d'interrupteur). Chaque interrupteur possède donc une fonction de gâchette déduite de l'inspection décrite ci-dessus.
De part sa simplicité, un tel modèle nécessite un pas de calcul suffisamment petit Dufour *et al.* [2007a] Dufour *et al.* [2007b] et ne demande aucune méthode particulière de compensation d'évènement.

Cependant, la modélisation instantanée par la logique combinatoire n'est pas correcte lorsque la charge devient génératrice et qu'aucun des interrupteurs du convertisseur n'est commandé Gole *et al.* [1997] Parma et Dinavahi [2007]. Ceci est dû à la relation de causalité figée lors de la modélisation par fonction de commutation. En fait, le convertisseur ainsi modélisé impose toujours sa tension (son courant) à sa charge, ce qui n'est pas vrai dans tous les cas.
Une solution est d'enrichir la représentation par une fonction de commutation gérée par une machine d'état Rakotozafy *et al.* [2011] (ou par un réseau de Pétri) qui casse la causalité figée. Cette méthode nécessite cependant une modélisation spécifique de la charge. Dans le cas d'un onduleur de tension, une fois que les interrupteurs sont ouverts et non commandés, la charge par réciprocité causale imposerait la tension de l'onduleur à courant nul.
D'ailleurs, les outils de modélisation des systèmes séquentiels avec des interrupteurs idéaux (source de courant nul ou source de tension nulle) pour l'analyse des systèmes électrotechniques ont été utilisés depuis le siècle précédent notamment :

- les réseaux de Pétri et les graphes de lien Hautier et Manesse [1989] Kato et Tachibana [1998] Ammous [2002] (Annexes) Maksimovic [2010]. A partir des réseaux de Pétri d'un interrupteur, il est possible de reconstituer le réseau Pétri d'un convertisseur ;
- le Graphe Informationnel Causal (GIC) Hautier et Caron [1999].

Le modèle instantané peut être également affiné en incluant dans les équations du système les

tensions seuils des interrupteurs et leur résistance en conduction. Les pertes par commutation peuvent être évaluées en fonction du nombre de commutations, du niveau de courant et de tension ; les pertes par conduction sont évaluées à partir des résistances internes des composants.

1.5.1.2 Modélisation par interrupteurs impédants

1.5.1.2.1 Interrupteur à résistance variable

Dans le logiciel SPICE3 Version 3CI, les interrupteurs sont modélisés par une faible résistance R_{ON} à la fermeture et une résistance élevée R_{OFF} à l'ouverture. L'idée de représenter les interrupteurs par une résistance élevée lorsqu'ils sont ouverts a aussi été reprise par *Mitsubishi* Harakawa *et al.* [2009].

Un choix non adapté de la valeur de ces résistances introduit des instabilités numériques et donc des problèmes de convergence. Il peut aboutir à des simulations complètement erronées Bedrosian et Vlach [1992]. La valeur de cette impédance influe sur la stabilité numérique de la simulation car elle est susceptible de créer des pôles à hautes fréquences numériquement instables avec le pas de calcul utilisé (voir 1.3.1.6). Pour illustrer ce phénomène d'une façon simple mais approximative, on prend le cas d'un convertisseur de tension alimentant une charge inductive et on suppose que :

$$R_{OFF} \gg \sum \text{ toutes les résistances dans la boucle}$$

Le pôle p_{OFF} introduit sera approximativement :

$$p_{OFF} \approx -\frac{R_{OFF}}{\sum \text{toutes les inductances dans la boucle}}$$

La résistance doit-être suffisamment grande pour réellement modéliser un interrupteur ouvert. Par contre, sa valeur doit-être limitée pour ne pas introduire un pôle continu stable à haute fréquence, mais numériquement instable (voir section 1.3.1.6).

1.5.1.2.2 Interrupteur L/C

Cette approche est une approche purement discrète en utilisant les représentations discrètes des composants par les algorithmes de *Dommel* ou par le RCM (voir section 1.4.1)

L'interrupteur est modélisé par une inductance lorsqu'il conduit et par une capacité lorsqu'il est ouvert (Figure 1.8).

Les résistances discrètes de l'inductance et de la capacité doivent être égales pour éviter d'avoir plusieurs matrices impédances (dans la représentation nodale) à inverser. Comme la matrice admittance est constante, elle s'inverse une seule fois hors ligne. Le temps de calcul pour une modélisation L/C des interrupteurs est donc quasiment le même que pour les systèmes linéaires. La conduction et l'ouverture se limitent à un changement de vecteur d'injection de courant.

Le courant d'injection est calculé comme le courant d'injection d'une inductance (respectivement d'une capacité) lorsque l'interrupteur est fermé (respectivement ouvert). Les expressions des résistances discrètes et des courants d'injection selon la méthode d'intégration choisie peuvent être établies à l'aide de la Table 1.2. La Table 1.3 résume les expressions des résistances discrètes et les courants d'injection d'un interrupteur L/C avec les trois principales méthodes d'intégration numériques utilisées dans le cadre de la simulation Temps réel.

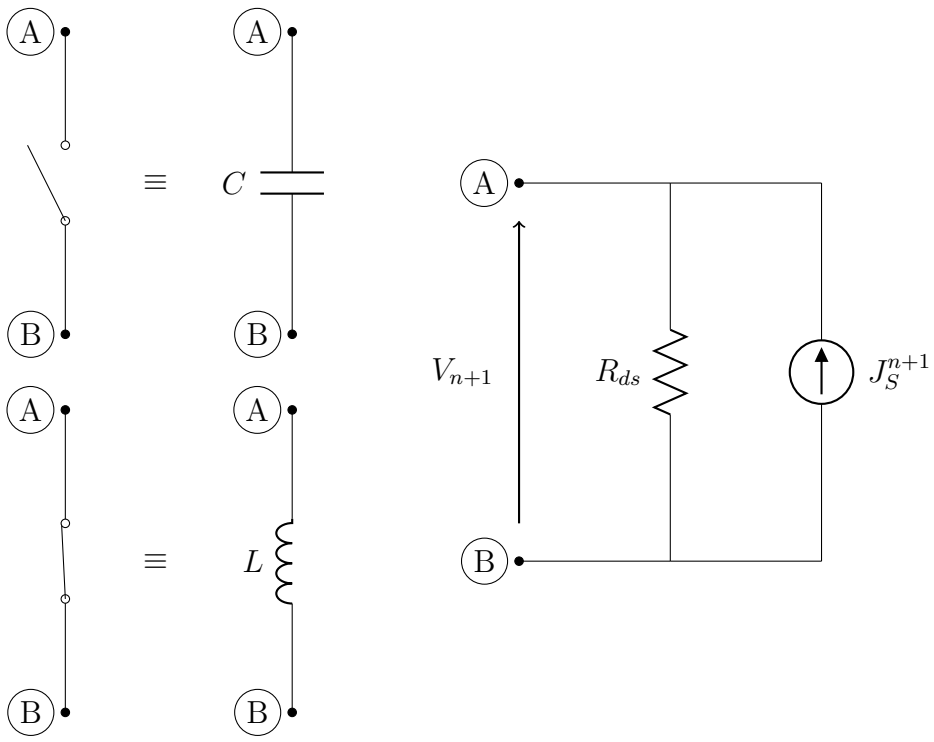


FIGURE 1.8 – Interrupteur de type L/C.

TABLE 1.3 – Expression de la conductance et du courant d'injection d'un interrupteur L/C selon le type d'intégration numérique.

Méthode d'intégration numérique	Résistance discrète R_{dS} de l'interrupteur L/C	Courant d'injection J_S^{n+1}	
		Ouvert	Fermé
Euler arrière	$R_{dS} = \frac{T_s}{C} = \frac{L}{T_s}$	$\frac{1}{R_{dS}} V_n$	$-I_n$
Trapézoïdal (bilinéaire)	$R_{dS} = \frac{T_s}{2C} = \frac{2L}{T_s}$	$I_n + \frac{1}{R_{dS}} V_n$	$-I_n - \frac{1}{R_{dS}} V_n$
Gear du second ordre	$R_{dS} = \frac{2T_s}{3C} = \frac{3L}{2T_s}$	$\frac{4}{3R_{dS}} V_n - \frac{1}{3R_{dS}} V_{n-1}$	$-\frac{4}{3} I_n + \frac{1}{3} I_{n-1}$

On définit par R_{dS} la résistance discrète de l'interrupteur.

Nous allons maintenant étudier le comportement de cette représentation en utilisant la méthode trapézoïdale. Considérons un bras de convertisseur de tension alimentant une charge source de courant I_l. Souvent, lorsqu'un interrupteur est en train de se fermer, un autre interrupteur est en train de s'ouvrir. On étudie le cas illustré sur la Figure 1.9 où un interrupteur est fermé et un autre ouvert.

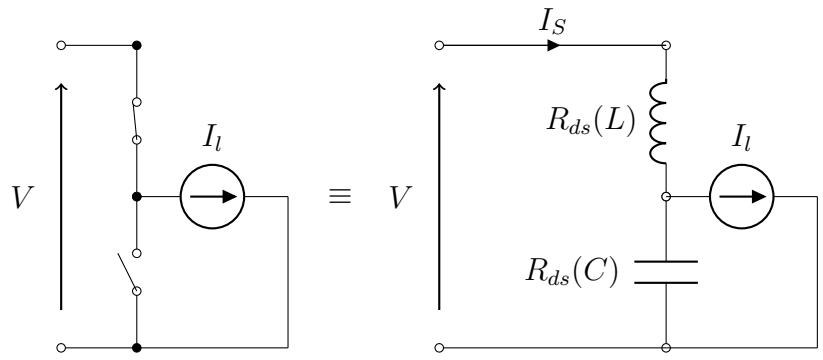


FIGURE 1.9 – Etude du comportement d'un interrupteur L/C selon la méthode trapézoïdale

Le système étudié est linéaire, nous nous intéressons à la fonction de transfert liant le courant I_s au courant de la charge I_l :

$$I_S(s) = \frac{LCs^2}{LCs^2 + 1} I_l(s) \tag{1.95}$$

Les paramètres L et C de l'interrupteur pour la méthode trapézoïdale sont liés par la relation (voir table 1.3) :

$$R_{dS} = \frac{T_s}{2C} = \frac{2L}{T_s} \tag{1.96}$$

La version discrète de l'équation (1.95) est alors (en utilisant (1.96) et par substitution de s par z^{-1} en se référant à la table 1.1) :

$$I_S = \frac{\left(-1 + z^{-1}\right)^2}{2\left(1 + z^{-2}\right)} I_l \tag{1.97}$$

La représentation L/C de l'interrupteur par la méthode trapézoïdale crée deux pôles complexes conjugués ($z_{1,2} = \pm j$) sur le cercle unité. Ce pôle étant à la limite de stabilité, le système oscille. Ceci est prévisible car la version continue présente deux pôles complexes conjugués ($s_{1,2} = \pm\sqrt{LC}$) et la méthode trapézoïdale est A-stable.
Si on effectue la même démarche avec la méthode d'*Euler* arrière et de *Gear*, on constate que les pôles discrets sont à l'intérieur du cercle unité et donc amortis. La méthode d'*Euler* et *Gear* introduisent un amortissement numérique.

Les paramètres L et C de l'interrupteur L/C (explicitement de la résistance discrète R_{dS} de l'interrupteur L/C et le pas de calcul adapté T_s) définissent donc un pôle rapide, artificiellement rajouté par la représentation. Il est crucial de choisir correctement ces paramètres pour assurer la stabilité de la simulation (voir stabilité à la section 1.3.1.6) tout en tenant compte du pas de calcul utilisé.
Il est possible de déterminer la valeur de la conductance G_S et la valeur du pas de calcul T_S en choisissant des valeurs réelles de la capacité parasite $C = C_p$ et l'inductance totale $L = L_T$ (données du fabricant). Cependant, un interrupteur réel est beaucoup plus complexe qu'une inductance à la fermeture et une capacité à l'ouverture. En plus, ce choix de paramétrage impose pratiquement un pas de simulation très petit qui risque de ralentir la simulation hors Temps réel ou de ne pas être atteignable en Temps réel. Néanmoins, un pas de calcul extrêmement petit, de l'ordre de quelques dizaines de nanosecondes, est envisageable sur FPGA au prix d'une grande précision des variables, de la taille des mots et donc de la quantité d'éléments logiques utilisés. Pour la simulation Temps réel, il est préférable de choisir les valeurs de R_{dS} et T_s pour améliorer les performances de la simulation et de réduire l'erreur faite par rapport à des interrupteurs

supposés parfaits. Pejovic et Maksimovic [1994] démontrent que l'erreur que l'on fait en se référant à des interrupteurs parfaits diminue avec le nombre d'itérations. L'erreur sur le courant (respectivement la tension) est inversement proportionnel (respectivement proportionnel) à R_{dS}.

1.5.1.2.3 Interrupteur L/RC

La représentation L/C de l'interrupteur résout complètement le problème de changement de topologie. Cependant, le simulateur EMTP, dans sa version initiale de *Dommel*, utilise cependant la méthode trapézoïdale. Nous avions vu que la représentation L/C de l'interrupteur introduit des pôles asymptotiquement stables avec la méthode trapézoïdale.
Avec des méthodes A-stables comme la méthode trapézoïdale, on ajoute une résistance dans la représentation. Cette résistance permet de maîtriser l'amortissement du système. Le principe est toujours le même qu'avec la représentation L/C :

- avoir une seule admittance pour un interrupteur fermé ou ouvert ;
- jouer sur le vecteur d'injection de courant pour les états fermés et ouverts.

La Table 1.4 résume les expressions de la résistance discrète et le courant d'injection d'un interrupteur L/RC pour différentes méthodes d'intégration.

TABLE 1.4 – Expression de la conductance et du courant d"injection d'un interrupteur L/RC selon le type d'intégration numérique.

Méthode d'intégration numérique	**Résistance discrète R_{dS} de l'interrupteur L/RC**	**Courant d'injection J_S^{n+1}**	
		Ouvert	**Fermé**
Euler arrière	$R_{dS} = R + \frac{T_s}{C} = \frac{L}{T_s}$	$\frac{1}{R_{dS}} V_n - \frac{R}{R_{dS}} I_n$	$-I_n$
Trapézoïdal (bilinéaire)	$R_{dS} = R + \frac{T_s}{2C} = \frac{2L}{T_s}$	$\left(1 - 2\frac{R}{R_{dS}}\right) I_n + \frac{1}{R_{dS}} V_n$	$-I_n - \frac{1}{R_{dS}} V_n$
Gear du second ordre	$R_{dS} = R + \frac{2T_s}{3C} = \frac{3L}{2T_s}$	$\frac{4}{3R_{dS}} (V_n - RI_n)$ $-\frac{1}{3R_{dS}} (V_{n-1} - RI_{n-1})$	$-\frac{4}{3} I_n + \frac{1}{3} I_{n-1}$

Les paramètres R , C et L permettent de choisir le dépassement et la fréquence de coupure de l'ensemble RLC série.
On pourra aussi choisir comme critère la minimisation des pertes. En effet, la représentation RC série de l'état ouvert induit plus de pertes par commutation par rapport à l'interrupteur réel. De ce fait, Trevor et James [2005] propose de ne pas inclure les snubbers [13] dans le modèle du convertisseur.
La représentation L/RC de l'interrupteur est également utilisée par le simulateur Real-Time Digital Simulator (RTDS). Des applications avec compensateur statique d'énergie réactive (SVC) avec le groupe ASEA - Brown - Boveri & Cie (ABB) Larsson *et al.* [2007] ont été validées en utilisant cette technique.
Cependant, ce type de modélisation ne convient pas à toutes les applications. Dans le cas d'une application où l'on désire par exemple exploiter la résonance entre la capacité parasite de l'interrupteur et l'inductance du transformateur afin de faire de la commutation douce Deng *et al.* [2009], l'interrupteur L/RC ne convient pas.

Par ailleurs, les représentations par interrupteurs impédants rajoutent des pôles à fréquences élevées. La coexistence de ces pôles très rapides avec les pôles relativement lents du système augmente la plage de valeur des variables utilisées dans la modélisation du système. Dans

13. Circuit d'aide à la commutation, mis en parallèle des interrupteurs de puissance.

la représentation nodale par exemple, on constate de grandes admittances et de très petites admittances dans la même matrice admittance du système. L'utilisation d'un pas de calcul plus petit accentue ce phénomène. Ces représentations nécessitent alors une grande précision au niveau des formats numériques.

Nous avions vu jusqu'ici des solutions pour le premier problème (changement de topologie) introduit par les convertisseurs statiques. Ces changements de topologies non seulement compliquent la modélisation mais s'effectuent à des fréquences élevées. Nous verrons alors par la suite comment tenir compte de ces phénomènes fréquentiels tout en gardant un pas de calcul Temps réel compatible avec la technologie actuelle.

1.5.2 Solutions apportées au problème de résolution

1.5.2.1 Modèle moyen

Deng *et al.* [2009] proposent un modèle moyen de convertisseur DC/DC DTHB bidirectionnel pour une base matérielle RTDS. Rappelons ici que les modèles d'interrupteurs L/RC ne sont pas adaptés pour une application contenant des convertisseurs à commutation douce. De telles applications sont par exemple utilisées dans l'industrie des véhicules hybrides pour minimiser les pertes.
Les modèles moyens de certaines topologies simples sont faciles à développer. Par contre, pour un convertisseur complexe, le modèle moyen nécessite beaucoup de temps de développement Paquin *et al.* [2009]. Le modèle moyen peut être différent pour chaque topologie, parfois pour chaque type de charge ou pour chaque mode de fonctionnement Deng *et al.* [2009]. Dans le cas du hacheur série, la relation qui lie la tension de sortie avec le rapport cyclique est différente si on est en mode de conduction continue ou si on est en mode de conduction discontinue. Ce type de modélisation a une représentativité limitée mais ne demande pas une grande puissance de calcul Etxeberria-Otadui *et al.* [2002].
Le point de départ du développement du modèle moyen est le modèle d'interrupteur idéal Ammous [2002]. Le modèle moyen ne tient donc pas compte des pertes lors des commutations. Ammous *et al.* [2003] propose donc une méthode de calcul des pertes lors des commutations par un modèle moyen en se basant sur les caractéristiques linéarisées des tensions et des courants lors des commutations. Ces caractéristiques sont déduites des relevés expérimentaux ou directement des datasheets des interrupteurs.
L'idée d'une interface de pré-traitement des pulses par une moyenne glissante est aussi traitée par Gombert [2005]. On peut enrichir la composante moyenne des pulses par une composante harmonique. Cette méthode est valable quand il existe une relation explicite entre les signaux de gâchette et le niveau de tension. Cette relation est évidente dans le cas d'un hacheur série ou d'un onduleur à deux niveaux.

1.5.2.2 Interface de pré-traitement

Les signaux de commande des interrupteurs ont une très haute résolution. Dans le cas des applications de l'entreprise GE-PC, les impulsions sont précises à $33ns$ près. Le pas de calcul minimum sur des simulateurs à base de coeur avoisine aujourd'hui les $40\mu s$. Pour tenir compte des évènements dont la résolution ne peut être prise en compte avec le pas de calcul Dinavahi [2000] , on peut procéder en deux étapes :

- enregistrer les différents évènements susceptibles de changer la topologie du convertisseur (changement de signaux de gâchette, annulation d'un courant, dépassement d'un seuil de tension) ;
- traiter ces évènements un à un par des pas de calcul intermédiaires.

Les évènements sur les signaux de gâchette qui se déroulent entre les instants $t = t_k$ et $t = t_{k+1}$ sont enregistrés et datés par une interface de pré-traitement.

Ils sont ensuite traités et ils fixent les pas intermédiaires à effectuer entre $t = t_{k+1}$ et $t = t_{k+2}$. Si entre $t = t_{k+1}$ et $t = t_{k+2}$ on détecte un évènement susceptible de changer la topologie du convertisseur autre que les signaux de gâchette (évènement tension ou courant) :

- on détermine la date de cet évènement ;
- on retourne un pas en arrière sur la dernière configuration valide ;
- on fait un pas intermédiaire jusqu'à cet évènement ;
- on tente de faire un pas jusqu'au prochain évènement de signaux de gâchette ou bien jusqu'à la fin du pas de calcul si il n'y a plus d'évènement de signaux de gâchette ;
- on vérifie qu'aucun évènement susceptible de changer la topologie du convertisseur n'a eu lieu. Sinon, on réitère le processus jusqu'à sortir du pas de calcul avec une configuration valide.

Il faut noter que la mesure de correction par interface de pré-traitement ne peut pas corriger les valeurs calculées pendant le pas de calcul au cours duquel la commutation se produit Dufour et Belanger [2001]. Ces méthodes permettent uniquement d'obtenir un résultat correct au prochain pas de calcul Faruque et Dinavahi [2010]. La raison est que l'on ne peut pas retourner en arrière en Temps réel. Par ailleurs, cette méthode est coûteuse en temps quand le nombre d'évènements pendant un pas de calcul augmente.

Le simulateur Real-Time Laboratory (RT-LAB) utilise la solution avec interface de pré-traitement sous le nom de Time Stamped Bridge (TSB) Mohammed *et al.* [2007]. RTDS l'utilise sous le nom de Digital Input Times Stamp (DITS) (Digital Input Times Stamp) Forsyth et Kuffel [2007]. Ce type de solution pour simuler en Temps réel des convertisseurs statiques a par exemple été utilisée pour la simulation Temps réel d'une éolienne Guo *et al.* [2009] Pak et Dinavahi [2009].

1.6 Bases matérielles pour la simulation temps réel

On distingue deux principaux groupes de bases matérielles de calcul dans le domaine de la simulation Temps réel :

- les processeurs tels que les DSPs, les PowerPC, les microprocesseurs ;
- les FPGAs.

1.6.1 Simulateurs à base de cœurs

Cet ensemble regroupe tous les calculateurs à exécution séquentielle de code. Un programme donné est enregistré en mémoire. Chaque instruction est exécutée l'une après l'autre par une architecture matérielle figée. Le cœur de calcul possède des registres de calcul, des registres d'adressage, une ou plusieurs unités arithmétiques et logiques et un jeu d'instruction.
Quoiqu'il en soit, ces composants utilisés seuls ne peuvent pas tenir compte des évènements à haute résolution temporelle. Pour calculer la valeur moyenne des signaux de gâchette ou pour les horodater, des composants additionnels tels que des FPGAs, des Peripheral Interface Controllers (PICs), ou des compteurs binaires sont nécessaires.
Figueroa *et al.* [2004] utilisent une interface de calcul de la valeur moyenne des signaux de gâchette (avec un $PIC18F8720$ à 8 bits) pour tenir compte des commutations. Les tensions en sortie du convertisseur se calculent alors avec un pas beaucoup plus grand (50 µs) sur une station de travail Dell précision Workstation 450n sous Linux avec deux processeurs $2.666Ghz$ et 512Mo de RAM. L'utilisation d'un FPGA est proposée comme perspective d'avenir pour réaliser une interface de pré-traitement plus aisément contrôlable.
Les simulateurs commerciaux Temps réel d'aujourd'hui sont principalement basés sur des cœurs. On y trouve par exemple les Giga Processor Cards (GPCs), les Triple Processor Cards (3PCs) et les RISC Processor Cards (RPCs) de RTDS Kim *et al.* [2010] Trevor et James [2005]. Hormis les GPC, ces calculateurs ont des pas de calcul minimaux avoisinant les 50 µs Dufour *et al.*

[2005] Mohammed *et al.* [2007] Paquin *et al.* [2009] Faruque et Dinavahi [2010] Li *et al.* [2010].

1.6.2 Simulateurs sur FPGA

Les FPGAs ont été utilisés dans un premier temps pour le pré-traitement des signaux de gâchette pour des modèles de convertisseur tournant avec un pas de l'ordre de plusieurs dizaines de microsecondes. Dans le simulateur de RT-LAB par exemple, l'horodatage des signaux de gâchette se fait avec des FPGAs de Xilinx, via une librairie appelée RT-EVENTS.

Le marché offre des composants FPGAs de plus en plus performants avec des fréquences d'horloge élevées, plus d'éléments logiques voire même des blocs Digital Signal Processors (DSPs) de multiplication embarqués. Il devient alors intéressant d'utiliser les FPGAs en tant qu'éléments actifs de simulation Temps réel. La grande résolution temporelle de ces composants permet de modéliser des phénomènes jusqu'ici non pris en compte en Temps réel.

La finesse de représentation est améliorée par l'insertion de composant actif FPGA dans le simulateur. Dufour *et al.* [2007b] Dufour *et al.* [2007a] modélisent une machine synchrone à aimants permanents par les éléments finis. Le modèle à éléments finis est généré en utilisant le logiciel JMAG-RT mais pourrait être également généré par le logiciel FLUX2D. Ce modèle a permis de voir les effets d'encoches (l'harmonique 12 apparaît pour une machine à 24 encoches) et vérifier à l'avance le comportement d'une machine en cours de fabrication.

Les transitoires des commutations peuvent être simulés en Temps réel sur ces composants. Un exemple traité est la commutation des Insulated Gate Bipolar Transistors (IGBTs), modélisée avec des transitoires linéarisés Parma et Dinavahi [2007]. Un modèle plus proche de l'expérimental est développé dans Myaing et Dinavahi [2010] où les pertes au niveau des composants sont obtenues suite à des mesures sur le système réel et une mise en mémoire des dynamiques lors des commutations.

Des implantations uniquement sur cible FPGA ont été développées récemment dans la littérature
Parma et Dinavahi [2007]Faruque et Dinavahi [2010] Chen et Dinavahi [2009]. L'implantation 'tout' sur FPGA devient une tendance dans le développement de simulateurs Temps réel. En plus, des méthodes simples et accessibles sont proposées pour pouvoir faciliter la mise en œuvre de l'implantation sur FPGA Monmasson et Cirstea [2007] Paiz *et al.* [2007] Karimi *et al.* [2010].

1.6.3 Comparaison des deux types de bases matérielles

Un avantage de l'utilisation des processeurs est leur facilité de programmation. Une modification dans une partie de programme en C se fait d'une manière générale beaucoup plus rapidement qu'une modification dans une partie d'une description en Very High Speed Integrated Circuits Hardware Description Langage (VHDL). L'implémentation sur FPGA nécessite plus de temps qu'une implémentation sur processeur.

Les processeurs utilisés dans la simulation Temps réel utilisent des formats en simple ou double précision. Ces formats permettent de représenter une grande plage de valeurs. L'implémentation sur FPGA se fait généralement en virgule fixe. L'utilisation du format à virgule fixe nécessite une étude plus approfondie sur le nombre de bits nécessaires pour représenter correctement une variable.

Un avantage des FPGAs est le fait de pouvoir bénéficier du parallélisme sans avoir recours à d'autres calculateurs. Sur les simulateurs à base de processeurs, on est obligé de rajouter d'autres processeurs pour accroitre la capacité de calcul parallèle. L'architecture du FPGA est particulièrement adéquate pour les calculs parallèles et à pas multiples Crosbie *et al.* [2010].

Sur FPGA, le pas de calcul peut rester constant même si la taille du système augmente.

La seule limite est la capacité du FPGA cible Matar et Iravani [2010a], étant donné que le nombre d'éléments logiques utilisés augmente avec la taille du système.
A titre comparatif, sur processeur, on pourrait simuler n'importe quel système ayant n'importe quelle taille. Cependant, plus la taille du système augmente, plus la taille du programme augmente et plus le temps d'exécution augmente.

1.6.4 Approches et méthodes selon les bases matérielles

La représentation d'état aussi bien que la représentation nodale sont utilisées dans le domaine de la simulation Temps réel. Des études de génération automatique de modèles avec ces représentations ont été présentées.
Par exemple, Alimeling et Hammer [1999] proposent une méthode permettant de générer les matrices de la représentation d'état pour n'importe quelle configuration topologique du système, à partir d'une matrice généralisée.

La représentation nodale discrète du système offre un grand parallélisme. Plus la taille de la matrice admittance augmente, plus le niveau de parallélisme augmente et ne peut plus être réalisé sur processeurs. En effet, le nombre de multiplicateurs et d'additionneurs, le niveau de 'pipelining' sont fixes sur un processeur du fait de l'architecture figée. Par contre, on peut construire autant de multiplicateurs et d'additionneurs que nécessaire sur un FPGA tant que les ressources logiques élémentaires sont disponibles. La représentation nodale convient bien à une implantation sur FPGA.

Instinctivement, si l'on diminue le pas de calcul, on est obligé de pouvoir coder des variations très petites. Plus le rapport entre les constantes de temps du système de puissance et le pas de calcul T_s est grand, plus le nombre de bits nécessaires pour représenter les variables est élevé.
D'une façon analytique, les pôles en z d'un système discrétisé se rapprochent de plus en plus de 1 quand le pas de calcul diminue Newman et Holmes [2003]. Ces pôles sont donc à la limite de stabilité et une petite variation sur les coefficients ou les variables de calcul peut conduire à l'instabilité. Par ailleurs, deux systèmes différents peuvent avoir des réponses similaires si les formats utilisés ne sont pas adéquats. Il est alors nécessaire d'augmenter la précision en utilisant plus de bits pour le codage. Le nombre de bits est pourtant limité pour les calculateurs comme les DSPs Fang *et al.* [2005] à virgule fixe et les FPGAs. La réduction du temps de calcul augmente alors la quantité de ressources nécessaires.
Le problème ne concerne pas seulement les composants qui utilisent le format à virgule fixe. Une manifestation similaire à ces phénomènes est signalée par Trevor et James [2005] qui utilise une approche nodale (voir section 1.2.4) implémentée sur des processeurs de calcul. Avec des pas de simulation de l'ordre de 50µs, il obtient des grandes admittances et de très petites admittances dans la matrice générale du système. Son choix de pas de calcul de 2µs pour simuler des convertisseurs statiques accentue ce phénomène. Trevor et James [2005] conseille alors l'utilisation des processeurs à double précision flottante.
Cependant, la problématique reste la même pour une implémentation sur FPGA avec un petit pas de calcul sauf que l'implantation en double précision sur FPGA est aujourd'hui peu traitée et utilisée car peu avantageuse. Des applications avec des cœurs embarqués sur FPGA (Softcore) existent mais ces cœurs perdent l'intérêt qu'est le parallélisme. En effet, les cœurs embarqués exécutent comme les processeurs le code instruction par instruction.
L'implémentation sur FPGA demande donc une attention particulière quant au format des mots, surtout pour des pas de calcul faibles vis à vis des constantes de temps. La méthode delta Newman et Holmes [2003] Monmasson et Cirstea [2007] permet d'utiliser un pas de calcul suffisamment petit tout en minimisant le nombre de bits nécessaires.

1.7 Optimisation et réduction de coût

1.7.1 Techniques de partitionnement des modèles

On est parfois contraint de simuler un système sur différentes bases matérielles lorsque les ressources sur un calculateur ne suffisent plus.
L'idée d'interfacer deux simulateurs différents a commencé à faire l'objet de recherche depuis l'avènement du simulateur numérique. Pendant la transition de l'analogique vers le numérique, une partie du système de puissance était simulée en analogique. L'objectif de l'interfaçage est donc de découpler astucieusement le système pour obtenir un certain parallélisme. Les deux simulateurs peuvent être de natures différentes (numérique – analogique), ou de résolutions différentes (pas de calcul différents si numérique), ou de niveaux de puissance différents (cas du PHIL).
Le système est alors découplé par les inductances des transformateurs, par les lignes de transmission (Méthode de *Bergeron*) ou par partitionnement au niveau des capacités ou des inductances Paquin *et al.* [2009]. D'une façon simple, le point d'interfaçage est choisi de façon à ce que le retard de l'interface soit négligeable devant les constantes de temps des signaux échangés.
Nous allons voir par la suite ces différentes techniques qui permettent de partitionner un système.

1.7.1.1 Techniques de partitionnement

On peut utiliser les lignes de transmission pour partitionner un système. Une des propriétés des lignes de transmission est qu'elles divisent la matrice conductance du système de puissance en plusieurs blocs de matrices diagonales Kuffel *et al.* [1995].
Le délai de propagation d'une ligne de transmission est utilisé pour compenser le retard dû à l'interface d'interconnexion entre deux simulateurs. Verma *et al.* [2006] utilisent par exemple un découplage par une ligne courte. Le partitionnement se fait comme sur une inductance étant donné que la ligne courte a un comportement prédominant de type inductif.

On peut également effectuer le partitionnement au niveau des éléments réactifs Kato *et al.* [2010]. On suppose que le courant J d'une inductance ne varie pas brusquement. Tous les circuits en série avec l'inductance L sont donc traversés par le courant J et peuvent être partitionnés comme indiqué sur la Figure 1.10. Le même principe est appliqué pour une capacité en parallèle (Figure 1.11).

1.7.1.2 Effet du partitionnement

Le découplage du système suppose qu'une grandeur électrique (courant à travers une inductance ou tension aux bornes d'une capacité) évolue peu au cours du pas de calcul. Dans le cas de la Figure 1.11 par exemple, on suppose que la tension aux bornes de la capacité C varie peu en un pas de calcul. On effectue alors les calculs sur les sous-circuits avec la valeur de cette tension. La somme des courants obtenue pour les sous-circuits est utilisée pour actualiser la tension de la capacité au prochain calcul. Le découplage introduit du retard dans la modélisation car les valeurs actuelles des courants des sous-circuits sont calculées sur une valeur antérieure de la tension de la capacité. Un retard supplémentaire est introduit si les sous-circuits sont calculés sur deux bases différentes, utilisant une interface d'échange de données.
Le principal obstacle dans l'optique d'utiliser plusieurs simulateurs ou plusieurs processeurs est le retard engendré par l'interface qui assure l'échange de données. Le système entier est non seulement simulé avec des résolutions différentes, mais un retard supplémentaire venant de l'interface est inséré.

Dans le cas d'un simulateur hybride analogique-numérique, le retard vient des conversions numérique-analogique et analogique-numérique
Kuffel *et al.* [1995]. Pour un simulateur à plusieurs calculateurs, le retard vient de l'échange de

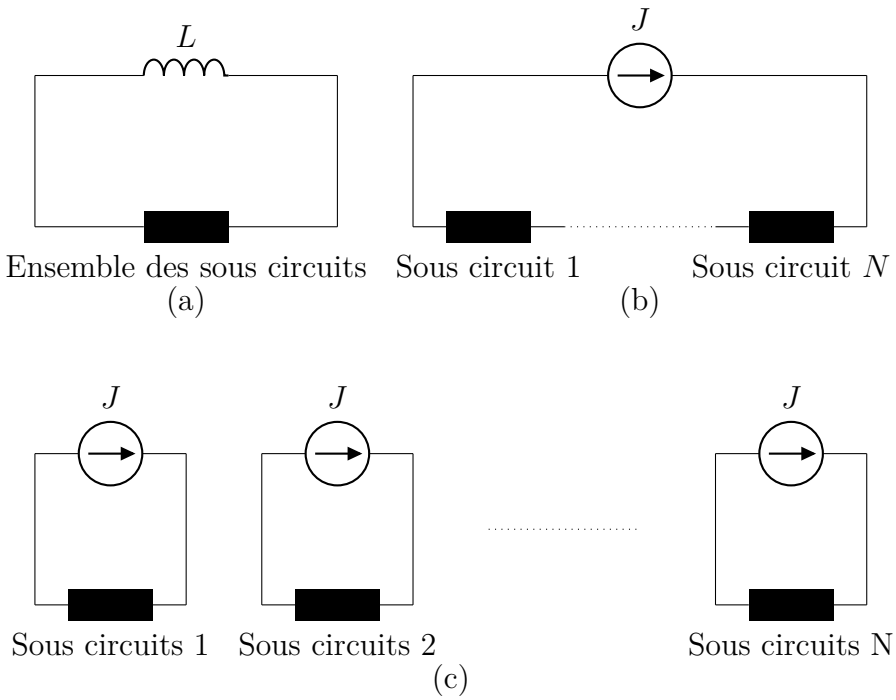


FIGURE 1.10 – Partitionnement par une inductance série : Circuit avec inductance (a) – Détection des sous circuits en série (b) – Circuits partitionnés équivalents (c).

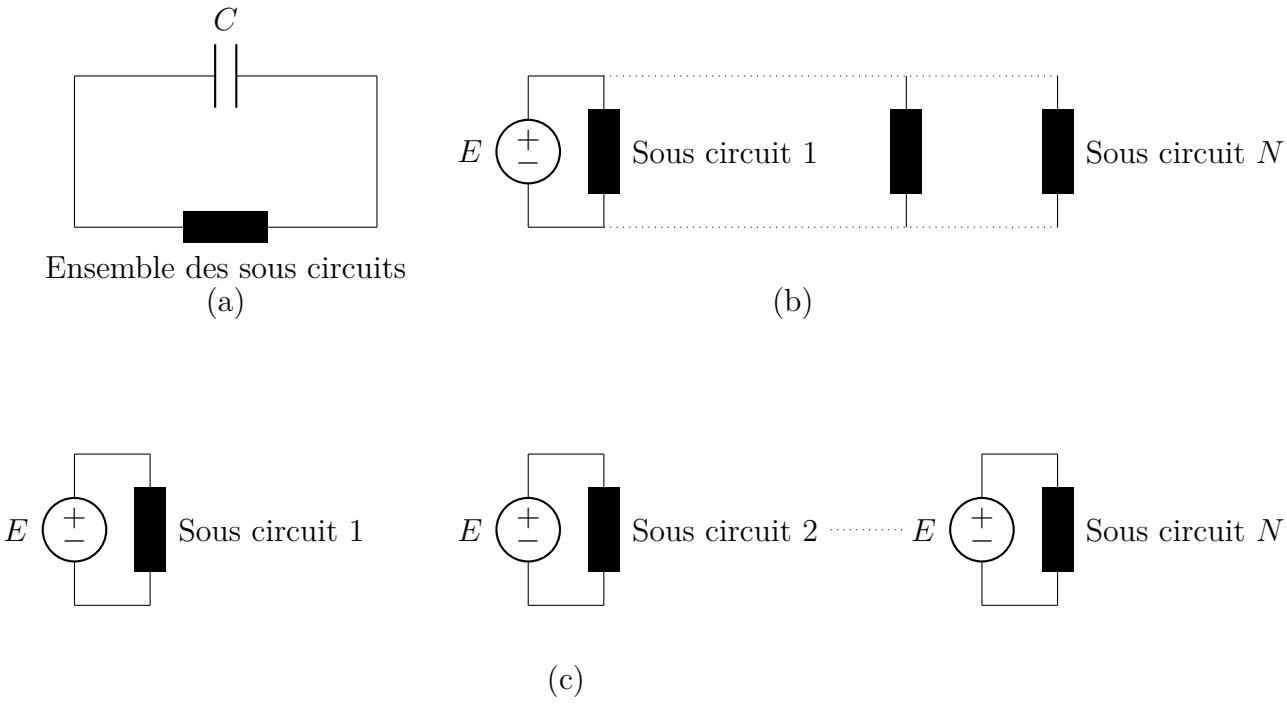


FIGURE 1.11 – Partitionnement par une capacité parallèle : Circuit avec capacité (a) – Détection des sous circuits en parallèle (b) – Circuits partitionnés équivalents (c).

données entre les deux calculateurs Mercier *et al.* [1995] (section VI – paragraphe 3). A cause du retard au niveau du point de découplage, des oscillations au niveau des variables de découplage prennent naissance Le-Huy [2006]. En fonction du retard introduit par l'interface et la grandeur caractéristique de l'élément réactif (capacité, inductance), l'effet du retard entre les simulateurs peut être plus ou moins grave.

Le partitionnement par une ligne de transmission permet de réaliser une interface théori-

quement stable tant que le délai de propagation d'un signal le long de la ligne est plus grand que le délai introduit par l'interface d'échange de données. Si la ligne n'est pas suffisamment longue (délai de propagation plus petit que le délai d'interfaçage), il est impossible d'interfacer les deux simulateurs. Les conclusions sont les mêmes pour les partitionnements par les éléments réactifs, le délai dû au partitionnement doit-être négligeable par rapport à la dynamique de la variable de découplage.

Il est alors intéressant d'utiliser une interface suffisamment rapide vis à vis de la dynamique des variables de découplage. L'utilisation de la mémoire cache par exemple est avantageuse dans le cas des processeurs à cœurs multiples. Dans le cas du simulateur RTDS, la liaison entre deux cartes actives est faite avec une liaison optique. Lorsqu'on divise un système en deux sous-systèmes simulés avec des pas de calcul différents, un des systèmes est sur-échantillonné et l'autre sous-échantillonné.

1.7.1.3 Sur-échantillonnage

Les grandeurs venant du système sous-échantillonné peuvent être gardées constantes vis-à-vis du système sur-échantillonné.
Une approche plus performante consiste à faire une prédiction à partir de l'historique du signal sous échantillonné (d'où l'utilisation d'une méthode d'intégration Explicite) pour les points intermédiaires et de faire une mise à jour à chaque période du système lent.

1.7.1.4 Sous-échantillonnage

La solution la plus aisée est de prendre en compte une valeur venant du système sur-échantillonné et de négliger les autres. Cette technique est simple mais n'a aucune représentativité du point de vue électrotechnique.
En général, les systèmes électrotechniques fonctionnent en valeur moyenne au niveau d'un courant à travers une inductance ou d'une charge stockée dans une capacité Pimentel [2006]. La valeur vue par le système sous-échantillonné sera alors une moyenne des valeurs du système sur-échantillonné sur la période du système lent.

1.7.2 Traitement des matrices creuses

Les matrices de représentation des systèmes électrotechniques comportent souvent des zéros. L'enregistrement de ces zéros augmente la taille de la mémoire nécessaire. Par ailleurs, lors du traitement de ces matrices, on a recours à différentes techniques de réduction de matrices. Il existe des techniques pour minimiser le nombre de zéro au cours de la réduction de la matrice. Dans le domaine de la simulation numérique, on peut faire appel à diverses techniques de traitement des matrices creuses pour minimiser les ressources utilisées George et Liu [1989] Amestoy *et al.* [1996] Davis *et al.* [2004] Reißig [2007].

1.7.3 Algorithmes de réduction du nombre d'opération de calcul

Les calculs nécessaires pour la simulation des circuits électriques utilisent des multiplications matricielles. Il existe des méthodes réduisant le nombre de multiplications nécessaires pour la multiplication de deux matrices. Les instructions de multiplication sur des cœurs prennent plus de temps que les additions.
La multiplication de deux matrices de taille $(n \times n)$ nécessite n^3 opérations. Le point d'entrée des méthodes de réduction du nombre de multiplication est la découverte par *Strassen* d'une méthode permettant de multiplier deux matrices de taille (2x2) en utilisant sept (au lieu de huit) multiplications.

Soit deux matrices A et B définies comme suit : $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ et $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$

On calcule le produit $C = A.B$ et on note : $C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$

Strassen calcule sept produits :

- $\gamma_1 = a_{11}b_{11}$
- $\gamma_2 = a_{12}b_{21}$
- $\gamma_3 = (a_{11} + a_{12} - a_{21} - a_{22})b_{22}$
- $\gamma_4 = a_{22}(b_{11} - b_{12} - b_{21} + b_{22})$
- $\gamma_5 = (a_{21} + a_{22})(-b_{11} + b_{12})$
- $\gamma_6 = (-a_{11} + a_{21} + a_{22})(b_{11} - b_{12} + b_{22})$
- $\gamma_7 = (a_{11} - a_{21})(-b_{12} + b_{22})$

On obtient la matrice C : $C = \begin{pmatrix} \gamma_1 + \gamma_2 & \gamma_1 + \gamma_3 + \gamma_5 + \gamma_6 \\ \gamma_1 - \gamma_4 + \gamma_6 + \gamma_7 & \gamma_1 + \gamma_5 + \gamma_6 + \gamma_7 \end{pmatrix}$

Le nombre $w = log_2(7) = 2.807$ est appelé exposant de l'algorithme de *Strassen*. L'algorithme de base a un exposant 3. Les algorithmes de réduction du coût de calcul dans une multiplication matricielle ont évolué depuis la méthode de *Strassen* :

- l'algorithme de *Strassen* d'exposant 2.807 Strassen [1969] ;
- l'algorithme de Bini *et al.* [1979] d'exposant 2.779 ;
- le théorème de Schönhage [1981] avec un exposant de 2.522 ;
- la méthode du laser de *Strassen* Strassen [1988] avec un exposant de 2.48 ;
- la méthode de Coppersmith et Winograd [1990] avec un exposant inférieur ou égal à 2.376 ;
- et récemment Cohn *et al.* [2005] avec un exposant inférieur ou égal à 2.376.

1.7.4 Pré-enregistrement des matrices de transfert discrètes à durée quantifiée

Dans un environnement Temps réel, on peut dater les transitions par approximation à des instants diviseurs multiples du pas de calcul Wong *et al.* [1987] Luciano et Strollo [1990]. On calcule avant la simulation différentes matrices discrètes donnant la dynamique du système durant un pas de temps diviseur du pas de calcul. Cette technique permet d'éviter de recalculer en ligne les matrices discrètes en effectuant des bonds correspondant aux durées précalculées.

Si le vecteur de commande U reste constant pendant la période où la configuration j est valide, la solution des équations d'état (1.2) et (1.3) pour un pas de temps $\triangle t_j$ et un vecteur d'état initial $X(t_0)$ se met sous la forme connue Chen [1993](page 53) :

$$X(t_0 + \triangle t_j) = e^{A_j \triangle t_j} X(t_0) + A_j^{-1}(e^{A_j \triangle t_j} - I)B_j U_j \tag{1.98}$$

A_j est la matrice de dynamique valide entre $t = t_0$ et $t = t_0 + \triangle t_j$;
B_j est la matrice de dynamique valide entre $t = t_0$ et $t = t_0 + \triangle t_j$.
On pose :

$A_j^i = e^{A_j \triangle t_j}$ la matrice de dynamique discrète valide entre $t = t_0$ et $t = t_0 + \triangle t_j$;

$B_j^i = A_j^{-1}(e^{A_j \triangle t_j} - I)B_j$ la matrice de commande valide entre $t = t_0$ et $t = t_0 + \triangle t_j$;

On divise alors l'intervalle de temps T_s en K intervalles $\triangle t_j = \frac{T_s}{K}$. On obtient les matrices :

$A_K^i = A_j^i(\triangle t_j = \frac{T_s}{K})$

$B_K^i = B_j^i(\triangle t_j = \frac{T_s}{K})$

Les différentes valeurs des matrices A_K^i et B_K^i sont alors enregistrées à l'avance en mémoire pour différentes valeurs de K. En choisissant une valeur de K puissance de deux ($K = 2^n$), on peut utiliser un pointeur binaire dont la valeur est fonction des évènements datés par l'interface

de pré-traitement des signaux de gâchette pour adresser vers les bonnes matrices discrètes. Cet astuce permet d'éviter d'inverser des matrices et de calculer leurs inverses en ligne.

1.8 Conclusion

Nous avons abordé dans ce chapitre les différentes notions utiles pour le développement d'un simulateur Temps réel de dispositifs électrotechniques. La modélisation du système de puissance peut se faire de deux manières :

- soit à partir d'une modélisation en continu ;
- soit directement par une modélisation discrète.

Dans le cas d'une modélisation continue, il est nécessaire de discrétiser le système. Les méthodes de discrétisation présentées dans ce chapitre permettent de discrétiser un modèle de système continu. Nous avons montré qu'il est important d'examiner la stabilité de la méthode d'intégration avant de l'utiliser.

Plusieurs méthodes de modélisation directement discrète ont été traitées. Ces méthodes peuvent faire gagner du temps lors du développement d'un simulateur. Toutefois, l'usage dépendra de l'expérience et des habitudes des utilisateurs.

Les problématiques de changement de topologies et de résolution introduites par les commutations des convertisseurs statiques nécessitent une prise en charge particulière.
Le changement de topologie est traité de deux façons :

- soit par une modélisation spéciale des interrupteurs (interrupteurs impédants) qui met en œuvre une topologie fixe capable de prendre en compte toutes les topologies ;
- soit en traitant par intervalle de temps au cours duquel une topologie donnée est valide.

La première solution promet une bonne performance de calcul car la simulation des convertisseurs par cette solution est identique à la simulation de système linéaire. Cependant, une attention particulière doit-être accordée aux paramètres du modèle des interrupteurs et la méthode d'intégration choisie.
La deuxième solution se rapproche plus de l'interrupteur idéal mais nécessite de traiter un à un les changements de topologies.
Le problème de résolution concerne beaucoup plus la simulation Temps réel que la simulation hors Temps réel. Les contraintes Temps réel imposent une durée de traitement inférieure au pas de calcul Temps réel et exigent donc des performances de calcul. Cette performance ne peut pas être atteinte aujourd'hui uniquement avec des processeurs Matar et Iravani [2010a]. On fait alors appel à des composants (type FPGA) plus rapides qui au minimum se chargent de dater les événements.
Les performances temporelles atteintes par les FPGAs conduisent à développer des simulateurs sur cible unique FPGA. L'idée est prometteuse, mais un grand effort doit être fait pour automatiser la génération de modèles de systèmes électrotechniques pour ces composants. La génération automatique de code C pour des processeurs est aujourd'hui chose faite. Des outils de programmation évolués pour programmer des FPGAs existent déjà. La difficulté pour des composants FPGAs est qu'il faudrait une étude de sensitivité [14] ou des essais de simulation hors ligne pour avoir des formats à virgule fixe corrects.

La dernière partie de ce chapitre a présenté des axes d'orientation pour d'éventuelles améliorations et méthodes de réduction du coût du matériel. Elle fait référence au partitionnement

14. Une étude de sensitivité analyse la variation d'une grandeur résultant de la variation d'une autre grandeur. Dans le cas d'une implémentation à virgule fixe, une étude de sensitivité donne l'erreur sur une grandeur due à une erreur de quantification d'un paramètre.

de modèles, au traitement des matrices creuses, aux algorithmes de réduction de multiplications et aux techniques d'accélération Temps réel par pré-discrétisation hors ligne.

Nous avons ainsi examiné des outils et des méthodes variés dans le cadre de la simulation des systèmes électrotechniques sur processeurs et sur FPGAs. Dans les chapitres 2 et 3, nous verrons leurs applications dans le cadre de la *validation temps réel de contrôleur industriel par interface logicielle* et la *validation temps réel de contrôleur industriel par interface matérielle*.

Chapitre 2

Validation temps réel d'un contrôleur par interface logicielle

2.1 Introduction

Un contrôleur industriel standard est composé de deux parties principales (Figure 2.1) :
- une Unité Logicielle (UL) qui exécute l'algorithme de contrôle proprement dit. Cette partie est composée d'un programme séquentiel exécuté sur processeurs, ou DSPs, ou PowerPC ...
- une Unité de Gestion des Entrées/Sorties (UGES) qui génère les signaux physiques de contrôle et les applique au système de puissance. Cette partie est généralement faite sur FPGAs pour une question de flexibilité et possibilité de pré-traitement des signaux.

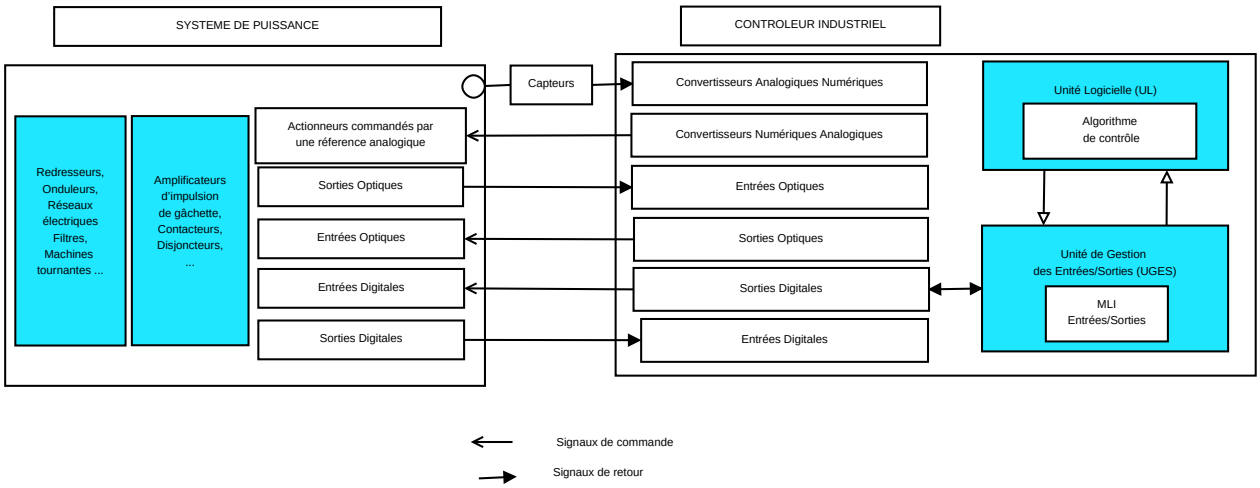


Figure 2.1 – Système électrotechnique avec son contrôleur industriel.

Dans un processus de développement classique, le contrôleur est d'abord conçu puis testé avec un modèle du système de puissance dans un environnement hors temps réel comme Matlab/Simulink. Cette simulation hors temps réel peut être transposée en temps réel sur des bases matérielles spécifiques comme dSPACE dans le cadre de prototypage rapide. Il n'en est pas de même pour les contrôleurs industriels qui sont souvent plus complexes et ne peuvent pas être transposés dans un tel environnement.
Dans les applications que nous traitons, l'algorithme de contrôle est composé de plusieurs tâches tournant avec des pas de calcul différents. Dans ces applications, les périodes des tâches peuvent changer en ligne à cause de changements de stratégies de MLI. Dans le contexte industriel, ces applications sont développées dans un environnement propriétaire pour réduire le coût du

produit final et ne pas avoir à acheter une licence pour chaque produit vendu. Refaire le contrôle dans un autre environnement s'avère donc difficile et onéreux.

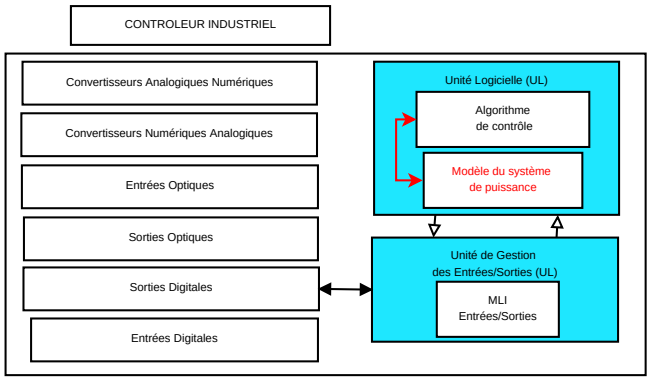


FIGURE 2.2 – Validation par Interface Logicielle

On propose alors une validation que nous appelons Validation par Interface Logicielle (VIL).

2.2 Principe de la VIL

Le principe de la VIL consiste à implémenter sur la même base matérielle le contrôleur et le modèle du système de puissance qu'il pilote (voir Figure 2.2). Les bases matérielles industrielles permettent l'implémentation du modèle du système de puissance en plus de l'algorithme de contrôle sur la même base car elles sont dimensionnées pour pouvoir supporter diverses applications (selon la commande du client) avec une certaine marge sur la charge du processeur.

2.2.1 Cas où les instants de commutation sont calculés *a priori* par l'UL

Pour éviter d'avoir à synchroniser l'algorithme de contrôle et la gestion des entrées/sorties, dans certaines applications, les instants de commutation sont calculés à l'avance sur processeurs. L'interface de gestion des entrées/sorties ne fait que recevoir les niveaux de tension et leurs durées respectives dans un ordre chronologique pour les appliquer sur les interrupteurs du convertisseur statique. Les niveaux de tension et leurs durées respectives à appliquer entre deux instants t_{n+1} et t_{n+2} sont reçus entre t_n et t_{n+1}.

Pour illustrer cette idée, nous allons prendre deux cas :
- le cas d'une MLI précalculée à N angles ;
- le cas d'une MLI avec porteuse et référence synchronisées.

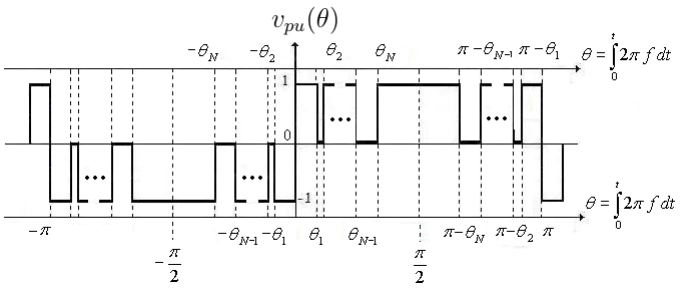


FIGURE 2.3 – Forme d'onde type d'une tension trois niveaux $v_{pu}(\theta)$ pour une MLI précalculée.

Dans le cas d'une MLI précalculée (Figure 2.3), on commence par déterminer la décomposition en série de Fourier de la forme d'onde pour un nombre donné d'angles, N. La forme d'onde est généralement symétrique par rapport à une droite passant par $\theta = \frac{\pi}{2}$ et antisymétrique par rapport à $\theta = \pi$, pour éviter d'avoir des harmoniques de rangs pairs. Par ailleurs, l'angle θ_1 est généralement nul pour éviter de commuter sur deux tensions de signes opposés de la fin d'une période complète à la période suivante.

On choisit $N-1$ composantes harmoniques quelconques que l'on veut annuler. Ces $N-1$ composantes harmoniques, avec l'amplitude de la fondamentale, forment N équations non linéaires dont les inconnues sont les N angles $\theta_{i=2..N+1}$ (θ_1 exclu si fixé à zéro). Ces équations non linéaires sont ensuite résolues pour plusieurs valeurs de la tension fondamentale par des méthodes itératives pour déterminer les angles θ_i.

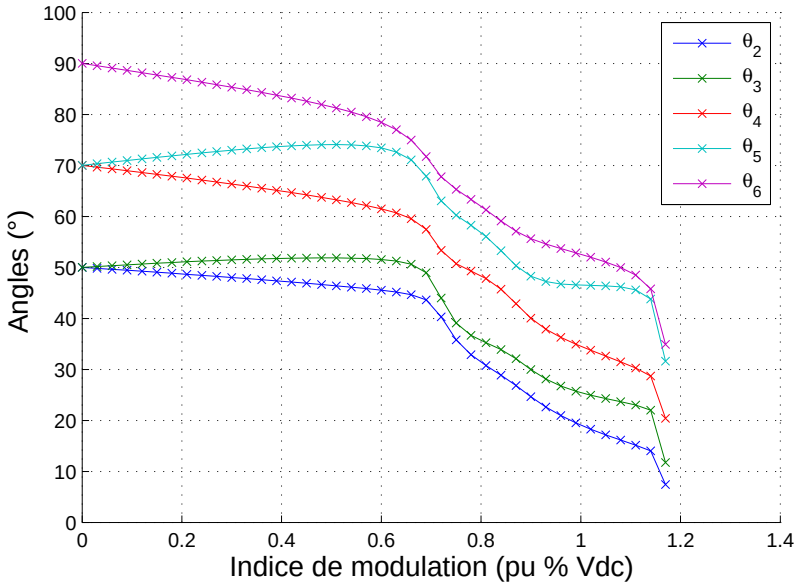


FIGURE 2.4 – Angles de commutation dans le cas d'une MLI précalculée à 5 angles.

La Figure 2.4 montre les angles de commutations en fonction de l'indice de modulation dans le cas d'une MLI précalculée à 5 angles. Les 5 angles donnent 5 degrés de liberté et permettent :

- d'imposer une tension fondamentale ;
- d'imposer l'amplitude de 4 harmoniques (dans notre cas les harmoniques 5, 7, 11 et 13 comme illustré par la Figure 2.5). Les harmoniques pairs sont nuls car la forme d'onde est antisymétrique par rapport à $\theta = \pi$. Les harmoniques multiples de trois ne sont pas vues par la charge si le neutre n'est pas relié.

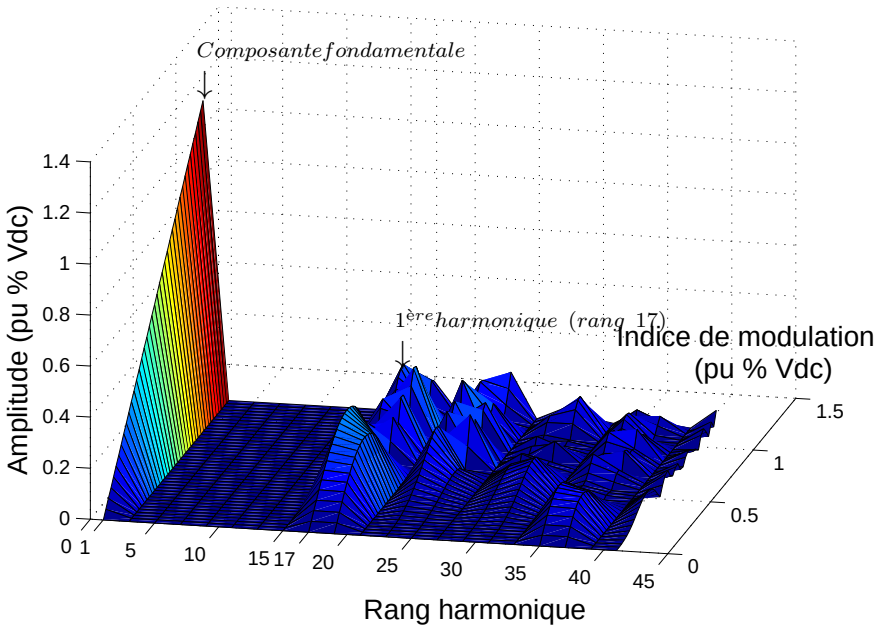


FIGURE 2.5 – Spectre harmonique pour différents indices de modulation.

La Figure 2.6 présente les formes d'ondes de tension dans le cas triphasé pour une indice de modulation égale à 0.6 et une fréquence fondamentale de $37.8Hz$.

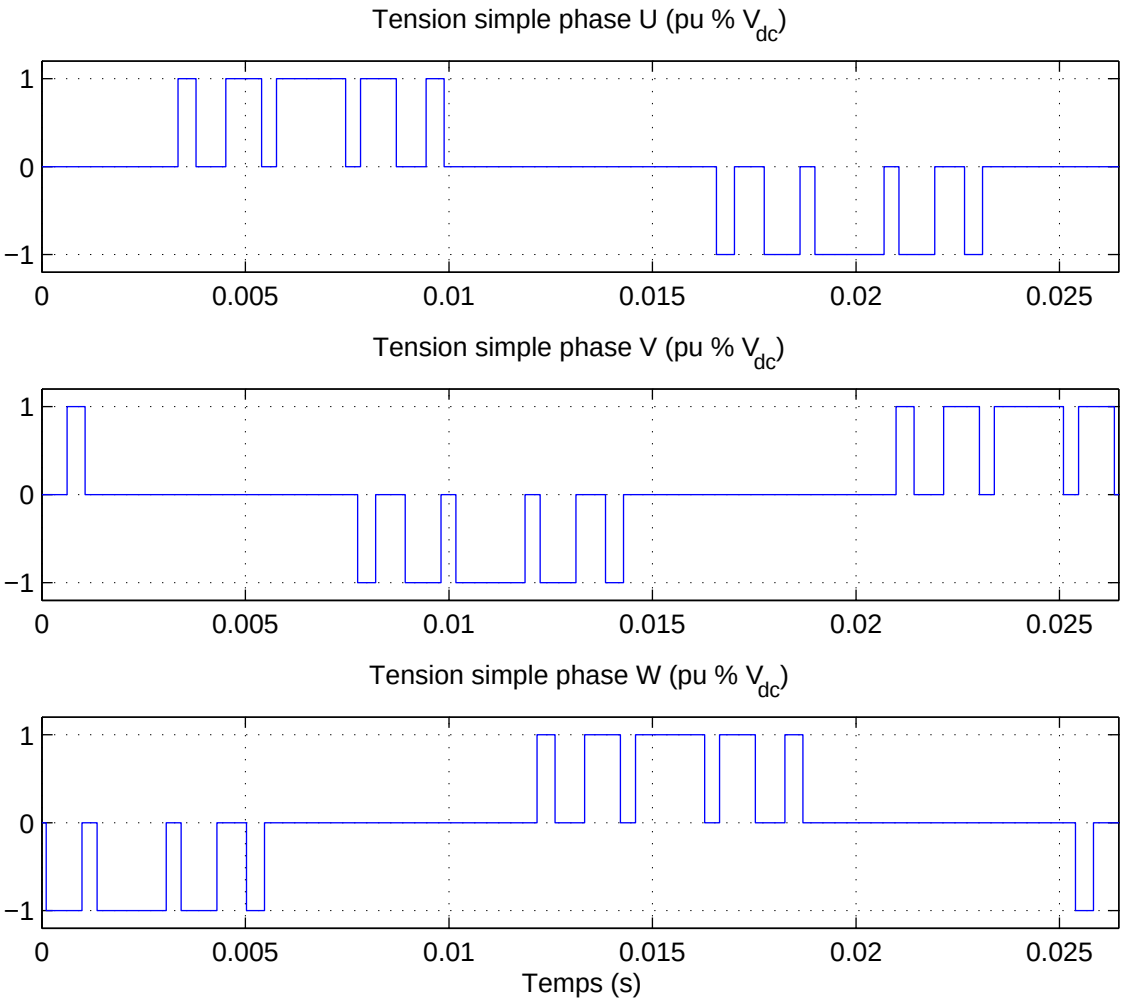


FIGURE 2.6 – Formes d'ondes des tensions simples pour une MLI à 5 angles.

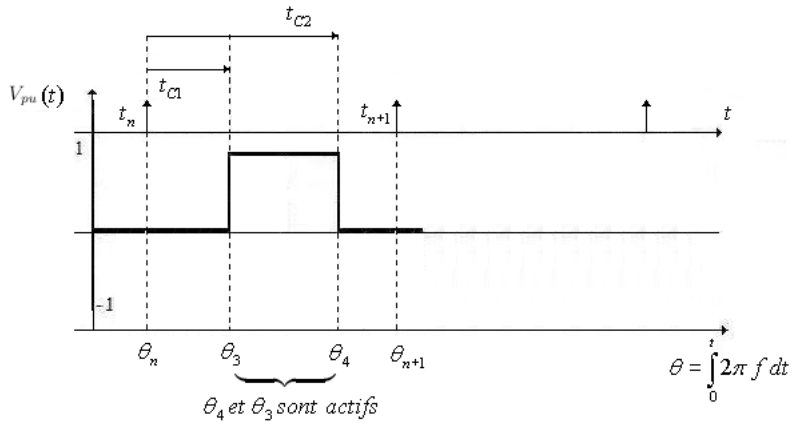


FIGURE 2.7 – Détermination temps réel des instants de commutation T_{Ck}.

Les instants de commutation T_{Ck} entre t_n et t_{n+1} sont déduits des angles θ_{actifs} qui appartiennent à la fenêtre de temps ($(\theta_n = \int_0^{t_n} 2\pi f dt) < \theta_{actifs} < (\theta_{n+1} = \int_0^{t_{n+1}} 2\pi f dt)$). Pour un type donné de MLI précalculée, les angles sont préalablement enregistrés en mémoire en fonction de la fréquence f et du temps t. La Figure 2.7 représente par exemple un cas où deux commutations aux instants T_{C1} et T_{C2} doivent avoir lieu entre t_n et t_{n+1}. Ces instants de commutation sont calculés sur processeur et envoyés à l'unité UGES.

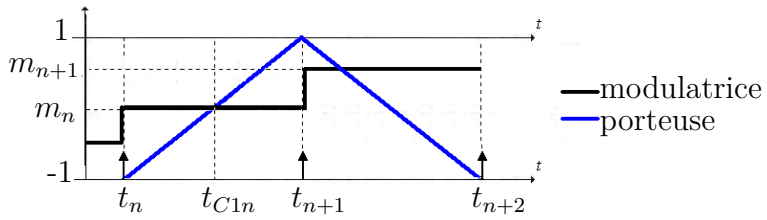


FIGURE 2.8 – Détermination temps réel des instants de commutation T_{Ck} dans le cas d'une MLI à porteuse triangulaire.

Dans le cas d'une MLI avec porteuse (Figure 2.8), entre deux instants discrets t_n et t_{n+1}, la référence de tension (modulatrice m_n) est comparée avec un signal triangulaire pour déterminer les instants de commutation. Nous reprenons ici le cas où la référence de tension est échantillonnée aux sommets de la porteuse. La porteuse est ici bipolaire. Dans ce cas, la période de la tâche du contrôleur correspond à la demi-période de la porteuse. On a une seule commutation par période aux instants T_{C1n} définis comme suit :

$$T_{C1n} = \begin{cases} |m_n|(t_{n+1} - t_n) = |m_n|T_s & \text{si la porteuse a une pente positive} \\ (1 - |m_n|)(t_{n+1} - t_n) = (1 - |m_n|)T_s & \text{si la porteuse a une pente négative} \end{cases} \quad (2.1)$$

Dans plusieurs cas, les instants de commutation et les niveaux de tensions peuvent être calculés à l'avance par l'unité UL de contrôle. Dans ces cas là, l'unité UGES ne fait qu'appliquer les combinaisons adéquates des signaux de gâchette. Cette solution permet un changement de stratégie MLI en ligne sans compliquer la programmation de l'UGES. En effet, les données d'entrées pour l'UGES restent les mêmes quelle que soit la stratégie MLI utilisée. La VIL prenant en compte la présence de convertisseurs statiques devient alors simplifiée dans ce cas car les instants de commutations et les niveaux de tension sont connus.

2.2.2 Cas où les instants de commutation sont traités sur la UGES

Cependant, dans certains cas, l'UGES effectue des traitement liés à la MLI. Un premier exemple est quand l'UL calcule seulement les références de tension et les envoie à l'UGES. Un autre cas par exemple est quand l'UGES assure des durées minimales de conduction et des temps morts et n'applique pas directement les instants de commutations imposés par l'UL. Les traitements faits par l'UGES peuvent être plus ou moins importants selon le cahier de charge de validation.
Si ces traitements sont considérés comme importants, pour simuler exactement le comportement du système de puissance, il est nécessaire de modéliser l'UGES pour faire de la VIL.

2.2.3 Avantages et limitations de la VIL

Hormis le fait qu'on rajoute le modèle du système de puissance sur la même base matérielle sur laquelle s'exécute l'algorithme de contrôle, le contrôleur réel n'est pas modifié. Les signaux en sorties du contrôleur sont dupliqués pour devenir des signaux d'entrées du modèle numérique du système de puissance. Les valeurs des signaux d'entrées (L'UGES effectue toujours les acquisitions) sont écrasées par celles calculées par le simulateur temps réel embarqué.

La VIL ne génère pas de surcoût matériel car le contrôleur réel est validé sur sa propre base matérielle. Les modèles de puissance sont développés dans l'environnement de l'utilisateur et la librairie peut être utilisée par tout ingénieur désirant valider le contrôleur avec des paramètres et des profils de tests donnés. Elle peut être utilisée après la phase de simulation hors temps réel, ou dans le cas d'un paramétrage du contrôleur sur des nouveaux paramètres du système de puissance.
Comme le contrôleur et le modèle du système de puissance sont mis en œuvre dans le même environnement, la prise en main par l'utilisateur final (qui est celui qui développe et teste son contrôleur) est d'autant plus simplifiée.
Cependant, la VIL ne valide pas l'UGES. Elle assure que les signaux théoriques au niveau logiciel sont corrects. Par contre, elle ne vérifie pas si ces signaux sont réellement appliqués sur le système de puissance ou pas. Les problèmes de communication entre l'UL et l'UGES, les défauts et limitations des convertisseurs analogiques/numériques et numériques/analogiques (par exemple l'offset, l'erreur de quantification, la bande passante, un dysfonctionnement, ...), les défauts (par exemple un défaut de synchronisation des impulsions de commande) du circuit de génération de signaux MLI, une mauvaise configuration de l'UGES (comme cette partie du contrôleur industriel est souvent reconfigurable) ne peuvent pas être détectés lors d'une VIL.
La validation du contrôleur incluant l'UL et l'UGES est faite par ce que nous appelons *validation par interface matérielle* traitée au chapitre 3.

Après avoir présenté les notions principales au sujet de la VIL nous allons maintenant reprendre les méthodologies présentées au premier chapitre et qui sont adéquates pour la VIL. La méthode de mise en œuvre est ensuite expliquée. A partir d'une étude de cas par l'approche exacte pour le cas d'une VIL, des résultats de simulation hors temps réel et temps réel sont présentés.

2.3 Modélisation du système de puissance

Nous avions vu au chapitre 1 que dans le cas où le système de puissance comprend des convertisseurs statiques, lors de la simulation temps réel, les convertisseurs sont modélisés :

- soit par des modèles moyens ;
- soit par des modèles instantanés ;
- soit par les interrupteurs impédants.

Notre choix pour une application de VIL porte sur la modélisation instantanée du convertisseur pour les raisons suivantes :

- les résultats sont plus précis que ceux obtenus avec un modèle moyen du convertisseur ou avec une représentation par interrupteur impédant ;
- le développement du modèle moyen nécessite d'abord la connaissance du modèle instantané.

La modélisation instantanée nécessite un horodatage des évènements comme cela a été discuté au chapitre 1 (section 1.5.2.2). Comme expliqué dans les sections 2.2.1 et 2.2.2, les instants auxquels ces évènements arrivent sont dans certains cas disponibles directement dans l'UL ou doivent être déterminés en modélisant l'UGES. Nous verrons dans ce qui suit comment traiter ces instants pour faire de la VIL.

2.3.1 Principe de l'interpolation évènementielle

On suppose maintenant que les instants de commutation et donc les niveaux de tension sont connus. On fait correspondre à chaque intervalle de temps sans commutation un modèle du convertisseur. Dans un intervalle de temps sans commutation, le convertisseur peut être vu comme une source de courant constant ou une source de tension constante, selon sa nature :

- le courant de l'inductance du bus continu d'un onduleur de courant varie très peu entre deux commutations ;
- la(les) tension(s) de(s) capacité(s) du bus continu d'un onduleur de tension varie (varient) très peu entre deux commutations.

La démarche d'interpolation évènementielle dans le cadre de la VIL d'un contrôleur de système incluant un convertisseur de tension à IGBT est décrite à la Figure 2.9. Le principe reste le même pour un convertisseur à thyristor Siala *et al.* [2004].

Si le convertisseur est actif, les niveaux de tension appliqués et leurs durées sont supposés connus (soit parce que l'UL les prend en charge, soit parce qu'un modèle de l'UGES a été fait). Ces évènements sont d'abord triés dans un ordre chronologique. On effectue alors des calculs intermédiaires pour prendre en compte les commutations des composants dans cet ordre chronologique. Une fois tous les évènements traités, les dernières valeurs sont envoyées au contrôleur pour qu'il puisse recalculer ses références au prochain pas. Ce processus est exécuté à chaque période de calcul temps réel.

Si le convertisseur n'est plus actif, le contrôleur laisse le système évoluer de lui-même. On fait alors un pas jusqu'à la fin de la période de calcul. On analyse ensuite si l'état des interrupteurs est cohérent avec les résultats obtenus (Par exemple, une diode ne peut pas être fermée si la tension à ses bornes est positive). Si l'état des interrupteurs n'est pas cohérent, un ou plusieurs évènements susceptibles de changer la topologie du convertisseur ont eu lieu, on retourne en arrière pour repartir d'une configuration valide puis on effectue un pas vers le premier de ces évènements, Gayed *et al.* [1994].
Ces évènements viennent du fait d'une annulation de courant donnée (impliquant l'ouverture d'un interrupteur) ou d'une tension qui a atteint une certaine valeur (impliquant la fermeture d'une diode). Il est nécessaire de corriger les valeurs calculées avant de continuer les calculs jusqu'à la fin de la période T_S.

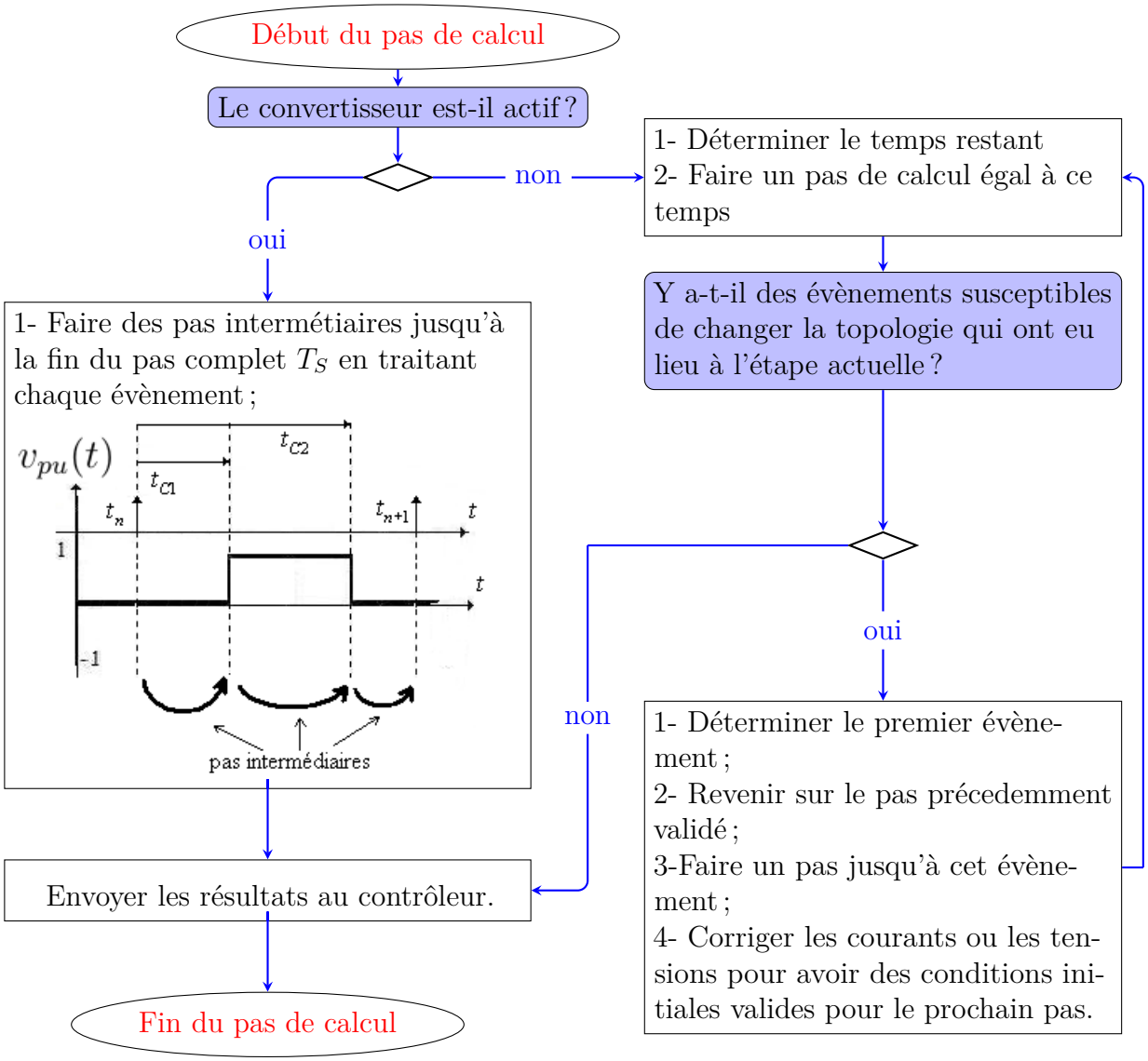


FIGURE 2.9 – Interpolation évènementielle pour la VIL.

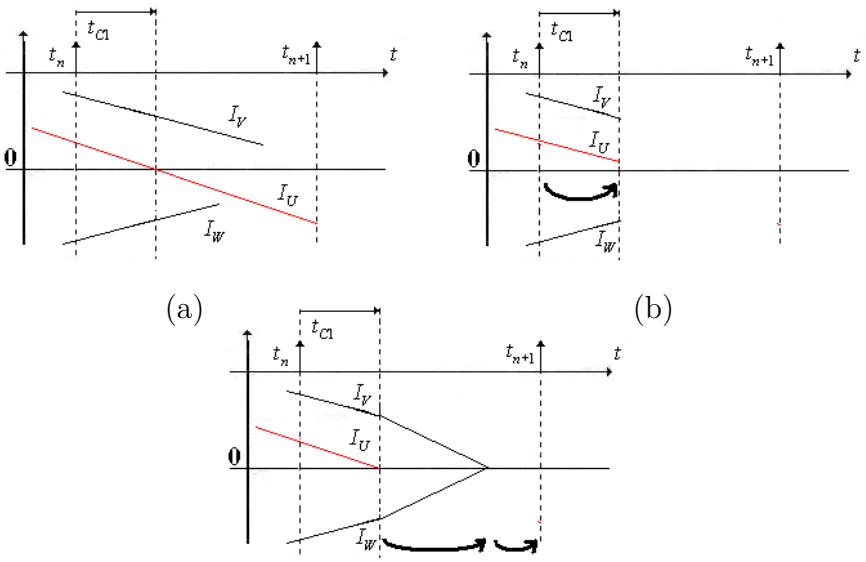


FIGURE 2.10 – Détection de l'annulation d'un courant (a) - Pas intermédiaire jusqu'au temps d'annulation calculé (b) - Correction des conditions initiales et poursuite du calcul jusqu'à la fin du pas de simulation (c)

Nous illustrons la nécessité de correction des résultats dans le cas d'un onduleur de tension triphasé dont le neutre n'est pas relié. On détecte une annulation du courant I_U (Figure 2.10 -a) par interpolation linéaire, Le Doeuff et Benkhoris [1995]. On effectue un pas jusqu'à cet instant, le courant numériquement calculé n'est pas complètement nul (Figure 2.10 -b). Il est alors nécessaire de corriger les trois courants de façon à ce que la somme soit nulle (Figure 2.10 -c) avant de continuer le calcul. Cette mesure permet d'éviter les problèmes liés à la réduction du nombre de variables d'état évoquée à la section 1.5 du premier chapitre.

Maintenant que nous avons déterminé comment prendre en compte le convertisseur statique dans le système de puissance pour une VIL, nous allons étudier comment modéliser la partie sans commutation de ce système de puissance.

2.3.2 Modèle discrétisé pour la VIL

Dans le cas de la VIL, le contrôle prime sur le simulateur temps réel. En effet, le pas de calcul doit être celui du contrôleur. On choisit ce pas en fonction de la stratégie MLI désirée et des performances du système de régulation. Le pas de calcul est néanmoins susceptible de changer en ligne. En aucun cas, le simulateur embarqué dédié à la VIL ne doit imposer le pas de calcul. En résumé :

$$T_s = T_{control} \tag{2.2}$$

avec :
T_s pas de calcul du simulateur ;
$T_{control}$ pas de calcul du contrôleur.

Par conséquent, assurer une stabilité de la simulation numérique pour n'importe quel pas de calcul $T_{control}$ est obligatoire. Il est même possible qu'aucune commutation n'ait lieu durant un pas de calcul entier. Dans ces cas, le modèle du système de puissance est alors exécuté avec un pas égal au pas total $T_{control}$. Deux choix de modèles peuvent être envisagés :

- un modèle exact ;
- un modèle approximatif.

2.3.2.1 Modèle discret exact

Nous privilégions l'utilisation d'un modèle discret exact dans le cas d'une VIL. En effet, l'utilisation d'un tel modèle ne nécessite pas de précaution particulière quant au pas de calcul. Le modèle exact se comporte comme le système réel indépendamment du pas de calcul utilisé. Les différentes manières d'obtenir un modèle discret exact sont décrites à la section 1.3.2 du premier chapitre. Cependant, le développement de ce modèle nécessite du temps et plus d'attention qu'un modèle approximatif.

Dans le cas où on ne peut pas avoir un modèle exact, on peut utiliser un modèle approximatif.

2.3.2.2 Modèle discret approximatif

Le modèle discret approximatif peut être utilisé dans le cadre de la VIL en prenant des précautions particulières. Ces modèles sont moins complexes à obtenir et sont mis en œuvre dans bon nombre de simulateurs temps réel commerciaux.

Le cas de la VIL est plus complexe que celui où le contrôleur et le modèle temps réel du système de puissance sont séparés. Quand les deux entités sont séparées, il est possible d'affecter directement un pas de calcul convenable au simulateur.

En VIL, il faut alors assurer une stabilité numérique quelle que soit le pas de calcul $T_{control}$ du contrôleur. Dans ce cas, nous proposons une limitation interne du pas de calcul. Le principe consiste à effectuer des pas intermédiaires dont la longueur n'excède pas une certaine valeur lorsque le pas est trop grand. Ce pas est déterminé de façon à avoir une méthode d'intégration numérique stable compte tenu des pôles du système de puissance.

Le pas maximum autorisé, T_{smax}, doit être suffisamment petit pour assurer la stabilité numérique mais pas trop petit non plus pour ne pas conduire à des problèmes aux niveaux des coefficients du modèle et ne pas avoir à effectuer trop de pas intermédiaires qui engendreraient trop de calculs.

La Figure 2.11 illustre un cas où trois évènements arrivent pendant un pas de calcul de simulation. Le pas de calcul, comme signalé précédemment, correspond au pas de calcul du contrôleur dans le cas d'une VIL :

- Les pas internes t_{C1} et T_{C4} supérieurs au pas T_{smax} sont donc décomposés en un ou plusieurs pas de durée égale à T_{smax} et des pas résiduels t_{C1R} et t_{C4R}. Les durées de ces pas résiduels sont inférieures au pas maximum T_{smax} ;
- Les pas t_{C2} et t_{C2} sont effectués sans être décomposés car leurs durées sont inférieures à T_{smax}.

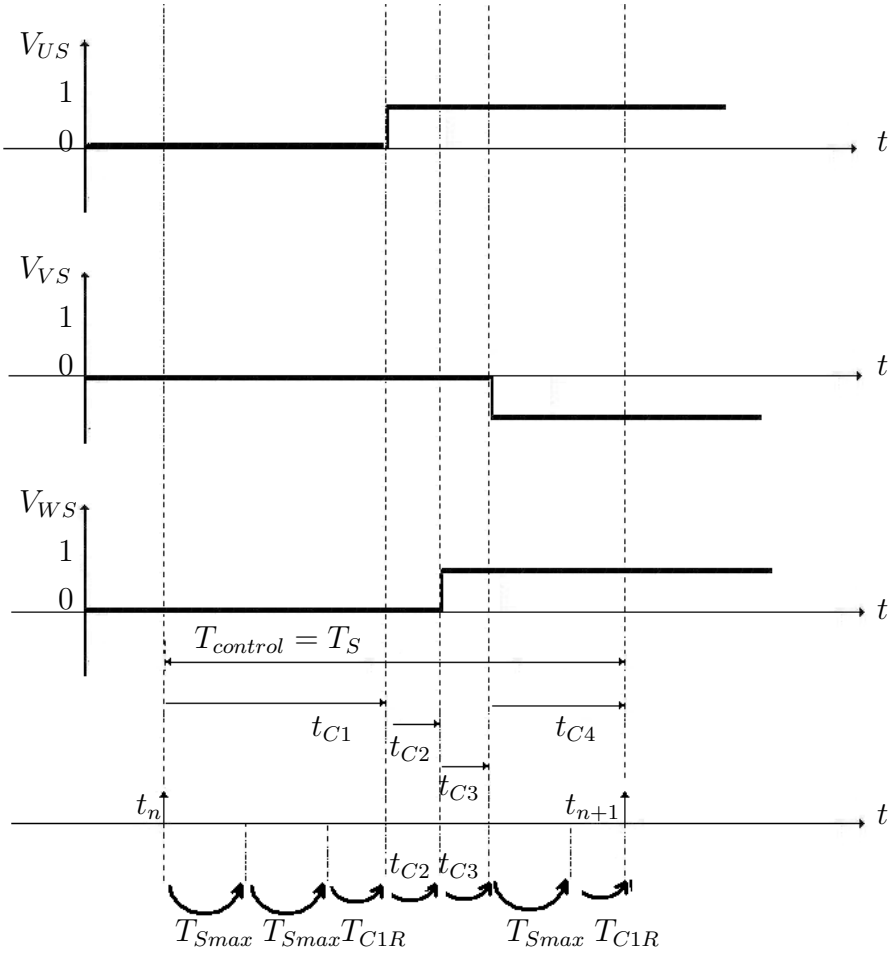


FIGURE 2.11 – Limitation des pas internes à T_{smax}.

Dans ce cas, nous lançons alors une réflexion sur la comparaison entre une méthode d'intégration d'ordre élevé avec un T_{smax} plus grand et une méthode d'intégration de premier ordre avec T_{smax} plus petit que l'on exécute pour avoir les résultats sur un pas $T_{pas} > T_{smax}$:

- une méthode d'intégration élevée utilise plus d'opérations mais sera exécutée moins de fois sur T_{pas} (car T_{smax} est plus grand) ;

– une méthode d'intégration du premier ordre utilise peu d'opérations mais sera exécuté plus de fois sur T_{pas} (car T_{smax} est plus petit).

2.4 Mise en œuvre de la validation

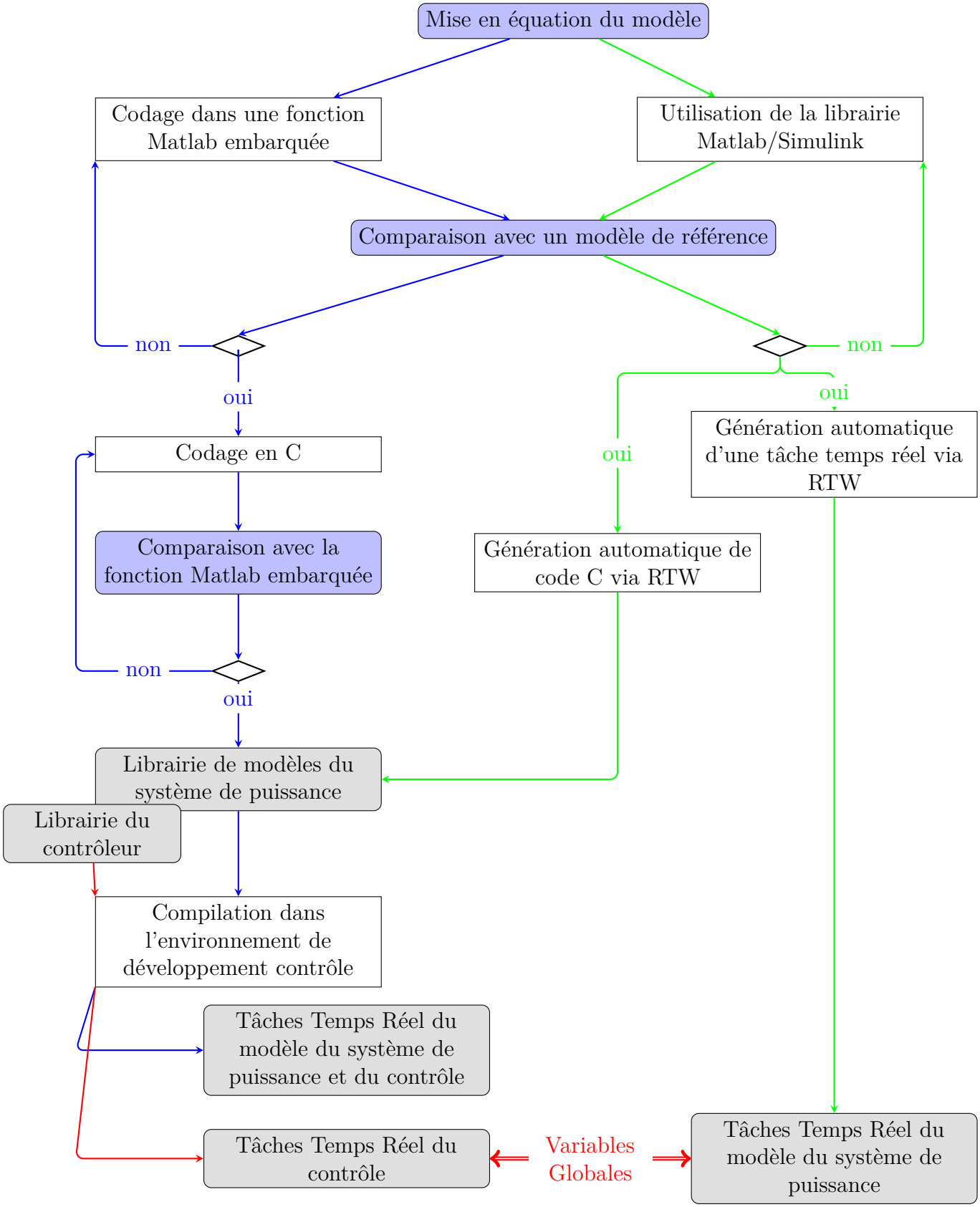


FIGURE 2.12 – Mise en œuvre de la VIL.

La réalisation d'une VIL dans un environnement de développement industriel diffère un peu de celle dans le cas d'un environnement de prototypage rapide. Pour des raisons de coût, de lisibilité de code, les applications industrielles sont généralement développées dans des environnements propriétaires qui ne sont pas forcément commerciaux. La question de lisibilité de code reste quand même un sujet ouvert puisque des efforts ont été faits pour que les codes C générés automatiquement à partir de Real Time Workshop (RTW) conservent les commentaires des codes Matlab originaux, mais le code final reste quand même difficile à lire et donc difficile à modifier.

La Figure 2.12 présente une méthode à risques limités pour développer le modèle du système de puissance en C, implantable sur une cible type processeur. Les codes sont vérifiés au fur et à mesure de la mise en œuvre. Un codage directement en C n'est pas conseillé sans vérification par comparaison avec un modèle de référence.

On commence par la détermination des équations discrètes modélisant le comportement du système de puissance. Ce travail a été discuté au premier chapitre. Les précautions à prendre dans le cas d'une VIL avec modèle approximatif ont été examinées au paragraphe précédant. Nous supposons maintenant que l'étape de mise en équation a été menée avec succès.

Nous proposons deux flots séparés puisqu'une VIL doit, dans la mesure du possible être développée dans le même environnement que le contrôleur. Selon la nature du contrôleur, nous présentons deux flots de conception différents :

- cas d'un contrôleur développé dans un environnement de prototypage rapide ;
- cas d'un contrôleur développé dans un environnement de prototypage industriel.

La mise en œuvre dans le cas d'un prototypage rapide est moins contraignante que celui d'un prototypage industriel. Nous prenons le cas d'un prototypage rapide utilisant RTW pour une cible utilisant le système d'exploitation temps réel *Vxworks*. Cette méthode permet d'utiliser certains blocs de la librairie Matlab/Simulink. Une fois le modèle du système de puissance construit sous Simulink, on peut procéder de deux façons :

- en générant un fichier objet d'extension *.lo* qui sera chargé et exécuté comme une tâche sur la cible ;
- en générant du code C qui pourra être recompilé dans l'environnement de prototypage industriel.

La première approche consiste à faire coexister deux tâches (la tâche contrôle et la tâche de simulation) sur une même base matérielle. Les deux tâches échangent alors à travers des variables globales. Les limitations de cette approche sont les suivantes :

- les seules variables visibles sont les variables globales : le debugage du modèle du système de puissance devient difficile et la version obtenue par prototypage rapide convient à un contrôleur donné. En effet, l'échange à travers les variables globales est seulement possible quand la structure des données du contrôleur est connue à l'avance, et que l'on ait réservé à l'avance certaines adresses pour faire de la VIL.
- dans les applications sur lesquelles nous travaillons, le pas de calcul du contrôleur change en ligne. Nous n'avons pas pu utiliser tel quel le résultat obtenu par prototypage rapide en VIL avec de tel contrôleur. Pour pouvoir utiliser le résultat par prototypage rapide en VIL sur ces contrôleurs, on réveille plutôt la tâche de simulation par la tâche contrôle. Mais cette solution n'est pas industriellement envisageable car elle nécessite une modification dans les fichiers sources de la toolbox RTW et au niveau du contrôleur lui-même.

Cependant, la toolbox RTW peut-être utilisée pour générer du code C adapté à un environnement industriel. Cela nécessite une connaissance profonde de ces deux environnements pour formaliser les structures et formats de données échangés. A l'heure actuelle, des efforts sont faits dans ce cadre là, mais des modifications manuelles au niveau du code automatiquement généré sont encore nécessaires. Par ailleurs, les générateurs de code automatique sont très critiqués quant

aux performances des codes qu'ils fournissent.

Dans le cas d'un prototypage industriel, nous proposons la démarche qui suit.

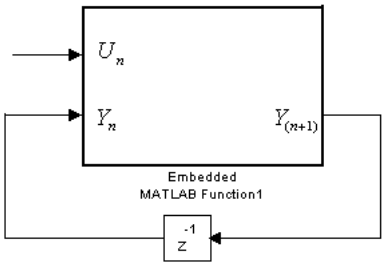


FIGURE 2.13 – Utilisation d'une fonction Matlab embarquée.

On commence par construire le modèle du système et on l'integre dans une fonction Matlab embarquée (Figure 2.13), exécutée avec une période égale au pas de calcul temps réel T_s. Les relations discrètes vues à la section 1.3 liant les entrées de commande U_i et les sorties $Y_{(n+1)}$ sont implémentées comme indiqué à la Figure 2.13. Ces relations utilisent des valeurs antérieures des sorties. L'avantage de l'utilisation d'une fonction embarquée Matlab est qu'elle est proche du code C final contrairement à un schéma utilisant des blocs fonctionnels. Les résultats peuvent être directement comparés avec une simulation de référence comme SimPowerSystem (SPS) ou PLECS. Par ailleurs, cette fonction impose moins de contrainte de programmation par rapport aux *sfunction*s embarquant du code C et permet donc de gagner du temps.
Une fois la fonction embarquée validée par rapport au modèle de référence, on passe au codage en C. L'intérêt du passage par la fonction embarquée se manifeste encore ici. En effet, les seules modifications à faire à ce niveau sont :

- le rajout de l'entête de déclaration des variables et des fonctions ;
- la correction des indices des variables multidimensionnelles car l'indexation commence par zéro en C, ce qui n'est pas le cas dans Matlab ;
- le code C final peut-être testé par comparaison avec la fonction embarquée en leur donnant les mêmes données d'entrées. En choisissant un format en double précision pour les grandeurs réelles, les résultats du code Matlab et du code en C doivent-être exactement les mêmes.

Une fois le code C validé, il est mis en forme et est inclus dans une librairie dédiée à l'environnement de prototypage industriel. Elle peut-être ensuite utilisée par n'importe quel utilisateur familier au développement de contrôle dans l'environnement industriel. La VIL devient similaire à un développement de contrôleur pour les utilisateurs finaux.

2.5 VIL d'un ensemble convertisseur/machine

Nous étudierons le cas du système de puissance illustré par la Figure 2.14.

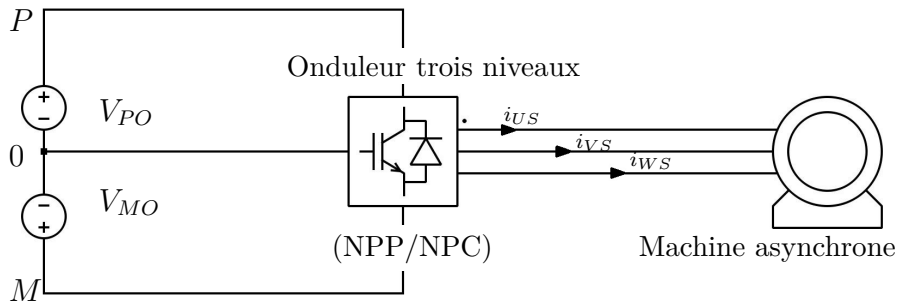


FIGURE 2.14 – Système de puissance simulé dans le cadre d'une VIL.

Le système de puissance est composé d'un onduleur triphasé trois niveaux et une machine asynchrone.

2.5.1 Modèle du convertisseur

Dans les applications de l'entreprise GE-PC, on retrouve les onduleurs triphasés trois niveaux (Neutral Point Piloted converter (NPP) ou Neutral Point Clamped converter (NPC)) dont les bras sont décrits à la Figure 2.15 ou la Figure 2.16.

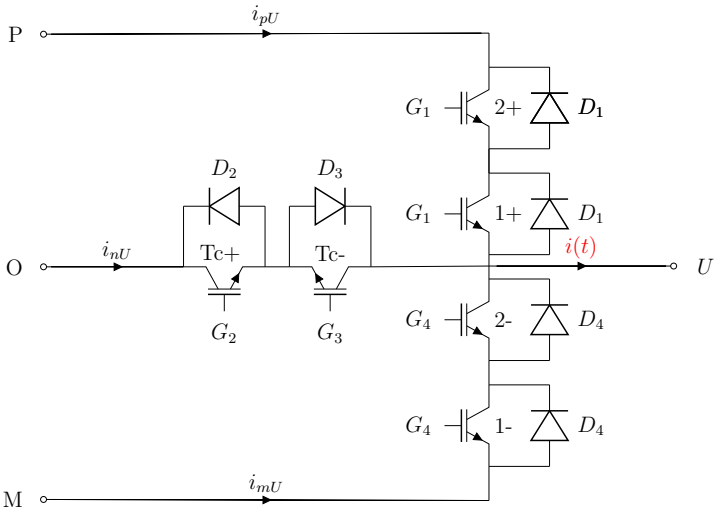


FIGURE 2.15 – Bras de la phase U d'un convertisseur de type NPP - i(t) : Courant de la charge.

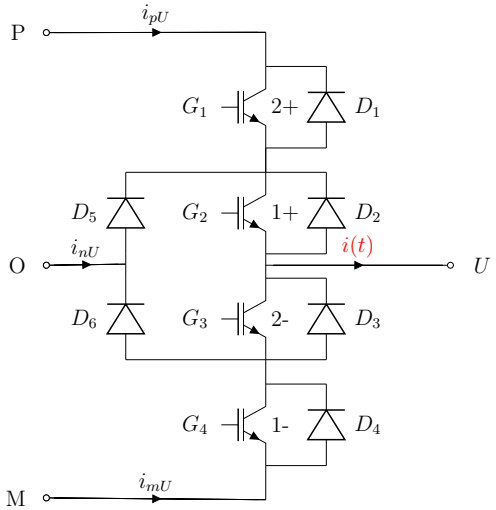


FIGURE 2.16 – Bras de la phase U d'un convertisseur de type NPC.

Nous verrons par la suite que les deux convertisseurs ont des fonctionnements très similaires sauf que les interrupteurs du convertisseur de type NPP subissent moins de contrainte en tension. Le convertisseur triphasé est modélisé par des fonctions combinatoires dont les entrées logiques sont :

- les signaux de commande des interrupteurs ;
- les signes des trois courants de chaque phase de la charge.

Chaque interrupteur commandé est composé d'un IGBT incluant une diode en anti-parallèle. Les niveaux de tension en fonction des signaux de gâchettes pour des onduleurs de

type NPC ou NPP sont consignés dans la table 2.1.

$$G_i = \begin{cases} 1 & \textit{L'interrupteur correspondant est commandé à la fermeture} \\ 0 & \textit{L'interrupteur n'est pas commandé} \end{cases} \tag{2.3}$$

Quand l'interrupteur n'est pas commandé, il peut être fermé ou ouvert selon l'état de la diode.

TABLE 2.1 – Niveaux de tension de la phase U en fonction du courant et des signaux de gâchette

Config	[G_1 G_2 G_3 G_4]	signe(i)	i_{pU}		i_{mU}		i_{nU}		Composants fermés		V_{UO}	
			NPC	NPP	NPC	NPP	NPC	NPP	NPC	NPP	NPC	NPP
0	[0 0 0 0]	+	0	0	I	I	0	0	D3, D4	D4	V_{MO}	V_{MO}
		-	I	I	0	0	0	0	D1, D2	D1	V_{PO}	V_{PO}
1	[0 0 0 1]	+	0	0	I	I	0	0	D3, D4	D4	V_{MO}	V_{MO}
		-	I	0	0	I	0	0	D1, D2	2-, 1-	V_{PO}	V_{MO}
2	[0 0 1 0]	+	0	0	I	I	0	0	D3, D4	D4	V_{MO}	V_{MO}
		-	0	0	0	0	I	I	2- , D6	Tc-, D2	*0*	*0*
3	[0 0 1 1]	+	0	0	I	I	0	0	D3, D4	D4	V_{MO}	V_{MO}
		-	0	0	I	I	0	0	2-, 1-	2-, 1-	V_{MO}	V_{MO}
4	[0 1 0 0]	+	0	0	0	0	I	I	1+, D5	Tc+, D3	*0*	*0*
		-	I	I	0	0	0	0	D1, D2	D1	V_{PO}	V_{PO}
5	[0 1 0 1]	+	0		0		I		1+, D5		*0*	
		-	I		0		0		D1, D2		V_{PO}	
6	[0 1 1 0]	+	0	0	0	0	I	I	1+, D5	Tc+, D3	*0*	*0*
		-	0	0	0	0	I	I	2-, D6	Tc-, D2	*0*	*0*
7	[0 1 1 1]											
8	[1 0 0 0]	+	0	I	I	0	0	0	D3, D4	2+, 1+	V_{MO}	V_{PO}
		-	I	I	0	0	0	0	D1, D2	D1	V_{PO}	V_{PO}
9	[1 0 0 1]	+	0		I		0		D3, D4		V_{MO}	
		-	I		0		0		D1, D2		V_{PO}	
10	[1 0 1 0]	+	0		I		0		D3, D4		V_{MO}	
		-	0		0		I		2-, D6		*0*	
11	[1 0 1 1]	+	0		I		0		D3, D4		V_{MO}	
		-	0		I		0		2-, 1-		V_{MO}	
12	[1 1 0 0]	+	I	I	0	0	0	0	2+, 1+	2+, 1+	V_{PO}	V_{PO}
		-	I	I	0	0	0	0	D1, D2	*D1*	V_{PO}	V_{PO}
13	[1 1 0 1]	+	I		0		0		2+, 1+		V_{PO}	
		-	I		0		0		D1, D2		V_{PO}	
14	[1 1 1 0]											
15	[1 1 1 1]											

Légende de la table 2.1 :
- Configurations de fonctionnement principal
- Configurations utilisées pendant les temps morts
- Courts-circuits

Les courants totaux i_p, i_m et i_n du bus provenant des trois phases U, V et W sont les sommes des courants du bus provenant de chaque phase :

$$i_p = i_{pU} + i_{pV} + i_{pW} \tag{2.4}$$

$$i_m = i_{mU} + i_{mV} + i_{mW} \tag{2.5}$$

$$i_n = i_{nU} + i_{nV} + i_{nW} \tag{2.6}$$

Les configurations prévues en fonctionnement normal d'un NPP ou d'un NPC sont les configurations fonctionnelles 0,2,3,4,6,12. Les configurations 2 et 4 sont prévues pour la gestion des temps morts car en réalité les interrupteurs commandés ne se ferment et ne s'ouvrent pas instantanément. En effet, des temps morts sont insérés lor du passage d'un niveau de tension à un autre (Par exemple, le passage de la configuration 3 à la configuration 6 transite pendant 20µs à 30µs par la configuration 2). Les fonctionnements des convertisseurs de type NPP et NPC sont tout à fait identiques d'un point de vue macroscopique si on regarde les relations entre niveaux de tension, les combinaisons de signaux de gâchette et le courant pour les configurations précédemment citées.

Toutes les configurations restantes (1, 5, 7, 8, 9, 10, 11, 13, 14 et 15) doivent être signalées en cas d'apparition car elles engendrent soit des surtensions sur les interrupteurs commandés, soit des courts-circuits.

Les niveaux de tension sont définis en fonction des combinaisons de signaux de gâchette et du signe du courant. Cependant, le cas où le courant s'annule est litigieux dans certaines configurations comme la configuration 0 et les configurations de temps morts 2 et 4. Dans ces conditions (courant nul et configurations litigieuses), le niveau de tension dépend de la charge du convertisseur. La tension dépend plus précisément de la force électromotrice de la charge. Dans notre cas d'application, cette force électromotrice correspond à la force électromotrice de la machine.

Comme il s'agit d'un calcul temps réel, nous n'allons pas effectuer des itérations successives pour déterminer les instants d'annulation de courant. En l'absence de signaux de commande, ces instants sont susceptibles de changer la topologie de l'onduleur et donc de changer le niveau de tension appliqué à la machine.

On détermine ces instants par interpolation linéaire Le Doeuff et Benkhoris [1995] en fonction des deux valeurs consécutives échantillonnées du courant, de signes opposés, comme illustré par la Figure 2.17.

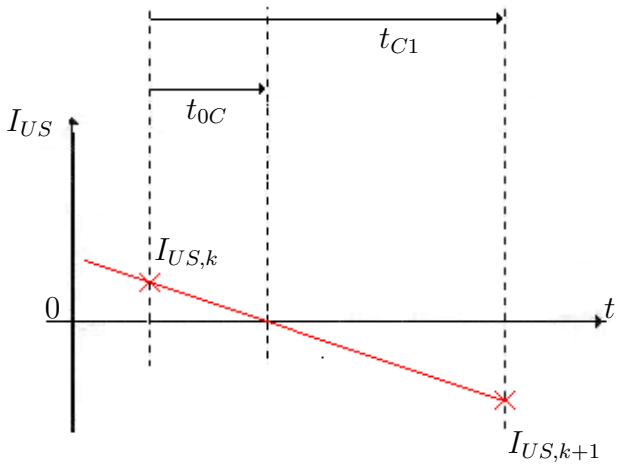


FIGURE 2.17 – Détermination de l'annulation du courant par linéarisation.

L'instant t_{0C} d'annulation du courant est alors :

$$t_{0C} = t_{C1} \left| \frac{I_{US,k}}{I_{US,k} - I_{US,k+1}} \right| \tag{2.7}$$

Une fois l'instant d'annulation du courant déterminé, on procède comme exposé à la section 2.3.1 et illustré par la Figure 2.10. On effectue un pas de temps de t_{0C}. Comme le courant

calculé n'est pas absolument nul du fait de l'approximation, on recorrige les courants restants de façon à reprendre les calculs suivants avec des conditions initiales corrigées.

On procède de la même façon pour déterminer l'instant de conduction des diodes dans le cas où le convertisseur n'est pas commandé. Dans ce cas, le convertisseur de type NPP ou NPC se comporte comme un pont redresseur à diode triphasé classique. Les diodes se ferment quand les tensions entre phase excèdent la tension du bus continu et s'ouvrent à l'annulation du courant. Cet aspect séquentiel des diodes est pris en compte dans la modélisation en mémorisant l'état précédant du convertisseur :

$$\begin{cases} \textit{Si la diode était fermée, on surveille l'annulation du courant;} \\ \textit{Si la diode était ouverte, on surveille la tension entre ses bornes.} \end{cases}$$

Nous venons d'étudier la modélisation du convertisseur pour notre exemple d'étude. Nous allons maintenant modéliser la machine asynchrone alimentée par ce convertisseur.

2.5.2 Modèle de la machine asynchrone

Considérons le cas d'une machine asynchrone à rotor court-circuité et à neutre du stator non-relié. Nous reprenons le modèle dont les variables d'état sont les courants statoriques et les flux rotoriques lorsque le repère $(DQ0)$ est confondu avec le repère stationnaire $(\alpha\beta 0)$ lié au stator $(\dot{\theta}_S = 0)$. D'autres choix de variables d'état conduiraient au même résultat car les matrices de dynamique de deux modèles équivalents ont les mêmes valeurs propres (voir (1.14) du chapitre 1).

Le modèle de départ est celui qui est défini par les équations (D.64), (D.64) et (D.66) (Annexe D). Suite aux différents choix décrits au paragraphe précédent, on obtient le modèle :

- en imposant $\dot{\theta}_S = 0$ dans l'expression de $\overline{A_{mas}}$ (repère lié au stator fixe) ;
- en supprimant les dynamiques de I_{0S} et I_{0R}, c'est à dire les lignes 3 et 6 de $\overline{A_{mas}}$ et $\overline{B_{mas}}$ (neutre non relié) ;
- en éliminant les colonnes 3 et 4 de $\overline{B_{mas}}$ (rotor court-circuité).

Nous obtenons alors :

$$\underbrace{\frac{d}{dt}\begin{pmatrix} I_{DQS} \\ \phi_{DQR} \end{pmatrix}}_{\frac{d}{dt}X_{simp}} = \underbrace{\begin{pmatrix} -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & 0 & \frac{m_{SR}R_R}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}\dot{\theta}_R}{-l_{CR}l_{CS} + {m_{SR}}^2} \\ 0 & -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & -\frac{m_{SR}\theta_R}{-l_{CR}l_{CS} + {m_{SR}}^2} & \frac{m_{SR}R_R}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} \\ \frac{m_{SR}R_R}{l_{CR}} & 0 & -\frac{R_R}{l_{CR}} & \dot{\theta}_R \\ 0 & \frac{m_{SR}R_R}{l_{CR}} & -\dot{\theta}_R & -\frac{R_R}{l_{CR}} \end{pmatrix}}_{A_{simp}}$$

$$\underbrace{\begin{pmatrix} I_{DQS} \\ \phi_{DQR} \end{pmatrix}}_{X_{simp}} + \underbrace{\begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{B_{simp}} \underbrace{V_{DQS}}_{U_{simp}} \tag{2.8}$$

2.5.2.1 Discrétisation de la partie électrique de la machine asynchrone

2.5.2.1.1 Discrétisation exacte du modèle de la partie électrique de la machine asynchrone

Nous ne prétendons pas développer un modèle exact incluant la partie mécanique. Néanmoins, sur un pas de calcul, on peut considérer que la vitesse de rotation mécanique est constante. Les constantes de temps mécaniques sont largement plus grandes que les celles de la partie

électrique. Le modèle de la machine asynchrone devient donc linéaire si la vitesse est localement constante. Comme présenté au premier chapitre, le modèle linéaire peut donc être discrétisé :

- soit par des méthodes d'intégrations numériques ou autres approximations ;
- soit par la discrétisation exacte.

Nous présentons dans cette section une approche discrète exacte de modélisation de la machine asynchrone, en supposant que la vitesse de rotation est constante pendant le pas de calcul. La première étape consiste à déterminer les valeurs propres de la matrice de dynamique. Comme décrit à la section 1.3.2.2.2 , la détermination de ses valeurs propres permettra de calculer l'exponentielle et par ailleurs l'inverse de la matrice de dynamique en utilisant le théorème de Cayley-Hamilton.

Le calcul des valeurs propres d'une matrice d'ordre 4 nécessite généralement la résolution d'une équation polynomiale d'ordre 4. Une telle résolution se fait par la méthode de *Ferrari* (polynôme d'ordre 4) qui fait appel à la méthode de Cardan (polynôme d'ordre 3). Nous proposons une réduction de l'ordre du système en utilisant des variables d'état complexes.

Modèle d'état complexe réduit Cette représentation utilise uniquement des coefficients réels. On peut ainsi réduire l'ordre du système en passant par une représentation complexe (Figure 2.18).

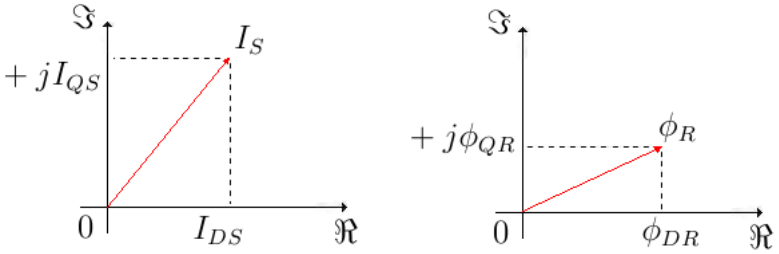


FIGURE 2.18 – Composition de variables d'état complexes.

On définit les variables d'état complexes I_S et ϕ_R comme suit :

$$I_S = I_{DS} + jI_{QS} \Leftrightarrow \Re(I_S) = I_{DS} \text{ et } \Im(I_S) = I_{QS} \tag{2.9}$$

$$\phi_R = \phi_{DR} + j\phi_{QR} \Leftrightarrow \Re(\phi_R) = \phi_{DR} \text{ et } \Im(\phi_R) = \phi_{QR} \tag{2.10}$$

N'importe quelle autre combinaison complexe des variables d'état réelles est mathématiquement possible et pourrait être utilisée. Mais une combinaison entre flux rotorique et courant statorique n'aurait aucun sens physique. Nous garderons donc la composition complexe illustrée par la Figure 2.18.

Pour obtenir la forme d'état complexe, on utilise la matrice de transformation $P_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)}$ décrite comme suit :

$$P_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)} = \begin{pmatrix} 1 & j & 0 & 0 \\ 0 & 0 & 1 & j \\ 1 & -j & 0 & 0 \\ 0 & 0 & 1 & -j \end{pmatrix} \tag{2.11}$$

avec son inverse :

$$P^{-1}_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)} = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ -\frac{j}{2} & 0 & \frac{j}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & -\frac{j}{2} & 0 & \frac{j}{2} \end{pmatrix} \tag{2.12}$$

d'où l'on établit :

$$\underbrace{\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix}}_{X_{comp}} = P_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)} \underbrace{\begin{pmatrix} I_{DS} \\ I_{QS} \\ \phi_{DR} \\ \phi_{QR} \end{pmatrix}}_{X_{simp}} \tag{2.13}$$

On définit ici les variables d'état conjuguées :
I_S^* est le conjugué de la variable d'état complexe I_S ;
ϕ_S^* est le conjugué de la variable d'état complexe ϕ_S.

La nouvelle représentation d'état s'écrit alors :

$$\frac{d}{dt}X_{comp} = A_{comp}X_{comp} + B_{comp}V_{DQS} \tag{2.14}$$

On calcule la matrice de dynamique complexe A_{comp} à partir de l'équation (1.12) (page 12) :

$$\begin{aligned} A_{comp} =& P_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)} A_{simp} P^{-1}_{(I_{DQS},\phi_{DQR})} \\ =& \begin{pmatrix} -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}(R_R + j l_{CR}\dot\theta_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} & 0 & 0 \\ \frac{m_{SR}R_R}{l_{CR}} & -\frac{R_R}{l_{CR}} - j\dot\theta_R & 0 & 0 \\ 0 & 0 & -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}(R_R - j l_{CR}\dot\theta_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} \\ 0 & 0 & \frac{m_{SR}R_R}{l_{CR}} & -\frac{R_R}{l_{CR}} + j\dot\theta_R \end{pmatrix} \end{aligned} \tag{2.15}$$

On écrit :

$$A_{comp} = \begin{pmatrix} A_* & 0_{2\times2} \\ 0_{2\times2} & A_*^* \end{pmatrix} \tag{2.16}$$

avec :

$$A_* = \begin{pmatrix} -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}(R_R + j l_{CR}\dot\theta_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} \\ \frac{m_{SR}R_R}{l_{CR}} & -\frac{R_R}{l_{CR}} - j\dot\theta_R \end{pmatrix} \tag{2.17}$$

On note A_*^* le conjugué de A_*.

La matrice de commande complexe B_{comp} est la suivante :

$$\begin{aligned} B_{comp} =& P_{(I_{DQS},\phi_{DQR})\to(I_S,\phi_R)} B_{simp} \\ =& \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} & \frac{j l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} \\ 0 & 0 \\ \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} & -\frac{j l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} \\ 0 & 0 \end{pmatrix} \end{aligned} \tag{2.18}$$

On écrit alors :

$$\frac{d}{dt}\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} = \begin{pmatrix} A_* & 0_{2\times2} \\ 0_{2\times2} & A_*^* \end{pmatrix} \begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} + \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} & \frac{j l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} \\ 0 & 0 \\ \frac{l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} & -\frac{j l_{CR}}{l_{CR}l_{CS} - {m_{SR}}^2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_{DS} \\ V_{QS} \end{pmatrix} \tag{2.19}$$

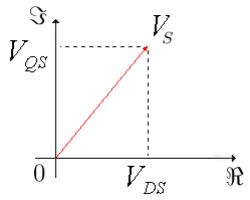


FIGURE 2.19 – Composition d'une entrée tension complexe V_{DQS}.

De la même façon que l'on a composé les variables d'état complexes, nous composons l'entrée de commande complexe V_{DQS} comme indiqué sur la Figure 2.19, on obtient :

$$\frac{d}{dt}\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} = \begin{pmatrix} A_* & 0_{2\times2} \\ 0_{2\times2} & A_*^* \end{pmatrix}\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} + \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} \\ 0 \\ \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} \\ 0 \end{pmatrix}\begin{pmatrix} V_{DQS} \\ V_{DQS}^* \end{pmatrix} \tag{2.20}$$

De même que la demi-matrice de dynamique complexe, en utilisant l'entrée de commande complexe V_{DQS}, la demi-matrice de commande complexe B_* peut s'écrire :

$$B_* = \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} \\ 0 \end{pmatrix} \tag{2.21}$$

d'où :

$$\frac{d}{dt}\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} = \begin{pmatrix} A_* & 0_{2\times2} \\ 0_{2\times2} & A_*^* \end{pmatrix}\begin{pmatrix} I_S \\ \phi_R \\ I_S^* \\ \phi_R^* \end{pmatrix} + \begin{pmatrix} B_* \\ B_*^* \end{pmatrix}\begin{pmatrix} V_{DQS} \\ V_{DQS}^* \end{pmatrix} \tag{2.22}$$

avec B_*^* est le conjugué de B_*

L'équation (2.20) montre que la dynamique des variables d'état complexes est indépendante de la dynamique de leurs conjugués. Dès lors, on peut les séparer en deux représentations d'état et les deux représentations obtenues sont par conjugaison exactement les mêmes.
Un demi-modèle contient donc toutes les informations nécessaires sur les axes direct et en quadrature. On retrouve les grandeurs réelles sur ces deux axes en récupérant les parties réelles et imaginaires des variables d'état complexes. En plus, nous passons d'une représentation d'état d'ordre 4 à coefficients réels à une représentation d'ordre 2 à coefficients complexes. Un système d'équation d'ordre 2 est beaucoup plus facile à résoudre et résolvable avec des notions mathématiques moins compliquées.

L'équation d'état continue complexe réduite est alors :

$$\frac{d}{dt}\begin{pmatrix} I_S \\ \phi_R \end{pmatrix} = A_*\begin{pmatrix} I_S \\ \phi_R \end{pmatrix} + B_* V_{DQS} \tag{2.23}$$

Il faut alors trouver les valeurs propres de la demi-matrice A_* pour trouver ensuite la matrice de dynamique discrète $A_{*d} = e^{A_* t}$. On déterminera ensuite la matrice de commande B_{*d}.

Nous allons calculer l'exponentielle de A_* et en déduire le modèle discrèt exact de la machine asynchrone. Pour cela, on commence par déterminer l'équation caractéristique de cette matrice.

Équation caractéristique de la demi matrice complexe A_*

Valeurs propres de la matrice A_*
Les valeurs propres de A_* sont les solutions de :

$$f(\lambda) = Det\,(A_* - \lambda I) = \frac{1}{a}(a\lambda^2 + b\lambda + c) = 0 \tag{2.24}$$

On note :

$$a = l_{CR}l_{CS} - m_{SR}^2$$
$$b = l_{CS}R_R + l_{CR}R_S + j\dot{\theta}_R(-m_{SR}^2 + l_{CR}l_{CS})$$
$$c = R_R R_S + j\dot{\theta}_R(l_{CR}l_{CS} - m_{SR}^2)$$

$$\triangle = b^2 - 4ac \tag{2.25}$$

Cas où $\triangle \neq 0$:

$$\Re(|\triangle|) = -\Im(b)^2 + \Re(b)^2 - 4a\Re(c) \tag{2.26}$$

$$\Im(|\triangle|) = -4\Im(c)a + 2\Im(b)\Re(b) \tag{2.27}$$

$$\triangle = |\triangle|e^{j\theta_\triangle} \tag{2.28}$$

$$|\triangle| = \sqrt{\Re(|\triangle|)^2 + \Im(|\triangle|)^2} \tag{2.29}$$

$$\sin(\theta_\triangle) = \frac{\Im(|\triangle|)}{|\triangle|} \tag{2.30}$$

$$\cos(\theta_\triangle) = \frac{\Re(|\triangle|)}{|\triangle|} \tag{2.31}$$

Nous allons chercher un (il peut en exister deux mais nous avons besoins d'un seul) nombre complexe $\delta = |\delta|e^{j\theta_\delta}$ vérifiant :

$$\delta^2 = \triangle \tag{2.32}$$

$$\delta^2 = |\delta|^2 e^{j2\theta_\delta} = |\triangle|e^{j\theta_\triangle} = \triangle \tag{2.33}$$

De l'équation précédente, nous déduisons les deux équations qui suivent :

$$|\delta|^2 = |\triangle| \Leftrightarrow |\delta| = \sqrt{|\triangle|} \tag{2.34}$$

et

$$\cos(2\theta_\delta) + j\sin(2\theta_\delta) = \cos(\theta_\triangle) + j\sin(\theta_\triangle) \tag{2.35}$$

En identifiant les parties imaginaire et réelle de chaque côté de l'équation (2.35), on déduit les deux équations suivantes :

$$\cos(2\theta_\delta) = \cos^2(\theta_\delta) - \sin^2(\theta_\delta) = -1 + 2\cos^2(\theta_\delta) = 1 - 2\sin^2(\theta_\delta) = \cos(\theta_\triangle) \tag{2.36}$$

$$\sin(2\theta_\delta) = 2\cos(\theta_\delta)\sin(\theta_\delta) = \sin(\theta_\triangle) \tag{2.37}$$

De l'équation (2.36), nous déduisons les valeurs *probables* de $\cos(\theta_\delta)$ et $\sin(\theta_\delta)$:

$$\cos(\theta_\delta) = \pm\sqrt{\frac{1}{2}(1+\cos(\theta_\triangle))} = \pm\sqrt{\frac{1}{2}\left(1+\frac{\Re(|\triangle|)}{|\triangle|}\right)} \tag{2.38}$$

$$\sin(\theta_\delta) = \pm\sqrt{\frac{1}{2}(1-\cos(\theta_\triangle))} = \pm\sqrt{\frac{1}{2}\left(1-\frac{\Re(|\triangle|)}{|\triangle|}\right)} \tag{2.39}$$

L'équation (2.37) donne 4 doublons $\{\cos(\theta_\triangle, \sin(\theta_\delta))\}$. Pour identifier les doublons à conserver, on utilise le signe du produit de $\cos(\theta_\delta)$ et $\sin(\theta_\delta)$ ($\sin(\theta_\triangle)$ étant préalablement calculé). Il existe deux doublons $\{\cos(\theta_\triangle, \sin(\theta_\delta))\}$ qui vérifient l'équation (2.37). Nous allons choisir le doublon ayant :

$$\cos(\theta_\delta) = \sqrt{\frac{1}{2}\left(1+\frac{\Re(|\triangle|)}{|\triangle|}\right)} \geq 0 \tag{2.40}$$

Le doublon $\{\cos(\theta_\triangle, \sin(\theta_\delta))\}$ ayant $\cos(\theta_\delta) = \sqrt{\frac{1}{2}\left(1+\frac{\Re(|\triangle|)}{|\triangle|}\right)} \leq 0$ et qui satisfait la ségrégation par le signe ((2.37)) conduira au même résultat final.

$$\sin(\theta_\delta) = \begin{cases} \sqrt{\frac{1}{2}\left(1-\frac{\Re(|\triangle|)}{|\triangle|}\right)} & \text{si } \Im(\triangle) > 0 \quad (\sin(\theta_\triangle > 0) \\ -\sqrt{\frac{1}{2}\left(1-\frac{\Re(|\triangle|)}{|\triangle|}\right)} & \text{si } \Im(\triangle) \leq 0 \quad (\sin(\theta_\triangle \leq 0) \end{cases} \tag{2.41}$$

Les parties réelles et imaginaires de δ sont alors :

$$\Re(\delta) = \sqrt{\frac{|\triangle|}{2}\left(1+\frac{\Re(|\triangle|)}{|\triangle|}\right)} \tag{2.42}$$

$$\Im(\delta) = \text{signe}(\Im(\triangle))\sqrt{\frac{|\triangle|}{2}\left(1-\frac{\Re(|\triangle|)}{|\triangle|}\right)} \tag{2.43}$$

Les parties réelles et imaginaires des valeurs propres λ_0 et λ_1 sont :

$$\Re(\lambda_0) = \frac{-\Re(b)-\Re(\delta)}{2a} \tag{2.44}$$

$$\Im(\lambda_0) = \frac{-\Im(b)-\Im(\delta)}{2a} \tag{2.45}$$

$$\Re(\lambda_1) = \frac{-\Re(b)+\Re(\delta)}{2a} \tag{2.46}$$

$$\Im(\lambda_1) = \frac{-\Im(b)+\Im(\delta)}{2a} \tag{2.47}$$

Cas où $\triangle = 0$: λ_0 et λ_1 sont égaux dans le cas où :

$$\Re(\delta) = \Im(\delta) = 0 \Leftrightarrow \delta = 0 \Leftrightarrow \triangle = 0 \tag{2.48}$$

Dans ce cas :

$$\lambda_0 = \lambda_1 = -\frac{b}{2a} \tag{2.49}$$

$$\Re(\lambda_0) = \Re(\lambda_1) = -\frac{\Re(b)}{2a} \tag{2.50}$$

$$\Im(\lambda_0) = \Im(\lambda_1) = -\frac{\Im(b)}{2a} \tag{2.51}$$

La Figure 2.20 et la Figure 2.21 présentent les tracés des parties réelles et des parties imaginaires des deux valeurs propres λ_0 et λ_1 pour les paramètres électriques suivants :

TABLE 2.2 – Exemple de paramètres machine

Paramètres	**Valeurs**
L_{CS}	$72.32277mH$
L_{CR}	$70.66819mH$
m_{SR}	$69.0924mH$
R_S	0.034Ω
R_R	0.015Ω
p	6
f_n	$13.13Hz$
$\dot{\theta}_{Rnom}$	$2 \times \pi \times \frac{13.13}{6} rad.s^{-1} = 13.75 rad.s^{-1}$

f_n et $\dot{\theta}_{Rnom}$ sont respectivement la fréquence et la pulsation de rotation mécanique nominale de la machine. Cette machine est une machine à faible vitesse et donc à nombre de pôles important (machine type pour les applications marines de l'entreprise GE-PC). Les machines de bateau sont à faible vitesse pour éviter la cavitation. Les résistances statoriques et rotoriques sont les résistances de bobinage mesurées à la température de fonctionnement.
Les valeurs propres ont des parties réelles négatives sur toute la plage de vitesse de fonctionnement et la machine alimentée par des tensions statoriques sinusoïdales à fréquence et amplitude constante est stable.

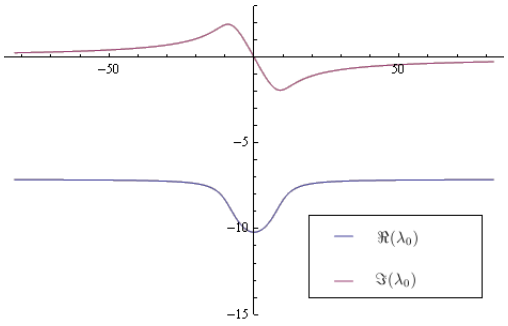


FIGURE 2.20 – Parties réelle et imaginaire de λ_0.

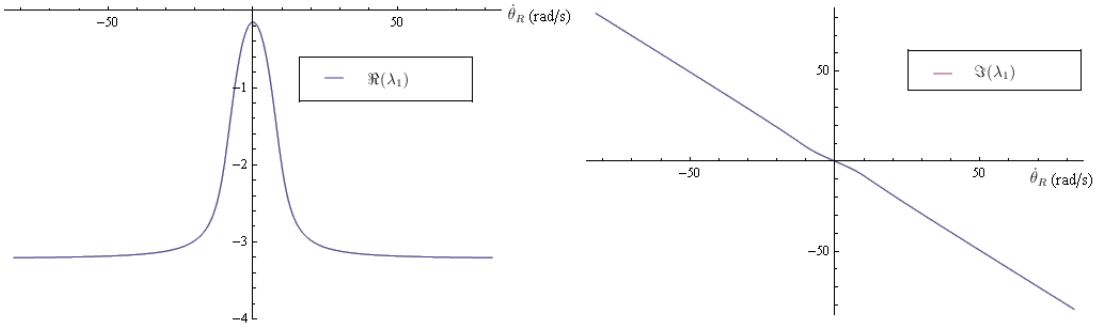


FIGURE 2.21 – Parties réelle et imaginaire de λ_1.

Exponentielle de la matrice A_*

Cas où $\triangle \neq 0$ **:** Dans ce cas, comme vu précédemment, les deux valeurs propres sont distinctes. On utilise alors l'équation générique (1.79) du premier chapitre :

$$\begin{pmatrix} e^{\lambda_0 t} \\ e^{\lambda_1 t} \end{pmatrix} = \begin{pmatrix} \sum_{k=0}^{1} \alpha_k \lambda_0^k \\ \sum_{k=0}^{1} \alpha_k \lambda_1^k \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \lambda_0 \\ 1 & \lambda_1 \end{pmatrix}}_{M_{Vandermonde}} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \tag{2.52}$$

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \underbrace{\frac{1}{(\lambda_1 - \lambda_0)} \begin{pmatrix} \lambda_1 & -\lambda_0 \\ -1 & 1 \end{pmatrix}}_{M^{-1}_{Vandermonde}} \begin{pmatrix} e^{\lambda_0 t} \\ e^{\lambda_1 t} \end{pmatrix} \tag{2.53}$$

$$\Re\left(\frac{1}{\lambda_1 - \lambda_0}\right) = \frac{\Re(\lambda_1) - \Re(\lambda_0)}{\Im(\lambda_1)^2 - 2\Im(\lambda_0)\Im(\lambda_1) + \Im(\lambda_0)^2 + (\Re(\lambda_1) - \Re(\lambda_0))^2}$$

$$\Im\left(\frac{1}{\lambda_1 - \lambda_0}\right) = -\frac{\Im(\lambda_1) - \Im(\lambda_0)}{\Im(\lambda_1)^2 - 2\Im(\lambda_0)\Im(\lambda_1) + \Im(\lambda_0)^2 + (\Re(\lambda_1) - \Re(\lambda_0))^2}$$

$$\Re\Big(\lambda_1 e^{\lambda_0 t}\Big) = e^{\Re(\lambda_0)t}\Big(\Re(\lambda_1)\cos\left(\Im(\lambda_0)t\right) - \Im(\lambda_1)\sin\left(\Im(\lambda_0)t\right)\Big)$$

$$\Im\Big(\lambda_1 e^{\lambda_0 t}\Big) = e^{\Re(\lambda_0)t}\Big(\Im(\lambda_1)\cos\left(\Im(\lambda_0)t\right) + \Re(\lambda_1)\sin\left(\Im(\lambda_0)t\right)\Big)$$

$$\Re\Big(\lambda_0 e^{\lambda_1 t}\Big) = e^{\Re(\lambda_1)t}\Big(\Re(\lambda_0)\cos\left(\Im(\lambda_1)t\right) - \Im(\lambda_0)\sin\left(\Im(\lambda_1)t\right)\Big)$$

$$\Im\Big(\lambda_0 e^{\lambda_1 t}\Big) = e^{\Re(\lambda_1)t}\Big(\Im(\lambda_0)\cos\left(\Im(\lambda_1)t\right) + \Re(\lambda_0)\sin\left(\Im(\lambda_1)t\right)\Big)$$

$$\Re\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) = \Re\Big(\lambda_1 e^{\lambda_0 t}\Big) - \Re\Big(\lambda_0 e^{\lambda_1 t}\Big)$$

$$\Im\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) = \Im\Big(\lambda_1 e^{\lambda_0 t}\Big) - \Im\Big(\lambda_0 e^{\lambda_1 t}\Big)$$

$$\Re\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) = e^{\Re(\lambda_1)t}\cos\left(\Im(\lambda_1)t\right) - e^{\Re(\lambda_0)t}\cos\left(\Im(\lambda_0)t\right)$$

$$\Im\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) = e^{\Re(\lambda_1)t}\sin\left(\Im(\lambda_1)t\right) - e^{\Re(\lambda_0)t}\sin\left(\Im(\lambda_0)t\right)$$

Finalement, les parties réelles et imaginaires des coefficients α_0 et α_1 permettant de calculer l'exponentielle de A_* sont définies comme suit :

$$\Re(\alpha_0) = \Re\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Re\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) - \Im\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Im\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) \tag{2.54}$$

$$\Im(\alpha_0) = \Re\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Im\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) + \Im\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Re\Big(\lambda_1 e^{\lambda_0 t} - \lambda_0 e^{\lambda_1 t}\Big) \tag{2.55}$$

$$\Re(\alpha_1) = \Re\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Re\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) - \Im\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Im\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) \tag{2.56}$$

$$\Im(\alpha_1) = \Re\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Im\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) + \Im\left(\frac{1}{\lambda_1 - \lambda_0}\right)\Re\Big(e^{\lambda_1 t} - e^{\lambda_0 t}\Big) \tag{2.57}$$

Cas où $\triangle = 0$ **:**

Dans ce cas, le polynôme caractéristique $f(\lambda)$, défini par l'équation (2.24), admet une racine double λ_0 définie par l'équation (2.49). Remarquons que le terme $\frac{1}{\lambda_1 - \lambda_0}$ n'est pas défini pour une valeur propre de multiplicité 2. Comme vu au premier chapitre, on utilise dans le cas d'une ou plusieurs valeurs propres multiples, l'équation (1.82) pour calculer les coefficients α_0 et α_1. On obtient :

$$\begin{pmatrix} e^{\lambda_0 t} \\ te^{\lambda_0 t} \end{pmatrix} = \begin{pmatrix} 1 & \lambda_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \tag{2.58}$$

$$\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix} = \begin{pmatrix}1 & -\lambda_0\\ 0 & 1\end{pmatrix}\begin{pmatrix}e^{\lambda_0 t}\\ te^{\lambda_0 t}\end{pmatrix} = \begin{pmatrix}(1-\lambda_0 t)e^{\lambda_0 t}\\ te^{\lambda_0 t}\end{pmatrix} \tag{2.59}$$

$$\Re(\alpha_0) = e^{\Re(\lambda_0)t}\Big(\big(1-\Re(\lambda_0)t\big)\cos\left(\Im(\lambda_0)t\right) + \Im(\lambda_0)t\sin\left(\Im(\lambda_0)t\right)\Big) \tag{2.60}$$

$$\Im(\alpha_0) = e^{\Re(\lambda_0)t}\Big(\big(1-\Re(\lambda_0)t\big)\sin\left(\Im(\lambda_0)t\right) - \Im(\lambda_0)t\cos\left(\Im(\lambda_0)t\right)\Big) \tag{2.61}$$

$$\Re(\alpha_1) = te^{\Re(\lambda_0)t}\cos\left(\Im(\lambda_0)t\right) \tag{2.62}$$

$$\Im(\alpha_1) = te^{\Re(\lambda_0)t}\sin\left(\Im(\lambda_0)t\right) \tag{2.63}$$

Nous venons de traiter tous les cas possibles pour calculer α_0 et α_1. Une fois α_0 et α_1 calculés, on utilise l'équation (1.78) pour retrouver $e^{A_* t}$. Chaque élément de A_* est défini dans l'équation (2.17) que nous rappelons ici :

$$A_* = \begin{pmatrix}-\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}(R_R + j l_{CR}\dot\theta_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)}\\ \frac{m_{SR}R_R}{l_{CR}} & -\frac{R_R}{l_{CR}} - j\dot\theta_R\end{pmatrix}$$

La matrice de dynamique discrète complexe réduite, notée A_{*d} s'exprime alors par :

$$\begin{aligned} A_{*d} = e^{A_* t} =& \alpha_0 I + \alpha_1 A_* \\ =& \begin{pmatrix}\Re(\alpha_0) - \Re(\alpha_1)\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}\Big(\Re(\alpha_1)R_R - \Im(\alpha_1) l_{CR}\dot\theta_R\Big)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)}\\ \Re(\alpha_1)\frac{m_{SR}R_R}{l_{CR}} & \Re(\alpha_0) - \Re(\alpha_1)\frac{R_R}{l_{CR}} + \Im(\alpha_1)\dot\theta_R\end{pmatrix} \\ &+ j\begin{pmatrix}\Im(\alpha_0) - \Im(\alpha_1)\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}\Big(\Im(\alpha_1)R_R + \Re(\alpha_1) l_{CR}\dot\theta_R\Big)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)}\\ \Im(\alpha_1)\frac{m_{SR}R_R}{l_{CR}} & \Im(\alpha_0) - \Im(\alpha_1)\frac{R_R}{l_{CR}} - \Re(\alpha_1)\dot\theta_R\end{pmatrix} \end{aligned} \tag{2.64}$$

Pour alléger l'écriture, on peut écrire :

$$A_{*d} = e^{A_* t} = \begin{pmatrix}\Re\Big(A_{*d(1,1)}\Big) + j\Im\Big(A_{*d(1,1)}\Big) & \Re\Big(A_{*d(1,2)}\Big) + j\Im\Big(A_{*d(1,2)}\Big)\\ \Re\Big(A_{*d(2,1)}\Big) + j\Im\Big(A_{*d(1,1)}\Big) & \Re\Big(A_{*d(2,2)}\Big) + j\Im\Big(A_{*d(1,2)}\Big)\end{pmatrix} \tag{2.65}$$

Matrice de commande discrète B_{*d}

Nous allons établir l'expression de la matrice de commande discrète B_{*d}. Comme la machine asynchrone est alimentée par un onduleur de tension, le vecteur d'entrées $U_* = V_{DQS}$ peut être considéré comme constant entre deux commutations.

$$B_{*d} = {A_*}^{-1}(e^{A_* t} - I)B_* = {A_*}^{-1}(A_{*d} - I)B_* \tag{2.66}$$

Calcul de ${A_*}^{-1}$:

On utilisera ici encore le théorème de Cayley-Hamilton pour calculer A_*^{-1}. La matrice A_* satisfait son polynôme caractéristique (chapitre 1 - page 26) défini par l'équation (2.24). On écrit alors :

$$aA_*^2 + bA_* + cI = 0 \Leftrightarrow aA_* + bI + cA_*^{-1} = 0 \tag{2.67}$$

Pour rappel, nous précisons ici que :

$$a = l_{CR}l_{CS} - m_{SR}^2$$

$$b = l_{CS}R_R + l_{CR}R_S + j\dot{\theta}_R(-m_{SR}^2 + l_{CR}l_{CS})$$
$$c = R_R R_S + j\dot{\theta}_R(l_{CR}l_{CS} - m_{SR}^2)$$
$$A_* = \begin{pmatrix} -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} & \frac{m_{SR}(R_R + j l_{CR}\dot{\theta}_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} \\ \frac{m_{SR}R_R}{l_{CR}} & -\frac{R_R}{l_{CR}} - j\dot{\theta}_R \end{pmatrix}$$

On en déduit alors l'expression suivante :

$$A_*^{-1} = -\frac{a}{c}A_* - \frac{b}{c}I \tag{2.68}$$

avec :

$$\Re(-\frac{a}{c}) = -\frac{a\Re(c)}{\Im(c)^2 + \Re(c)^2}$$
$$\Im(-\frac{a}{c}) = \frac{a\Im(c)}{\Im(c)^2 + \Re(c)^2}$$
$$\Re(-\frac{b}{c}) = -\frac{\Im(b)\Im(c)}{\Im(c)^2 + \Re(c)^2} - \frac{\Re(b)\Re(c)}{\Im(c)^2 + \Re(c)^2}$$
$$\Im(-\frac{b}{c}) = \frac{\Im(c)\Re(b)}{\Im(c)^2 + \Re(c)^2} - \frac{\Im(b)\Re(c)}{\Im(c)^2 + \Re(c)^2}$$

d'où finalement :

$$\begin{aligned} A_*^{-1} = &\begin{pmatrix} -\Re(-\frac{a}{c})\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} + \Re(-\frac{b}{c}) & \frac{m_{SR}(\Re(-\frac{a}{c})R_R - \Im(-\frac{a}{c})l_{CR}\dot{\theta}_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} \\ \Re(-\frac{a}{c})\frac{m_{SR}R_R}{l_{CR}} & -\Re(-\frac{a}{c})\frac{R_R}{l_{CR}} + \Im(-\frac{a}{c})\dot{\theta}_R + \Re(-\frac{b}{c}) \end{pmatrix} \\ &+ j\begin{pmatrix} -\Im(-\frac{a}{c})\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR}{m_{SR}}^2} + \Im(-\frac{b}{c}) & \frac{m_{SR}(\Im(-\frac{a}{c})R_R + \Re(-\frac{a}{c})l_{CR}\dot{\theta}_R)}{l_{CR}(l_{CR}l_{CS} - {m_{SR}}^2)} \\ \Im(-\frac{a}{c})\frac{m_{SR}R_R}{l_{CR}} & -\Im(-\frac{a}{c})\frac{R_R}{l_{CR}} - \Re(-\frac{a}{c})\dot{\theta}_R + \Im(-\frac{b}{c}) \end{pmatrix} \end{aligned} \tag{2.69}$$

Pour pouvoir comparer la modélisation qui vient d'être proposée à une approche classique utilisant un solveur numérique, nous allons développer par la suite le modèle discretisé par la méthode trapézoïdale.

2.5.2.1.2 Discrétisation approximative de la partie électrique de la machine asynchrone

Nous allons maintenant utiliser la méthode trapézoïdale pour avoir un modèle discret approximatif.

On recherche d'abord la solution en régime libre X_l de la représentation d'état continue de (1.2) en utilisant les relations de la table 1.1 pour le cas d'une intégration par la méthode trapézoïdale :

$$\dot{X}_l = \frac{dX_l}{dt} = A_* X_l \tag{2.70}$$

$$\frac{2}{T_S}\frac{(1 - z^{-1})}{(1 + z^{-1})}X_{n+1} = A_* X_{n+1} \tag{2.71}$$

$$2X_{n+1} - 2z^{-1}X_{n+1} = A_* T_S X_{n+1} + A_* T_S z^{-1} X_{n+1} \tag{2.72}$$

$$2X_{n+1} - 2X_n = A_* T_S X_{n+1} + A_* T_S z^{-1} X_n \tag{2.73}$$

D'où finalement la solution discrète en régime libre par la méthode trapézoïdale :

$$X_{n+1} = e^{A_* T_S} X_n \simeq A_{*d,trap} X_n \tag{2.74}$$

avec

$$A_{*d,trap} = (2I - T_S A_*)^{-1}(2I + A_* T_S)$$

Les éléments du vecteur d'entrées sont du type échelon si les tensions du bus continu varient très peu entre deux commutations. La solution particulière à des entrées échelons de l'équation d'état (1.2) est traitée en annexe C. On retrouve la solution en imposant $X(0) = 0$ dans l'équation (C.7). Cette solution s'écrit :

$$X_{n+1} = A_*^{-1}(e^{A_* t} - I)BU_{n+1} \simeq A_*^{-1}(A_{*d,trap} - I)B_* U_{n+1} \tag{2.75}$$

Le modèle de la machine asynchrone par la méthode trapézoïdale est obtenu en additionnant la solution en régime libre et la solution particulière. On écrit alors :

$$X_{n+1} \simeq A_{*d,trap} X_n + A_*^{-1}(A_{*d,trap} - I)BU_{n+1} \tag{2.76}$$

2.5.2.2 Comparaison des deux modèles discrets de la partie électrique

Les figures (2.22) et (2.23) présentent le tracé des coefficients de la matrice de dynamique discrète obtenus par la méthode exacte et par la méthode trapézoïdale. Les sous-figures (a) et (b) de chacune de ces figures montrent deux éléments de la matrice discrète sur la plage de vitesse de fonctionnement pour différents pas de calcul. L'erreur diminue avec le pas de calcul T_S utilisé. L'erreur relative varie d'une façon non linéaire par rapport à la vitesse. Un pas de calcul Ts de $200\mu s$ permet d'avoir une erreur relative sur les coefficients inférieure à 1%.

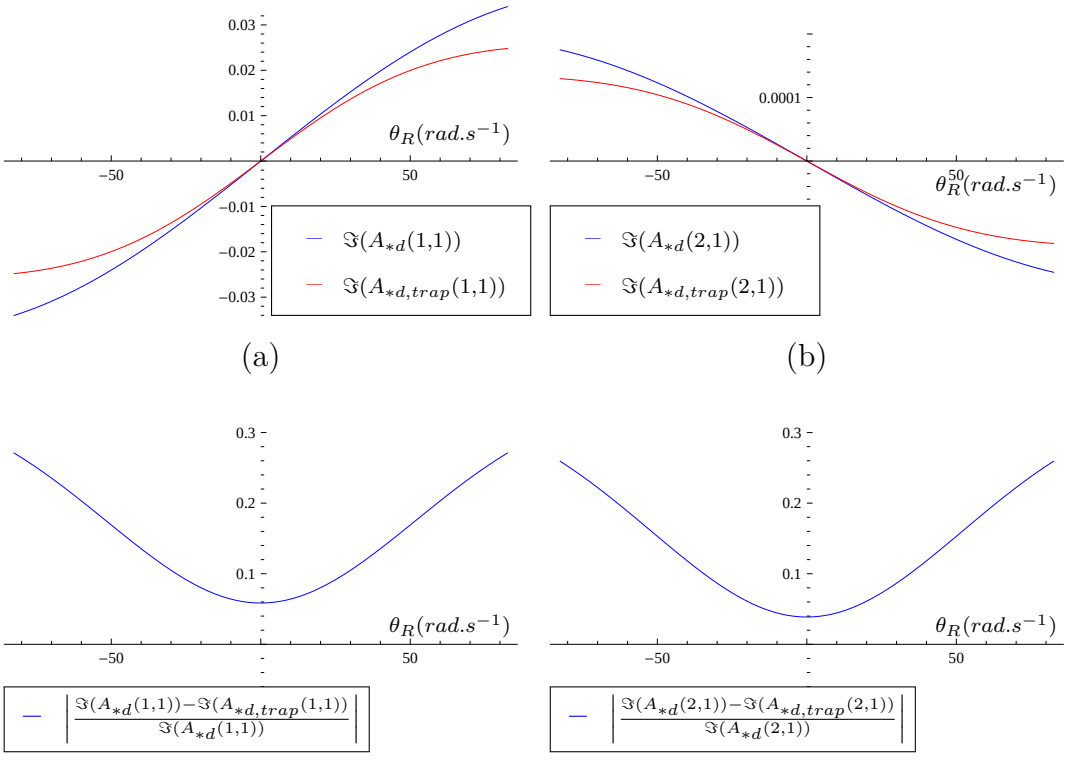


FIGURE 2.22 – Comparaison des éléments de la matrice discrète pour un pas de calcul $T_s = 20ms$.

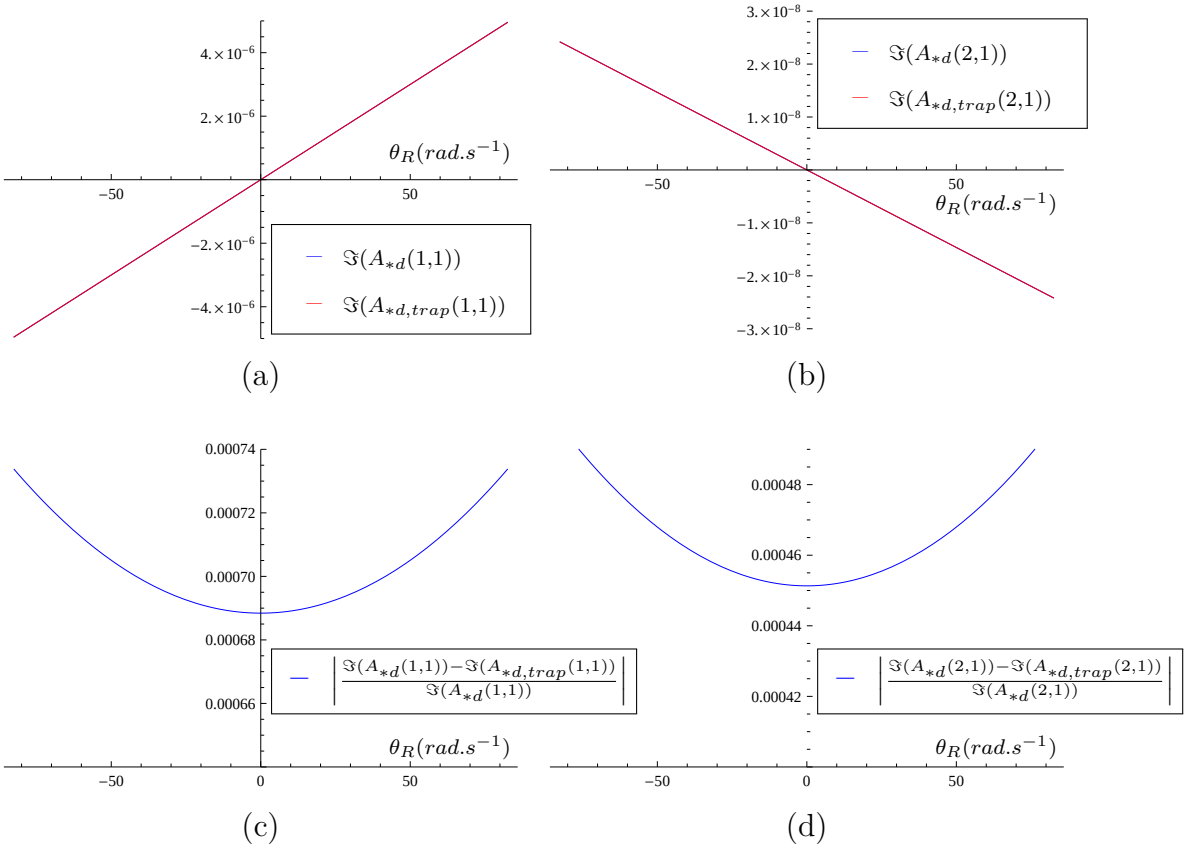


FIGURE 2.23 – Comparaison des éléments de la matrice discrète pour un pas de calcul $T_s = 0.2ms$.

Le modèle continu de la machine asynchrone est stable car les parties réelles des valeurs propres sont négatives sur toute la plage de vitesse de fonctionnement (voir Figure 2.20 et Figure 2.21). Comme la méthode trapézoïdale est A-stable, le modèle discret reste toujours stable car le modèle continu l'était.
La seule source d'instabilité possible serait les erreurs de précision de format lorsque le pas de calcul est trop petit. En effet, lorsque le pas de calcul est trop petit, les points $q_i = T_S\lambda_i$ se rapprochent de l'axe imaginaire qui limite la région de stabilité de la méthode trapézoïdale (voir la section 1.3.1.6.3 du premier chapitre). Une erreur sur les coefficients pourrait légèrement déplacer ces points q_i dans la région instable du demi plan droit.

D'une façon générale, l'erreur augmente avec le pas de calcul. Par conséquent, si l'on utilise le modèle discret approximé par la méthode trapézoïdale, on doit fixer un pas maximum T_{Smax} interne autoris,é comme discuté à la section 2.3.2.2. Cela permettra d'assurer une erreur tolérable au niveau du modèle approximatif. Dans ce cas, T_{Smax} sera un paramètre d'entrée du modèle et pourra être changé en fonction de la dynamique de la machine simulée. Ce problème n'a pas lieu dans le cas d'une modélisation exacte. Nous avions donc choisi d'implémenter le modèle exact de la machine asynchrone avec la méthode d'interpolation évènementielle.

2.5.3 Résultats

Nous étudions le cas du système de puissance illustré précédemment par la Figure 2.14.

Les paramètres de la machine asynchrone sont consignés dans la table 2.3.

TABLE 2.3 – Paramètres de la machine asynchrone.

Paramètres	**Valeurs**
L_{CS}	$72.32277mH$
L_{CR}	$70.66819mH$
m_{SR}	$69.0924mH$
R_S	0.034Ω
R_R	0.015Ω
p	6
f_n	$13.13Hz$
$\dot{\theta}_{Rnom}$	$2 \times \pi \times \frac{13.13}{6} rad.s^{-1} = 13.75 rad.s^{-1}$
J	$59000kg.m^2$
$f_{\dot{\theta}}$	$3923.016Nms$

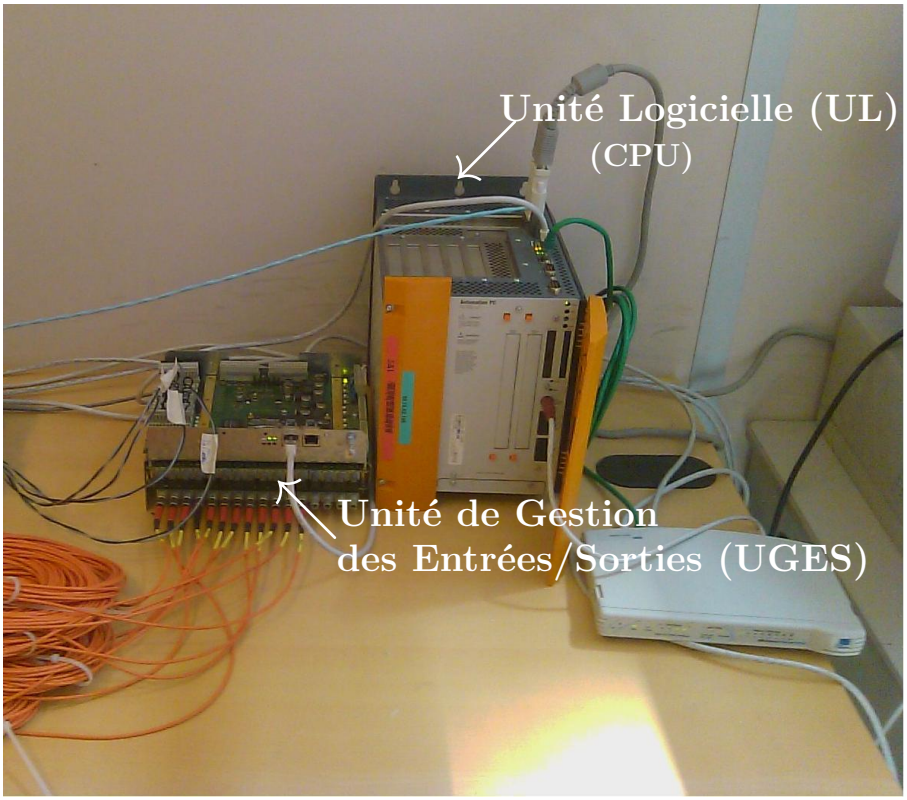


FIGURE 2.24 – Contrôleur testé en VIL.

La Figure 2.24 décrit le matériel dédié au contrôle de l'ensemble convertisseur/machine. L'unité UL est un APC810 - 5 slots PCI - 2.5GHz. L'unité UGES est une carte industrielle propriétaire appelée Power Interface Board EtherCAT (PIBe). Cette carte PIBe intègre un composant FPGA EP2C20F484 de la famille Altera et peut gérer en même temps :

- 8 entrées analogiques ;
- 4 sorties analogiques ;
- 2 cartes filles de 16 entrées/16 sorties optiques, ou 16 entrées/16 sorties pulses électriques ou une carte fille d'estimation de flux. Dans le cas de la Figure 2.24, la PIBe gère une carte pulses optiques.

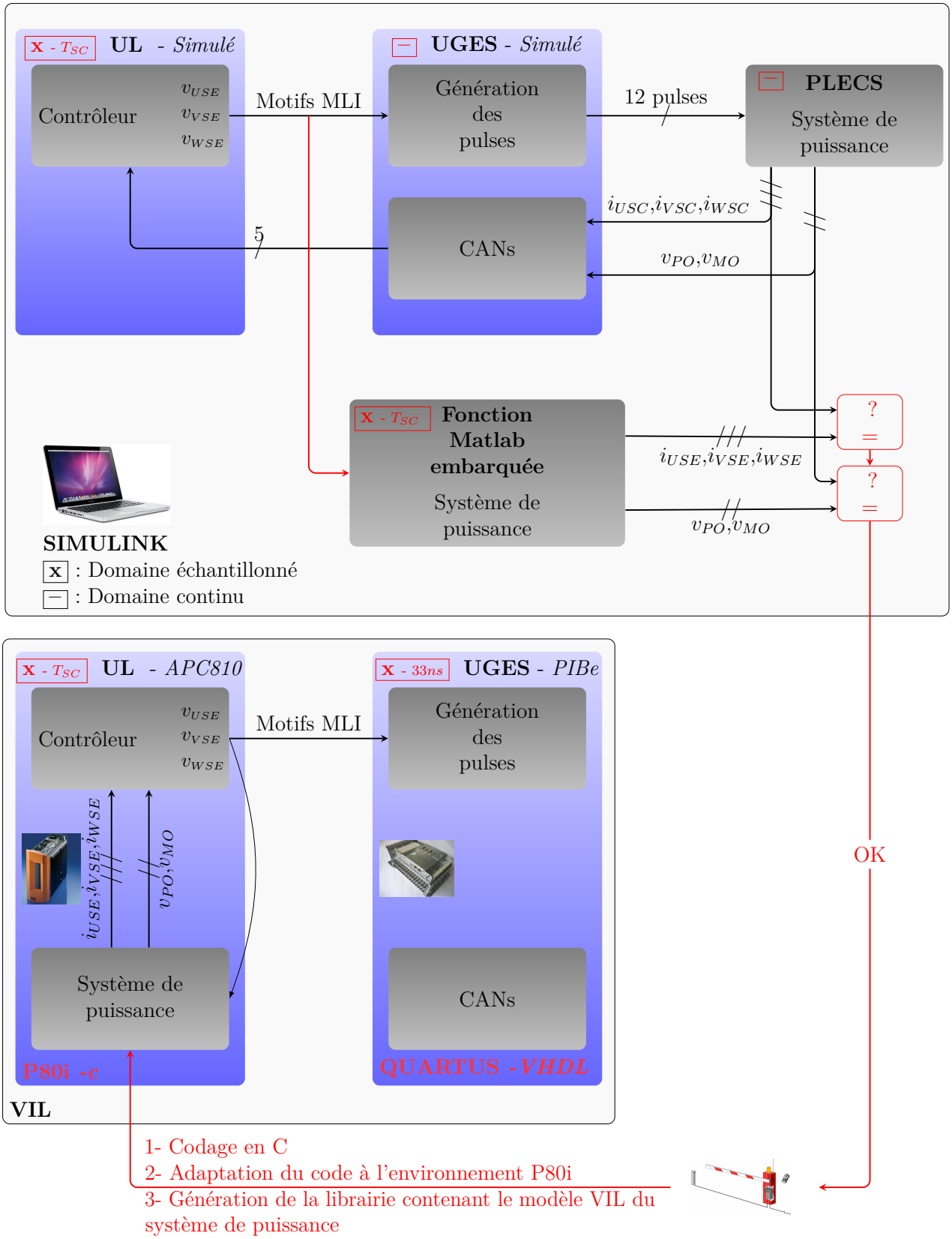


FIGURE 2.25 – Mise en oeuvre de la VIL d'un contrôleur industriel pour l'entraînement à vitesse variable d'une machine asynchrone.

La Figure 2.25 illustre la mise en oeuvre de la VIL pour notre cas d'étude.

Dans un premier temps, un modèle simplifié du contrôleur industriel est modélisé à l'aide de Simulink dans l'environnement Matlab. Le contrôleur utilise la commande vectorielle sans

capteur de vitesse avec orientation du flux rotorique. Les références de tension v_{USE},v_{VSE} et v_{USE} sont traduites en motifs MLI selon le type de MLI choisi.

A chaque phase correspond 4 mots qui définissent les motifs MLI et donc les commutations à executer pendant un pas de calcul discret T_{SC} du contrôleur. A chaque mot correspond une commutation. Chaque mot de 32 bits est défini comme suit :

b_{31} b_{30}	b_{29} .. b_0
Niveau de tension	Instant de commutation en ns

Les deux bits de poids fort définissent le niveau de tension requis :

$$b_{31}b_{30} = \begin{cases} 11 & \textit{La tension appliquée est } v_{PO} \\ 10 & \textit{Pas de commutation} \\ 01 & \textit{La tension appliquée est } v_{MO} \\ 00 & \textit{La tension appliquée est } 0 \end{cases} \tag{2.77}$$

L'unité UGES (dans notre cas la PIBe) applique alors ces tensions définies par b_{31} b_{30} aux instants définis par b_{29} .. b_0. Dès lors, l'UGES peut générer les MLIs indépendamment de la stratégie MLI. Dans un premier temps, un modèle simplifié de l'UGES est développé dans Simulink.

L'ensemble UL et UGES simulé sous Simulink pilote le modèle du système de puissance développé avec la toolbox PLECS. Ce modèle est utilisé comme modèle de référence. PLECS est alors exécuté en mode continu.

Une version du modèle de puissance que nous avions décrit est codé dans une fonction embarquée Matlab. Le code de la fonction embarquée ressemble fortement à du code en C, à quelques détails près. Le modèle ainsi codé est comparé avec le modèle de réference sous PLECS.

Nous allons présenter dans cette partie des résultats de simulation (hors ligne) et des résultats de VIL (temps réel). Les résultats ici présentés correspondent au point de fonctionnement suivant :

- La vitesse de référence est de $0.35pu$.
- Le flux rotorique de référence de $0.65pu$.
- Le moteur fonctionne à vide.
- A ce point de fonctionnement, la machine est alimentée avec des tensions homopolaires ajoutées aux références de tension de régulation.

La Figure 2.26 montre les résultats de simulation sous Matlab/Simulink. Le contrôleur tourne avec un pas $T_{control} = 667\mu s$. Les courants mesurés au niveau du contrôleur ont un contenu spectral moins riche car les valeurs intermédiaires du courant lors des commutations entre deux instants d'échantillonnage ne sont pas disponibles. Sur cette figure, nous voyons les courants statoriques i_{USC}, i_{VSC}, i_{WSC} obtenus avec modèle de référence à pas variable (continu sous PLECS) et les courants statoriques i_{USE}, i_{VSE}, i_{WSE} vus par le contrôleur. Comme dit précédemment, les tensions de références v_{USE}, v_{VSE}, v_{WSE} comportent une composante homopolaire additionnelle qui permet d'optimiser et d'uniformiser les pertes thermiques à faible taux de modulation.

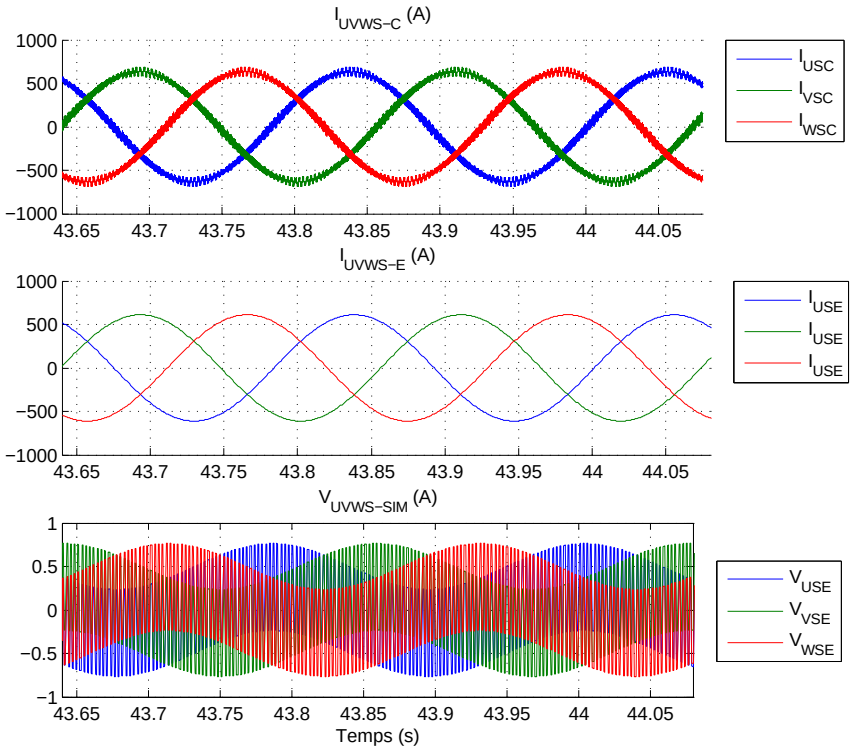


FIGURE 2.26 – De haut en bas : Courants statoriques en simulation à pas variable (continu) i_{USC}, i_{VSC}, i_{WSC} – Courants statoriques du modèle construit avec une fonction matlab (vu du contrôleur en simulation) – i_{USE}, i_{VSE}, i_{WSE} – Tensions de référence en simulation v_{USE}, v_{VSE}, v_{WSE} – Conditions de fonctionnement : Vitesse de référence = 0.35 pu - Flux rotorique de référence = 0.65 pu

Une fois les résultats cohérents avec le modèle de référence, on passe au codage en C. Le code en C est obtenu :

- en déclarant les variables utilisées (ce qui n'était pas nécessaire dans la fonction embarquée Matlab) ;
- en rajoutant une variable fictive d'indice 0 car l'indexation pour les variables multidimensionnelles commence par 0 en C (et par 1 sous Matlab).

Pour s'assurer de la conformité du code C, on teste les deux codes (code C et code de la fonction Matlab embarquée) avec les mêmes valeurs d'entrées pour avoir les mêmes sorties. Si les constantes et les variables réelles utilisées en C sont définis au format *double*, les résultats doivent-être exactement les mêmes. Les résultats peuvent être différents même si le code C est correct si on utilise la simple précision (*float*).

Après avoir validé le code C, on effectue une adaptation qui est propre à l'environnement propriétaire P80i de GE-PC pour en faire une librairie réutilisable.

Sur la même base matérielle, on exécute le modèle du système de puissance précédement développé et l'algorithme de contrôle. Les motifs MLI sont alors repris pour être utilisés comme données d'entrée du modèle du système de puissance afin que ce dernier puisse effectuer les calculs selon les instants de commutation spécifiés.

La Figure 2.27 représente les courants statoriques i_{US-SIL}, i_{VS-SIL}, i_{WS-SIL} obtenus en VIL et les références de tension v_{US-SIL}, v_{VS-SIL}, v_{WS-SIL} du contrôleur physique réel. Ces résultats sont cohérents si l'on se réfère aux résultats de simulation hors temps réel de la Figure 2.26.

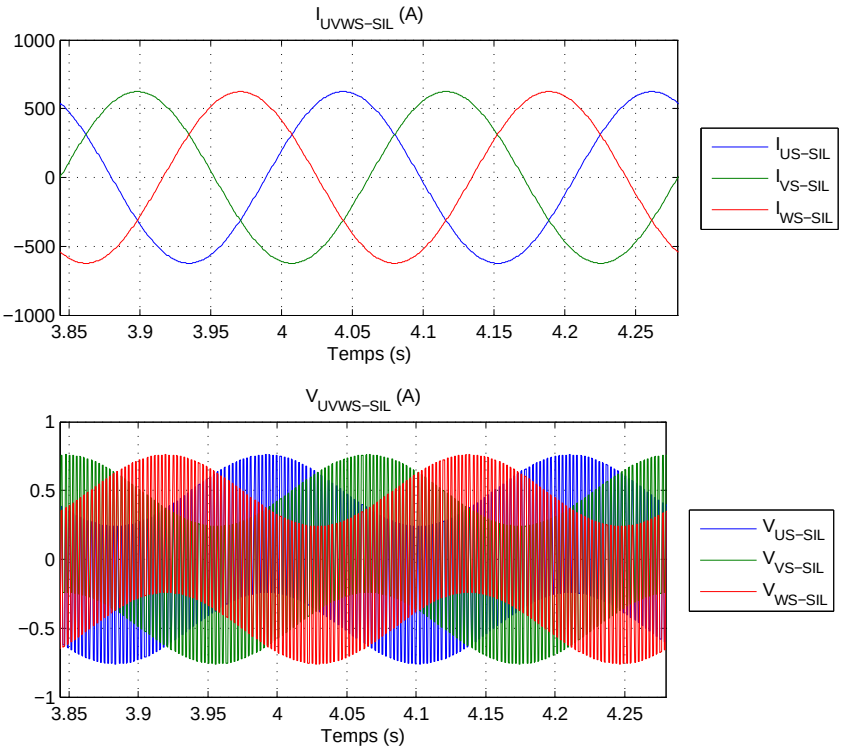


FIGURE 2.27 – Courants statoriques en VIL i_{US-SIL}, i_{VS-SIL}, i_{WS-SIL} – Tensions de référence en VIL v_{US-SIL}, v_{VS-SIL}, v_{WS-SIL} – Conditions de fonctionnement : Vitesse de référence = 0.35 pu - Flux rotorique de référence = 0.65 pu

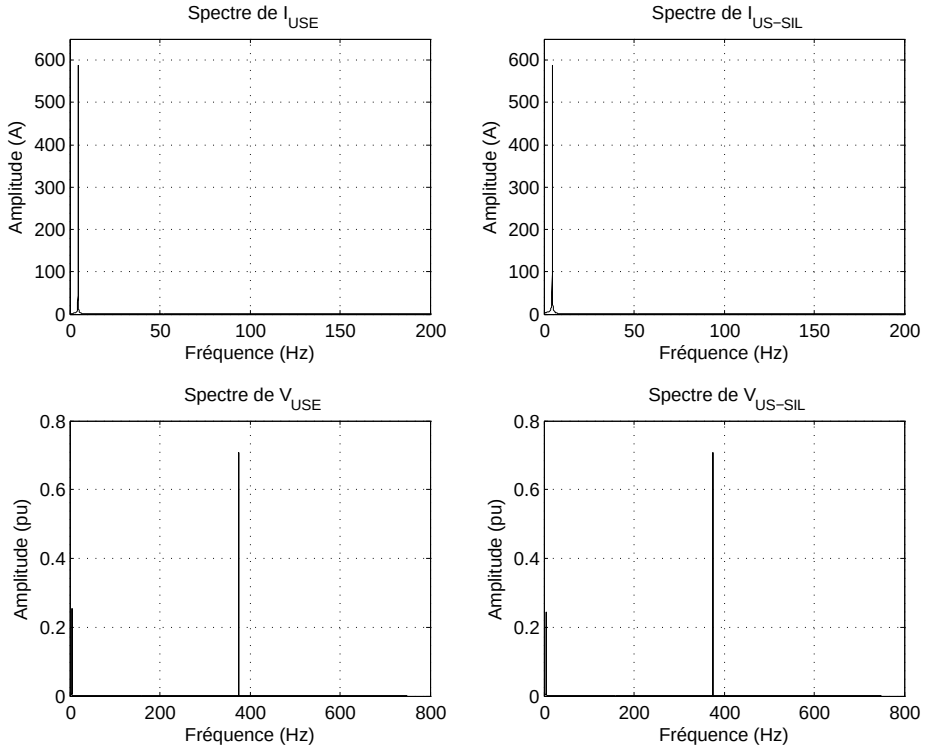


FIGURE 2.28 – En haut : spectre du courant i_{USE} de la phase U vu du contrôleur en simulation – spectre du courant i_{US-SIL} de la phase U vu du contrôleur en VIL
En bas : spectre de la tension statorique v_{USE} de la phase U en simulation – spectre de la tension statorique v_{US-SIL} de la phase U en en VIL

La Figure 2.28 présente les spectres des courants statoriques ainsi que les tensions de référence obtenus en simulation et en VIL. Nous comparons ici les grandeurs qui sont échantillonnées à la période du contrôleur comme nous sommes dans le cas d'une VIL. Nous constatons une

composante harmonique sur les tensions à $375Hz$. Cette composante harmonique de tension ne se répercute pas sur les courants car le neutre est flottant dans ce cas d'étude. En réalité, cette composante harmonique de tension n'est vue que par le circuit homopolaire et donc par l'isolation de la machine et par le circuit de mise à la terre si l'on considère un cas où le neutre n'est pas flottant.

En terme de charge, sur un calculateur industriel avec une horloge de $2.13GHz$, le modèle d'un onduleur triphasé avec machine asynchrone s'exécute entre $12\mu s$ et $24\mu s$ dans le cas de six événements et donc 7 pas internes du simulateur à traiter. Le temps d'exécution de la tâche est ici mesuré en regardant le contenu d'un compteur en début et en fin de tâche. Cette méthode de mesure n'indique pas exactement combien de temps le processeur a passé en effectuant une tâche car entre deux lectures du contenu du compteur, la tâche peut être interrompue par une tâche plus prioritaire comme par exemple une tâche système.

2.6 Conclusion

Dans le cas de la VIL, la partie linéaire sans commutation peut se modéliser soit :
- par une discrétisation exacte ;
- par une discrétisation approximative.

Pour les raisons suivantes, nous recommandons, dans la mesure du possible, une discrétisation exacte même si elle nécessite un peu plus de temps de développement :
- le modèle discret exact n'est pas directement sensible au pas de simulation alors qu'un modèle discret approximatif peut produire des instabilités numériques en fonction du pas de calcul. Dans une VIL, le pas de calcul est imposé par le contrôleur réel et en aucun cas la partie simulation ne doit rajouter des contraintes sur le pas de calcul du contrôleur. Le pas de calcul du contrôleur est déjà contraint par les types de MLI à utiliser qui eux même varient en fonction du point de fonctionnement.
- en utilisant les modèles exacts, on pourrait simuler en temps réel la présence de filtres anti-repliement quoique leurs dynamiques ne peuvent pas être prise en compte par les méthodes d'intégrations numériques (problème de stabilité avec le pas de calcul du contrôleur). La présence de filtres anti-repliement pourrait-être prise en compte car les réponses temporelles exactes des filtres passe bas sont connues.

Nous avons montré l'utilité et les avantages de la VIL :
- la VIL ne nécessite pas de matériel additionnel pour faire de la simulation temps réel et réduit donc le coût et augmente ainsi le nombre d'utilisateurs ;
- comme le modèle de la partie puissance est implémenté sur le contrôleur réel, une reproduction du contrôleur dans un autre environnement n'est pas nécessaire une fois le modèle du système de puissance validé. Hormis le fait que les entrées/sorties sont fournies par le modèle du système de puissance à la place des entrées/sorties physiques, le contrôleur est gardé tel qu'il est expérimentalement et pratiquement ;
- la VIL procure un gain de temps énorme. Une simulation sur $100s$ de l'ensemble contrôleur/onduleur de tension à trois niveaux/machine asynchrone sous Matlab/Simulink dure $19min\ 10s\ 50''$ sur un ordinateur usuel ayant un processeur de $2.5GHz$. Parfois, pour gagner du temps, on augmente la bande passante des régulateurs en simulation hors temps réel pour accélérer la convergence en s'autorisant des dépassements qui n'ont pas lieu d'être en réalité. Sinon, pour des machines de forte puissance, on diminue l'inertie pour arriver rapidement à un point de fonctionnement donné. Une initialisation à partir des conditions initiales est théoriquement envisageable mais cette solution devient facilement compliquée quand le système de puissance est complexe, et le contrôleur simulé doit lui aussi être correctement initialisé dans ce cas.

Par contre, la VIL ne valide pas l'interface d'entrées/sorties du contrôleur mais s'assure seulement que la partie algorithmique, sans interférence avec l'interface d'entrées/sorties, est correcte. Par exemple, avec l'injection d'homopolaire précédemment utilisée, lorsque la fréquence de la tension fondamentale statorique est multiple de la fréquence des porteuses, de courtes durées de conduction sont générées. Ces courtes durées de conduction sont en réalité inhibées par l'interface de génération des signaux de gâchette pour protéger les interrupteurs commandés. Sans précautions particulières, ces phénomènes apparaissent également dans le cas d'un onduleur trois niveaux quand le taux de modulation avoisine 0, 1 ou -1 avec une MLI à porteuse triangulaire.

Pour prendre en compte l'interface d'entrées/sorties :

- soit l'on la modélise comme on modélise le système de puissance et on fait du VIL. Cette solution ne valide pas l'interface physique réelle car elle sera substituée. En plus, les interfaces d'entrées/sorties industrielles sont programmables et ces programmes sont en perpétuelles évolution au niveau de leurs fonctionnalités spécifiques ;
- soit on passe à une validation nécessitant un investissement matériel autre que le contrôleur physique et donc plus chère, qui est la validation par interface matérielle.

Chapitre 3

Validation temps réel de contrôleurs industriels par Interface Matérielle

Simulateur 'Tout sur FPGA' - Simulateur FPGA/ Processeur rapide - Simulateurs FPGA / Processeurs lent

3.1 Introduction

Dans le chapitre précédent, nous avons vu que la Validation par Interface Logicielle (VIL) permet de valider l'algorithme de contrôle et son implantation matérielle, en ayant uniquement recours à la base matérielle dédiée au contrôle (UL- BnR APC620). Cependant, cette approche ne permet pas de valider les signaux qui sont réellement générés par l'interface d'entrée/sortie (carte PIBe) car cette interface n'est pas prise en compte par la VIL.

Dans certains cas, l'interface d'entrée/sortie effectue de légères modifications des signaux de commande. Par exemple, pour des fins de protection des composants de puissance en commutation, les durées de conduction minimale et de temps morts sont imposées. Dans le cas d'un onduleur trois niveaux, un séquentiel bien déterminé assure les passages entre les trois niveaux de tension en prenant en compte les temps morts.

Pour valider en même temps le contrôle et les signaux physiques générés par l'interface réelle, on procède à ce que l'on appelle Validation par Interface Matérielle (VIM), de l'anglais Hardware In the Loop (HIL). La VIM permet de valider le contrôleur physique final et son interface tel qu'ils seront sur site avec le reste de l'équipement.

Pour pouvoir valider en même temps le contrôleur et son interface d'entrée/sortie, l'utilisation de composants rapides comme les FPGAs est obligatoire. En effet, les temps de cycle des calculateurs exécutant des instructions préenregistrées en mémoire (processeur, PowerPC, DSP,...) ne permettent pas de prendre en compte l'ensemble des phénomènes liés aux commutations. Pour prendre en compte ces phénomènes, on utilise les FPGAs :

- soit en élément actif de simulation ;
- soit pour l'horodatage des signaux de gâchette.

Dans ce chapitre, nous allons présenter diverses approches possibles, illustrant l'utilisation des FPGAs pour la simulation temps réel en VIM. Nous aborderons successivement la VIM réalisée sur une unique cible FPGA, puis la VIM réalisée par une association FPGA/Processeur.

3.2 Simulateurs 'Tout FPGA'

Nous présentons dans cette partie une solution de type VIM utilisant une unique cible matérielle de type FPGA. Comme illustré par la Figure 3.1, nous utilisons une carte industrielle

spécifique à GE-PC, nommée Power Interface Board EtherCAT (PIBe), incluant un composant FPGA EP2C20F484 de la famille Altera. Cette carte PIBe est initialement dédiée à la génération des signaux de commande et aux acquisitions de données nécessaires au contrôle pour imposer une loi de commande. La même carte a été utilisée en VIL selon le principe de la Figure 2.24 du deuxième chapitre.

Comme mentionné dans le deuxième chapitre, cette carte PIBe comprend :
- 8 entrées analogiques ;
- 4 sorties analogiques ;
- 2 cartes filles de 16 entrées/16 sorties optiques, ou 16 entrées/16 sorties pulses électriques ou une carte fille d'estimation de flux. Dans le cas de la Figure 2.24, la PIBe gère une carte de génération d'impulsions de commande optiques. Les impulsions optiques sont utilisées pour commander les interrupteurs des convertisseurs statiques.

Les caractéristiques du composant FPGA EP2C20F484 intégré dans la carte PIBe 20X4550 sont comparées avec celles du composant FPGA EP1S80B956C7 (carte type utilisée dans la simulation temps réel sur cible FPGA) sur la Table 3.1.

TABLE 3.1 – Comparaison des composants FPGAs EP2C20F484 et EP1S80B956C7.

Cyclone II EP2C20F484	Quantité
Éléments logiques	18 752
Pins	315
Mémoires	239 616 bits
PLLs	4
Blocs DSP	0

Stratix EP1S80	Quantité
Éléments logiques	79 040
Pins	1 203
Mémoires	7 427 520 bits
PLLs	12
Blocs DSP	22

Les cartes FPGA utilisées dans le cadre de la simulation temps réel possèdent un nombre important d'éléments logiques et de blocs DSP. En effet, les modèles de systèmes électrotechniques développés pour les applications temps réel demandent une capacité de calcul importante. La série de carte PIBe n'est donc pas spécifiquement dédiée et optimisée pour la simulation temps réel. Nous allons donc devoir effectuer des optimisations pour pouvoir utiliser ce type de carte pour la VIM.

Néanmoins, l'utilisation de cette carte diminue le temps de développement puisque sa validation a déjà été réalisée dans le cadre de développements liés à l'implémentation matérielle du contrôleur. Les programmes nécessaires pour exploiter les ressources d'interface entrée/sortie gérées par ses cartes sont également déjà validés.
Les caractéristiques des signaux d'entrées sont compatibles avec les signaux de sorties. Par exemple, les sorties analogiques (+/- 10V) peuvent être rebouclées sur les entrées analogiques (en mode +/- 10V). Les sorties optiques peuvent aussi être rebouclées sur les entrées optiques. Dans le même esprit, nous verrons également à la section dédiée à la simulation sur FPGA/Processeur que le matériel dédié au contrôle sera utilisé dans le cadre de la VIM incluant processeur et FPGA.

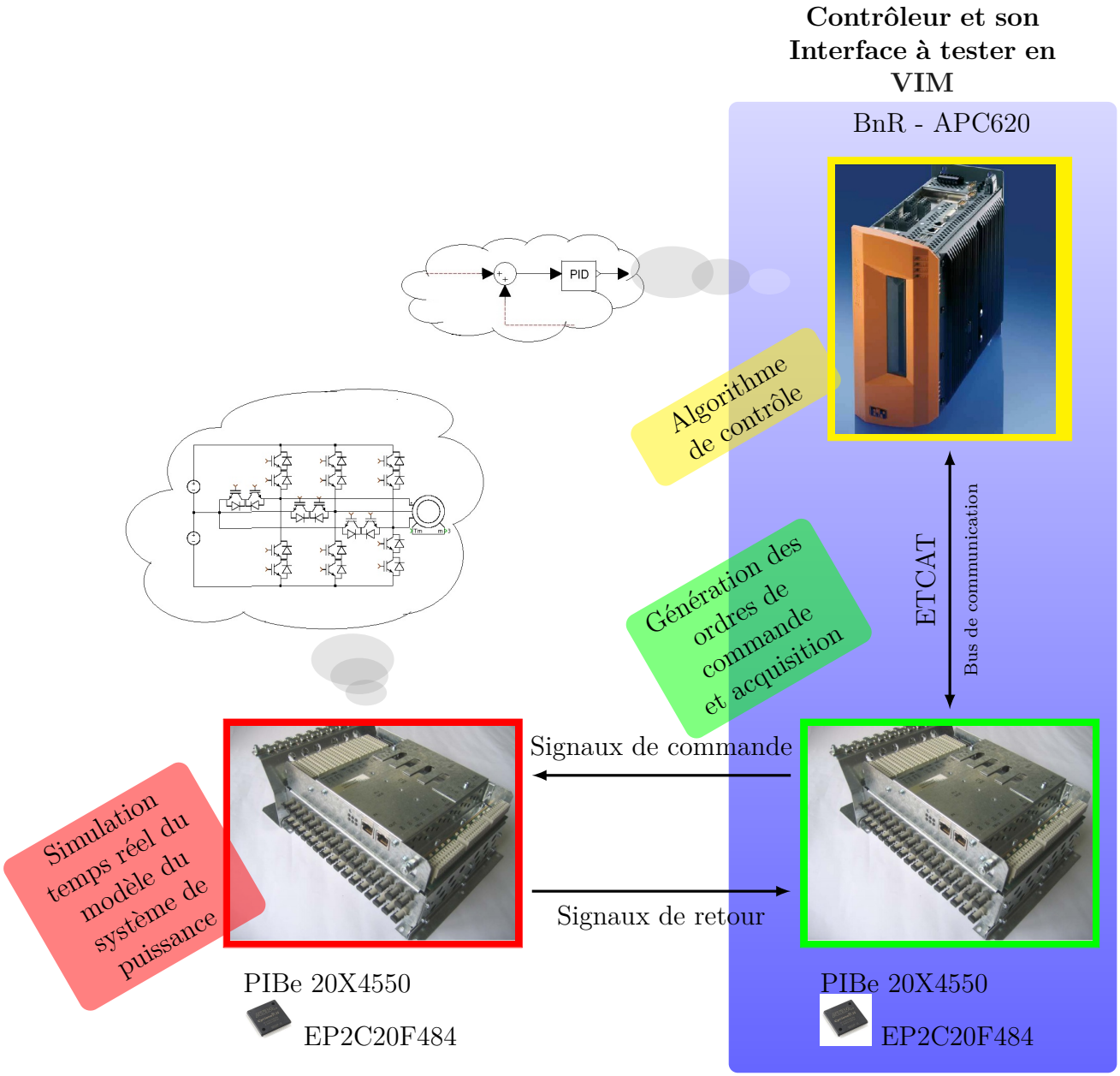


FIGURE 3.1 – VIM réalisée sur une unique cible FPGA.

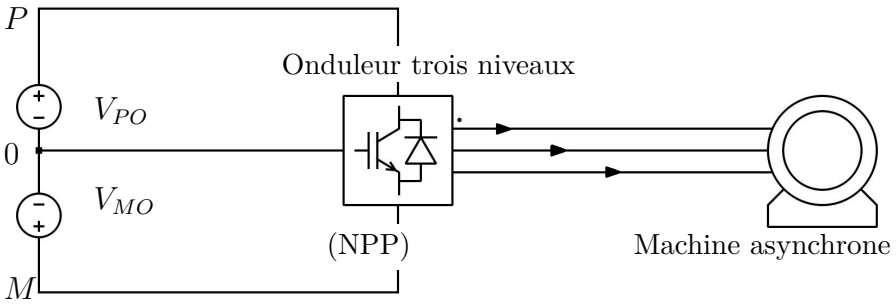


FIGURE 3.2 – Système de puissance simulé dans le cadre d'une VIM.

Nous reprenons dans cette section le même système de puissance (Figure 3.2) que celui utilisé comme exemple d'application au deuxième chapitre. Le système de puissance est constitué d'un onduleur NPP trois niveaux et d'une machine asynchrone.

3.2.1 Modélisation de l'onduleur

Nous étudierons le cas d'un onduleur triphasé de type NPP dont un des trois bras est représenté à la Figure 3.3. La table 2.1 du deuxième chapitre consigne les relations entre les signaux de gâchette, les tensions d'onduleur et les courants de ligne.

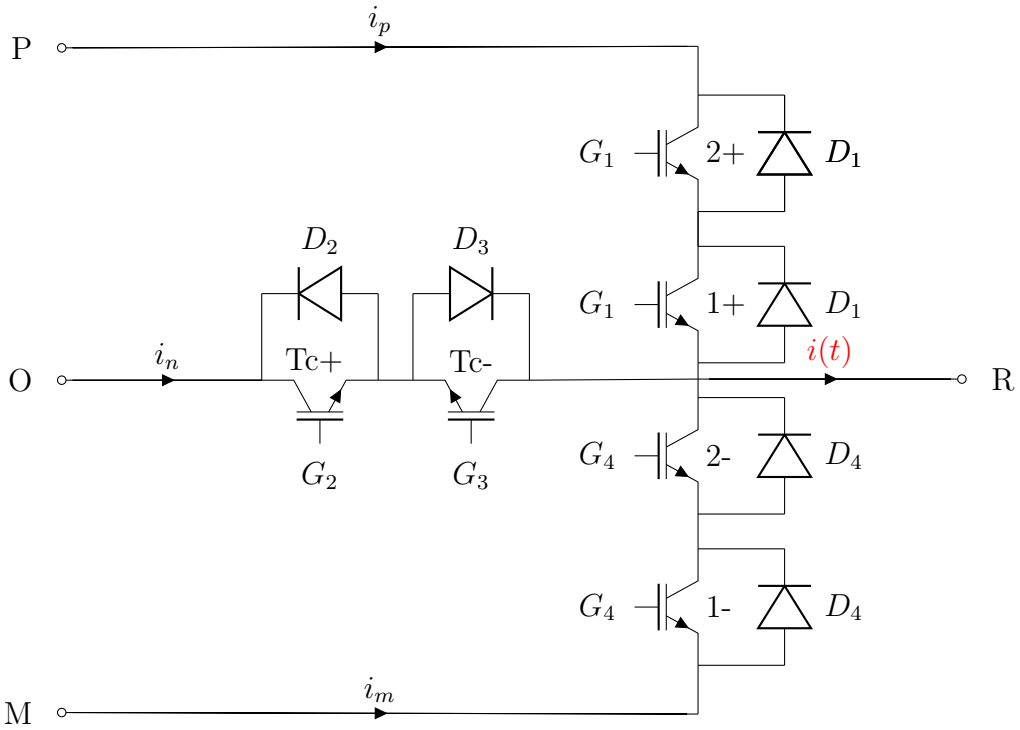


FIGURE 3.3 – Bras d'un convertisseur de type NPP - $[G_1G_2G_3G_4]$: Combinaison des Signaux de Gâchette (CSG) - i(t) : Courant de la charge.

Pour simplifier la modélisation, on s'intéresse seulement au comportement macroscopique du convertisseur. La modélisation détaillée de chaque interrupteur compliquerait davantage la modélisation et aurait un intérêt uniquement dans les cas où l'on désire étudier les contraintes des composants, ou bien encore les transitoires de fermeture et d'ouverture.
On note cependant que la prise en compte des pertes au niveau des composants en commutation est parfois importante. Pour des applications d'entraînement à vitesse variable où le bus continu est alimenté par un ou plusieurs ponts de diodes sans système de dissipation, la puissance régénérée est uniquement dissipée par les pertes (les pertes mécaniques, les pertes des résistances dans le circuit, les pertes des snubbers et également les pertes par commutation et par conduction des interrupteurs). Dans ce cas, la dissipation due aux pertes limite la montée de la tension du bus en cas de régénération de puissance à travers ce dernier. Ce cas arrive quand le cahier des charges n'impose pas une rampe de décélération contraignante et lorsqu'on laisse la machine s'arrêter par ses propres pertes par frottement.

Le modèle que nous proposons est un ensemble de fonctions logiques combinatoires et séquentielles. De telles fonctions conviennent parfaitement à une implémentation sur cible FPGA car les FPGAs embarquent de milliers d'éléments logiques de base.

La Figure 3.4 présente le principe de la modélisation. Trois modes indépendants (mode normal - mode haute impédance - mode redresseur) s'exécutent en parallèle pour définir l'état du convertisseur. Un seul mode de fonctionnement est valide à un instant donné. Le mode actif est sélectionné grâce à de la logique séquentielle (machine d'états illustrée par la Figure 3.8 et discutée à la section 3.2.1.4). Le choix de l'état actif du convertisseur dépend de la Force

ElectroMotrice (FEM) de la charge du convertisseur, des signaux de gâchette (CSG) et du courant de la charge (i(t)).

A cause des erreurs de quantification liées aux formats binaires spécifiques utilisés, le courant de la charge du convertisseur n'est jamais exactement nul une fois le modèle exécuté. On utilise alors une bande de courant δ_I positive et de faible valeur, en plus de la détection du changement de signe du courant. Lorsque la valeur du courant est comprise dans l'intervalle $[-\delta_I, +\delta_I]$, le courant est considéré étant comme nul. Par principe de continuité, un courant ayant changé de signe est bien passé au moins une fois par zéro. Notons qu'un calcul "rétroactif" de l'instant d'annulation du courant est nécessaire pour des pas de calcul type VIL mais n'est pas obligatoire dans le cas d'une VIM "tout sur FPGA" étant donné les pas de calcul extrêmement petits vis à vis des constantes de temps mises en jeu.

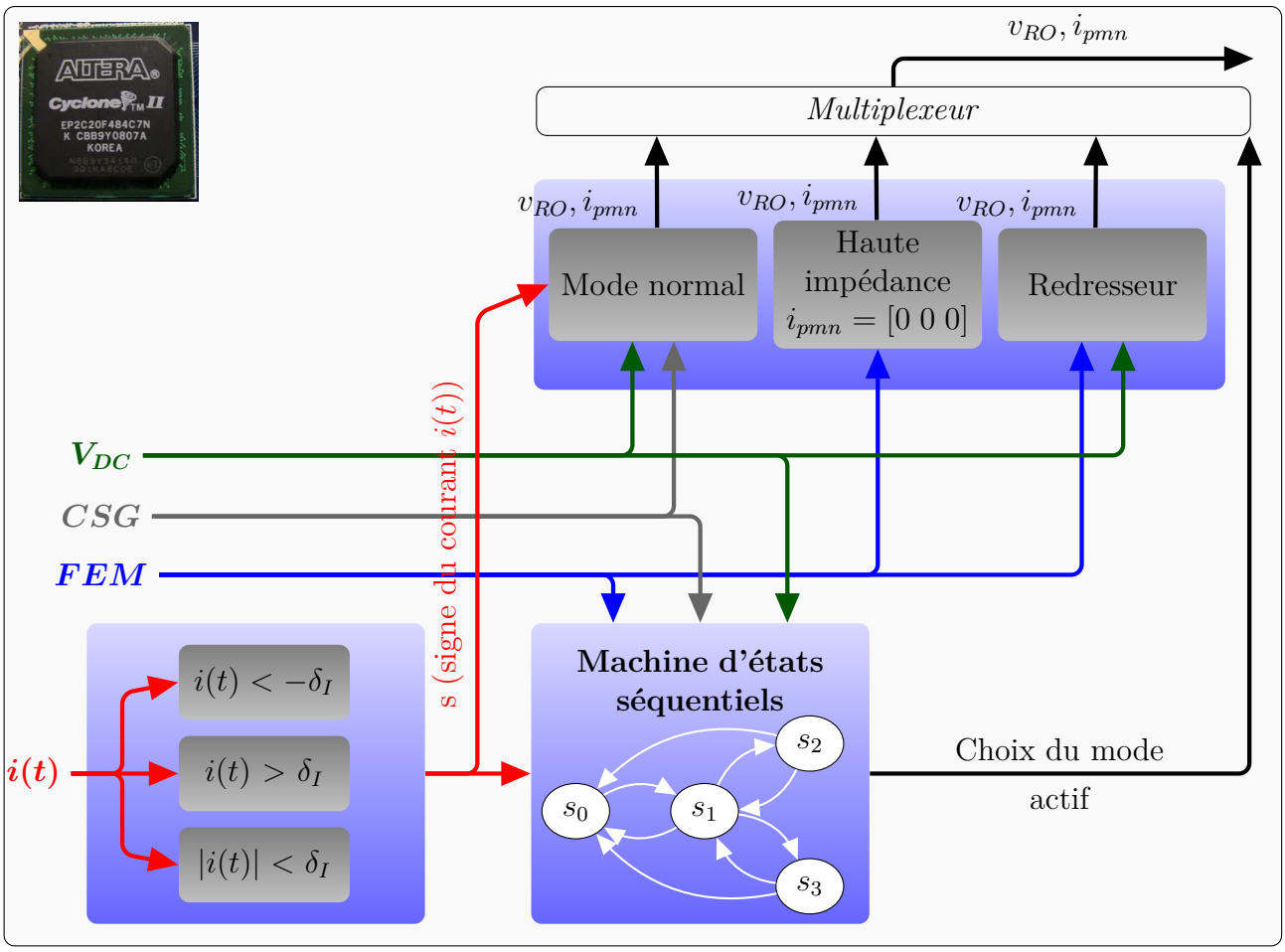


FIGURE 3.4 – Illustration fonctionnelle du modèle d'un bras du convertisseur implémenté sur FPGA $v_{RO}(t)$: Tension du bras - i_{pmn} : [i_p i_m i_n] Courants du bus continu - $V_{DC} = [V_{PO}, V_{MO}]$.

3.2.1.1 Mode de fonctionnement normal

Soit s la variable qui indique le signe du courant $i(t)$, définie comme suit :
- si $i < 0$ alors $s = 1$
- sinon $s = 0$

On définit les variables v_p, v_m et v_o qui permettent d'établir la tension de la charge en mode de fonctionnement normal :
- si $v_p = 1$ alors $V_{RO} = V_{PO}$;
- si $v_m = 1$ alors $V_{RO} = V_{MO}$;

– si $v_o = 1$ alors $V_{RO} = 0$.

La modélisation du mode de fonctionnement normal est établie sur la base de la logique combinatoire. Le cas d'un onduleur NPP est consigné au niveau de la table 3.2. Les tables de vérité des signaux v_p, v_m et v_o sont construites à partir des résultats de la table 2.1 (voir chapitre 2).

Les résultats résumés dans cette table ont été obtenus en établissant le chemin pris par le courant $i(t)$ en fonction des signaux de gâchette et de son signe. A titre d'exemple, retrouvons l'expression logique qui détermine le signal binaire v_p. Les cas de figures correspondant à une tension de sortie V_{PO} dans le cas d'un NPP sont :
– $Config = 0$ et $i < 0$;
– $Config = 4$ et $i < 0$;
– $Config = 8$ ou $Config = 12$.
On résume les cas précédents dans deux tables de vérités. Chaque table correspond à un signe de courant (s) déterminé.

TABLE 3.2 – Table de vérité du signal v_p.

$s = 0$ $(i > 0)$

G_1G_2 \ G_3G_4	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	0
10	1	0	0	0

$s = 1$ $(i < 0)$

G_1G_2 \ G_3G_4	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	0	0	0
10	1	0	0	0

On déduit alors l'expression de v_p suivante :

$$v_p = s.\overline{G_3}.\overline{G_4} + \overline{s}.G_1.\overline{G_3}.\overline{G_4} = (s + \overline{s}.G_1).\overline{G_3}.\overline{G_4} \tag{3.1}$$

Certaines combinaisons de signaux de gâchette engendrent des court-circuits ou des surtensions sur les composants en commutation. Ces cas erratiques (Dans le cas NPP : configurations 5,7,9,10,11 ...) sont détectés et signalés à l'utilisateur par le signal S_c. Ces cas ne sont pas simulés d'un point de vue électrotechnique.
$S_c = 1$ si un cas erratique est survenu, sinon $S_c = 0$.

La table 3.3 résume les expressions des signaux v_p, v_m, v_o et S_c :

TABLE 3.3 – Expressions logiques des signaux v_p, v_m, v_o et S_c dans le cas d'un onduleur trois niveaux NPP.

v_p	$(s + \overline{s}.G_1).\overline{G_3}.\overline{G_4}$
v_m	$(s.G_4 + \overline{s}).\overline{G_1}.\overline{G_2}$
v_o	$(s.G_3 + \overline{s}.G_2).\overline{G_1}.\overline{G_4}$
S_c	$G_1.G_3 + G_1.G_4 + G_2.G_4$

De même, les courants du bus continu i_p, i_m et i_n en mode de fonctionnement normal dépendent des variables v_p, v_m et v_o :

- si $v_p = 1$, le courant circule à travers (2+, 1+) ou à travers les 2 diodes D1 alors $i_p = i$ sinon $i_p = 0$.
- si $v_m = 1$, le courant circule à travers (2-, 1-) ou à travers les 2 diodes D4 alors $i_m = i$ sinon $i_m = 0$.
- si $v_o = 1$, le courant circule à travers (Tc-, D2) ou (Tc+,D3) alors $i_n = i$ sinon $i_n = 0$.

La logique combinatoire ne prend pas en compte le comportement "redresseur" du convertisseur dans le cas d'une charge devenue génératrice. En effet, la fonction combinatoire impose une relation de causalité figée entre le convertisseur et sa charge. Avec le mode de fonctionnement normal (causalité figée), l'onduleur impose toujours sa tension aux bornes de la charge.
Une machine électrique en rotation présente des FEMs au moment quand on coupe les signaux de gâchette comme le flux ne s'annule pas instantanément. Ces FEMs sont susceptibles de rendre passantes les diodes en fonction du niveau de tension du bus continu. Le cas où les diodes restent ouvertes est traité dans le mode de fonctionnement haute impédance.

3.2.1.2 Mode de fonctionnement haute impédance

Le mode haute impédance est un cas de fonctionnement où aucun interrupteur n'est commandé et aucune diode ne conduit. D'un point de vue macroscopique, le comportement du convertisseur se résume au comportement des diodes qui ne conduisent pas tant que la tension à leurs bornes n'est pas positive (ou supérieure à la tension seuil si les diodes ne sont pas supposées idéales).
Dans ce cas, la tension à appliquer au modèle de la charge doit assurer un courant nul au prochain pas de calcul. Cette tension en mode haute impédance dépend donc de la charge elle même. Un bonne prise en compte du mode de fonctionnement haute impédance implique donc un développement plus détaillé du modèle de la charge du convertisseur.
Des travaux allant dans ce sens ont été réalisés en modélisant les interrupteurs par des résistances R_{ON} à l'état fermé et R_{OFF} à l'état ouvert. Un mauvais choix des résistances dans ce type de modélisation peut fausser les résultats, voire conduire à des instabilités. Le mode haute impédance annule le courant ou au moins le garde à un niveau suffisamment bas.
Nous allons voir par la suite que cette tension en mode haute impédance est aussi nécessaire pour déterminer les états des diodes.

3.2.1.3 Mode de fonctionnement redresseur

Ce mode est actif quand aucun interrupteur n'est commandé. Le comportement du convertisseur est toujours ramené au comportement des diodes. La différence avec le mode haute impédance est qu'une fois que les FEMs de la charge excèdent la tension du bus continu, les diodes (D1 ou D4 dans le cas NPP) se mettent à conduire. Les diodes du bras central (D2 et D3 dans le cas NPP) ne peuvent conduire tant que les IGBTs associés (respectivement Tc+ et Tc-) ne sont pas commandés à la fermeture. Dans ce cas, un courant s'établit une fois que des diodes se ferment. Les diodes conduisent jusqu'à l'annulation du courant.

La logique du mode redresseur varie selon que le point neutre de la charge triphasée est flottant ou impédant.

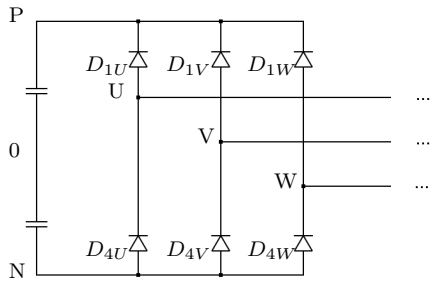


FIGURE 3.5 – Redresseur avec neutre non connecté.

Dans le cas où ce point neutre est flottant, au minimum deux des trois bras du convertisseur conduisent. Les diodes se ferment quand les tensions entre phases sont supérieures à la tension du bus. Les diodes s'ouvrent ensuite à l'annulation du courant. La Figure 3.5 montre le cas d'un redresseur triphasé à diodes. Les diodes D_{1U} et D_{4V} conduisent si $V_{UV} > V_{PN}$ dans le cas où l'on considère les diodes idéales (pas de tension seuil).

Dans le cas où le point neutre est impédant et est relié au point milieu des capacités du bus (Figure 3.6), un bras peut conduire tout seul indépendamment des autres et le courant passe alors par l'impédance Z_h. On calcule alors à chaque pas la tension simple que l'on compare avec la demi-tension du bus.

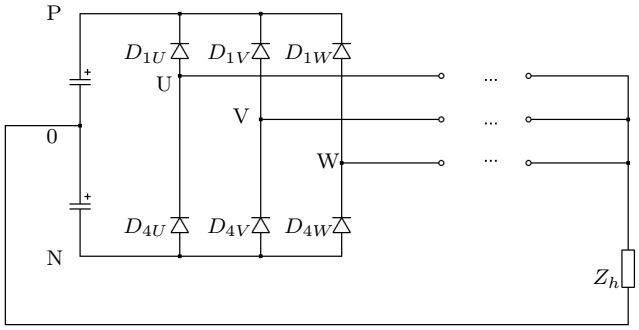


FIGURE 3.6 – Redresseur avec neutre impédant connecté.

3.2.1.4 Fonction séquentielle

Le comportement redresseur est dû aux diodes. La diode se ferme quand la tension V_{AC} à ses bornes est positive. Une fois fermée, la diode s'ouvre à l'annulation du courant (Figure 3.7).

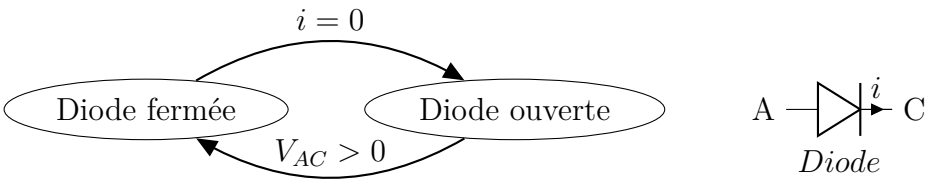


FIGURE 3.7 – Machine d'états modélisant le fonctionnement d'une diode.

La fonction séquentielle assure que sans signaux de commande des interrupteurs, les diodes se ferment par seuil de tension et s'ouvrent par annulation du courant. La fonction séquentielle active soit le mode normal, le mode haute impédance ou le mode redresseur. Cette fonction

séquentielle est représentée à la Figure 3.8. Cette machine séquentielle respecte et provient de la machine d'états de base pour une diode (Figure 3.7).

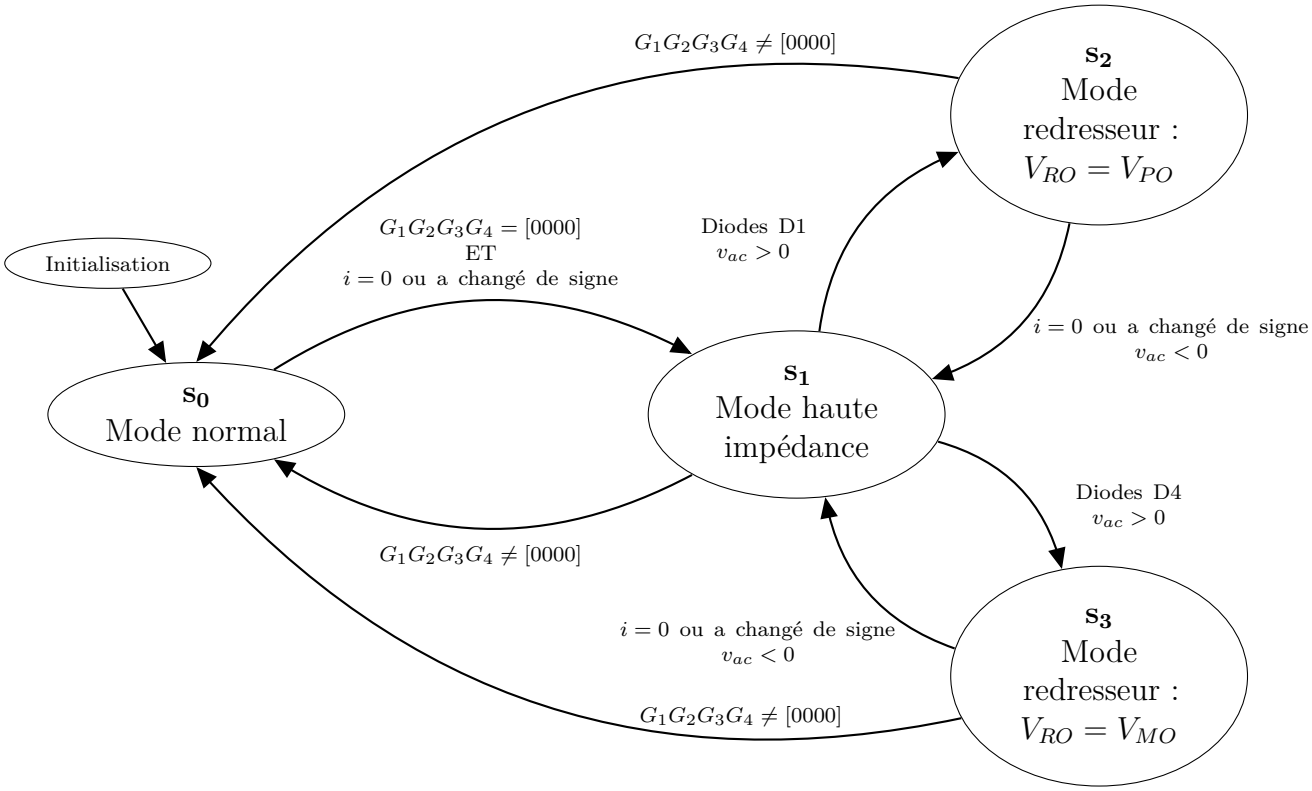


FIGURE 3.8 – Machine d'états pour la gestion des différents modes de fonctionnement.

3.2.2 Modélisation de la machine asynchrone

Le modèle continu de la machine asynchrone que nous allons utiliser ici sera le modèle dont les variables d'état sont les courants statoriques et les courants rotoriques. Les équations de départ sont les équations matricielles données par l'équation (D.56) de l'annexe D :

$$\frac{d}{dt}\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} = \underbrace{-L_{mas}^{-1}R_{mas}}_{A_{mas}} \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{X_{mas}} + \underbrace{L_{mas}^{-1}}_{B_{mas}} \underbrace{\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}}_{V_{mas}} \tag{3.2}$$

avec :

$$A_{mas} = \begin{pmatrix} a_{11} & a_{12} & a_{14} & a_{15} \\ -a_{12} & a_{11} & -a_{15} & a_{14} \\ a_{41} & a_{42} & a_{44} & a_{45} \\ -a_{42} & a_{41} & -a_{45} & a_{44} \end{pmatrix}$$

$$a_{11} = -\frac{l_{CR}R_S}{l_{CR}l_{CS} - {m_{SR}}^2} \qquad a_{41} = \frac{m_{SR}R_S}{l_{CR}l_{CS} - {m_{SR}}^2}$$

$$a_{12} = \frac{{m_{SR}}^2\dot{\theta}_R - l_{CR}l_{CS}\dot{\theta}_S}{-l_{CR}l_{CS} + {m_{SR}}^2} \qquad a_{42} = \frac{-l_{CS}m_{SR}\dot{\theta}_R + l_{CS}m_{SR}\dot{\theta}_S}{-l_{CR}l_{CS} + {m_{SR}}^2}$$

$$a_{14} = \frac{m_{SR}Rr}{l_{CR}l_{CS} - {m_{SR}}^2} \qquad a_{44} = -\frac{l_{CS}Rr}{l_{CR}l_{CS} - {m_{SR}}^2}$$

$$a_{15} = \frac{-l_{CR}m_{SR}\dot{\theta}_R + l_{CR}m_{SR}\dot{\theta}_S}{l_{CR}l_{CS} - {m_{SR}}^2} \qquad a_{45} = \frac{l_{CR}l_{CS}\dot{\theta}_R - {m_{SR}}^2\dot{\theta}_S}{l_{CR}l_{CS} - {m_{SR}}^2}$$

et :

$$B_{mas} = \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} \\ \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 & \frac{l_{CS}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 \\ 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 & \frac{l_{CS}}{l_{CR}l_{CS}-{m_{SR}}^2} \end{pmatrix}$$

On utilisera un modèle lié au stator donc $\dot{\theta}_S = 0$ et $\dot{\theta}_R = -p.w_{meca}$. En utilisant l'opérateur de Laplace, l'équation (3.2) s'écrit :

$$s.X_{mas} = A_{mas}.X_{mas} + B_{mas}.V_{mas} \tag{3.3}$$

Dans notre cas d'implémentation sur FPGA, nous utilisons la méthode d'Euler arrière présentée à la section 1.3.1.1.2. Une méthode d'intégration numérique d'ordre un est suffisante si le pas de calcul choisi est suffisamment petit par rapport aux constantes de temps mises en jeu. L'avantage de l'utilisation d'un FPGA est justement la possibilité d'avoir un pas de simulation de faible valeur, plus de cent fois plus petit que les constantes de temps électriques. Cependant, une attention doit être portée sur le choix des formats des mots qui doivent être adaptatifs et augmenter de taille quand le pas diminue.

En utilisant donc la relation (1.29) du chapitre 1, on construit les intégrateurs d'ordre 1 pour les variables d'état (courants statoriques et courants rotoriques) :

$$y_{n+1} = \frac{T_s}{1 - z^{-1}}.x_{n+1} \tag{3.4}$$

$$y_{n+1} = T_s.x_{n+1} + y_n \tag{3.5}$$

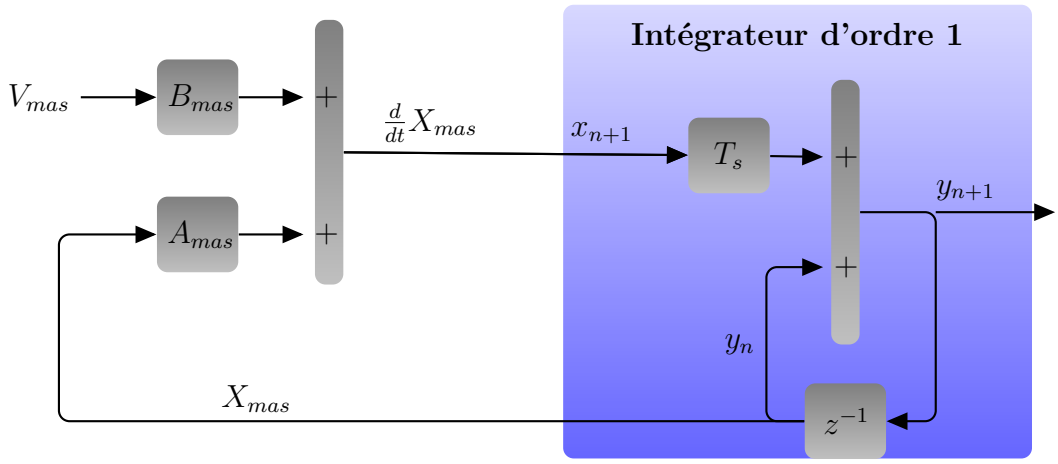


FIGURE 3.9 – Discrétisation du modèle de la machine asynchrone pour une implémentation sur FPGA.

3.2.3 Implémentation

Les différentes actions à entreprendre pour pouvoir implémenter un modèle de l'ensemble onduleur trois niveaux et de la machine asynchrone sur une carte à faible ressources logiques sont représentées à la Figure 3.10.

Tout d'abord, le convertisseur statique, la partie électrique et la partie mécanique de la machine sont séparés et constituent chacun un domaine caractérisé par son propre pas de calcul.

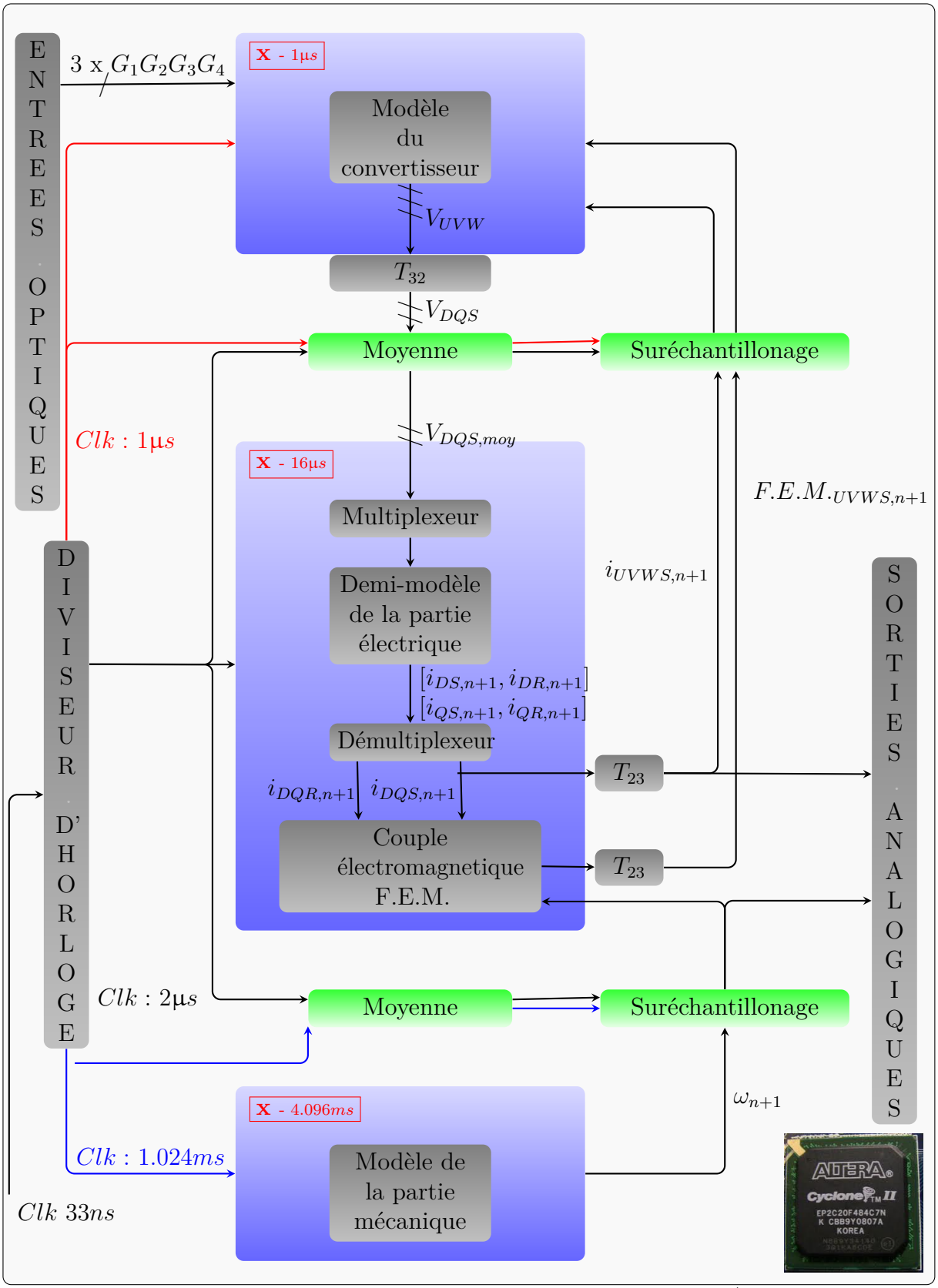


FIGURE 3.10 – Implémentation sur FPGA de l'ensemble convertisseur/machine asynchrone.

L'utilisation de plusieurs pas de calcul réduit considérablement les ressources logiques nécessaires pour l'implémentation d'un modèle temps réel sur FPGA. Les pas de calcul sont choisis en fonction des constantes de temps mises en jeu.
Les temps de fermeture et d'ouverture des IGBTs sont de l'ordre de la microseconde. Un pas de calcul d'une microseconde pour la simulation du convertisseur est donc adapté. Nous ne simulons pas les transitoires, mais le choix d'un pas d'une microseconde introduit autant d'erreur que si

l'on supposait les interrupteurs parfaits, si l'on se réfère aux interrupteurs réels.
La partie mécanique est simulée avec un pas de calcul beaucoup plus grand comme les constantes de temps mécaniques sont grandes comparées aux constantes de temps électriques. Par exemple, dans le cas de la machine simulée en VIL, nous avons une constante de temps mécanique $T_{meca} = \frac{J}{f_{\dot{\theta}}} = 15.04s$ alors que la constante de temps électrique statorique vaut $T_{stat} = \frac{L_{CS}}{R_S} = 2.13ms$.

Il convient donc d'utiliser trois horloges synchronisées pour prendre en compte les divers domaines à résolutions différentes. Les grandeurs allant d'un système rapide à un système lent sont moyennées. Les grandeurs allant d'un système lent vers un système rapide pourraient être suréchantillonnées en utilisant des extrapolations entre deux pas de calcul lents. Les grandeurs suréchantillonnées reprennent les valeurs données par la partie lente à chaque fois que les signaux sont établis. Néanmoins, dans notre cas, le suréchantillonnage n'est pas impératif bien qu'il permette d'améliorer les résultats de simulation. Ne disposant que de ressources matérielles limitées, nous n'avons pas utilisé le suréchantillonnage afin de limiter l'utilisation des ressources logiques.

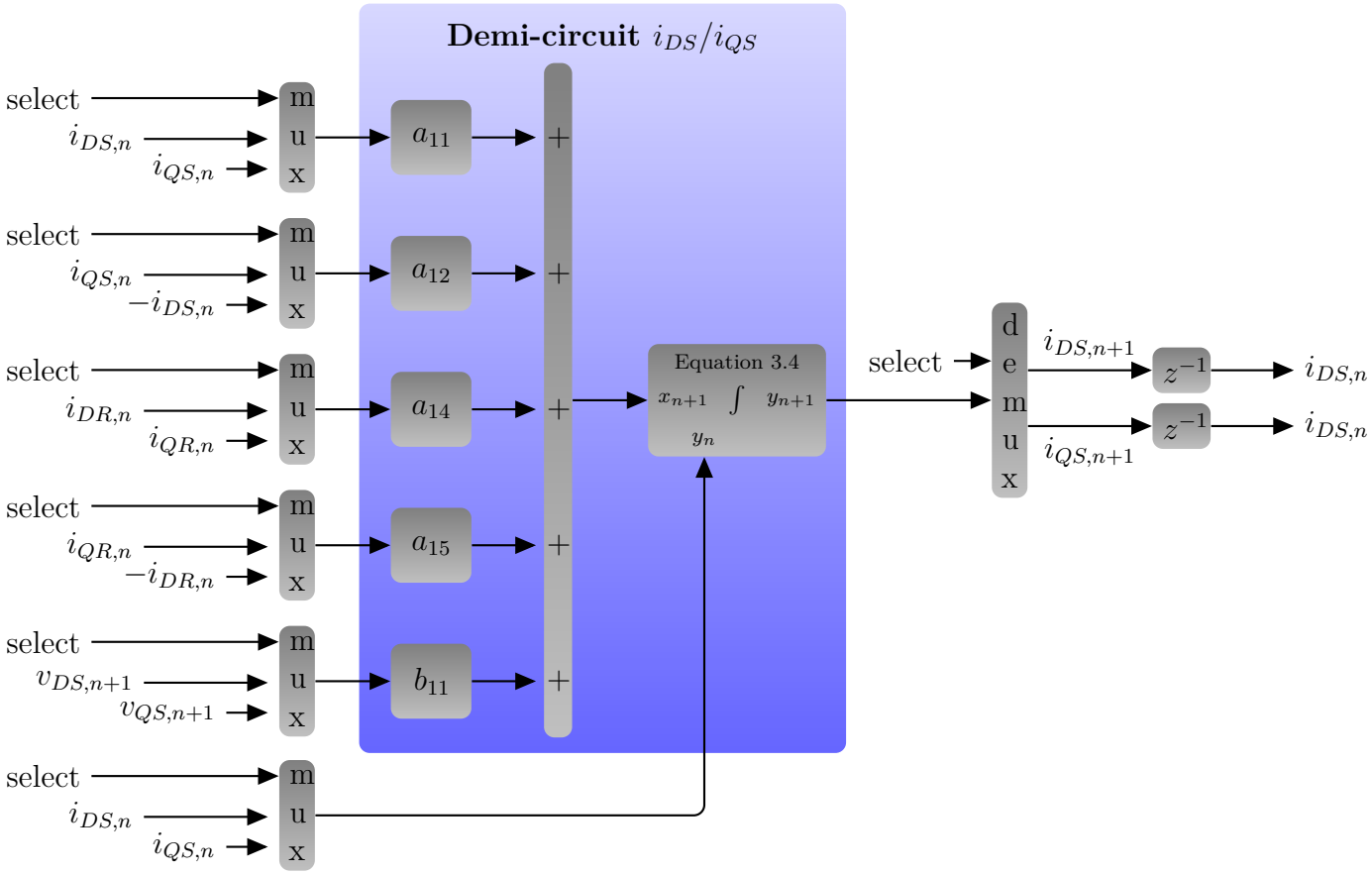


FIGURE 3.11 – Exemple de demi-circuit pour le calcul de i_{DS} et i_{QS}.

Par ailleurs, nous remarquons que les équations qui régissent la dynamique de la partie électrique de la machine asynchrone présentent une certaine similitude selon les axes D et Q. Nous allons exploiter cette similitude pour réduire davantage les ressources logiques nécessaires pour implémenter notre modèle sur un FPGA à faible ressource. Comme montré sur la Figure 3.10, on utilise un demi-circuit qui calcule successivement les résultats sur l'axe D et sur l'axe Q.

Un exemple pour le calcul des courants i_{DS} et i_{QS} est présenté à la Figure 3.11. Le même principe est utilisé pour les courants rotoriques i_{DR} et i_{QR}. Les entrées de ce demi circuit sont multiplexées et les sorties démultiplexées. Dans notre cas, les grandeurs électriques sont rafraîchies toutes les 16μs mais on attaque le demi-circuit électrique avec un horloge beaucoup

plus rapide (2μs). Cette horloge rapide permet de réaliser les différentes étapes (multiplexage, calcul et démultiplexage) durant les 16μs. On réduit encore les ressources logiques nécessaires en utilisant le même circuit pour le calcul de deux grandeurs d'état.

Pour minimiser les ressources liées au calcul des valeurs moyennes, on choisit des pas de calcul multiples d'une puissance de deux sur la base de l'horloge la plus rapide (1 μs dans notre cas). La valeur moyenne de 2^n échantillons est alors tout simplement dans ce cas la somme des 2^n échantillons interprétés autrement vu du codage logique (position de la virgule). Le résultat de la somme est tout simplement traduit dans un autre format en augmentant la taille de la partie fractionnaire de n, soit alors en déplaçant la virgule.

3.2.4 Résultats

Nous reprenons le cas du système de puissance validé en VIL au chapitre 2 que nous simulons cette fois ci sur FPGA. Le système de puissance ayant été validé en VIL au chapitre 2, il sera également implanté au niveau du contrôleur (UL - BnR APC620) afin de servir de référence.

Comme présenté à la Figure 3.12, le contrôleur et la génération des impulsions de commande des interrupteurs sont rebouclés sur le modèle temps réel exécuté sur FPGA. Le modèle développé en VIL tourne en esclave sur le contrôleur et deux sorties courants sont envoyées sur les sorties analogiques non utilisées de la carte d'interface entrée/sortie du contrôleur. Ces courants (réferences) sont ensuite comparés avec les courants calculés avec le modèle exécuté sur FPGA en VIM.

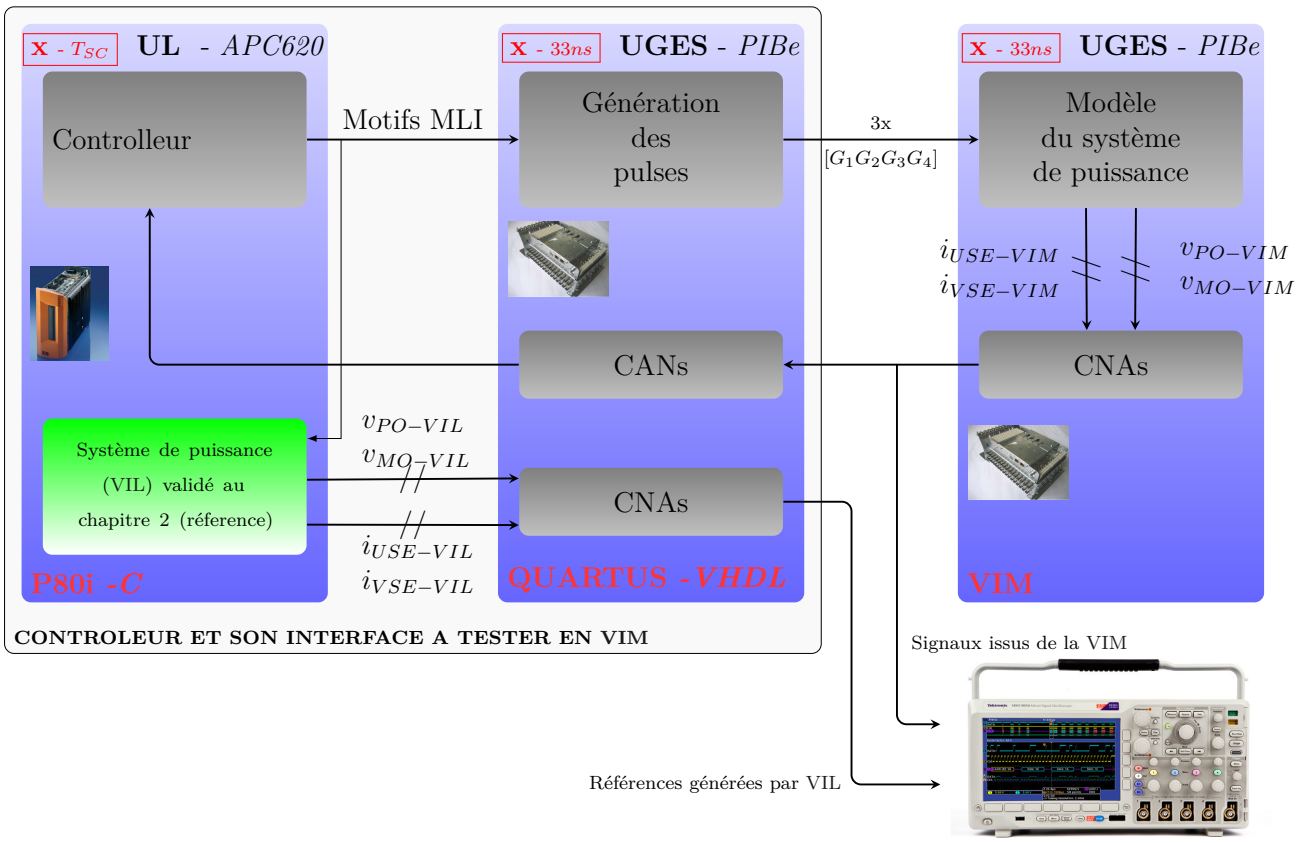


FIGURE 3.12 – Validation du modèle développé sur FPGA en VIM par comparaison avec le modèle de l'ensemble convertisseur/machine simulé en VIL (référence).

La Figure 3.13 présente les courants issus du modèle exécuté en VIL et ceux issus du

modèle implémenté sur FPGA.

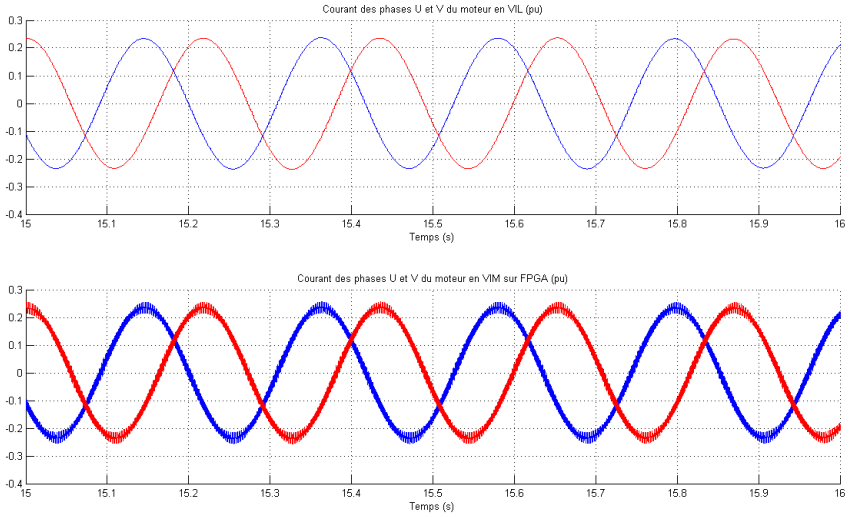


FIGURE 3.13 – Courants des phases U et V du moteur en VIL et en VIM calculés sur FPGA.

Comme le modèle en VIL s'exécute avec le même pas de calcul que le contrôleur, les commutations n'apparaissent pas sur ces courants. En effet, le modèle en VIL est en parfaite synchronisation avec le contrôle et seuls les courants en fin de cycle sont envoyés à l'oscilloscope, en sortie du CNA (Figure 3.12). L'implémentation sur FPGA permet d'avoir un contenu spectral beaucoup plus riche. Les effets des temps morts et des durées minimales de conduction sont pris en compte. La table 3.4 résume les taux d'utilisation des différentes ressources du FPGA ciblé.

TABLE 3.4 – Taux d'utilisation des ressources logiques.

Cyclone II EP2C20F484	Taux d'utilisation
Éléments logiques	58% (10831/18752)
Pins	22% (68/315)
Mémoires	0% (0/239616 bits)
PLLs	0% (0/4)

L'implémentation sur FPGA fournit de meilleures performances temporelles (période de 16μs) que l'implémentation sur processeur (période de 50μs sur simulateur temps réel et supérieure à 380μs dans le cadre de la VIL). Cependant, tout changement des paramètres moteurs nécessite une recompilation, voir même une vérification des formats de mots utilisés pour s'assurer d'avoir une bonne représentation du système de puissance simulé.

3.3 Simulateur FPGA/Processeur

3.3.1 Simulateurs FPGA/Processeur rapide

Les travaux présentés par la suite ont été développés dans le cadre d'une VIM d'un contrôleur d'un bateau casseur de glace. L'architecture du système de puissance est décrite à la Figure 3.14. La machine peut-être alimentée par un ou deux onduleurs trois niveaux de type NPC. La machine est alimentée à travers des inductances L_{circ} et R_{circ} qui limitent les

courants de circulation entre les deux convertisseurs. Ces courants de circulation peuvent-être générés à cause d'une désynchronisation des impulsions de commande ou d'une différence entre les demi-tensions des deux convertisseurs.

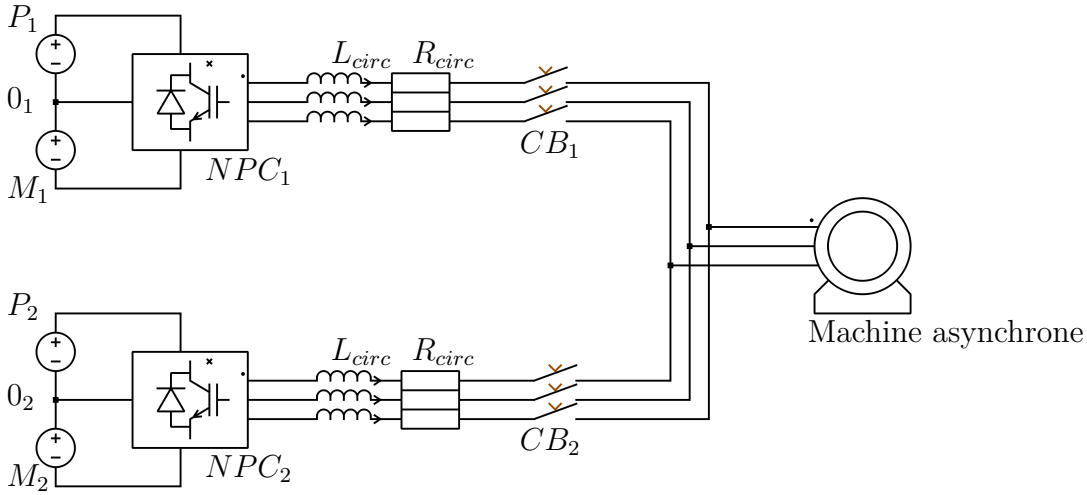
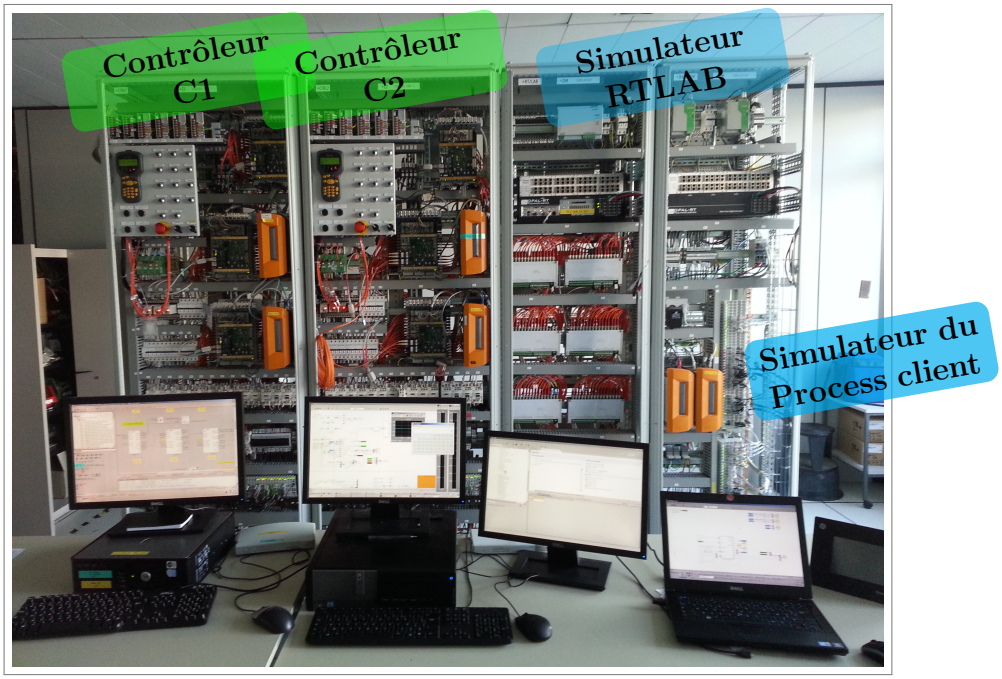


FIGURE 3.14 – Cas d'étude pour la validation temps réel sur FPGA/processeur rapide.



OP5600 (Simulateur)

PIBe - 20x4550 (Unité de Gestion des Entrées/Sorties)

PIB810 (Conversions Impulsions Optiques <-> Impulsions Électriques)

BnR - APC620 (Unité Logicielle)

FIGURE 3.15 – Plateforme de validation de contrôleurs industriels incluant deux tours contrôle configurables et le simulateur temps réel.

La plateforme de validation temps réel est montrée sur la Figure 3.15. Cette plateforme inclue :

– deux tours de contrôle $C1$ et $C2$ reconfigurables selon l'application à tester. Ces deux tours peuvent constituer une réplique de tout les armoires de contrôle.
– un simulateur temps réel RT-LAB qui exécute le modèle du système de puissance.

Les cartes PIBes génère des impulsions de commande optiques. Les entrées digitales du simulateur temps réel sont de type électrique. Les cartes PIB810 effectuent les conversions optiques/électriques.

Le simulateur temps réel RT-LAB de la société OPAL-RT peut atteindre des pas de calcul de l'ordre d'une cinquantaine de microsecondes. Ce simulateur intègre une carte mère ATX avec 8 cœurs de processeurs à $3.3GHz$, ainsi qu'une carte à base de FPGA Spartan 3 de la famille Xilinx. Il sera notre base matérielle pour cette étude.

L'objectif est :
– de répondre rapidement à une demande de VIM et donc de développer un modèle temps réel pour une validation avant la mise en service ;
– de développer un modèle simple mais qui garde une représentativité correcte du système de puissance.

Pour réaliser ce travail, nous allons présenter une manière originale permettant de modéliser le convertisseur de tension trois niveaux. L'approche classique consiste à dater les évènements sur les signaux de gâchette, les arranger dans l'ordre chronologique et les traiter un à un par les algorithmes d'interpolation. Un tel travail nécessite énormément de temps de développement et de nombreux tests de validation, vu la complexité de l'algorithmique.

Dans cette partie, nous allons voir qu'à condition d'utiliser des processeurs rapides, on peut établir une expression simple, quasi-linéaire, de la tension de sortie de l'onduleur. En effet, le choix d'un processeur rapide permet de limiter le nombre d'évènements pouvant avoir lieu en un pas de calcul temps réel vu que les durées de conductions minimales et les durées des temps morts sont connues pour la gamme de convertisseurs concernés ($20\mu s$ à $35\mu s$). Comme le nombre d'évènements est limité, il est possible de faire une analyse exhaustive des différents cas (que nous appelerons les groupes de fenêtres). Les détails pour établir l'expression de la tension de sortie à partir de la valeur moyenne des impulsions sont développés par la suite.

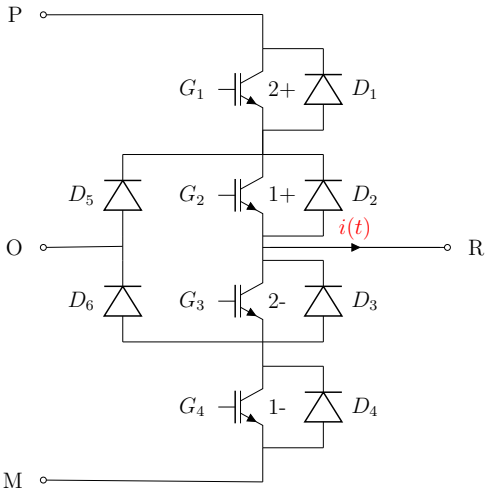


FIGURE 3.16 – Bras d'un convertisseur de type NPC.

3.3.1.1 Modélisation de l'onduleur trois niveaux

Nous prenons comme exemple le cas de l'onduleur trois niveaux de type NPC dont un bras est représenté à la Figure 3.16.

Comme la résolution du simulateur à base de processeur ne permet pas un traitement instantané des impulsions de commande, le modèle du fonctionnement normal d'un bras du convertisseur (Figure 3.4), développé pour une simulation temps réel sur base FPGA, ne convient pas. Le modèle en fonctionnement normal sera remplacé par une modélisation au sens de la valeur moyenne de la tension de sortie que nous allons présenter par la suite. Les modes de fonctionnement autres que le mode normal restent inchangés pour modéliser le convertisseur NPC.

3.3.1.1.1 Méthode basée sur la tension moyenne

Une première approche possible consiste à modéliser le convertisseur en implémentant son modèle instantané sur FPGA. Ce modèle reçoit alors les signaux de commande et le courant de charge. Ce modèle repose partiellement sur le modèle du convertisseur développé pour une implémentation complète sur FPGA (voir section 3.2.1).
La valeur moyenne de la tension sur une période de calcul est alors calculée localement par le FPGA et transmise au processeur à chaque pas de calcul. Cette solution est relativement simple à implémenter et permet de s'affranchir des calculs d'interpolation.
Dans le cas d'application qui suit, nous implémentons sur une base matérielle de RTLAB. Il est possible d'implémenter ses propres modèles sur le FPGA embarqué dans le simulateur RT-LAB mais cela nécessite l'utilisation d'un outil complémentaire appelé Xilinx System Generator (XSG). Nous ne disposons pas de cet outil et proposons une solution alternative (méthode par les impulsions moyennées) qui ne nécessite pas de développer un programme spécifique au FPGA gérant les cartes d'entrée sortie du simulateur RT-LAB.

3.3.1.1.2 Méthode par les impulsions moyennées

Hypothèses : Dans cette partie, comme nous allons implémenter sur une base de calcul matérielle rapide, les courants sont supposés constants pendant un pas de calcul. Cette supposition n'est pas validée pour des charges à dynamiques rapides. Dans nos applications, les constantes de temps électriques avoisinent les 50ms. Des pas de calcul de l'ordre de $50\mu s$ permettent donc de faire cette supposition sur les courants.

Les signaux de commande des interrupteurs vérifient les conditions suivantes :
- la durée minimale de conduction et la durée des temps morts sont supérieures au tiers du pas de calcul T_S du simulateur temps réel RT-LAB, le nombre maximal de configurations ($Config$[1]) est donc limité à 3 ;
- ces signaux de commandes respectent les temps morts (aucune transition directe du niveau de tension positive et négative n'est permise : voir Figure 3.18).

La deuxième condition doit être systématiquement vérifiée pour le convertisseur de tension trois niveaux car en réalité les interrupteurs ne se ferment pas instantanément.
La première condition nous impose donc une limite supérieure lors du choix du pas de calcul. Cette limite permettra de limiter le nombre d'évènements et de valider le résultat de l'analyse des fenêtres de temps décrites par la suite.
Pour les convertisseurs dédiés aux applications de GE-PC, les temps morts et les durées de conduction minimales sur les onduleurs trois niveaux sont d'environ $20\mu s$. Un pas de calcul de $40\mu s$ vérifie donc la première condition.

Une surveillance par de la logique simple est donc suffisante pour vérifier les conditions à

1. $Config$ correspond à la combinaison des signaux de commande $[G_1G_2G_3G_4]$. Les configurations de fonctionnements autorisées sont les configurations 2, 3, 4, 6 et 12.

valider préalablement à toute simulation du système de puissance.

Nous allons voir dans ce qui suit pourquoi ces deux conditions simplifient énormément la modélisation du convertisseur.

Valeurs moyennes des impulsions de commande La Figure 3.17 présente une impulsion de commande appliquée à un semi-conducteur du convertisseur. La valeur de l'impulsion (1 ou 0) est moyennée mais nous allons montrer que les informations ne sont pas perdues tant que les deux conditions précédemment décrites sont vérifiées.

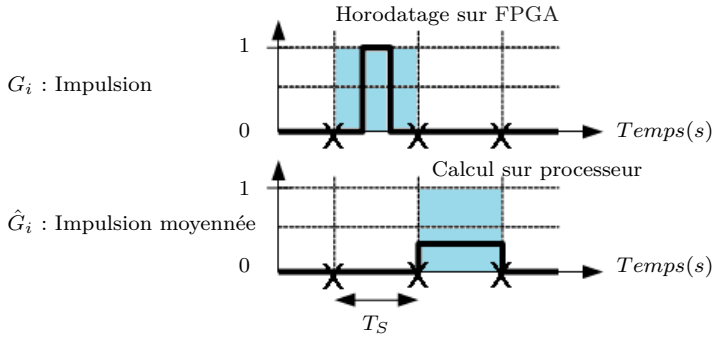


FIGURE 3.17 – Principe de la méthode utilisant les pulses moyennées.

On définit la valeur moyenne $\hat{G}_i$ d'une impulsion G_i sur la période de calcul T_s par :

$$\hat{G}_i = \frac{1}{T_s}.\int_{t_n}^{t_n+T_s} G_i\,.dt \tag{3.6}$$

Nous allons montrer une méthode simple permettant de simuler un onduleur trois niveaux avec des pas de l'ordre de $40\mu s$. Cette méthode aboutira à une expression de la tension de sortie fonction des valeurs moyennes des signaux G_1, G_2, G_3 et G_4. Afin de minimiser la complexité de la modélisation, nous allons d'abord analyser en détail le fonctionnement de ce convertisseur.

Séquences de fonctionnement Les séquences de commande de l'onduleur trois niveaux sont représentées à la Figure 3.18. La durée minimale de conduction varie en fonction de la gamme de convertisseur mais est généralement supérieure ou égale à $20\mu s$.

La tension en sortie de l'onduleur trois niveaux peut être positive ($V_{RO} = V_{PO}$), nulle ($V_{RO} = 0$) ou négative ($V_{RO} = V_{MO}$). Comme les interrupteurs réels ne se ferment pas instantanément, un temps mort ($Config$ 4) est inséré entre une tension positive ($Config$ 12) et une tension nulle ($Config$ 6). Le temps mort de la configuration 2 est utilisée entre une tension négative ($Config$ 3) et une tension nulle ($Config$ 6). Le passage direct d'une tension positive à une tension négative peut endommager l'onduleur.

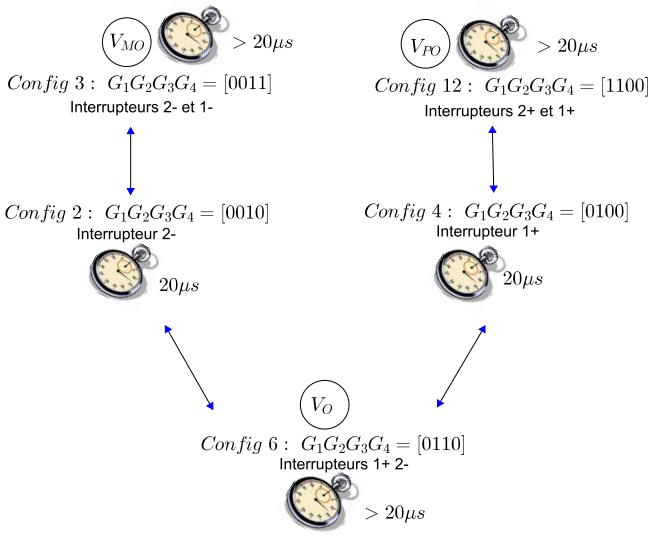


FIGURE 3.18 – Séquence de commande de l'onduleur trois niveaux de type NPC avec les configurations autorisées (table 2.1 , chapitre 2).

Cela implique que sur une fenêtre de temps de $40\mu s$, nous allons voir l'un des cinq groupes de fenêtre décrits par la suite.

Les prochaines figures présentent différentes allures de la tension instantanée V_{RO} en rappelant les successions de configurations ($Config$) possibles en une fenêtre de temps égale au pas de calcul du simulateur. Nous allons les étudier pour pouvoir déterminer une relation simple liant les valeurs moyennes des impulsions à la valeur moyenne de la tension sur une fenêtre de temps égale au pas de calcul.

La succession d'évènements est soumise :
- au passage par les temps morts (config 2 et 4) selon la séquence de commande décrite sur la Figure 3.18 ;
- à la durée minimale de conduction.

Groupe de fenêtres I Ce groupe de fenêtres illustré à la Figure 3.19 concerne la conduction des interrupteurs (1+,2-). La tension est nulle sur une partie du pas de calcul.
Cette conduction :
- est suivie ou précédée d'un temps mort de transition vers une tension négative V_{MO} dans le cas de la Figure 3.19-a et de la Figure 3.19-b.
- est entourée de deux temps morts de transition vers ou à partir d'une tension négative V_{MO} dans le cas de la Figure 3.19-c.
- correspond à une tension nulle sur tout le pas de calcul dans le cas de la Figure 3.19-d.

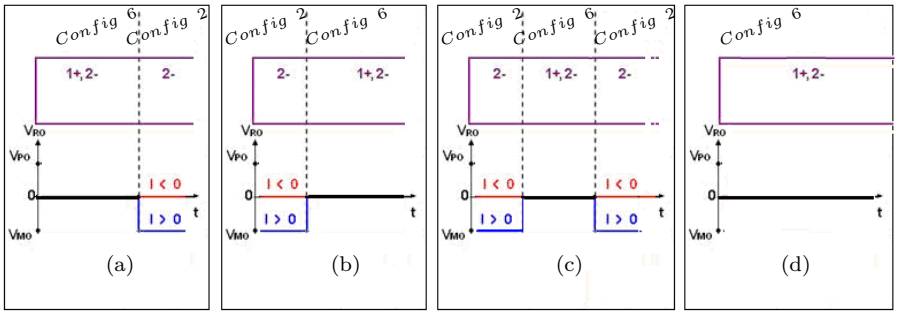


FIGURE 3.19 – Groupe de fenêtres I.

Groupe de fenêtres II Ce groupe est identique au groupe I sauf que les temps morts sont présents avant ou après des transitions vers une tension positive V_{PO}.

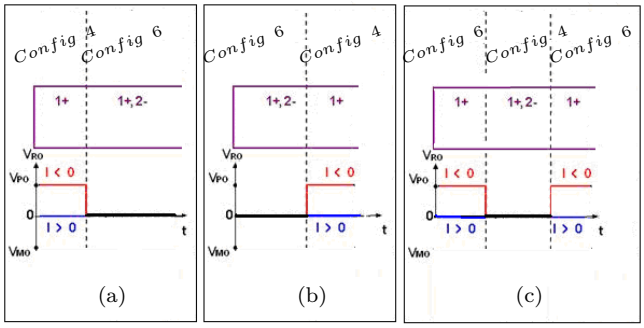


FIGURE 3.20 – Groupe de fenêtres II.

La conduction des interrupteurs (1+,2-) :
- est suivie ou précedée d'un temps mort de transition vers une tension positive V_{PO} dans le cas de la Figure 3.20-a et de la Figure 3.20-b ;
- est entourée de deux temps morts de transition vers ou à partir d'une tension positive V_{PO} dans le cas de la Figure 3.20-c.

Groupe de fenêtres III Ce groupe de fenêtres est illustré à la Figure 3.21 ; il décrit la conduction des interrupteurs (2-,1-). La tension V_{RO} est négative (et vaut V_{MO}) sur une partie du pas de calcul.

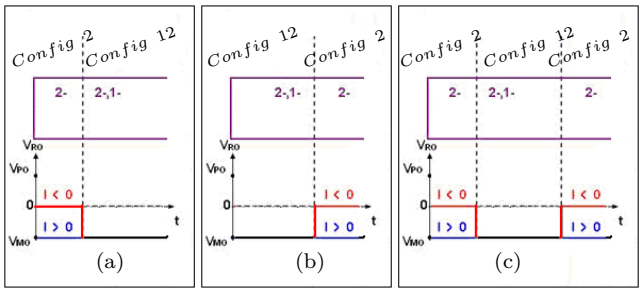


FIGURE 3.21 – Groupe de fenêtres III.

La conduction de (2-,1-) :
- se produit suite à un temps mort dédié à la transition d'une tension nulle vers une tension V_{MO} dans le cas de la Figure 3.21-a ;
- est suivie d'un temps mort de passage vers une tension nulle dans le cas de la Figure 3.21-b ;
- se produit entre deux temps morts de transition entre une tension négative V_{MO} et une tension nulle comme illustré par la Figure 3.21-c.

Groupe de fenêtres IV Ce groupe de fenêtres illustré par la Figure 3.22 concerne la conduction des interrupteurs (2+,1+). La tension V_{RO} est positive sur une partie du pas de calcul.
La conduction de (2+,1+) :
- se produit suite à un temps mort dédié à la transition d'une tension nulle à une tension V_{PO} dans le cas de la Figure 3.22-a ;
- est suivie d'un temps mort de passage vers une tension nulle dans le cas de la Figure 3.22-b ;
- est située entre deux temps morts de transition vers et à partir d'une tension nulle dans le cas de la Figure 3.22-c.

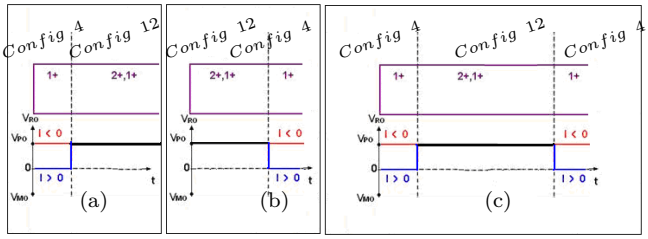


FIGURE 3.22 – Groupe de fenêtres IV.

Groupe de fenêtres V La Figure 3.23-a présente les niveaux de tension positive et nulle ainsi que le temps mort correspondant. L'ordre des séquences positive et nulle peut s'inverser autour de la séquence de temps mort mais cela ne change rien du point de vue de l'approche par valeur moyenne.
La Figure 3.23-b montre une fenêtre composée des niveaux de tension négative et nulle ainsi que le temps mort correspondant. L'ordre des séquences négative et nulle peut également s'inverser autour de la séquence de temps mort mais cela ne change rien du point de vue de l'approche par valeur moyenne.
La Figure 3.23-c montre le cas où l'on assiste à une transition d'une tension négative à une tension positive. Ce cas de figure se produit notamment lorsque la modulatrice change de signe. Dans ce cas, une tension nulle égale à la durée minimale de conduction est insérée entre deux temps morts.

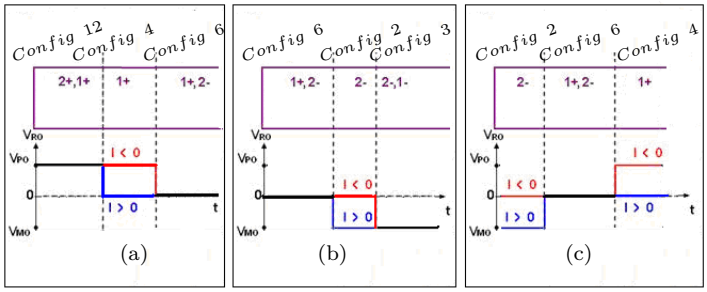


FIGURE 3.23 – Groupe de fenêtres V.

Groupe de fenêtres VI Le groupe VI rassemble des cas de figures pour lesquels les niveaux de tension nulle (Figure 3.24-a) , de tension négative (Figure 3.24-b) et de tension positive (Figure 3.24-c) sont entourés pendant une durée de conduction minimale de plusieurs séquences (notamment des temps morts et d'autres niveaux de tension).

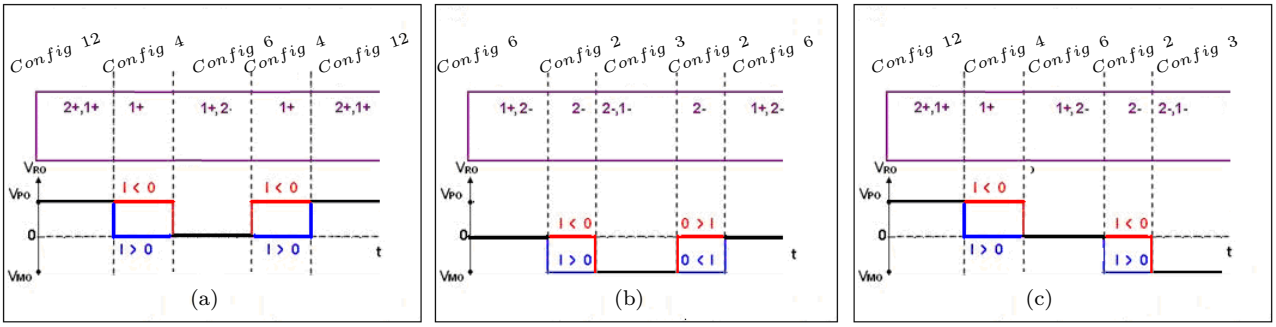


FIGURE 3.24 – Groupe de fenêtres VI.

Equations de groupe La table 3.5 consigne les expressions des tensions moyennes du convertisseur pour chaque groupe de fenêtres.

TABLE 3.5 – Expressions des tensions moyennes pour chaque groupe.

Groupes	signe($i(t)$)	Expression de la tension moyenne de sortie V_{RO}
Groupe I	+	(1-$\hat{G}_2$).V_{MO}
Groupe I	-	0
Groupe II	+	0
Groupe II	-	(1-$\hat{G}_3$).V_{PO}
Groupe III	+	V_{MO}
Groupe III	-	$\hat{G}_3$.V_{MO}
Groupe IV	+	$\hat{G}_1$.V_{PO}
Groupe IV	-	V_{PO}
Groupe V.a et Groupe VI.a	+	$\hat{G}_1$.V_{PO}
Groupe V.a et Groupe VI.a	-	(1-$\hat{G}_3$).V_{PO}
Groupe V.b et Groupe VI.b	+	(1-$\hat{G}_2$).V_{MO}
Groupe V.b et Groupe VI.b	-	$\hat{G}_4$.V_{MO}
Groupe V.c	+	(1-$\hat{G}_2$).V_{MO}
Groupe V.c	-	(1-$\hat{G}_3$).V_{PO}
Groupe VI.c	+	$\hat{G}_1$.V_{PO}+(1-$\hat{G}_2$).V_{MO}
Groupe VI.c	-	$\hat{G}_4$.V_{MO}+(1-$\hat{G}_3$).V_{PO}

Pendant les combinaisons litigieuses (cas des temps morts où la tension dépend du signe du courant), si le courant est nul, la tension appliquée est la tension dite de haute impédance V_{FEM}. Cette tension V_{FEM} dépend de la charge. Des compositions de fenêtre incluant une tension en mode haute impédance peuvent apparaître pendant les temps morts.

Si la durée minimale de conduction des interrupteurs commandés et la durée des temps morts sont supérieures au tiers du pas de calcul T_S, l'expression générale de la tension moyenne en sortie du convertisseur est alors :

$$\hat{V} = \hat{G}_1.V_{PO} + i_m.V_{PO}(1 - \hat{G}_1 - \hat{G}_3) + i_p.V_{MO}.(1 - \hat{G}_2 - \hat{G}_4) + \hat{G}_4.V_{MO} + i_z.V_{FEM}.(2 - \hat{G}_1 - \hat{G}_2 - \hat{G}_3 - \hat{G}_4) \quad (3.7)$$

avec :

- $i_p = 1$ si le courant $i(t)$ est positif, $i_p = 0$ sinon ;
- $i_m = 1$ si le courant $i(t)$ est négatif, $i_m = 0$ sinon ;
- $i_z = 1$ si le courant $i(t)$ est nul, $i_z = 0$ sinon.

L'équation (3.7) remplace alors l'expression en mode normal développée dans l'implémentation sur FPGA. La machine d'états séquentiels est gardée pour représenter les modes hautes impédances et redresseurs du convertisseur à trois niveaux de tension.

3.3.1.2 Validation du modèle du convertisseur NPC sur une charge *RLE*

Le modèle précédemment proposé est dans un premier temps validé sur une charge *RLE* comme illustré par la Figure 3.25. Le modèle est exécuté dans l'environnement Matlab à l'aide de la toolbox Simulink et comparé avec un modèle de référence (PLECS).

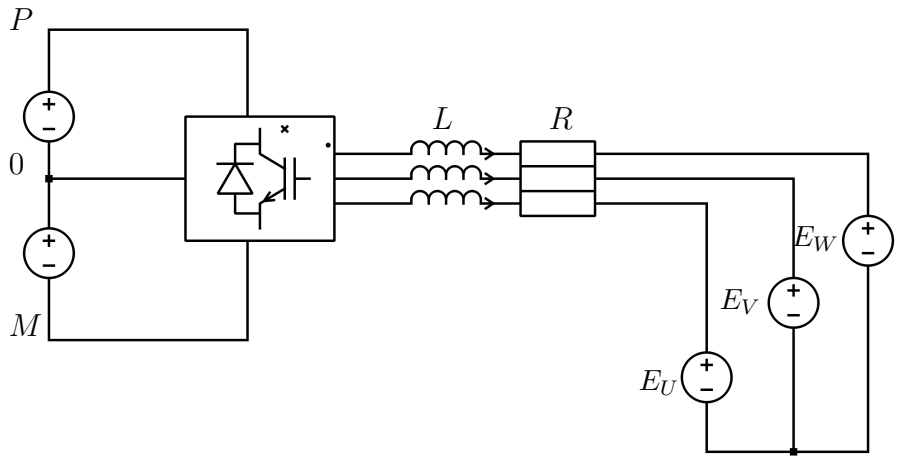


FIGURE 3.25 – Système de puissance utilisé pour valider le modèle du convertisseur trois niveaux NPC à impulsion de commande moyennée.

Les modèles sont exécutés avec un pas de $40\mu s$. La charge est de type RLE avec les paramètres suivants :

- $R = 1\Omega$
- $L = 100mH$
- $E_{UVW} = -2000V \begin{pmatrix} sin(2.\pi.60.t) \\ sin(2.\pi.60.t + \frac{2\pi}{3}) \\ sin(2.\pi.60.t + \frac{4\pi}{3}) \end{pmatrix}$ entre $t = 0.07s$ et $t = 0.1s$
- $E_{UVW} = -4000V \begin{pmatrix} sin(2.\pi.60.t) \\ sin(2.\pi.60.t + \frac{2\pi}{3}) \\ sin(2.\pi.60.t + \frac{4\pi}{3}) \end{pmatrix}$ entre $t = 0.1s$ et $t = 0.15s$

Le profil des FEMs E_{UVW} est choisi de façon à pouvoir tester les modes haute impédance et redresseur.

L'onduleur est un onduleur de $3.3kV$ avec une demi-tension de bus nominale de $2333V$. La fréquence de la porteuse est de $1250Hz$, les durées minimales de conduction et de temps morts sont chacunes égales à $20\mu s$. Les résultats du modèle proposé avec un pas de $40\mu s$ sont comparés avec ceux du modèle continu simulé sous PLECS.

Pour tester le mode de fonctionnement normal, on active les signaux de commande de gâchette à t=0.02s et on impose un taux de modulation de 0.9. Les signaux de commande sont ensuite désactivés à t=0.05s. La Figure 3.26 et la Figure 3.27 montrent les formes d'ondes de tension et de courant pour ce mode de fonctionnement.

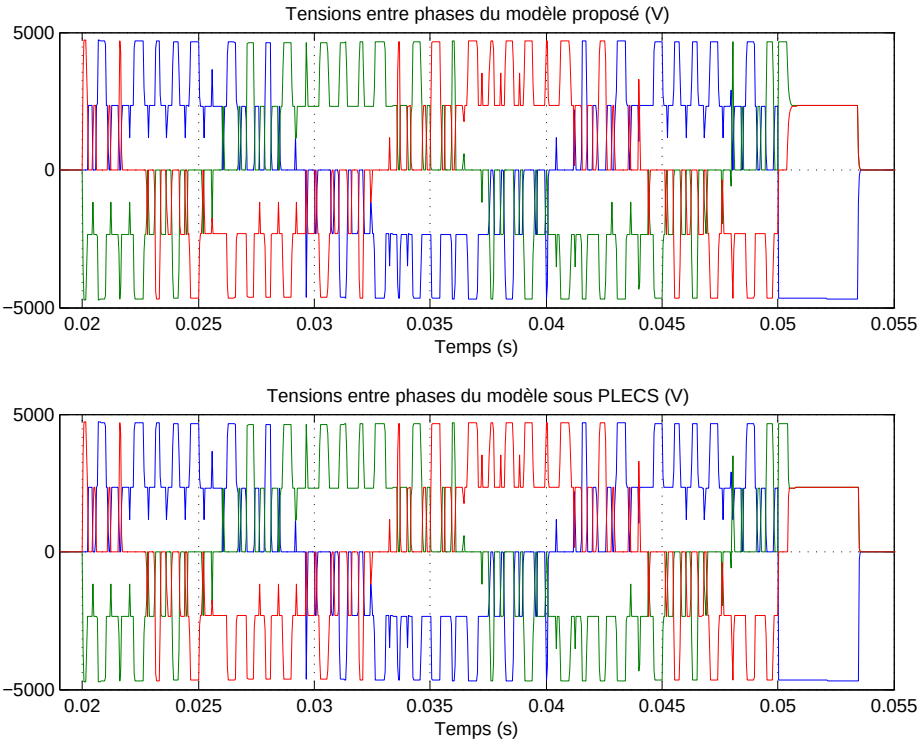


FIGURE 3.26 – Tensions entre phases du convertisseur en mode normal.

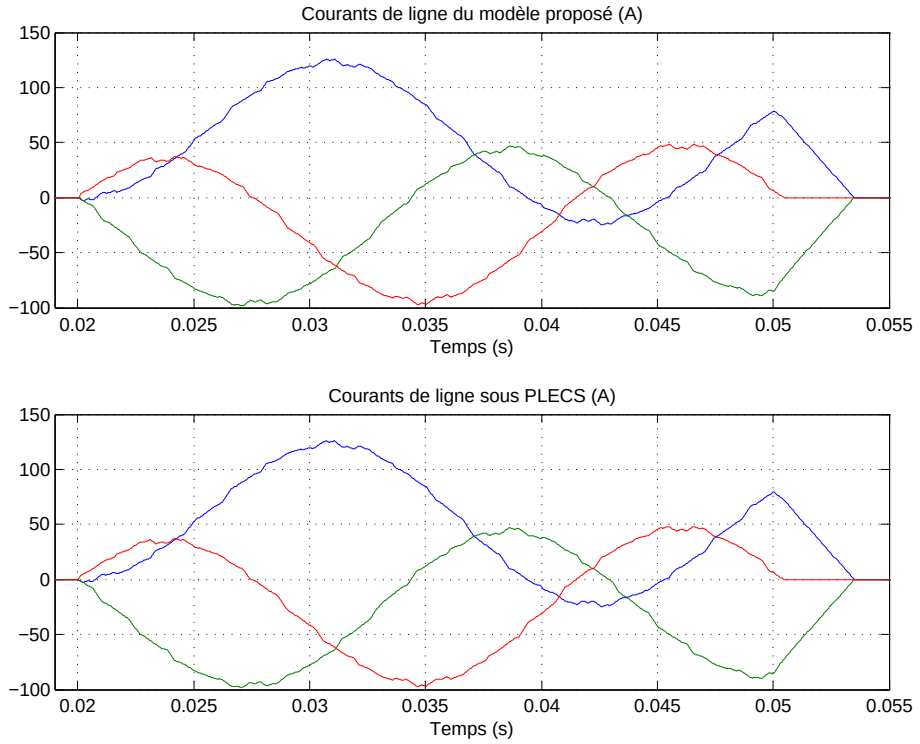


FIGURE 3.27 – Courants de ligne en mode normal.

On teste ensuite le mode haute impédance. A partir de t=0.05s jusqu'à t=0.1s, aucun signal de commande n'est envoyé aux interrupteurs commandés. Les tensions en sorties du pont correspondent au FEM de la charge(Figure 3.28). Les courants restent nuls (Figure 3.29) car les tensions entre phases n'excèdent pas la tension totale V_{PM} du bus.

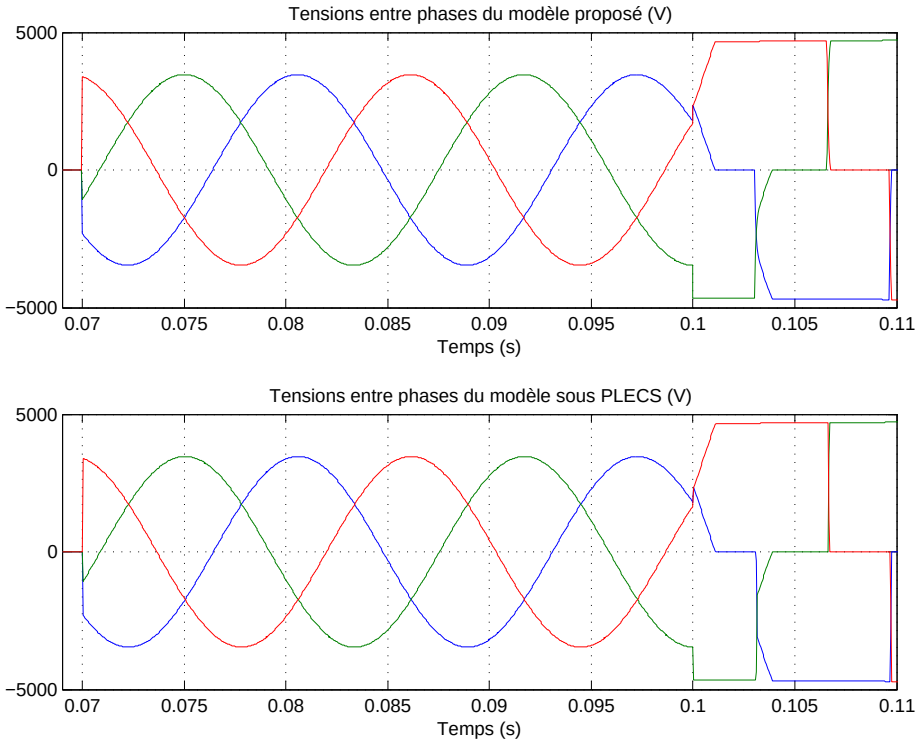


FIGURE 3.28 – Tensions entre phases du convertisseur en mode haute impédance.

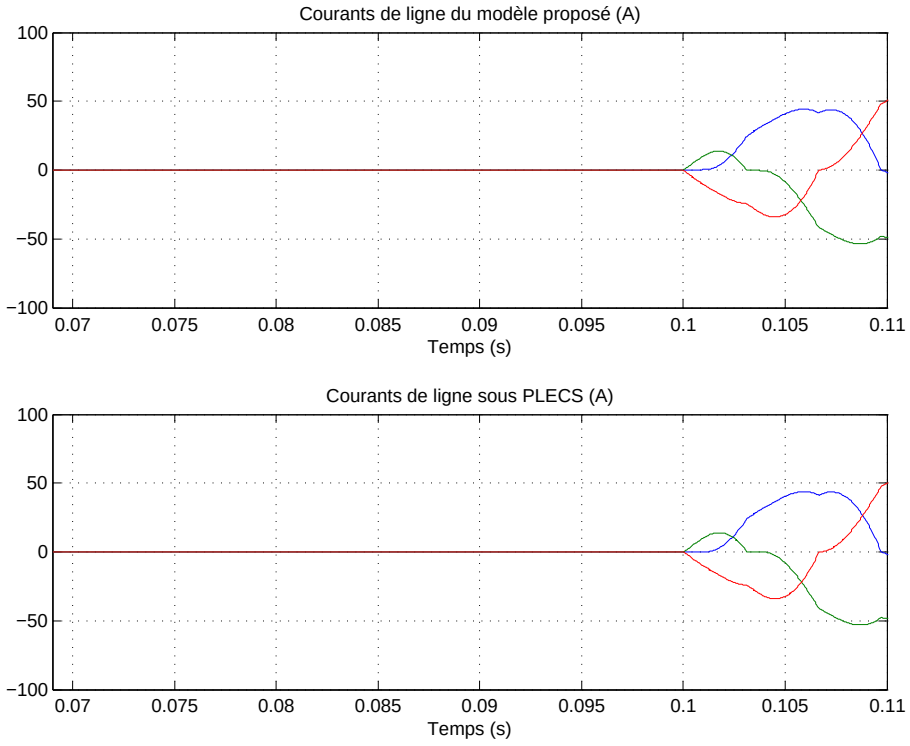


FIGURE 3.29 – Courants de ligne en mode haute impédance.

Entre t=0.1s et t=0.15s, les tensions entre phases excèdent la tension du bus et on observe un phénomène de redressement. Le convertisseur se comporte comme un redresseur.

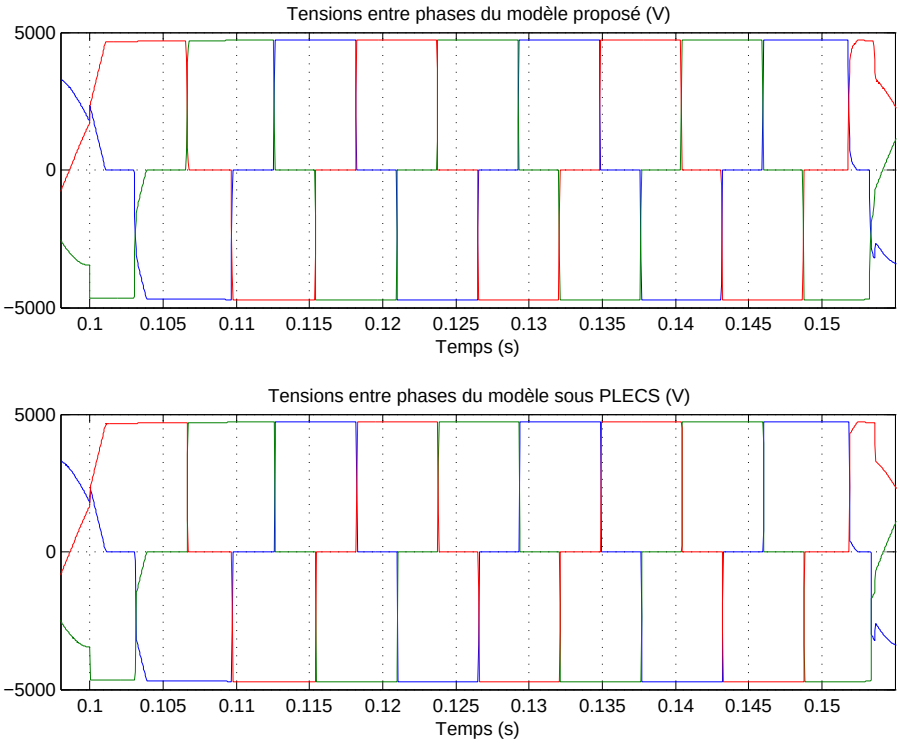


FIGURE 3.30 – Tensions entre phases du convertisseur en mode redresseur.

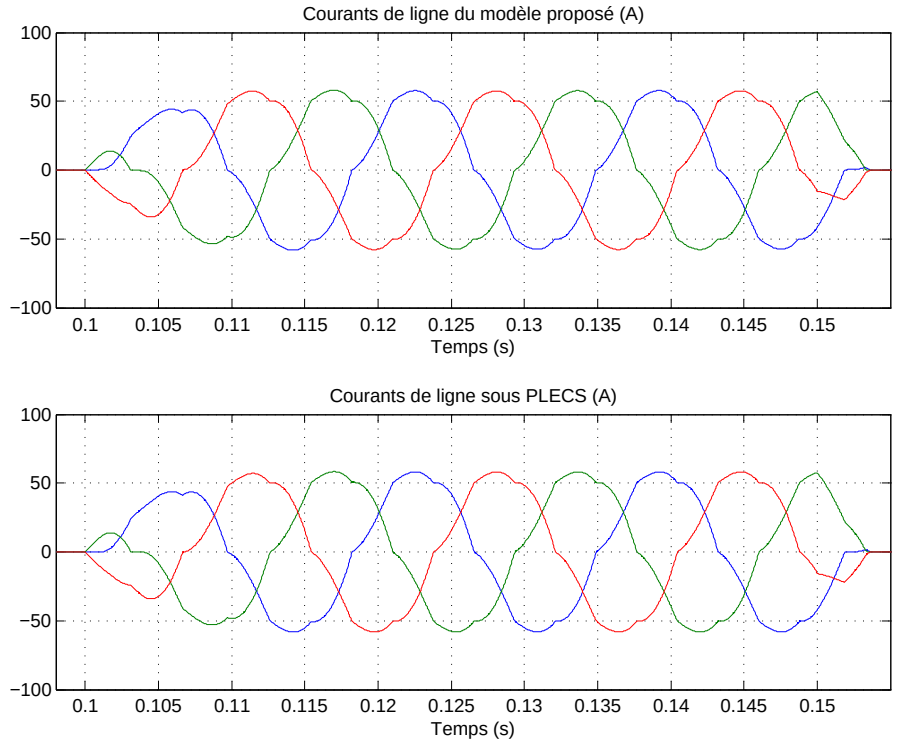


FIGURE 3.31 – Courants de ligne en mode redresseur.

La Figure 3.32 illustre les limites de la modélisation par analyse exhaustive des fenêtres de tension précédemment décrites. Nous remarquons sur cette figure le changement de signe du courant I_W de la phase W. Le modèle continu détecte avec bonne précision l'instant d'annulation du courant. Le modèle tournant à $40\mu s$ ne voit pas ce phénomène comme nous avions supposé que le courant reste constant durant un pas de calcul. De plus, on considère que le courant est nul dans la bande morte de courant $\delta_I = 2A$. L'erreur sur le courant I_W induit une erreur sur la tension V_W. Une erreur sur la tension V_W se manifeste seulement sur les tensions de ligne U_{VW} et U_{WU}. La tension U_{UV} n'est pas impactée car cette tension ne dépend pas du courant I_W. Les courants I_U et I_V sont en dehors de la bande $\delta_I = 2A$.

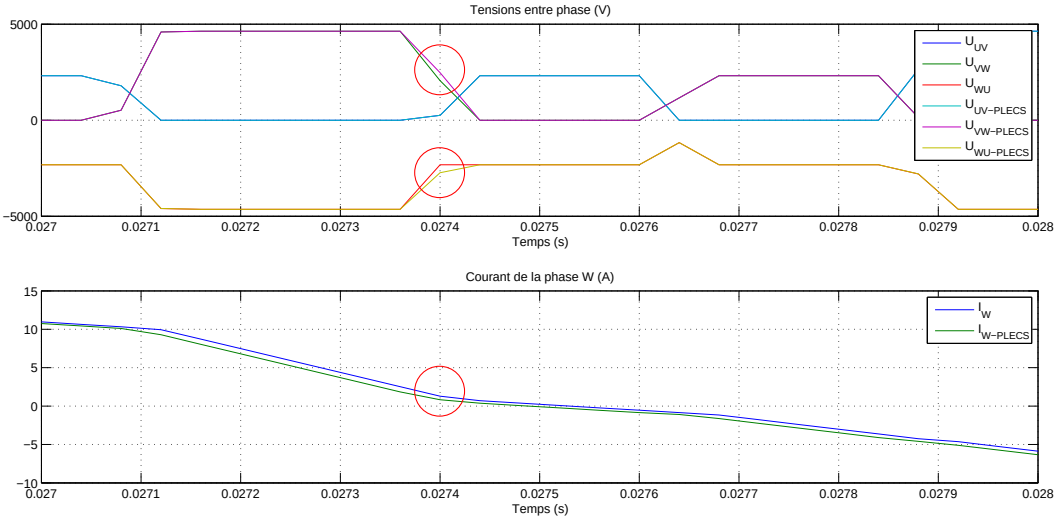


FIGURE 3.32 – Différence entre le modèle proposé et le modèle continu sous PLECS.

Le bloc onduleur étant implémenté en blocs élémentaires Simulink, il n'est pas possible de revenir en arrière pour recalculer l'instant d'annulation du courant. On utilise alors une bande de courant que l'on suppose nulle et on surveille les changements de signe du courant.

Ce problème n'apparaît pas si l'onduleur et la charge sont fondus dans un seul script Matlab embarqué (Embedded Matlab Function). Dans ce cas, on peut retourner en arrière pour recalculer l'instant d'annulation du courant à l'aide d'une boucle de programmation.

La représentation graphique de l'onduleur et sa charge par des blocs élémentaires Simulink fournit une meilleure lisibilité au détriment de la modélisation. C'est la raison pour laquelle certains logiciels utilisent une description nodale du système et calculent seulement le modèle global à partir de cette description.

3.3.1.3 Cas d'une machine asynchrone alimentée par deux onduleurs

Nous allons maintenant étudier la cas d'une machine asynchrone potentiellement alimentée par deux convertisseurs NPC (Figure 3.14).

La machine peut être alimentée par un seul convertisseur ou deux convertisseurs en même temps.

Comme le changement de topologie est purement électrique, les équations mécaniques de la machine restent les mêmes.

Dans cette étude, on considère que les tolérances des inductances (L) et des résistances (R) sont suffisamment petites. On considère que ces résistances et ces inductances sont parfaitement égales pour chaque phase et pour chaque convertisseur.

Un convertisseur sera dit **non actif** si aucun de ses bras ne conduit ou si il est déconnecté de la machine, autrement il est dit **actif**.
Dans le cas où un seul convertisseur est actif, on définit la tension **active** comme la tension du convertisseur qui est actif. L'autre convertisseur voit entre ses phases les tensions entre les nœuds des inductances connectées à la machine. La tension active est la tension utilisée pour calculer les états de la machine.

3.3.1.3.1 Modèle de la charge dans le cas où un seul onduleur est actif

Le cas où la machine asynchrone est alimentée par un seul convertisseur peut se produire dans le cas d'un défaut sur l'autre convertisseur. Dans ce cas, le convertisseur défectueux est

simplement déconnecté du reste du système.

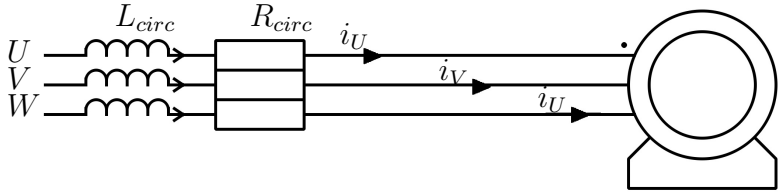


FIGURE 3.33 – Machine asynchrone alimentée par un seul onduleur.

Le système de puissance est dans ce cas équivalent à une machine à laquelle la résistance statorique et l'inductance de fuite statorique ont été augmentées des quantités R_{circ} et L_{circ}.

Nous reprenons ici l'équation de l'annexe D (D.56) en augmentant les résistances statoriques R_S et les inductances cycliques statoriques l_{CS} de l'inductance L_{circ} et de la résistance R_{circ} initialement destinées à limiter les courants de circulation en mode d'alimentation à deux convertisseurs :

$$\frac{d}{dt}\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} = A_{mas,I} \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{X_{mas}} + B_{mas,I} \underbrace{\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}}_{V_{mas,act}} \tag{3.8}$$

$$A_{mas,I} = \begin{pmatrix} ka_{11} & ka_{12} & ka_{14} & ka_{15} \\ -ka_{12} & ka_{11} & -ka_{15} & ka_{14} \\ ka_{41} & ka_{42} & ka_{44} & ka_{45} \\ -ka_{42} & ka_{41} & -ka_{45} & ka_{44} \end{pmatrix}$$

$$B_{mas,I} = \begin{pmatrix} \frac{l_{CR}}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2} \\ \frac{m_{SR}}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2} & 0 & \frac{(l_{CS}+L_{circ})}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} & 0 \\ 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2} & 0 & \frac{(l_{CS}+L_{circ})}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} \end{pmatrix}$$

$$ka_{11} = -\frac{l_{CR}(R_S+R_{circ})}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} \qquad ka_{41} = \frac{m_{SR}(R_S+R_{circ})}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2}$$

$$ka_{12} = \frac{{m_{SR}}^2\dot{\theta}_R - l_{CR}(l_{CS}+L_{circ})\dot{\theta}_S}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2} \qquad ka_{42} = \frac{-(l_{CS}+L_{circ})m_{SR}\dot{\theta}_R + (l_{CS}+L_{circ})m_{SR}\dot{\theta}_S}{-l_{CR}(l_{CS}+L_{circ})+{m_{SR}}^2}$$

$$ka_{14} = \frac{m_{SR}R_R}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} \qquad ka_{44} = -\frac{(l_{CS}+L_{circ})R_R}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2}$$

$$ka_{15} = \frac{-l_{CR}m_{SR}\dot{\theta}_R + l_{CR}m_{SR}\dot{\theta}_S}{l_{CR}(l_{CS}+L_{circ})-{m_{SR}}^2} \qquad ka_{45} = \frac{l_{CR}(l_{CS}+L_{circ})\dot{\theta}_R - {m_{SR}}^2\dot{\theta}_S}{l_{CR}(l_{CS}+L)-{m_{SR}}^2}$$

Comme un tour mécanique correspond à p période(s) électrique(s) :

$$\theta_R = p\theta_M \tag{3.9}$$

Dans les équations précédentes :

– θ_R est la pulsation électrique équivalente de la partie mécanique $(rad.s^{-1})$;

– θ_M est la pulsation de rotation mécanique avec $\theta_M = 2\pi f_M$ $(rad.s^{-1})$;
– f_M est la vitesse de rotation mécanique (Hz) ;
– p est le nombre de paires de pôles.

Dans ce cas, les courants en sortie du convertisseur actif, les courants de la machine et les courants des inductances du convertisseur actif sont égaux. Le convertisseur non actif a ses courants nuls.

3.3.1.3.2 Modèle de la charge dans le cas de deux onduleurs actifs

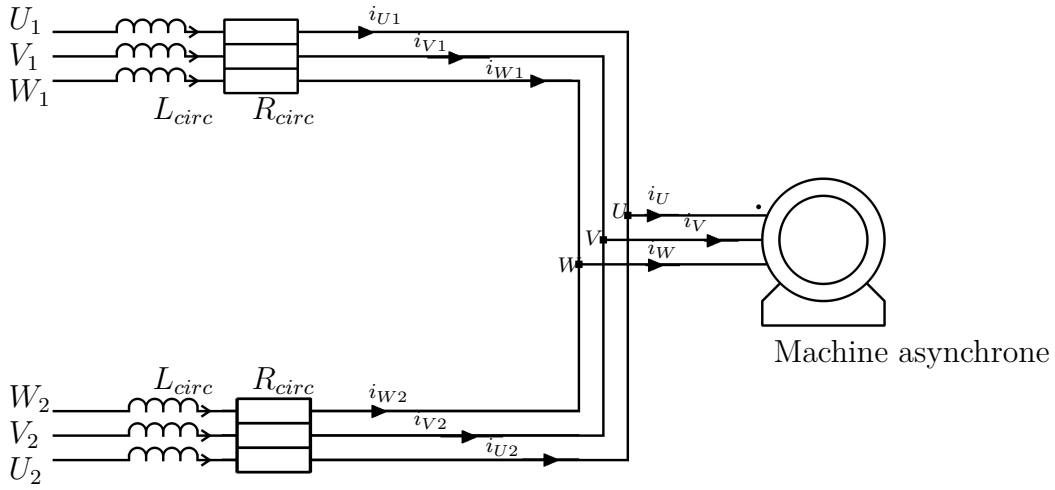


FIGURE 3.34 – Machine asynchrone alimentée par deux onduleurs.

La Figure 3.34 décrit la machine asynchrone alimentée par deux convertisseurs triphasés. La Figure 3.35 présente le schéma équivalent pour la phase U. Ce schéma équivalent est obtenu en recalculant les sources de tensions de Thévenin et l'impédance de Thévenin.

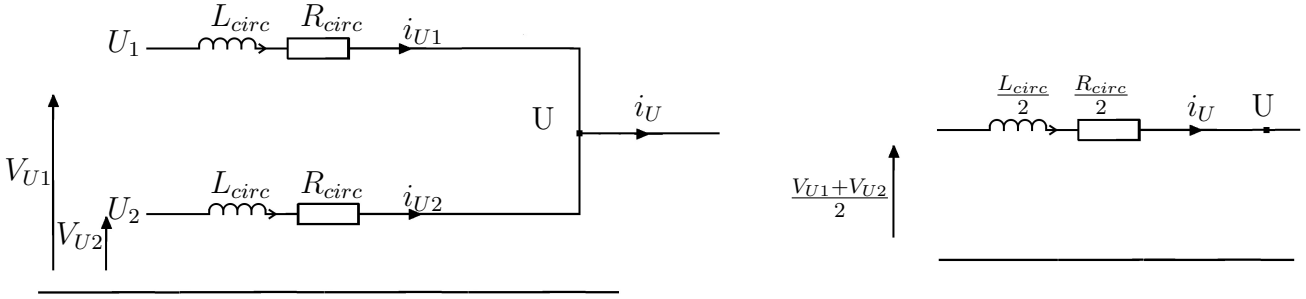


FIGURE 3.35 – Schéma de base pour le calcul des courants de ligne des convertisseurs.

La machine équivalente est une charge de nature courant vue des convertisseurs lorsque ces derniers sont en mode de fonctionnement normal (les convertisseurs imposent leurs tensions de sortie). Le circuit de limitation des courants entre les deux convertisseurs (les résistances R_{circ} et les inductances L_{circ} en sortie de chaque convertisseur de tension) peut-être considéré comme un diviseur de tension avec un courant débité au point de connexion vers la machine

(Figure 3.35). L'équation d'état régissant le système est dans ce cas :

$$\frac{d}{dt}\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} = A_{mas,II}\underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{X_{mas}} + B_{mas,II}.\underbrace{\frac{1}{2}.\left(\begin{pmatrix} V_{DQS,1} \\ V_{DQR} \end{pmatrix} + \begin{pmatrix} V_{DQS,2} \\ V_{DQR} \end{pmatrix}\right)}_{V_{mas}} \tag{3.10}$$

$$A_{mas,II} = \begin{pmatrix} kb_{11} & kb_{12} & kb_{14} & kb_{15} \\ -kb_{12} & kb_{11} & -kb_{15} & kb_{14} \\ kb_{41} & kb_{42} & kb_{44} & kb_{45} \\ -kb_{42} & kb_{41} & -kb_{45} & kb_{44} \end{pmatrix}$$

$$B_{mas,II} = \begin{pmatrix} \frac{l_{CR}}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} \\ \frac{m_{SR}}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} & 0 & \frac{(l_{CS}+\frac{L_{circ}}{2})}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & 0 \\ 0 & \frac{m_{SR}}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} & 0 & \frac{(l_{CS}+\frac{L_{circ}}{2})}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} \end{pmatrix}$$

$$\begin{aligned} kb_{11} &= -\frac{l_{CR}(R_S+\frac{R_{circ}}{2})}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & kb_{41} &= \frac{m_{SR}(R_S+\frac{R_{circ}}{2})}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} \\ kb_{12} &= \frac{{m_{SR}}^2\dot{\theta}_R - l_{CR}(l_{CS}+\frac{L_{circ}}{2})\dot{\theta}_S}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} & kb_{42} &= \frac{-(l_{CS}+\frac{L_{circ}}{2})m_{SR}\dot{\theta}_R + (l_{CS}+\frac{L_{circ}}{2})m_{SR}\dot{\theta}_S}{-l_{CR}(l_{CS}+\frac{L_{circ}}{2})+{m_{SR}}^2} \\ kb_{14} &= \frac{m_{SR}R_R}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & kb_{44} &= -\frac{(l_{CS}+\frac{L_{circ}}{2})R_R}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} \\ kb_{15} &= \frac{-l_{CR}m_{SR}\dot{\theta}_R + l_{CR}m_{SR}\dot{\theta}_S}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} & kb_{45} &= \frac{l_{CR}(l_{CS}+\frac{L_{circ}}{2})\dot{\theta}_R - {m_{SR}}^2\dot{\theta}_S}{l_{CR}(l_{CS}+\frac{L_{circ}}{2})-{m_{SR}}^2} \end{aligned}$$

Le modèle d'état précédemment décrit, comparé au modèle d'une machine asynchrone seule montre que la machine asynchrone alimentée par deux convertisseurs via des résistances R_{circ} et des inductances L_{circ} de limitation de courant de circulation est équivalente à une machine asynchrone dont les résistances statoriques et les inductances statoriques ont été respectivement augmentées de $\frac{R_{circ}}{2}$ et $\frac{L_{circ}}{2}$.

L'onduleur équivalent est alors un onduleur dont la tension est la valeur moyenne des deux onduleurs actifs.

3.3.1.3.3 Courants de ligne de chaque convertisseur

Dans cette partie, nous allons calculer les courants traversant les inductances L_{circ} et les résistances R_{circ}. Ces courants sont nécessaires pour calculer les tensions de sortie de chaque convertisseur de tension alimentant la machine.

A partir de la Figure 3.35, on déduit le courant de chaque phase du convertisseur (par exemple i_{U1} et i_{U2}) en fonction des tensions des convertisseurs (par exemple v_{U1} et v_{U2}) et du courant de la machine équivalente (par exemple iu). Comme la transformation que nous avions choisie conserve les impédances, on peut écrire, en utilisant la variable de Laplace :

$$i_{D1} = \frac{V_{D1} - V_{D2}}{2\left(R_{circ} + L_{circ}.s\right)} + \frac{i_{DS}}{2} \tag{3.11}$$

$$i_{Q1} = \frac{V_{Q1} - V_{Q2}}{2\left(R_{circ} + L_{circ}.s\right)} + \frac{i_{QS}}{2} \tag{3.12}$$

$$i_{D2} = \frac{V_{D2} - V_{D1}}{2\left(R_{circ} + L_{circ}.s\right)} + \frac{i_{DS}}{2} \tag{3.13}$$

$$i_{Q2} = \frac{V_{Q2} - V_{Q1}}{2\left(R_{circ} + L_{circ}.s\right)} + \frac{i_{QS}}{2} \tag{3.14}$$

3.3.1.3.4 Courants de circulation

Comme nous l'avons vu dans les équations précédentes, les courants en sortie de chaque convertisseur comportent :

- une composante $\frac{I_{DQS}}{2}$ qui circule vers la machine équivalente ;
- une composante $\frac{V_{DQ1}-V_{DQ2}}{2(R_{circ}+L_{circ}.s)}$ qui circule entre les deux convertisseurs, appelée courant de circulation.

Comme un changement de configuration peut se faire à chaud (les changements entre une configuration avec deux onduleurs et une configuration avec un seul onduleur peut s'effectuer sans arrêter le moteur et sans attendre l'annulation des courants), la réinitialisation des variables d'état (de la nouvelle équation d'état définie par l'équation (3.8) ou l'équation (3.10)) est nécessaire lorsque l'on passe d'un onduleur actif à deux onduleurs actifs (et vice versa).

3.3.1.3.5 Procédure de validation

Le but de la validation suivante est de vérifier le comportement du modèle du système de puissance de la Figure 3.34. Ce modèle a été développé en vue de son implantation sur la plateforme de simulation temps réel RT-LAB.

Le modèle des convertisseurs trois niveaux utilisant les impulsions de commande moyennées est conçu à partir de blocs élémentaires Simulink. Ces mêmes blocs Simulink sont ensuite compilés pour générer un code temps réel pour le simulateur RT-LAB. Les équations régissant les variables d'état de la machine asynchrone (équations (3.8) et (3.10)) ainsi que le calcul des courants de chaque convertisseur (équations (3.11) à (3.14) si les deux convertisseurs sont actifs) sont implémentés par des fonctions de transfert.

3.3.1.3.6 Résultats

Les paramètres du système étudié pour la validation sont consignés dans la table 3.6.

TABLE 3.6 – Paramètres du système étudié.

Paramètres électriques	Valeurs
V_{PO}	2620V
V_{MO}	-2620V
l_{CS}	0.72122mH
l_{CR}	0.70668mH
m_{SR}	0.69092mH
R_S	$19m\Omega$
R_R	$15m\Omega$
R_{circ}	$1.05m\Omega$
L_{circ}	0.2mH

Paramètres mécaniques	Valeurs
p	6
J	$59000kg.m^2$
$f_{\dot{\theta}}$	1200 $Nm/rad/s^2$

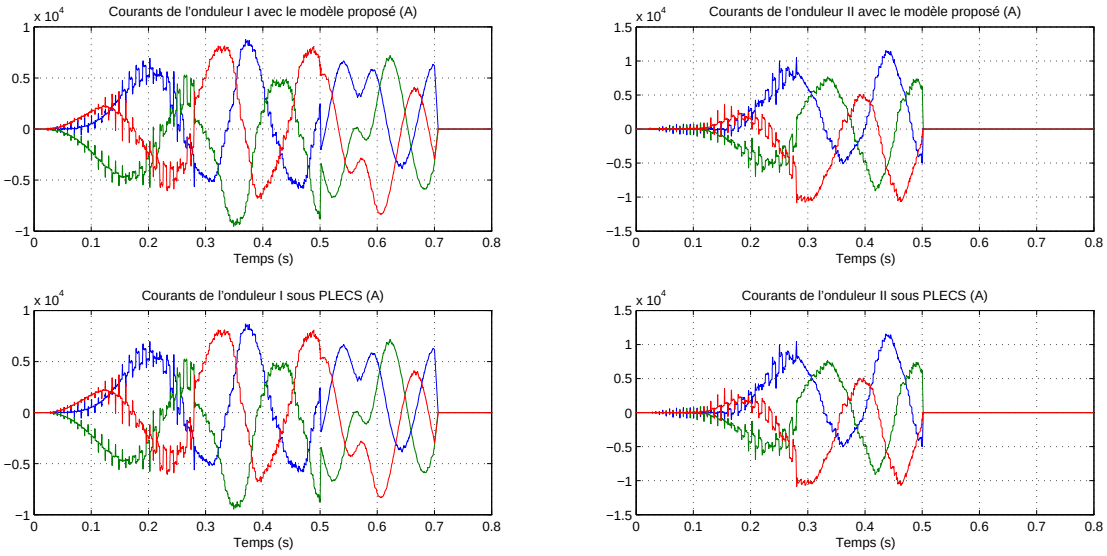


FIGURE 3.36 – Comparaison du modèle proposé pour la machine asynchrone alimentée par un ou deux onduleurs avec le modèle de référence (PLECS).

Les impulsions de commande des interrupteurs ont été décalées volontairement pour vérifier le comportement du système. Les deux onduleurs sont commandés avec une MLI avec injection de signal homopolaire carré (permettant de réduire et de partager les pertes sur tous les interrupteurs du bras à trois niveaux de tension) entre t=0s et t=0,3s. Le signal homopolaire est remplacé ensuite par un signal contenant les harmoniques 3 et multiples. L'onduleur II est ensuite rendu inactif à t=0,5s. L'onduleur I continue à être actif jusqu'à t=0,7s.

La Figure 3.36 présente les formes d'ondes de courant des deux onduleurs soumis au scénario précédemment décrit. Le décalage des impulsions de commande induit davantage d'oscillations sur les courants dans le cas de l'injection d'un signal homopolaire carré, puisqu'on décale des signaux à front raide. Les pics de courant sont moins importants dans le cas d'un signal homopolaire contenant les harmoniques 3 et multiples de 3. Le deuxième type d'homopolaire ne comporte pas de discontinuité.

L'approche utilisant les valeurs moyennes de pulses aboutit à une expression relativement simple des tensions en sortie d'onduleur. Trois modèles d'onduleurs de ce type avec leurs charges (réseau, transformateur, moteurs ...) peuvent tourner avec un pas de 30μs sur la carte OP5600 de RT-LAB. La carte OP5600 que nous utilisons embarque un processeur à 8 cœurs de 3.3GHz. Elle gère à travers des cartes modulaires d'entrées/sorties logiques horodatées ou non et d'entrées/sorties analogiques. Cependant, le modèle ainsi construit est seulement valide quand la durée des temps morts et de la durée minimale de conduction sont inférieures au tiers du pas de calcul T_S.

Dans les applications qui nous concernent, ces durées minimales de conduction sont de l'ordre de 20μs. Dans ce cas, le pas de calcul du simulateur doit être de l'ordre de 40μs. Cette contrainte peut-être atteinte sur des simulateurs dédiés comme RT-LAB. Cependant, si l'on désire, comme dans le cas de la VIL, utiliser le même type de matériel que celui utilisé pour le contrôle, l'approche par valeurs moyennes des impulsions ne convient pas puisque la durée minimale de conduction et la durée des temps morts seront inférieures au tiers du pas de calcul. Dans ce cas, on risque d'avoir plus de 3 évènements, et donc des cas autres que ceux étudiés avec les groupes de fenêtres précédentes.

3.3.2 Simulateur sur FPGA/Processeur lent

On désire maintenant étudier la faisabilité d'un développement d'un système de validation temps réel sur la même base matérielle que le contrôle. On reprendra par la suite le même système de puissance que celui étudié dans le cadre de la VIL (Figure 3.37). Ce système est constitué d'un ensemble machine asynchrone et d'un convertisseur trois niveaux de type NPP.

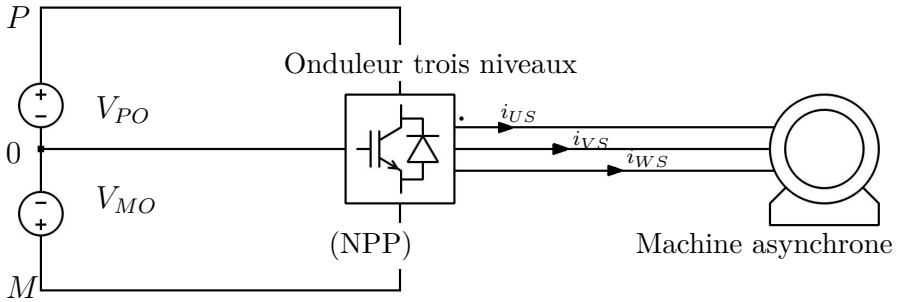


FIGURE 3.37 – Système de puissance simulé dans le cadre d'une VIL repris pour la VIM sur FPGA/Processeur lent.

Nous voyons à la Figure 3.38 que la partie simulateur temps réel lors de la VIM est constitué du même matériel que le contrôleur industriel.

L'implémentation sur la même base matérielle réduit les coûts de développement annexes comme les suivis de versions, les programmes de gestion des ressources d'entrée/sortie. En plus, l'utilisation de la même base matérielle augmente le potentiel d'utilisation en masse de la validation.

La difficulté majeure d'un tel développement est que la base matérielle dédiée au contrôle ne peut atteindre les mêmes performances temporelles que les simulateurs que l'on rencontre dans le commerce (le contrôleur fonctionne avec des périodes de tâche supérieures à $380\mu s$ alors que les simulateurs temps réels actuels fonctionnent à environ $50\mu s$).

Dans le cas où la durée des temps morts et la durée minimale de conduction sont inférieures au tiers du pas de calcul, la méthodologie d'interpolation évènementielle qui a été expliquée dans le cas d'une VIL doit être reprise pour la VIM. La modélisation du convertisseur par les impulsions moyennées ne convient plus dans ce cas.

Dans les approches classiques, l'horodatage se fait sur les signaux de commande. Le processeur effectue ensuite des sauts internes en recalculant à chaque fois le niveau de tension en fonction des signaux de commande datés.

Dans notre cas, nous connaissons déjà le modèle du convertisseur. Ce modèle est le même que celui qui a été utilisé pour une implémentation complète sur unique FPGA. Nous proposons donc plutôt d'implémenter le modèle du convertisseur sur FPGA et d'horodater la tension sur FPGA. L' horodatage des tensions à la place des signaux de commande d'interrupteurs réduit le nombre de données échangées entre le FPGA et le processeur. En effet, la communication entre le FPGA et le processeur consomme une grande partie des temps de cycle des tâches processeurs. Par exemple, un modèle de machine asynchrone avec un onduleur trois niveaux sur un processeur $2.5GHz$ peut s'exécuter en $150\mu s$ (dont $30\mu s$ pour le modèle et $120\mu s$ pour la communication) en échangeant 128 mots de 16 bits sur une liaison ETCAT.

Par ailleurs, si nous utilisons le même format que le format des instants de commutation envoyé par l'UL à la UGES (équation 2.77 - chapitre 2), on peut réutiliser le modèle du système de puissance initialement développé pour la VIL pour effectuer de la VIM. Le travail à faire est

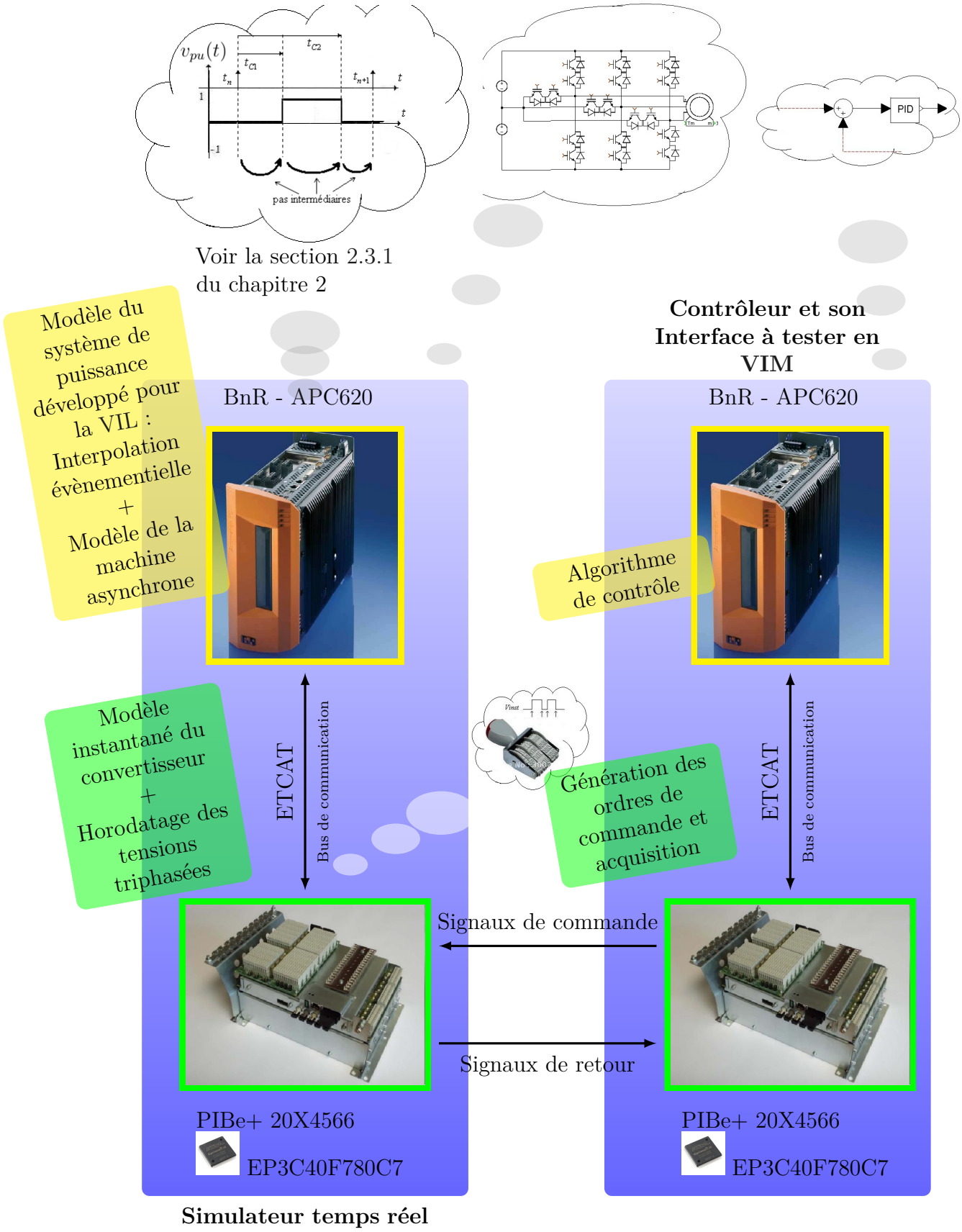


FIGURE 3.38 – VIM par horodatage de la tension sur FPGA.

donc de réimplementer le modèle du convertisseur sur FPGA et mettre les mots obtenus par horodatage de la tension sous le même format que l'équation 2.77.

Nous allons donc présenter en détails le principe d'horodatage de la tension.

3.3.2.1 Horodatage de la tension

On reprend le modèle de l'onduleur trois niveaux NPC précédemment traité pour une implémentation sur cible FPGA (voir la section 3.2.1).

D'abord, on choisit un codage sur deux bits noté V_{inst} des différents niveaux de tensions onduleurs :

- '11' pour une tension positive ;
- '00' pour une tension négative ;
- '01' pour une tension nulle ;
- '10' pour une tension en mode haute impédance ou inconnue.

V_{inst} donne donc une information sur la tension instantanée.
T_{inst} est le temps en tic horloge (l'horloge principale de l'UGES est de $33ns$) si on prend comme origine de temps le début du pas de calcul.

Un mot d'horodatage est donc formé comme décrit par la table 3.7.

TABLE 3.7 – Composition des mots de 32 bits d'horodatage.

Niveau de tension : b_{31} b_{30}	Instant de commutation en tic horloge : b_{29} .. b_0
V_{dated}	T_{dated}

Un mot (V_{inst} | T_{inst}) de 32 bits est formé à partir du niveau de tension codé V_{inst} et du temps T_{inst}. Il est interprété comme suit :
"La tension instantannée a la valeur correspondant à V_{inst} (par exemple V_{PO} si $V_{inst} = 11$) à $t = T_{inst}$."

Comme cela est présenté à la Figure 3.39, le *Timer* est un compteur de 30 bits. Sa sortie T_{inst} de 30 bits est utilisée pour compter le temps entre deux pas de calcul du processeur et contribue directement à l'affectation des 30 bits de poids faible (V_{dated}) du mot d'horodatage.

Un signal IT est reçu après un échange de données entre la UL et la UGES. Ce signal IT réinitialise le contenu de ce compteur (*Timer*) à chaque fois que le temps correspondant à un pas de calcul processeur est écoulé.

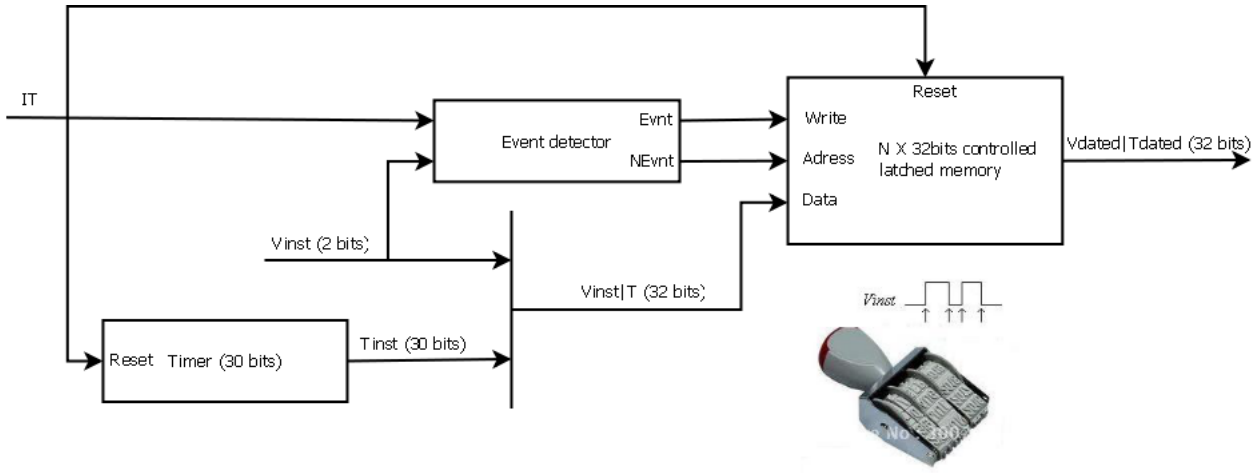


FIGURE 3.39 – Circuit d'horodatage des tensions d'un convertisseur trois niveaux.

Un détecteur d'évènements *Event detector* surveille le signal de sortie V_{inst} du modèle de l'onduleur et génère un évènement sur le signal binaire *Evnt* en cas de changement de V_{inst}. Le signal *Evnt* est utilisé pour déclencher une écriture en mémoire de la valeur de $(V_{inst} \mid T_{inst})$.

Le nombre d'évènements *NEvnt* est compté par le bloc *Event detector* et mis à disposition en sortie avec le signal *Evnt*.

Comme précédemment discuté, le signal *Evnt* déclenche un enregistrement dans la mémoire de N mots de 32 bits (*controlled latched memory*) à l'adresse localement définie par le nombre d'évènement *NEvnt*. La donnée enregistrée $(V_{dated} \mid T_{dated})$ correspond à la dernière valeur de $(V_{inst} \mid T_{inst})$ qui contient l'information sur le dernier niveau de tension et l'instant d'apparition de l'évènement.

La zone mémoire est contrôlée de façon à éviter les conflits de lecture et d'écriture, soit en passant par des mémoires tampons, soit par un buffer circulaire. La profondeur de la mémoire (N) détermine le nombre maximal d'évènements que l'on puisse autoriser. Dans notre cas, nous avions choisi $N = 4$ pour être compatible avec le modèle développé en VIL (voir page 83 - chapitre 2).

Par ailleurs, on génère systématiquement et volontairement un évènement sur *Evnt* par le signal IT. Cela permet, en cas de remplissage de la zone mémoire dédiée, de reprendre la bonne valeur de tension au prochain pas de calcul.

En sortie du circuit d'horodatage, le mot $(V_{dated} \mid T_{dated})$ est interprété comme suit :
"La tension est passée de sa valeur précédente à la valeur correspondant à V_{dated} (même interprétation que V_{inst}) à $t = T_{dated}$."

TABLE 3.8 – Taux d'utilisation des ressources logiques.

Cyclone III EP3C40F780C7	Taux d'utilisation
Éléments logiques	4% (1665 / 39600)
Pins	84% (450/536)
Mémoires	0% (0/1161216 bits)
PLLs	0% (0/4)

La table 3.8 rassemble les taux d'utilisation du FPGA utilisé (Cyclone III). L'horodatage utilise peu de ressources logiques. Le nombre de pins semble important. Les sorties du circuit d'horodatage (tensions horodatées) seront en réalité réenvoyées par Ethercat au processeur dédié à la simulation temps réel et aucune pin du FPGA ne sera utilisée pour affecter les sorties.

3.3.2.2 Validation de l'horodatage

Le modèle du système de puissance de la Figure 3.2 a déjà été validé au deuxième chapitre portant sur la VIL. Ce modèle utilise les algorithmes d'interpolation évènementielle.
De même, le modèle instantané de l'onduleur trois niveaux sur FPGA a été traité lors de l'étude du simulateur sur une unique cible FPGA. Il reste donc à valider le circuit d'horodatage de la tension, précédemment décrit.

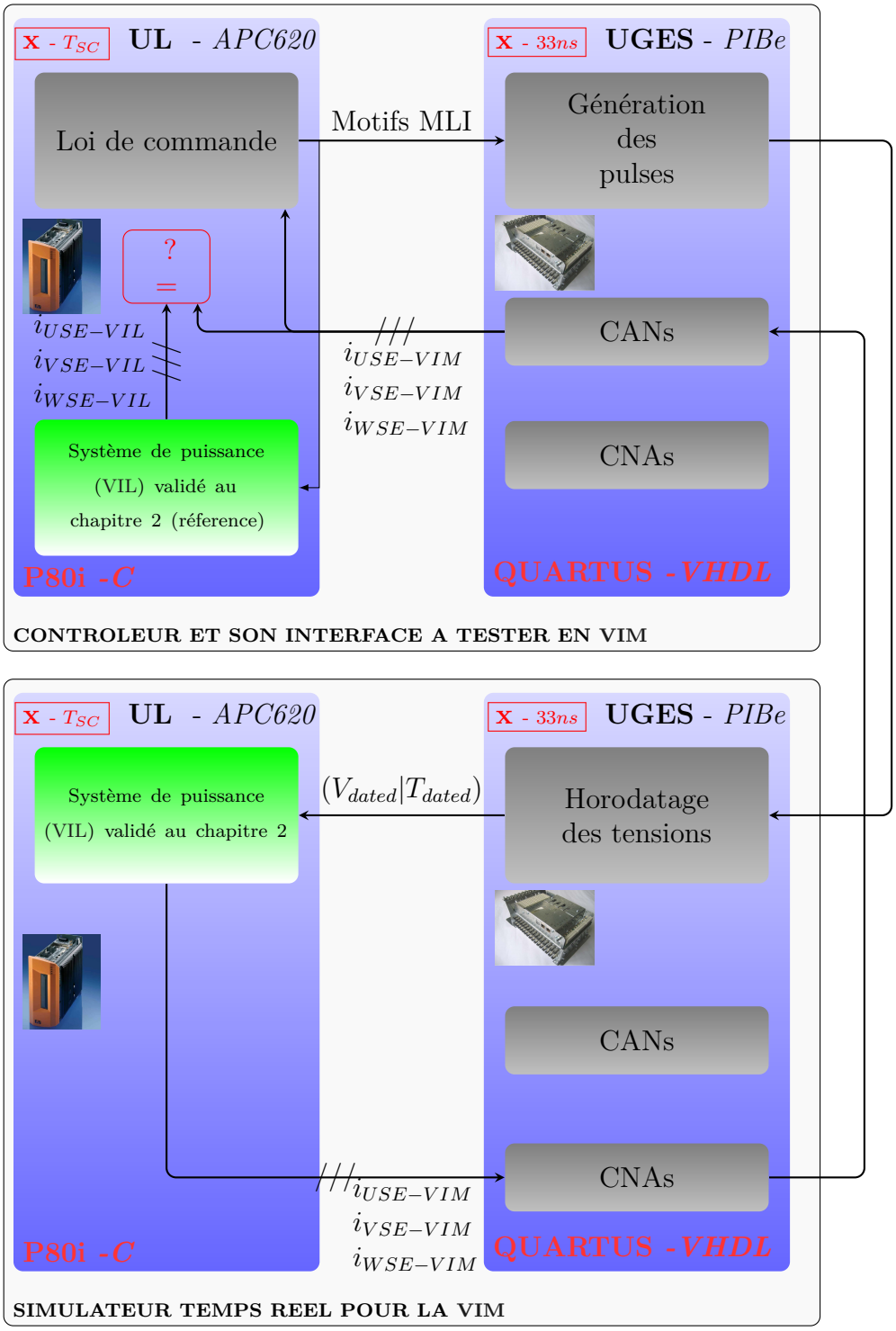


FIGURE 3.40 – Validation de l'horodatage par comparaison avec le modèle développé pour la VIL.

Pour cela, nous procédons comme décrit à la Figure 3.40. Les retours des courants de l'ensemble constitué de l'onduleur trois niveaux et de la machine asynchrone sur le simulateur en VIM sont comparés en temps réel avec les retours qu'auraient générés le modèle en VIL.

Le simulateur fonctionne avec un pas de calcul de 130μs.

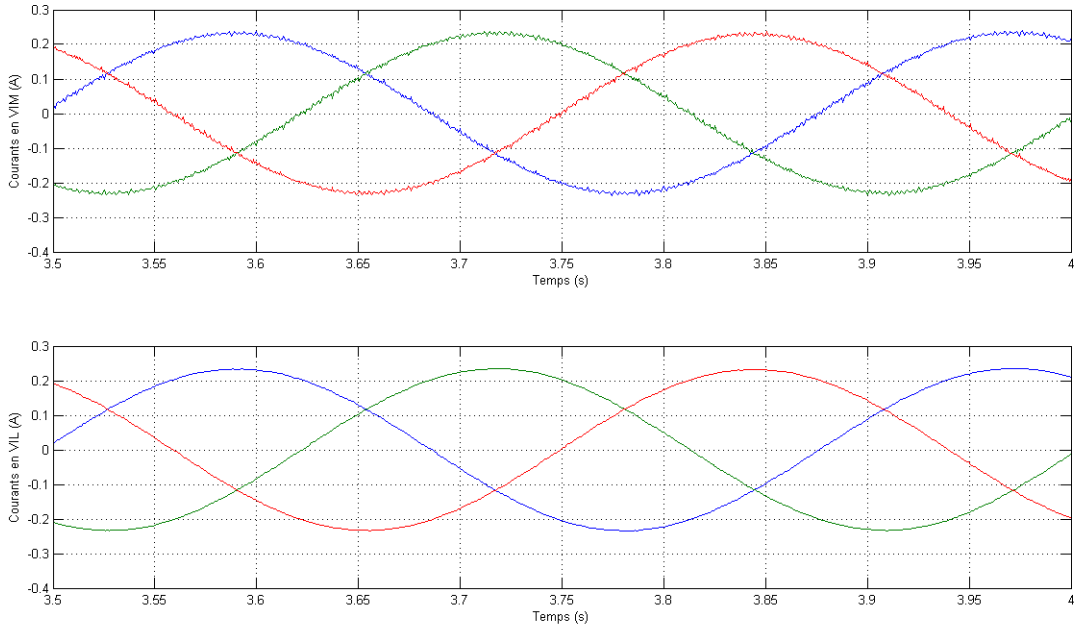


FIGURE 3.41 – Courants statoriques de la machine en VIL et VIM.

Les formes d'onde obtenues en VIM et VIL sont très proches. Les différences sont dues :
- au comportement de l'UGES qui introduit des temps morts ;
- au fait que le contrôleur et le simulateur tournent avec des périodes différentes. La MLI utilisée est une MLI à porteuse triangulaire. Avec ce type de MLI, le contrôleur échantillonne au sommet des porteuses. Cet échantillonnage fait office de filtrage et la VIL assure une synchronisation des mesures par rapport aux porteuses. Par contre, des retards dus à l'horodatage, au traitement par interpolation évènementielle et aux échanges de données entre l'UL et l'UGES de la partie simulateur corrompt la synchronisation des mesures par rapport au sommet des porteuses. Il est parfois nécessaire d'introduire une compensation des retards par avance de phase sur les valeurs calculées en VIM avec processeur lent quand la période de tâche du contrôleur est du même ordre de grandeur que la période de tâche du simulateur.

3.4 Conclusion

L'implémentation complète de la simulation par VIM sur FPGA permet d'avoir une haute résolution temporelle que l'on ne peut obtenir avec des simulateurs à base de cœur de processeurs. Dans ce type d'implémentation, le pas de calcul est complètement indépendant de la taille du système. Cependant, la limitation se situe au niveau des ressources logiques disponibles sur le type de composant utilisé. Dans ce travail, nous avons été contraints par les capacités du composant FPGA utilisé sachant que ce dernier était initialement dédié à des tâches de contrôle liées aux traitements basiques de signaux (acquisitions de signaux, valeurs moyennes, génération de signaux de gâchettes et quelques entrées/sorties logiques). Un énorme effort a été fait pour réduire au maximum les ressources utilisées pour faire du temps réel. Par ailleurs, l'implémentation complète sur FPGA requiert beaucoup plus de temps de conception qu'une implémentation sur processeur.

Une simulation temps réel plus classique utilisant conjointement FPGA et processeur

s'implémente beaucoup plus rapidement. Cependant, on peut simuler un système de taille quelconque au détriment du pas de calcul. En effet, plus le système est grand, plus le nombre de lignes de code augmente et plus le pas de calcul temps réel augmente. Le fait d'augmenter le pas de calcul peut induire à des instabilités numériques comme cela a été présenté aux chapitres précédents. En plus, l'horodatage introduit des retards au niveau des signaux de retour du contrôleur qui peuvent créer des instabilités artificielles qui n'ont lieu d'être dans la réalité.

Il serait intéressant de concilier les avantages des deux types d'approches, la rapidité et flexibilité de programmation des simulateurs processeurs/FPGA et les performances temporelles élevées sur FPGA.
On pourrait imaginer par exemple un calcul à haute résolution sur FPGA pour les courants de la charge des convertisseurs de tension trois niveaux. Comme la charge des convertisseurs de tension est de nature inductive, il sera possible d'implémenter un modèle de charge de type RLE dont les résistances et inductances seraient des résistances équivalentes de machine asynchrone, synchrone ou de filtre.
La FEM et les coefficients discrets liés au modèle RL seront par contre calculés sur processeur selon le type de charge. De cette façon, l'implémentation sur FPGA reste inchangée. Le type de charge ainsi que les paramètres sont calculés au niveau du processeur.

Conclusion Générale

Le premier chapitre de cette thèse traite des notions théoriques sur les différentes étapes nécessaires au développement de modèles pour la simulation numérique de systèmes électrotechniques. Différentes méthodes de discrétisation ont été présentées. Les méthodes de discrétisation les plus simples (d'ordre le moins élevé) requièrent un pas de calcul temps réel plus petit que les méthodes d'ordre élevé pour garantir la précision requise. La méthode la plus sûre et précise, qui est aussi la plus difficile, est la recherche de l'équation temporelle exacte des équations différentielles du système. Cette dernière solution a l'avantage de ne pas engendrer de problèmes d'instabilité contrairement aux méthodes d'intégrations numériques.

La présence d'interrupteurs dans le système de puissance simulé complexifie la modélisation. Les commutations ne peuvent pas être simulées en temps réel sur des cibles de type processeur sans avoir recours aux méthodes d'interpolation évènementielle. D'autres techniques d'accélération comme la modélisation des interrupteurs par des éléments impédants existent. La modélisation des interrupteurs par résistance discrète constante est par exemple utilisée dans le RCM et la même philosophie peut-être utilisée dans la modélisation par ligne de transmission. Les interrupteurs impédants ont récemment été utilisés sur des cibles type FPGA.

Les FPGAs offrent les meilleures performances temporelles et permettent d'avoir une haute résolution mais également d'avoir moins de retard de calcul possible. Cependant, le développement sur ces cibles est plus difficile que sur un processeur. La génération de code n'arrive pas au même niveau d'automatisation que pour les processeurs où des codes de simulation de systèmes électrotechniques peuvent se faire via une interface de description topologique du circuit électrique. Les modèles implantés sur FPGA requièrent une attention particulière sur les formats de codage des mots en fonction du pas de calcul utilisé et des constantes de temps mises en jeu. Par ailleurs, le reparamétrage est plus facile sur des processeurs.

Une mise en pratique des méthodologies présentées est faite dans les deux derniers chapitres.

Si seule une validation de l'algorithme de contrôle est nécessaire, une VIL suffit. Le type de validation que nous proposons permet de pré-valider un contrôleur final sans surplus de matériel pour la simulation temps réel. Ce type de validation doit être impératif dans les processus de développement de contrôleurs industriels. En effet, cette validation ne coûte quasiment rien hormis le coût de développement initial des modèles et la librairie de simulation. Elle est la candidate la plus adaptée pour être utilisée par des ingénieurs en phase de conception de contrôleurs industriels. Cependant, elle peut surcharger le contrôleur et ne convient pas à des applications où le taux de charge du processeur est trop important. Dans le cas d'applications industrielles, des marges sont généralement gardées. Ce type de validation a permis de reproduire des phénomènes sur site et même de faire une démonstration devant le client avant la mise en service de son équipement. Elle permet d'éliminer par avance certains doutes en cas de mauvais fonctionnement du contrôleur.

Cependant, la VIL ne valide pas l'interface physique dédiée à la génération de signaux

de commande et à l'acquisition de données. Pour prendre en compte cette interface physique d'entrées/sorties, soit on modélise cette interface que l'on exécute en VIL, soit on passe à la VIM. La modélisation de l'interface d'entrée/sortie a peu d'intérêt car ces cartes évoluent tout le temps et sont souvent programmables.

Au dernier chapitre, nous avons présenté deux types de VIM : une implémentation sur FPGA et une implémentation sur processeur. L'implémentation sur FPGA se fait avec une méthode d'intégration d'ordre 1 car on peut imposer un pas de calcul suffisamment petit. Ce pas ne doit pas être trop petit pour éviter d'avoir à utiliser des mots à larges formats et de trop augmenter la quantité de ressources logiques requises. Cependant, lorsque la taille du système simulé augmente, on peut garder sur FPGA le même pas de calcul mais les ressources logiques requises augmenteront.

Une implémentation sur FPGA a été faite sur des cartes initialement dédiées au contrôle. De ce fait, une forte optimisation a été nécessaire pour que l'on puisse faire de la simulation temps réel sur ces cartes. L'utilisation d'une implémentation multipas ainsi que la recherche de similarité dans les équations à implanter réduisent la quantité de ressources logiques nécessaires. Cette implémentation est avantageuse en terme de résolution temporelle mais elle est plus coûteuse en temps de développement et elle est plus difficilement reparamétrable.

Une implémentation utilisant un processeur en tant que cœur de simulation et un FPGA est plus facile à réaliser. Le changement de paramètres ou du système à simuler se fait plus rapidement. Il existe même des plateformes de génération automatique de code temps réel. Cette implémentation utilise un pas de calcul plus important qu'une implémentation sur FPGA. Si l'on utilise la même base matérielle que celle utilisée par le contrôle, le pas de calcul du contrôleur et celui du simulateur seront du même ordre de grandeur. Dans ce cas, l'horodatage introduit un retard d'un pas du simulateur qui n'existe pas dans la réalité. Ce retard peut-être à l'origine d'instabilité artificielle du contrôleur. Il est nécessaire dans ce cas d'introduire une avance de phase dans la simulation.

Il serait intéressant par la suite de retrouver un compromis qui puisse faire bénéficier de l'avantage en terme de résolution du FPGA et de la facilité de reparamétrage et de reprogrammation des processeurs.
Dans les applications incluant des convertisseurs de tension, la charge est de nature inductive. On pourrait donc représenter ces charges par une charge RLE. La résistance R et l'inductance L seront des résistances équivalentes et la FEM E sera défini selon la nature de la charge directe du convertisseur (filtres, machine asynchrone ou machine synchrone ...).
La charge RLE sera donc simulée sur FPGA pour avoir une haute résolution. Par contre, la FEM E sera générée par le processeur selon le type de charge. La partie RLE simulée sur FPGA devra être reconfigurable depuis le processeur. Ce dernier enverra les coefficients discrets vers le FPGA par le bus de communication. Nous aboutirons ainsi à un simulateur hybride fournissant une haute résolution pour les grandeurs à dynamique rapide (courant). Ce simulateur hybride est paramétrable via le processeur.

Par ailleurs, des efforts devront être faits dans la génération automatique de code de synthèse pour FPGA. Des logiciels permettent de faciliter la génération de code de description matériel mais ils n'automatisent pas la génération de modèles de systèmes électrotechniques. La plus grande difficulté réside dans la détermination des formats à virgule fixe, nécessaires pour une implémentation sur FPGA.

Parmi les validations traitées dans ce travail, la VIL est la meilleure candidate pour une utilisation en masse car elle ne demande pas de matériel supplémentaire hormis le matériel dédié au contrôleur.
De nos jours, nous disposons simultanément des machines virtuelles et des processeurs multicœurs. Les machines virtuelles permettent de faire tourner un système d'exploitation sur un autre.

On est aujourd'hui capable de faire tourner les programmes temps réel de contrôleurs industriels sur nos ordinateurs à usage commun en utilisant une machine virtuelle. La machine virtuelle tourne sur un des cœurs du processeur. On peut alors faire une VIL qui utilise la même plateforme de programmation et le même code mais tournant sur une autre base matérielle qui est l'ordinateur grand public d'aujourd'hui.

Appendices

Annexe **A**

Rappel sur le calcul des résidus

Cas d'un pôle simple

Si $Y_i(s) = \frac{Num(s)}{Den(s)}$ et s_0 est un zéro simple de $Den(s)$, le résidu en s_0 $Res(s_0)$ est :

$$Res(s_0) = \frac{Num(s_0)}{Den'(s_0)} \tag{A.1}$$

$$Den(s_0) = 0 \text{ et } Den^{'}(s_0) \neq 0$$

Exemple :
Nous prenons comme exemple un cas mono-entrée mono-sortie. Soit une fonction de transfert $H_{11}(s) = \frac{1}{(1+s^2)}$ à laquelle on applique une entrée de type échelon d'amplitude unitaire $U_1(s) = \frac{1}{s}$:

$$Y_1(s) = \frac{1}{s(1+s^2)}$$

$$y_1(t) = \sum Residus\left(\frac{e^{st}}{s(1+s^2)}\right)$$

$Num_1(s) = e^{st}$
$Den_1(s) = s(1+s^2)$ et $Den_i^{'}(s) = 1 + 3s^2$
$Res(0) = 1$
$Res(j) = \frac{e^{jt}}{-2}$
$Res(-j) = \frac{e^{-jt}}{-2}$
D'où alors :

$$y_1(t) = 1 - \frac{e^{it} + e^{-it}}{2} = 1 - \frac{\cos t + j\sin t + \cos t - j\sin t}{2} = 1 - \cos t$$

La réponse est oscillatoire entretenue et est prévisible comme les pôles de $H_{in}(s)$ sont purement imaginaires.

Cas d'un pôle multiple de multiplicité m

Dans le cas où le pôle s_0 est de multiplicité m :

$$Res(s_0) = \frac{1}{(m-1)!} lim_{s\to s_0} \frac{d^{m-1}}{ds^{m-1}} \left((s-s_0)^m Y_i(s) e^{st}\right) \tag{A.2}$$

Exemple : Soit une fonction de transfert double intégrateur $H_{11}(s) = \frac{1}{s^2}$ à laquelle on applique une entrée de type échelon d'amplitude unitaire $U_1(s) = \frac{1}{s}$:

$$Y_1(s) = \frac{1}{s^3}$$

$s_0 = 0$ est de multiplicité $m = 3$

$$y_1(t) = Res_{(s_0=0)} \left(\frac{e^{st}}{s^3} \right)$$

$\frac{d^2}{ds^2} \left((s-0)^3 \frac{1}{s^3} e^{st} \right) = t^2 e^{st}$

$$y_1(t) = \frac{1}{(3-1)!} lim_{s \to 0} t^2 e^{st} = \frac{1}{2} t^2$$

La réponse est une parabole vu que l'on intègre deux fois l'échelon.

Annexe **B**

Exemples pour la deuxième méthode des résidus

Nous reprenons l'exemple correspondant à la méthode des résidus I de l'annexe A avec :

$$H_{11}(s) = \frac{1}{1+s^2} \tag{B.1}$$

On se propose de calculer une version discrète $H^0_{11}(z)$ par invariance impulsionnelle.

$$H^0_{11}(z) = \mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{1}{1+s^2}\right)\right) \tag{B.2}$$

On choisira la méthode des résidus I pour calculer $\mathcal{L}^{-1}\left(\frac{1}{1+s^2}\right)$:

$$\mathcal{L}^{-1}\left(\frac{1}{1+s^2}\right) = \sum Res\left(\frac{e^{st}}{1+s^2}\right) = \frac{e^{jt}}{2j} + \frac{e^{-jt}}{-2j} = \sin t \tag{B.3}$$

$$H^0_{11}(z) = \mathcal{Z}\left(\sin t\right) = \frac{z^{-1}\sin T_s}{1 - 2z^{-1}\cos T_s + z^{-2}} \tag{B.4}$$

On se propose maintenant de calculer une version discrète $H_{11}(z)$ par invariance indicielle.

$$H_{ij}(z) = (1-z^{-1})\mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{1}{s(1+s^2)}\right)\right) \tag{B.5}$$

La partie à inverser par la transformée de Laplace a été calculée par la méthode des résidus I et vaut :

$$\mathcal{L}^{-1}\left(\frac{1}{s(1+s^2)}\right) = 1 - \cos t \tag{B.6}$$

$$H_{ij}(z) = (1-z^{-1})\mathcal{Z}\left(1-\cos t\right) = (1-z^{-1})\left(\frac{1}{1-z^{-1}} - \frac{1-z^{-1}\cos T_s}{1-2z^{-1}\cos T_s + z^{-2}}\right) \tag{B.7}$$

$$H_{ij}(z) = 1 - \frac{(1-z^{-1})(1-z^{-1}\cos T_s)}{1-2z^{-1}\cos T_s + z^{-2}} \tag{B.8}$$

Annexe C

Exemples pour la méthode temporelle exacte

L'équation (1.73) est une équation générale. Nous traitons ici le cas où les éléments U_i du vecteur de commande U ont une variation déterminée par une fonction connue. Nous traitons ici les variations type impulsion, échelon et rampe à titre d'exemple, mais l'étude pourrait également être menée pour une entrée de type harmonique.

Réponse impulsionnelle

On définit une fonction impulsion $\delta_{T_i}^{U_i}$:

$$\delta_{T_i}^{U_i}(t = T_i) = \begin{cases} \frac{U_i}{T_i} & \text{si } 0 \leq t \leq T_i \\ 0 & \text{sinon} \\ \text{avec} & T_i \to 0 \end{cases} \tag{C.1}$$

Si les éléments U_i de $U(t)$ sont de type impulsion, l'équation (1.73) s'écrit :

$$X(t) = e^{At}X(0) + e^{At}\left(\int_0^{T_i} e^{-At}dt\right)B\frac{U}{T_i} \tag{C.2}$$

$$X(t) = e^{At}X(0) + e^{At}\left[-A^{-1}e^{-At}\right]_0^{T_i} B\frac{U}{T_i} = e^{At}X(0) + e^{At}(-A^{-1}e^{-AT_i} + A^{-1})B\frac{U}{T_i}$$

$$X(t) = e^{At}X(0) - e^{At}A^{-1}(e^{-AT_i} - I)B\frac{U}{T_i}$$

$$\frac{e^{-AT_i} - I}{T_i} = \frac{I - AT_i + \frac{(-AT_i)^2}{2!} + \frac{(-AT_i)^3}{3!} + ... - I}{T_i}$$

L'impulsion parfaite est obtenue quand $T_i \to 0$:

$$lim_{T_i \to 0}\frac{e^{-AT_i} - I}{T_i} = -A$$

$$X(t) = e^{At}X(0) - e^{At}A^{-1}(-A)BU$$

Finalement, pour un vecteur de commande de type impulsion :

$$X(t) = e^{At}X(0) + e^{At}BU \tag{C.3}$$

Pour des conditions initiales nulles ($X(0) = 0$, la réponse impulsionnelle correspond à la fonction de transfert de système.

Réponse indicielle

Si les éléments U_i de $U(t)$ sont constants durant une période de calcul (ce qui est souvent le cas dans le cadre de notre étude sur la validation de contrôleur, les références restant constantes entre deux pas de calcul), l'équation (1.73) s'écrit :

$$X(t) = e^{At}X(0) + e^{At}\left(\int_0^t e^{-At}dt\right)BU \tag{C.4}$$

$$X(t) = e^{At}X(0) + e^{At}\left[-A^{-1}e^{-At}\right]_0^t BU = e^{At}X(0) + e^{At}(-A^{-1}e^{-At} + A^{-1})BU \tag{C.5}$$

$$X(t) = e^{At}X(0) + e^{At}A^{-1}(-e^{-At} + I)BU \tag{C.6}$$

Comme $e^{At}A = Ae^{At}$, $A^{-1}e^{At}A = e^{At}$ et donc $A^{-1}e^{At} = e^{At}A^{-1}$ d'où finalement la solution exacte pour des commandes de type échelon :

$$X(t) = e^{At}X(0) + A^{-1}(e^{At} - I)BU \tag{C.7}$$

Le cas de la réponse indicielle sera retenu dans la partie applicative sur les convertisseurs de tension pour deux raisons :

- la (les) tension(s) du bus continu varie (varient) très peu pendant une période de calcul
- la tension appliquée à la charge du convertisseur reste la même tant que la topologie du circuit ne change pas.

Réponse à une rampe

Si les éléments U_i de $U(t)$ évoluent d'une façon linéaire :

$$U(t) = \frac{U(T_s) - U(0)}{T_s}t + U(0) = Kt + U(0) \tag{C.8}$$

L'équation(1.73) s'écrit alors comme suit :

$$X(t) = e^{At}X(0) + e^{At}\int_0^t e^{-At}B(Kt + U(0))dt \tag{C.9}$$

$$X(t) = e^{At}X(0) + e^{At}\int_0^t te^{-At}dtBK + e^{At}\int_0^t e^{-At}dtBU(0)$$

$$X(t) = e^{At}X(0) + e^{At}\int_0^t te^{-At}dtBK + A^{-1}(e^{At} - I)BU(0)$$

On calcule alors l'intégrale par intégration par partie et on obtient :

$$X(t) = e^{At}X(0) + e^{At}\left(t(-A^{-1})e^{-At} + (A^{-2})\left(e^{-At} - I\right)\right)BK + A^{-1}(e^{At} - I)BU(0)$$

Comme $(-A^{-1})e^{-At} = e^{-At}(-A^{-1})$ alors $(-A^{-1})(-A^{-1})e^{-At} = (-A^{-1})e^{-At}(-A^{-1}) = e^{-At}(-A^{-1})(-A^{-1})$

$$X(t) = e^{At}X(0) + \left(t(-A^{-1}) + (A^{-2})\left(I - e^{At}\right)\right)BK + A^{-1}(e^{At} - I)BU(0)$$

$$X(t) = e^{At}X(0) + \left(t(-A^{-1}) - (A^{-2})\left(e^{At} - I\right)\right)BK + A^{-1}(e^{At} - I)BU(0)$$

L'équation(1.73) s'écrit alors dans le cas où le vecteur de commande évolue linéairement :

$$X(t) = e^{At}X(0) - A^{-1}\left(t + (A^{-1})\left(e^{At} - I\right)\right)BK + A^{-1}(e^{At} - I)BU(0) \tag{C.10}$$

K est le vecteur des pentes des éléments U_i.

Dans le cadre de la simulation Temps réel, les calculs se font à des instants discrets :

$$X(T_s) = e^{AT_s}X(0) - A^{-1}\left(T_s + (A^{-1})\left(e^{AT_s} - I\right)\right)B\left(\frac{U(T_s) - U(0)}{T_s}\right) + A^{-1}(e^{At} - I)BU(0)$$

D'où finalement :

$$X(T_s) = e^{AT_s}X(0) + A^{-1}\left(e^{At}BU(0) - BU(T_s)\right) - A^{-2}\left(e^{AT_s} - I\right)B\left(\frac{U(T_s) - U(0)}{T_s}\right) \tag{C.11}$$

A partir de l'équation (C.11), on retrouve facilement le cas de la réponse indicielle de l'équation (C.7) en posant $K = 0$ ou $U(T_s) = U(0)$.

Annexe **D**

Modèle continu de la machine asynchrone

D.1 Hypothèses

On suppose que :
- le bobinage permet d'avoir des forces électromotrices sinusoïdales pour des courants sinusoïdaux ;
- la machine travaille en régime non saturé ;
- les trois phases des bobinages statoriques (idem pour les bobinages rotoriques) présentent les mêmes caractéristiques (résistances, inductances propres et inductances mutuelles).

Les enroulements statoriques et rotoriques sont disposés comme indiqués sur la Figure D.1.

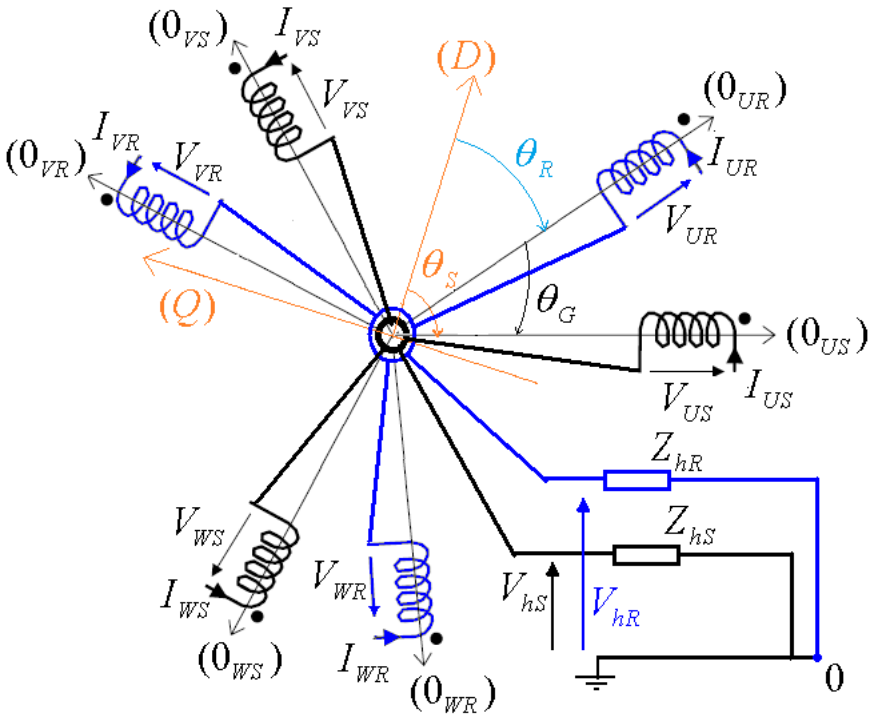


Figure D.1 – Disposition des enroulements de la machine asynchrone triphasée.

On désigne par :
θ_S l'angle entre l'axe (D) et l'axe de la phase de référence du stator 0_{US}
θ_R l'angle entre l'axe (D) et la phase de référence du rotor 0_{UR}
θ_G l'angle entre l'axe 0_{UR} et l'axe 0_{US}. Les grandeurs statoriques ne tournent pas avec la même

fréquence que les grandeurs rotoriques à cause du *glissement*.

Nous considérons dans un premier temps que les axes de modélisation (D) et (Q) sont quelconques. Nous verrons par la suite qu'il peuvent-être choisi de façon à coïncider aux axes statoriques, rotoriques ou au flux d'entrefer.

D.2 Modélisation

D.2.1 Tensions statoriques et rotoriques

On commence par les équations des tensions statoriques :

$$\begin{aligned} V_{UVWS,0} =& V_{UVWS} + V_{hS} = \begin{pmatrix} V_{US} \\ V_{VS} \\ V_{WS} \end{pmatrix} + V_{hS} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ =& R_S \begin{pmatrix} I_{US} \\ I_{VS} \\ I_{WS} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \phi_{US} \\ \phi_{VS} \\ \phi_{WS} \end{pmatrix} + Z_{hS} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} I_{US} \\ I_{VS} \\ I_{WS} \end{pmatrix} \\ =& R_S . I_{UVWS} + \frac{d\phi_{UVWS}}{dt} + Z_{hS} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} . I_{UVWS} \end{aligned} \tag{D.1}$$

V_{UVWS} est le vecteur des tensions triphasées appliquées aux bobinages statoriques ;
$V_{UVWS,0}$ est le vecteur des tensions triphasées statoriques par rapport au nœud de référence ;
I_{UVWS} est le vecteur des courants triphasés statoriques ;
R_S est un scalaire et correspond à la résistance de chaque bobinage statorique ;
ϕ_{UVWS} est le vecteur des flux statoriques ;
Z_{hS} est l'impédance qui lie le point commun des bobinages statoriques au noeud de référence 0.

De même pour le rotor :

$$\begin{aligned} V_{UVWR,0} =& V_{UVWR} + V_{hR} \\ =& \begin{pmatrix} V_{UR} \\ V_{VR} \\ V_{WR} \end{pmatrix} + V_{hS} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ =& R_R \begin{pmatrix} I_{UR} \\ I_{VR} \\ I_{WR} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} \phi_{UR} \\ \phi_{VR} \\ \phi_{WR} \end{pmatrix} + Z_{hR} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} I_{UR} \\ I_{VR} \\ I_{WR} \end{pmatrix} \\ =& R_R . I_{UVWR} + \frac{d\phi_{UVWR}}{dt} + Z_{hR} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} . I_{UVWR} \end{aligned} \tag{D.2}$$

V_{UVWR} est le vecteur des tensions triphasées rotoriques ;
I_{UVWR} est le vecteur des courants triphasés rotoriques ;
$V_{UVWS,0}$ est le vecteur des tensions triphasées statoriques par rapport au nœud de référence ;
R_R est un scalaire et correspond à la résistance de chaque bobinage rotorique ;
ϕ_{UVWR} est le vecteur de flux rotoriques ;
Z_{hR} est l'impédance qui lie le point commun des bobinages rotoriques au nœud de référence 0.

D.2.2 Flux statoriques et rotoriques

Les flux statoriques et rotoriques s'écrivent :

$$\begin{pmatrix} \phi_{UVWS} \\ \phi_{UVWR} \end{pmatrix} = \left(\begin{array}{c|c} L_S & M_{SR(\theta_G)} \\ \hline M^T_{SR(\theta_G)} & L_R \end{array}\right) \begin{pmatrix} I_{UVWS} \\ I_{UVWR} \end{pmatrix} \tag{D.3}$$

l_S est l'inductance propre statorique ;
m_S est l'inductance mutuelle statorique ;
l_R est l'inductance propre rotorique ;
m_R est l'inductance mutuelle rotorique ;
m_{Smax} est la valeur de l'inductance mutuelle entre deux phases du stator si elles étaient face à face sur le même axe ;
m_{Rmax} est la valeur de l'inductance mutuelle entre deux phases du rotor si elles étaient face à face sur le même axe ;
m_{SRmax} est la valeur de l'inductance mutuelle entre une phase du stator et une phase du rotor si elles étaient face à face sur le même axe.

$$L_S = \begin{pmatrix} l_S & m_{Smax}\cos\frac{2\pi}{3} & m_{Smax}\cos\frac{-2\pi}{3} \\ m_{Smax}\cos\frac{-2\pi}{3} & l_S & m_{Smax}\cos\frac{2\pi}{3} \\ m_{Smax}\cos\frac{2\pi}{3} & m_{Smax}\cos\frac{-2\pi}{3} & l_S \end{pmatrix} = \begin{pmatrix} l_S & m_S & m_S \\ m_S & l_S & m_S \\ m_S & m_S & l_S \end{pmatrix}$$

$$L_R = \begin{pmatrix} l_R & m_{Rmax}\cos\frac{2\pi}{3} & m_{Rmax}\cos\frac{-2\pi}{3} \\ m_{Rmax}\cos\frac{-2\pi}{3} & l_R & m_{Rmax}\cos\frac{2\pi}{3} \\ m_{Rmax}\cos\frac{2\pi}{3} & m_{Rmax}\cos\frac{-2\pi}{3} & l_R \end{pmatrix} = \begin{pmatrix} l_R & m_R & m_R \\ m_R & l_R & m_R \\ m_R & m_R & l_R \end{pmatrix}$$

$$M_{SR(\theta_G)} = \begin{pmatrix} m_{SRmax}\cos(\theta_G) & m_{SRmax}\cos\left(\theta_G+\frac{2\pi}{3}\right) & m_{SRmax}\cos\left(\theta_G-\frac{2\pi}{3}\right) \\ m_{SRmax}\cos\left(\theta_G-\frac{2\pi}{3}\right) & m_{SRmax}\cos(\theta_G) & m_{SRmax}\cos\left(\theta_G+\frac{2\pi}{3}\right) \\ m_{SRmax}\cos\left(\theta_G+\frac{2\pi}{3}\right) & m_{SRmax}\cos\left(\theta_G-\frac{2\pi}{3}\right) & m_{SRmax}\cos(\theta_G) \end{pmatrix}$$

Les matrices d'inductances statoriques L_S et rotoriques L_R sont diagonalisables. La manipulation de matrices diagonales est de loin plus simple et nécessite moins de calcul.

D.2.3 Diagonalisation des matrices inductances L_S et L_R

Nous allons traité le cas de la matrice d'inductance statorique L_S. La diagonalisation de la matrice inductance rotorique se fait de la même façon.
Les valeurs propres de L_S sont les λ_i vérifiant :

$$Det(L_S - \lambda I) = 0 \tag{D.4}$$

$$Det\begin{pmatrix} l_S-\lambda & m_S & m_S \\ m_S & l_S-\lambda & m_S \\ m_S & m_S & l_S-\lambda \end{pmatrix} = 0 \tag{D.5}$$

$$(l_S-\lambda)\left((l_S-\lambda)^2 - m_S^2\right) - m_S\left(m_S(l_S-\lambda) - m_S^2\right) + m_S\left(-m_S(l_S-\lambda) + m_S^2\right) = 0 \tag{D.6}$$

$$(l_S-\lambda)(l_S-\lambda-m_S)(l_S-\lambda+m_S) - 2m_S^2(l_S-\lambda-m_S) = 0 \tag{D.7}$$

$$(l_S - \lambda - m_S)\left((l_S - \lambda)(l_S - \lambda + m_S) - 2m_S^2\right) = 0 \tag{D.8}$$

On obtient le produit d'un terme du premier ordre qui s'annule pour $\lambda = \lambda_1$ et d'un terme du deuxième ordre qui s'annule pour $\lambda = \lambda_2$ ou $\lambda = \lambda_3$ avec :

$$\begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{pmatrix} = \begin{pmatrix}l_S - m_S\\ l_S - m_S\\ l_S + 2m_S\end{pmatrix} \tag{D.9}$$

$\lambda_1 = l_S - m_S$ est de multiplicité algébrique $m = 2$. Selon que l'on a deux chaines de Jordan dimension 1 ou une seule chaine de Jordan de dimension deux, la matrice (ou quasi) diagonale équivalente vaut alors :

$$\begin{pmatrix}l_S - m_S & 0 & 0\\ 0 & l_S - m_S & 0\\ 0 & 0 & l_S + 2m_S\end{pmatrix} \text{ ou } \begin{pmatrix}l_S - m_S & 1 & 0\\ 0 & l_S - m_S & 0\\ 0 & 0 & l_S + 2m_S\end{pmatrix}$$

Pour faire le choix de l'expression, nous avons besoin de déterminer le nombre de chaines de Jordan. La multiplicité géométrique donne le nombre de chaines de Jordan. La multiplicité géométrique q est :

$$q = 3 - rang\left(L_S - (l_S - m_S)\, I\right) = 3 - rang\begin{pmatrix}m_S & m_S & m_S\\ m_S & m_S & m_S\\ m_S & m_S & m_S\end{pmatrix} = 3 - 1 = 2$$

On a donc deux chaines de Jordan de dimension un pour la matrice d'inductance L_S de dimension $n = 3$. La matrice diagonale équivalente à L_S s'écrit donc :

$$L_{Sdiag} = \begin{pmatrix}l_S - m_S & 0 & 0\\ 0 & l_S - m_S & 0\\ 0 & 0 & l_S + 2m_S\end{pmatrix} = \begin{pmatrix}l_{CS} & 0 & 0\\ 0 & l_{CS} & 0\\ 0 & 0 & l_{CS} + 3m_S\end{pmatrix} \tag{D.10}$$

l_{CS} est l'inductance cyclique statorique. En effet, l'inductance propre peut être décomposée en somme d'une inductance cyclique l_{CS} et d'une mutuelle m_S. L'inductance mutuelle m_S est égale à la mutuelle entre deux phases du stator décalées de $\frac{2\pi}{3}$.

$$l_S = l_{CS} + m_S \tag{D.11}$$

De même, la matrice diagonale équivalente à L_R s'écrit :

$$L_{Rdiag} = \begin{pmatrix}l_R - m_R & 0 & 0\\ 0 & l_R - m_R & 0\\ 0 & 0 & l_R + 2m_R\end{pmatrix} = \begin{pmatrix}l_{CR} & 0 & 0\\ 0 & l_{CR} & 0\\ 0 & 0 & l_{CR} + 3m_R\end{pmatrix} \tag{D.12}$$

où l_{CR} est l'inductance cyclique rotorique.

D.2.3.1 Matrices de transformation P pour diagonaliser

On calcule maintenant la matrice de changement de base P pour arriver à cette forme diagonale :

$$P = \begin{pmatrix}V_{\lambda_1} & V_{\lambda_2} & V_{\lambda_3}\end{pmatrix} = \begin{pmatrix}V_{\lambda_1,1} & V_{\lambda_2,1} & V_{\lambda_3,1}\\ V_{\lambda_1,2} & V_{\lambda_2,2} & V_{\lambda_3,2}\\ V_{\lambda_1,3} & V_{\lambda_2,3} & V_{\lambda_3,3}\end{pmatrix} \tag{D.13}$$

Les éléments de P sont les vecteurs propres V_{λ_i} de L_S et vérifient donc :

$$L_S.V_{\lambda_i} = \lambda_i V_{\lambda_i} \tag{D.14}$$

Vecteurs propres associées à $\lambda_1 = \lambda_2 = l_{CS} = l_S - m_S$: $L_S \begin{pmatrix} V_{\lambda_1,1} \\ V_{\lambda_1,2} \\ V_{\lambda_1,3} \end{pmatrix} =$

$$\begin{pmatrix} l_S & m_S & m_S \\ m_S & l_S & m_S \\ m_S & m_S & l_S \end{pmatrix} \begin{pmatrix} V_{\lambda_1,1} \\ V_{\lambda_1,2} \\ V_{\lambda_1,3} \end{pmatrix} = (l_S - m_S) \begin{pmatrix} V_{\lambda_1,1} \\ V_{\lambda_1,2} \\ V_{\lambda_1,3} \end{pmatrix}$$

soit
$$\begin{pmatrix} l_S V_{\lambda_1,1} + m_S V_{\lambda_1,2} + m_S V_{\lambda_1,3} \\ m_S V_{\lambda_1,1} + l_S V_{\lambda_1,2} + m_S V_{\lambda_1,3} \\ m_S V_{\lambda_1,1} + m_S V_{\lambda_1,2} + l_S V_{\lambda_1,3} \end{pmatrix} = \begin{pmatrix} (l_S - m_S) V_{\lambda_1,1} \\ (l_S - m_S) V_{\lambda_1,2} \\ (l_S - m_S) V_{\lambda_1,3} \end{pmatrix}$$

d'où
$$V_{\lambda_1,1} + V_{\lambda_1,2} + V_{\lambda_1,3} = 0$$

Tout système triphasé équilibré est donc un vecteur propre de la matrice inductance L_S. Pour avoir deux vecteurs propres indépendants et normés, on choisit :

$$V_{\lambda_1} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix} \tag{D.15}$$

$$V_{\lambda_2} = \sqrt{\frac{2}{3}} \begin{pmatrix} -\sin(\theta) \\ -\sin\left(\theta - \frac{2\pi}{3}\right) \\ -\sin\left(\theta + \frac{2\pi}{3}\right) \end{pmatrix} \tag{D.16}$$

Vecteurs propres associés à $\lambda_3 = l_{CS} + 3m_S = l_S + 2m_S$:
Pour λ_3, on écrit :

$$\begin{pmatrix} l_S V_{\lambda_3,1} + m_S V_{\lambda_3,2} + m_S V_{\lambda_3,3} \\ m_S V_{\lambda_3,1} + l_S V_{\lambda_3,2} + m_S V_{\lambda_3,3} \\ m_S V_{\lambda_3,1} + m_S V_{\lambda_3,2} + l_S V_{\lambda_3,3} \end{pmatrix} = \begin{pmatrix} (l_S + 2m_S) V_{\lambda_3,1} \\ (l_S + 2m_S) V_{\lambda_3,2} \\ (l_S + 2m_S) V_{\lambda_3,3} \end{pmatrix}$$

soit
$$\begin{pmatrix} V_{\lambda_3,2} + V_{\lambda_3,3} \\ V_{\lambda_3,1} + V_{\lambda_3,3} \\ V_{\lambda_3,1} + V_{\lambda_3,2} \end{pmatrix} = \begin{pmatrix} 2V_{\lambda_3,1} \\ 2V_{\lambda_3,2} \\ 2V_{\lambda_3,3} \end{pmatrix}$$

d'où
$$V_{\lambda_3,1} = V_{\lambda_3,2} = V_{\lambda_3,3}$$

On choisit le vecteur normé :

$$V_{\lambda_3} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{D.17}$$

La matrice de transformation P s'écrit alors :

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos(\theta) & -\sin(\theta) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{pmatrix} \tag{D.18}$$

$P(\theta)$ est appelée transformation de $Park$
$P(\theta = 0)$ est la transformation de $Clark - Concordia$

$P(\theta)$ correspond au fait à une rotation d'un angle θ.

On écrit alors $L_{Sdiag} = P(\theta)^{-1} L_S P(\theta)$ en encore $P(\theta) L_{Sdiag} P(\theta)^{-1} = L_S$.
La rotation du repère (0_{UVWS}) lié au stator vers le repère $(DQ0)$ est une rotation d'un angle $\theta = \theta_S$ (voir Figure D.1) :

$$X_{DQ0S} = P(\theta_S)^{-1} X_{UVWS} \Leftrightarrow X_{UVWS} = P(\theta_S) X_{DQ0S} \tag{D.19}$$

De même la rotation du repère (0_{UVWR}) lié au rotor vers le repère $(DQ0)$ est une rotation d'un angle $\theta = \theta_R = \theta_S - \theta_G$ (voir Figure D.1) :

$$X_{DQ0R} = P(\theta_S - \theta_G)^{-1} X_{UVWR} \Leftrightarrow X_{UVWR} = P(\theta_S - \theta_G) X_{DQ0R} \tag{D.20}$$

D.2.4 Représentation d'état de la partie électrique de la machine asynchrone

D.2.4.1 Flux statoriques et rotoriques dans le repère $(DQ0)$

La partie statorique de l'équation (D.3), en ramenant les grandeurs statoriques et rotoriques dans le même repère $(DQ0)$, s'écrit alors :

$$\phi_{UVWS} = L_S I_{UVWS} + M_{SR(\theta_G)} I_{UVWR} = P(\theta_S) L_{Sdiag} \underbrace{P(\theta_S)^{-1} I_{UVWS}}_{I_{DQ0S}} + M_{SR(\theta_G)} I_{UVWR} \tag{D.21}$$

$$P(\theta_S)\phi_{DQ0S} = P(\theta_S) L_{Sdiag} I_{DQ0S} + M_{SR(\theta_G)} P(\theta_S - \theta_G) I_{DQ0R} \tag{D.22}$$

$$\phi_{DQ0S} = L_{Sdiag} I_{DQ0S} + P(\theta_S)^{-1} M_{SR(\theta_G)} P(\theta_S - \theta_G) I_{DQ0R} \tag{D.23}$$

On démontre que :

$$P(\theta_S)^{-1} M_{SR(\theta_G)} P(\theta_S - \theta_G) = \frac{3}{2} m_{SRmax} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = m_{SR} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{D.24}$$

$m_{SR} = \frac{3}{2} m_{SRmax}$ est appelée inductance mutuelle cyclique entre le stator et le rotor.
D'où finalement l'équation des flux statoriques :

$$\phi_{DQ0S} = L_{Sdiag} I_{DQ0S} + m_{SR} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0R} \tag{D.25}$$

Le même principe est appliqué pour avoir les équations des flux rotoriques. La partie rotorique de l'équation (D.3), en ramenant les grandeurs statoriques et rotoriques dans le même repère $(DQ0)$, s'écrit alors :

$$\phi_{UVWR} = L_R I_{UVWR} + M^T_{SR(\theta_G)} I_{UVWS} = P(\theta_R) L_{Sdiag} \underbrace{P(\theta_R)^{-1} I_{UVWR}}_{I_{DQ0R}} + M^T_{SR(\theta_G)} I_{UVWS} \tag{D.26}$$

$$P(\theta_R)\phi_{DQ0R} = P(\theta_R) L_{Rdiag} I_{DQ0R} + M^T_{SR(\theta_G)} P(\theta_S) I_{DQ0S} \tag{D.27}$$

$$\phi_{DQ0R} = L_{Rdiag} I_{DQ0R} + P(\theta_R)^{-1} M^T_{SR(\theta_G)} P(\theta_S) I_{DQ0S} \tag{D.28}$$

$$\phi_{DQ0R} = L_{Rdiag} I_{DQ0R} + P(\theta_S - \theta_G)^{-1} M^T_{SR(\theta_G)} P(\theta_S) I_{DQ0S} \tag{D.29}$$

$P(\theta)$ est une matrice orthonormée donc :

$$P^{-1}(\theta) = P^T(\theta) \tag{D.30}$$

$$\begin{aligned} P(\theta_S - \theta_G)^{-1} M^T_{SR(\theta_G)} P(\theta_S) &= \left(P^T(\theta_S) M_{SR(\theta_G)} (P^T(\theta_S - \theta_G))^{-1}\right)^T \\ &= \left(P^{-1}(\theta_S) M_{SR(\theta_G)} (P(\theta_S - \theta_G))\right)^T \\ &= m_{SR} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

D'où finalement l'équation des flux rotoriques :

$$\phi_{DQ0R} = L_{Rdiag} I_{DQ0R} + m_{SR} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0S} \tag{D.31}$$

D.2.4.2 Equations des tensions statoriques et rotoriques dans le repère $(DQ0)$

En utilisant une rotation $P(\theta = \theta_S)$, l'équation (D.1) devient :

$$P(\theta_S) V_{DQ0S} = R_S . P(\theta_S) I_{DQ0S} + \frac{d}{dt}\left(P(\theta_S)\phi_{DQ0S}\right) + Z_{hS} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} P(\theta_S) I_{DQ0S} \tag{D.32}$$

$$P(\theta_S) V_{DQ0S} = R_S . P(\theta_S) I_{DQ0S} + \frac{d}{dt}\left(P(\theta_S)\right)\phi_{DQ0S} + P(\theta_S)\frac{d}{dt}\phi_{DQ0S} + Z_{hS} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} P(\theta_S) I_{DQ0S} \tag{D.33}$$

$$V_{DQ0S} = R_S . I_{DQ0S} + \dot{\theta}_S P^{-1}(\theta_S)\frac{d}{d\theta}\left(P(\theta_S)\right)\phi_{DQ0S} + \frac{d}{dt}\phi_{DQ0S} + Z_{hS} P^{-1}(\theta_S) \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} P(\theta_S) I_{DQ0S} \tag{D.34}$$

$P(\theta)$ est une matrice orthonormée donc :

$$P^{-1}(\theta) = P^T(\theta) = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \tag{D.35}$$

$$\frac{d}{d\theta}\left(P(\theta)\right) = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos\left(\theta + \frac{\pi}{2}\right) & -\sin\left(\theta + \frac{\pi}{2}\right) & 0 \\ \cos\left(\theta + \frac{\pi}{2} - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{\pi}{2} - \frac{2\pi}{3}\right) & 0 \\ \cos\left(\theta + \frac{\pi}{2} + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{\pi}{2} + \frac{2\pi}{3}\right) & 0 \end{pmatrix} \tag{D.36}$$

On a alors :

$$P^{-1}(\theta)\frac{d}{d\theta}\left(P(\theta)\right) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{D.37}$$

et :

$$P^{-1}(\theta_S)\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}P(\theta_S)=\begin{pmatrix}0&0&0\\0&0&0\\0&0&3\end{pmatrix} \tag{D.38}$$

$$V_{DQ0S}=R_S.I_{DQ0S}+\dot{\theta}_S\begin{pmatrix}0&-1&0\\1&0&0\\0&0&0\end{pmatrix}\phi_{DQ0S}+\frac{d}{dt}\phi_{DQ0S}+Z_{hS}\begin{pmatrix}0&0&0\\0&0&0\\0&0&3\end{pmatrix}I_{DQ0S} \tag{D.39}$$

En utilisant une rotation $P(\theta=\theta_R)$, l'équation (D.2) devient :

$$P(\theta_R)V_{DQ0R}=R_R.P(\theta_R)I_{DQ0R}+\frac{d}{dt}\left(P(\theta_R)\phi_{DQ0R}\right)+Z_{hR}\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}P(\theta_R)I_{DQ0R} \tag{D.40}$$

$$P(\theta_R)V_{DQ0R}=R_R.P(\theta_R)I_{DQ0R}+\frac{d}{dt}\left(P(\theta_R)\right)\phi_{DQ0R}+P(\theta_R)\frac{d}{dt}\phi_{DQ0R}+Z_{hR}\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}P(\theta_R)I_{DQ0R} \tag{D.41}$$

$$V_{DQ0R}=R_R.I_{DQ0R}+\dot{\theta}_R P^{-1}(\theta_R)\frac{d}{d\theta}\left(P(\theta_R)\right)\phi_{DQ0R}+\frac{d}{dt}\phi_{DQ0R}+Z_{hR}P^{-1}(\theta_R)\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}P(\theta_R)I_{DQ0R} \tag{D.42}$$

$$V_{DQ0R}=R_R.I_{DQ0R}+\dot{\theta}_R\begin{pmatrix}0&-1&0\\1&0&0\\0&0&0\end{pmatrix}\phi_{DQ0R}+\frac{d}{dt}\phi_{DQ0R}+Z_{hR}\begin{pmatrix}0&0&0\\0&0&0\\0&0&3\end{pmatrix}I_{DQ0R} \tag{D.43}$$

En utilisant l'équation (D.25) et (D.39), on a :

$$\begin{aligned}V_{DQ0S}=&R_S.I_{DQ0S}+\dot{\theta}_S\begin{pmatrix}0&-1&0\\1&0&0\\0&0&0\end{pmatrix}\left(L_{Sdiag}I_{DQ0S}+m_{SR}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}I_{DQ0R}\right)\\&+\frac{d}{dt}\left(L_{Sdiag}I_{DQ0S}+m_{SR}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}I_{DQ0R}\right)+Z_{hS}\begin{pmatrix}0&0&0\\0&0&0\\0&0&3\end{pmatrix}I_{DQ0S}\end{aligned} \tag{D.44}$$

$$\begin{aligned}V_{DQ0S}=&R_S.I_{DQ0S}+L_{Sdiag}\begin{pmatrix}0&-\dot{\theta}_S&0\\\dot{\theta}_S&0&0\\0&0&0\end{pmatrix}I_{DQ0S}+m_{SR}\begin{pmatrix}0&-\dot{\theta}_S&0\\\dot{\theta}_S&0&0\\0&0&0\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}I_{DQ0R}\\&+L_{Sdiag}\frac{d}{dt}I_{DQ0S}+m_{SR}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}\frac{d}{dt}I_{DQ0R}+Z_{hS}\begin{pmatrix}0&0&0\\0&0&0\\0&0&3\end{pmatrix}I_{DQ0S}\end{aligned} \tag{D.45}$$

$$V_{DQ0S} = R_S.I_{DQ0S} + \begin{pmatrix} 0 & -l_{CS}\dot{\theta}_S & 0 \\ l_{CS}\dot{\theta}_S & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0S} + \begin{pmatrix} 0 & -m_{SR}\dot{\theta}_S & 0 \\ m_{SR}\dot{\theta}_S & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0R} + \begin{pmatrix} l_{CS} & 0 & 0 \\ 0 & l_{CS} & 0 \\ 0 & 0 & l_{CS}+3m_S \end{pmatrix} \frac{d}{dt} I_{DQ0S} + \begin{pmatrix} m_{SR} & 0 & 0 \\ 0 & m_{SR} & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{d}{dt} I_{DQ0R} + Z_{hS} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix} I_{DQ0S} \tag{D.46}$$

$$V_{DQ0S} = \begin{pmatrix} R_S & -l_{CS}\dot{\theta}_S & 0 \\ l_{CS}\dot{\theta}_S & R_S & 0 \\ 0 & 0 & R_S \end{pmatrix} I_{DQ0S} + \begin{pmatrix} 0 & -m_{SR}\dot{\theta}_S & 0 \\ m_{SR}\dot{\theta}_S & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0R} + \begin{pmatrix} l_{CS} & 0 & 0 \\ 0 & l_{CS} & 0 \\ 0 & 0 & l_{CS}+3m_S \end{pmatrix} \frac{d}{dt} I_{DQ0S} + \begin{pmatrix} m_{SR} & 0 & 0 \\ 0 & m_{SR} & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{d}{dt} I_{DQ0R} + Z_{hS} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix} I_{DQ0S} \tag{D.47}$$

De même pour le rotor, on a :

$$V_{DQ0R} = \begin{pmatrix} R_R & -l_{CR}\dot{\theta}_R & 0 \\ l_{CR}\dot{\theta}_R & R_R & 0 \\ 0 & 0 & R_R \end{pmatrix} I_{DQ0R} + \begin{pmatrix} 0 & -m_{SR}\dot{\theta}_R & 0 \\ m_{SR}\dot{\theta}_R & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} I_{DQ0S} + \begin{pmatrix} l_{CR} & 0 & 0 \\ 0 & l_{CR} & 0 \\ 0 & 0 & l_{CR}+3m_R \end{pmatrix} \frac{d}{dt} I_{DQ0R} + \begin{pmatrix} m_{SR} & 0 & 0 \\ 0 & m_{SR} & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{d}{dt} I_{DQ0S} + Z_{hR} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix} I_{DQ0R} \tag{D.48}$$

Les deux dernières équations se mettent sous la forme matricielle :

$$\begin{pmatrix} V_{DQ0S} \\ V_{DQ0R} \end{pmatrix} = \left(\begin{array}{ccc|ccc} R_S & -l_{CS}\dot{\theta}_S & 0 & 0 & -m_{SR}\dot{\theta}_S & 0 \\ l_{CS}\dot{\theta}_S & R_S & 0 & m_{SR}\dot{\theta}_S & 0 & 0 \\ 0 & 0 & R_S+3Z_{hS} & 0 & 0 & 0 \\ \hline 0 & -m_{SR}\dot{\theta}_R & 0 & R_R & -l_{CR}\dot{\theta}_R & 0 \\ m_{SR}\dot{\theta}_R & 0 & 0 & l_{CR}\dot{\theta}_R & R_R & 0 \\ 0 & 0 & 0 & 0 & 0 & R_R+3Z_{hR} \end{array}\right) \begin{pmatrix} I_{DQ0S} \\ I_{DQ0R} \end{pmatrix} + \left(\begin{array}{ccc|ccc} l_{CS} & 0 & 0 & m_{SR} & 0 & 0 \\ 0 & l_{CS} & 0 & 0 & m_{SR} & 0 \\ 0 & 0 & l_{CS}+3m_S & 0 & 0 & 0 \\ \hline m_{SR} & 0 & 0 & l_{CR} & 0 & 0 \\ 0 & m_{SR} & 0 & 0 & l_{CR} & 0 \\ 0 & 0 & 0 & 0 & 0 & l_{CR}+3m_R \end{array}\right) \frac{d}{dt} \begin{pmatrix} I_{DQ0S} \\ I_{DQ0R} \end{pmatrix} \tag{D.49}$$

Les parties homopolaires sont complètement indépendantes des composantes directe et en quadrature. Les circuits homopolaires équivalents correspondants sont représentés à la Figure D.2.

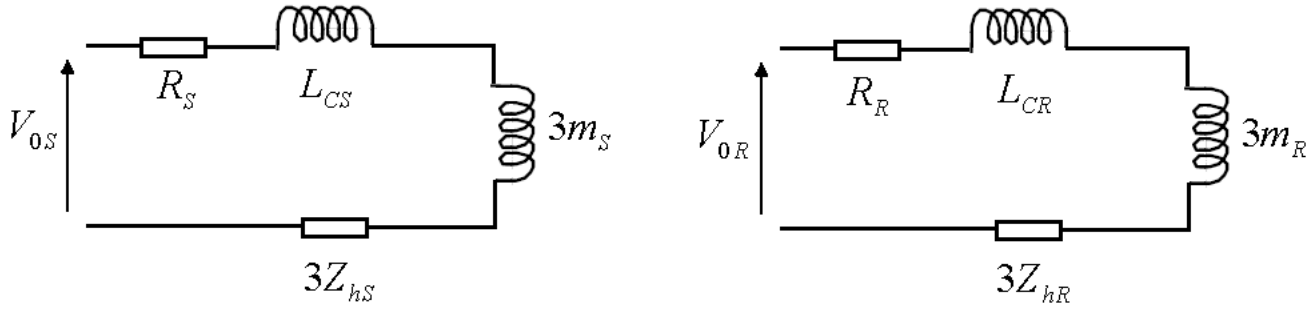


FIGURE D.2 – Circuits électriques équivalents de la partie homopolaire

Les calculs des grandeurs homopolaires peuvent donc être séparés des grandeurs dans les repères direct et en quadrature et dépendront de la connexion des neutres statorique et rotorique. On écrit alors l'expression des grandeurs statoriques et rotoriques directes et en quadrature comme suit :

$$\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix} = \underbrace{\left(\begin{array}{cc|cc} R_S & -l_{CS}\dot{\theta}_S & 0 & -m_{SR}\dot{\theta}_S \\ l_{CS}\dot{\theta}_S & R_S & m_{SR}\dot{\theta}_S & 0 \\ \hline 0 & -m_{SR}\dot{\theta}_R & R_R & -l_{CR}\dot{\theta}_R \\ m_{SR}\dot{\theta}_R & 0 & l_{CR}\dot{\theta}_R & R_R \end{array}\right)}_{R_{mas}} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} + \underbrace{\left(\begin{array}{cc|cc} l_{CS} & 0 & m_{SR} & 0 \\ 0 & l_{CS} & 0 & m_{SR} \\ \hline m_{SR} & 0 & l_{CR} & 0 \\ 0 & m_{SR} & 0 & l_{CR} \end{array}\right)}_{L_{mas}} \frac{d}{dt} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} \tag{D.50}$$

D.2.4.3 Équations d'état dans le repère (DQ) en utilisant les courants statoriques et rotoriques

La représentation d'état explicitée au premier chapitre consiste à exprimer les dérivées des variables d'état en fonction des variables d'état. Dans ce cas d'étude, nous allons devoir inverser la matrice L_{mas} de l'équation précédente pour retrouver la représentation d'état de la machine. Pour simplifier les calculs de l'inverse de la matrice inductance machine L_{mas}, les parties homopolaires étant indépendantes des composantes directe et en quadrature et peuvent donc être traitées à part. On inversera ainsi une matrice carrée de dimension 4. Nous obtenons :

$$L_{mas}^{-1} = \begin{pmatrix} \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} \\ \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 & \frac{l_{CS}}{l_{CR}l_{CS}-{m_{SR}}^2} & 0 \\ 0 & \frac{m_{SR}}{-l_{CR}l_{CS}+{m_{SR}}^2} & 0 & \frac{l_{CS}}{l_{CR}l_{CS}-{m_{SR}}^2} \end{pmatrix} \tag{D.51}$$

Les inductances cycliques statoriques l_{CS} et rotoriques l_{CR} sont dans certains ouvrages décomposées en inductances de fuites et inductance mutuelle m_{SR} entre le stator et rotor. Cette inductance mutuelle est à l'origine du flux utile qui se traduit par le couple électromagnétique.

$$l_{CS} = L_{fS} + m_{SR} \tag{D.52}$$

$$l_{CR} = L_{fR} + m_{SR} \tag{D.53}$$

L_{fS} est l'inductance de fuite statorique ;
L_{fR} est l'inductance de fuite rotorique.

Certains auteurs utilisent le coefficient de dispersion de Blondel σ défini comme suit :

$$\sigma = \frac{l_{CS}l_{CR} - m_{SR}^2}{l_{CS}l_{CR}} \tag{D.54}$$

La matrice inductance inverse L_{mas}^{-1} décrite par l'équation (D.51) peut alors s'écrire :

$$L_{mas}^{-1} = \begin{pmatrix} \frac{1}{\sigma l_{CS}} & 0 & -\frac{m_{SR}}{\sigma l_{CS} l_{CR}} & 0 \\ 0 & \frac{1}{\sigma l_{CS}} & 0 & -\frac{m_{SR}}{\sigma l_{CS} l_{CR}} \\ -\frac{m_{SR}}{\sigma l_{CS} l_{CR}} & 0 & \frac{1}{\sigma l_{CR}} & 0 \\ 0 & -\frac{m_{SR}}{\sigma l_{CS} l_{CR}} & 0 & \frac{1}{\sigma l_{CR}} \end{pmatrix} \tag{D.55}$$

Il en résulte la représentation d'état suivante de la machine asynchrone :

$$\frac{d}{dt}\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} = \underbrace{-L_{mas}^{-1}R_{mas}}_{A_{mas}} \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{X_{mas}} + \underbrace{L_{mas}^{-1}}_{B_{mas}} \underbrace{\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}}_{V_{mas}} \tag{D.56}$$

avec :

$$A_{mas} = \begin{pmatrix} a_{11} & a_{12} & a_{14} & a_{15} \\ -a_{12} & a_{11} & -a_{15} & a_{14} \\ a_{41} & a_{42} & a_{44} & a_{45} \\ -a_{42} & a_{41} & -a_{45} & a_{44} \end{pmatrix} \text{ et } B_{mas} = L_{mas}^{-1}$$

et :

$$\begin{aligned} a_{11} &= -\frac{l_{CR}R_S}{l_{CR}l_{CS} - {m_{SR}}^2} & a_{41} &= \frac{m_{SR}R_S}{l_{CR}l_{CS} - {m_{SR}}^2} \\ a_{12} &= \frac{{m_{SR}}^2\dot{\theta}_R - l_{CR}l_{CS}\dot{\theta}_S}{-l_{CR}l_{CS} + {m_{SR}}^2} & a_{42} &= \frac{-l_{CS}m_{SR}\dot{\theta}_R + l_{CS}m_{SR}\dot{\theta}_S}{-l_{CR}l_{CS} + {m_{SR}}^2} \\ a_{14} &= \frac{m_{SR}Rr}{l_{CR}l_{CS} - {m_{SR}}^2} & a_{44} &= -\frac{l_{CS}Rr}{l_{CR}l_{CS} - {m_{SR}}^2} \\ a_{15} &= \frac{-l_{CR}m_{SR}\dot{\theta}_R + l_{CR}m_{SR}\dot{\theta}_S}{l_{CR}l_{CS} - {m_{SR}}^2} & a_{45} &= \frac{l_{CR}l_{CS}\dot{\theta}_R - {m_{SR}}^2\dot{\theta}_S}{l_{CR}l_{CS} - {m_{SR}}^2} \end{aligned}$$

Comme un tour mécanique correspond à p période(s) électrique(s), on a :

$$\theta_R = p\theta_M \tag{D.57}$$

θ_R est la pulsation électrique équivalente de la partie mécanique $(rad.s^{-1})$;
θ_M est la pulsation de rotation mécanique : $\theta_M = 2\pi f_M$ $(rad.s^{-1})$;
f_M est la fréquence de rotation mécanique (Hz) ;
p est le nombre de paires de pôles ;
Dans cette représentation, les variables d'état sont les courants statoriques et rotoriques des axes direct (D) et en quadrature (Q). Nous allons présenter par la suite différentes variantes de modélisation de la machine.

D.2.5 Modélisation de la partie mécanique

Si le rotor est court-circuité, la puissance est intégralement fournie via le stator de la machine. La puissance instantanée fournie à la machine dans le cas général est :

$$\begin{aligned}
P_{inst} =& V_{UVWS}^T I_{UVWS} + V_{UVWR}^T I_{UVWR} \\
=& \left(P\left(\theta_S\right) V_{DQ0S}\right)^T \left(P\left(\theta_S\right) I_{DQ0S}\right) + \left(P\left(\theta_R\right) V_{DQ0R}\right)^T \left(P\left(\theta_R\right) I_{DQ0R}\right) \\
=& V_{DQ0S}^T P\left(\theta_S\right)^T P\left(\theta_S\right) I_{DQ0S} + V_{DQ0R}^T P\left(\theta_R\right)^T P\left(\theta_R\right) I_{DQ0R} \\
=& V_{DQ0S}^T I_{DQ0S} + V_{DQ0R}^T I_{DQ0R} \\
=& \underbrace{\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{P_{inst,DQ}} + \underbrace{V_{0S}^T I_{0S} + V_{0R}^T I_{0R}}_{P_{inst,0}}
\end{aligned}$$

avec :
$P\left(\theta_S\right)^T P\left(\theta_S\right) = P\left(\theta_R\right)^T P\left(\theta_R\right) = I$ car nous avions choisi des matrices de transformation orthogonales et normées.
$P_{inst,0}$ ne contribue pas au couple car cette puissance transite uniquement soit par le stator, soit par le rotor. Cette puissance n'est pas échangée entre le stator et le rotor.

A partir de l'équation (D.50), on peut alors écrire :

$$\begin{aligned}
\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}^T &= \left(R_{mas} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} + L_{mas} \frac{d}{dt} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} \right)^T \\
&= \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{mas}^T + \frac{d}{dt} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T L_{mas}^T
\end{aligned}$$

avec

$$\begin{aligned}
R_{mas} &= \left(\begin{array}{cc|cc} R_S & -l_{CS}\dot{\theta}_S & 0 & -m_{SR}\dot{\theta}_S \\ l_{CS}\dot{\theta}_S & R_S & m_{SR}\dot{\theta}_S & 0 \\ \hline 0 & -m_{SR}\dot{\theta}_R & R_R & -l_{CR}\dot{\theta}_R \\ m_{SR}\dot{\theta}_R & 0 & l_{CR}\dot{\theta}_R & R_R \end{array}\right) \\
&= \underbrace{\begin{pmatrix} R_S & 0 & 0 & 0 \\ 0 & R_S & 0 & 0 \\ 0 & 0 & R_R & 0 \\ 0 & 0 & 0 & R_R \end{pmatrix}}_{R_{elec}} + \underbrace{\left(\begin{array}{cc|cc} 0 & -l_{CS}\dot{\theta}_S & 0 & -m_{SR}\dot{\theta}_S \\ l_{CS}\dot{\theta}_S & 0 & m_{SR}\dot{\theta}_S & 0 \\ \hline 0 & -m_{SR}\dot{\theta}_R & 0 & -l_{CR}\dot{\theta}_R \\ m_{SR}\dot{\theta}_R & 0 & l_{CR}\dot{\theta}_R & 0 \end{array}\right)}_{R_{meca}}
\end{aligned}$$

Ainsi :

$$\begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}^T = \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{elec}^T + \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{meca}^T + \frac{d}{dt} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T L_{mas}^T \tag{D.58}$$

et

$$\begin{aligned}
P_{inst,DQ} &= \begin{pmatrix} V_{DQS} \\ V_{DQR} \end{pmatrix}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{elec}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{P_J} + \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{meca}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{P_{MECA}} + \underbrace{\frac{d}{dt} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T L_{mas}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{P_{EM}}
\end{aligned} \tag{D.59}$$

P_J correspond aux pertes joule statoriques et rotoriques ;
P_{MECA} est la puissance mécanique ;
P_{EM} correspond à la variation d'énergie électromagnétique dans le fer.

Sachant que :

$$P_{MECA} = T_{EM}\dot{\theta}_{meca}$$

On peut écrire :

$$T_{EM} = \frac{1}{\dot{\theta}_{meca}} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R_{meca}^T \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}$$

Soit :

$$T_{EM} = \frac{1}{\dot{\theta}_{meca}}(I_{DS}I_{QR}m_{SR} - I_{DR}I_{QS}m_{SR})\dot{\theta}_R + (-I_{DS}I_{QR}m_{SR} + I_{DR}I_{QS}m_{SR})\dot{\theta}_S \tag{D.60}$$

D.2.6 Différentes variantes du modèle d'état de la machine asynchrone

D.2.6.1 Variantes en fonction des variables d'état

Nous avions vu au premier chapitre que la représentation d'état n'est pas unique. La représentation d'état de l'équation (D.56) utilise les courants statoriques et rotoriques directs et en quadrature. On peut obtenir d'autres représentations en utilisant l'équation (1.12)établie au premier chapitre. La matrice de transformation choisie définit le lien entre deux représentations équivalentes mais utilisant des variables d'état différentes.

Par exemple, la matrice de passage du modèle de l'équation (D.56) dont le vecteur d'état X_{mas} est composé des courants statoriques et les rotoriques, à un modèle dont le vecteur d'état $\overline{X_{mas}}$ est composé des courants statoriques et les flux rotoriques est :

$$P_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ m_{SR} & 0 & l_{CR} & 0 \\ 0 & m_{SR} & 0 & l_{CR} \end{pmatrix} \tag{D.61}$$

Car la relation liant $\overline{X_{mas}}$ à X_{mas} est :

$$\underbrace{\begin{pmatrix} I_{DQS} \\ \phi_{DQR} \end{pmatrix}}_{\overline{X_{mas}}} = P_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} \underbrace{\begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}}_{X_{mas}} \tag{D.62}$$

et on inverse est :

$$P^{-1}_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{m_{SR}}{l_{CR}} & 0 & \frac{1}{l_{CR}} & 0 \\ 0 & -\frac{m_{SR}}{l_{CR}} & 0 & \frac{1}{l_{CR}} \end{pmatrix} \tag{D.63}$$

La matrice de transformation est très importante car elle permet de retrouver des expressions d'une représentation d'état à une autre.
En utilisant l'équation (1.12), on retrouve la matrice de dynamique équivalente $\overline{A_{mas}}$ et la

matrice de commande équivalente $\overline{B_{mas}}$:

$$\begin{aligned}\overline{A_{mas}} &= P_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} A_{mas} P^{-1}_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} \\ &= \begin{pmatrix} -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR} {m_{SR}}^2} & \dot{\theta}_S & \frac{m_{SR} R_R}{{l_{CR}}^2 l_{CS} - l_{CR} {m_{SR}}^2} & \frac{m_{SR}(\dot{\theta}_R - \dot{\theta}_S)}{-l_{CR} l_{CS} + {m_{SR}}^2} \\ -\dot{\theta}_S & -\frac{{m_{SR}}^2 R_R + {l_{CR}}^2 R_S}{{l_{CR}}^2 l_{CS} - l_{CR} {m_{SR}}^2} & -\frac{m_{SR}(\theta_R - \dot{\theta}_S)}{-l_{CR} l_{CS} + {m_{SR}}^2} & \frac{m_{SR} R_R}{{l_{CR}}^2 l_{CS} - l_{CR} {m_{SR}}^2} \\ \frac{m_{SR} R_R}{l_{CR}} & 0 & -\frac{R_R}{l_{CR}} & \dot{\theta}_R \\ 0 & \frac{m_{SR} R_R}{l_{CR}} & -\dot{\theta}_R & -\frac{R_R}{l_{CR}} \end{pmatrix} \end{aligned} \tag{D.64}$$

$$\overline{B_{mas}} = \begin{pmatrix} \frac{l_{CR}}{l_{CR} l_{CS} - {m_{SR}}^2} & 0 & -\frac{m_{SR}}{l_{CR} l_{CS} - {m_{SR}}^2} & 0 \\ 0 & \frac{l_{CR}}{l_{CR} l_{CS} - {m_{SR}}^2} & 0 & -\frac{m_{SR}}{l_{CR} l_{CS} - {m_{SR}}^2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{D.65}$$

On a la relation d'état équivalente :

$$\frac{d}{dt}\overline{X_{mas}} = \overline{A_{mas}}.\overline{X_{mas}} + \overline{B_{mas}}.V_{mas} \tag{D.66}$$

On utilise encore une fois la matrice de transformation pour retrouver l'expression du couple :

$$\begin{aligned} T_{EM} =& \frac{1}{\dot{\theta}_{meca}} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix}^T R^T_{meca} \begin{pmatrix} I_{DQS} \\ I_{DQR} \end{pmatrix} \\ =& \frac{1}{\dot{\theta}_{meca}} \begin{pmatrix} I_{DQS} \\ \phi_{DQR} \end{pmatrix}^T {P^{-1}}^T_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} R^T_{meca} P^{-1}_{(I_{DQS},I_{DQR})\to(I_{DQS},\phi_{DQR})} \begin{pmatrix} I_{DQS} \\ \phi_{DQR} \end{pmatrix} \end{aligned}$$

$$T_{EM} = \frac{1}{\dot{\theta}_{meca}} \left(-\frac{m_{SR}}{l_{CR}} \left(I_{QS}\phi_{DR} - I_{DS}\phi_{QR} \right) \dot{\theta}_R + \frac{m_{SR}}{l_{CR}} \left(I_{QS}\phi_{DR} - I_{DS}\phi_{QR} \right) \dot{\theta}_S \right) \tag{D.67}$$

D.2.6.2 Variante en fonction du positionnement du repère (DQ)

Nous n'avons fait aucune supposition sur les angles de rotation θ_S et θ_R. Selon l'application, on peut positionner le repère $(DQ0)$ de différentes manières. Le choix de ces angles peut être quelconque. Nous rappelons ici les cas les plus utilisés.

Cas du modèle lié au stator Le stator ne tourne pas. On peut placer le repère DQ de façon à avoir $\theta_S = 0$. Dans ce cas, l'axe (D) du repère (DQ) est confondu avec l'axe U_S de la phase U du stator. Ce repère est appelé repère de Clark et généralement noté $(\alpha\beta 0)$ au lieu de (DQ).

Dans ce cas, la matrice de dynamique A_{MAS} de la machine asynchrone prend la forme :

$$A_{MAS} = A_{mas}(\dot{\theta_S} = 0) \tag{D.68}$$

Dans ce cas, la relation d'auto-pilotage s'écrit :

$$\dot{\theta_S} = 0 = \dot{\theta}_R + p.\dot{\theta}_{meca} \Leftrightarrow \dot{\theta}_R = -p.\dot{\theta}_{meca} \tag{D.69}$$

Le couple électromagnétique s'exprime alors comme suit :

$$T_{EM} = \frac{1}{\dot{\theta}_{meca}} (I_{DS} I_{QR} m_{SR} - I_{DR} I_{QS} m_{SR}) \dot{\theta}_R = p.m_{SR}.(I_{DR} I_{QS} - I_{DS} I_{QR}) \tag{D.70}$$

De même, si l'on utilise par exemple le modèle courants statoriques et flux rotoriques avec un repère (DQ) fixe et lié au stator, on obtient :

$$T_{EM} = \frac{1}{\dot{\theta}_{meca}} \left(-\frac{m_{SR}}{l_{CR}} \left(I_{QS}\phi_{DR} - I_{DS}\phi_{QR} \right) \dot{\theta}_R \right)$$

$$T_{EM} = p\frac{m_{SR}}{l_{CR}} \left(I_{QS}\phi_{DR} - I_{DS}\phi_{QR} \right) \tag{D.71}$$

Cas du modèle lié au rotor On peut placer l'axe (D) du repère (DQ) sur la phase U_R du rotor de la machine. Dans ce cas l'angle θ_R est nul et le repère (DQ) tourne avec le rotor. La matrice de dynamique A_{MAS} de la machine asynchrone a alors pour expression :

$$A_{MAS} = A_{mas}(\dot{\theta_R} = 0) \tag{D.72}$$

La matrice de dynamique dépend seulement de $\dot{\theta_S}$.

Dans ce cas, la relation d'auto-pilotage s'écrit :

$$\dot{\theta_S} = \dot{\theta}_R + p.\dot{\theta}_{meca} = p.\dot{\theta}_{meca} \tag{D.73}$$

Cas du modèle lié au champ tournant On peut également envisager un modèle tel que le repère (DQ) tourne avec une certaine pulsation $\dot{\theta}$. Si cette pulsation correspond à la pulsation statorique $\dot{\theta_S}$, les grandeurs statoriques dans le repère DQ deviennent alors constantes. Dans ce cas, la relation d'auto-pilotage est :

$$\dot{\theta_S} = \dot{\theta}_R + p.\dot{\theta}_{meca} \tag{D.74}$$

On positionne le repère DQ en fonction de l'application.
Par exemple, le modèle lié au stator est utilisé à des fins de simulation de la machine asynchrone comme la pulsation statorique $\dot{\theta_S}$ n'est pas une grandeur explicitement accessible. $\dot{\theta_S}$ est une information contenue dans les tensions de commandes V_{mas}. Ce modèle utilise seulement $\dot{\theta_R}$ qui peut être calculé plus facilement en rajoutant le modèle de la partie mécanique.
Pour le contrôle de la machine asynchrone par exemple, on préfère le modèle lié au champ tournant car réguler des grandeurs continues est plus facile.

D.2.6.3 Variante en fonction de la tension rotorique

Selon la nature de l'application, le rotor peut-être court-circuité (cas des machines d'entrainement) ou pas (cas des éoliennes connectés au réseau). L'expression de la matrice de commande B_{mas} peut alors se simplifier en ne prenant que les deux premières colonnes si le rotor est court-circuité. Dans ce cas, le vecteur de commande est composé uniquement des tensions statoriques sur l'axe (D) et l'axe (Q).

La matrice de commande reste telle quelle si la machine est simultanément alimentée par le stator et par le rotor.

D.2.6.4 Variante machine asynchrone à neutre non relié ou fortement impédant

Le neutre du stator et du rotor sont souvent non reliés dans les schémas de validation de contrôleur. Le schéma homopolaire est réalisé de manière à limiter les courants homopolaires car ces courants ne contribuent pas à fournir du couple et nuisent à l'isolation de la machine. En plus, ces courants sont à haute fréquence et le contrôle ne peut réagir à ces fréquences. Si l'on désire uniquement valider le contrôleur sur sa bande passante, on peut négliger les courants homopolaires.
Dans le cas où les dynamiques homopolaires statorique et rotorique sont négligées, l'ordre du modèle est réduit à 4.

Pour une modélisation beaucoup plus fine, on peut supposer que le neutre du stator est relié à la carcasse de la machine par son impédance d'isolation. Cette isolation, dans le cas le plus simple, peut-être modélisée par la capacité d'isolation. A cette impédance d'isolation se rajoute l'impédance de mise à la terre de l'onduleur dans le schéma homopolaire. Comme les dynamiques des composantes homopolaires statorique et rotorique sont indépendantes des composantes directes et en quadrature, elles peuvent être simulé séparément pour réduire l'ordre du modèle.

Bibliographie

Alimeling, J. et Hammer, W. (1999). PLECS-piece-wise linear electrical circuit simulation for simulink. volume 1, pages 355–360 vol.1.

Allegre, A., Bouscayrol, A., Verhille, J., Delarue, P., Chattot, E. et El-Fassi, S. (2010). Reduced-Scale-Power Hardware-in-the-Loop simulation of an innovative subway. *Industrial Electronics, IEEE Transactions on*, 57(4) :1175–1185.

Amestoy, P., Davis, T. et Duff, I. (1996). An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4) :886–905.

Ammous, A., Ammous, K., Ayedi, M., Ounajjar, Y. et Sellami, F. (2003). An advanced PWM-switch model including semiconductor device nonlinearities. *Power Electronics, IEEE Transactions on*, 18(5) :1230–1237.

Ammous, K. (2002). *Contribution à la construction systématique des modèles moyens de convertisseurs de puissance*. Thèse de doctorat, Institut national des sciences appliquées de Lyon, Lyon.

Antsaklis, P. J. et Michel, A. N. (2005). *Linear Systems*. Springer.

Bauer, W. F. (1958). Aspects of Real-Time Simulation. *IEEE Transactions on Electronic Computers*, EC-7 :134–136.

Bedrosian, D. et Vlach, J. (1992). Time-domain analysis of networks with internally controlled switches. *Circuits and Systems I : Fundamental Theory and Applications, IEEE Transactions on*, 39(3) :199–212.

Bini, D., Capovani, M., Romani, F. et Lotti, G. (1979). < i> o</i>(< i> n</i>< sup> 2.7799</sup>) complexity for< i> n</i>×< i> n</i> approximate matrix multiplication. *Information processing letters*, 8(5) :234–235.

Borne, P., Rotella, F., Zambettakis, I. et Richard, J. (1992). *Modélisation et identification des processus*. Editions TECHNIP.

Brogan, W. L. (1991). *Modern control theory*. Prentice Hall.

Chen, C. (1993). *Analog and Digital Control System Design : Transfer-function, State-space and Algebraic Methods*. Saunders College Publishing/Harcourt Brace.

Chen, Y. et Dinavahi, V. (2009). FPGA-Based Real-Time EMTP. *IEEE Transactions on Power Delivery*, 24(2) :892–902.

Cohn, H., Kleinberg, R., Szegedy, B. et Umans, C. (2005). Group-theoretic algorithms for matrix multiplication. *In Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 379–388. IEEE.

Coppersmith, D. et Winograd, S. (1990). Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3) :251–280.

Courtecuisse, V., Mokadem, M. E., Guillaud, X., Salha, F. et Robyns, B. (2008). Use of real time simulation to validate primary frequency control with wind turbine. pages 1–8.

Crosbie, R., Zenor, J., Bednar, R., Word, D. et Hingorani, N. (2010). Using attached processors to achieve High-Speed Real-Time simulation. pages 140–143, Nice, France.

Crosbie, R. E., Zenor, J. J., Bednar, R., Word, D., Hingorani, N. G. et Ericsen, T. (2004). High-speed, scalable, real-time simulation using DSP arrays. *In Workshop on Principles of Advanced and Distributed Simulation*, pages 52–59.

Davis, T., Gilbert, J., Larimore, S. et Ng, E. (2004). A column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 30(3) :353–376.

Deng, Y., Foo, S. et Li, H. (2009). Real time simulation of power flow control strategies for fuel cell vehicle with energy storage by using real time digital simulator (RTDS). pages 2323–2327.

Dinavahi, V. (2000). *Real-t ime Digital Simulation of Switching Power Circuits*. Thèse de doctorat, University of Toronto.

Ding, H., Liu, W., Li, Y., Duan, X., Wu, D., Lin, Y., Chen, J., Zhang, R. et Zhang, J. (2008). Development of an active power filter physical simulator connected with RTDS. pages 1206–1211.

Dommel, H. (1969). Digital computer solution of electromagnetic transients in single-and multiphase networks. *Power Apparatus and Systems, IEEE Transactions on*, PAS-88(4) :388–399.

Dufour, C. et Belanger, J. (2001). Discrete time compensation of switching events for accurate real-time simulation of power systems. volume 2, pages 1533–1538 vol.2.

Dufour, C., Belanger, J., Abourida, S. et Lapointe, V. (2007a). FPGA-Based Real-Time simulation of Finite-Element analysis permanent magnet synchronous machine drives. pages 909–915.

Dufour, C., Belanger, J., Abourida, S. et Lapointe, V. (2007b). Real-time simulation of finite-element analysis permanent magnet synchronous machine drives on a FPGA card. pages 1–10.

Dufour, C., Bélanger, J., Ishikawa, T. et Uemura, K. (2005). Advances in Real-Time simulation of fuel cell hybrid electric vehicles.

Etxeberria-Otadui, I., Manzo, V., Bacha, S. et Baltes, F. (2002). Generalized average modelling of FACTS for real time simulation in ARENE. volume 2, pages 864–869 vol.2.

Everett, R. R. (1951). The Whirlwind I computer. *In AFIPS Joint Computer Conferences*, pages 70–74.

Fang, Z., Carletta, J. et Veillette, R. (2005). A methodology for FPGA-based control implementation. *Control Systems Technology, IEEE Transactions on*, 13(6) :977–987.

Faruque, M. et Dinavahi, V. (2010). Hardware-in-the-Loop simulation of power electronic systems using adaptive discretization. *Industrial Electronics, IEEE Transactions on*, 57(4) : 1146–1158.

Figueroa, H., Monti, A. et Wu, X. (2004). An interface for switching signals and a new real-time testing platform for accurate hardware-in-the-loop simulation. volume 2, pages 883–887 vol. 2.

Forsyth, P. et Kuffel, R. (2007). Utility applications of a RTDS® simulator. pages 112–117.

Gayed, A., Benkhoris, M., Siala, S. et Le Doeuff, R. (1994). Simulation tool for the study of pmsm control techniques. *In Computers in Power Electronics, 1994., IEEE 4th Workshop on*, pages 215–220. IEEE.

George, A. et Liu, J. (1989). The evolution of the minimum degree ordering algorithm. *Siam review*, 31(1) :1–19.

Gole, A., Keri, A., Kwankpa, C., Gunther, E., Dommel, H., Hassan, I., Marti, J., Martinez, J., Fehrle, K., Tang, L., McGranaghan, M., Nayak, O., Ribeiro, P., Iravani, R. et Lasseter, R. (1997). Guidelines for modeling power electronics in electric power engineering applications. *Power Delivery, IEEE Transactions on*, 12(1) :505–514.

Gombert, C. (2005). *Simulation temps-réel des dispositifs d'Electronique de Puissance dédiés aux Réseaux d'Energie Electrique*. Thèse de doctorat, INPG, Grenoble.

Gray, H. J., Rubinoff, M. et Sohon, H. (1953). A technique for real time simulation of a rigid body problem. *Mathematics of Computation*, 7 :73–73.

Guo, H., Zhou, B., Li, J., Cheng, F. et Zhang, L. (2009). Real-time simulation of BLDC-based wind turbine emulator using RT-LAB. pages 1–6.

Harakawa, M., Dufour, C., Nishimura, S. et Nagano, T. (2009). Real-time simulation of a PMSM drive in faulty modes with validation against an actual drive system. pages 1–9.

Hautier, J. et Caron, J. (1999). *Convertisseurs statiques : méthodologie causale de modélisation et de commande*. Editions TECHNIP.

Hautier, J. et Manesse, G. (1989). Analyse des systèmes électrotechniques par réseaux de pétri. *Les techniques de l'Ingénieur*, pages 1–20.

Horn, R. A. et Johnson, C. R. (1990). *Matrix Analysis*. Cambridge University Press.

huei Lu, T. et Meyer, W. S. (1987). *Electro-magnetic transient program (EMTP) Theory book*.

Hui, S. et Christopoulos, C. (1990). A discrete approach to the modeling of power electronic switching networks. *Power Electronics, IEEE Transactions on*, 5(4) :398–403.

Kailath, T. (1980). Linear systems. *Prentice-Hall information and system sciences series*.

Karimi, S., Poure, P. et Saadate, S. (2010). An HIL-Based reconfigurable platform for design, implementation, and verification of electrical system digital controllers. *Industrial Electronics, IEEE Transactions on*, 57(4) :1226–1236.

Kato, T., Inoue, K. et Fujiwara, Y. (2010). Multirate analysis method for a power electronic system by multi-domain partitioning. pages 1–8, Boulder, CO, USA.

Kato, T. et Tachibana, W. (1998). Periodic steady-state analysis of an autonomous power electronic system by a modified shooting method. *IEEE Transactions on Power Electronics*, 13(3) :522–527.

Kim, G., Kim, Y., Park, M., Yu, I. et Song, B. (2010). RTDS-based real time simulations of grid-connected wind turbine generator systems. pages 2085–2090.

Kim, K. H., Nguyen, C. et mo Park, C. (1996). Real-Time Simulation Techniques Based on the RTO.k Object Modeling. *In International Computer Software and Applications Conference*, pages 176–183.

Kuffel, R., Wierckx, R., Duchen, H., Lagerkvist, M., Wang, X., Forsyth, P. et Holmberg, P. (1995). Expanding an analogue HVDC simulator's modelling capability using a Real-Time digital simulator (RTDS. page 199.

Larsson, T., Hasler, J. et Paul Forsyth, T. M. (2007). Voltage source converter modeled in RTDS - experiences and comparison with field results. pages IPST07 – 76.

Le Doeuff, R. et Benkhoris, M. (1995). General principles and new trends in the simulation of static converters and drives. *Mathematics and Computers in Simulation*, 38(4–6) :263 – 270.

Le-Huy, H. (2001). Modeling and simulation of electrical drives using MATLAB/Simulink and power system blockset. volume 3, pages 1603–1611 vol.3.

Le-Huy, P. (2006). *Simulation Temps Réel De Convertisseurs De Puissance À L'aide De FPGA*. Thèse de doctorat, Université Laval.

Li, W., Joos, G. et Belanger, J. (2010). Real-Time simulation of a wind turbine generator coupled with a battery supercapacitor energy storage system. *Industrial Electronics, IEEE Transactions on*, 57(4) :1137–1145.

Lu, B., Wu, X., Figueroa, H. et Monti, A. (2007). A Low-Cost Real-Time Hardware-in-the-Loop testing approach of power electronics controls. *Industrial Electronics, IEEE Transactions on*, 54(2) :919–931.

Luciano, A. et Strollo, A. (1990). A fast time-domain algorithm for the simulation of switching power converters. *IEEE Transactions on Power Electronics*, 5(3) :363–370.

Maksimovic, D. (2010). Automated steady-state analysis of switching power converters using a general-purpose simulation tool. pages 1352–1358, St. Louis, MO, USA.

Marti, J. (1982). Accurate modelling of Frequency-Dependent transmission lines in electromagnetic transient simulations. *IEEE Transactions on Power Apparatus and Systems*, PAS-101(1) :147–157.

Massarini, A., Reggiani, U. et Kazimierczuk, M. (1997). Analysis of networks with ideal switches by state equations. *Circuits and Systems I : Fundamental Theory and Applications, IEEE Transactions on*, 44(8) :692–697.

Matar, M. et Iravani, R. (2010a). FPGA implementation of the power electronic converter model for Real-Time simulation of electromagnetic transients. *Power Delivery, IEEE Transactions on*, 25(2) :852–860.

Matar, M. et Iravani, R. (2010b). A modified multiport Two-Layer network equivalent for the analysis of electromagnetic transients. *Power Delivery, IEEE Transactions on*, 25(1) :434–441.

Mercier, P., Gagnon, C., Tetreault, M. et Toupin, M. (1995). Real-Time digital simulation of power systems at Hydro-Quebec. page 9.

Mohammed, O., Abed, N. et Ganu, S. (2007). Real-Time simulations of electrical machine drives with Hardware-in-the-Loop. pages 1–6.

Monmasson, E. et Cirstea, M. (2007). FPGA design methodology for industrial control Systems—A review. *Industrial Electronics, IEEE Transactions on*, 54(4) :1824–1842.

Myaing, A. et Dinavahi, V. (2010). FPGA-Based Real-Time emulation of power electronic systems with detailed representation of device characteristics. *IEEE Transactions on Industrial Electronics*.

Newman, M. et Holmes, D. (2003). Delta operator digital filters for high performance inverter applications. *Power Electronics, IEEE Transactions on*, 18(1) :447–454.

Paiz, C., Kettelhoit, B. et Porrmann, M. (2007). A design framework for FPGA-based dynamically reconfigurable digital controllers. pages 3708–3711.

Pak, L. et Dinavahi, V. (2009). Real-Time simulation of a wind energy system based on the Doubly-Fed induction generator. *Power Systems, IEEE Transactions on*, 24(3) :1301–1309.

Paquin, J., Li, W., Belanger, J., Schoen, L., Peres, I., Olariu, C. et Kohmann, H. (2009). A modern and open real-time digital simulator of All-Electric ships with a multi-platform co-simulation approach. pages 28–35.

Parma, G. et Dinavahi, V. (2007). Real-Time digital hardware simulation of power electronics and drives. *Power Delivery, IEEE Transactions on*, 22(2) :1235–1246.

Pejovic, P. et Maksimovic, D. (1994). A method for fast time-domain simulation of networks with switches. *Power Electronics, IEEE Transactions on*, 9(4) :449–456.

Pekarek, S., Wasynczuk, O., Walters, E., Jatskevich, J., Lucas, C., Wu, N. et Lamm, P. (2004). An efficient multirate simulation technique for Power-Electronic-Based systems. *IEEE Transactions on Power Systems*, 19(1) :399–409.

Pimentel, J. G. (2006). Implementation of simulation algorithms in FPGA for real time simulation of electrical networks with power electronics devices. pages 1–8, San Luis Potosi, Mexico.

Rakotozafy, M., Poure, P., Saadate, S., Bordas, C. et Leclere, L. (2011). Real-time digital simulation of power electronics systems with neutral point piloted multilevel inverter using FPGA. *Electric Power Systems Research*, 81(2) :687–698.

Reissig, G. (2007). Fill reduction techniques for circuit simulation. *Electrical Engineering*, 90(2) :143–146.

Robbins, A. H. et College, M. R. R. C. (2003). *Circuit Analysis : Theory and Practice*. Delmar Cengage Learning, 2nd revised edition édition.

Rubinoff, M. (1955). Digital Computers for Real-Time Simulation. *Journal of The ACM*, 2 :186–204.

Schilders, W. et Maten, E. T. (2005). *Numerical Methods in Electromagnetics : Special Volume*. Elsevier.

Schönhage, A. (1981). Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3) :434–455.

Schutt-Aine, J. (2001). Latency insertion method (LIM) for the fast transient simulation of large networks. *Circuits and Systems I : Fundamental Theory and Applications, IEEE Transactions on*, 48(1) :81–89.

Siala, S., Bordas, C., Pouliquen, J. et Benkhoris, M. (2004). Embedded real time simulator for complex power system. *In Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, volume 5, pages 3628–3634. IEEE.

Singhal, K., Vlach, J. et Vlach, M. (1975). Numerical inversion of multidimensional laplace transform. *Proceedings of the IEEE*, 63(11) :1627–1628.

Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4) :354–356.

Strassen, V. (1988). The asymptotic spectrum of tensors. *Journal für die reine und angewandte Mathematik*, 384 :102–152.

Tao, H., Yong-hua, Y., Wei-ping, J., Yi-ying, Z. et Jin-hua, Z. (2008). Study on method of Digital/Analog hybrid simulation for the Ultra-High voltage grid. pages 1–5.

Thiberville, A. J. (1958). Facilities and instrumentation required for real-time simulation involving system hardware. *In AFIPS Joint Computer Conferences*, pages 96–100.

Thompson, C. A. (1992). A study of numerical integration techniques for use in the companion circult method of transient circuit analysis. *ECE Technical Reports*, page 297.

Trevor, M. et James, G. (2005). Small time-step ($< 2\mu$Sec) VSC model for the real time digital simulator. pages IPST05 – 168.

Uchida, T. et McPhee, J. (2011). Triangularizing kinematic constraint equations using Gröbner bases for real-time dynamic simulation. *Multibody System Dynamics*, 25 :335–356.

Verma, S., Odani, H., Ogawa, S., Kuroda, K. et Kono, Y. (2006). Real time interface for interconnecting fully digital and analog simulators using short line or transformer. page 7 pp.

Vlach, J. et Opal, A. (1997). Modern CAD methods for analysis of switched networks. *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, 44(8) :759–762.

Vlach, J. et Singhal, K. (1983). *Computer methods for circuit analysis and design*. Van Nostrand Reinhold Company, University of Waterloo.

Wong, R. C., Owen, H. A. et Wilson, T. G. (1987). An efficient algorithm for the Time-Domain simulation of regulated Energy-Storage DC-to-DC converters. *Power Electronics, IEEE Transactions on*, PE-2(2) :154–168.

Résumé

Les contrôleurs industriels font l'objet de changements de paramètres, de modifications, d'améliorations en permanence. Ils subissent les évolutions technologiques aussi bien matérielles que logicielles (librairies, système d'exploitation, loi de commande...). Malgré ces contraintes, ces contrôleurs doivent obligatoirement assurer toutes les fonctionnalités recouvrant le séquentiel, les protections, l'interface homme machine et la stabilité du système à contrôler. Ces fonctionnalités doivent être couvertes pour une large gamme d'applications.

Chaque modification (matérielle ou logicielle) quoique mineure est risquée. Le debogage, l'analyse et la programmation sur site sont énormément coûteux surtout pour des sites de type offshore ou marine. Les conditions de travail sont difficiles et les tests sont réduits au strict minimum.

Cette thèse propose deux niveaux de validation en plateforme d'expérimentation :
– un niveau de validation algorithmique que l'on appelle Validation par Interface Logicielle (VIL) traitée au chapitre 2 ;
– un niveau de validation physique que l'on appelle Validation par Interface Matérielle (VIM) traitée au chapitre 3.

La VIL valide uniquement l'aspect algorithme, la loi de commande et la conformité des références au niveau calcul sans prendre en compte les signaux de commande physiques et les signaux de retour gérés par l'Unité de Gestion des Entrées/Sorties (UGES). Un exemple de validation d'un contrôleur industriel d'un ensemble convertisseur trois niveaux et machine asynchrone est traité dans le deuxième chapitre avec une modélisation particulièrement adaptée à la VIL.

Le dernier chapitre traite la VIM sur différentes bases matérielles (Field Programmable Gate Array (FPGA), processeurs). Cette validation prend en compte l'aspect algorithme et les signaux de commande physique ainsi que les signaux de retour. On y présente plusieurs approches de modélisation, choisies selon la base matérielle d'implémentation du simulateur temps réel.

Ces travaux ont contribué aujourd'hui à au processus de validation des contrôleurs dédiés aux applications Oil and Gaz et Marine de General Electric - Power Conversion © (GE-PC).

Mots clés : simulation temps réel, électronique de puissance, convertisseurs statiques, validation de contrôleurs industriels, validation par interface logicielle, validation par interface matérielle, FPGA, processeur

Abstract

Industrial controllers are always subjected to parameters change, modifications and permanent improvements. They have to follow off-the-shelf technologies as well as hardware than software (libraries, operating system, control regulations ...).
Apart from these primary necessities, additional aspects concerning the system operation that includes sequential, protections, human machine interface and system stability have to be implemented and interfaced correctly. In addition, these functions should be generically structured to be used in common for wide range of applications.

All modifications (hardware or software) even slight ones are risky. In the absence of a prior validation system, these modifications are potentially a source of system instability or damage. On-site debugging and modification are not only extremely expensive but can be highly risky, cumulate expenditure and reduce productivity. This concerns all major industrial applications, Oil & Gas installations and Marine applications. Working conditions are difficult and the amount of tests that can be done is strictly limited to the mandatory ones.

This thesis proposes two levels of industrial controller validation which can be done in experimental test platform :
– an algorithm validation level called Software In the Loop (SIL) treated in the second chapter ;
– a physical hardware validation called Hardware In the Loop (HIL) treated in the third chapter.

The SIL validates only the control algorithm, the control law and the computed references without taking into account neither the actual physical commands nor the physical input feedbacks managed by the Input/Output boards. SIL validation of the system where industrial asynchronous motor is fed and regulated by a three level Variable Speed Drive with a three level voltage source converter is treated in the second chapter with a particular modeling approach adapted to such validation.

The last chapter presents the HIL validation with various hardware implementations (Field Programmable Gate Array (FPGA), processors). Such validation checks both the control algorithm and the actual physical Input/Output signals generated by the dedicated boards. Each time, the modeling approach is chosen according to the hardware implementation.

Currently this work has contributed to the system validation used by General Electric - Power Conversion © (GE-PC) as part of their validation phase that is mandatory for Oil & Gas projects and Marine applications.

Keywords : real time simulation, power electronics, switching converters, industrial controller validation, Hardware In the Loop, Software In the Loop, FPGA, processor.