



HAL
open science

Amélioration de la vitesse de calcul et de la précision des outils de simulation des phénomènes transitoires électromagnétiques sur les réseaux électriques

Boris Bruned

► **To cite this version:**

Boris Bruned. Amélioration de la vitesse de calcul et de la précision des outils de simulation des phénomènes transitoires électromagnétiques sur les réseaux électriques. Energie électrique. Nantes Université, 2023. Français. NNT : 2023NANU4003 . tel-04087034

HAL Id: tel-04087034

<https://theses.hal.science/tel-04087034>

Submitted on 2 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DOSSIER DE VAE DOCTORALE

NANTES UNIVERSITE

ECOLE DOCTORALE N° 641

*Mathématiques et Sciences et Technologies du numérique,
de l'Information et de la Communication*

Spécialité : Génie Electrique

Par

Boris BRUNED

Amélioration de la vitesse de calcul et de la précision des outils de simulation des phénomènes transitoires électromagnétiques sur les réseaux électriques

VAE Doctorale présentée et soutenue à Saint-Nazaire, le 26/01/2023

Rapporteurs avant soutenance :

Mario PAOLONE
Seddik BACHA

Full Professor, EPFL Lausanne
Professeur des Universités, Université Grenoble-Alpes

Composition du Jury :

Président : Didier TRICHET
Examineurs : Corinne MIRAL
Jean MAHSEREDJIAN
Nicolas BRACIKOWSKI
Sébastien DENNETIERE
Réf. VAE HDR : François AUGER

Professeurs des Universités, Nantes Université
Maître de conférences HDR, Nantes Université
Professeur Titulaire, Polytechnique Montréal
Maître de conférences, Nantes Université
Docteur, RTE Lyon
Professeurs des Universités, Nantes Université

Invité(s)

Bertrand CLERC Ingénieur, RTE Lyon

Remerciements

Tout d'abord, je tiens à remercier l'encadrement dont j'ai bénéficié tout le long de cette VAE doctorale. Il a contribué indéniablement à la réussite de ce projet.

Danièle Ferré, consultante VAE, a permis de lancer cette rédaction dans la mise en valeur de mon parcours professionnel et de mes compétences de chercheur.

François Auger est un enseignant référent sérieux et exigeant. Sa relecture minutieuse a largement magnifié mes premiers jets. J'ai appris la rigueur qui s'impose à toute production scientifique et comment rendre accessible et lisible mes travaux de recherche. J'ai pu réaliser tout ce que l'on attend d'un futur docteur.

Nicolas Bracikowski a été un premier contact qui m'a permis d'initier cette démarche de VAE doctorale auprès de l'Université de Nantes. Il m'a adressé aux bonnes personnes pour constituer et déposer le dossier de recevabilité. Il a également veillé au bon déroulement et avancement de cette démarche.

Je n'oublie pas le service VAE de l'Université de Nantes qui a toujours répondu rapidement à mes questions et dont la réactivité a été précieuse.

Une carrière professionnelle est faite de rencontres et d'opportunités. Dès mes premiers échanges avec Jean Mahseredjian, professeur à Polytechnique Montréal, j'ai compris qu'une nouvelle orientation s'offrait à moi. Il m'a beaucoup appris sur le domaine de la simulation des transitoires électromagnétiques des réseaux électriques. Inlassablement, il m'a expliqué les mystères de la Jacobienne et la genèse de la Compensation. Le cœur des travaux présentés dans cette VAE doctorale est issu de cette riche collaboration. De Paris à Montréal en passant par Lyon, d'un opéra à simplement un verre, Jean Mahseredjian m'a surtout transmis le goût de la recherche où questionnement perpétuel, rigueur scientifique et perfectionnisme procurent satisfactions et joies.

Une autre rencontre marquante est celle de Sébastien Dennetière. A RTE, il m'a initié à la recherche appliquée. Ce travail n'aurait pas vu le jour si Sébastien ne m'avait pas sollicité en 2016 pour rédiger mon tout premier article en vue de la conférence IPST. Ce fut le début d'une collaboration scientifique riche et fructueuse. Sébastien m'a montré qu'une recherche de pointe est possible au sein d'une entreprise et qu'elle peut mener au doctorat.

Transmission et inspiration, c'est à cela que l'on reconnaît un mentor. Encore merci à vous deux.

Je souhaite également exprimer ma profonde gratitude aux rapporteurs et membres du jury qui ont accepté de lire mon manuscrit : Mario Paolone, Seddik Bacha, Corinne Miral et Didier Trichet.

Je tiens à remercier les chercheurs de l'institut de recherche d'Hydro-Québec : Jean-Claude Soumagne, François Guay, Olivier Tremblay, Philippe Le-Huy, Eric Lemieux. Ils m'ont fait découvrir le monde de la simulation temps réel qui est au cœur de la plupart de mes travaux. Je remercie aussi Omar Saad et Anas Abusalah pour nos discussions approfondies sur la parallélisation. Je n'oublie pas Loic Schoen d'Opal-RT Technologie qui m'a guidé dans les méandres de la compilation d'HYPERSIM.

Cette VAE doctorale n'aurait pas été possible sans un soutien indéfectible de ma hiérarchie : Bertrand Clerc, Samuel Nguéfeu et Olivier Mazerolles. Dans un poste précédent, Bertrand Clerc m'a aidé également à faire mes premiers pas dans le code d'HYPERSIM.

J'ai pu bénéficier d'un cadre stimulant et chaleureux au sein de mon équipe D3EP à RTE. Particulièrement, je tiens à remercier mes compagnons de code, César, Mehdi, nos illustres chercheurs inspirants (thèse ou pas thèse ?), Hani, Pierre, nos spécialistes d'études Ambroise, Julien, Yannick, Xavier-Marie, Quentin, Mickael, Fabien, de maintenance, Samy, Mathieu, Elias, Vinicius, Benjamin, Baptiste, de projet, Marco, Janek, Phuong, Laure. Tous vous avez favorisé la réussite de ce projet. Ma gratitude va également à Ian, Manuel et Jean pour une collaboration interne productive sur la répartition des tâches.

Enfin, je remercie mes parents et mes deux frères qui m'ont constamment soutenu durant ce projet. Souvent inspireurs, Chantal et José ont toujours favorisé mes espérances. Yvain et Vianney, bonheur de mon enfance, vous êtes mes fidèles compagnons de route.

Sigles et Abréviations

Δt	Pas de temps de la simulation
Y_n	Matrice d'admittance issue de l'analyse nodale
v_n	Vecteur des tensions aux nœuds
i_n	Vecteur des injections de courant aux nœuds
v_{km}	Tension de branche entre les nœuds k et m
(Z_{th}, v_{th})	Equivalent de Thévenin, impédance et tension de Thévenin
i_c	Vecteur des courants de compensation
$e\%$	Erreur relative en pourcentage
$J^{(k)}$	Matrice Jacobienne à l'itération k
$V(\mathbf{G})$	Ensemble des sommets d'un graphe \mathbf{G}
$E(\mathbf{G})$	Ensemble des arêtes d'un graphe \mathbf{G}
R	Matrice d'adjacence de la fermeture transitive d'un graphe \mathbf{G}
φ	Fonction du bloc du feedback ou de rétroaction
e	Variables de coupe ou de feedback (rétroaction)
A*	Heuristique de recherche A*
AC	Courant alternatif
BBD	Forme matricielle diagonale bordée par blocs
BDF	Forme matricielle diagonale par blocs
BTF	Forme matricielle triangulaire par blocs
CIGRE	Conseil International des Grands Réseaux Electriques
CM	Méthode de Compensation
CSPR	Compensateur Statique de Puissance Réactive
DAE	Equations différentielles Algébriques
DB	Méthode de découplage basée sur l'insertion d'un délai artificiel
DC	Courant Continu
DFIG	Modèle d'éolienne à générateur asynchrone à double alimentation

DFS	Depth-First Search (Parcours en Profondeur)
DFSOpt	Variante du DFS pour résoudre le FAS minimal
EMT	Electromagnetic Transients (Transitoires Electromagnétiques)
FAS	Feedback Arc Set (Ensemble des arcs de rétroaction)
FSC	Modèle éolien à convertisseur pleine échelle
FDNE	Frequency Dependent Network Equivalent (Equivalent Fréquentiel de Réseau)
GPA	Algorithme glouton pour le partitionnement local de graphe
GreedyFAS	Algorithme glouton pour la résolution du FAS minimal
HIL	Hardware-In-the-Loop
HVDC	High Voltage Direct Current (Courant Continu à très Haute Tension)
IGBT	Transistors bipolaires à grille isolée
IO	Boitier d'entrée/sorties
JM	Méthode basée sur la formulation de la matrice Jacobienne
LCC	Convertisseur en source de courant
LD	Méthode de découplage basée sur le Délai de propagation des Lignes
MANA	Analyse nodale augmentée modifiée
ML	Multi-niveaux
PLL	Boucle à verrouillage de phase
RB	Bipartition Réursive
RJM	Méthode basée sur la formulation de la Jacobienne réduite
RTE	Réseau de Transport d'Electricité
SCC	Composantes fortement connexes d'un graphe
Seed	Graine aléatoire
SEQ	Méthode séquentielle sans parallélisation pour les équations de réseaux, séquence selon le DFS pour les équations de contrôle
SortFAS	Tri par insertion pour résoudre le FAS minimal
TAP	Task Allocation Problem (Problème d'Allocation de Tâches)
VSC	Convertisseur en source de tension

SOMMAIRE

REMERCIEMENTS	I
SIGLES ET ABREVIATIONS	III
INTRODUCTION.....	1
CHAPITRE 1 MON PARCOURS	2
1.1 INTRODUCTION.....	2
1.2 NOTICE BIOGRAPHIQUE	3
1.3 DESCRIPTION DU PARCOURS	3
1.4 FORMATIONS	4
1.4.1 <i>Formation initiale</i>	4
1.4.2 <i>Formation professionnelle et continue</i>	6
1.5 MON PARCOURS PROFESSIONNEL.....	8
1.5.1 <i>Ingénieur Logiciel chez FACTSET (2013-2015)</i>	8
1.5.2 <i>Ingénieur R&D chez RTE (2015-)</i>	9
1.5.2.1 Contexte et enjeux.....	10
1.5.2.2 Mes Débuts (2015-2016)	14
1.5.2.3 Maîtrise et Encadrement (2017-2018).....	15
1.5.2.4 Expertise et Recherche (2019-).....	16
1.5.2.5 Compétences acquises.....	17
1.6 CONCLUSION GENERALE SUR LES EXPERIENCES DE MON PARCOURS.....	18
CHAPITRE 2 INTRODUCTION GENERALE AUX CONTRIBUTIONS.....	19
2.1 CONTEXTE ET MOTIVATIONS	19
2.2 RESUME DES TRAVAUX	20
2.3 CONTRIBUTIONS	21
2.3.1 <i>Liste des publications</i>	22
2.3.1.1 Articles de Journal.....	22
2.3.1.2 Articles de Conférence.....	23
2.3.1.3 Rapports Internes	23
2.3.1.4 Prépublications de journal	23
CHAPITRE 3 EQUIVALENTS FREQUENTIELS DE RESEAUX	24
3.1 INTRODUCTION.....	24
3.2 RESOLUTION DES EQUATIONS DE RESEAUX.....	24
3.2.1 <i>Réseaux linéaires</i>	24
3.2.2 <i>Analyse Nodale Augmentée Modifiée</i>	30
3.2.3 <i>Réseaux non-linéaires</i>	30
3.2.4 <i>Principe de la simulation temps réel</i>	32
3.3 PRINCIPE DES EQUIVALENTS DE RESEAUX.....	34

3.4	GENERATION DES EQUATIONS D'ETAT D'UN EQUIVALENT FREQUENTIEL	35
3.5	INTEGRATION DES EQUATIONS D'ETAT A LA RESOLUTION DE RESEAU	36
3.6	IMPLEMENTATION POUR LE TEMPS REEL.....	37
3.7	PERFORMANCE ET VALIDATION	38
3.7.1	<i>Cas d'un réseau triphasé simple</i>	38
3.7.2	<i>Cas de la modélisation du câble d'une liaison HVDC</i>	40
3.8	CONCLUSION.....	41
CHAPITRE 4 PARALLELISATION DES RESOLUTIONS DE RESEAUX : SEPARATION EN TACHES		43
4.1	PRINCIPE DE LA PARALLELISATION DES RESOLUTIONS DE RESEAUX.....	43
4.2	PRINCIPE DE LA SEPARATION EN TACHES	43
4.3	PARALLELISATION AVEC DELAI	44
4.3.1	<i>Délai de propagation des lignes</i>	44
4.3.2	<i>Délai artificiel de lignes virtuelles</i>	45
4.4	PARALLELISATION SANS DELAI	45
4.4.1	<i>Approche topologique</i>	45
4.4.2	<i>Méthode de Compensation</i>	46
4.4.3	<i>Approche matricielle</i>	48
4.4.3.1	Forme BBD (diagonale bordée par blocs)	48
4.4.3.2	Résolution en parallèle forme BBD.....	49
4.4.4	<i>Automatisation de la séparation en tâches</i>	52
4.5	IMPLEMENTATION DE LA METHODE DE COMPENSATION.....	53
4.5.1	<i>Mécanisme de parallélisation</i>	53
4.5.2	<i>Méthode de Compensation non-linéaire</i>	54
4.5.3	<i>Multiplicateur de courant idéal avec découplage</i>	56
4.6	PERFORMANCE ET VALIDATION	58
4.6.1	<i>Réseaux linéaires</i>	59
4.6.1.1	Réseaux de distribution	59
4.6.1.2	Réseaux HVDC avec commutations	61
4.6.2	<i>Réseaux non-linéaires</i>	63
4.6.2.1	Réseaux non-linéaires avec commutations	63
4.6.2.2	Réseaux avec des caractéristiques non-linéaires.....	74
4.7	CONCLUSION.....	76
CHAPITRE 5 PARALLELISATION DES RESOLUTIONS DE RESEAUX : REPARTITION DES TACHES SUR LES PROCESSEURS		78
5.1	PRINCIPE	78
5.2	METHODES DE RESOLUTION	79
5.2.1	<i>Approche du Bin Packing</i>	79
5.2.2	<i>Approche TAP</i>	81
5.2.2.1	Recherche A*	82

5.2.2.2	Partitionnement de graphe.....	85
5.3	PARTITIONNEMENT MULTI-NIVEAUX DE GRAPHE.....	86
5.3.1.1	Principe de l'algorithme.....	86
5.3.2	<i>Un algorithme rapide de partitionnement.....</i>	86
5.3.2.1	Limites.....	88
5.3.3	<i>Recherche d'un paramétrage optimal de l'algorithme.....</i>	89
5.3.4	<i>Validation de l'optimalité avec les méthodes exactes.....</i>	89
5.4	ETUDES DES PERFORMANCES.....	90
5.4.1	<i>Etudes comparatives des heuristiques de partitionnement de graphe sur de grandes instances</i> <i>90</i>	
5.4.2	<i>Recherche du paramétrage optimal pour le partitionnement de graphe multi-niveaux.....</i>	93
5.4.3	<i>Validation de la qualité des solutions obtenues.....</i>	94
5.4.4	<i>Performances temps réel.....</i>	96
5.4.4.1	Cas réseau HVDC CIGRE.....	96
5.4.4.2	Cas réseau HVDC projet Best Paths.....	101
5.5	CONCLUSION.....	104
CHAPITRE 6 METHODES D'ACCELERATION DES RESOLUTIONS DE SYSTEMES DE CONTROLE.....		105
6.1	INTRODUCTION.....	105
6.2	PRINCIPE DE RESOLUTION DES EQUATIONS DE CONTROLE.....	106
6.3	FORMULATION JACOBIENNE POUR RESOUDRE LES BOUCLES DE RETROACTION.....	108
6.4	FORMULATION OPTIMISEE DE LA JACOBIENNE REDUITE.....	111
6.4.1	<i>Principe.....</i>	111
6.4.2	<i>Calcul des séquences de boucle.....</i>	112
6.4.3	<i>Détection des coupes.....</i>	115
6.5	VALIDATION ET PERFORMANCES.....	118
6.5.1	<i>Analyse des graphes de contrôle.....</i>	119
6.5.2	<i>Validation.....</i>	120
6.5.2.1	Cas de la simulation d'une PLL.....	120
6.5.2.2	Cas de la simulation de l'arc secondaire d'un disjoncteur.....	121
6.5.2.3	Cas de la simulation de parcs éoliens.....	124
6.5.3	<i>Analyse des Performances.....</i>	126
6.6	CONCLUSION.....	129
CHAPITRE 7 CONCLUSION SUR LES CONTRIBUTIONS.....		130
CONCLUSION.....		132
REFERENCES 133		
ANNEXES 143		
A.	RESOLUTION D'UN PLL.....	143

A.1	<i>Formulation de la méthode Jacobiennne</i>	144
A.2	<i>Formulation de la Jacobiennne réduite</i>	146
B.	ALGORITHME DE KOSARAJU POUR LES COMPOSANTES FORTEMENT CONNEXES	151
C.	RECAPITULATIF DE LA DEMANDE DE VALIDATION DES ACQUIS DE L'EXPERIENCE	152
D.	ANNEXES « PEDAGOGIQUES » ILLUSTRATIVES ET LIEES A LA DESCRIPTION ET A L'ANALYSE DES EXPERIENCES	153

Introduction

Séoul, à la fin du mois de Juin 2017. D'une voix hésitante, je commence ma présentation en anglais dans la salle principale de la conférence internationale bisannuelle sur les Transitoires Electromagnétiques des systèmes électriques. Il s'agit de mon premier travail de recherche, qui porte sur l'unification de la modélisation utilisée par deux environnements de simulation, le temps réel et le temps différé. Des mois auparavant, ce travail a suivi le processus de sélection de la conférence : envoi d'un résumé, soumission d'un manuscrit final et réponses aux commentaires des arbitres. A travers cette première contribution scientifique, j'ai découvert en quoi consistent les compétences d'un chercheur dans la conception et l'élaboration d'une démarche de recherche et développement. J'ai pris goût à cette activité où, tout en contribuant au savoir scientifique, je suis capable de le transférer directement dans des applications pratiques que sont la simulation des transitoires électromagnétiques des réseaux électriques. Cinq années se sont écoulées depuis. Cette première conférence en a appelé plusieurs. Mes contributions scientifiques se sont multipliées en devenant des articles de revue. Ainsi, je suis maintenant en mesure de proposer une validation par les acquis de l'expérience d'un doctorat qui met en valeur mes contributions en recherche appliquée dans la simulation des réseaux électriques.

Dans un premier temps, je reviens sur mon parcours professionnel. Je présente mon cheminement professionnel, de mes études d'ingénieur à mon premier poste dans le développement informatique, puis à des activités de recherche sur la simulation des réseaux électriques. Mais avant tout, je mets en exergue les compétences de chercheur que j'ai acquises au cours de mon poste actuel à travers mes travaux de recherche, mes encadrements et la diffusion de mon savoir, que ce soit en l'interne ou à l'externe de mon entreprise.

Dans un second temps, je présente le contenu de mes recherches qui ont été regroupées au sein d'une même thématique, l'accélération de la vitesse de calcul et l'amélioration de la précision des simulations de réseaux électriques. Cette thématique est mise en relation avec les grands enjeux de la transition énergétique qui impactent les réseaux électriques. En effet, ils doivent s'adapter à l'intégration massive des énergies renouvelables. Pour chaque problématique de recherche présentée, mes contributions originales sont distinguées afin de mieux apprécier le travail effectué.

Chapitre 1 Mon Parcours

1.1 Introduction

Au cours de mes études supérieures, j'ai toujours été intéressé par des sujets techniques alliant à la fois les mathématiques appliquées et l'informatique. Travailler en tant que futur ingénieur en recherche et développement était une voie naturelle pour mon projet professionnel. Une thèse pouvait m'intéresser en me faisant approfondir un sujet technique tout en gagnant en rigueur scientifique. Sorti diplômé de Centrale Nantes en 2012, je n'ai pas eu l'opportunité de commencer une thèse. J'ai directement intégré le marché du travail dans le développement informatique. Par la suite, mon évolution professionnelle m'a fait redécouvrir mes premières aspirations. En 2015, j'ai intégré RTE (Réseau de Transport d'Electricité) en tant qu'ingénieur en Recherche et Développement, poste que j'occupe toujours actuellement. Je travaille sur des sujets techniques très intéressants tels que la simulation de Transitoires Electromagnétiques de réseau, qui allient à la fois des mathématiques appliqués, de l'informatique et de la physique. Mais surtout, un contexte favorable et fécond m'a amené à développer mon goût pour la recherche appliquée. La présence de nombreux docteurs au sein de mon équipe actifs dans la publication scientifique, ainsi que le partenariat avec une Chaire à Polytechnique Montréal, ont été une source d'émulation.

Mon goût pour la recherche appliquée, c'est avant tout mon envie d'apporter une contribution au savoir scientifique, d'échanger et de collaborer pour avancer, de partir du théorique vers une application concrète, de travailler sur des problèmes difficiles en croisant plusieurs domaines (informatique, génie électrique et mathématiques) pour les résoudre, de diffuser ses connaissances nouvelles via des conférences ou des publications, de transmettre via l'encadrement.

Tout cela, je l'ai trouvé dans mon poste actuel. C'est tout naturellement que je souhaite présenter une thèse par la validation des acquis de l'expérience. Cette thèse sera pour moi une occasion de revenir sur mon parcours en y montrant les différentes compétences acquises liées à la recherche. Surtout, cela rassemble mes différents travaux pour les valoriser dans leurs apports scientifiques. C'est aussi une reconnaissance de mon aptitude à mener des sujets de recherche dont la qualité a été approuvée par mes pairs. Indéniablement, cette validation doctorale sera une réussite dans ma carrière professionnelle, pour ses apports à la recherche appliquée, mais également sur le plan personnel par la stimulation intellectuelle qu'elle peut m'apporter.

Dans un premier temps, mon parcours professionnel sera abordé sous l'angle de mes formations. L'accent sera mis sur ma formation initiale d'ingénieur de l'Ecole Centrale de Nantes. Cette dernière m'a permis d'acquérir un socle de connaissances scientifiques solides et une expérience conséquente de l'entreprise. Seront aussi présentées les formations que j'ai pu suivre au cours de ma carrière professionnelle qui ont permis mon intégration chez RTE.

Dans un second temps, les différents éléments de ma carrière professionnelle seront exposés, que ce soit mon premier poste d'ingénieur logiciel chez FACTSET ou mon expérience chez RTE. Leur analyse permettra de comprendre les étapes qui m'ont amené du développement logiciel pour la finance aux méthodes de simulation pour les réseaux électriques, du métier d'ingénieur logiciel à celui de chercheur.

1.2 Notice biographique

ANNEE	FORMATIONS	EXPERIENCES PROFESSIONNELLES ET/OU SOCIALES ET PERSONNELLES
	FORMATIONS INITIALES	
2007	BACCALAUREAT BAC S, MENTION TRES BIEN, LYCÉE KÉRICHEN	
2007- 2009	CLASSE PREPARATOIRES AUX GRANDES ECOLES FILIERE MP, OPTION INFORMATIQUE LYCÉE KÉRICHEN	
2009 - 2012	DIPLOME D'INGENIEUR CENTRALE NANTES, OPTION INFORMATIQUE	Stage ouvrier (6 semaines) <i>Société Industrielle du Ponant (Landerneau)</i>
		Stage ingénieur (15 semaines) <i>La Poste (Nantes)</i>
		Stage de Fin d'Etudes (6 mois) <i>AMADEUS (Londres)</i>
2013- 2015		Ingénieur Développement Logiciel <i>FACTSET</i>
	FORMATIONS CONTINUES	
2015	ERSO <i>Electrotechnique de réseau</i>	Ingénieur Recherche et Développement <i>RTE</i> <i>Poste actuel</i>
2016	AUTO <i>Analyse et commande des systèmes dynamiques</i>	
2017	SSFDR <i>Sensibilisation à la Sûreté Système par le fonctionnement dynamique des réseaux</i>	
2019- 2020	ICIP2019 <i>International Cross-Innovation Program sur l'Agilité</i> Etude EMTP® <i>Etude des lieux d'impédances harmoniques pour Aquind</i>	

1.3 Description du parcours

Diplômé de Centrale Nantes, j'ai débuté ma carrière en tant qu'ingénieur logiciel chez FACTSET, un éditeur de logiciels américain spécialisé dans l'information financière. Cela me permettait de valoriser l'option informatique que j'avais suivie durant ma formation d'ingénieur. Plus précisément, j'ai travaillé sur un outil d'aide à la saisie d'informations financières pour Factset Estimates

(prévisions des résultats financiers des entreprises) et Factset Economics (prévisions économiques). En tant qu'ingénieur logiciel, je prenais part à des cycles de développement de 3 mois qui allaient de la prise en compte des besoins des utilisateurs à l'implémentation et aux tests de nouvelles fonctionnalités. Si cette expérience professionnelle a été très formatrice dans le développement informatique, elle ne répondait pas totalement à mes attentes en termes d'évolution professionnelle et de stimulation intellectuelle. Les cycles de développement étaient répétitifs, et la faible taille du groupe en France n'offrait pas beaucoup de perspectives d'évolution.

En 2015, j'ai décidé de rejoindre RTE (Réseau de Transport d'Electricité), le gestionnaire du réseau français de transport d'électricité, en tant qu'ingénieur R&D, poste que j'occupe toujours actuellement. Au départ, le poste proposé capitalisait sur l'expertise que j'avais acquise à FACTSET dans le développement informatique chez un éditeur de logiciels. Je participe au développement d'HYPERSIM [1], un logiciel de simulation temps réel des transitoires électromagnétiques sur le réseau, dans le cadre d'une collaboration avec Hydro-Québec, le producteur et transporteur d'électricité au Québec. Mais progressivement, mes activités ont glissé vers le domaine de la R&D. En effet, j'ai dû apprendre un nouveau domaine qu'est la simulation des transitoires électromagnétiques des réseaux électriques. J'ai dû ainsi comprendre la physique des modèles de transitoires et les méthodes de résolution issues des mathématiques appliquées. Le projet de renouvellement des outils m'a permis de faire une démarche de recherche dans l'identification des méthodes de résolution qui seront implémentées. Ceci a amené à des publications scientifiques et à leur présentation lors de conférences internationales. Un travail de veille a également enrichi cette démarche en étant arbitre pour des revues scientifiques. Par ailleurs, j'ai effectué des encadrements de stage de fin d'études d'école d'ingénieurs sur des sujets techniques et de recherches. J'ai encadré un alternant ingénieur pendant trois ans.

1.4 Formations

1.4.1 Formation initiale

Intéressé par les matières scientifiques (mathématiques et physique), j'ai obtenu un baccalauréat série S en 2007 avec une mention très bien au lycée Kérichen à Brest. J'ai poursuivi des études scientifiques dans le même lycée en classes préparatoires mathématiques et physique, option informatique. A la suite des concours d'écoles d'ingénieurs, j'ai intégré en 2009 Centrale Nantes, une école d'ingénieurs. Tout en satisfaisant mon goût pour les matières scientifiques, ceci me permettait d'affiner mon projet professionnel grâce aux aspects généraliste et pratique (nombreux stages) de cette formation. Un tronc commun de 18 mois m'a permis de consolider mes connaissances en mathématiques appliquées, mécanique et en informatique. En dernière année, je me suis spécialisé en informatique. De connaissances plus conceptuelles, je suis passé à une approche plus pratique de l'informatique dans le développement concret de programmes via différents langages (C, Java, C++) et la découverte de différentes technologies (développement web, base de données...).

Cette formation a également été professionnalisante. J'ai réalisé sur les trois ans 11 mois de stage dont 6 mois à l'étranger.

Mon premier stage s'est déroulé pendant 6 semaines dans l'usine de l'Industrielle du Ponant à Landerneau, spécialisée dans la tôlerie-chaudronnerie. En tant qu'exécutant, j'ai participé à

différentes activités comme l'usinage et le conditionnement de pièces, la peinture ou le pliage. J'ai pu appréhender les contraintes en matière d'organisation et de sécurité d'un atelier mécanique.

En deuxième année, j'ai réalisé un stage ingénieur à La Direction Technique du Courier de La Poste à Nantes. Ce stage consistait à appliquer des techniques de recherche opérationnelle pour optimiser le tri du courrier. Cette activité de recherche et développement répondait à mes attentes en me permettant de mettre en œuvre mes compétences acquises lors de ma formation comme le développement informatique et les techniques d'optimisation mathématique.

En dernière année de formation, je suis parti à Londres pendant 6 mois pour réaliser mon stage de fin d'études chez Amadeus, un éditeur de logiciels pour l'industrie du voyage. J'ai travaillé sur l'optimisation du processus d'Onload qui est lancé juste avant l'embarquement d'un vol pour gérer les priorités vis-à-vis de la surréservation et du sur-classement. Comme pour mon précédent stage, j'ai utilisé des techniques de recherche opérationnelle pour optimiser tous les critères du problème. J'ai pu découvrir l'organisation d'un éditeur de logiciels où chaque équipe de développement s'occupe d'une fonctionnalité spécifique et a son équivalent fonctionnel pour définir le besoin du client. Mon niveau en anglais s'est sensiblement amélioré en travaillant dans un environnement international.

A travers ces premières expériences professionnelles, j'ai ainsi découvert des domaines d'activité différents (industrie, services, informatique pour l'aérien) et monté en compétences, du rôle d'exécutant à celui d'ingénieur. J'ai également surtout développé une appétence pour des sujets techniques qui allient à la fois de l'informatique et des mathématiques appliquées comme en témoignent mes deux stages en recherche opérationnelle. Je souhaitais retrouver ces deux composantes dans ma carrière professionnelle.

Diplôme	Dates établissement	Résumé du diplôme / compétences acquises
Bac S	Lycée Kérichen	Série Scientifique, option Mathématiques
L2	Lycée Kérichen	Classes préparatoires scientifiques, filière MP, Option Informatique
Diplôme d'Ingénieur	Centrale Nantes	Option Informatique

DIPLOME D'INGENIEUR DE CENTRALE NANTES
<p>Objectifs</p> <p>Diplôme d'Ingénieur</p> <p>Programme</p> <p>18 mois de tronc commun : mathématiques, mécanique et informatique 1 semestre différencié : Web marketing</p>

Dernière année de spécialisation en informatique
11 mois de stage dont 3 mois obligatoires à l'étranger

Stage

2010 : Stage ouvrier de 6 semaines à l'Industrielle du Ponant (Landernau)

2011 : Stage ingénieur de 15 semaines à la Direction Technique de Courrier de La Poste (Nantes)
Recherche Opérationnelle, optimisation du tri départ

2012 : Stage de Fin d'Etude de 6 mois à l'étranger chez AMADEUS (Londres)
Optimisation du Processus d'Onload des passagers.

Travaux écrits réalisés

Rapports de stage

Apports

Formation à la fois théorique sur les savoirs d'un ingénieur généraliste (mathématiques, mécanique et informatique) mais également très professionnalisante par la grande part accordée aux stages en entreprise qui préparent à l'insertion dans la vie active.

1.4.2 Formation professionnelle et continue

Dans le cadre de mon poste actuel à RTE, j'ai suivi des formations internes (ERSO, AUTO et SSFDR) pour découvrir des domaines techniques en rapport avec la gestion d'un réseau électrique. J'ai pu acquérir des connaissances dans l'électrotechnique de réseau qui n'avaient pas été abordées lors de ma formation initiale. Ceci m'a permis d'asseoir ma compréhension des enjeux techniques dans mon poste en matière de stabilité du réseau. Plus particulièrement, j'ai mené une étude harmonique sous EMTP® [2], logiciel utilisé à RTE pour simuler les transitoires électromagnétiques, pour répondre à une demande client sur un projet d'interconnexion à courant continu (projet Aquind). Cela répondait à la fois à un besoin opérationnel et à un besoin de formation. J'ai pu me mettre à la place des utilisateurs des outils de simulation de transitoires électromagnétiques que je développe et ainsi mieux comprendre leurs besoins.

Par ailleurs, j'ai eu l'opportunité de participer à un programme d'innovation sur l'Agilité, en partenariat avec Transnet BW, un gestionnaire de réseau allemand basé à Stuttgart. Dans un premier temps, ce programme proposait aux participants des deux entreprises une découverte théorique des concepts autour de l'agilité : méthodes agiles, design thinking, design sprint, business modelling et lean startup. Puis, il fallait mettre en pratique une idée d'innovation sur plusieurs mois en utilisant la méthode de projet SCRUM. Cela m'a permis d'appréhender l'efficacité des méthodes agiles dans le cadre d'un projet concret transverse. Dans une équipe internationale, j'ai appris de passer d'une idée d'innovation à sa première maquette via le design sprint. Puis, j'ai participé à son développement via la méthodologie SCRUM vers une première version fonctionnelle. C'est un apport indéniable à la conduite d'un projet d'innovation dans une démarche de recherche et développement.

Intitulé	Durée	Dates - établissement	Résumé de la formation
Electrotechnique de réseau (ERSO)	3 jours	17/11-19/11 2015 Formation Interne	Maitriser les connaissances de base en électrotechnique de réseau.
Analyse et commande des systèmes dynamiques (AUTO)	4 jours	19/01-22/01 2016 Formation Interne	Connaître les notions de bases du fonctionnement dynamique des systèmes utilisés dans la gestion des réseaux électriques.
Sensibilisation à la Sûreté Système par le fonctionnement dynamique des réseaux (SSFDR)	3 jours	27/03-31/03 2017 Formation Interne	Connaître les interactions sur le réseau et comprendre l'intérêt des simulations dynamiques pour la sûreté du système.
ICIP2019	1 an	08/04/2019-29/04/2020 Die Organisations Gestalter	Apprendre et mettre en œuvre les méthodes agiles dans le cadre d'un projet d'innovation international.
Etude des lieux d'impédances harmoniques pour Aquind	3 mois	01/12/2019-01/03/2020 Formation Interne	Apprendre à faire une étude de lieux d'impédances harmoniques sous EMTP®.

N°1	ICIP2019
<p>Objectifs : Apprendre et mettre en œuvre les méthodes agiles dans le cadre d'un projet d'innovation commun aux deux entreprises (RTE et Transnet BW)</p> <p>Programme : Participants pour moitié RTE (8 personnes) et Transnet BW (7 personnes) 4 semaines de formation théorique en anglais Concept d'agilité et méthodes agiles (SCRUM), Design Thinking, Design Sprint, Lean startup et business modelling Mise en pratique sur un projet d'innovation : COBAMA (content-based matching) plateforme web de mise en relation entre salariés de gestionnaires de réseau sur des sujets d'intérêt commun. Projet SCRUM sur 7 sprints d'un mois. Une réunion physique par mois pour faire le bilan du sprint et lancer le suivant.</p> <p>Durée : 1 an</p> <p>Apports : Travailler dans un environnement international Mettre en œuvre une démarche d'innovation en mode agile : faire émerger une idée lors d'un design sprint et la développer en appliquant la méthode SCRUM.</p>	

N°2	ETUDE DES LIEUX D'IMPEDANCES HARMONIQUES POUR AQUIND
<p>Objectifs :</p> <p>Apprendre à faire une étude des lieux d'impédances harmoniques sous EMTP® dans le cadre d'une demande client suite à un projet d'insertion d'une liaison HVDC (Courant Continu à très Haute Tension) : projet Aquind entre la France et l'Angleterre.</p> <p>Programme :</p> <p>Modélisation du réseau autour du point de raccordement sous EMTP®. Définition des scénarios pertinents à simuler en lien avec le service exploitation de RTE. Simulations paramétriques via EMTP® et la suite ParametricStudio. Analyse des résultats et extraction des lieux d'impédances harmoniques qui seront envoyés au client (Aquind). Rédaction d'une note technique interne retraçant le dérouler de l'étude.</p> <p>Durée : 3 mois</p> <p>Apports :</p> <p>Comprendre les besoins des utilisateurs d'outils de simulation tels qu'EMTP® en réalisant une étude pratique. Allier à la fois une formation qui permet de découvrir la modélisation et la simulation harmonique sous EMTP® et une étude concrète qui répond à un besoin client (Aquind).</p>	

1.5 Mon parcours professionnel

1.5.1 Ingénieur Logiciel chez FACTSET (2013-2015)

FACTSET est une entreprise américaine spécialisée dans l'édition de logiciels et la fourniture d'informations financières. Ces principaux concurrents sont Bloomberg et Thomson Reuters. Employant près de 6000 employés en 2013, elle a des bureaux dans les principales places financières (Etats-Unis, Europe, Londres...) mais également en Inde et aux Philippines. J'ai travaillé dans la filiale française à Paris, FACSET JCF (effectif de 80 salariés), qui s'occupe de la constitution de deux bases de données financières : Factset Estimates et Factset Economics. La première regroupe l'ensemble des prévisions des analystes financiers sur les résultats des entreprises. La seconde est constituée de prévisions macro-économiques. Cela répondait à mes attentes de commencer ma carrière dans le développement informatique en capitalisant sur ma spécialité d'école d'ingénieurs. L'environnement financier de par sa complexité et ses modélisations mathématiques m'attirait. J'ai intégré l'équipe en charge du développement de l'outil de saisie des données financières pour alimenter les bases Factset Estimates et Factset Economics. Cet outil permettait de semi-automatiser la lecture de rapports financiers d'analystes et la saisie des données pertinentes dans les bases données. Outre la vitesse de saisie, le plus important restait la qualité de la donnée où des équipes dédiées étaient chargées de la vérifier.

Cette expérience a été très formatrice en m'apprenant à développer et à maintenir de façon industrielle une application en C++ avec une gestion de bases de données (SQL). Elle m'a permis également de mettre en pratique mes connaissances dans la gestion de projets informatiques. Les développements étaient organisés en mode projet de 3 mois. Je prenais part à la traduction

technique des nouveaux besoins et leur développement. Je corrigeais également les bugs venant des tests et de la production. J'ai travaillé dans un environnement international en interagissant avec les équipes en Inde qui utilisaient et testaient l'outil de saisie de données financières. Le tableau ci-dessous résume les compétences que j'ai acquises lors de cette expérience

Savoirs, connaissances	Savoir-faire technique, procédural	Savoir-faire relationnel
<p>Maitriser les bases de la programmation et de l'algorithmique.</p> <p>Maîtriser les bases de données.</p> <p>Connaître les différentes étapes d'un projet informatique.</p> <p>Connaître l'utilisation des données financières collectées.</p>	<p>Savoir coder dans un langage de programmation (C++), utiliser un langage de requête pour les bases de données (langage SQL).</p> <p>Comprendre le besoin des utilisateurs et le reformuler en fonctionnalités techniques.</p> <p>Investiguer et corriger les bugs détectés lors de la phase de test ou en production.</p> <p>Estimer le temps nécessaire pour le développement des nouvelles fonctionnalités.</p> <p>Tenir les délais d'un cycle de développement.</p>	<p>Adapter son niveau de langage à ses interlocuteurs.</p> <p>Prendre en compte les différences culturelles.</p>
<p>Les ressources externes</p> <ul style="list-style-type: none"> - l'équipe de développement. - recherche internet pour les bonnes pratiques de programmation et surtout la résolution de bugs. - les testeurs pour comprendre et reproduire les bugs. - les utilisateurs issus de l'équipe qualité des données pour comprendre les besoins qui ont été définis. 		

Après la montée en compétences sur les techniques de développement et sur les interactions à avoir lors du cycle du projet, ce poste me paraissait répétitif en étant ponctué tous les trois mois par les mêmes phases. Il ne me stimulait plus intellectuellement en étant exclusivement du développement informatique. Outre son aspect répétitif, la petite taille de la filiale française (80 personnes) n'offrait pas de perspectives d'évolution. J'ai décidé de rechercher une nouvelle opportunité professionnelle dans un grand groupe où je ne ferai pas que du développement informatique. En effet, cette première expérience était essentiellement du développement informatique et impliquait peu d'ouverture sur la finance et très peu de mathématiques appliquées. En 2015, j'ai donc ainsi intégré RTE en tant Ingénieur R&D.

1.5.2 Ingénieur R&D chez RTE (2015-)

RTE (Réseau de Transport d'Electricité) est un grand groupe public (près de 9000 employés) en charge de l'exploitation et de la maintenance du réseau de transport d'électricité français. On retrouve ces deux grandes fonctions dans son organisation (Figure 1-1) : un pôle gestion de l'infrastructure pour la maintenance, un pôle Clients – Conception et opération des Systèmes pour

l'exploitation du réseau mais également pour le système d'information, la R&D et l'Innovation. Les deux autres grands pôles concernent les finances, les achats et les ressources humaines (Transformation – Environnement Salarié).

Je travaille dans le Centre National d'Expertise Réseau (CNER), direction en charge de l'expertise technique sur les matériels installés sur le réseau au cours de leur cycle de vie. Cette direction est un appui naturel à la maintenance et se trouve ainsi dans le pôle gestion de l'infrastructure. Plus précisément, le département Postes (80 personnes) s'occupe de l'expertise sur les postes électriques. Près de 2500 postes électriques sont installés sur le réseau.

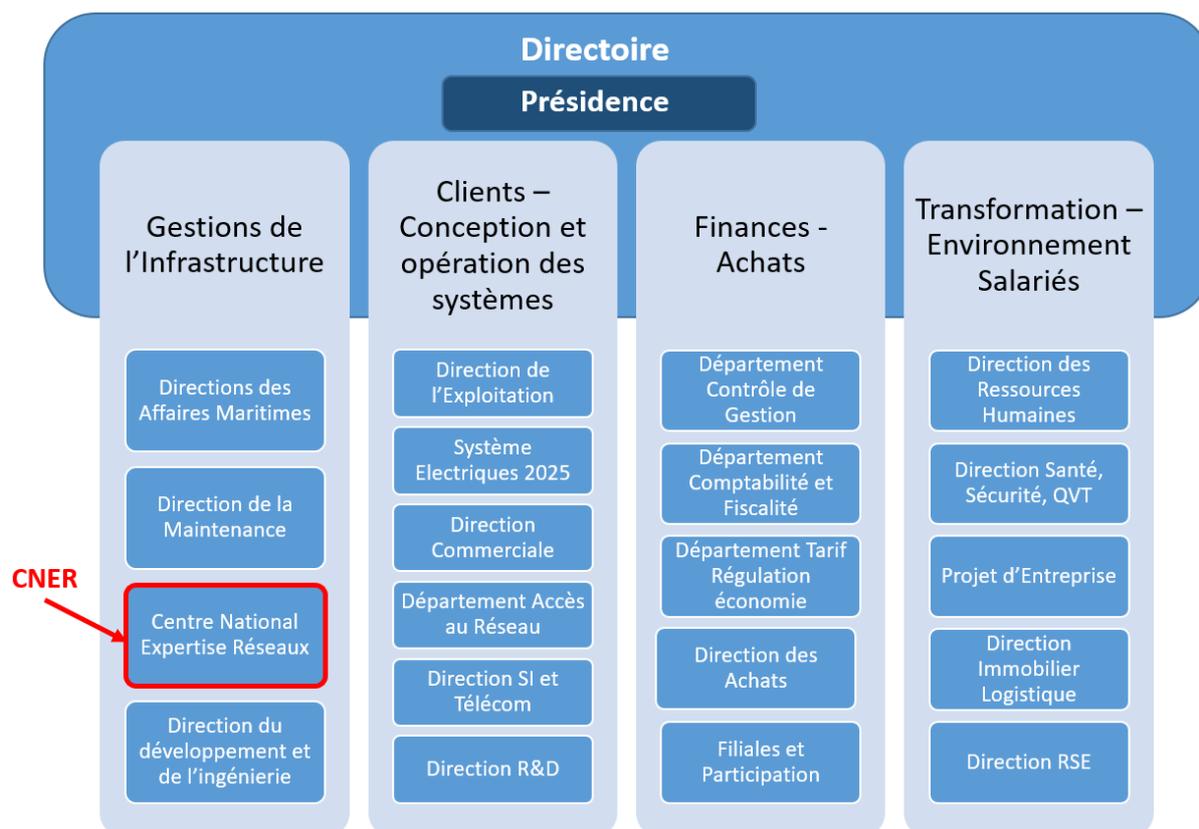


Figure 1-1 : Organigramme de RTE au 1^{er} février 2021.

1.5.2.1 Contexte et enjeux

Mon équipe (24 personnes), la division Etudes Electromagnétiques et Electronique de Puissance (D3EP) rattachée au département Postes, a en charge plus spécifiquement des postes à base d'électronique de puissance. Ce type d'équipements répond aux enjeux de la Transition Energétique. En effet, pour répondre à l'objectif d'une stratégie bas-carbone, la consommation d'une électricité décarbonée doit passer en 2050 de 25% à 55% de l'énergie consommée au total. Dans *Les Futurs Energétiques 2050* [3], RTE a noté, que quel que soit le mix énergétique choisi, il faudra augmenter considérablement la production d'énergies renouvelables. Pour ce faire, les technologies à base d'électronique de puissance sont incontournables sur le réseau pour assurer l'intégration massive des énergies renouvelables. Cela se traduit sur le réseau par une multiplication de projets d'interconnexion à courant continu entre la France et ses voisins [4]. Encouragé par l'Europe pour assurer la sécurité d'approvisionnement [5], cela permet aussi une évacuation de la production renouvelable et leur intégration au marché européen d'électricité. Comme le montre la Figure 1-2, de nombreux projets ont été mis en service par RTE ces dernières années : une liaison France-

Espagne INELFE en 2015, une liaison France-UK IFA2 (2021). Certains sont en cours de réalisation (liaison France-Italie prévue pour 2022) ou en projet (une autre liaison France-Espagne dans le Golfe de Gascogne et une avec l'Irlande, Celtic). De l'électronique de puissance est nécessaire pour assurer la conversion de la tension du continu vers l'alternatif, traditionnellement utilisé sur les réseaux électriques. La Figure 1-3 montre un aperçu de l'électronique de puissance présente dans un ce type de poste de conversion tension continue/alternative.

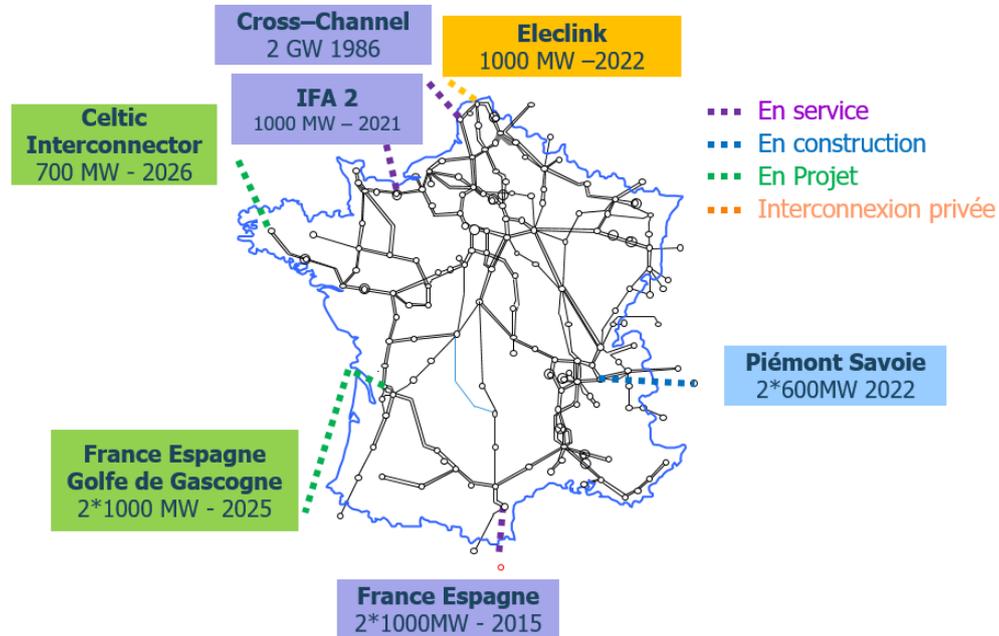


Figure 1-2 : Projet d'interconnexion à courant continu sur le réseau de transport d'électricité français (source RTE).



Figure 1-3 : Aperçu du hall d'un poste d'électronique de puissance pour la conversion continu/alternatif (liaison France-Espagne, source RTE).

Pour répondre à ces défis, mon équipe est constituée de quatre pôles de compétences (Figure 1-4) : Projets, Maintenance, Etudes et Codeurs. Les deux premiers concernent l'ensemble du cycle de vie d'un poste d'électronique de puissance, de sa spécification à sa mise en service, jusqu'à sa maintenance lors de son exploitation.

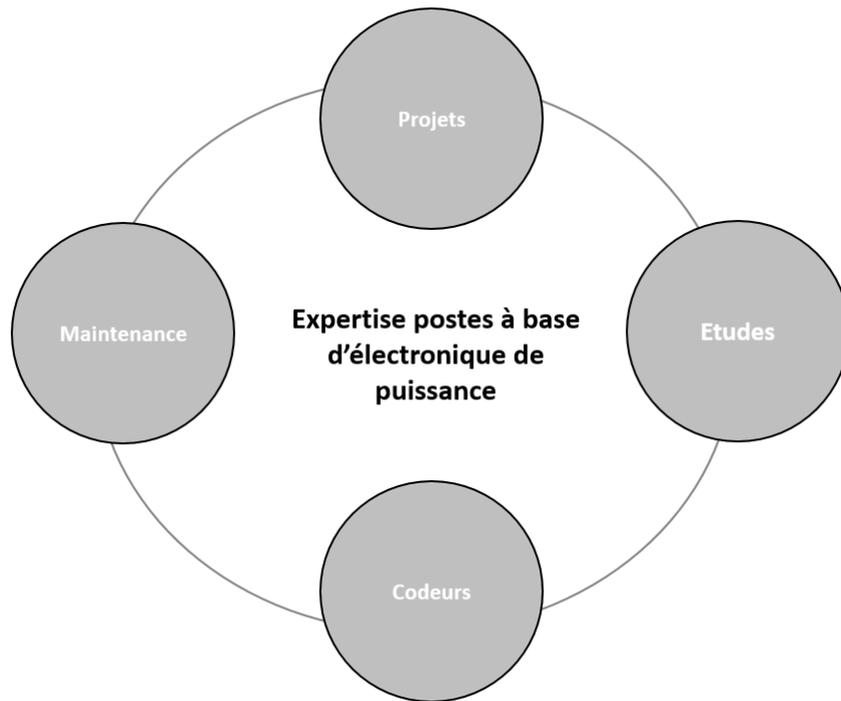


Figure 1-4 : Pôles de Compétences de mon équipe D3EP.

Les deux derniers, Etudes et Codeurs, permettent d'appréhender les transitoires électromagnétiques qui peuvent arriver sur le réseau lors de l'insertion de nouveaux équipements ou lors d'incidents. Le pôle Etudes reproduit ces phénomènes en utilisant deux outils de simulation, HYPERSIM pour la simulation temps réel et EMTP® pour le temps différé. La simulation temps réel implique une simulation numérique qui interagit directement avec du matériel. En effet, si les modèles numériques de réseau sont assez précis, ce n'est pas le cas pour les modèles de contrôle-commande de stations de conversion. Ils ne permettent pas pour le moment d'avoir la même précision que sur site. Pour ce faire, un laboratoire de simulation temps réel a été mis en place sur le Campus Transfo de RTE à Jonage dans la région de Lyon. Pour chaque projet d'électronique de puissance, un deuxième contrôle-commande identique à celui installé sur site (une réplique) est acheté pour être connecté à simulateur au laboratoire et servir in fine à de la simulation temps réel. La Figure 1-5 illustre une réplique connectée à un simulateur au sein du Campus Transfo.

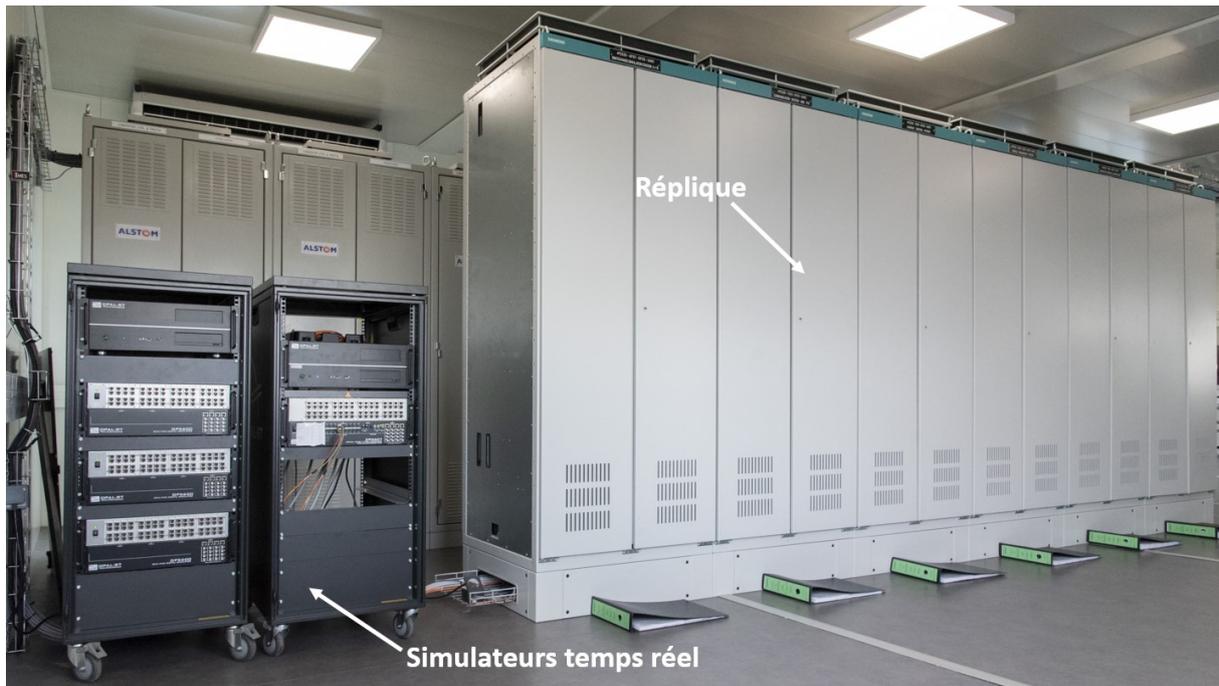


Figure 1-5 : Installation de simulation temps réel (Répliques + Simulateur) au Campus Transfo de RTE.

Je travaille dans le pôle des Codeurs, qui a en charge le développement des outils de simulation de transitoires électromagnétiques qu'utilise le pôle Etudes (HYPERSIM et EMTP®). Le développement et la commercialisation de ces outils se fait via des partenariats externes. Ainsi, EMTP® est en co-développement avec EDF, Hydro-Québec et PGSTech (Montréal). La commercialisation est faite par Powersys et PGSTech. Le travail sur HYPERSIM se fait également dans le cadre d'un accord de co-développement avec Hydro-Québec. La société Opal-RT (Montréal) développe aussi et commercialise cet outil. Par ailleurs, le développement de ces outils est alimenté par les travaux de recherche de la chaire industrielle financé en partie par RTE de l'École Polytechnique de Montréal sur la *Simulation multi-échelle de temps des transitoires dans les réseaux électriques de grandes dimensions*. Ils permettent d'enrichir les méthodes de résolution ainsi que les modèles utilisés avec les dernières avancées de la recherche dans le domaine.

Mon poste concerne initialement le développement de nouvelles fonctionnalités dans HYPERSIM. RTE a ainsi bien capitalisé sur ma précédente expérience d'ingénieur développement logiciel pour la prise en main du développement d'HYPERSIM. Mon profil d'ingénieur généraliste était également recherché pour être en mesure de bien comprendre le domaine d'application des transitoires électromagnétiques via des connaissances scientifiques. Pour ma part, RTE, en étant un grand groupe, m'offrirait davantage d'évolution de carrière que ma précédente expérience. Le poste répondait à mes appétences en alliant à la fois une composante informatique et une composante mathématiques appliquées sur la résolution des transitoires électromagnétiques. Trois grandes étapes ont ponctué ma montée en compétences sur cette expérience professionnelle :

- Mes débuts (2015-2016)
- Maîtrise et Encadrement (2017-2018)
- Expertise et Recherche (2019-)

Ces différentes étapes ont progressivement transformé un travail d'ingénierie en un travail de recherche. Comme le montre la Figure 1-6, mes compétences sur les outils de simulation se sont orientées de l'informatique (Interfaçage externe...) vers des activités de recherche, plus cœur de métier, comme le moteur de calcul.

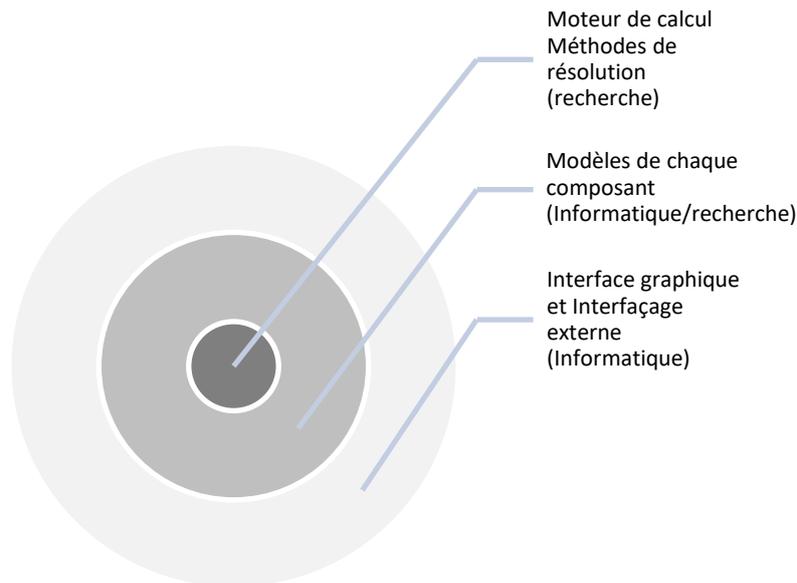


Figure 1-6 : Activités sur les outils de simulation, de celles orientées Informatique (cercle périphérique) à celles liées plus à la recherche (cercle intérieur).

1.5.2.2 Mes Débuts (2015-2016)

Chez RTE, j'ai changé de domaine d'application en passant de l'information financière à la simulation de transitoires électromagnétiques, de l'informatique de gestion au calcul scientifique. J'ai dû apprendre ce nouveau domaine. Ainsi, des formations en interne m'ont permis d'acquérir les bases sur l'électrotechnique de réseau et sur la stabilité d'un système électrique. J'ai dû également comprendre les techniques de résolution utilisées par les outils de simulation (EMTP® et HYPERSIM) ainsi que la physique des modèles. Concernant le développement informatique d'HYPERSIM, j'ai pu bénéficier des acquis de ma précédente expérience en tant ingénieur logiciel. Durant ces premières années, j'ai mis en place deux fonctionnalités, une interface HYPERSIM-SIMULINK et un modèle d'équations d'état.

La première fonctionnalité était un interfaçage externe avec le logiciel MATLAB/SIMULINK [6], le plus utilisé dans la modélisation de contrôles. C'était entièrement du développement informatique. Peu de connaissances étaient requises sur le fonctionnement de l'outil HYPERSIM en général. La difficulté résidait en ce que cet interfaçage n'avait pas été mis à jour depuis 10 ans et qu'il n'était pas robuste à la sortie de nouvelles versions de SIMULINK. Le rendre compatible avec une nouvelle version prenait du temps. Une refonte des techniques utilisées a permis de simplifier le support des nouvelles versions de SIMULINK qui est maintenant plus rapide et robuste. C'est une des fonctionnalités les plus utilisées que j'ai développées dans HYPERSIM. La plupart des utilisateurs pour la modélisation du contrôle passe par cet import. La deuxième fonctionnalité m'a permis de découvrir la physique des modèles en implémentant un modèle d'équations d'état. Une aide précieuse des développeurs d'HYPERSIM m'a permis de comprendre comment interfacier un modèle de composant de réseau avec le moteur de calcul temps réel. Ce fut une première approche pour comprendre les interactions entre les modèles et le moteur de calcul qui contient les méthodes de résolution.

Bien sûr, j'ai dû me familiariser avec les partenaires externes du projet de développement : Hydro-Québec, Opal-RT (société qui commercialise HYPERSIM). Un séjour d'une semaine en 2016 à Montréal m'a permis de collaborer plus étroitement avec ces acteurs sur les fonctionnalités que j'ai

développées. J'ai découvert également les travaux de la Chaire de Polytechnique Montréal dont les travaux sont intégrés par la suite dans les outils de simulation (EMTP® et HYPERSIM). Conquis et enchanté par ce poste qui répondait à mes attentes, du développement informatique avec des équations physiques, j'avais passé le cap de la découverte.

1.5.2.3 *Maîtrise et Encadrement (2017-2018)*

Maîtrisant l'environnement technique de mon poste, j'ai été en mesure de prendre du recul et de valoriser mes contributions, pas uniquement en tant que fonctionnalités d'HYPERSIM mais aussi en tant que contributions scientifiques. J'ai pu également m'intéresser aux techniques de parallélisation et rentrer dans la compréhension du moteur de calcul.

Mes travaux sur l'interface HYPERSIM-Simulink et le modèle d'équations d'état ont fait l'objet d'une communication [7] que j'ai présenté à la conférence IPST (Conférence Internationale sur les Transitoires Electromagnétiques) en 2017 à Séoul. C'est une conférence du domaine de la simulation des transitoires électromagnétiques des réseaux électriques qui a lieu tous les deux ans. Ce fut ma première expérience de publication d'un article scientifique dans une conférence internationale. J'ai dû me plier aux exigences de rédaction (Structuration, bibliographie complète, cas de test pertinents) et de sélection (soumission d'un résumé et rédaction d'un manuscrit final, révisions) d'un papier. J'ai également commencé à travailler sur la parallélisation d'HYPERSIM pour accélérer la simulation.

Au départ centré sur le répartiteur de tâches, mes travaux étaient principalement une activité d'ingénierie. Il fallait donner davantage de flexibilité aux utilisateurs via l'implémentation d'une répartition manuelle. Puis, l'aspect R&D est venu dans une étude des techniques automatiques de répartition des tâches. Cette étude a permis de simuler en temps réel pour la première fois le modèle CIGRE de Courant Continu. Ce travail a fait l'objet d'un article [8] dans lequel j'ai collaboré. Il a été accepté à la conférence généraliste PSCC sur les Calculs dans les systèmes électriques en 2018. Je l'ai présenté lors de la tenue de cette conférence à Dublin. Il a fait également l'objet d'une présentation à la conférence RT2018 sur la simulation temps réel organisée par Opal-RT à Paris.

J'ai également appris à diffuser ma culture scientifique à différents publics. Ainsi, j'ai vulgarisé dans la revue technique interne de RTE [9] l'algorithme de répartition des tâches d'HYPERSIM. J'ai été conférencier showroom pendant 2 ans, où je devais présenter des sujets de R&D de RTE, qui n'étaient pas forcément rattachés à mon activité, face à un public interne ou externe. J'ai dû adapter mon discours suivant le public et faire preuve de pédagogie pour bien vulgariser les sujets de R&D et en faire la promotion.

La continuation de mes travaux sur la répartition des tâches et la maîtrise de mon poste m'ont permis d'entreprendre une démarche de recherche et développement en proposant un stage niveau M2 recherche en 2018. J'ai participé à toute les phases du stage : de la conception du sujet, au recrutement des candidats et à l'encadrement au cours du stage. J'ai pris goût à l'encadrement et à la transmission de mon expertise. Le sujet concernait la recherche d'un algorithme efficient de répartition des tâches. Des collaborations fructueuses ont permis de faire avancer ce sujet. Hydro-Québec a partagé son expertise sur les techniques de répartition des tâches ainsi que sur les architectures de simulation. J'ai pris contact avec les experts d'optimisation du département Recherche et Développement de RTE pour avoir leur éclairage sur le problème de répartition des tâches. Ils ont fourni des méthodes et des outils (solveurs d'optimisation, supercalculateur) pour

vérifier l'optimalité de la répartition obtenue. Par ailleurs, le projet R&D européen Best Paths, piloté dans mon équipe sur l'interopérabilité des stations de conversion, a apporté un cas pratique innovant de simulation temps réel de réseaux à courant continu. J'ai ainsi appris que l'on ne doit jamais travailler seul en recherche mais qu'au contraire les collaborations sont une source d'enrichissement dans un croisement de plusieurs expertises : simulation en parallèle d'une résolution de réseau, optimisation mathématique et modélisation d'un réseau à courant continu.

1.5.2.4 Expertise et Recherche (2019-)

En 2019, un projet de renouvellement des outils de simulation par un outil unique qui ferait à la fois le temps réel (HYPERSIM) et le temps différé (EMTP[®]) a été lancé. Cela a été un tournant dans mon poste en lui donnant une dimension recherche et développement. L'objectif n'était plus de développer une fonctionnalité pour un outil spécifique mais d'identifier et de prototyper de nouvelles méthodes de résolution. J'ai donc entrepris une démarche R&D dans l'identification et l'implémentation de ces méthodes.

Dans ce cadre, les précédents travaux du stage de M2 recherche sur la répartition des tâches ont été enrichis et valorisés par un article de conférence [10] présenté à l'IPST en 2019 à Perpignan organisé par RTE. J'ai également pris part à l'organisation de cette conférence. J'étais en charge de la prospection de partenariats institutionnels, de l'accueil des participants et de la bonne tenue des sessions. Au vu de la qualité du travail présenté lors de la conférence, le papier sur la répartition des tâches a été sélectionné pour être soumis au journal *EPSR (Electric Power Systems Research)*. J'ai dû ainsi faire l'apprentissage des standards d'un journal : montrer clairement sa contribution vis-à-vis de la littérature existante, apporter des réponses détaillées aux exigences des arbitres. Pour ce faire, le travail initial a dû être augmenté d'une étude détaillée sur différents algorithmes de répartition des tâches. Cet article [11] a été publié dans la revue *EPSR* en 2020.

Avant tout, une collaboration plus étroite avec la chaire de recherche de Polytechnique Montréal et Hydro-Québec m'a permis d'entreprendre deux sujets de recherche d'envergure. Le premier porte sur la méthode de compensation qui permet une parallélisation sans délai d'une solution de réseau. C'est le plus ambitieux que j'ai fait. Dans les années 70s, cette méthode avait été tout d'abord théorisée pour les fonctions non-linéaires puis pour la parallélisation. Cependant, aucune implémentation industrielle n'avait été proposée pour tester cette méthode dans le cadre de la parallélisation. Elle était restée principalement dans le cadre universitaire. La force de cette méthode de parallélisation générique est de pouvoir séparer n'importe où un réseau en tâches en vue d'une simulation parallèle. Ceci est très utile pour les réseaux d'électronique de puissance où l'on ne peut pas utiliser la méthode classique des délais de propagation des lignes de transmission. Une collaboration avec la chaire (professeur Jean Mahseredjian) sur ce sujet a permis de dégager les fondements théoriques de la méthode. Puis, grâce à mon expertise acquise sur les développements des outils de simulation, j'ai implémenté un prototype en temps réel en intégrant cette méthode de parallélisation générique au cœur de calcul d'HYPERSIM. L'environnement temps réel d'HYPERSIM m'a permis d'avoir accès à de nombreux cas d'applications industrielles qui faisaient défaut dans la littérature existante. Ainsi, les performances de la méthode de Compensation ont pu être validées avec les répliques de contrôle-commande de station de conversion dans le laboratoire temps réel de RTE. Ce travail a fait l'objet d'une présentation en ligne à la conférence IPST en 2021. L'article associé [12] a été publié dans la revue *EPSR*. Un autre papier [13], sur le traitement des réseaux non-linéaires avec cette méthode a été accepté dans la revue scientifique *IEEE Transactions on Power Delivery*.

Une contribution à ce sujet [14] a été présentée lors de la Session CIGRE de Paris en 2022. De plus, la méthode de compensation a permis de mettre en place une plateforme de simulation temps réel pour étudier les éventuelles interactions entre deux ouvrages d'électronique de puissance [15]. Cela a fait l'objet d'une communication lors de la conférence IECON 2022.

Le deuxième sujet concerne une formulation en Jacobienne réduite pour résoudre les boucles de contrôle, toujours en collaboration avec la chaire. Ce sujet faisait suite aux travaux d'une thèse de 2013 à Polytechnique Montréal. La méthode avait été introduite théoriquement et prototypée dans EMTP®. J'ai apporté deux optimisations de la méthode : l'une sur une formulation plus générique et simplifiée et l'autre sur une minimisation du nombre de variables à considérer. Là encore, une connexion à d'autres domaines de recherche a été très utile. Les techniques de minimisation du nombre de cycles d'un graphe utilisées pour le Web ont permis d'optimiser cette méthode de résolution des boucles de contrôle. Ces améliorations ont été implémentées dans EMTP®. Un article [16] a été soumis dans un journal du domaine.

Toujours dans l'objectif de diffusion de mes travaux scientifiques, j'ai continué à vulgariser en interne mes travaux dans la revue technique RTE Tech sur la mise en place d'une plateforme de simulation distribuée [17] et sur la méthode de Compensation [18]. Un travail de veille a également été entrepris en devenant arbitre pour deux journaux du domaine, *EPSR* et *IEEE Transactions on Power Delivery*. Ma contribution a été pour l'un d'entre eux distinguée en tant qu'Outstanding reviewer 2020 (voir certificats en annexe). En parallèle, j'ai continué l'encadrement d'un stagiaire de fin d'études en 2019 et d'un alternant peu après sur trois ans, en étant toujours motivé pour partager mon expertise.

1.5.2.5 Compétences acquises

L'ensemble des compétences que j'ai acquises dans mon poste actuel est résumé dans le tableau ci-dessous.

Savoirs, connaissances	Savoir-faire technique, procédural	Savoir-faire relationnel
<p>Maitriser les bases de la programmation et de l'algorithmique.</p> <p>Connaître les bases de l'électrotechnique de réseau.</p> <p>Maitriser les méthodes mathématiques de résolution d'équations différentielles pour la simulation de transitoires électromagnétiques.</p> <p>Comprendre la physique des modèles.</p>	<p>Savoir coder dans des langages de programmation (C++, Java, Python, Fortran).</p> <p>Savoir implémenter et tester un nouveau modèle.</p> <p>Investiguer et corriger les bugs détectés par les utilisateurs sur les fonctionnalités développés.</p> <p>Mettre en place une démarche de R&D dans le prototypage et le test de nouvelles méthodes de résolution dans les outils de simulation.</p> <p>Mener une veille technologique via des conférences ou la revue</p>	<p>Adapter son niveau de langage à ses interlocuteurs.</p> <p>Faire preuve de pédagogie pour transmettre et promouvoir des sujets techniques.</p> <p>Travailler dans un environnement international.</p> <p>Savoir collaborer sur une thématique de recherche.</p>

	<p>de papiers scientifiques.</p> <p>Valoriser ses travaux de R&D via des publications scientifiques et un transfert des méthodes dans les outils.</p> <p>Diffuser sa culture scientifique vers l'interne et l'externe.</p> <p>Encadrer des stagiaires/alternants sur des sujets techniques et de recherche.</p>	
<p>Les ressources externes</p> <ul style="list-style-type: none"> - Membres de l'équipe (utilisateurs des outils de simulation et codeurs des outils). - Equipe de développement d'HYPERSIM chez Hydro-Québec et Opal-RT. - Equipe de développement d'EMTP® chez PSGTech et Hydro-Québec. - Chaire de Polytechnique de Montréal pour le suivi et le transfert des travaux dans les outils de développement. - Recherche internet pour les bonnes pratiques de programmation et surtout la résolution de bugs. - IEEE explore pour l'accès aux articles scientifiques (veille et review de communications scientifiques). 		

1.6 Conclusion générale sur les expériences de mon parcours

Mon parcours professionnel s'est tout d'abord focalisé sur le développement de compétences dans le développement informatique. Ces compétences initiées dans mon premier poste chez FACTSET ont largement bénéficié à mon second poste chez RTE. Dans mon poste actuel, mon profil s'est enrichi en s'orientant vers le domaine de la recherche et développement dans la simulation de transitoires électromagnétiques. Ma formation initiale scientifique d'ingénieur généraliste m'a fortement aidé pour bien appréhender des nouveaux domaines d'application que sont l'électrotechnique de réseau et les méthodes de résolution pour la simulation de transitoires électromagnétiques. Cela correspondait davantage à mon goût pour les mathématiques appliquées que l'était un poste purement informatique. Cependant, mon expertise dans l'informatique constitue un avantage indéniable dans un domaine où très peu de personnes sont en mesure de comprendre à la fois les équations de la physique et de les implémenter directement dans les outils de simulation.

Cette alliance d'expertises en informatique et en mathématiques appliquées m'a permis de réaliser une démarche R&D sur de nouvelles techniques de simulation pour les transitoires électromagnétiques. J'ai pu ainsi acquérir les compétences que l'on attend d'un futur docteur. Ma recherche a été valorisée par des publications scientifiques de conférences et de revues. Je l'ai diffusée en interne dans des articles vulgarisés et à l'externe lors de présentations dans des conférences internationales. Je tiens à la mettre à jour des nouvelles avancées du domaine comme témoigne mon travail d'arbitre. J'apprécie de partager ma recherche et de la transmettre dans le cadre d'encadrements de stagiaires.

Chapitre 2 Introduction Générale aux Contributions

2.1 Contexte et motivations

Atteindre la neutralité carbone en 2050, c'est l'objectif que s'est fixé la France pour juguler le changement climatique. Cet objectif ambitieux [19] fait suite à l'accord de Paris pour le climat lors de la 21^{ème} Conférence des Parties (COP21) en 2015 pour limiter le réchauffement climatique à 1,5°C. La réussite de cette transition énergétique passe par un virage vers les énergies bas-carbone, comme le recommande la Stratégie Nationale Bas-Carbone (SNBC). L'électricité avec des moyens de production décarbonés en fait partie. Sa consommation est amenée à fortement augmenter, passant de 25% de la consommation totale d'énergie à 55% en 2050. Quel que soit le mix énergétique étudié, cette augmentation de la consommation d'électricité représente un grand défi pour les réseaux électriques [3] afin d'accueillir les nouveaux moyens de production d'énergies renouvelables dans le mix énergétique. Leur insertion nécessite le développement d'un réseau à courant continu de plus en plus prépondérant, reposant sur de l'électronique de puissance.

Cela traduit par un développement des interconnexions à courant continu de très haute tension (HVDC) entre pays européens [5]. En interconnectant les réseaux européens, l'objectif est de davantage sécurisé l'approvisionnement en électricité de chaque pays. Cela permet ainsi d'évacuer le surplus de production renouvelable sur le marché européen. De nombreux projets HVDC ont été mis en service sur le réseau électrique français [4]: IFA2000 (1986), INELFE (2015), IFA2 (2021) ou sont en cours : Golfe de Gascogne, Celtic. Le développement de parcs éoliens en mer est également en plein essor. 10 GW de production sont prévus d'être raccordés d'ici 2030 en France. Des raccordements en HVDC [20] sont mis en place pour intégrer de fortes puissances.

Les technologies à base d'électronique de puissance nécessitent des études de comportement [21] tout le long de leur cycle de vie sur le réseau, des études de raccordement, d'interactions aux analyses d'incident. Les constantes de temps des phénomènes étudiés étant très rapides, de l'ordre de la micro seconde (μ s), une modélisation des transitoires électromagnétiques doit être réalisée. Les outils de type EMT (transitoires électromagnétiques) tels que EMTP[®] [2] pour le temps différé ou HYPERSIM [1] pour le temps réel permettent de simuler numériquement le réseau en se basant sur cette modélisation avec des pas d'intégration numérique de l'ordre de quelques μ s. Les besoins d'études sont à la fois en temps différé mais également en temps réel.

En effet, le temps différé permet d'avoir une meilleure précision de simulation des phénomènes. A contrario, le temps réel est plus rapide pour respecter la contrainte temporelle de synchronisation avec le pas de calcul de l'horloge réel (1 s est simulée en 1 s). La simulation en temps réel doit appliquer des approximations. Cela permet de communiquer avec du matériel (généralement des systèmes de contrôle-commande) dans une simulation dite HIL (Hardware-In-The-Loop). La nécessité de simulations temps réel a été démontrée [22] dans l'impossibilité des modèles numériques de contrôle actuels de reproduire avec précision le comportement du matériel réel. Cela est également utile lors de la phase projet et d'exploitation d'équipements d'électronique de puissance [22][23]

telles que les stations de conversion HVDC. Cependant, la rapidité du temps réel se fait au détriment de la précision des modèles numériques.

Ainsi, les défis actuels de l'étude des phénomènes transitoires liés au développement de l'électronique de puissance sont d'associer les bénéfices des deux environnements de simulation :

- **La précision des modèles et des méthodes de résolution** pour l'électronique de puissance: faire de la simulation en temps réel avec la précision de la simulation en temps différé.
- **L'amélioration de la vitesse de calcul.** En effet, ces technologies d'électronique de puissance nécessitent des modèles de plus en plus détaillés afin d'avoir la précision escomptée dans l'étude des phénomènes transitoires. Cela passe par une diminution du pas de temps et donc par un allongement du temps de simulation. Ainsi, une simulation plus rapide devient un enjeu à la fois pour le temps réel, pour tenir la contrainte temporelle, mais aussi pour le temps différé pour raccourcir le temps de simulation.

Les travaux qui constituent cette thèse en VAE ont contribué à ces deux points en mettant principalement l'accent sur l'accélération du temps de simulation.

2.2 Résumé des travaux

I – Equivalents Fréquentiels optimisés pour la simulation EMT (Chapitre 3)

Le premier axe de recherche concerne **l'utilisation d'équivalents de réseau fréquentiel comme le FDNE pour le temps réel**. Outre la portabilité de réseaux, cela permet surtout d'accélérer la simulation tout en gardant une bonne précision [24]. Le FDNE donne une représentation linéaire et fréquentielle de l'impédance d'un sous-réseau. La méthode de Vector Fitting (VF) [25][26] permet à partir d'un scan fréquentiel de l'impédance de réseaux de proposer un équivalent de réseau sous la forme d'équations d'état. Par la suite, ce modèle d'équations d'état peut être directement intégré à la modélisation de réseaux [27]. L'intégration d'un modèle d'équations d'état à la réalisation nodale fait intervenir des matrices creuses. Une représentation matricielle adéquate et l'utilisation de la génération de code permettent une optimisation des calculs pour le temps réel.

II – Méthodes d'accélération des résolutions de réseaux (Chapitre 4 et Chapitre 5)

Le deuxième axe de recherche porte sur **l'accélération des méthodes de résolution de réseaux** tout en assurant un haut niveau de précision. Pour y parvenir, **les techniques de parallélisation d'une simulation EMT** sont étudiées en tirant partie de multi-cœurs des plateformes de simulation actuelles (Ordinateur avec le système d'exploitation Windows pour le temps différé, simulateur avec le système d'exploitation Linux pour le temps réel). On distingue deux types de parallélisation : celle de la résolution de réseaux via une formulation nodale des équations et celle des modèles dans le calcul des valeurs historiques de chaque modèle qui découlent de la méthode d'intégration numérique. Ces deux éléments peuvent être regroupés dans la notion de tâche de réseau : un sous-réseau où sont effectuées la résolution nodale et la mise à jour des historiques des composants le constituant. **La parallélisation d'une simulation EMT se fait en deux temps** : le réseau est d'abord **séparé en tâches**, puis une **répartition de ces tâches** est effectuée sur les processeurs disponibles pour la simulation. **Ces deux solutions sont étudiées.**

1^{er} Solution : Séparation en tâches (Chapitre 4)

Traditionnellement cette séparation est faite via les lignes de transmission lorsque le délai de propagation est supérieur au pas de temps. Néanmoins, dans certains cas comme pour les réseaux de distribution ou les montages à base d'électronique de puissance contenant un grand nombre de composants, ce découplage n'est plus possible. D'autres techniques existent comme **la méthode de compensation** [28] et son extension via une **formulation hybride** [29]-[33] en permettant de séparer en tâches n'importe où sans l'ajout de délai artificiel. Ces méthodes restent une interprétation physique d'autres techniques purement mathématiques dans la parallélisation de résolution de systèmes linéaires qui mettent la matrice d'admittance sous une forme diagonale par blocs frontières [34]. La **méthode de compensation** y est plus précisément présentée et a fait l'objet d'une implémentation dans un environnement temps réel. De nombreux exemples tirés de cas concrets de réseaux linéaires et non-linéaires incluant de l'électronique de puissance permettent de valider ses performances.

2^{eme} Solution : Répartition des tâches sur les processeurs (Chapitre 5)

Ce problème est NP-complet [35] et nécessite l'utilisation d'heuristiques pour optimiser à la fois l'équilibrage de la charge des processeurs et le coût des communications. Une **première heuristique basée sur un algorithme A*** avait été utilisée pour le temps réel [36]. Mais récemment **les techniques de partitionnement de graphe** [37][38] se sont révélées plus efficaces dans leur rapidité et leur optimisation combinée de l'équilibrage de la charge des processeurs et de la communication inter-processeurs.

III – Méthodes d'accélération des résolutions de systèmes de contrôle (Chapitre 6)

Outre la parallélisation de la résolution du réseau, **l'accélération du moteur de calcul du contrôle** est abordée comme sa précision pour les systèmes non-linéaires complexes. Via une modélisation input-output par bloc [39], la résolution d'un système de contrôle sans boucle de rétroaction est immédiate en ordonnant topologiquement les blocs à exécuter. Mais dans le cas de boucles de rétroaction, une formulation Jacobienne est nécessaire [40] pour résoudre avec une bonne précision les boucles algébriques et les non linéarités sans ajout de délai artificiel. **Une méthode de Jacobienne réduite** [41][42] a été introduite pour réduire la taille du système linéaire à résoudre et mettre à jour les autres variables via des séquences de boucles. Cette méthode y est optimisée dans la détection et la suppression de boucles [43] mais aussi dans une description plus générique de l'algorithme de résolution.

2.3 Contributions

Les contributions de cette thèse sont détaillées ci-dessous. Les implémentations ont été faites en lien avec des outils EMT préexistants (EMTP[®] pour le temps différé et HYPERSIM pour le temps réel) afin de tester les méthodes proposées dans un contexte industriel et réaliste.

- Implémentation d'un modèle d'équations d'état pour intégrer l'utilisation d'un équivalent fréquentiel (FDNE) à la simulation temps réel. Utilisation de la génération de code pour accélérer le calcul des matrices creuses. (Chapitre 3)

- Application du modèle d'équations d'état pour simuler en temps réel l'équivalent fréquentiel d'un câble d'une liaison HVDC. (Chapitre 3)
- Implémentation de la méthode de Compensation pour la simulation temps réel pour les réseaux linéaires et non-linéaires. (Chapitre 4)
- Etude des performances de la méthode de Compensation sur des cas industriels HVDC de simulation HIL linéaires et non-linéaires. (Chapitre 4)
- Etude comparative des heuristiques pour résoudre le problème de répartition des tâches sur de grands réseaux. (Chapitre 5)
- Etude de l'optimalité du partitionnement de graphe multi-niveaux dans son paramétrage et une comparaison avec les méthodes exactes. (Chapitre 5)
- Validation des performances des heuristiques sur des cas HVDC de simulation temps réel sur le réseau CIGRE et le projet Best Paths. (Chapitre 5)
- Optimisation de la méthode de Jacobienne réduite pour la résolution simultanée des boucles de rétroaction des équations de contrôle. Minimisation du nombre de variables de coupe à considérer en appliquant des heuristiques qui résolvent le problème du Feedback Arc Set minimal. (Chapitre 6)
- Formulation générique et simplifiée de l'obtention des séquences de boucle pour la Jacobienne réduite. (Chapitre 6)
- Implémentation de la Jacobienne réduite optimisée dans un outil générique de simulation EMT en temps différé. (Chapitre 6)
- Validation et test des performances de la Jacobienne réduite optimisée sur de grands réseaux. (Chapitre 6)

2.3.1 Liste des publications

2.3.1.1 Articles de Journal

- **B. Bruned**, J. Mahseredjian, S. Dennetière, J. Michel, M. Schudel, N. Bracikowski, "Compensation Method for parallel and iterative real-time simulation of electromagnetic transients," accepted to **IEEE Transactions on Power Delivery**, 2023. [doi:10.1109/TPWRD.2023.3238422](https://doi.org/10.1109/TPWRD.2023.3238422).
- **B. Bruned**, S. Dennetière, J. Michel, M. Schudel, J. Mahseredjian, N. Bracikowski, "Compensation Method for parallel real-time EMT studies," **Electric Power Systems Research**, Volume 198, 2021. doi.org/10.1016/j.epsr.2021.107341.
- **B. Bruned**, I. M. Martins, P. Rault, S. Dennetière, "Use of Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," **Electric Power Systems Research**, Volume 189, 2020. [doi:10.1016/j.epsr.2020.106604](https://doi.org/10.1016/j.epsr.2020.106604).

2.3.1.2 *Articles de Conférence*

- **B. Bruned**, S. Dennetière, Y. Vernay, V. Oiring De Castro Cesar, H. Saad, "Optimized real-time setup for Interaction study between VSC-HVDC and SVC on the French Network," **Proceedings of the IEEE Industrial Electronics Society Conference (IECON)**, Bruxelles, Belgique, 2022. [doi:10.1109/IECON49645.2022.9969119](https://doi.org/10.1109/IECON49645.2022.9969119).
- **B. Bruned**, "Compensation Method to speedup offline and real-time Electromagnetic Transients simulation of inverter-based transmission networks," **Contributions of CIGRE Session**, Paris, France, 2022.
- **B. Bruned**, I. M. Martins, P. Rault, S. Dennetière, "Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," **Proceedings of the International Conference on Power Systems Transients 2019 (IPST 2019)**, Perpignan, France, 2019. https://www.ipstconf.org/papers/Proc_IPST2019/19IPST035.pdf
- S. Dennetière, **B. Bruned**, H. Saad, E. Lemieux, "Task separation for real-time simulation of the CIGRE DC grid benchmark," **Proceedings of the Power Systems Computation Conference 2018 (PSCC 2018)**, Dublin, Ireland, 2018. [doi:10.23919/PSCC.2018.8442556](https://doi.org/10.23919/PSCC.2018.8442556).
- **B. Bruned**, C. Martin, S. Dennetière, Y. Vernay, "Implementation of a unified modelling between EMT tools for Network Studies," **Proceedings of the International Conference on Power Systems Transients 2017 (IPST 2017)**, Seoul, Republic of Korea, 2017. https://www.ipstconf.org/papers/Proc_IPST2017/17IPST128.pdf

2.3.1.3 *Rapports Internes*

- **B. Bruned**, S. Dennetière, "Méthode de Compensation pour paralléliser la simulation temps réel de réseaux électriques," **RTE Tech**, N°27, April, 2021.
- **B. Bruned**, S. Dennetière, "Simulation temps réel de grands réseaux électriques : solutions de parallélisation," **RTE Tech**, N°16, Janvier, 2018.

2.3.1.4 *Prépublications de journal*

- **B. Bruned**, J. Mahseredjian, S. Dennetière and N. Bracikowski, "Optimized reduced Jacobian formulation for simultaneous solution of control systems in Electromagnetic Transients simulations," to be revised in **IEEE Transactions on Power Delivery**, 2023.

Chapitre 3 Equivalents fréquentiels de réseaux

3.1 Introduction

La simulation de transitoires électromagnétiques est devenue indispensable pour un gestionnaire de réseau tel que RTE, afin de faire face à la transition énergétique [21]. En effet, l'arrivée massive des énergies renouvelables provoque une augmentation importante du nombre de dispositifs d'électronique de puissance sur le réseau, tels que les liaisons HVDC d'interconnexion ou les Compensateurs Statiques de Puissance Réactive (CSPR) pour régler la tension. L'étude de leurs phénomènes transitoires est nécessaire pour maîtriser les risques de l'introduction de ces dispositifs dans le réseau. Dans la littérature [44]-[46], les phénomènes transitoires les plus étudiés sont les surintensités et les surtensions. Les surintensités peuvent endommager les équipements électriques par une forte dissipation de chaleur due à l'effet Joule. De leur côté, les surtensions peuvent mettre à mal l'isolation de l'équipement. L'origine de ces transitoires est multiple : choc de foudre, énergisation, commutations, défaut sur le réseau...

Ce chapitre va présenter un élément essentiel d'une étude de transitoires électromagnétiques : l'équivalent fréquentiel de réseau. On ne modélise généralement pas tout le réseau en détail car cela nuirait aux performances de la simulation. Une zone d'étude est sélectionnée pour être modélisée en détail et le reste du réseau est représenté sous la forme d'un équivalent. Pour comprendre ce principe d'équivalent de réseau, les méthodes de résolution des équations de réseaux dans le domaine temporel sont présentées. Néanmoins, la résolution dans le domaine fréquentiel des équations de réseaux est utile pour certaines études [47], comme les perturbations harmoniques que peut générer l'insertion d'un ouvrage sur le réseau tel qu'un parc éolien. Cela peut également servir pour bien initialiser une simulation temporelle. Seule la résolution temporelle sera considérée ici. Les phénomènes transitoires étant à dynamique rapide (quelques ms), il faut, pour les reproduire, intégrer numériquement les équations différentielles issues des lois de Kirchhoff avec un pas de temps de l'ordre de la microseconde. Une représentation en nombres complexes, dans le domaine phaseur, ne serait pas pertinente car la dynamique serait trop lente avec un pas de temps de simulation de l'ordre de la milliseconde. Dans ce chapitre, après avoir rappelé les méthodes de résolution, l'intérêt de la simulation temps réel sera introduit à travers le retour d'expérience d'un gestionnaire de réseau comme RTE. La plupart des réalisations de cette thèse ont été faites pour la simulation temps réel. Puis, est exposé le principe d'un équivalent fréquentiel de réseau, de sa génération à son implémentation pour la simulation temps réel. Des techniques de représentation en matrices creuses sont utilisées pour optimiser les performances, comme ce qui avait été fait pour le temps différé [27].

3.2 Résolution des équations de réseaux

3.2.1 Réseaux linéaires

L'analyse nodale est la méthode la plus utilisée dans les outils de simulation EMT (Transitoires Electromagnétiques) pour résoudre les équations de réseaux dans le domaine temporel. Tout

d'abord, le temps de simulation est discrétisé suivant un pas de temps Δt pour intégrer les équations différentielles de réseau. Puis, à chaque instant t discrétisé, l'analyse nodale formalise le système linéaire à résoudre : calculer la tension aux nœuds $\mathbf{v}_n(t)$ fonction des injections de courant associées $\mathbf{i}_n(t)$ en résolvant un système linéaire de la forme :

$$\mathbf{Y}_n(t)\mathbf{v}_n(t) = \mathbf{i}_n(t) \quad (3-1)$$

Les caractères en gras dans les équations seront utilisés pour les matrices et les vecteurs. Ici, $\mathbf{Y}_n(t)$ est la matrice d'admittance du réseau calculée à l'instant t . L'indice n indique que le formalisme nodal est utilisé. Par la suite, pour simplifier les notations, la dépendance temporelle t sera implicite. Pour les réseaux linéaires, \mathbf{Y}_n et \mathbf{i}_n sont soit constantes au cours du temps, soit ne dépendent que de t (pas de dépendance vis-à-vis de \mathbf{v}_n). On peut remarquer que cette formulation est aussi valable en complexe, d'où l'utilisation de dénominations plus générales telles qu'admittance et impédance. Par la suite, ces dénominations sont gardées même si pour le cas temporel, les admittances et les impédances sont réelles et égalent respectivement à leur conductance et résistance.

Le remplissage de \mathbf{Y}_n et \mathbf{i}_n est automatisé par composant. Pour chaque composant, connecté aux nœuds k et m , est calculé un équivalent de Norton \mathbf{Y}_{nor} , \mathbf{i}_{nor} qui va contribuer aux nœuds associés dans \mathbf{Y}_n et \mathbf{i}_n . Cet équivalent de Norton discrétisé est appelé le circuit compagnon du modèle.

$$\begin{bmatrix} i_k \\ i_m \end{bmatrix} = \mathbf{Y}_{nor} \begin{bmatrix} v_k \\ v_m \end{bmatrix} + \mathbf{i}_{nor} \quad (3-2)$$

$$\mathbf{Y}_{nor} = \begin{bmatrix} Y_{kk} & Y_{km} \\ Y_{mk} & Y_{mm} \end{bmatrix}, \mathbf{i}_{nor} = \begin{bmatrix} \hat{i}_k \\ \hat{i}_m \end{bmatrix}$$

On retrouve la tension à chaque nœud v_k et v_m ainsi que les courants externes associés i_k et i_m qui sont injectés aux nœuds k et m . \hat{i}_k et \hat{i}_m sont les courants connus qu'injecte le composant aux nœuds k et m . Généralement, l'admittance entre les deux nœuds est symétrique $Y_{km} = Y_{mk}$.

Prenons l'exemple simple d'une résistance formalisée dans les deux conventions (branche et nœud).

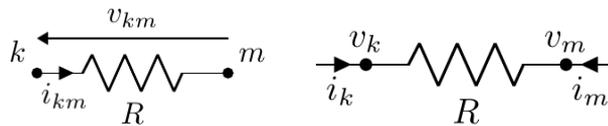


Figure 3-1 : Convention courant/tension de branche pour une résistance (gauche), convention courant/tension aux nœuds (droite).

Tout d'abord, on applique la loi d'ohm avec les valeurs de branche :

$$v_{km} = R i_{km} \quad (3-3)$$

$$i_{km} = \frac{v_k - v_m}{R} \quad (3-4)$$

Puis, on écrit la loi des nœuds en k et en m avec les valeurs aux nœuds :

$$i_k = i_{km} = \frac{v_k - v_m}{R} \quad (3-5)$$

$$i_m = -i_{km} = \frac{v_m - v_k}{R} \quad (3-6)$$

On réécrit ces équations sous la forme d'un système linéaire d'où l'on peut déduire l'équivalent de Norton correspondant. Le courant de Norton est nul car la résistance, un élément passif, n'apporte pas de contribution supplémentaire aux courants des nœuds.

$$\begin{bmatrix} \frac{1}{R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{1}{R} \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} i_k \\ i_m \end{bmatrix} \quad (3-7)$$

Plus généralement, les termes Y_{ij} de la matrice d'admittance \mathbf{Y}_n (matrice carrée de dimension N) de l'analyse nodale sont définis comme suit :

$$Y_{ij} = \begin{cases} y_i + \sum_{\substack{k=1, \dots, N \\ k \neq i}} y_{ik}, & \text{si } i = j \\ -y_{ij}, & \text{si } i \neq j \end{cases} \quad (3-8)$$

Où y_{ij} est l'admittance entre les nœuds i et j , y_{ik} l'admittance entre les nœuds i et k , y_i l'admittance de Norton entre le nœud i et la terre. Les admittances y_{ik} et y_{ij} ont été obtenues respectivement en sommant les admittances des branches en parallèle entre les nœuds i et k , et entre les nœuds i et j .

En sommant les différents courants de Norton arrivant au nœud k , on obtient les éléments i_k des injections de courants connues de \mathbf{i}_n . Pour $k \in \llbracket 1, N \rrbracket$,

$$i_k = \sum_{b \in brch(k)} -\hat{i}_{k_b} \quad (3-9)$$

Où $brch(k)$ est l'ensemble des branches contenant le nœud k et \hat{i}_{k_b} le courant connu qui est injecté dans le nœud k depuis la branche b . Le signe moins devant les courants provient de la convention de signe issue de la définition de l'équivalent de Norton dans l'équation (3-2).

Par ailleurs, les équations électriques, faisant intervenir des équations différentielles, sont intégrées numériquement pour déterminer son circuit compagnon (équivalent de Norton discrétisé). La méthode d'intégration numérique la plus utilisée est la méthode trapézoïdale. Elle reste un bon compromis entre précision et stabilité numérique pour la résolution d'équations différentielles de ce type.

Pour une inductance L , sa loi physique donne pour le formalisme de branche de la Figure 3-2 :

$$v_{km} = L \frac{di_{km}}{dt} \quad (3-10)$$

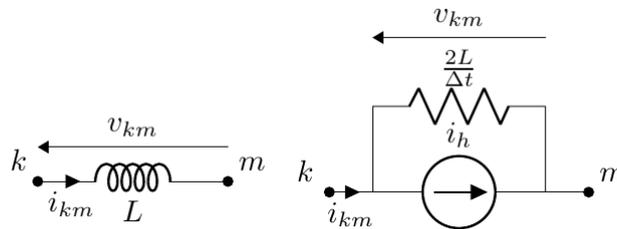


Figure 3-2 : Inductance et son circuit compagnon.

On intègre cette équation différentielle sur la durée d'un pas de temps Δt et on utilise la méthode du trapèze pour approximer le calcul de la première intégrale.

$$\int_{t-\Delta t}^t v_{km} dt = \int_{t-\Delta t}^t L \frac{di_{km}}{dt} dt \quad (3-11)$$

$$\frac{v_{km}(t) + v_{km}(t - \Delta t)}{2} \Delta t = L(i_{km}(t) - i_{km}(t - \Delta t)) \quad (3-12)$$

$$i_{km}(t) = \frac{\Delta t}{2L} v_{km}(t) + \frac{\Delta t}{2L} v_{km}(t - \Delta t) + i_{km}(t - \Delta t) \quad (3-13)$$

On introduit un terme d'historique i_h , appelé aussi courant historique, qui ne dépend que des valeurs des pas de temps précédents ($v_{km}(t - \Delta t)$ et $i_{km}(t - \Delta t)$).

$$i_{km}(t) = \frac{\Delta t}{2L} v_{km}(t) + i_h(t) \quad (3-14)$$

$$i_h(t) = \frac{\Delta t}{2L} v_{km}(t - \Delta t) + i_{km}(t - \Delta t)$$

Après discrétisation, une inductance est équivalente à une source de courant de valeur i_h en parallèle avec une résistance de valeur $\frac{2L}{\Delta t}$. On en déduit l'équivalent de Norton associé :

$$\mathbf{Y}_{nor} = \begin{bmatrix} \frac{\Delta t}{2L} & -\frac{\Delta t}{2L} \\ -\frac{\Delta t}{2L} & \frac{\Delta t}{2L} \end{bmatrix}, \mathbf{i}_{nor} = \begin{bmatrix} i_h \\ -i_h \end{bmatrix} \quad (3-15)$$

Une fois que les tensions aux nœuds ont été calculées en t , il faut mettre à jour les courants historiques pour le prochain pas de temps $t + \Delta t$, à partir de (3-14) en calculant v_{km} via les tensions aux nœuds v_k et v_m . $i_{km}(t)$ est également calculée à partir de (3-14).

$$i_h(t + \Delta t) = \frac{\Delta t}{2L} (v_k(t) - v_m(t)) + i_{km}(t) \quad (3-16)$$

Pour une capacité C , de même on approxime l'intégrale du courant entre $t - \Delta t$ et t :

$$\int_{t-\Delta t}^t i_{km} dt = \int_{t-\Delta t}^t C \frac{dv_{km}}{dt} dt \quad (3-17)$$

$$\frac{i_{km}(t) + i_{km}(t - \Delta t)}{2} \Delta t = C (v_{km}(t) - v_{km}(t - \Delta t)) \quad (3-18)$$

$$i_{km}(t) = \frac{2C}{\Delta t} v_{km}(t) + i_h(t) \quad (3-19)$$

$$i_h(t) = - \left(\frac{2C}{\Delta t} v_{km}(t - \Delta t) + i_{km}(t - \Delta t) \right)$$

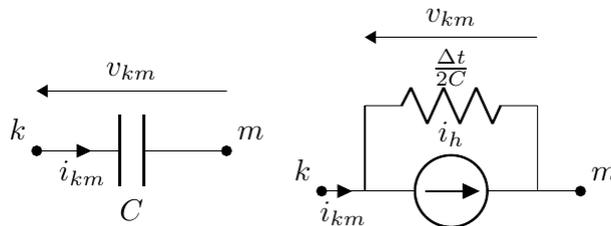


Figure 3-3 : Capacité et son circuit compagnon.

On en déduit l'équivalent de Norton associé :

$$\mathbf{Y}_{nor} = \begin{bmatrix} \frac{2C}{\Delta t} & -\frac{2C}{\Delta t} \\ -\frac{2C}{\Delta t} & \frac{2C}{\Delta t} \end{bmatrix}, \mathbf{i}_{nor} = \begin{bmatrix} i_h \\ -i_h \end{bmatrix} \quad (3-20)$$

Appliquons l'analyse nodale pour le réseau linéaire suivant pour simuler l'énergisation d'une ligne modélisée en PI ayant comme valeurs R_2, C_1, C_2 et L .

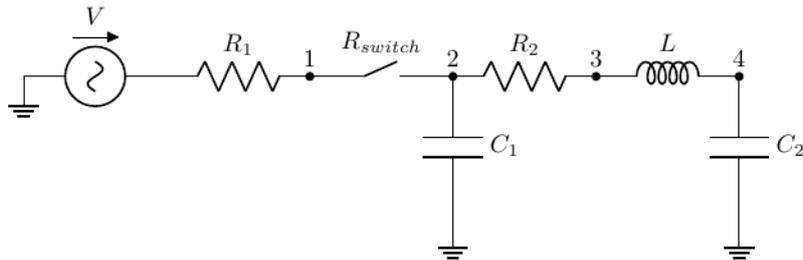


Figure 3-4 : Energisation d'une ligne modélisée en PI.

L'interrupteur a été modélisé par une résistance dont la valeur R_{switch} dépend de son état : ouvert (R_{open}) ou fermé (R_{off}). Dans la figure ci-dessous, chaque branche RLC discrétisée est représentée par son circuit compagnon. La source de tension V avec sa résistance R_1 (équivalent de Thévenin) a été transformée en équivalent de Norton pour pouvoir s'interfacer à la résolution nodale.

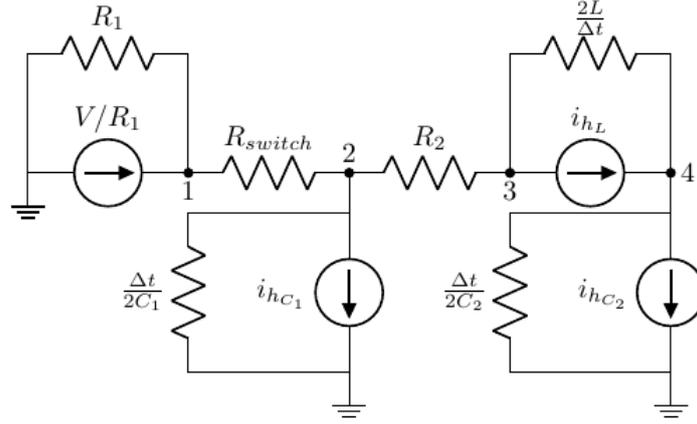


Figure 3-5 : Représentation discrétisée en circuits compagnons de l'énergisation d'une ligne modélisée en PI.

Les équations ci-dessous sont utilisées pour calculer les sources de courants historiques en appliquant les formules (3-14)(3-19) à l'instant t . Le nœud terre est désigné par 0 ($v_0 = 0$).

$$\begin{aligned}
 i_{h_{c_1}}(t) &= -\left(\frac{2C_1}{\Delta t} v_2(t - \Delta t) + i_{20}(t - \Delta t)\right) \\
 i_{h_{c_2}}(t) &= -\left(\frac{2C_2}{\Delta t} v_4(t - \Delta t) + i_{40}(t - \Delta t)\right) \\
 i_{h_L}(t) &= \frac{\Delta t}{2L} (v_3(t - \Delta t) - v_4(t - \Delta t)) + i_{34}(t - \Delta t)
 \end{aligned} \tag{3-21}$$

Avec comme valeurs pour les courants de branche à $t - \Delta t$:

$$\begin{aligned}
 i_{20}(t - \Delta t) &= \frac{2C_1}{\Delta t} v_2(t - \Delta t) + i_{h_{c_1}}(t - \Delta t) \\
 i_{40}(t - \Delta t) &= \frac{2C_2}{\Delta t} v_4(t - \Delta t) + i_{h_{c_2}}(t - \Delta t) \\
 i_{34}(t - \Delta t) &= \frac{\Delta t}{2L} (v_3(t - \Delta t) - v_4(t - \Delta t)) + i_{h_L}(t - \Delta t)
 \end{aligned} \tag{3-22}$$

A partir des circuits compagnons, la matrice d'admittance \mathbf{Y}_n est assemblée en utilisant (3-8) et on écrit la loi des nœuds pour les nœuds pour les nœuds 1, 2, 3 et 4 :

$$\begin{bmatrix} \frac{1}{R_{switch}} + \frac{1}{R_1} & \frac{-1}{R_{switch}} & 0 & 0 \\ \frac{-1}{R_{switch}} & \frac{2C_1}{\Delta t} + \frac{1}{R_{switch}} + \frac{1}{R_2} & \frac{-1}{R_2} & 0 \\ 0 & \frac{-1}{R_2} & \frac{1}{R_2} + \frac{\Delta t}{2L} & \frac{-\Delta t}{2L} \\ 0 & 0 & \frac{-\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{2C_2}{\Delta t} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} -\frac{V}{R_1} \\ i_{h_{c_1}} \\ i_{h_L} \\ i_{h_{c_2}} - i_{h_L} \end{bmatrix} = 0 \tag{3-23}$$

On en déduit le système linéaire à résoudre $\mathbf{Y}_n \mathbf{v}_n = \mathbf{i}_n$ à chaque pas de temps :

$$\begin{bmatrix} \frac{1}{R_{switch}} + \frac{1}{R_1} & \frac{-1}{R_{switch}} & 0 & 0 \\ \frac{-1}{R_{switch}} & \frac{2C_1}{\Delta t} + \frac{1}{R_{switch}} + \frac{1}{R_2} & \frac{-1}{R_2} & 0 \\ 0 & \frac{-1}{R_2} & \frac{1}{R_2} + \frac{\Delta t}{2L} & \frac{-\Delta t}{2L} \\ 0 & 0 & \frac{-\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{2C_2}{\Delta t} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} \frac{V}{R_1} \\ -i_{h_{c_1}} \\ -i_{h_L} \\ i_{h_L} - i_{h_{c_2}} \end{bmatrix} \tag{3-24}$$

Une fois que l'on a formé \mathbf{Y}_n et \mathbf{i}_n via l'analyse nodale et la discrétisation en circuits compagnons, il faut résoudre le système linéaire. La méthode utilisée est la décomposition LU de \mathbf{Y}_n [48][49].

$$\mathbf{Y}_n = \mathbf{L}_n \mathbf{U}_n \tag{3-25}$$

Où U_n une matrice triangulaire supérieure et L_n est une matrice triangulaire inférieure. Les matrices U_n et L_n vont permettre de stocker les multiplicateurs de l'élimination gaussienne afin de ne pas les recalculer à chaque pas de temps. La factorisation est gardée si Y_n n'a pas changé. Une fois que Y_n (matrice carrée de dimension N) est factorisée, la résolution s'effectue en deux étapes en introduisant la variable intermédiaire $U_n v_n = x_n$:

- Une **substitution avant** (calculs des x_{n_i} de 1 à N) en résolvant

$$L_n x_n = i_n \quad (3-26)$$

- Puis, une **substitution arrière** (calculs des v_{n_i} de N à 1) en résolvant

$$U_n v_n = x_n \quad (3-27)$$

Comme on peut le voir avec le réseau linéaire précédent, Y_n est une matrice creuse. Des algorithmes d'algèbre linéaire pour les matrices creuses sont utilisés pour optimiser les performances [48][49]. Lors de chaque pas de temps, la factorisation LU est recalculée partiellement sur les valeurs de Y_n qui ont changé. Une stratégie a été mise en place en temps réel [1][50] afin limiter les refactorisations. Elle consiste à ordonner les valeurs de Y_n , qui sont susceptibles de changer, en bas à droite. Pour l'exemple de la Figure 3-4 en appliquant cet ordonnancement, on obtient pour le système linéaire $Y_n v_n = i_n$:

$$\begin{bmatrix} \frac{1}{R_2} + \frac{\Delta t}{2L} & -\frac{\Delta t}{2L} & 0 & -\frac{1}{R_2} \\ -\frac{\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{2C_2}{\Delta t} & 0 & 0 \\ 0 & 0 & \frac{1}{R_{switch}} + \frac{1}{R_1} & -\frac{1}{R_{switch}} \\ -\frac{1}{R_2} & 0 & \frac{-1}{R_{switch}} & \frac{2C_1}{\Delta t} + \frac{1}{R_{switch}} + \frac{1}{R_2} \end{bmatrix} \begin{bmatrix} v_3 \\ v_4 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -i_{hL} \\ i_{hL} - i_{hC2} \\ V \\ -i_{hC1} \end{bmatrix} \quad (3-28)$$

Ainsi, seule la sous-matrice 2x2 en bas en droite, qui contient la résistance variable de l'interrupteur R_{switch} , nécessite une refactorisation LU.

Une fois que les tensions aux nœuds ont été calculées, les termes historiques sont mis à jour dans i_n à partir de ces nouvelles valeurs, pour la résolution du prochain pas de temps. La Figure 3-6 résume l'ensemble des étapes de la résolution des équations d'un réseau linéaire au cours d'un pas de temps.

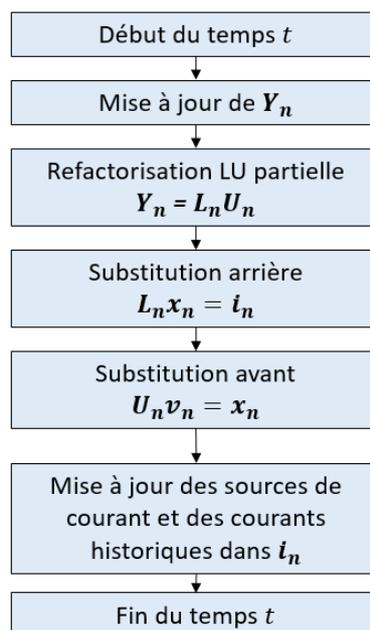


Figure 3-6 : Résolution complète des équations de réseaux pour le temps t .

3.2.2 Analyse Nodale Augmentée Modifiée

L'analyse nodale augmentée modifiée (MANA) [2] permet de résoudre plusieurs limitations de l'analyse nodale. En effet, via une analyse nodale simple, il n'est pas possible de simuler des circuits comportant des sources de tension idéales (une résistance en série est nécessaire), des interrupteurs ou des transformateurs idéaux et plusieurs autres composants qui n'ont pas des relations nodales. Pour ce faire, de nouvelles variables sont considérées. Considérons le cas le plus utilisé où des variables de courants branche à calculer i_x sont rajoutées avec des tensions de branche imposées v_x .

$$\begin{aligned} A_n x_n &= b_n \\ \begin{bmatrix} Y_n & A_c \\ A_d & A_r \end{bmatrix} \begin{bmatrix} v_n \\ i_x \end{bmatrix} &= \begin{bmatrix} i_n \\ v_x \end{bmatrix} \end{aligned} \quad (3-29)$$

Dans la partie augmentée de la matrice, les matrices A_c et A_d permettent de faire la conversion pour les courants et les tensions d'un formalisme de branche (loi des mailles) vers un formalisme de nœud (loi des nœuds) et vis-et-versa. Elles sont composées de -1 et de 1. La matrice A_r représente l'impédance des branches parcourues par le courant i_x . Dans le cas générique, d'autres variables en dehors des courants de branche peuvent être considérées comme par exemples des variables de flux pour les transformateurs, des variables d'états pour les disjoncteurs... Par la suite uniquement l'analyse nodale classique sera utilisée même si certains résultats peuvent être étendus à ce formalisme. Cette méthode est utilisée pour le temps différé [2], mais elle n'est pas disponible dans le temps réel de [1] où seule l'analyse nodale est utilisée. En effet, MANA n'a jamais été implémentée dans HYPERSIM. Reprenons l'exemple de la Figure 3-4. L'analyse nodale augmentée va permettre de pouvoir utiliser un interrupteur idéal sans résistance. Pour se faire, la variable de courant de branches i_{12} forme la partie augmentée. Lorsque l'interrupteur est fermé ($v_1 = v_2$), on obtient :

$$\begin{bmatrix} \frac{1}{R_1} & 0 & 0 & 0 & 1 \\ 0 & \frac{2C_1}{\Delta t} + \frac{1}{R_2} & \frac{-1}{R_2} & 0 & -1 \\ 0 & \frac{-1}{R_2} & \frac{1}{R_2} + \frac{\Delta t}{2L} & \frac{-\Delta t}{2L} & 0 \\ 0 & 0 & \frac{-\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{2C_2}{\Delta t} & 0 \\ 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ i_{12} \end{bmatrix} = \begin{bmatrix} \frac{V}{R_1} \\ -i_{nc1} \\ -i_{nL} \\ i_{nL} - i_{nc2} \\ 0 \end{bmatrix} \quad (3-30)$$

Lorsqu'il est ouvert ($i_{12} = 0$), la matrice devient :

$$\begin{bmatrix} \frac{1}{R_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2C_1}{\Delta t} + \frac{1}{R_2} & \frac{-1}{R_2} & 0 & 0 & 0 \\ 0 & \frac{-1}{R_2} & \frac{1}{R_2} + \frac{\Delta t}{2L} & \frac{-\Delta t}{2L} & 0 & 0 \\ 0 & 0 & \frac{-\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{2C_2}{\Delta t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ i_{12} \end{bmatrix} = \begin{bmatrix} \frac{V}{R_1} \\ -i_{nc1} \\ -i_{nL} \\ i_{nL} - i_{nc2} \\ 0 \end{bmatrix} \quad (3-31)$$

3.2.3 Réseaux non-linéaires

Pour le cas des réseaux non-linéaires, Y_n et i_n ne vont pas dépendre que du temps t . Une dépendance peut exister avec v_n et d'autres paramètres du réseau. Prenons le cas où Y_n et i_n dépendent de la variable v_n . L'équation à résoudre devient :

$$f_n(v_n) = Y_n(v_n)v_n - i_n(v_n) = \mathbf{0} \quad (3-32)$$

Pour résoudre ce type d'équation non-linéaire, on utilise une méthode itérative de Newton. En notant J_n la matrice jacobienne de f_n (variables v_n), on obtient à l'itération k :

$$f_n(v_n^{(k)}) + J_n^{(k)}(v_n^{(k+1)} - v_n^{(k)}) = \mathbf{0} \quad (3-33)$$

Pour le cas générique, la formulation MANA permet de prendre en compte d'autres dépendances non-linéaires autres que v_n . En pratique, la jacobienne n'est pas calculée directement. On cherche à linéariser les composants non-linéaires autour d'un point de fonctionnement [51]. Cela revient comme pour le cas linéaire à fournir un équivalent de Norton linéarisé. Lorsque la linéarisation n'est plus valable, il faut la recalculer. Dans ce cas-là, la matrice d'admittance est exactement la matrice jacobienne ($Y_n = J_n$). Pour une résolution nodale classique, deux types de non-linéarité sont traités de cette manière : les composants de commutation non-linéaire et les composants à caractéristique non-linéaire.

Le premier concerne les diodes, les commutations d'électronique de puissance comme les thyristors, les transistors bipolaires à grille isolée (IGBT). Un modèle d'interrupteur linéarisé est utilisé qui prend deux valeurs de résistance : R_{open} pour l'état ouvert et R_{close} pour l'état fermé. On ne fournit qu'une admittance de Norton. A la différence du modèle linéaire précédent, l'état de l'interrupteur ne dépend pas que du temps mais également de la valeur de la tension.

Le second type se retrouve, par exemple, dans les composants avec saturation comme les transformateurs, les inductances non-linéaires, les parafoudres. Une approximation linéaire par morceaux est utilisée pour la caractéristique non-linéaire. Le calcul de l'équivalent de Norton linéarisé dépend du numéro de segment seg ($Y(seg), i(seg)$). Ainsi, à chaque itération, les nouvelles valeurs des tensions aux nœuds permettent de déterminer le nouveau segment et de mettre à jour la linéarisation.

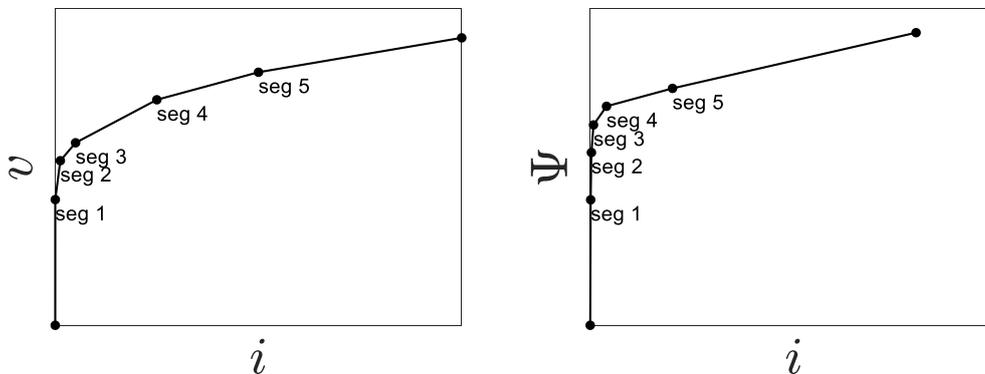


Figure 3-7 : Caractéristique non-linéaire linéarisée par morceau pour un parafoudre (à gauche) et un transformateur saturable (à droite).

La Figure 3-8 présente une résolution nodale classique itérative qui prend en compte les non-linéarités. Une fois que les itérations s'arrêtent pour calculer les v_n définitives, les sources de courant et les historiques sont mis à jour avant de passer au pas de temps suivant. Le critère d'arrêt peut être une combinaison de plusieurs facteurs :

- Un nombre maximum d'itérations fixé par l'utilisateur pour ne pas trop ralentir la simulation.
- Les équivalents linéarisés ne bougent plus : pas de changement des états des interrupteurs ou des numéros de segments.
- Un critère de convergence de l'algorithme de Newton sur $\|v_n^{(k+1)} - v_n^{(k)}\| \leq \varepsilon$ avec ε très petit fixé par l'utilisateur.

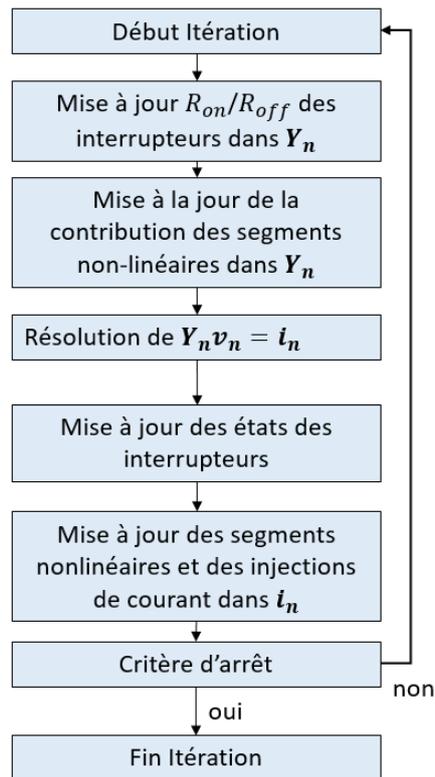


Figure 3-8 : Résolution nodale classique itérative pour les réseaux non-linéaires.

3.2.4 Principe de la simulation temps réel

Le principe de la simulation temps réel est de faire interagir du matériel avec un modèle numérique. On parle alors de simulation Hardware-In-the-Loop (HIL) [52]. Cela permet d'avoir une plus grande précision vis-à-vis de la simulation en temps différé qui n'inclut qu'un modèle numérique. Afin de mimer la réalité, la communication entre le matériel et le modèle numérique se fait par l'intermédiaire d'une horloge temps réel qui cadence la simulation. A chaque pas de temps une contrainte temporelle de synchronisation avec l'horloge réelle est imposée. La simulation numérique doit effectuer ses calculs dans le temps imparti à chaque pas de temps mais ne doit pas aller plus vite pour ne pas se désynchroniser de l'horloge : 1 s simulation temps réel doit durer une 1 s. Lorsque cette contrainte n'est pas respectée, on parle d'un overrun : le matériel connecté à la simulation va recevoir en retard les données de réseaux. Si cela se multiplie, cela peut rendre instable la simulation. A contrario, une simulation en temps différé peut être plus lente ou beaucoup plus rapide que le temps réel. Pour pouvoir respecter la contrainte temps réel, les outils de simulation EMT ont dû mettre en place l'architecture suivante :

- Une machine cible (Target, simulateur) où s'exécute en parallèle la simulation temps réel. La parallélisation est faite soit sur une architecture multiprocesseur (CPU) et/ou FPGA. L'environnement de simulation utilise un système d'exploitation compatible avec le temps réel (Linux).
- Une machine hôte (Host, pc de l'utilisateur) qui permet de préparer le modèle en temps différé, puis de le compiler et de le charger sur la target. Elle sert également pour le suivi du déroulement de la simulation.
- Un réseau de communication pour les échanges entre la target et le host.
- Des entrées/sorties pour communiquer avec le matériel.

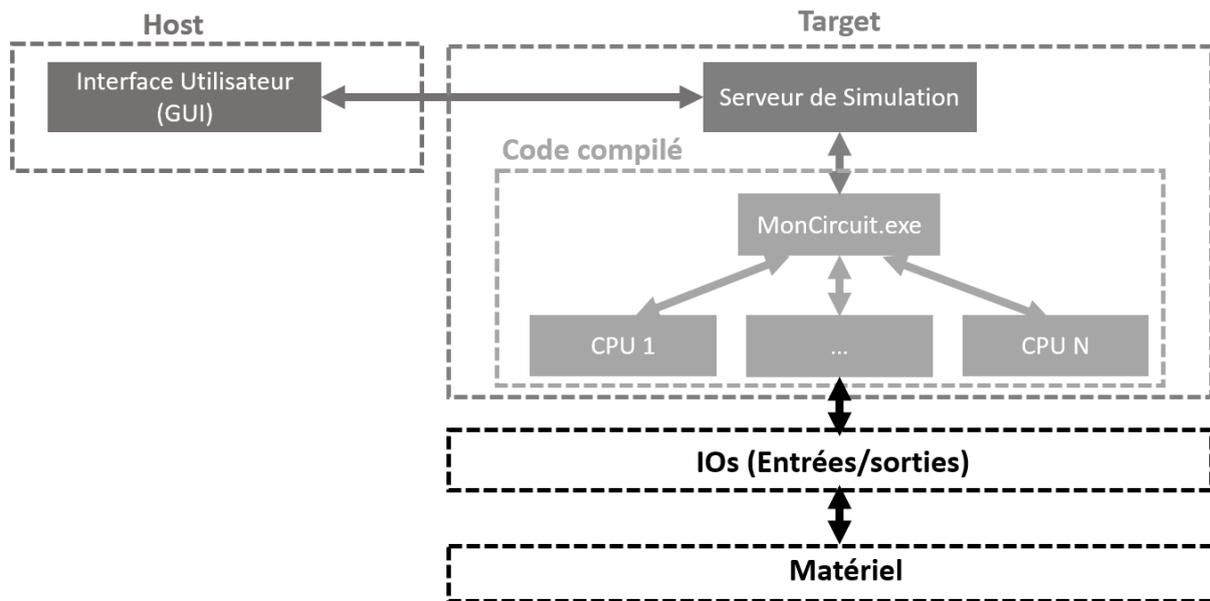


Figure 3-9 : Architecture de simulation EMT temps réel.

Dans le cadre de l'EMT, le matériel dans la boucle de simulation est le plus souvent des systèmes de contrôle et de protections d'équipements de réseau. Le retour d'expérience du gestionnaire de réseau, RTE [21], permet de comprendre l'utilité pratique de la simulation temps réel. En effet, face à la multiplication des ouvrages d'électronique de puissance, la simulation temps réel est utilisée pour maîtriser les risques tout le long de leur cycle de vie. Ainsi pour chaque projet, le système de contrôle et de protections est acheté en double (Figure 3-10) : un exemplaire sera installé sur site, l'autre (la réplique) servira à la simulation temps réel en laboratoire. La partie réseau est modélisée numériquement à l'aide de l'outil de simulation temps réel HYPERSIM. La réplique reçoit les mesures de ce réseau numérique et envoie les ordres correspondant du contrôle. Depuis 2011, RTE a mis en place un laboratoire de simulation temps réel. On y trouve les répliques des contrôleurs suivants :

- 5 Compensateurs Statiques de Puissance Réactive (CSPR),
- 1 liaison HVDC en LCC (convertisseur commuté par ligne),
- 5 liaisons HVDC en VSC (convertisseur en source de tension).

Voici ses avantages qui ont été identifiés vis-à-vis de la simulation en temps différé :

- Une meilleure précision est obtenue en ayant le contrôleur installé sur site dans la boucle de simulation. L'utilisation de modèles génériques ou fournis par les constructeurs de contrôle-commande en temps différé ne permet pas en général d'avoir la même précision. Les fonctionnalités des contrôles et des protections des équipements d'électronique de puissance comme la séquence de démarrage ou la capacité de démarrage autonome (Black Start) sont simplifiées. Néanmoins, certains constructeurs [53][54] sont maintenant capables de fournir des modèles de contrôle qui embarque le code source utilisé sur site pour le temps différé. Cela améliore grandement la précision.
- Une meilleure durabilité est constatée vis-à-vis des modèles en temps différé. Au cours de la vie de l'équipement, le contrôle-commande peut être mis à jour. Les modèles en temps différé ne prendront pas en compte ces changements. De plus, si des correctifs sont mis en place sur le contrôle-commande, ils peuvent être testés en simulation (tests de non-régression) avant d'être appliqués sur site. Le constructeur n'aura pas forcément gardé après la mise en service une plateforme de simulation temps réel du contrôle-commande.

- Elle est utile pour la formation pour la maintenance. Les gestes métiers sont appris et réalisés d'abord sur la réplique du contrôle.

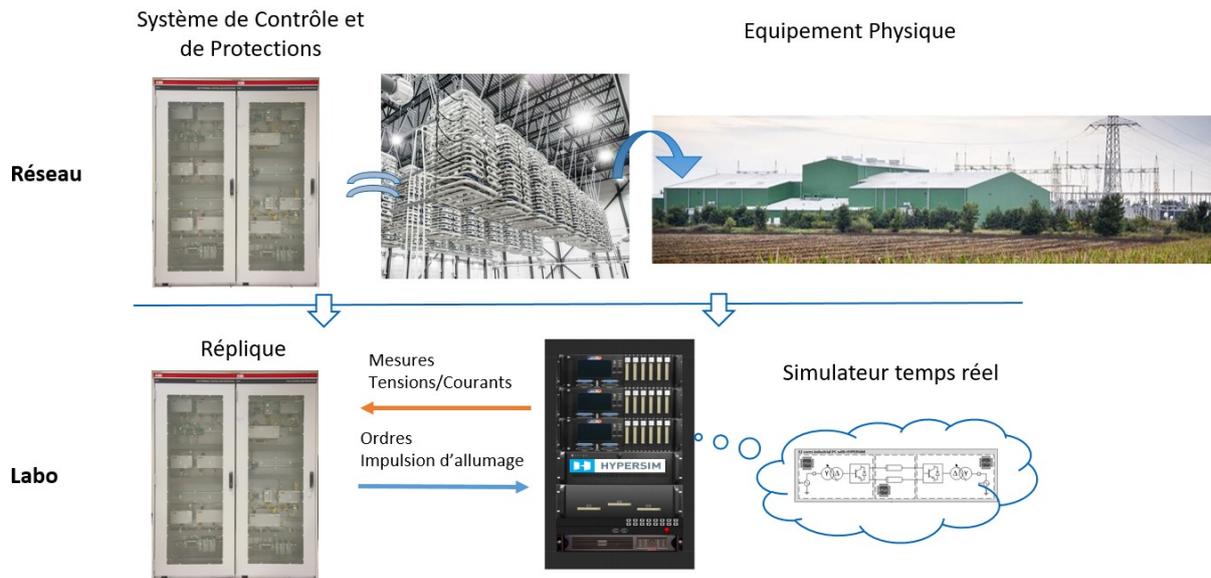


Figure 3-10 : Principe de la simulation HIL avec des répliques de contrôle-commande.

Par la suite, les implémentations et les tests en temps réel seront faits avec HYPERSIM. Pour la simulation en temps différée, EMTP® sera utilisé.

3.3 Principe des équivalents de réseaux

Lors d'une étude de transitoires électromagnétiques, le réseau n'est pas entièrement modélisé en détail [45]. En effet, dans le cas contraire, la taille du système linéaire à résoudre issu de l'analyse nodale serait trop importante en étant égale au nombre de nœuds électriques. La simulation serait très lente. Une zone d'étude est ainsi définie. Chaque composant y est modélisé en détail. Tout autour de cette zone étude, des équivalents de réseau sont utilisés. Cela permet de limiter le nombre de nœuds électriques modélisés. Un équivalent de réseau le plus simple est celui de Thévenin. Il permet de représenter le réseau sous la forme d'une source tension v_{th} et d'une impédance Z_{th} calculées à la fréquence du réseau (50 Hz en Europe, 60 Hz en Amérique du Nord). Des outils de planification permettent d'importer [55] une partie du réseau. A la frontière du réseau importé, les équivalents de Thévenin du réseau restant sont automatiquement calculés.

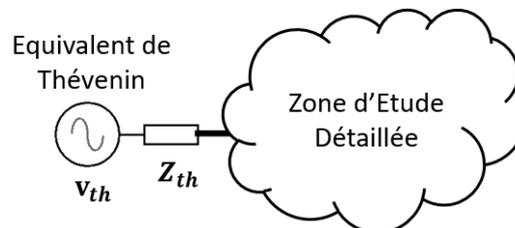


Figure 3-11 : Réduction de la zone hors étude en équivalent de Thévenin.

La principale limitation de cette approche est de fournir une impédance uniquement à la fréquence du réseau. Les phénomènes transitoires étudiés sont souvent de haute fréquence. Par exemple pour l'étude de sursensions voici les plages de fréquence utilisées par origine du phénomène du plus rapide au plus lent : défaut d'isolement de 100 kHz à 50 MHz, choc de foudre de 10 kHz à 3 MHz, sursension de manœuvre de 50 Hz à 20 kHz. Pour pallier à ce problème, un équivalent fréquentiel de

réseau (FDNE, Frequency Dependent Network Equivalent) est utilisé [24]. Via un formalisme d'équations d'état, il représente les impédances du réseau sur une large plage de fréquence.

Un autre intérêt des équivalents de réseau est leur portabilité entre divers outils de simulation tout en préservant la confidentialité. En effet, une représentation en équations d'état de l'impédance du réseau ne révèle rien sur ses données topologiques. Si chaque outil de simulation dispose d'un modèle d'équations d'état, l'équivalent fréquentiel peut être facilement intégré. Ce cas sera illustré avec une comparaison des résultats entre le temps réel et le temps différé.

3.4 Génération des équations d'état d'un équivalent fréquentiel

L'équivalent de réseau fréquentiel (FDNE), qui constituera la représentation d'état, est obtenu à travers les étapes suivantes décrites dans [24] :

- Tout d'abord, un scan fréquentiel est lancé pour mesurer les impédances du réseau à des points précis (les futurs terminaux de l'équivalent) dans le domaine fréquentiel. Pour chaque fréquence f , la matrice d'impédance ci-dessous est obtenue pour un équivalent à trois ports (a , b et c).

$$\mathbf{Z}(s = j\omega) = \begin{bmatrix} Z_{aa}(j\omega) & Z_{ab}(j\omega) & Z_{ac}(j\omega) \\ Z_{ba}(j\omega) & Z_{bb}(j\omega) & Z_{bc}(j\omega) \\ Z_{ca}(j\omega) & Z_{cb}(j\omega) & Z_{cc}(j\omega) \end{bmatrix} \quad (3-34)$$

Où ω est la fréquence angulaire, $\omega = 2\pi f$.

- Puis, un Vecteur Fitting [25][26] est réalisé à partir de ces impédances fréquentielles. Ce dernier consiste à trouver une décomposition en élément simples (recherche des pôles a_i et de la matrice des résidus \mathbf{R}_i) de la matrice d'admittance $\mathbf{Y}(s)$ (l'inverse de la précédente matrice $\mathbf{Z}(s)$) dans le domaine fréquentiel. Le nombre de pôles n à fitter est fixé par l'utilisateur. Il doit faire le compromis entre précision (n grand) et performance (n petit). Ce choix peut être automatisé [26].

$$\mathbf{Y}(s) \cong \sum_{i=1}^n \frac{\mathbf{R}_i}{s - a_i} + \mathbf{D} + s\mathbf{D}_1 \quad (3-35)$$

En réorganisant l'équation, on peut exhiber les matrices de la représentation d'état.

$$\mathbf{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} + s\mathbf{D}_1 \quad (3-36)$$

- Enfin, si nécessaire, on force la passivité de l'équivalent de réseau après le Vector Fitting pour s'assurer qu'un élément passif est obtenu [56][57].

Au final, on obtient une représentation en équations d'état de l'équivalent fréquentiel dans le domaine temporel :

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} \\ \mathbf{i} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{v} + \mathbf{D}_1\dot{\mathbf{v}} \end{aligned} \quad (3-37)$$

Où \mathbf{x} est la variable d'état d'ordre n , \mathbf{v} le vecteur de commande (tensions aux nœuds), \mathbf{i} le vecteur de sorties (courants injectés aux nœuds). On retrouve également les paramètres d'une formulation équations d'état classique : \mathbf{A} la matrice d'état, \mathbf{B} la matrice de commande, \mathbf{C} la matrice d'observation et \mathbf{D} la matrice de transmission directe. La matrice \mathbf{D}_1 a été rajoutée à une formulation classique pour que le vecteur de commande corresponde aux tensions des nœuds externes. Ce terme correspond à l'aspect capacitif du réseau ainsi réduit.

3.5 Intégration des équations d'état à la résolution de réseau

Les équations du domaine temporel (3-37) [58] sont discrétisées par la méthode trapézoïdale. En notant le pas de temps Δt , on obtient pour la première équation :

$$\mathbf{x}(t) = \boldsymbol{\varphi}\boldsymbol{\beta}\mathbf{x}(t - \Delta t) + \frac{\Delta t}{2}\boldsymbol{\varphi}\mathbf{B}\mathbf{v}(t - \Delta t) + \frac{\Delta t}{2}\boldsymbol{\varphi}\mathbf{B}\mathbf{v}(t) \quad (3-38)$$

En définissant :

$$\boldsymbol{\varphi} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}\right)^{-1} \quad (3-39)$$

$$\boldsymbol{\beta} = \mathbf{I} + \frac{\Delta t}{2}\mathbf{A} \quad (3-40)$$

Quant à la deuxième équation de (3-37), on obtient :

$$\mathbf{i}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{v}(t) + \mathbf{i}_c(t) \quad (3-41)$$

En introduisant la définition du courant capacitif :

$$\mathbf{i}_c(t) = \frac{2}{\Delta t}\mathbf{D}_1\mathbf{v}(t) - \frac{2}{\Delta t}\mathbf{D}_1\mathbf{v}(t - \Delta t) - \mathbf{i}_c(t - \Delta t) \quad (3-42)$$

Cette deuxième équation (3-41) peut être réécrite sous le formalisme de résolution bien connu en EMT :

$$\mathbf{i}(t) = \mathbf{G}_{eq}\mathbf{v}(t) + \mathbf{i}_h(t) \quad (3-43)$$

Avec la contribution à la matrice d'admittance du réseau général \mathbf{G}_{eq}

$$\mathbf{G}_{eq} = \frac{\Delta t}{2}\mathbf{C}\boldsymbol{\varphi}\mathbf{B} + \mathbf{D} + \frac{2}{\Delta t}\mathbf{D}_1 \quad (3-44)$$

Et le calcul des courants historiques :

$$\mathbf{i}_h(t) = \mathbf{C}\boldsymbol{\varphi}\boldsymbol{\beta}\mathbf{x}(t - \Delta t) + \left[\frac{\Delta t}{2}\mathbf{C}\boldsymbol{\varphi}\mathbf{B} - \frac{2}{\Delta t}\mathbf{D}_1\right]\mathbf{v}(t - \Delta t) - \mathbf{i}_c(t - \Delta t) \quad (3-45)$$

Tout d'abord, il faut dissocier les calculs qui se feront lors de la préparation de ceux qui seront exécutés en temps réel. La matrice \mathbf{G}_{eq} est calculée lors de la préparation du modèle en étant la contribution à la matrice d'admittance utilisée pour la résolution nodale. A chaque pas de temps (temps réel), les valeurs des courants historiques à chaque nœud \mathbf{i}_h seront calculées à partir de leurs tensions \mathbf{v}_t . Introduisons des variables intermédiaires \mathbf{i}_H et \mathbf{i}_{D1h} .

$$\mathbf{i}_h = \mathbf{i}_H + \mathbf{i}_{D1hist} \quad (3-46)$$

$$\mathbf{i}_H = \mathbf{C}\boldsymbol{\varphi}\boldsymbol{\beta}\mathbf{x} + \frac{\Delta t}{2}\mathbf{C}\boldsymbol{\varphi}\mathbf{B}\mathbf{v} \quad (3-47)$$

$$\mathbf{i}_{D1h} = -\frac{2}{\Delta t}\mathbf{D}_1\mathbf{v} - \mathbf{i}_c \quad (3-48)$$

Cherchons maintenant à faire apparaître $\mathbf{i}_{D1h}(t - \Delta t)$ dans le calcul de $\mathbf{i}_{D1h}(t)$.

A partir de l'expression (3-42) de $\mathbf{i}_c(t)$, on obtient :

$$-\frac{2}{\Delta t}\mathbf{D}_1\mathbf{v}(t) - \mathbf{i}_c(t) = -\frac{4}{\Delta t}\mathbf{D}_1\mathbf{v}(t) - \left(-\frac{2}{\Delta t}\mathbf{D}_1\mathbf{v}(t - \Delta t) - \mathbf{i}_c(t - \Delta t)\right) \quad (3-49)$$

D'après l'expression de \mathbf{i}_{D1h} (3-48) on a :

$$\mathbf{i}_{D1h}(t) = -\frac{4}{\Delta t}\mathbf{D}_1\mathbf{v}(t) - \mathbf{i}_{D1h}(t - \Delta t) \quad (3-50)$$

Et à $t = 0$,

$$\mathbf{i}_{D1h}(0) = -\frac{2}{\Delta t}\mathbf{D}_1\mathbf{v}(0) - \mathbf{i}_c(0) \quad (3-51)$$

Avant de lancer la simulation (partie préparation), on calcule les matrices suivantes qui sont constantes au cours de la simulation :

$$\mathbf{phibeta} = \boldsymbol{\varphi}\boldsymbol{\beta} \quad (3-52)$$

$$\begin{aligned}\mathbf{phiB} &= \frac{\Delta t}{2} \boldsymbol{\varphi B} \\ \mathbf{cphibeta} &= \mathbf{C}\boldsymbol{\varphi\beta} \\ \mathbf{cphiB} &= \frac{\Delta t}{2} \mathbf{C}\boldsymbol{\varphi B}\end{aligned}$$

On peut alors calculer \mathbf{G}_{eq} :

$$\mathbf{G}_{eq} = \mathbf{cphiB} + \mathbf{D} + \frac{2}{\Delta t} \mathbf{D}_1 \quad (3-53)$$

Et à partir des valeurs initiales $(\mathbf{v}(0), \mathbf{x}(0))$ et $\mathbf{i}_c(0)$, on calcule également :

$$\mathbf{i}_{D1h}(0) = -\frac{2}{\Delta t} \mathbf{D}_1 \mathbf{v}(0) - \mathbf{i}_c(0) \quad (3-54)$$

$$\mathbf{i}_H(0) = \mathbf{cphibeta} \mathbf{x}(0) + \mathbf{cphiB} \mathbf{v}(0) \quad (3-55)$$

$$\mathbf{i}_h(0) = \mathbf{i}_H(0) + \mathbf{i}_{D1h}(0) \quad (3-56)$$

Pour la partie temps réel au cours de la simulation, on définit :

- 1 entrée $\mathbf{v}(t)$,
- 1 sortie $\mathbf{i}_h(t)$,
- 4 valeurs historiques (valeurs modifiées à chaque pas de temps) : $\mathbf{phiB} \mathbf{v}(t - \Delta t)$, $\mathbf{x}(t - \Delta t)$, $\mathbf{x}(t)$ et $\mathbf{i}_{D1h}(t)$.

A chaque pas de temps, les calculs suivants sont exécutés dans l'ordre :

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{phibeta} \mathbf{x}(t - \Delta t) + \mathbf{phiB} \mathbf{v}(t - \Delta t) + \mathbf{phiB} \mathbf{v}(t) \\ \mathbf{i}_H(t) &= \mathbf{cphibeta} \mathbf{x}(t) + \mathbf{cphiB} \mathbf{v}(t) \\ \mathbf{i}_{D1h}(t) &= -\frac{4}{\Delta t} \mathbf{D}_1 \mathbf{v}(t) - \mathbf{i}_{D1h}(t - \Delta t) \\ \mathbf{i}_h(t) &= \mathbf{i}_H(t) + \mathbf{i}_{D1h}(t)\end{aligned} \quad (3-57)$$

3.6 Implémentation pour le temps réel

Comme nous l'avons noté précédemment, le calcul effectué par la partie temps réel comporte uniquement des multiplications matrice-vecteur. Les inversions de matrice ne sont calculées que lors de la préparation du modèle. Ainsi, des algorithmes standards d'analyse numérique sont utilisés pour les calculs matriciels effectués une seule fois lors de la préparation :

- BLAS (Basic Linear Algebra Subprograms) librairie fortran pour les calculs simples de type matrice-vecteur (multiplications...).
- LAPACK (Linear Algebra PACKage) librairie fortran spécialisée dans la résolution de systèmes linéaires. Utile pour l'inversion de matrices réelles ou complexes.

Comme il a été déjà montré lors de l'optimisation du modèle d'équations d'état en temps différé [27], le caractère creux de certaines matrices augmente le temps de calcul pour des algorithmes classiques. Cela est le cas pour la partie temps réel où les matrices suivantes sont souvent creuses : $\mathbf{phibeta}$, \mathbf{phiB} et $\mathbf{cphibeta}$.

Ainsi à chaque pas de temps, trois multiplications matrice creuse-vecteur sont effectuées : $\mathbf{phibeta} \mathbf{x}(t - \Delta t)$, $\mathbf{phiB} \mathbf{v}(t)$, $\mathbf{cphibeta} \mathbf{x}(t)$. On verra plus exactement dans l'analyse des performances qu'un algorithme classique de multiplication matrice-vecteur est très lent pour avoir un modèle qui s'exécute en temps réel. Au vu de la simplicité de l'algorithme à implémenter (multiplication matrice-vecteur), une représentation creuse en triplet de la matrice a été choisie à la place de la compression creuse par colonne (CSC) [49] utilisée dans l'optimisation de [27]. Une matrice creuse \mathbf{A} est représentée sous la forme d'un triplet (i, j, p) où p est la liste des éléments non nuls, i et j les listes respectives des indices de ligne et de colonne des éléments non nuls dans l'ordre correspondant de p .

3.7 Performance et Validation

3.7.1 Cas d'un réseau triphasé simple

Le cas triphasé est la réduction fréquentielle en FDNE d'un réseau 225 kV assez simple de lignes en PI. La représentation d'état du FDNE a été obtenue en mesurant l'impédance fréquentielle aux BUS1 et BUS3 et en y exécutant le Vector Fitting (25 pôles). Ainsi, seules les deux lignes en PI centrales sont gardées dans le réseau. On remarque également que les sources de tension ont été remplacées par des équivalents de Norton en cohérence avec la modélisation en équations d'état qui est considérée comme une impédance. Un défaut monophasé (phase A) de 10 Ohm est appliqué entre les deux lignes en PI centrales à $t = 30$ ms et il est éliminé à $t = 60$ ms.

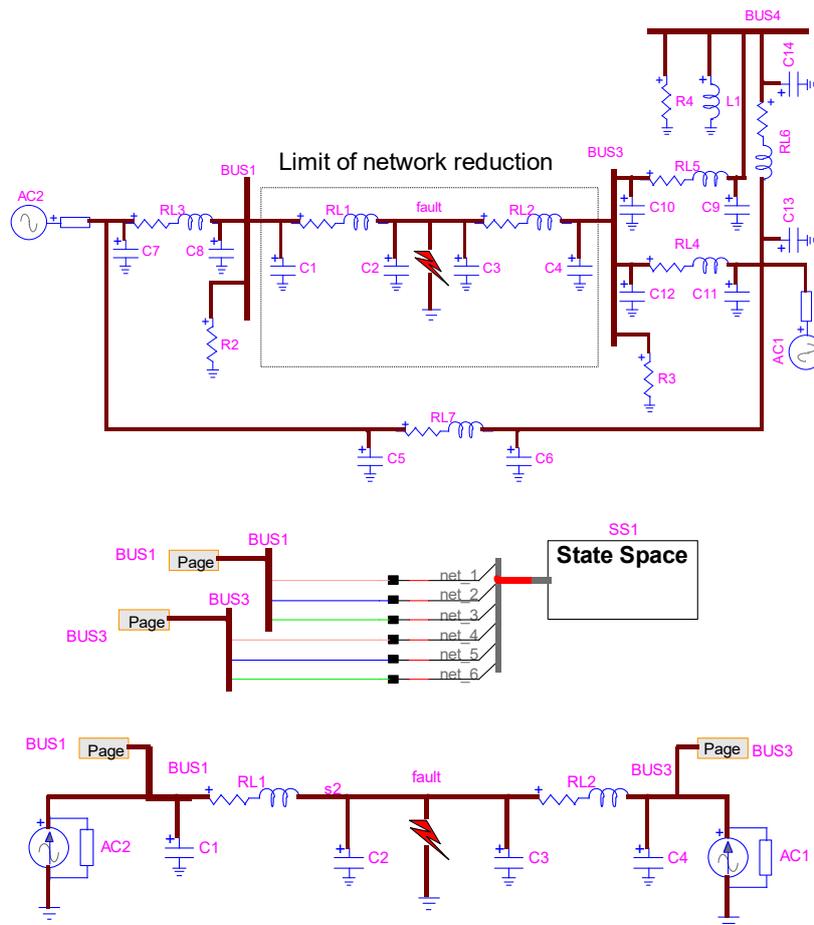


Figure 3-12. Réseau 225 kV avant (en haut) et après (en bas) réduction en FDNE.

Les résultats de la simulation (courant et tension de défaut) ont été superposés pour le réseau complet (RT2) et réduit (RT1) en temps réel ainsi que pour le réseau réduit (EMT) en temps différé. Les erreurs relatives suivantes sont utilisées :

$$\varepsilon = \left| \frac{RT1 - RT2}{RT2} \right|, \quad \alpha = \left| \frac{EMT - RT1}{RT1} \right| \quad (3-58)$$

Les performances temps réel du modèle sont étudiées avec un pas de temps avec le cas simple du réseau 225 kV. Il s'agit de valider la bonne exécution du modèle en temps réel en veillant que les temps d'exécution soit de l'ordre de 50 μ s (pas de temps d'une simulation temps réel). Il est intéressant de voir également si la représentation en matrices creuses procure des gains de performance. Durant les tests, on fait varier le nombre de pôles de la représentation d'état. En effet, le nombre de pôles caractérise sa précision mais aussi sa complexité. Plus ce nombre sera important,

plus les résultats seront précis et l'exécution de la simulation lente. Les tests ont été menés sur un simulateur temps réel linux OP5031 [59] doté de 32 cœurs (E5-2697A v4 Intel Xeon CPU à 2.60 GHz). Le Tableau 3-1 montre l'impact du nombre de pôles sur le temps de simulation ainsi que les algorithmes de matrices creuses.

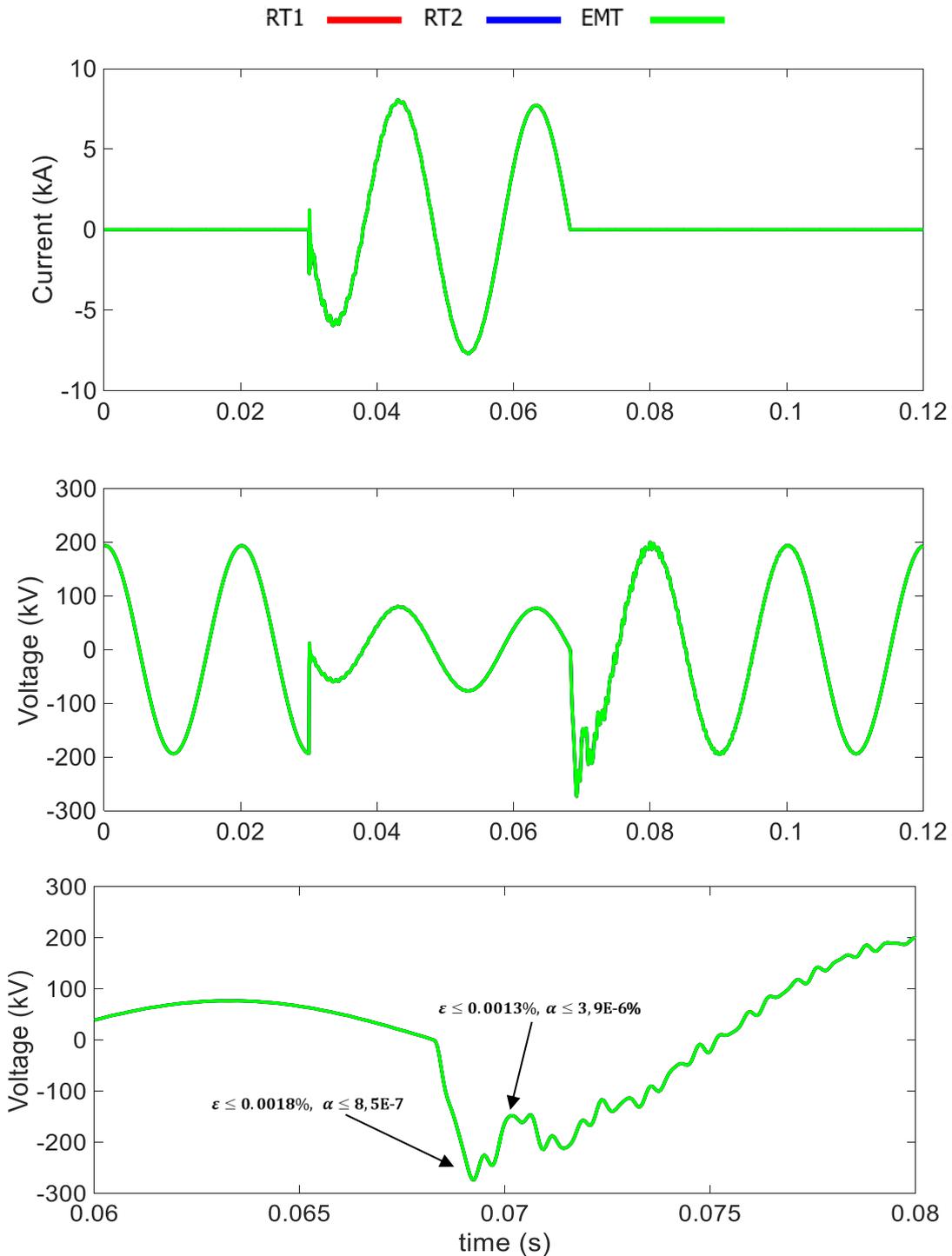


Figure 3-13 : Superposition des résultats entre le réseau entier et sa réduction.

Tableau 3-1 : Comparaison des temps CPU pour la résolution temporelle.

Nombre de Pôles	FDNE	FDNE Algorithme en creux
25	32.1 μ s	3.40 μ s
50	133.6 μ s	5.76 μ s
100	630 μ s	10.16 μ s
150	1396 μ s	14.1 μ s

On remarque que sans la représentation en matrices creuses le modèle échoue à tourner en temps réel lorsqu'il y a plus de 25 pôles ($> 50 \mu$ s). Avec l'algorithme en creux de multiplication matrice-vecteur, les performances sont nettement améliorées et permettent de tenir le temps réel. Même pour 150 pôles, on est en dessous de 20 μ s. Dans le cas présent, il est à noter que seulement 25 pôles suffisent pour avoir une bonne précision du modèle (Figure 3-13 sur la validation du modèle). Généralement, on va rarement au-delà de 150 pôles pour avoir une bonne précision. Ces différents éléments valident le modèle d'équations d'état du FDNE qui tient bien les contraintes temps réel.

3.7.2 Cas de la modélisation du câble d'une liaison HVDC

Un autre cas d'application est la modélisation d'un câble de grande longueur. Les modèles de câble à dépendance fréquentielle, comme le Wideband [27][60], représentent bien les effets de propagation. Cependant, cette modélisation détaillée contraint le pas de temps qui ne doit pas dépasser le minimum des délais de propagation du câble. Le pas temps est ainsi très petit si chaque section du câble est modélisée. Ceci pose un problème de performance pour la simulation temps réel. Le FDNE est une solution pour accélérer le modèle tout en gardant de la précision. Un exemple de ce type de solution est le projet HVDC Savoie-Piémont d'interconnexion entre la France et l'Italie [61]. Les 58 sections des 2*190 km câble DC sont modélisés chacune par un modèle Wideband en temps différé (Figure 3-14). La longueur très courte des sections impose un pas de temps de simulation de 10 μ s, trop rapide pour le temps réel. Les stations VSC ont été représentées avec des modèles détaillés de MMC (Modular Multilevel Converter). La solution pour le temps réel est de faire un équivalent fréquentiel FDNE du câble à 4 ports avec 180 pôles. Des stublines (lignes virtuelles) [62] sont rajoutées pour découpler la résolution de chaque convertisseur et faire tourner la simulation temps réel à un pas de temps de 20 μ s sur 3 CPUs (1 pour chaque convertisseur VSC et 1 pour le FDNE).

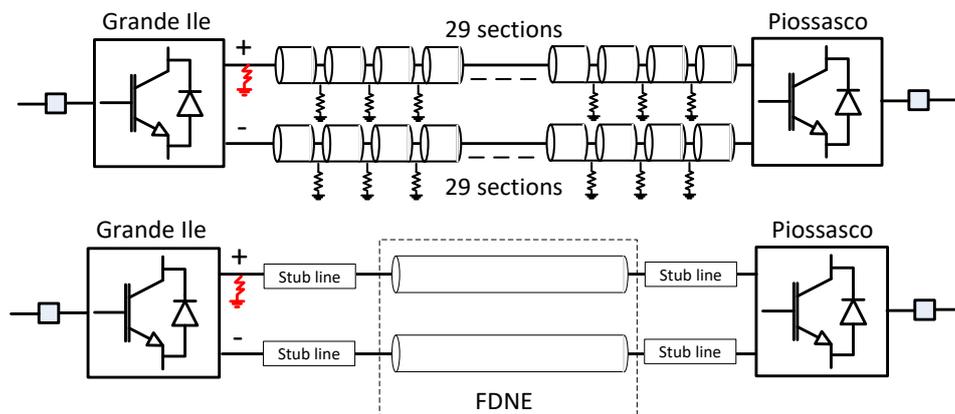


Figure 3-14 : Modélisation des 58 sections d'un câble DC (en haut) et sa réduction en FDNE (en bas).

Les résultats de la simulation temps réel (RT) sont comparés avec la modélisation en 58 Widebands des sections du câble en temps différé (EMT). De légères différences sont observées sur la tension DC pole-terre lors de la séquence de démarrage de la liaison (Figure 3-15), qui proviennent de la réduction du réseau en FDNE.

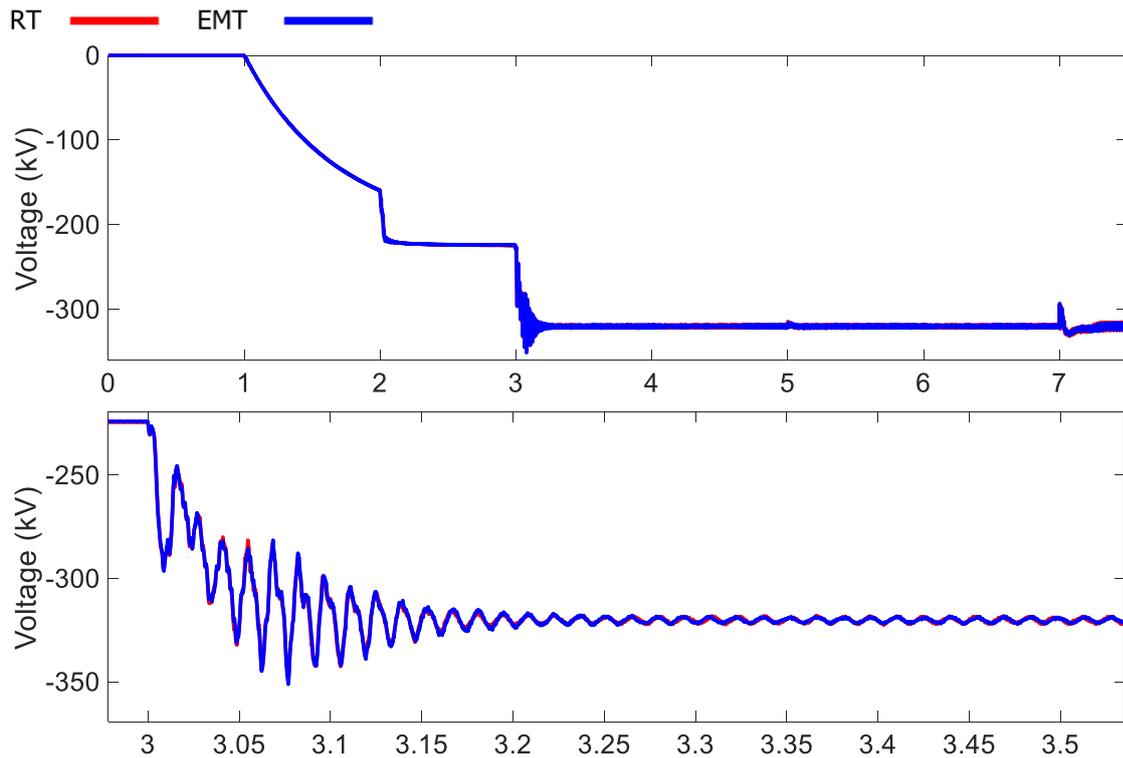


Figure 3-15 : Tension DC du pôle négatif avec zoom durant la séquence de démarrage.

Puis, un défaut pole-terre est simulé à $t = 8$ s sur le pôle positif. La tension sur le pôle sain est représentée sur la Figure 3-16.

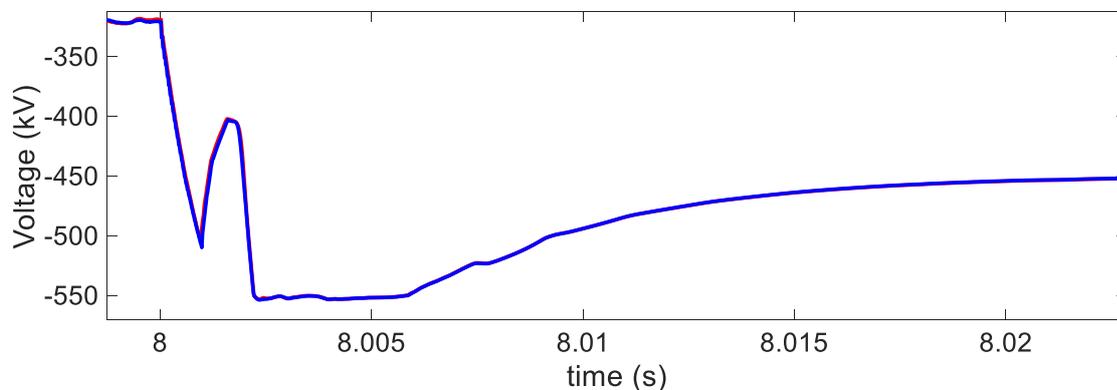


Figure 3-16 : Tension DC au pôle sain durant un défaut pôle terre.

La bonne superposition des deux courbes valide la réduction en FDNE du modèle de câble détaillé pour ce cas précis. Cette approche n'est pas généralisable et doit être utilisée avec prudence. En effet, les phénomènes physiques de propagation le long du câble ne sont pas représentés. Au final, pour ce cas, la réduction en FDNE a permis d'avoir une bonne précision pour l'étude de transitoires en temps réel avec un pas de temps de $20 \mu\text{s}$.

3.8 Conclusion

Ce travail est un premier pas vers une convergence des environnements de simulation entre le temps différé et le temps réel grâce à la portabilité du FDNE. Via un interfaçage en équations d'état, il est possible d'avoir les mêmes résultats de simulation garantis par une même modélisation de l'équivalent fréquentiel. L'utilisation d'une représentation en creux des matrices a permis de tenir les

performances pour le temps réel, contraintes par le pas de temps choisi. La réduction des sections d'un câble DC est un exemple intéressant d'un bon compromis entre performance et précision. L'utilisation du FNDE en temps réel est ainsi incontournable pour les études qui nécessitent une impédance à toutes les fréquences des phénomènes transitoire à simuler tout en optimisant les performances.

Ce premier axe de recherche fut une occasion de découvrir les techniques de modélisation d'une étude de transitoires électromagnétiques et de bien comprendre les besoins en termes de réduction de réseau via un équivalent fréquentiel. L'optimisation des performances du FNDE en temps différé avait déjà été faite dans [27], où des algorithmes des matrices creuses [49] avaient été utilisés pour accélérer le modèle d'équations d'état. La difficulté était d'interfacier ce modèle à la génération de code du temps réel. L'interfaçage à l'analyse nodale via un équivalent de Norton linéaire est assez classique mais nécessite de comprendre les techniques EMT de résolution de réseaux dans leur ensemble (analyse nodale, équivalent de Norton discrétisé, calculs des historiques). L'identification des calculs en temps réel des termes historiques ainsi que l'optimisation des calculs en creux via la génération de code est une contribution originale de ce travail pour satisfaire aux performances temps réel. Le cas d'étude sur la modélisation des sections d'un câble DC en est également une. La représentation en équations d'état de l'impédance fréquentielle de chaque section du câble permet de tenir la contrainte temps réel tout en ayant une bonne précision.

Chapitre 4 Parallélisation des résolutions de réseaux : Séparation en tâches

4.1 Principe de la parallélisation des résolutions de réseaux

Une des techniques les plus utilisées pour accélérer la simulation EMT est la parallélisation de la résolution des équations de réseaux. Elle tire parti de l'architecture moderne des ordinateurs qui est multicœurs que ce soit pour le temps différé ou le temps réel. Une parallélisation s'effectue en deux étapes :

- une séparation en tâches qui sont exécutées en parallèle,
- une répartition des tâches sur les cœurs disponibles de l'architecture de simulation. Les tâches sont regroupées au sein d'un thread qui est exécuté sur un cœur de simulation.

Si on reprend l'introduction aux méthodes de résolution de réseaux du paragraphe 3.2, elle se divise en deux étapes à chaque pas de temps :

1. la résolution d'un système linéaire issu de l'analyse nodale,
2. la mise à jour des valeurs historiques dans les modèles issues de leur discrétisation.

On peut remarquer que la 2^{ème} étape concernant les historiques est naturellement parallélisable. Les mises à jour des historiques sont indépendantes les unes des autres. La séparation étant faite, il ne reste plus qu'à les répartir. Il est en tout autrement pour la résolution des équations nodales qui nécessite un découplage en sous-réseaux. Pour avoir une parallélisation complète, une seule tâche sera identifiée à un sous-réseau où sont regroupées la résolution nodale et la mise à jour des historiques des modèles constituant le sous-réseau. Ainsi, ce chapitre abordera la séparation en tâches de la résolution nodale. Dans le Chapitre 5, sera présenté le problème de répartition des tâches sur les cœurs de simulation afin d'optimiser les performances de la simulation.

4.2 Principe de la Séparation en tâches

Le principe de la séparation en tâches est de découpler le réseau en sous-réseaux qui constituent les tâches à simuler en parallèle. Les méthodes utilisées par les outils de simulation sont basées sur un délai de propagation pour découpler les équations de réseaux. Le délai peut se trouver dans le modèle des lignes de transmission [1][63] ou être ajouté artificiellement pour forcer le découplage [62]. Cependant, lorsqu'il n'y a pas de délai naturel de transmission, l'ajout d'un délai artificiel peut dégrader la précision de la simulation vis-à-vis d'une simulation sans parallélisation [64]. En effet, la résolution des équations de réseaux n'est plus simultanée. C'est souvent le cas pour des réseaux à base d'électronique de puissance ou de distribution, où l'absence de délai naturel ne permet pas une parallélisation avec une bonne précision. Pour ce faire, ce chapitre va présenter une méthode sans délai, la Méthode de Compensation (CM) [28], qui permet un découplage n'importe où tout en gardant une solution simultanée. Dans la présentation des méthodes de cette thèse, deux approches seront abordées : une approche topologique et une approche matricielle. L'approche topologique

sépare le réseau à simuler en sous-réseaux (tâches) et va donner une justification physique au découplage : délai de propagation des lignes pour la séparation avec délai, équivalent de Thévenin et théorème de superposition pour CM. L'approche matricielle est purement mathématique. Elle consiste à paralléliser directement la résolution du système linéaire de l'analyse nodale en mettant la matrice sous une forme diagonale par blocs (avec délai) ou diagonale bordée par blocs (BBD). L'approche topologique est plutôt utilisée pour le temps réel [1] car elle convient bien à la modularité voulue par la génération de code : un fichier source par tâche/sous-réseau. La matrice globale de l'analyse nodale n'est jamais constituée en entier. Pour le temps différé [2][63], l'approche matricielle est privilégiée avec la matrice globale issue de l'analyse nodale.

Un apport important de ce chapitre est de proposer une implémentation de CM au sein d'un environnement temps réel. Si la théorie de ce type de méthode [28][29] avait déjà été énoncée dans les années 70, très peu d'implémentations industrielles avaient été faites pour tester les performances de la méthode en temps réel. Une méthode CM intégrée dans [1] permet d'avoir accès à de nombreux cas industriels de simulation temps réel HIL qui ont été testés dans ce chapitre pour valider la méthode que ce soit pour des réseaux linéaires ou pour des réseaux non-linéaires.

4.3 Parallélisation avec délai

4.3.1 Délai de propagation des lignes

Généralement, les méthodes de parallélisation pour les simulations EMT utilisent le délai de propagation τ des lignes de transmission pour séparer le réseau en tâches. En effet, si le pas de temps Δt est inférieur au délai de propagation τ alors on peut couper le réseau en tâches au niveau de la ligne et tirer parti de ce délai naturel tout en gardant la même précision. La Figure 4-1 ci-dessous résume la modélisation d'une ligne à paramètres constants sans perte entre les nœuds k et m après discrétisation en équivalents de Norton lorsque $\Delta t \leq \tau$. Elle se compose de deux sources de courant historiques dont les valeurs dépendent des courants et tensions de l'autre extrémité (phénomène de propagation).

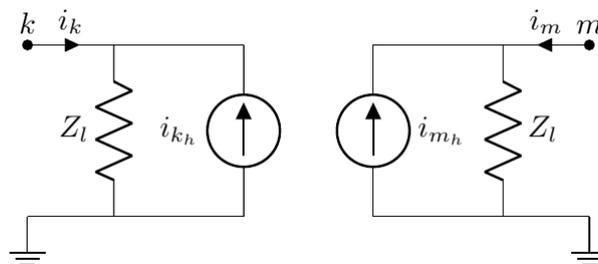


Figure 4-1 : Modèle de ligne à paramètres constants sans perte.

$$\begin{aligned} i_{kh}(t) &= \frac{v_m(t-\tau)}{Z_l} + i_m(t-\tau) \\ i_{mh}(t) &= \frac{v_k(t-\tau)}{Z_l} + i_k(t-\tau) \end{aligned} \quad (4-1)$$

Où Z_l est l'impédance de la ligne. Il n'y a aucune incidence sur la précision des calculs si $\Delta t \leq \tau$ car les sources de courant auront le temps de récupérer les bonnes valeurs de $t - \tau$. On peut noter que par cette modélisation, les nœuds k et m ont été séparés topologiquement. Ils appartiennent à deux sous-réseaux distincts.

Une première **approche topologique**, via une analyse des composants du réseau, identifie les lignes de transmission comme éléments de découplage et crée les sous-réseaux correspondants

assimilables à des tâches. Les sous-réseaux via le modèle de ligne ont été séparés topologiquement. C'est cette méthode qui est utilisée pour le temps réel [1].

Une deuxième **approche matricielle** [63] sépare les tâches directement dans la matrice d'admittance Y_n de l'ensemble du réseau. Cette méthode a déjà été mise en œuvre pour le temps différé dans les outils EMT. Elle consiste à mettre Y_n sous forme BDF (diagonale par blocs). Cette forme est obtenue en appliquant un ordonnancement BTF [65] (Triangulaire par blocs) à la matrice symétrisée Y_n^s de Y_n , ($Y_n^s = Y_n + Y_n^T$). L'ordonnancement obtenu appliqué sur Y_n la met sous forme BDF. Cet algorithme d'ordonnancement de matrices est disponible dans les solveurs LU d'algèbre linéaire [66]. L'équation ci-dessous illustre une matrice d'admittance sous forme BDF :

$$Y_n = \begin{bmatrix} Y_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Y_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & Y_m \end{bmatrix} \quad (4-2)$$

Chaque sous-matrice de la diagonale est la matrice d'admittance d'un sous-réseau qui a été séparé topologiquement par des lignes de transmission comme dans la Figure 4-1. Il ne reste plus qu'à résoudre en parallèle les m systèmes linéaires.

4.3.2 Délai artificiel de lignes virtuelles

Pour de nombreux cas, que ce soient des réseaux de distribution ou des réseaux DC, les délais de propagation des lignes ne peuvent pas être exploités pour paralléliser. En effet, soit les lignes sont très courtes (réseaux de distribution) avec une propagation extrêmement rapide ($\Delta t > \tau$) ou soit inexistantes pour l'électronique de puissance (convertisseurs HVDC...). Pour y remédier, on peut toujours utiliser le principe de découplage d'une ligne de transmission. On insère, à l'endroit où l'on veut découpler le réseau, une ligne virtuelle appelée aussi stubline [62] avec un délai de propagation $\tau = \Delta t$. Généralement, les valeurs de résistance R et d'inductance L de la ligne doivent être fournies et la capacité C est automatiquement calculée pour avoir $\tau = \Delta t$. R et L peuvent être prises à partir d'un enroulement d'un transformateur pour limiter l'impact de la stubline sur la précision. Dans le cas contraire, R doit être très petite et L doit être de l'ordre du mH. Avec l'ajout de ce délai artificiel, la résolution n'est plus simultanée ce qui peut avoir un impact sur la précision. La présence d'une capacité fictive et éventuellement d'une inductance fictive peut dégrader la précision via des résonances et des oscillations non présentes dans la modélisation d'origine.

4.4 Parallélisation sans délai

Afin de remédier à l'imprécision de l'ajout de délais artificiels via les stublines, des techniques de découplage sans délai doivent être envisagées pour préserver une résolution simultanée des équations de réseaux. On a deux grandes approches :

- une **approche topologique** qui va se baser sur les équivalents de réseaux des sous-réseaux séparés (méthode de compensation [28], formulation hybride [29]-[32]).
- une **approche matricielle** qui va mettre la matrice d'admittance sous une forme propice à la parallélisation (forme BBD [34]).

Bien sûr, ces deux approches se recoupent, la première donnant une interprétation physique de la seconde.

4.4.1 Approche topologique

Le principe de ces méthodes topologiques de découplage sans délai est de fournir, au point de coupe, des équivalents de réseaux des deux réseaux découplés. Puis, dans un second temps, il faut

réconcilier les connexions réelles, via les équivalents fournis par les deux extrémités, en appliquant le principe de superposition. Généralement, les types d'équivalent utilisés sont :

- **Equivalent de Thévenin** pour les **méthodes de compensation** [28][67][68] et leur dérivée **MATE** [69][70] (Multi-area Thévenin Equivalent). Initialement utilisées pour la résolution de systèmes non-linéaires, en s'inspirant du principe de la Diakoptique [71][72], elles sont adaptables à la parallélisation de réseaux. Il est important de mentionner que la méthode MATE a été présentée seulement pour les réseaux linéaires. Les auteurs de ces travaux n'ont pas reconnu la similarité évidente avec la méthode de compensation existante.
- **Equivalent de Norton** en appliquant la **Diakoptique en coupant avec des sources de tension** [71][72]. A l'inverse de la première méthode, des équivalents de Norton sont utilisés pour calculer des sources de tension aux points de coupe qui vont être par la suite superposées via un équivalent de Thévenin à la résolution de réseaux. Cela s'applique très bien pour **une formulation en équations de boucle (loi des mailles)**. D'après [72], cette formulation serait la plupart du temps plus optimale que la précédente en diminuant le nombre de variables de coupe à utiliser.
- **Equivalent de Thévenin et de Norton** via la **formulation Hybride** [30][31]. Cette méthode est plus générale que les méthodes de compensation car elle permet de modéliser des sources de tension non-linéaires. Via la Diakoptique, elle a été aussi appliquée à la parallélisation [29] en permettant de combiner les deux types d'équivalents. Plus récemment, cette formulation est à la base de la méthode SSN (State-Space Nodal) [32][33] qui permet de paralléliser n'importe où en couplant analyse nodale et représentation en équations d'état. En effet, il est plus facile et plus courant pour un sous-réseau en équations d'état de fournir un équivalent de Norton au lieu d'un équivalent de Thévenin pour la parallélisation. Ainsi, la méthode SSN permet de combiner plusieurs modélisations. Via la formulation Hybride, on pourrait également interfacer une modélisation en analyse nodale (équivalent de Thévenin) avec une modélisation en équations de boucle (équivalent de Norton).

La méthode de compensation va être par la suite détaillée, étant la plus utilisée dans la littérature dans une approche topologique de la parallélisation. En effet, les équivalents de Thévenin conviennent bien à la résolution nodale classique via l'utilisation d'un équivalent de Norton côté réseaux séparés.

4.4.2 Méthode de Compensation

La méthode de Compensation (CM) [28][67][68] est basée sur deux principes : l'équivalence Thévenin/Norton et le théorème de superposition. Elle consiste à couper des composants de réseaux pour découpler le réseau en plusieurs sous-réseaux. La coupe est généralement faite sur des fils à zéro impédance mais elle peut se faire à travers des branches impédantes linéaires ou non-linéaires. La Figure 4-2 ci-dessous montre ce principe où le découplage en deux sous-réseaux (N_1 et N_2) a été fait à travers des fils. Bien-sûr, cela peut être généralisé à n'importe quel nombre de sous-réseaux qui deviennent indépendants après la coupe de compensation. La sélection des coupes est manuelle.

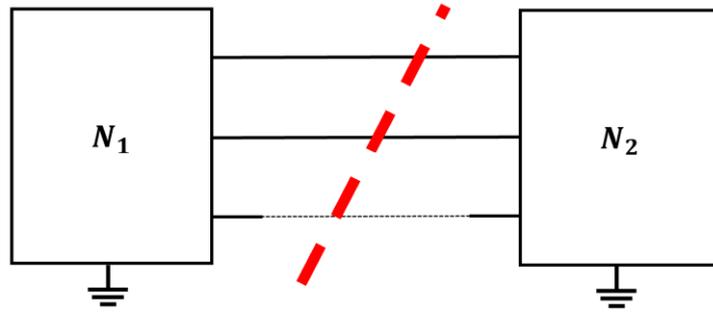


Figure 4-2 : Séparation en deux sous-réseaux à travers des fils par la méthode de Compensation.

Chaque sous-réseau peut être résolu en parallèle avec la formulation nodale et la discrétisation des modèles avec les équivalents de Norton. Ces calculs parallèles doivent être nécessairement complétés par la résolution des branches de coupes dont le courant va compenser la contribution de N_2 dans N_1 et vice versa. La compensation via le courant de branche est illustrée par la Figure 4-3 ce qui permet de réconcilier la dynamique de chaque réseau. Les fils de coupe ont été remplacés par des sources de courant idéales qui injectent le courant de compensation.

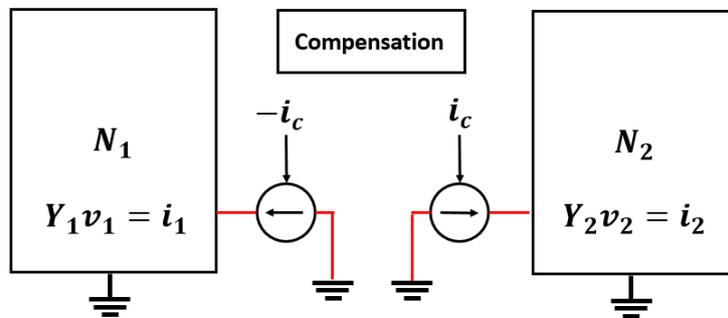


Figure 4-3 : Branches de la compensation remplacées par des sources de courant idéales.

Considérons n_c branches de coupe. On peut définir deux ensembles de nœuds : n_{c1} pour les nœuds à l'extrémité gauche des branches et n_{c2} pour l'extrémité droite. Ils sont de taille n_c . En prenant le vecteur de courant $\mathbf{i}_c = 0$ dans la Figure 4-3, les sous-réseaux N_1 et N_2 sont résolus en parallèle à l'**Etape 1** avec l'analyse nodale.

$$\begin{aligned} Y_1 \mathbf{v}_{n1} &= \mathbf{i}_1 \\ Y_2 \mathbf{v}_{n2} &= \mathbf{i}_2 \end{aligned} \quad (4-3)$$

Où \mathbf{Y} sont les matrices d'admittance, les vecteurs \mathbf{v}_n sont les tensions nodales à calculer et les vecteurs \mathbf{i} sont les injections de courants aux nœuds issues des sources de courant ou des courants historiques. Les tensions de Thévenin \mathbf{v}_{th1} et \mathbf{v}_{th2} aux ensembles de nœuds n_{c1} et n_{c2} sont extraites directement à partir de \mathbf{v}_{n1} et \mathbf{v}_{n2} .

Lors de l'**Etape 2**, les matrices des impédances de Thévenin ($n_c \times n_c$), \mathbf{Z}_{th1} et \mathbf{Z}_{th2} , sont calculées à partir de (4-36) en utilisant la méthode de l'injection de courant unitaire. Pour chaque nœud k appartenant aux ensembles n_{c1} et n_{c2} , on résout :

$$\begin{aligned} Y_1 \mathbf{v}_{1n_{c1}}^{(k)} &= \mathbf{i}_1^{(k)} \\ Y_2 \mathbf{v}_{2n_{c1}}^{(k)} &= \mathbf{i}_2^{(k)} \end{aligned} \quad (4-4)$$

Avec $\mathbf{i}_1^{(k)}$ et $\mathbf{i}_2^{(k)}$ où l'ensemble des injections de courant aux nœuds sont nuls sauf au nœud k où elles sont unitaires égales à 1 A. Ensuite, il n'y a plus qu'à assembler les vecteurs de tensions calculées lors de la résolution des n_c systèmes linéaires précédents pour obtenir les impédances de Thévenin.

$$\mathbf{Z}_{th1} = \begin{bmatrix} \mathbf{v}_{1n_{c1}}^{(1)} & \dots & \mathbf{v}_{1n_{c1}}^{(n_c)} \end{bmatrix} \quad (4-5)$$

$$\mathbf{Z}_{th2} = \begin{bmatrix} \mathbf{v}_{2n_{c2}}^{(1)} & \dots & \mathbf{v}_{2n_{c2}}^{(n_c)} \end{bmatrix}$$

Cette étape est la plus coûteuse en temps de calcul pour la compensation car elle revient à calculer une inversion de matrice. S'il n'y a pas de changements topologiques pour N_1 et N_2 , les équivalents de Thévenin ne changent pas. Il n'est donc pas nécessaire de les recalculer au pas de temps courant.

A l'**Étape 3**, il est maintenant possible de calculer les courants de branche des coupes \mathbf{i}_c en utilisant les équivalents de Thévenin de chaque sous-réseau comme l'illustre la Figure 4-4.

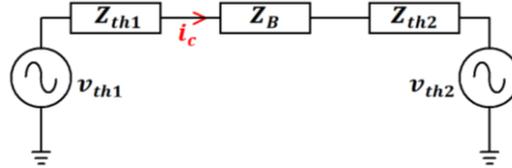


Figure 4-4 : Calcul du courant de compensation pour deux sous-réseaux.

$$\mathbf{Z}_c \mathbf{i}_c = \mathbf{v}_{th2} - \mathbf{v}_{th1} \quad (4-6)$$

Où $\mathbf{Z}_c = \mathbf{Z}_{th1} + \mathbf{Z}_{th2} + \mathbf{Z}_B$. \mathbf{Z}_B est la matrice d'impédance des branches de coupes. Elle est nulle lorsque les coupes ont été faites à travers de fils.

Lors de l'**Étape 4** dans la Figure 4-3, les courants \mathbf{i}_c sont injectés dans chaque sous-réseau pour appliquer la compensation en annulant toutes les autres sources de courant. Les tensions finales \mathbf{v}_n^{final} sont obtenues en superposant les contributions des courants internes \mathbf{v}_n aux contributions des courants de compensation \mathbf{v}_c :

$$\begin{aligned} \mathbf{v}_{n1}^{final} &= \mathbf{v}_{n1} + \mathbf{v}_{c1} \\ \mathbf{v}_{n2}^{final} &= \mathbf{v}_{n2} + \mathbf{v}_{c1} \end{aligned} \quad (4-7)$$

On peut constater que les Etapes 1, 2 et 4 peuvent être calculées en parallèle. L'Étape 3 est la partie séquentielle de la résolution.

4.4.3 Approche matricielle

Une deuxième approche matricielle permet d'avoir une formulation plus mathématique de la séparation de réseaux sans délai. Cette approche s'applique directement à la matrice d'admittance du réseau complet et utilise des techniques de parallélisation de solveurs de systèmes linéaires. On veillera à présenter la correspondance physique avec les méthodes précédemment décrites dans l'approche topologique.

4.4.3.1 Forme BBD (diagonale bordée par blocs)

Pour découpler selon les lignes de transmission, la matrice d'admittance était mise sous **forme BDF** (diagonale par blocs) [63]. Mais quand on ne peut pas découpler selon les lignes, une autre forme est utilisée, la **forme BBD** (diagonale bordée par blocs) qui utilise la notion de matrices frontières dans la séparation en sous matrices [34][63]. Ainsi, la résolution nodale sous forme générique $Ax = b$ peut se mettre sous la forme BBD générique suivante avec $n = m + 1$:

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_{1n} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_m & \mathbf{A}_{mn} \\ \mathbf{A}_{n1} & \dots & \dots & \mathbf{A}_{nm} & \mathbf{A}_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \vdots \\ \mathbf{x}_m \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \vdots \\ \mathbf{b}_m \\ \mathbf{b}_n \end{pmatrix} \quad (4-8)$$

Avec :

- \mathbf{A}_i sont les matrices d'admittance des m sous-réseaux séparés,
- \mathbf{x}_i sont les tensions des nœuds des m sous-réseaux,
- \mathbf{b}_i sont les sources de courant et historiques des m sous-réseaux,
- \mathbf{x}_n est le vecteur des variables d'interface du réseau frontière,
- \mathbf{b}_n sont les sources fixes et historiques du réseau frontière,

- A_{nn} est la matrice du réseau frontière,
- A_{ni} et A_{in} sont les matrices connexions entre le réseau frontière et les m sous-réseaux.

Deux méthodes principales existent pour obtenir cette forme BBD à partir de la formulation nodale suivant le choix des variables d'interface x_n :

- **Variable d'interface en courant.** Ce sont les courants des branches coupées qui sont choisis comme cela est fait pour les méthodes de compensation [63] appliquant le principe de Diakoptique [71][72].
- **Variable d'interface en tension.** Contrairement à la première méthode, aucune nouvelle variable n'est introduite. Les nœuds sont directement ordonnés pour former une forme BBD [73]-[75].

Après, l'utilisation d'une formulation MANA n'interdit pas dans le cas général de considérer d'autres variables d'interface issues de la partie augmentée en plus des courants et des tensions.

4.4.3.2 Résolution en parallèle forme BBD

Des approches purement mathématiques peuvent être utilisées pour résoudre en parallèle les sous-matrices et réconcilier les équations aux frontières dans la forme BBD. Mais à travers ces méthodes mathématiques, on retrouve la même interprétation physique qu'une méthode de compensation qui réconcilie des équivalents de Thévenin. Deux techniques seront présentées :

- Méthode de résolution BBD classique,
- Méthode de résolution BBD avec factorisation LU.

4.4.3.2.1 Méthode de résolution BBD classique

L'utilisation de la forme BBD est une méthode bien connue d'analyse numérique pour réduire la taille des systèmes linéaires à résoudre [48][76][77]. Elle permet également une résolution parallèle par sous-domaines pour des EDP linéaires [78][79]. En considérant une forme BBD en courant, les méthodes de compensation donnent une interprétation physique de cette méthode pour la résolution de réseaux électriques.

La résolution de la forme BBD de (4-8) en parallèle consiste à mettre la matrice sous une forme diagonale simplement bordée par blocs (SBBB).

$$\begin{pmatrix} \mathbf{1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_1 \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{1} & \mathbf{C}_m \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{C}_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_m \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \vdots \\ \vdots \\ \hat{b}_m \\ \vdots \\ \hat{b}_n \end{pmatrix} \quad (4-9)$$

Les différentes étapes sont décrites ci-dessous.

1. Déterminer les membres de droite \hat{b}_i de la forme SBBB (4-9). En enlevant les connexions entre chaque sous-domaine, résoudre en parallèle :

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_m & \mathbf{0} \\ \mathbf{A}_{n1} & \dots & \dots & \mathbf{A}_{nm} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \vdots \\ \vdots \\ \hat{b}_m \\ \hat{b}_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \\ b_n \end{pmatrix} \quad (4-10)$$

La dernière ligne correspond à l'agrégation des valeurs injectées par chaque sous-domaine à la frontière.

2. Calculer les matrices frontières C_i . Avec les connexions enlevées, résoudre en parallèle

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_m & \mathbf{0} \\ \mathbf{A}_{n1} & \dots & \dots & \mathbf{A}_{nm} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_m \\ \mathbf{C}_n \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{1n} \\ \vdots \\ \mathbf{A}_{mn} \\ \mathbf{A}_{nn} \end{pmatrix} \quad (4-11)$$

Cela correspond à trouver les impédances de thévenin/admittance de Norton pour les équivalents de chaque sous-domaine qui seront utilisés pour calculer les variables de frontière.

3. Utilisation de la forme SBBD (4-9) pour calculer les variables d'interfaces \mathbf{x}_n

$$\mathbf{C}_n \mathbf{x}_n = \widehat{\mathbf{b}}_n \quad (4-12)$$

4. Résoudre en parallèle pour chaque sous-réseau,

$$\mathbf{x}_i = \widehat{\mathbf{b}}_i - \mathbf{C}_i \mathbf{x}_n \quad (4-13)$$

Cela revient à appliquer le principe de superposition où $\widehat{\mathbf{b}}_i$ est la solution interne de chaque sous-domaine et $-\mathbf{C}_i \mathbf{x}_n$ la contribution issue des autres sous-domaines.

Ainsi, la résolution BBD a été décrite pour le cas général : la résolution des équations de réseaux utilise MANA et les équations d'interface peuvent être indifféremment des courants, des tensions ou d'autres variables qui sont présentés dans la partie augmentée. Si on se ramène dans le cas de la compensation avec une formulation nodale classique, on peut réécrire (4-8) sous la forme :

$$\begin{pmatrix} \mathbf{Y}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{S}_1^T \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{Y}_m & \mathbf{S}_m^T \\ \mathbf{S}_1 & \dots & \dots & \mathbf{S}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_m \\ \mathbf{i}_n \end{pmatrix} = \begin{pmatrix} \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_m \\ \mathbf{0} \end{pmatrix} \quad (4-14)$$

Les variables d'interface sont les courants \mathbf{i}_n de branche/compensation où l'on a découplé les sous-réseaux. Comme le découplage se fait à travers des fils, on a $\mathbf{A}_{nn} = \mathbf{0}$. Aucune tension de branche est imposée, $\mathbf{b}_n = \mathbf{0}$. Les matrices \mathbf{S}_i , pour $i \in \llbracket 1, m \rrbracket$, sont les connexions topologiques entre les sous-réseaux. Pour la compensation, ces matrices sont généralement symétriques, $\mathbf{S}_i = \mathbf{A}_{in} = \mathbf{A}_{ni}^T$. En effet, ces matrices de connectivité \mathbf{S}_i contiennent le coefficient +1 pour le nœud à gauche de la coupe et -1 pour celui à droite.

Si l'on considère le cas ($m = 2, n = 3$), on peut dresser la correspondance entre la compensation de la Figure 4-6 et la forme SBBD associée. Les contributions internes des sources de courant aux calculs des tensions sont définies par :

$$\begin{aligned} \widehat{\mathbf{b}}_1 &= \mathbf{v}_{n1} \\ \widehat{\mathbf{b}}_2 &= \mathbf{v}_{n2} \end{aligned} \quad (4-15)$$

A partir de la forme SBBD, on peut identifier les équivalents de Thévenin de chaque sous-réseau, $(\mathbf{Z}_{th1}, \mathbf{v}_{th1})$ et $(\mathbf{Z}_{th2}, \mathbf{v}_{th2})$.

$$\begin{aligned} (\mathbf{S}_1 \mathbf{C}_1, -\mathbf{S}_1 \widehat{\mathbf{b}}_1) &= (\mathbf{Z}_{th1}, \mathbf{v}_{th1}) \\ (\mathbf{S}_2 \mathbf{C}_2, \mathbf{S}_2 \widehat{\mathbf{b}}_2) &= (\mathbf{Z}_{th2}, \mathbf{v}_{th2}) \end{aligned} \quad (4-16)$$

Puis, le courant de compensation \mathbf{i}_c est calculé à partir des deux équivalents de Thévenin.

$$\begin{aligned} \widehat{\mathbf{b}}_3 &= \mathbf{v}_{th1} - \mathbf{v}_{th2} \\ \mathbf{C}_3 &= \mathbf{Z}_{th1} + \mathbf{Z}_{th2} \\ \mathbf{i}_3 &= \mathbf{i}_c \end{aligned} \quad (4-17)$$

Enfin, $-\mathbf{C}_1 \mathbf{i}_3$ et $-\mathbf{C}_2 \mathbf{i}_3$ sont les contributions en tensions (\mathbf{v}_{c1} et \mathbf{v}_{c2}) qui proviennent de l'injection de \mathbf{i}_c dans chaque sous-réseau. La superposition des tensions peut se faire.

$$\begin{aligned} \widehat{\mathbf{b}}_1 - \mathbf{C}_1 \mathbf{i}_3 &= \mathbf{v}_{n1} + \mathbf{v}_{c1} \\ \widehat{\mathbf{b}}_2 - \mathbf{C}_2 \mathbf{i}_3 &= \mathbf{v}_{n2} + \mathbf{v}_{c2} \end{aligned} \quad (4-18)$$

Il faut noter que toutes les résolutions de systèmes linéaires sont faites via une factorisation LU.

4.4.3.2.2 Méthode de résolution BBD avec factorisation LU

Cette approche avait été esquissée dans la classification des méthodes de compensation dans [67]. Habituellement, on utilise une méthode de post compensation : on résout d'abord les équations de réseaux et on compense par la suite. Mais la méthode de Compensation moyenne de [67] s'écrit directement sur la décomposition LU des matrices d'admittances. La compensation se fait sur la variable intermédiaire x lors de la substitution avant.

$$Lx = i \quad (4-19)$$

Récemment, [75] utilise cette variable intermédiaire en utilisant la décomposition LU où l'on retrouve les étapes de la compensation moyenne. En effectuant la décomposition LU de la formulation précédente, on obtient les matrices $L_i U_i U_{in} L_{ni}$ avec $n = m + 1$ telles que

$$\begin{pmatrix} A_1 & 0 & \dots & 0 & A_{1n} \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & A_m & A_{mn} \\ A_{n1} & \dots & \dots & A_{nm} & A_n \end{pmatrix} \begin{pmatrix} L_1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & L_m & 0 \\ L_{n1} & \dots & \dots & L_{nm} & L_{nn} \end{pmatrix} \begin{pmatrix} U_1 & 0 & \dots & 0 & U_{1n} \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \dots & \ddots & 0 & \vdots \\ \vdots & \dots & \ddots & U_m & U_{mn} \\ 0 & \dots & \dots & 0 & U_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_m \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \\ \vdots \\ b_n \end{pmatrix} \quad (4-20)$$

Voici les différentes étapes pour calculer cette décomposition LU d'après [73] en utilisant la méthode du bloc LU :

1. Décomposition LU des blocs diagonaux en parallèle pour $i \in \llbracket 1, m \rrbracket$,

$$A_i = L_i U_i \quad (4-21)$$

2. Pour $i \in \llbracket 1, m \rrbracket$, on en déduit la décomposition LU des blocs frontière verticaux U_{in} et horizontaux L_{ni} en résolvant en parallèle les systèmes linéaires suivants :

$\dim(A_{in})$ Substitutions avant pour chaque colonne de A_{in}

$$L_i U_{in} = A_{in} \quad (4-22)$$

$\dim(A_{ni})$ Substitutions arrière pour chaque ligne de A_{ni}

$$L_{ni} U_i = A_{ni} \quad (4-23)$$

3. Calcul de la décomposition LU du bloc frontière L_{nn}, U_{nn}

$$A_n = L_{nn} U_{nn} + \sum_{i=1}^m L_{ni} U_{in} \quad (4-24)$$

$$A_n - \sum_{i=1}^m L_{ni} U_{in} = L_{nn} U_{nn}$$

Cela revient à calculer la décomposition LU de $A_n - \sum_{i=1}^m L_{ni} U_{in}$.

On pose y_i comme variables intermédiaires lors de la substitution avant telles que :

$$\begin{pmatrix} L_1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & L_m & 0 \\ L_{n1} & \dots & \dots & L_{nm} & L_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \\ b_n \end{pmatrix} \quad (4-25)$$

Avec $y_i = U_i x_i$ pour $i \in \llbracket 1, m \rrbracket$ et $y_{nn} = U_{nn} x_n$

Et résolvons les équations d'interface :

$$L_{nn} y_n = b_n - \sum_{i=1}^m L_{ni} L_i^{-1} b_i \quad (4-26)$$

On peut alors calculer x_n .

$$U_{nn} x_n = y_n \quad (4-27)$$

Et on en déduit les autres x_i .

$$U_i x_i = y_i - U_{in} x_n \quad (4-28)$$

Ainsi, on retrouve ci-dessous les étapes de calcul décrites dans [67][73][75] en indiquant la correspondance avec les étapes de la compensation moyenne :

1. Calculer y_i pour chaque sous partie $i \in \llbracket 1, m \rrbracket$ en parallèle (**début résolution des équations de réseaux**)

$$L_i y_i = b_i \quad (4-29)$$

2. Calculer la variable intermédiaire frontière y_n

$$L_{nn} y_n = b_n - \sum_{i=1}^n L_{ni} y_i \quad (4-30)$$

3. Substitution arrière pour calculer la variable frontière x_n (**calcul vecteur de compensation**)

$$U_{nn} x_n = y_n \quad (4-31)$$

4. Résolution en parallèle des vecteurs x_i avec $i \in \llbracket 1, n \rrbracket$ à partir de x_n (**compensation et résolution complète des équations de réseaux**)

$$U_i x_i = y_i - U_{in} x_n \quad (4-32)$$

Comme énoncé dans [67], cette méthode de résolution présente moins de complexité qu'une méthode BBD classique et devrait être plus performante. Il semble difficile d'utiliser cette méthode pour une approche topologique car les matrices de bordure A_{ni} et A_{in} ne sont jamais explicitement formées.

4.4.4 Automatisation de la séparation en tâches

A l'image ce qui a déjà été fait pour la parallélisation par les lignes de transmission (analyse topologique [1] ou forme BDF [63]), de nombreux travaux se sont intéressés à l'automatisation de la séparation en tâches sans délai. Si au départ des heuristiques ont été proposées [80] pour résoudre ce type de problème, le partitionnement de graphe s'est imposé comme étant la plus utilisée que l'on retrouve implémentée dans les bibliothèques comme Metis [81][82] ou Scotch [37][83]. Comme cela sera analysé dans Chapitre 5, ces heuristiques se sont imposées dans le problème de partitionnement de graphe en étant très rapides via le multi-niveaux (partition sur un graphe plus petit en regroupant les nœuds) et en proposant différentes heuristiques pour améliorer la partition suivant le critère de load balancing (taille des partitions) et de minimisation des communications.

Pour l'**approche topologique**, comme dans [38], on peut estimer pour chaque élément de réseau le temps de calcul en flops au niveau de la mise à jour des historiques. Cette estimation semble suffire pour estimer la part de calcul dans la résolution nodale. On construit alors un graphe dont les sommets sont les éléments de réseaux avec le temps calcul estimé et les arcs représentent les échanges de signaux entre les éléments (valués par le nombre de signaux). Ce graphe est ensuite partitionné en m sous-réseaux via un algorithme de partitionnement (Metis ou Scotch). Le nombre m de sous-réseaux est fixé par l'utilisateur. Une fois les m partitions obtenues, il n'y a plus qu'à introduire des sources de courant pour couper physiquement les réseaux [84] et à mettre en place la méthode de compensation. Cette approche conviendrait très bien à un environnement temps réel [1] qui définit la notion de tâche de chaque sous-réseau en les séparant topologiquement.

Pour l'**approche matricielle**, la partition intervient directement sur la matrice d'admittance où A est mise sous forme de graphe. Si elle n'est pas symétrique, on prend sa forme symétrisée $A + A^T$. Le graphe obtenu (V, E) est tel que V contient l'ensemble des nœuds du réseau et $(v_i, v_j) \in E$ si $A_{ij} \neq 0$. En utilisant des outils de partitionnement de graphe comme Metis ou Scotch, on obtient m partitions de nœuds qui sont équilibrés en taille (load balancing), et qui minimisent le nombre de communications (taille des blocs frontières dans la matrice en forme BBD). A partir de ces partitions, on peut mettre la matrice d'admittance sous forme BBD :

- **Cas variables frontières de courant** (Compensation)

On introduit des sources de courant qui vont être les variables de courant aux branches qui séparent chaque partition. On obtient ainsi la forme BBD classique si on effectue des coupes à zéro impédance via la méthode de compensation.

- **Cas variables frontières de tension (Node Tearing)**

On détermine les nœuds frontières, qui retirés de chaque partition, séparent les partitions entre elles [73][75]. Ces nœuds frontières vont devenir les variables de tension d'interface. Une fois déterminé l'ensemble des nœuds frontières, il suffit de réordonner la matrice pour obtenir la forme BBD souhaitée.

Remarque : Ces méthodes ne prennent pas en compte la nature non-linéaire de certains éléments de réseaux dans les estimations du temps de calcul de la résolution nodale. Ces éléments ont un impact sur les performances en obligeant des refactorisations LU. Ainsi, pour le cas linéaire, cela semble pertinent, un peu moins pour le cas non-linéaire.

4.5 Implémentation de la méthode de Compensation

L'implémentation de la méthode de Compensation a été réalisée dans l'environnement temps réel de [1] en utilisant une approche topologique illustrée par la Figure 4-5. La séparation en tâches est faite manuellement. Un élément de découplage de Compensation permet d'indiquer les coupes à considérer. Pour le réseau ci-dessous, deux éléments de découplage de Compensation séparent le réseau en deux sous-réseaux.

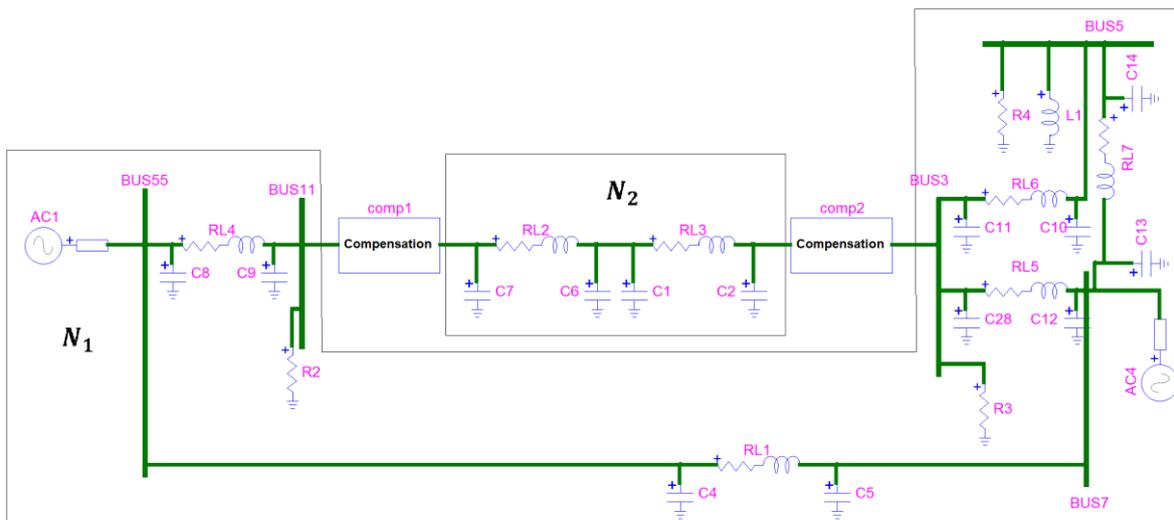


Figure 4-5 : Séparation en deux sous-réseaux via un élément de découplage de Compensation.

Ce paragraphe va détailler les mécanismes de parallélisation mis en œuvre pour la méthode de Compensation. L'algorithme de la solution parallélisation est présenté. Une version itérative est utilisée pour gérer les non-linéarités et reprend les étapes du cas linéaire. Enfin, une autre application intéressante est évoquée pour la Compensation. En plus du découplage, la Compensation permet de mettre en place un multiplicateur de courant très utile pour les équivalents de l'électronique de puissance.

4.5.1 Mécanisme de parallélisation

Dans le paragraphe 4.4.2, les calculs des équivalents de Thévenin (étapes 1 et 2) peuvent être faits en parallèle. Il en est de même pour la superposition (étape 4). Seule l'Étape 3 est séquentielle et est réalisée par une tâche de compensation. Le schéma ci-dessous (Figure 4-6) montre un algorithme en

trois tâches lorsqu'on découple en deux sous-réseaux N_1 et N_2 par la compensation. Il peut être réduit à deux tâches si la tâche de compensation est exécutée dans l'un des postes découplés. La synchronisation entre la tâche de compensation et celles des sous-réseaux, pour recevoir les équivalents de Thévenin et transmettre le courant de branche, utilise le mécanisme bien connu de barrière en calcul parallèle. Une barrière est un compteur partagé qui est incrémenté par un système de verrou lorsqu'un thread y arrive. Une fois que le dernier thread arrive, il libère la barrière et fait cesser l'attente des autres threads. Une barrière est déjà utilisée pour synchroniser les processeurs à la fin de chaque pas temps afin d'effectuer les communications entre processeurs. Ici, cela permet une communication à l'intérieur d'un pas de temps où la tâche compensation attend la réception de chaque équivalent de Thévenin et où chaque poste découplé attend le courant de branche pour réaliser la superposition.

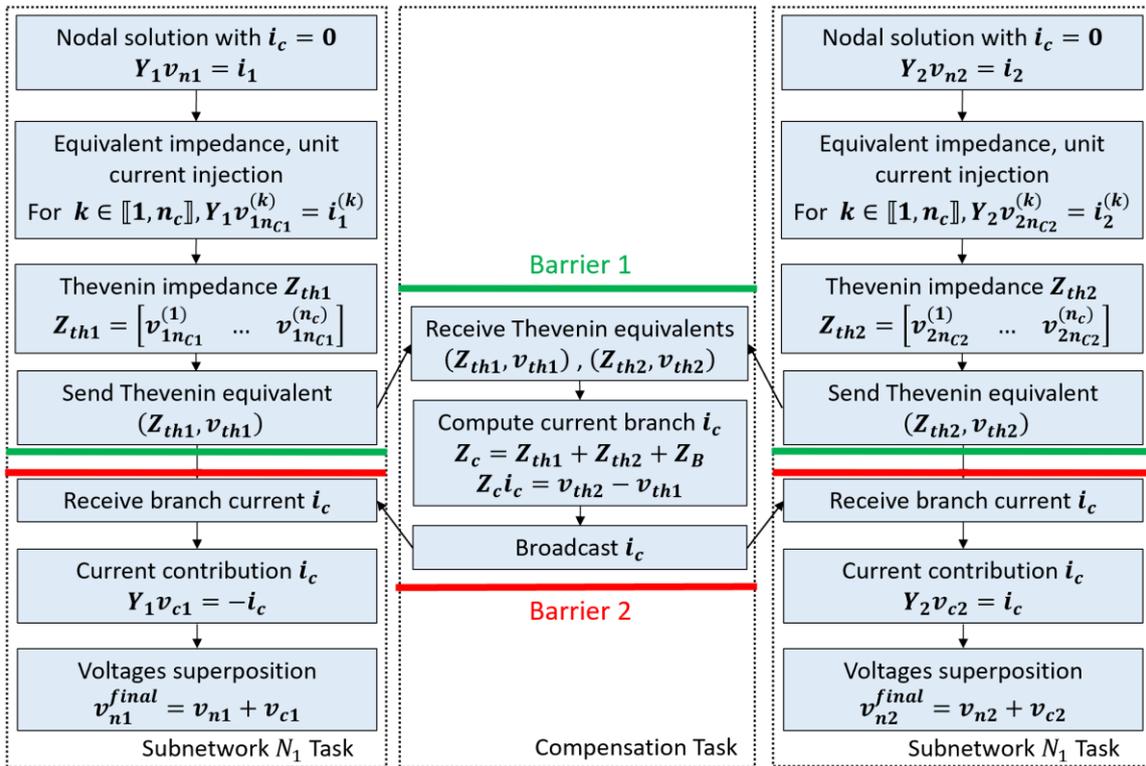


Figure 4-6 : Algorithme de Compensation en trois tâches pour la séparation en deux sous-réseaux.

A noter que dans ce schéma, le cas général a été considéré où la branche de coupe peut avoir une impédance Z_B .

4.5.2 Méthode de Compensation non-linéaire

Mettons sous la forme BBD, les équations impliquant un découplage de compensation entre les sous-réseaux N_1 et N_2 .

$$Ax = b \quad (4-35)$$

$$\begin{bmatrix} Y_1 & 0 & S_k^T \\ 0 & Y_2 & S_m^T \\ S_k & S_m & Z_B \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_c \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ 0 \end{bmatrix}$$

$$\Phi(v_1, v_2, i_c) = \begin{bmatrix} Y_1 & 0 & S_k^T \\ 0 & Y_2 & S_m^T \\ S_k & S_m & Z_B \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_c \end{bmatrix} - \begin{bmatrix} i_1 \\ i_2 \\ 0 \end{bmatrix} = 0 \quad (4-36)$$

Avec S_k et S_m les matrices de connectivité entre les sous-réseaux N_1 et N_2 à travers l'impédance de branche Z_B . Chaque ligne de S_k contient des coefficients égaux à +1 pour chaque nœud à gauche de

la coupe de branche. Chaque colonne de S_m contient des coefficients égaux à -1 pour chaque nœud à droite de la coupe de branche.

Si le réseau à simuler contient des non-linéarités, Φ est une fonction non-linéaire et une méthode de Newton itérative doit être utilisée pour résoudre (4-36). Comme aux paragraphes 3.2.2, les éléments non-linéaires vont être linéarisés autour d'un point de fonctionnement. Par définition, S_k et S_m sont constantes et linéaires et $Z_B = \mathbf{0}$ pour des coupes selon des fils (cas que l'on considère ici). Il n'est pas nécessaire de les linéariser. Y_1, Y_2, i_1, i_2 peuvent contenir des non-linéarités et doivent être linéarisés. Cela revient à linéariser les éléments non-linéaires de N_1 et N_2 pour que A soit identifiable à la matrice Jacobienne. Des équivalents de Norton linéarisés seront inclus dans Y_1 et i_1 , idem pour l'autre sous-réseau dans Y_2 et i_2 . Si le point de fonctionnement change, les équivalents devront être recalculés.

Une fois la forme linéarisée formalisée, il ne reste plus qu'à reprendre les étapes de la résolution linéaire tout en itérant.

1. Résoudre (4-35) en parallèle avec $i_c = \mathbf{0}$ comme dans l'équation (4-3) ;
2. Calculer les équivalents de Thévenin de chaque sous-réseau en parallèle ;
3. Calculer i_c à partir de l'équation (4-6);
4. Mettre à jour des tensions pour chaque sous-réseau à partir de (4-7) en parallèle ;
5. Les nouvelles valeurs des tensions nodales compensées servent à recalculer les équivalents de Norton linéarisés.
6. Si détection d'un changement de segment non-linéaire ou d'un changement d'état d'un interrupteur, l'équation (4-35) est mise à jour et retour à l'étape 1 pour effectuer l'itération suivante.
7. Avancer au prochain pas de temps.

Dans cette résolution non-linéaire, deux cas peuvent se présenter :

- Tous les sous-réseaux sont non-linéaires ;
- Certains sous-réseaux sont linéaires et d'autres non-linéaires.

Dans le premier cas, les résolutions de l'ensemble des sous-réseaux doivent itérer simultanément. Les itérations se terminent lorsque toutes les solutions des sous-réseaux ont convergé. Pour le second cas, les équivalents de Thévenin des réseaux non-linéaires sont constants durant les itérations d'un pas de temps. Il n'est pas besoin de faire participer les sous-réseaux linéaires aux itérations. Ils sont mis en attente jusqu'à la fin des itérations.

Au niveau de l'implémentation de la parallélisation, on retrouve les mêmes éléments que pour la résolution linéaire, comme l'illustre la Figure 4-7 . Ce qui change est l'ajout d'une troisième barrière pour synchroniser les itérations entre les différents sous-réseaux. Un statut sur les itérations (IterFlags) est diffusé pour les stopper (IterFlags=Faux) ou les continuer (IterFlags=Vrai). Une quatrième barrière additionnelle peut être utilisée pour le mixte de sous-réseaux non-linéaires et linéaires (Figure 4-8). Elle permet de mettre en attente les sous-réseaux linéaires qui ne participent pas au processus itératif. A la fin des itérations, cette barrière est levée et les sous-réseaux linéaires réalisent leur superposition des tensions.

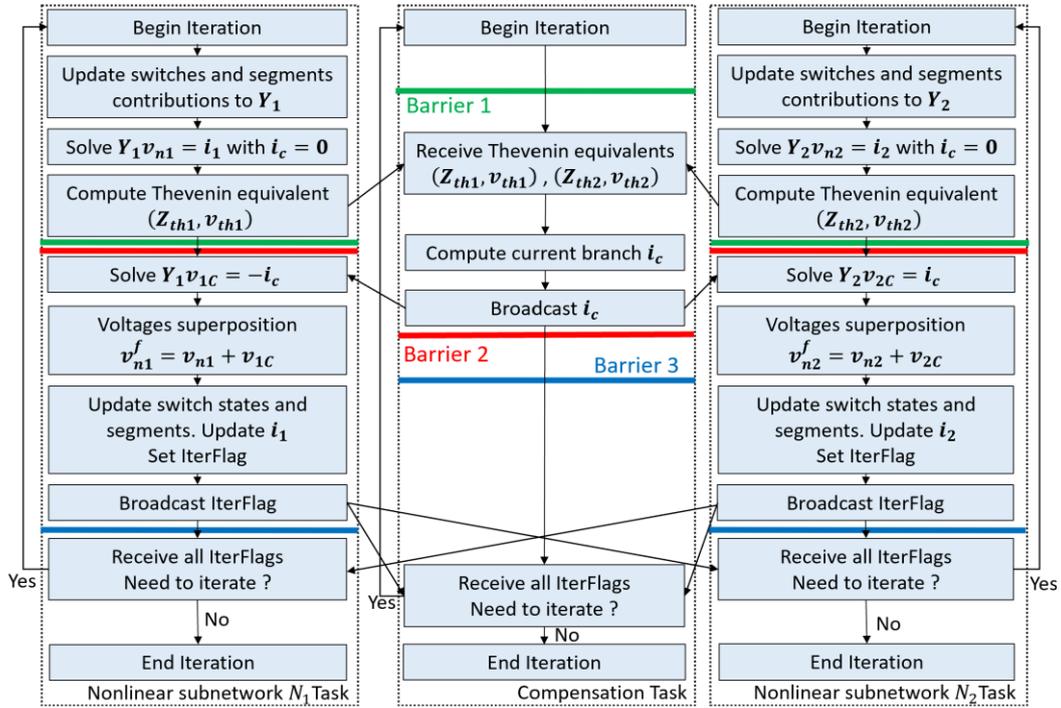


Figure 4-7: Proposition d'un algorithme itératif de Compensation en trois tâches pour deux sous-réseaux non-linéaires.

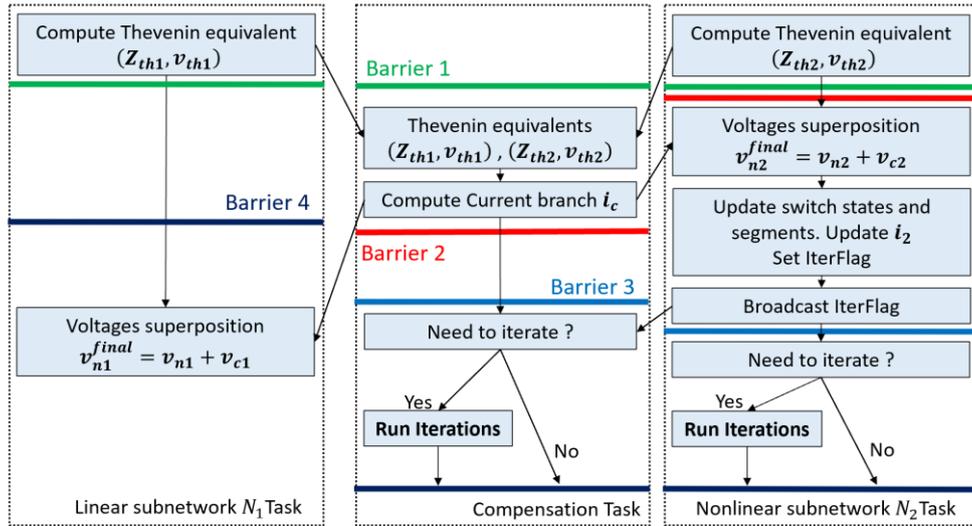


Figure 4-8: Proposition d'un algorithme de Compensation pour un sous-réseau linéaire et un autre non-linéaire.

4.5.3 Multiplicateur de courant idéal avec découplage

Une autre application de la méthode de Compensation est la mise en place d'un multiplicateur de courant. Le principe du multiplicateur de courant est de proposer une amplification du courant d'entrée vis-à-vis du courant de sortie tout en ayant la même tension en entrée et en sortie. La Figure 4-9 ci-dessous présente un multiplicateur de courant idéal entre les nœuds k et m .

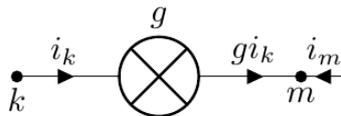


Figure 4-9 : Schéma d'un multiplicateur de courant.

$$\begin{aligned} i_m &= -gi_k \\ v_m &= v_k \end{aligned} \quad (4-34)$$

Cette multiplication de courant n'a pas de sens physique. Cela reviendrait à avoir un amplificateur de puissance sans apport d'énergie. Mais ce composant idéal est très utilisé pour faire des équivalents pour l'électronique de puissance. Un des cas pratiques est la modélisation d'un parc éolien où les g éoliennes sont modélisées en détail. Au lieu de modéliser les g éoliennes, ce qui ralentirait considérablement la simulation, une seule éolienne est représentée en étant connectée avec un multiplicateur de courant de coefficient entier g . Il permet d'avoir la contribution totale du parc en courant sur le réseau.

Avec la formulation MANA, un multiplicateur de courant revient à considérer un transformateur idéal qui garde la même tension en entrée et en sortie. La Figure 4-10 montre cette modélisation en une source de courant et une source de tension.

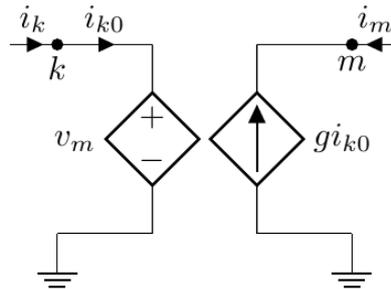


Figure 4-10 : Modélisation d'un multiplicateur de courant en formulation MANA.

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -g \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_k \\ v_m \\ i_{k0} \end{bmatrix} = \begin{bmatrix} i_k \\ i_m \\ v_{k0} \end{bmatrix} \quad (4-35)$$

Avec i_{k0} le courant de branche entre le nœud k et la terre (nœud 0) et v_{k0} la tension de branche imposée entre k et la terre. Or $v_{k0} = v_k$, on déduit le système suivant :

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -g \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_k \\ v_m \\ i_{k0} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} i_k \\ i_m \\ 0 \end{bmatrix} \quad (4-36)$$

En équivalent de Norton augmenté, cela donne :

$$\mathbf{A}_{Nort} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -g \\ 1 & -1 & 0 \end{bmatrix}, \mathbf{b}_{Nort} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4-37)$$

Pour la résolution nodale classique, une admittance fictive Y_f asymétrique doit être utilisée (une résistance de très faible valeur).

$$\begin{bmatrix} Y_f/g & -Y_f/g \\ -Y_f & Y_f \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix} = \begin{bmatrix} i_k \\ i_m \end{bmatrix} \quad (4-38)$$

Pour une analyse nodale classique, la méthode de Compensation peut proposer un multiplicateur de courant idéal sans ajout de résistance fictive mais également découpler le réseau pour la parallélisation. Lors du calcul du courant de branche i_c pour un multiplicateur de courant de coefficient g , il faut considérer g équivalents de Thévenin (Z_{th1}, v_{th1}) identiques en parallèle. On les agrège en un équivalent de Thévenin global ($Z_{th1}/g, v_{th1}$) comme le montre la Figure 4-11 ci-dessous :

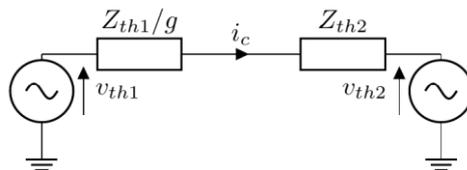


Figure 4-11 : Calcul du courant de compensation pour un multiplicateur de courant.

Une fois que le courant de branche i_c a été calculé, il faut injecter la part de courant qui revient à chaque élément condensé dans l'équivalent égale à $-i_c/g$.

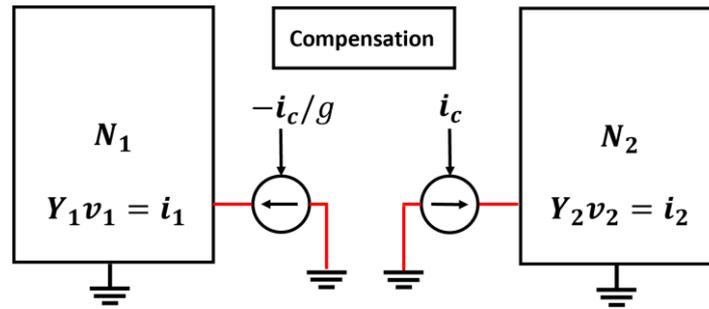


Figure 4-12 : Compensation pour un multiplicateur de courant.

4.6 Performance et Validation

Dans cette section, voici la liste des méthodes de résolution qui seront comparées pour valider la méthode de Compensation.

- SEQ Méthode séquentielle sans parallélisation de la résolution des équations de réseaux.
- LD Parallélisation selon le Délai des Lignes de transmission (Line-Delay).
- DB Ajout de délais virtuels (Delay-Based) via des stublines pour paralléliser.
- CM Méthode de Compensation sans itération pour les réseaux linéaires.

La méthode LD peut être combinée avec DB ou CM dans une parallélisation à deux niveaux. Tout d'abord, les lignes, avec un délai de propagation suffisamment grand, sont identifiées pour un premier découplage. Puis, DB ou CM sert à découpler les sous-réseaux issus de ce premier découplage. Ce type de méthode est noté LD+DB (Délai des lignes + Stubline) ou LD+CM (délai des lignes et compensation). Lorsque les itérations seront activées pour résoudre les non-linéarités, le mot clé Iterative précède le nom de la méthode : Iterative CM, Iterative LD, Iterative LD+CM, Iterative LD+DB...

Pour la comparaison des résultats, l'erreur relative est utilisée. Une mesure scalaire en %, $e\%$, entre la série temporelle \tilde{f} d'une solution et la série temporelle f d'une solution de référence est calculée en utilisant la norme euclidienne :

$$e\% = 100 \times \left(\|\tilde{f} - f\|_2 / \|f\|_2 \right) \quad (4-39)$$

Pour tracer la série temporelle e_t de l'erreur relative entre f et \tilde{f} , l'expression suivante est appliquée pour tout pas de temps t :

$$e_t = |\tilde{f}_t - f_t| / |f_t| \quad (4-40)$$

Afin d'éviter des valeurs nulles ou trop petites au dénominateur (si f s'annule), un maximum est utilisé sur une fenêtre temporelle couvrant une période $T = 1/freq$ où $freq$ est la fréquence du réseau. Pour un signal aperiodique, T couvre l'ensemble du temps de simulation.

$$e_t = |\tilde{f}_t - f_t| / \max_{t' \in [t-T/2, t+T/2]} |f_{t'}| \quad (4-41)$$

Pour les cas tests, les trois architectures de simulations temps réel [59], ci-dessous, sont utilisées en fonction de la réplique du contrôle-commande qui y est connectée. Ces architectures consistent en des PC sous linux adaptés pour la simulation temps réel. L'utilisation d'une version 64 bits de Linux (Arch1) permet la compilation de grands réseaux dont le code généré peut être volumineux.

- Arch1 : OP5031 64 bits sous Linux avec 32 cœurs (2 CPU Intel Xeon E5 3.2 GHz – 16 cœurs).
- Arch2 : OP5031 32 bits sous Linux avec 32 cœurs (2 CPU Intel Xeon E5 3.2 GHz – 16 cœurs).
- Arch3 : OP3031 32 bits sous Linux avec 16 cœurs (2 CPU Intel Xeon E5 3.2 GHz – 8 cœurs).

4.6.1 Réseaux linéaires

Deux cas d'application de réseaux linéaires seront testés pour valider la méthode de Compensation : un réseau de distribution et un réseau à base d'électronique de puissance. Si les matrices nodales de ces réseaux sont refactorisées pour des commutations, des manœuvres ou pour un défaut, il n'est pas nécessaire d'activer les itérations pour ces cas. Le modèle linéarisé de l'interrupteur, présenté au paragraphe 3.2.2, permet d'avoir une bonne précision sans itérations.

4.6.1.1 Réseaux de distribution

Les réseaux de distribution sont des cas d'application où le découplage par la méthode de Compensation est nécessaire pour paralléliser. Cela est dû à des lignes trop courtes ou à l'absence de délai de propagation. Le réseau microgrid GHOST est un cas pratique de distribution avec cette absence de délai naturel. Il est basé sur le cas de référence de Banshee, développé dans [85], qui représente un réseau concret de distribution. Au total, il comporte 663 nœuds. Ce réseau illustre le concept du fonctionnement de Ressources Distribuée d'Énergie (DER) dans la gestion de la puissance au sein de plusieurs feeders de microgrid. Ainsi, quatre types de source de production de puissance sont considérés : le Photovoltaïque (PV), le Stockage de Batteries d'Énergie (BESS), des Générateurs Diesel (DG) et un Combiné Chaleur-Puissance (CHP). Le réseau GHOST est représenté par la Figure 4-13. La connexion au réseau se fait via l'Utility Service.

Pour la simulation, la partie contrôle pour DG et CHP a été découplée de la résolution du réseau. Ainsi, pour la solution SEQ, deux cœurs sont utilisés : un pour les équations du réseau, un autre pour la partie contrôle. CM sépare la solution du réseau en quatre tâches (Procs 1-4) avec un découplage des feeders. Au total, cinq cœurs sont utilisés. Les systèmes de contrôle tournent sur le cinquième cœur. En plus de SEQ et CM, la solution DB est étudiée. Les stublines sont placées aux mêmes endroits que les coupes de la Compensation (faites sur des fils). Le même nombre de cœurs est utilisé.

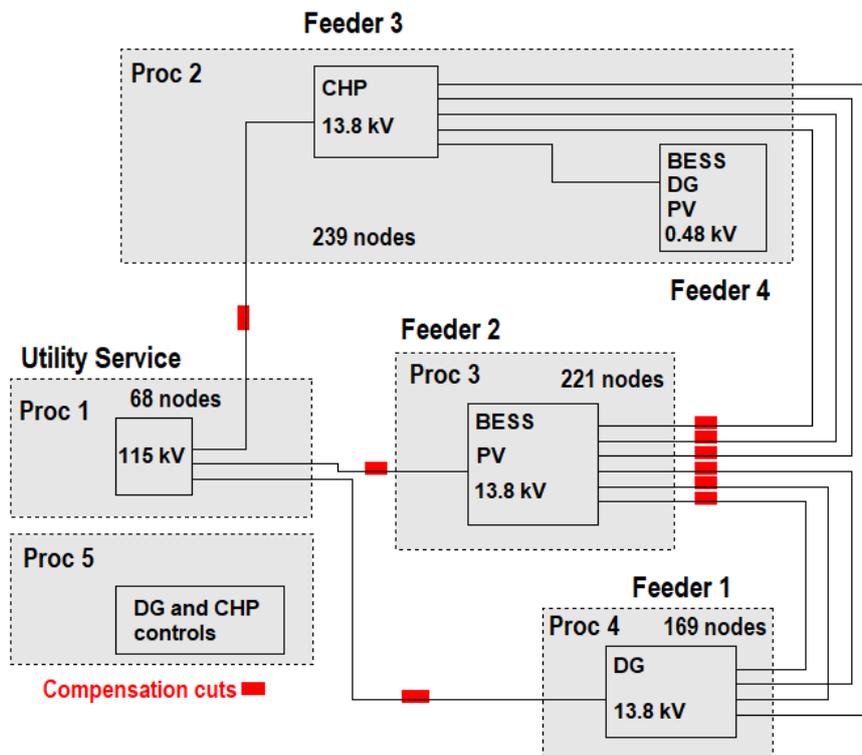


Figure 4-13 : Schéma du découplage par la méthode de Compensation du réseau microgrid GHOST.

Le scénario suivant est simulé pendant 90 s avec un pas de temps de 100 μ s :

- A $t = 0$, tous les DERs sont connectés au réseau.
- A $t = 40$ s, la puissance active du PV du Feeder 4 augmente de 0.2 à 0.8 pu.
- A $t = 59.96$ s, un défaut triphasé phase-terre se manifeste sur la partie réseau (Utility service) pendant une durée de 60 ms.
- A $t = 60$ s, les Feeders 1-3 sont déconnectés du réseau. Ils se mettent à fonctionner en mode isolé.
- A $t = 65$ s, une charge interruptible se déconnecte du Feeder 3 pour ramener la puissance active du CHP à 1 pu et ainsi maintenir l'équilibre offre-demande.

Ce cas est linéaire. La matrice nodale est mise à jour et refactorisée uniquement lors de manœuvres de disjoncteurs (connexions/déconnexions) ou lors du défaut. Le Tableau 4-1 montre les performances en temps différé et temps réel sur l'Arch1. Pour le temps différé, le pas temps moyen $\overline{\Delta t}$ est égal au temps d'exécution de la simulation sur le nombre total de pas de temps. Pour le temps réel, la contrainte temps réel est respectée sans matériel connecté à la simulation. Le pas de temps minimum sans overrun est affiché pour le régime permanent $\overline{\Delta t_{RT}}$ et lors des transitoires $\widetilde{\Delta t_{RT}}$. Le speedup ratio s_{RT} de $\widetilde{\Delta t_{RT}}$ par rapport à SEQ montre les performances de DB et CM lors des transitoires de la simulation temps réel. L'erreur relative $e\%$ par rapport à SEQ est affichée dans la dernière colonne.

Tableau 4-1 : Comparaison des performances et de la précision des méthodes SEQ, DB et CM pour la résolution du réseau microgrid GHOST.

Méthodes	Nombre de cœurs	$\overline{\Delta t}$	$\overline{\Delta t_{RT}}$	$\widetilde{\Delta t_{RT}}$	s_{RT}	$e\%$
SEQ	2	65 μ s	68 μ s	246 μ s	-	-
DB	5	24 μ s	29 μ s	46 μ s	5.35	2.35
CM	5	32 μ s	38s μ s	115 μ	2.14	5.4 x 10e-4

Ces résultats démontrent que CM est une méthode de parallélisation efficace pour les réseaux de distribution. La simulation est accélérée en abaissant le pas de temps admissible pour la simulation temps réel. Une bienne meilleure précision est obtenue en comparaison des délais artificiels de DB car la résolution des équations de réseaux par CM est simultanée. Cela se vérifie sur la superposition des courbes pour la tension RMS et la puissance réactive du Feeder 2.

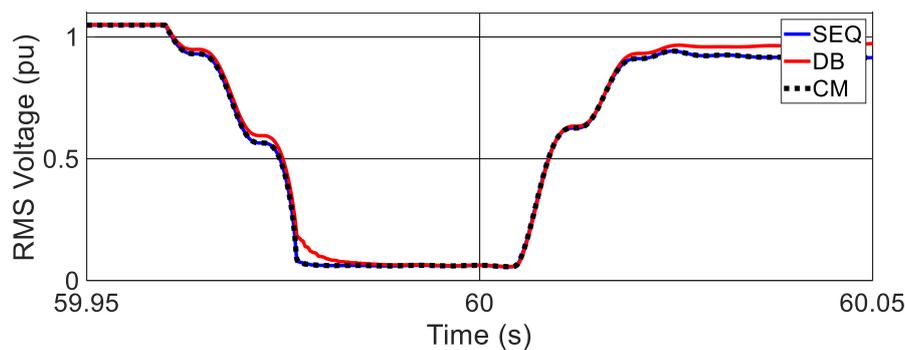


Figure 4-14 : Tension RMS du PV au Feeder 2.

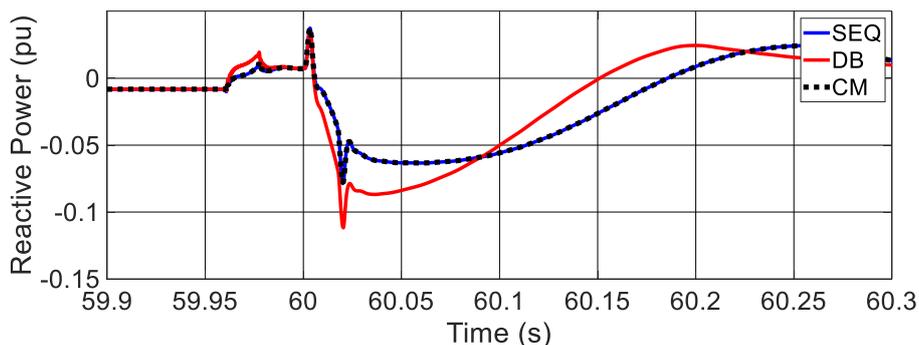


Figure 4-15 : Puissance réactive du PV au Feeder 2.

4.6.1.2 Réseaux HVDC avec commutations

Les réseaux à base d'électronique de puissance sont d'autres cas d'application de CM où les lignes de transmission ne permettent pas un découplage du réseau. L'exemple est donné par l'interconnexion HVDC IFA2000 [23], en service depuis 1986, entre la France (poste électrique Les Mandarins) et l'Angleterre (poste de Sellindge). Elle est composée de deux bipôles LCC de 1000 MW chacun. La tension opérée est fixée à +/- 272 kV. Pour les simulations EMT, ce réseau est considéré comme un réseau de commutations où les thyristors commutent plusieurs fois pour la conversion AC/DC. Comme décrit par la Figure 4-16, chaque pôle DC est représenté par deux ponts à six impulsions. Des modèles de câble à paramètres constants sont utilisés pour représenter le câble DC long de 73 km. Les filtres, nécessaires pour une liaison LCC, sont également modélisés en détail. Pour le système de contrôle, un modèle est importé depuis Matlab/Simulink [6] via une interface EMT-Simulink [7]. Les réseaux AC 400 KV aux extrémités de la liaison sont représentés par des équivalents Thévenin de court-circuit.

Pour cet exemple, un parallélisme à deux niveaux est utilisé (méthode LD+CM). Le délai naturel de propagation du câble DC permet de séparer la solution des deux stations de conversion. Ainsi, la méthode LD crée six tâches avec la résolution des lignes comprise. Cependant, l'électronique de puissance des stations de conversion ne peut pas être découplée avec une méthode avec délai. La méthode CM est appliquée après le découplage LD pour séparer la solution de chaque station en deux tâches comme le montre la Figure 4-16. Le découplage est dénommé pole-based LD+CM. Comme la solution de chaque station est indépendante, deux tâches de compensation sont créées pour calculer le courant de compensation correspondant. Au total six cœurs sont utilisés pour simuler le découplage pole-based LD+CM contre quatre cœurs pour le découplage LD sans compensation.

Le découplage CM peut être poussé un peu plus loin dans une séparation de chaque pont composant un pôle. Cette séparation bridge-based LD+CM est illustrée par la Figure 4-17. Au total, dix cœurs seront utilisés pour la simulation de ce découplage.

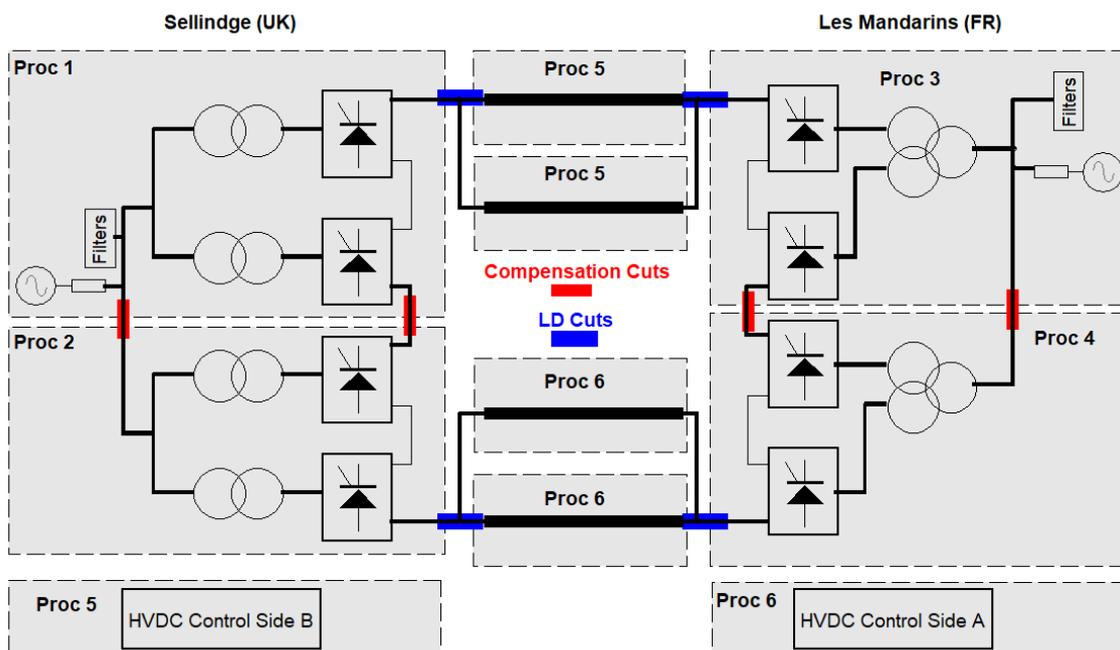


Figure 4-16 : Séparation en tâches d'IFA2000 pour le découplage pole-based LD+CM.

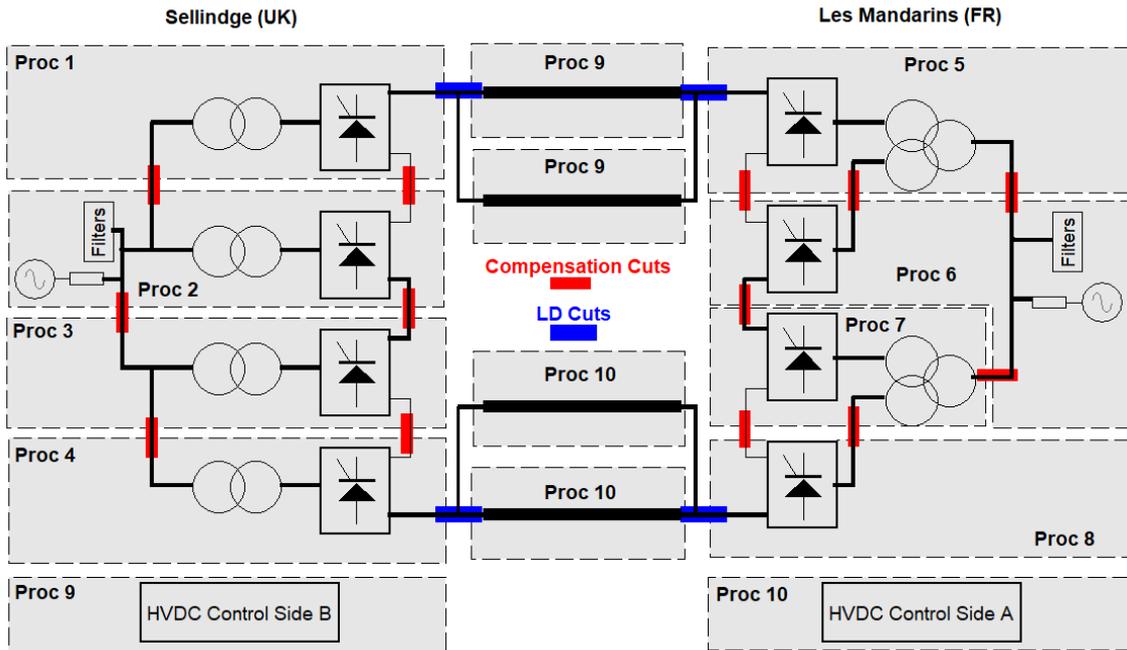


Figure 4-17 : Séparation en tâches d'IFA2000 pour le découplage bridge-based LD+CM.

Au début de la simulation, le découplage basé sur les ponts se révèle être beaucoup plus instable que celui basé uniquement sur les pôles au niveau de la station de Sellindge. En effet, la matrice nodale de Sellindge présente un conditionnement de 10^{23} beaucoup plus important que celle des Mandarins (10^{10}). Le découplage par Compensation va augmenter le conditionnement des matrices de ces sous-réseaux. Pour résoudre ce problème, deux résistances de très faible valeur ($10^{-10} \Omega$) sont insérées au niveau des coupes de la compensation côté Sellindge (Proc 2 et Proc 3). Aucun problème de stabilité n'est constaté pour le découplage par pôles. Ceci montre une limite du découplage par compensation. Les sous-réseaux résultants du découplage peuvent être numériquement plus instables que le réseau initial.

La séquence de démarrage est simulée pendant 10 s avec un pas de temps de $30 \mu\text{s}$. Le transit de puissance est fixé à 200 MW des Mandarins (FR) vers Sellindge (UK). L'architecture de simulation est l'Arch2. La simulation est d'abord lancée en temps différé pour vérifier la précision du découplage de la Compensation. Pour les deux types de découplage LD+CM par pont et par pôle, une bonne précision est obtenue par rapport à la méthode LD. L'erreur relative reste faible pour la tension DC, $e\% = 0.17$ pour le découplage CM par pont et $e\% = 0.18$ pour celui par pôles (voir aussi la Figure 4-18).

Les performances sont mesurées lors d'une simulation temps réel Software-In-the-Loop (SIL) n'incluant pas de matériel dans la boucle de simulation. Le temps d'exécution à chaque pas temps est ainsi tracé dans la Figure 4-19 pour les différentes méthodes de découplage pour le cœur simulation le plus chargé en tâches de calcul. La méthode LD ne permet pas de faire respecter la contrainte temps réel comme en témoignent les nombreux overruns (temps d'exécution au-delà de $30 \mu\text{s}$). Le découplage par pont LD+CM est plus performant que celui par pôles de 15% et de 47% par rapport au découplage LD. Ces gains de performance sont observés lorsque le temps d'exécution augmente après le démarrage de la liaison. La méthode CM est plus performante avec des refactorisations répétitives dues aux commutations. Comme les matrices des sous-réseaux issues du découplage CM sont de plus petites dimensions, la refactorisation est beaucoup plus rapide.

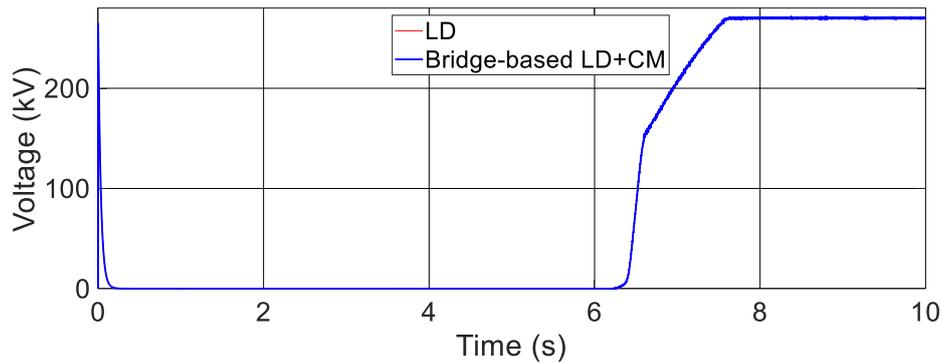


Figure 4-18 : Tension DC pendant la phase de démarrage pour les méthodes de découplage LD et bridge LD+CM.

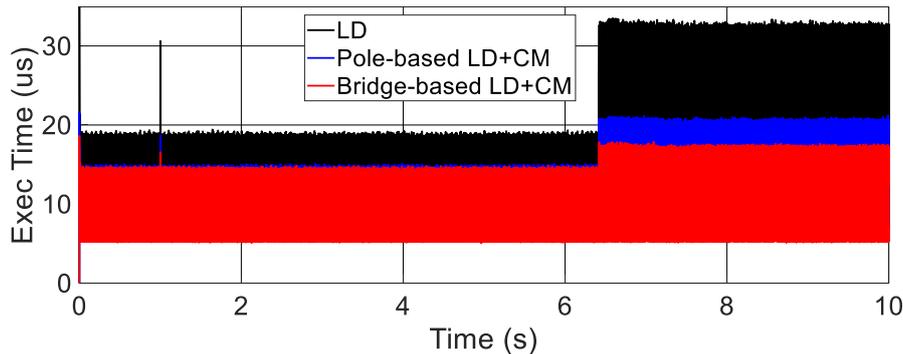


Figure 4-19 : Temps d'exécution du processeur le plus lent pour chaque pas de temps des méthodes LD, pole-based LD+CM et bridge-based LD+CM.

4.6.2 Réseaux non-linéaires

Contrairement aux exemples précédents, les cas tests vont nécessiter l'utilisation d'itérations pour la résolution des éléments non-linéaires. L'intérêt des itérations y est constaté pour deux types de non-linéarité : le changement d'état de diodes et la caractéristique linéarisée par segments d'un transformateur saturable. La plupart des exemples sont tirés d'applications industrielles d'études de réseau qui utilisent la simulation HIL avec les répliques de contrôle-commande du laboratoire temps réel du Campus Transfo de RTE.

4.6.2.1 Réseaux non-linéaires avec commutations

4.6.2.1.1 Réseau parc éolien et multiplicateur de courant

Un réseau simple qui bénéficie des itérations est le raccordement d'un parc éolien de 7,5 MW à un réseau de distribution de 25 kV. Le parc est composé de 5 éoliennes de 1,5 MW qui utilisent un convertisseur VSC back-to-back à deux niveaux pour se connecter au réseau. Le modèle d'éolienne à générateur asynchrone à double alimentation (DFIG) est de type III [86]. Les itérations sont nécessaires pour la résolution des états des diodes du convertisseur. C'est un cas d'application pour la compensation car la ligne en PI ne permet pas de découpler le parc du réseau. On découple à l'entrée du parc avec la compensation (Figure 4-20).

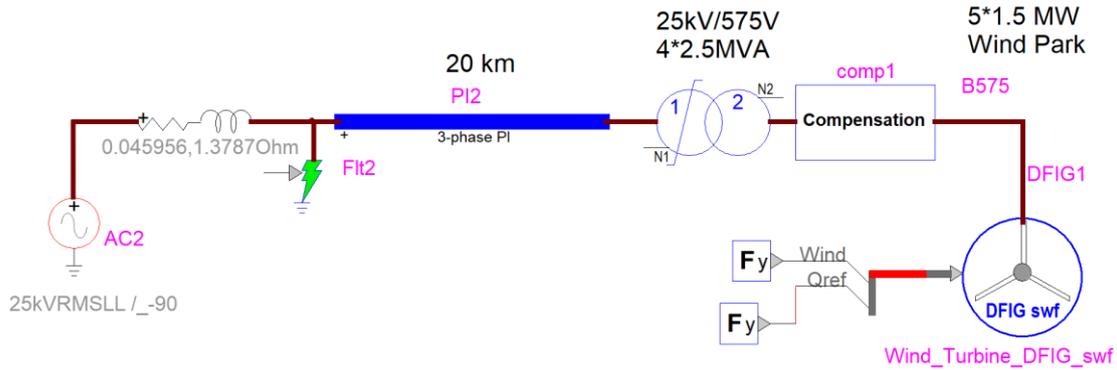


Figure 4-20 : Schéma d'un parc éolien DFIG connecté à un réseau de distribution de 25 kV.

On enregistre 1 s de simulation avec un pas de temps de 50 μ s. Un défaut franc (0.001 ohm) triphasé à l'entrée de la ligne a lieu à $t=0,5$ s et dure 100 ms. Les réponses (tension RMS mesurées au terminal du parc) des méthodes SEQ, Iterative SEQ et Iterative CM sont tracées à la Figure 4-21. On remarque une bonne superposition des courbes qui est vérifiée par une erreur relative très faible de l'ordre de 10^{-7} (Figure 4-22). Les itérations pour ce cas sont nécessaires pour éviter des oscillations numériques dues aux changements d'état des diodes.

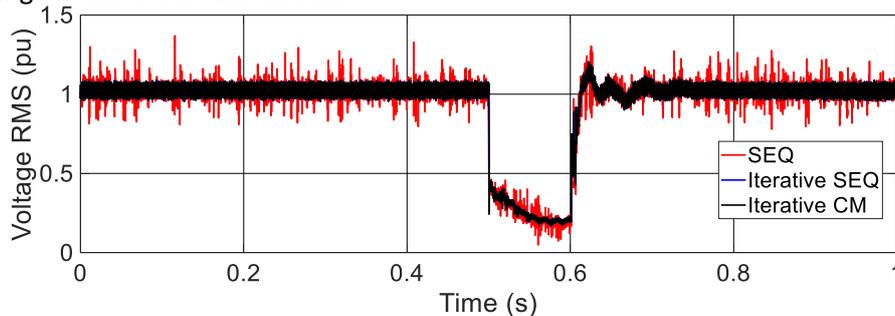


Figure 4-21 : Tension RMS du parc pour SEQ, Iterative SEQ et Iterative CM.

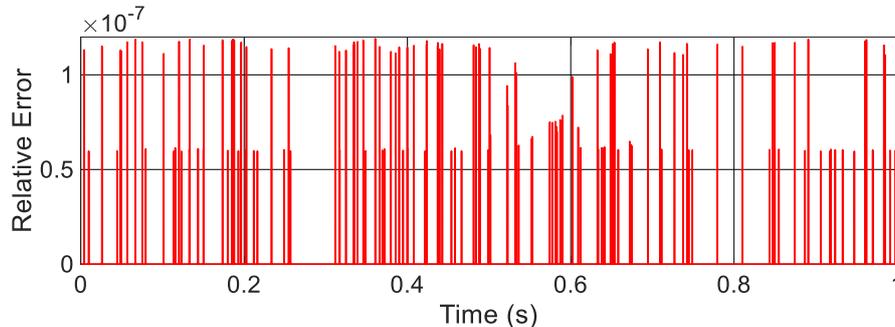


Figure 4-22 : Erreur relative de la tension RMS du parc d'Iterative CM par rapport à Iterative SEQ.

Iterative DB est également comparée avec Iterative SEQ en insérant une stubline au même endroit que le découplage de la compensation. Les valeurs R-L ont été prises sur le secondaire du transformateur pour la stubline. On constate que l'ajout d'une capacité fictive, afin de paralléliser la simulation, a un impact sur la simulation en détériorant la précision vis-à-vis d'une solution simultanée (Iterative CM).

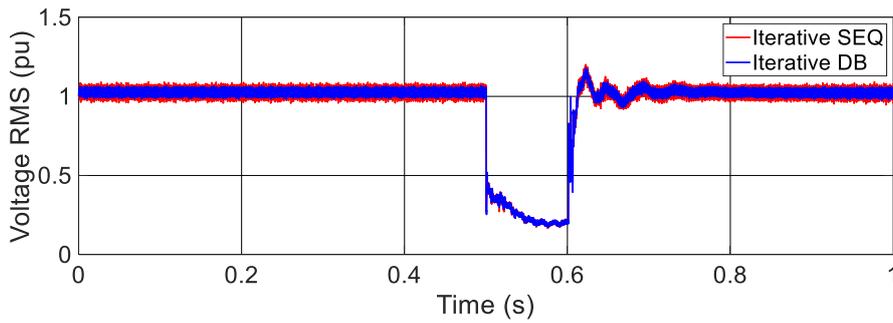


Figure 4-23 : Tension RMS du parc pour Iterative SEQ et Iterative DB.

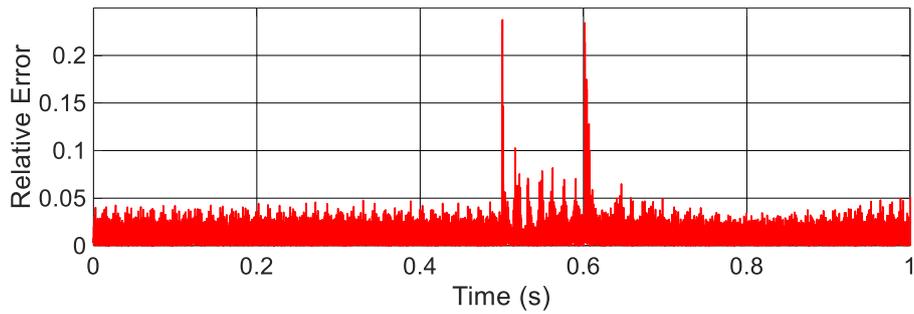


Figure 4-24 : Erreur relative de la tension RMS du parc de Iterative DB par rapport à Iterative SEQ.

Enfin, le multiplicateur de courant idéal découpleur via CM est testé. D'un côté, SEQ simule deux parcs contenant deux éoliennes chacun reliés au même point de raccordement, de l'autre, CM découple le réseau en tant que multiplicateur de courant de coefficient $g = 2$ relié à un seul parc de deux éoliennes (Figure 4-25).

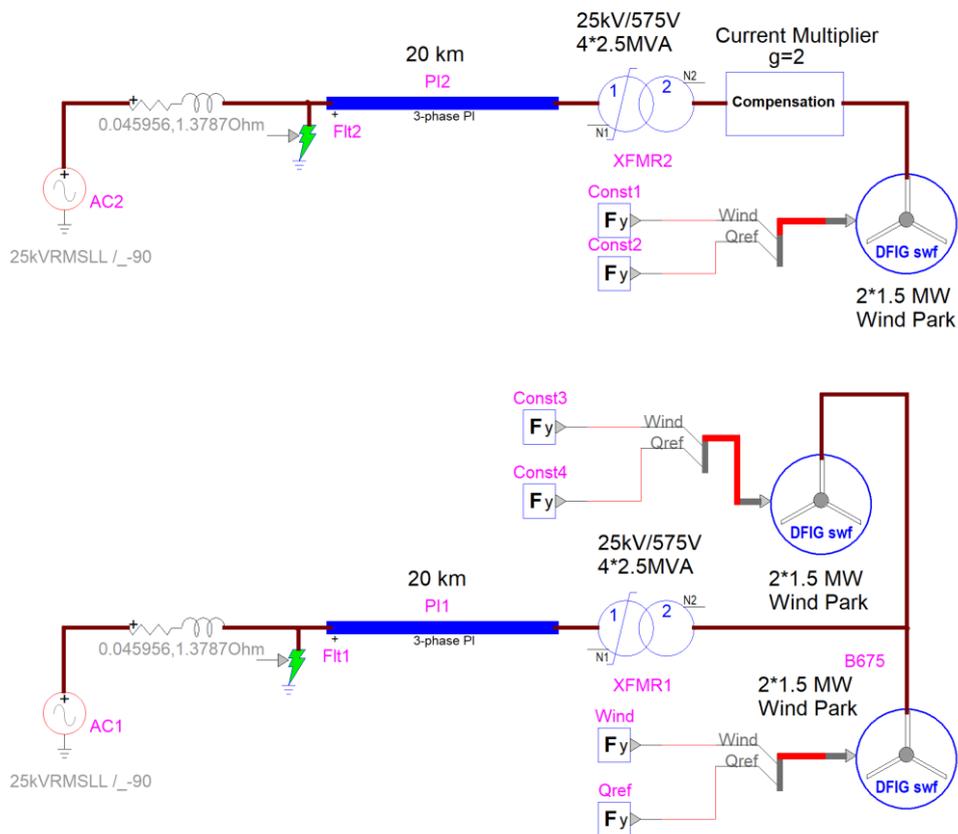


Figure 4-25 : Compensation en multiplicateur de courant pour agréger deux parcs.

Une précision similaire est obtenue entre Iterative SEQ et Iterative CM en tant que multiplicateur de courant (Figure 4-26 et Figure 4-27). Ceci valide l'équivalent de réseau ainsi construit.

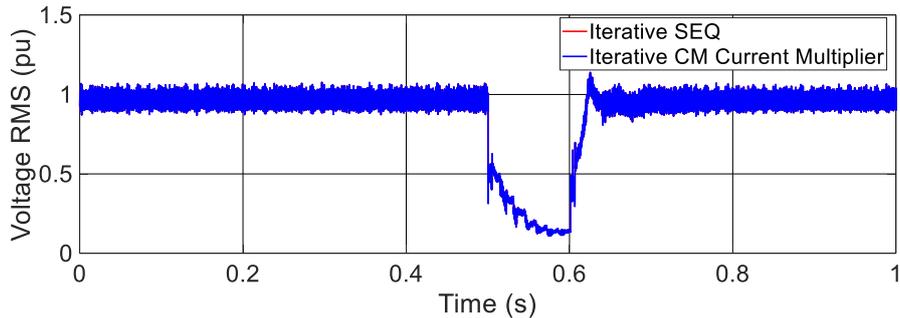


Figure 4-26 : Tension RMS des parcs pour Iterative SEQ et Iterative CM en multiplicateur de courant.

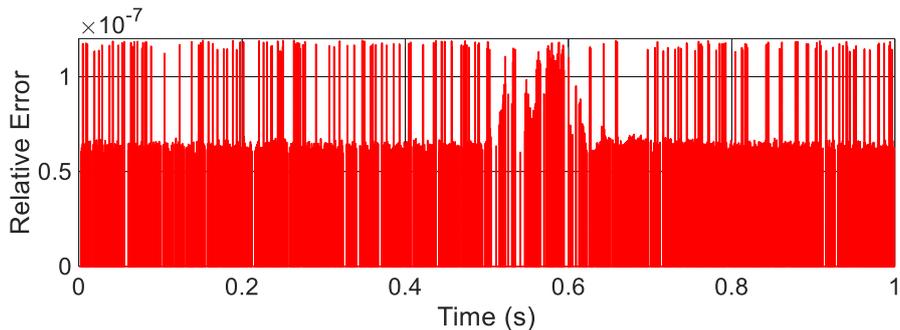


Figure 4-27 : Erreur relative entre la tension RMS des parcs d'Iterative CM en multiplicateur de courant par rapport à Iterative SEQ.

4.6.2.1.2 Liaison HVDC de type VSC

4.6.2.1.2.1 Cas Eleclink

Eleclink est un projet de technologie VSC-HVDC d'interconnexion [87] entre la France (poste Peuplingues) et l'Angleterre (poste Cheriton). Il est composé d'un monopôle symétrique de 1000 MW basé sur la technologie Modular Multilevel Converteur (MMC) de Siemens. La tension DC de fonctionnement est fixée à ± 320 kV. Les mêmes modèles de Convertisseur présentés dans [88] sont utilisés pour les simulations temps réel et temps différé. Un contrôle générique de station de conversion est importé de Matlab/Simulink [6] via une interface EMT-Simulink [7]. Un modèle Wideband de câble [27] est utilisé pour représenter le câble DC. Des équivalents de court-circuit de Thévenin modélisent les réseaux 400 kV AC à l'extrémité de la liaison. Le pas de temps est fixé à 10 μ s. Les itérations sont activées pour ce cas afin de résoudre les modèles de diode non-linéaires des IGBT. La séquence de démarrage est simulée en temps différé pendant une durée de 20 s sur l'Arch2. Le transit de puissance est de 500 MW. Dans un parallélisme à deux niveaux, LD découple chaque station de conversion via le câble DC et CM est utilisée pour découpler les réseaux AC des stations comme le montre la Figure 4-28. Sept cœurs sont utilisés par LD+CM contre cinq cœurs pour LD seule. Tous les sous-réseaux issus de la compensation itèrent comme dans la Figure 4-7.

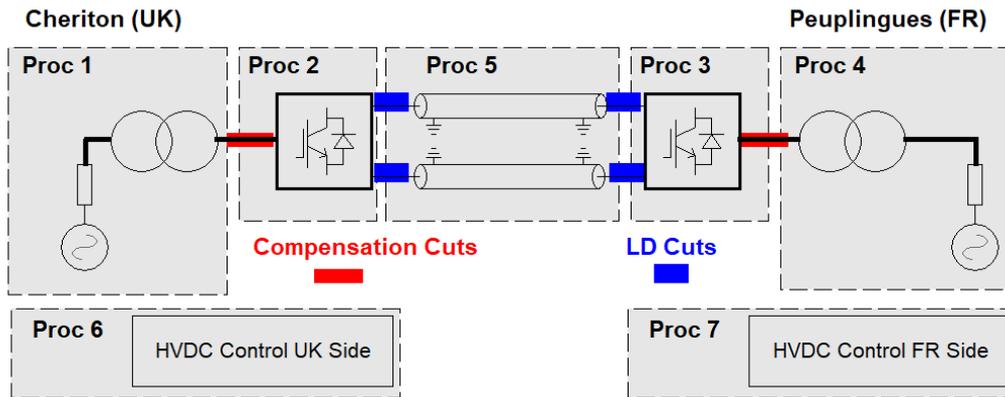


Figure 4-28: Séparation en tâches d'iterative LD+CM pour la simulation temps réel SIL de la liaison HVDC Elelink.

Les résultats montrent que les itérations sont bien nécessaires pour éliminer des oscillations numériques pendant la phase de démarrage (Figure 4-29). Ces oscillations viennent des erreurs de commutation des diodes incluses dans le modèle des IGBT. Le même problème est observé sans compensation. Iterative LD+CM donne une bonne précision (Figure 4-29, $e\%_{\text{Iterative LD+CM}} = 0.013$) pour les commutations non-linéaires tout en améliorant les performances de 40 %. En temps réel, cette méthode tient la contrainte temporelle avec un pas de temps de 10 μs alors que des overruns arrivent continuellement pour Iterative LD.

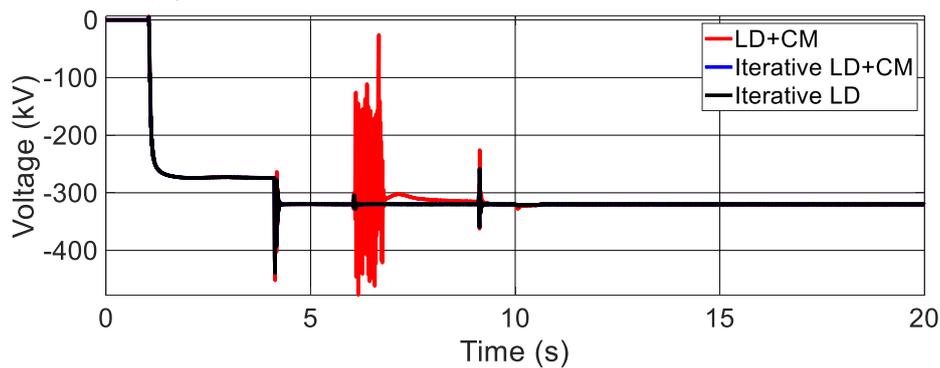


Figure 4-29 : Tension DC pendant la séquence de démarrage pour LD+CM, Iterative LD+CM et Iterative LD.

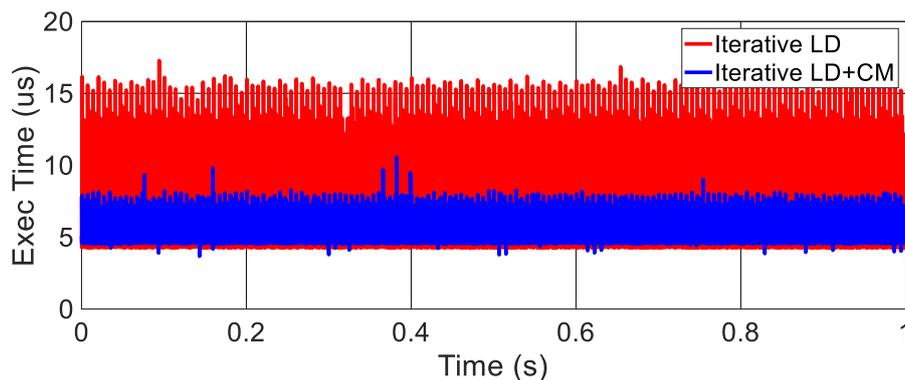


Figure 4-30 : Temps d'exécution du processeur le plus lent à chaque pas de temps pour Iterative LD et Iterative LD+CM.

Iterative LD+CM est ensuite testée en temps réel avec une plateforme HIL. Le contrôle générique sous Simulink est remplacé par une réplique des contrôleurs et des protections de la station (Figure 4-31).

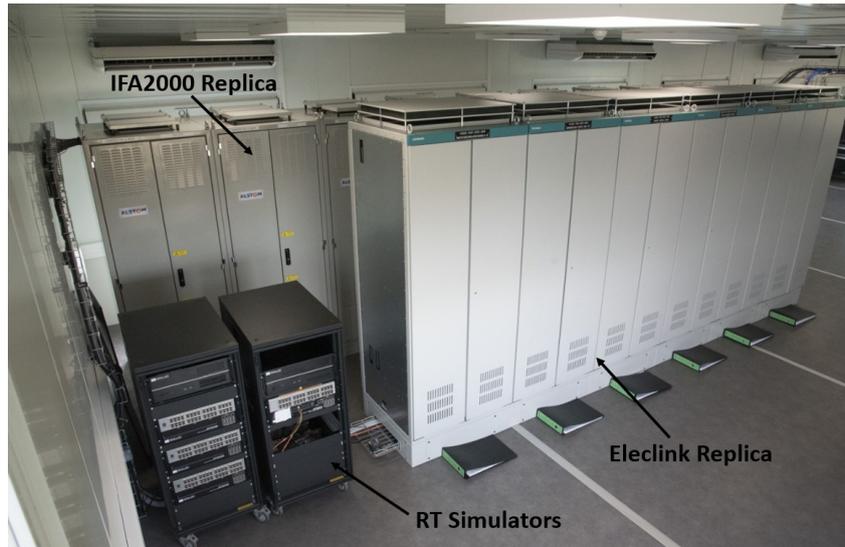


Figure 4-31 : Répliques du contrôle-commande des liaisons IFA2000 et Eleclink dans le laboratoire temps réel de RTE au Campus Transfo.

Des boîtiers d'IO (Entrées/Sorties) permettent l'interface entre le simulateur (Arch3) et la réplique du contrôle. Dans la Figure 4-32 ci-dessous, Iterative LD+CM a besoin de cinq cœurs, un pour chaque sous-station et un autre pour les IOs et le câble DC, alors qu'Iterative LD en requiert trois. Un défaut phase-terre côté français de 150 ms est simulé au transit maximal de puissance (1000 MW). Le pas de temps est fixé à 30 μ s.

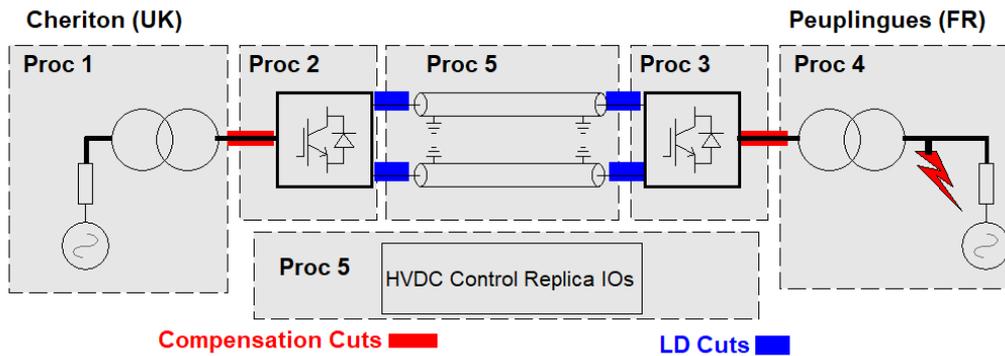


Figure 4-32 Séparation en tâches d'Iterative LD+CM pour la simulation temps réel HIL de la liaison HVDC Eleclink.

Les figures 4-33 et 4-34 montrent qu'Iterative LD+CM a permis d'améliorer les performances de 40% tout en ayant une réponse similaire sur la tension DC.

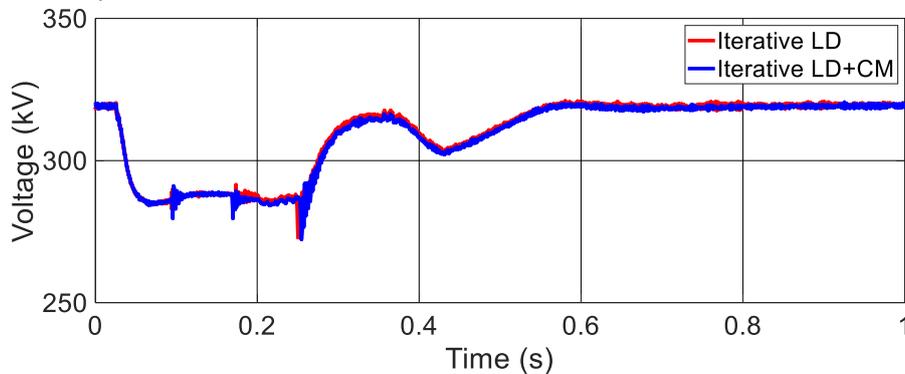


Figure 4-33 : Tension DC pour Iterative LD et Iterative LD+CM.

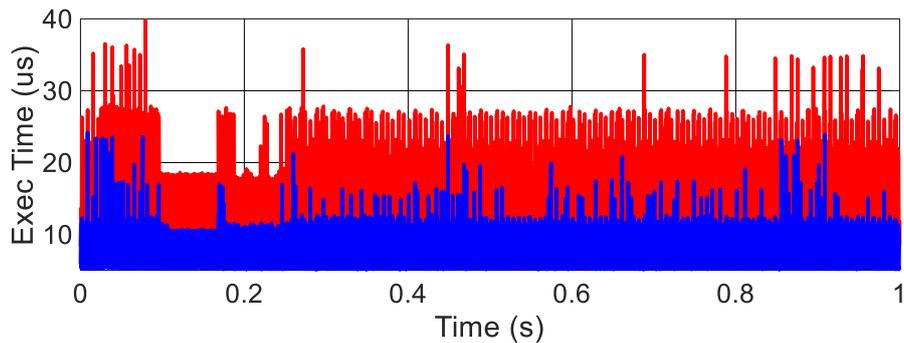


Figure 4-34 : Temps d'exécution pour le processeur le plus lent pour Iterative LD et Iterative LD+CM.

4.6.2.1.2.2 Etude d'interaction entre un CSPR et une liaison VSC-HVDC

La multiplication des équipements d'électronique de puissance peut avoir un impact sur la stabilité du réseau. Des études d'interaction sont réalisées pour évaluer les risques associés [89]. Afin d'avoir une bonne précision des phénomènes transitoires étudiés, la simulation EMT temps réel [21][90] est utilisée pour les études en intégrant dans la boucle de simulation une réplique des contrôleurs installés sur site. Un cas d'étude d'interaction est celui concernant la liaison VSC-HVDC IFA2 [91] entre la France (poste de Tourbe) et l'Angleterre (poste de Daedalus) et le CSPR (Compensateur Statique de Puissance Réactive) de Tourbe [92]. En effet, ces deux équipements d'électronique de puissance (Station de conversion et CSPR) sont situés sur le même poste électrique. Au vu de cette proximité, des éventuelles interactions doivent être étudiées. C'est un cas pratique pour la méthode de Compensation qui permet au poste de Tourbe un découplage de la solution de réseau avec une bonne précision. Sans découplage, la simulation ne serait pas en mesure de tenir le temps réel.

Le CSPR est localisé dans la partie 225 kV du poste électrique de Tourbe. Il a été mis en service en 2012. La puissance réactive de fonctionnement va de -250 MVar en capacitif à +100 MVar en inductif. Comme décrit dans la Figure 4-35 : Schéma détaillé du CSPR de Tourbe., on retrouve les éléments classiques d'un CSPR :

- Un transformateur principal de puissance ;
- Une inductance contrôlée par thyristors, Thyristor Controlled Reactor (TCR) ;
- Une capacité contrôlée par disjoncteur, Thyristor Switched Capacitor (TSC) ;
- Des filtres harmoniques pour le rang 5 ;
- Une capacité contrôlée par disjoncteur, Mechanically Switched Capacitor (MSC), qui permet d'étendre la plage de puissance réactive avec un temps de réponse plus lent.

Une modélisation détaillée est utilisée pour ces différents éléments ainsi qu'un contrôle générique qui a été adapté pour le CSPR de Tourbe.

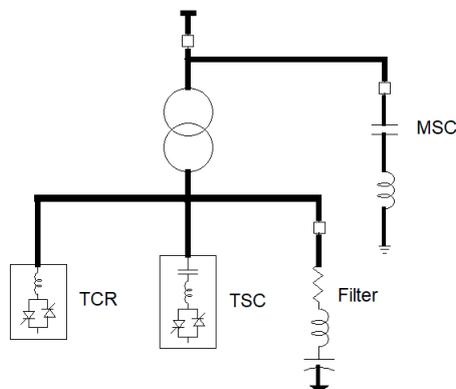


Figure 4-35 : Schéma détaillé du CSPR de Tourbe.

La liaison IFA2 a été mise en service en 2021. Elle est composée d'un monopôle symétrique de 1000 MW basé sur la technologie à deux niveaux en cascade (CTL) d'ABB. La tension DC de fonctionnement est fixée à ± 320 kV. Le modèle pour les stations de conversion est issu de [93]. Il tourne en temps réel sur une architecture CPU-FPGA. Sur FPGA, le pas de temps est fixé à $1 \mu\text{s}$. Les itérations sont nécessaires comme pour le cas précédent d'Eleclink pour résoudre les équations associées aux diodes non-linéaires incluses dans le modèle des IGBT. Chaque station de conversion est contrôlée, via des IOs du simulateur (Arch2), par la réplique du contrôleur et des protections de la liaison (Figure 4-36). Les câbles DC, longs de 228 km, sont représentés par des modèles Wideband [27] qui incluent une dépendance fréquentielle des paramètres. Le délai de propagation des câbles permet de découpler chaque station de conversion et de les résoudre sur des cœurs séparés (coupe LD sur la Figure 4-37). Les réseaux AC 400 KV sont représentés par des équivalents de Thévenin de court-circuit pour une simulation de IFA2 uniquement. Pour cette simulation, la méthode de compensation peut être utilisée comme pour Eleclink pour découpler le réseau AC de la station de conversion. Les IOs correspondantes à chaque sous-réseau sont associées au même cœur pour garder la même précision que LD en évitant d'introduire des délais supplémentaires. Au total, Iterative LD+CM utilise six cœurs, un pour le câble DC, quatre pour la résolution des sous-réseaux issus du découplage et un pour les IOs du FPGA. Le transit est fixé à 1000 MW de Tourbe vers Daedalus. Un défaut phase-terre de 200 ms est simulé côté français. Le pas de temps est de $30 \mu\text{s}$.



Figure 4-36 : Aperçu de la réplique du contrôle-commande de la liaison HVDC IFA2 dans le laboratoire temps réel du Campus Transfo de RTE.

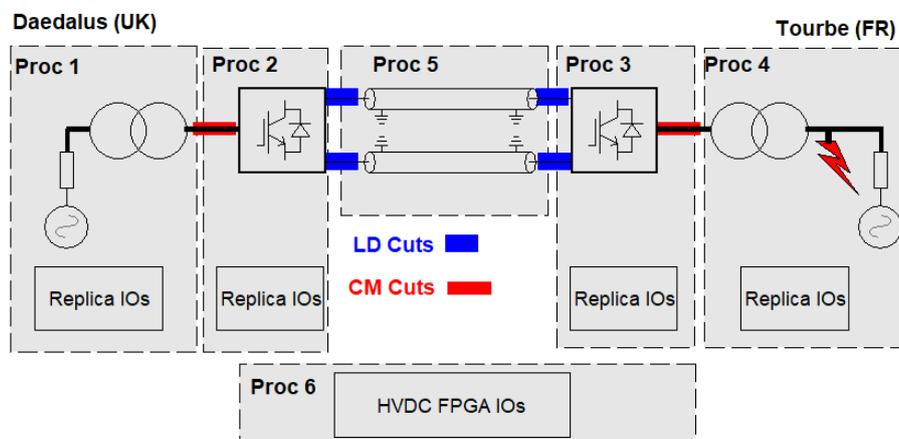


Figure 4-37 : Séparation en tâches d'Iterative LD+CM pour la simulation HIL temps réel de la liaison HVDC IFA2.

La Figure 4-38 montre une bonne superposition de la tension DC pour les deux méthodes, Iterative LD et Iterative LD+CM. Par ailleurs, les performances ont été augmentées de 25%. Ainsi, pour IFA2 on peut abaisser le pas de temps à 30 μ s grâce à la compensation.

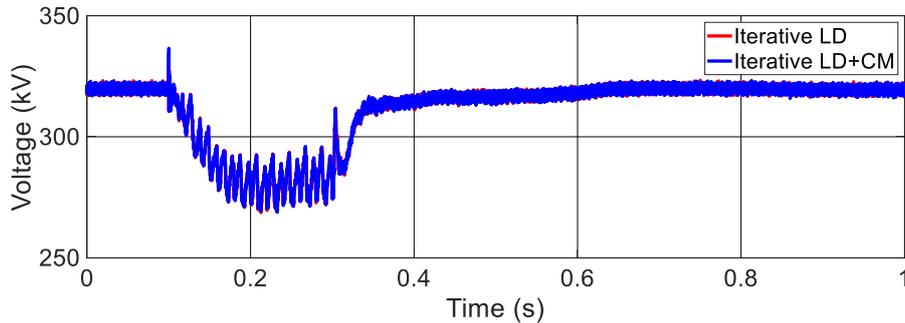


Figure 4-38 : Tension DC pour Iterative LD et Iterative LD+CM.

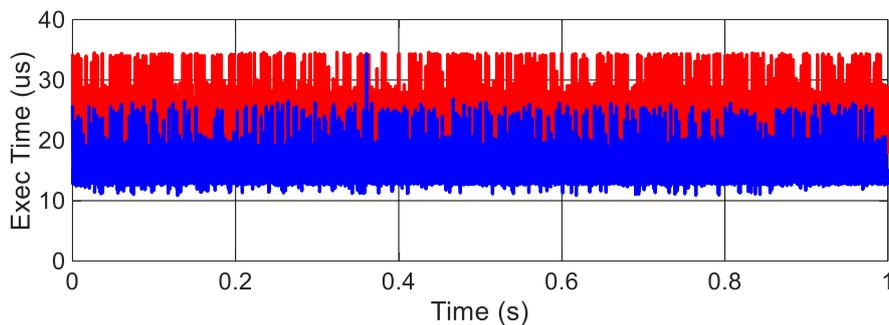


Figure 4-39 : Temps d'exécution pour le processeur le plus lent à chaque pas de temps pour Iterative LD et Iterative LD+CM.

Le système complet, CSPR + Liaison HVDC, pour l'étude de l'interaction entre le CSPR et la liaison VSC-HVDC est représenté sur la figure 23. Le réseau AC environnant est composé d'une zone de production de 400 kV autour du groupe nucléaire de Flamanville et d'une zone de consommation tout autour de la ville de Caen. Le reste du réseau 400 kV/225 kV est représenté par quatre équivalents de Thévenin de court-circuit. Les données de réseau ont été directement importées à partir d'un outil de planification de réseau [55]. La situation de réseau choisie est un creux d'été où la puissance de court-circuit est minimale. Le délai de propagation des lignes de transport permet de découpler le réseau en quatre sous-réseaux. Avec cinq cœurs utilisés par Iterative LD, la simulation est trop lente pour le temps réel. Le temps d'exécution pendant un pas de temps est de 122 μ s pour Proc 2. Cela s'explique par le fait que le sous-réseau de Proc 2 contient à la fois le modèle du CSPR et une station de conversion d'IFA2. Aucun découplage via un délai naturel n'est possible dans le poste de Tourbe.

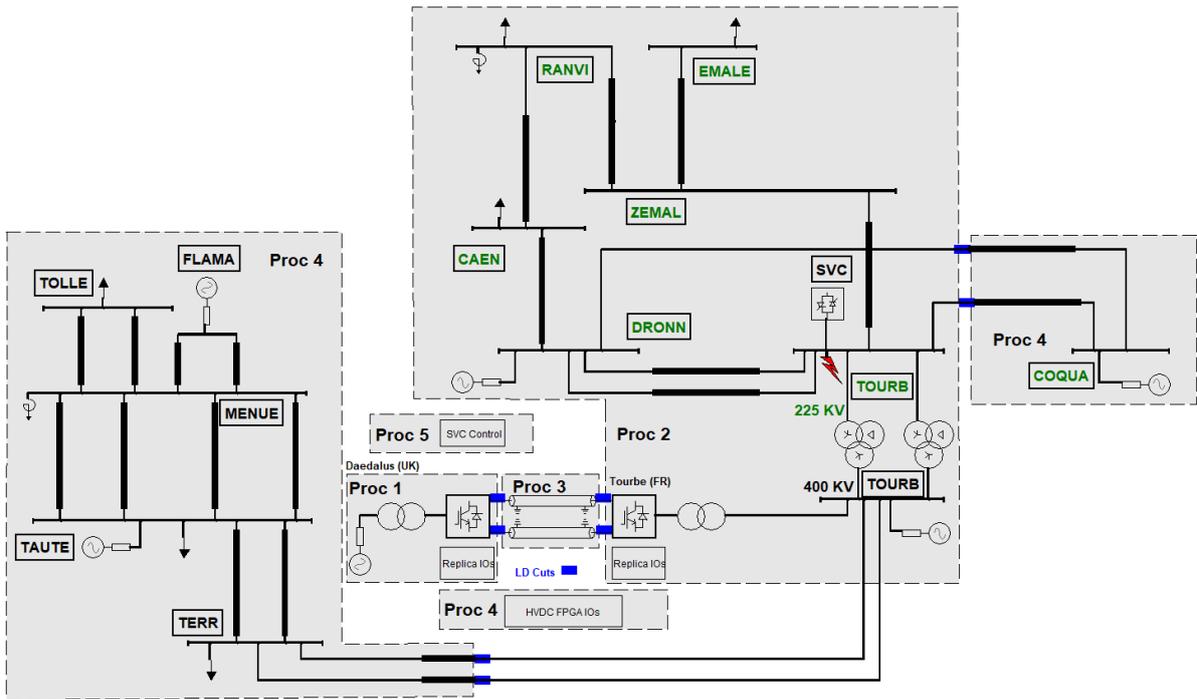


Figure 4-40 : Séparation en tâches d'Iterative LD du système complet pour l'étude d'interaction.

Pour remédier à ce problème, il faut utiliser la méthode de compensation pour découpler le poste de Tourbe. Au final, quatre coupes de compensation permettent de découpler le Proc 2 en quatre sous-réseaux comme le montre la Figure 4-41. Avec ce découplage (Iterative LD+CM), la simulation tourne en temps réel avec un pas de temps de 45 μ s en utilisant huit cœurs. Le pas temps est plus important que pour le cas d'IFA2 seule. Cela peut s'expliquer par l'augmentation du nombre de coupes de compensation (trois au lieu d'une). Le temps de calcul des courants de compensation augmente et in fine les tâches des sous-réseaux issus de CM sont ralenties.

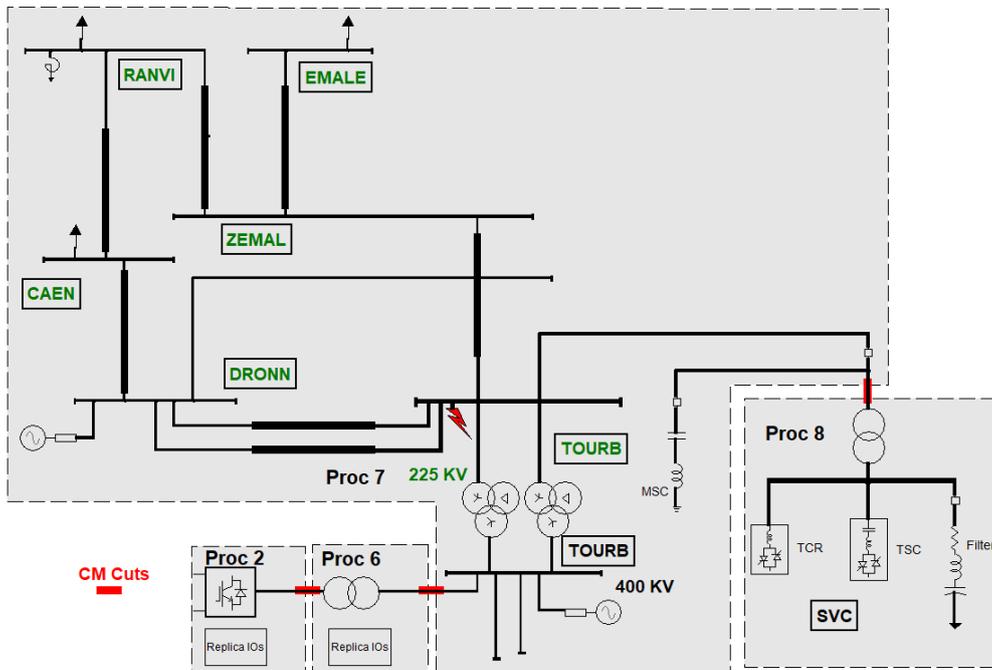


Figure 4-41 : Découplage par la méthode de Compensation (Iterative LD+CM) du Proc 2.

Mise à part CM, on peut tester la technique basée sur le délai artificiel d'une stubline, Iterative LD+DB. La Figure 4-42 montre la localisation de la ligne fictive ajoutée pour simuler sur six cœurs. C'est le meilleur emplacement pour optimiser les performances. Les valeurs RL du transformateur de la liaison d'IFA2 aurait pu être utilisées. Cependant, cela aurait ralenti la simulation en incluant les communications d'IOs dans la tâche de Tourbe. Pour comparer les deux méthodes, Iterative LD+CM et Iterative LD+DB, un défaut triphasé phase-terre de 80 ms est simulé au niveau du réseau 400 kV d'IFA2 avec un pas de temps de 45 μ s. Le transit de puissance est fixé à 1000 MW de Tourbe vers Daedalus. Pour avoir la même puissance réactive pour chaque méthode, le CSPR a pour consigne $Q=25$ MVar et la partie française d'IFA2 est en Q-ctrl (contrôle de la puissance réactive Q) avec $Q_{ref}=300$ MVar.

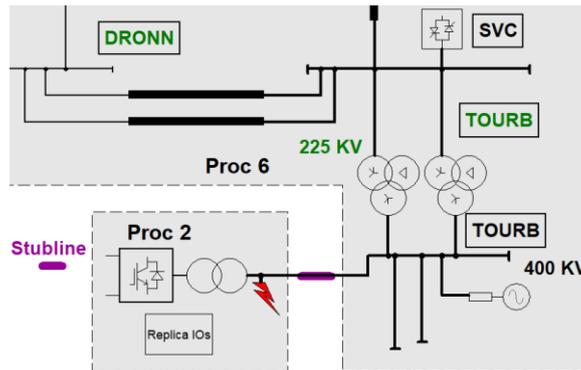


Figure 4-42 : Découplage du Proc 2 par une stubline (Iterative LD+DB).

La Figure 4-43 montre la tension d'une phase en défaut au point du défaut. Deux valeurs d'inductance ont été testées pour la stubline ($L = 1$ mH pour Iterative LD+DB 1, $L = 4$ mH pour Iterative LD+DB 2). Après l'élimination du défaut, la stubline crée une surtension plus élevée que Iterative CM+LD. L'ajout de capacités et d'inductances fictives ont modifié la dynamique du système par rapport à une résolution simultanée (Iterative LD+CM).

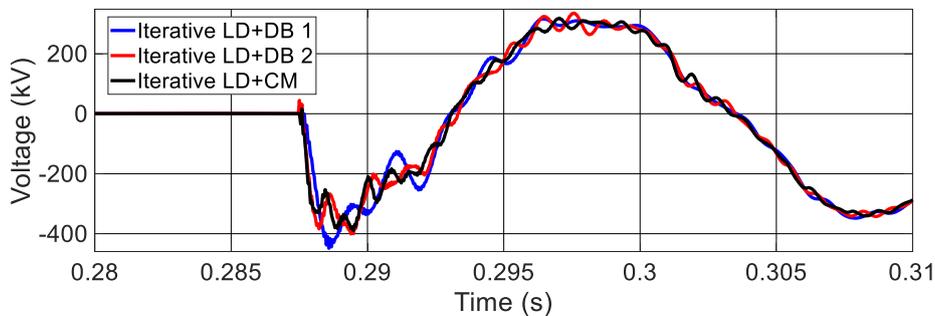


Figure 4-43 Tension AC après l'élimination du défaut Iterative LD+CM et Iterative LD+DB avec deux valeurs d'inductance).

Avec la méthode itérative LD+CM, on est en mesure de réaliser une étude d'interaction avec la réplique d'IFA2 comme celles décrites dans [89][90]. Quelques scénarios ont été simulés pour mesurer l'impact d'IFA2 sur le CSPR. La consigne du CSPR est fixée à $U_{ref}=238$ kV et la tension AC du réseau à 240 kV. Cela correspond à une absorption initiale de réactif de 25 MVar par le CSPR. Un défaut monophasé phase-terre est simulé au point de connexion du CSPR pour vérifier son bon fonctionnement. Tout d'abord, on simule sans IFA2 (Tableau 4-2, cas 1). Puis, l'impact d'IFA2 est analysé à travers cinq points de fonctionnement (Tableau 4-3). Au total, ce sont six cas d'une étude d'interaction qui sont simulés. La convention de signe est la suivante : un transit de puissance positif de Tourbe vers Daedalus, une puissance réactive capacitive négative pour le CSPR.

Tableau 4-2 : Réseaux simulés pour l'étude d'interaction.

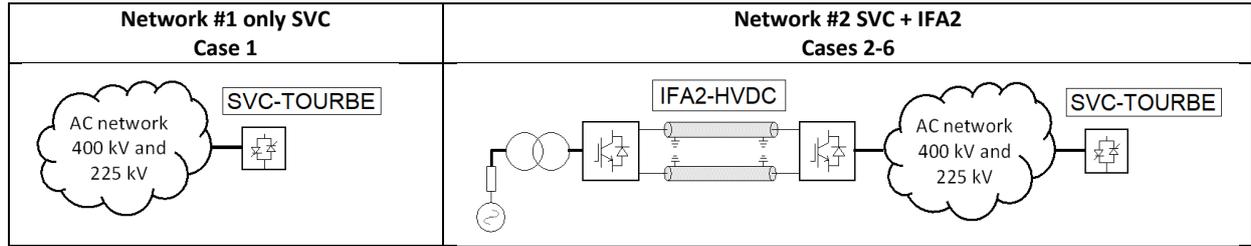


Tableau 4-3 : Configurations de fonctionnement d'IFA2 pendant le défaut AC.

Scénarios	Puissance réactive de IFA2	Puissance active de IFA2
Case 2	Pref = 1000 MW	U-ctrl, Uref = 440 kV
Case 3	Pref = 1000 MW	U-ctrl, Uref = 380 kV
Case 4	Pref = -1000 MW	U-ctrl, Uref = 440 kV
Case 5	Pref = -1000 MW	U-ctrl, Uref = 380 kV
Case 6	Pref = 0 MW	Q-ctrl, Qref = 0 MVAR

Les résultats des six scénarios sont montrés dans la Figure 4-44. Aucun effet adverse n'est observé. Dans chaque cas, le fonctionnement CPSR est stable vis-à-vis du défaut. Les valeurs contrôlées (Q_{svc} et V_{svc}) retournent à leur état initial. Par ailleurs, une surtension apparaît sur les phases saines (max RMS V_{svc}) une fois que défaut est éliminé. Cela vient de l'apport du CPSR en puissance réactive qui met des dizaines de ms à s'arrêter après l'élimination du défaut. La présence d'IFA2 semble atténuer ce pic. Ces résultats montrent que grâce à la méthode de compensation, on est capable de faire une étude de l'interaction entre la liaison HVDC IFA2 et le CPSR de Tourbe en temps réel avec les mêmes contrôleurs que ceux installés sur le site.

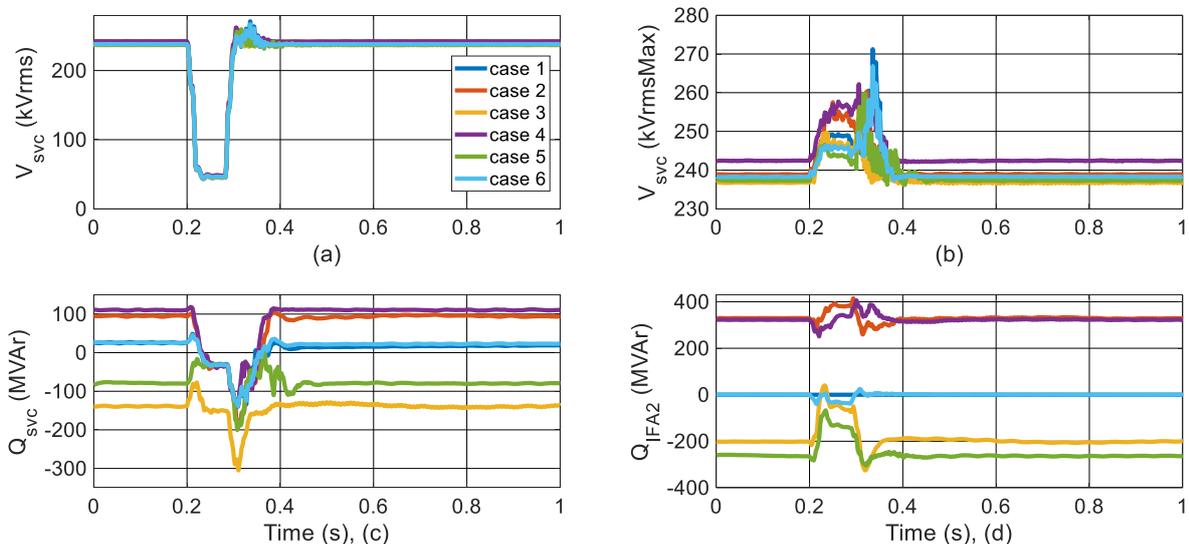


Figure 4-44 : Résultats des simulations, pour le CPSR, tension RMS de la phase en défaut (a), maximum des tensions RMS sur les phases saines (b), puissance réactive (c), et pour IFA2 la puissance réactive (d).

4.6.2.2 Réseaux avec des caractéristiques non-linéaires

L'énergisation de transformateurs de la liaison LCC-HVDC d'IFA2000 [94] permet d'illustrer la résolution des équations de la saturation non-linéaire par Iterative LD+CM lors d'une simulation HIL temps réel. La plateforme temps réel est la même que celle de [23] où le contrôle générique de Simulink a été remplacé par la réplique du contrôle-commande. Le cas de test consiste à l'énergisation de deux transformateurs au même nœud électrique. Cela correspond à l'énergisation du second bipôle. C'est un scénario critique qui peut impacter le bon fonctionnement de la liaison. Une caractéristique linéarisée à deux points est utilisée pour modéliser la saturation des

transformateurs. Avec ce type de caractéristique, la précision est plus sensible à une sélection erronée de segment. Le découplage CM par pôles est choisi en complément de celui des lignes (Iterative LD+CM). Il est plus stable que celui par ponts. Une coupe CM supplémentaire permet de découpler les transformateurs du poste des Mandarins. Huit cœurs sont requis pour Iterative LD+CM contre cinq pour Iterative LD. Les itérations peuvent être activées uniquement pour le côté français où tous les sous-réseaux y prennent part.

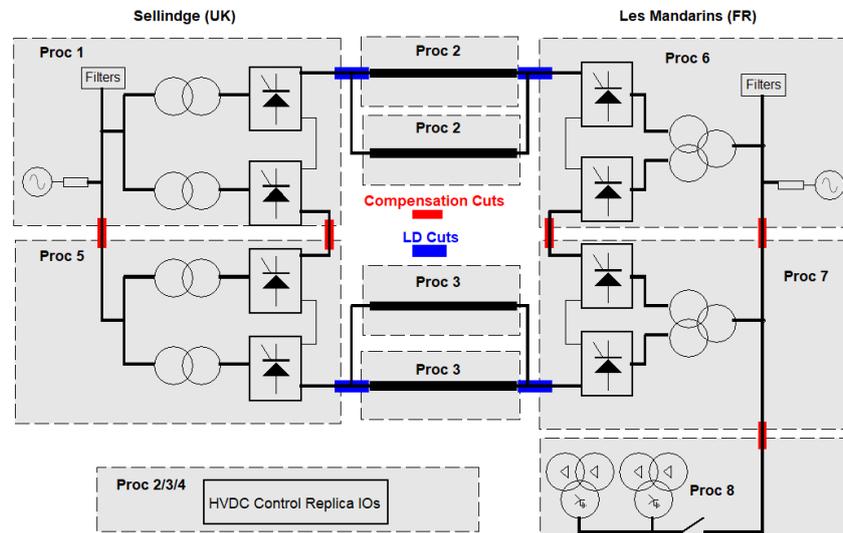


Figure 4-45 : Séparation en tâches d'Iterative LD+CM pour l'énergisation de transformateurs d'IFA2000.

Le transit de puissance est fixé à 300 MW de l'Angleterre vers la France. L'énergisation est déclenchée au cours d'une acquisition d'1 s une fois la consigne de transit atteinte. Les temps d'enclenchement pour chaque phase (A, B et C) sont respectivement $t_A = 128$ ms, $t_B = 129$ ms et $t_C = 119$ ms. L'énergisation amène à de fortes perturbations sur la tension AC et des ratés de commutation. Ce cas test tourne en temps réel avec un pas de temps de 40 μ s sans overruns. Les méthodes Iterative LD+CM et Iterative LD ont une réponse similaire pour la tension DC (Figure 4-46). Les performances sont améliorées de 40% avec Iterative LD+CM (Figure 4-47). Cela permet d'atténuer les pics du temps d'exécution lorsque plusieurs itérations sont nécessaires. Comme pour 4.6.1.2, les gains en performance sont d'autant plus importants qu'il faut faire des refactorisations pour les itérations.

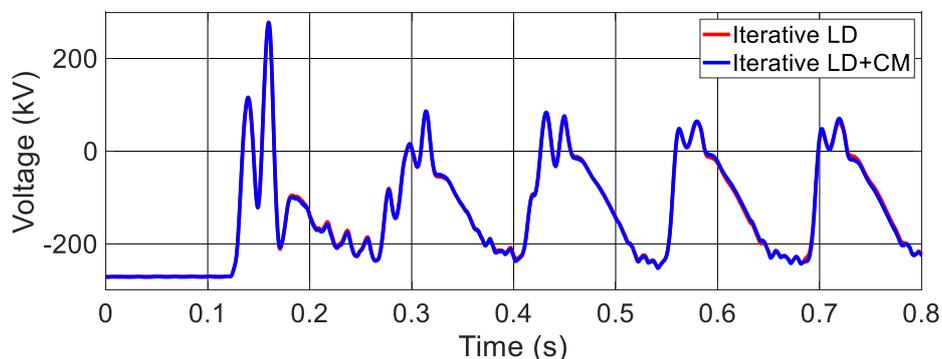


Figure 4-46 : Tension DC pour Iterative LD et Iterative LD+CM.

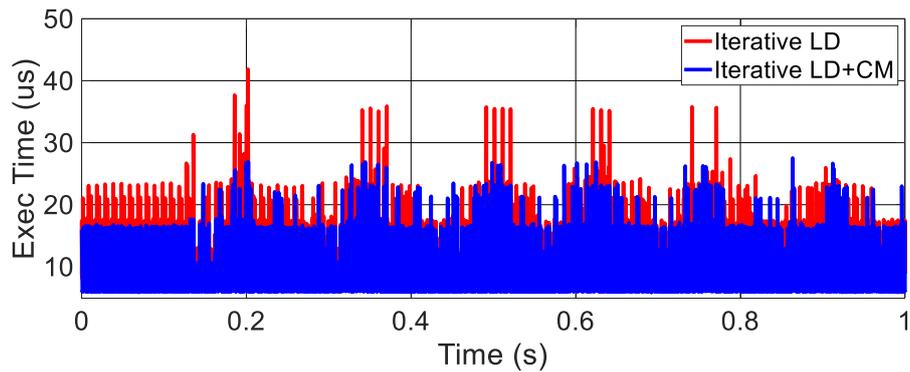


Figure 4-47 : Temps d'exécution du processeur le plus lent de chaque pas de temps pour Iterative LD et Iterative LD+CM.

La Figure 4-48 et son zoom (Figure 4-49) montre que les itérations sont nécessaires pour la précision de la simulation. Sans itérations, la sélection du mauvais segment crée des surtensions fictives pour les transformateurs.

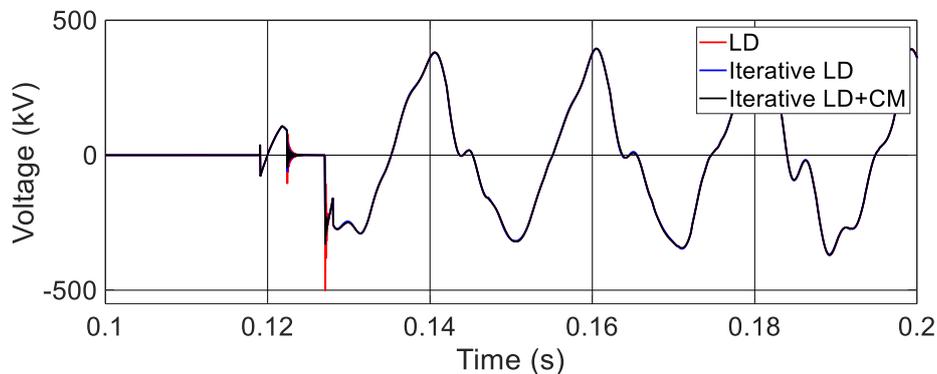


Figure 4-48 : Tension AC au début de l'énergisation pour LD, Iterative LD et Iterative LD+CM.

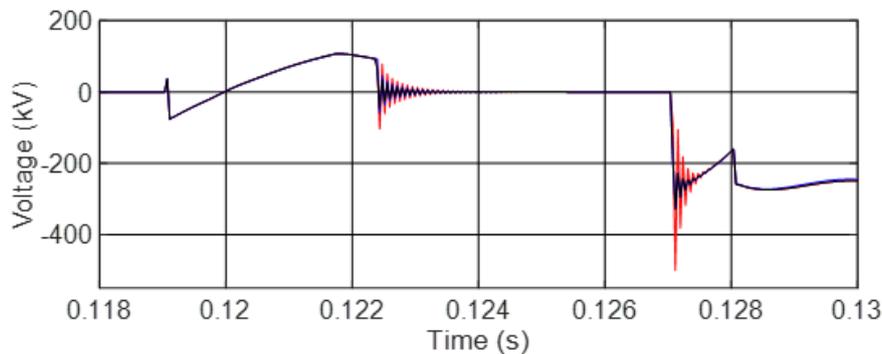


Figure 4-49 : Zoom sur les différences avec et sans itérations.

4.7 Conclusion

Ce chapitre a démontré l'efficacité de la méthode de Compensation pour découpler la solution des équations de réseaux quand aucun délai naturel de ligne n'est disponible, afin de les simuler en parallèle en temps réel. Cette méthode offre des gains de performance substantiels tout en gardant une bonne précision. Contrairement à l'utilisation de stublines qui rajoutent des délais artificiels, la solution reste simultanée. Un large éventail de cas d'applications industrielles (réseaux de distribution, LCC-HVDC, VSC-HVDC, interaction CPSR et VSC-HVDC) a validé les gains de performance en temps réel et démontré qu'il était possible de diminuer le pas de temps que ce soit pour une résolution linéaire ou non-linéaire avec la version itérative. Tout en respectant la contrainte temps réel, cela permet de gagner en précision. Des limites à la méthode de Compensation ont été

identifiées. On ne peut pas couper n'importe où car les matrices des sous-réseaux découplés peuvent être mal conditionnées. Cependant, en pratique, il est beaucoup plus facile d'identifier une coupe de compensation que de calibrer une stubline afin d'éviter des instabilités numériques.

Ce travail est un apport indéniable à la littérature existante qui n'avait pas mené une étude de performance aussi exhaustive. Peu de cas temps réel HIL avaient été étudiés. Principalement des cas linéaires avaient été considérés. Dans ce chapitre, les performances de la compensation ont été démontrées pour des cas non-linéaires qui nécessitent des itérations pour la précision de la simulation. Une revue de la littérature existante est également proposée pour bien comprendre les fondements théoriques de la compensation et les correspondances avec d'autres méthodes. Cela complète un autre document [68] qui avait déjà montré que des méthodes plus récentes comme MATE [69] se ramènent à faire de la Compensation.

Ce deuxième axe de recherche a été le plus ambitieux en termes d'implémentation informatique. En effet, aucun outil de simulation EMT ne propose actuellement un découplage sans délai qui préserve une solution simultanée comme la compensation. Afin d'avoir accès à des cas industriels HIL en temps réel, il a fallu implémenter directement la méthode de Compensation dans l'environnement temps réel d'HYPERSIM. Cela était équivalent à réimplémenter une solution de réseau complète pour gérer les différents éléments de découplage de la compensation. Une telle implémentation a demandé de bien maîtriser les différents éléments d'un outil de simulation EMT : identification topologique des éléments de découplage, mécanisme de parallélisation via les barrières, génération du code pour les calculs de la compensation.

Chapitre 5 Parallélisation des résolutions de réseaux : Répartition des tâches sur les processeurs

Dans ce chapitre, on va aborder le problème de répartition des tâches sur les cœurs/processeurs de l'architecture utilisée pour la simulation. Les tâches pour ce chapitre sont issues uniquement des techniques de parallélisation avec délai (délai des lignes ou lignes fictives). Pour les techniques sans délai, cela a été évoqué dans le paragraphe 4.4.4. Une fois que le réseau a été séparé en sous-réseaux/tâches, il ne reste plus qu'à répartir les tâches sur les processeurs de l'architecture multi-cœurs de simulation. Si pour de petits réseaux, une allocation manuelle des tâches est faisable afin d'optimiser les performances, elle devient beaucoup plus difficile pour de grands réseaux. L'objectif est de proposer un algorithme de répartition des tâches qui va automatiser cette répartition sur les processeurs.

Le premier critère à optimiser est le Load Balancing, qui consiste à une équipartition de la charge de calcul des processeurs afin de minimiser le temps de simulation. Ce seul critère a été optimisé en temps différé en utilisant l'approche du Bin Packing [63][95]. Néanmoins, un autre critère doit être pris en compte dans la parallélisation, c'est le coût de communication entre les différents processeurs. Ainsi, une formulation plus générale en Problème d'Allocation des Tâches (TAP) [35][96] est plus adéquate en prenant en compte ces deux critères. Comme TAP est un problème NP-Complet, des heuristiques ont été utilisées pour le temps réel. Initialement une recherche A* [36] a été utilisée dans [4]. Mais récemment (fin des années 90s), les algorithmes de partitionnement de graphes multi-niveaux [37][83] sont apparus comme étant les plus performants et les plus utilisés pour la simulation EMT.

La contribution de ce chapitre est de faire une étude comparative des différentes heuristiques pour résoudre un TAP sur de grands réseaux. Plus précisément, pour le partitionnement de graphes multi-niveaux, une étude d'optimalité est menée en le comparant à la résolution exacte et en cherchant le paramétrage le plus adéquat. Les performances des différentes heuristiques sont validées sur des cas HVDC de simulation temps réel.

5.1 Principe

On dispose d'un ensemble de tâches T_i de taille n avec un ensemble de processeurs P_j de taille m . L'objectif est d'affecter pour chaque tâche un seul processeur et de profiter au maximum de la parallélisation offerte par des machines multi-cœurs.

Deux grandes familles de répartition sont possibles :

- La **répartition statique** où les temps d'exécution de chaque tâche sont connus ou estimés à l'avance et ne varient pas au cours de la simulation. Ainsi, l'algorithme de répartition est lancé en offline (avant la simulation).
- La **répartition dynamique** où les temps d'exécution varient au cours de la simulation. Un algorithme va modifier la répartition au cours de la simulation si des temps d'exécution changent.

Dans le cadre des simulations EMT, la répartition statique est généralement utilisée. Cependant, la répartition dynamique peut avoir une utilité dans la gestion des éléments non-linéaires qui peuvent modifier le temps d'exécution d'une tâche au cours de la simulation.

Pour l'environnement temps réel d'HYPERSIM, les temps d'exécution sont estimés à l'avance. Pour chaque sous-réseau, le nombre de branches RLC donne une approximation du temps d'exécution. La plupart des modèles disposent d'une estimation de la charge de calcul à partir des performances pratiques. Bien sûr, ces temps d'exécution peuvent être ajustés en fonction de l'architecture de simulation. Néanmoins, l'augmentation du temps de calcul, dans le cadre de non linéarités, n'est pas prise en compte, ce qui peut faire varier l'exécution d'une tâche vis-à-vis de l'estimation.

Pour le temps différé [63], la charge de calcul de chaque tâche est également connue avant la simulation, en estimant le nombre de calculs qui sont nécessaires pour la résolution nodale. Pour mieux prendre en compte cette fluctuation de charge liée aux non-linéarités, des répartitions dynamiques ont été testées [95]. Pour le moment, il est apparu que la répartition statique reste plus rapide que la répartition dynamique. Par la suite, on considèrera uniquement les répartitions statiques.

5.2 Méthodes de résolution

Comme cela a été vu précédemment, c'est la répartition statique des tâches qui est principalement utilisée dans la parallélisation des outils EMT. Au niveau des tâches, on ne trouve pas de lien de préséance. Par exemple, une tâche T_i n'attend pas à l'instant t le résultat d'une autre tâche T_j calculé au même instant t pour commencer son calcul. Ainsi, il n'y a aucun ordonnancement des tâches à effectuer.

Les objectifs d'un algorithme de répartition statique des tâches sont multiples :

- **Minimiser le temps de simulation** en profitant au maximum de la parallélisation. Cela revient à minimiser le temps maximum d'exécution par processeur.
- Assurer un **Load Balancing** entre les processeurs (équilibrage de la charge de calcul). Cet équilibrage des charges assure de meilleures performances et limite les overruns potentiels. Il faut noter que le premier objectif n'amène pas forcément à un load balancing.
- **Minimiser les communications entre les tâches**. On parle de problème d'affectation des tâches. Cela dépend à la fois des interactions entre les tâches mais aussi de l'architecture (coût de communication entre les processeurs).
- Pour le temps réel, **respecter la contrainte de temps**. Le temps d'exécution sur chaque processeur ne doit pas excéder le pas de temps de la simulation.

5.2.1 Approche du Bin Packing

Pour le temps différé [63][95], la parallélisation a été faite uniquement pour la résolution matricielle de l'analyse nodale. La mise à jour des modèles (calculs des historiques) est séquentielle et est assurée par l'instance principale de l'application qui a lancé les différents threads, au plus un par processeur. Chaque thread (processeur) va exécuter les calculs de sous-blocs matriciels (les tâches issues de la parallélisation). Il n'y a pas à proprement parler de communications entre les threads. Ainsi, la répartition des tâches sur les processeurs va uniquement se concentrer sur la minimisation du temps de simulation via le critère du Load Balancing.

Le problème de répartition des tâches se rapproche d'un problème d'optimisation combinatoire bien connu qu'est le **Bin Packing**. Ce problème est NP-difficile et trouver l'optimum est très coûteux.

Généralement, on passe par une formulation de programmation linéaire en nombre entiers. Pour un grand nombre de tâches et de processeurs, les temps de résolution peuvent être extrêmement longs. On utilise des heuristiques plus rapides qui permettent d'arriver à une solution approchée (minimum local).

Soient P un ensemble de processeurs P_i (bin/ boîte de rangement) de taille m et T un ensemble de tâches T_j (objets à ranger dans les boîtes) de taille n auxquelles on associe un temps d'exécution w_j (la place occupée par l'objet). Soient les variables binaires x_{ij} qui indiquent si la tâche i est affectée ($x_{ij} = 1$) au processeur j .

On veut qu'une tâche soit affectée à un seul processeur :

$$\forall i \in \llbracket 1, n \rrbracket, \sum_{j=1}^m x_{ij} = 1 \quad (5-1)$$

A partir de là, deux approches sont possibles. La première approche consiste à fixer la durée maximale de simulation à w et à rechercher un nombre minimal de processeurs pour exécuter l'ensemble des tâches dans ce temps imparti. C'est un problème de Bin Packing classique. Soit y_i la variable binaire qui indique si le processeur i est utilisé ($y_i = 1$) ou pas ($y_i = 0$).

$$\forall i \in \llbracket 1, n \rrbracket, y_i = \left(\sum_{k=1}^m x_{kj} > 0 \right) \quad (5-2)$$

$$\forall j \in \llbracket 1, m \rrbracket, \sum_{i=1}^n w_i x_{ij} \leq w \quad (5-3)$$

$$\min \sum_{i=1}^m y_i \quad (5-4)$$

Ce problème s'applique plutôt au cas du temps réel où w correspond au pas de temps de la simulation que ne doit pas dépasser l'exécution des tâches.

La seconde approche, plus appropriée pour le temps différé, consiste à minimiser le temps de simulation en fixant le nombre de processeurs (généralement fixé par l'utilisateur) utilisable pour la parallélisation à M . On relâche la contrainte (5-3) et l'on utilise comme fonction objectif :

$$\min \left(\max_{j \in \llbracket 1, M \rrbracket} \sum_{i=1}^n w_i x_{ij} \right) \quad (5-5)$$

En revanche, si l'on veut un load balancing, il faut plutôt minimiser l'erreur à la moyenne,

$$\min_{k \in \llbracket 1, M \rrbracket} \left(\left| \sum_{i=1}^n w_i x_{ik} - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^n w_i x_{ij} \right| \right) \quad (5-6)$$

Ou pour être plus précis la variance,

$$\min \frac{1}{M} \sum_{k=1}^M \left(\sum_{i=1}^n w_i x_{ik} - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^n w_i x_{ij} \right)^2 \quad (5-7)$$

En pratique pour le temps différé, une heuristique simple du Bin Packing a été utilisée pour affecter les tâches aux processeurs [95]: la méthode de Best Fit Decreasing (BFD)[97]. On trie les tâches par ordre décroissant de temps d'exécution. On parcourt cette liste pour affecter les tâches sur le processeur le moins chargé (le plus faible temps d'exécution pris par les tâches). Très rapide, cet algorithme propose une solution approchée et ne prend pas forcément bien en compte le load balancing.

Une autre heuristique a été mise en place dans [63] pour tenir davantage compte du load balancing dans la parallélisation de la formulation nodale via la forme BDF. Ici, le temps d'exécution des blocs/tâches est assimilé au nombre d'opérations numériques à réaliser (flops).

1. Calcul de la répartition idéale de flops par processeur en divisant le nombre total de flops à répartir avec le nombre de processeurs restants.
2. On prend le premier processeur auquel aucune tâche n'est assignée. On y affecte la tâche du plus grand nombre de flops.
3. Si le nombre de flops du processeur est inférieur à la répartition idéale de 1., on assigne alors la tâche du plus petit nombre de flops.
4. On répète 3. jusqu'à ce que le nombre de flops du thread soit supérieur ou égal à la répartition idéale de 1.
5. Si le nombre de threads est inférieur à ce qu'a demandé l'utilisateur et que le nombre de flops restant est plus grand que le minimum de flops par thread, on retourne à l'étape 1.

L'inconvénient de l'approche du Bin Packing est qu'elle ne prend pas en compte les coûts de communication inter-processeurs. Lorsque la parallélisation est complète comme pour [1], des communications entre tâches/sous-réseaux sont nécessaires comme en témoigne le modèle de ligne de la Figure 4-1. En effet, les deux tâches doivent s'échanger à chaque pas de temps les tensions et les courants aux nœuds, pour pouvoir faire le calcul de la source de courant historique pour le prochain pas de temps. Ces coûts de communication sont présents pour les deux paradigmes de parallélisation : communication par mémoires partagées et communication par messages. La communication par mémoires partagées est la méthode de communication principalement utilisée dans la parallélisation EMT. Même si le temps de communication est moindre que le paradigme par message, il existe à travers le temps d'écriture et de lecture ainsi que la taille de la mémoire partagée.

5.2.2 Approche TAP

Dans un cadre plus complet, les contraintes de communication inter-tâches sont à considérer ainsi que l'architecture du simulateur où les coûts de communication entre processeurs peuvent être différents. Soit d_{ij} le coût de communication entre deux tâches i et j (par exemple le nombre de signaux échangés), c_{kl} le coût de communication entre deux processeurs k et l (par exemple le temps de communication pour échanger un signal). Ainsi, si les deux tâches i et j sont affectées respectivement à ces deux processeurs, le coût de communication est de $d_{ij}c_{kl}$. En prenant en compte les contraintes de load balancing et des coûts de communication dans la fonction objectif, on obtient une formulation complète d'un Problème d'Allocation des Tâches (TAP) avec la contrainte d'unicité de l'allocation d'une tâche sur un processeur :

$$\min \frac{1}{M} \sum_{k=1}^M \left(\sum_{i=1}^n w_i x_{ik} - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^n w_i x_{ij} \right)^2 + \sum_{i=1}^n \sum_{j>i}^n \sum_{k=1}^M \sum_{l \neq k}^M d_{ij} c_{kl} x_{ik} x_{jl} \quad (5-8)$$

$$\forall i \in \llbracket 1, n \rrbracket, \sum_{j=1}^M x_{ij} = 1 \quad (5-9)$$

Où w_i est le temps estimé pour effectuer la tâche, x_{ij} variable binaire d'allocation vaut 1 si la tâche i est allouée sur le processeur j sinon 0, M le nombre de processeurs choisi par l'utilisateur, n le nombre de tâches. Pour le temps réel, on rajoute la contrainte de temps suivante :

$$\forall j \in \llbracket 1, M \rrbracket, \sum_{i=1}^n w_i x_{ij} \leq w \quad (5-10)$$

Cela revient ainsi à résoudre un problème d'optimisation quadratique en nombres entiers qui est NP-difficile. Les différentes méthodes de résolution de ce type de problème d'allocation sont décrites dans [35][97]. Plusieurs heuristiques classiques ont été proposées pour ce type de problème (relaxation lagrangienne, algorithme génétique, recuit simulé...). On peut également envisager de relâcher la contrainte de temps pour trouver des solutions plus rapidement au risque d'être sous-optimal.

Par ailleurs, cette formulation ne propose pas à l'utilisateur une optimisation du nombre de processeurs utilisés par la simulation. Il doit le fournir à l'application où l'algorithme augmente le nombre de cœurs s'il ne trouve pas de solution faisable. Une formulation via le Bin Packing propose cette optimisation (il est également possible de le combiner avec du Load Balancing [98]) mais cependant elle ne prend pas en compte les coûts de communication entre les tâches.

Deux types d'heuristique seront présentés par la suite pour résoudre ce problème :

- Une recherche A* dans un graphe d'état,
- Des heuristiques pour le partitionnement de graphe.

5.2.2.1 Recherche A*

Le premier algorithme utilisé par [1] pour la répartition des tâches est basé sur une heuristique de fouille/recherche. Cette méthode avait été proposée dans le cadre d'une thèse [36] et avait directement été implémentée dans [1]. Le problème est modélisé par un espace d'états (arbre/graphe). Chaque état est une configuration du système où un certain nombre de tâches ont été allouées à un processeur. Soient un ensemble T de tâches T_i de taille m à allouer et P un ensemble de processeurs P_i de taille n . Par définition, un état e contient p affectations de tâches distinctes allouées à un processeur. L'affectation de la tâche T_i avec le processeur P_j est représenté par le tuple (T_i, P_j) . Ainsi, l'état e est l'ensemble contenant les p tuples des affectations comme suit :

$$e = \left\{ (T_{k_i}, P_{l_i}) \right\}_{\substack{i \in \llbracket 1, p \rrbracket \\ \forall i, j \in \llbracket 1, p \rrbracket \quad l_i \neq l_j}} \quad (5-11)$$

Pour la répartition, on utilise les notations ci-dessous.

E Ensemble E des états s .

e_0 Etat initial $e_0 \in E$, $e_0 = \{\emptyset\}$ où aucune tâche n'a été assignée à un processeur.

F Ensemble d'états finaux $F \subset E$ tel que $e_f \in F \Leftrightarrow \text{card}(s_f) = m$. Pour tout état final, cela revient à avoir alloué l'ensemble des m tâches.

$Succ$ Opérateur qui permet d'obtenir les successeurs d'un état e . Un successeur e_{succ} de e est défini comme suit, $\exists k, l \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket / e_{succ} = e \cup \{(T_k, P_l)\}$ avec T_k pas encore allouée dans e . Choisir un successeur de e revient à allouer une tâche supplémentaire.

M Ensemble de règles heuristiques pour sélectionner les successeurs admissibles de e .

S Stratégie pour choisir le prochain état courant dans la liste des états déjà parcourus.

En reprenant ces notations, l'algorithme de recherche générique [99] ci-dessous est utilisé par la fouille heuristique :

Algorithm 1 Recherche générique

```
1:  $ListeEtats \leftarrow \{\emptyset\}$ 
2:  $e \leftarrow e_0$ 
3: while  $e \notin F$  do
4:   for  $succ \in Succ(e)$  do
5:     if  $succ$  satisfait  $M$  then
6:        $ListeEtats \leftarrow Insérer(succ)$ 
7:     end if
8:   end for
9:   Choisir  $e_1 \in ListeEtats$  selon  $S$ 
10:   $e \leftarrow e_1$ 
11: end while
```

Deux éléments restent à définir dans cet algorithme générique : la stratégie S de choix de l'état courant et les règles (heuristiques M) de génération des successeurs.

Pour S , on définit une mesure d'utilité $f: E \rightarrow \mathbb{R}$. On sélectionnera l'état dans $ListeEtats$ qui minimise cette mesure. On introduit deux autres fonctions qui vont permettre de calculer la mesure d'utilité :

- $g: E \rightarrow \mathbb{R}$, $g(e)$ représente le coût mis pour atteindre l'état e depuis l'état initial e_0 ,
- $h: E \rightarrow \mathbb{R}$, $h(e)$ représente le coût estimé pour aller à un état final depuis e .

Voici les méthodes de recherche selon l'expression de f choisie en fonction de g et h :

- la **recherche uniforme** si $f = g$,
- la **recherche gloutonne** si $f = h$,
- la **recherche A*** si $f = g + h$. Ici, $f(e)$ correspond au coût total estimé pour aller d'un état initial vers un état final en passant par l'état e . C'est l'heuristique qui a été implémentée à l'origine dans [1].

C'est dans le calcul de la fonction d'utilité f que sont pris en compte le load balancing ainsi que la minimisation des coûts communication.

Pour $g(e)$, voici des exemples de critères de pénalité par ordre de priorité pour l'état e :

1. Pénalisation des processeurs dont le taux d'occupation est important (Load Balancing) avec la répartition de e .
2. Pénalisation du coût des communications entre processeurs généré par la répartition des tâches de l'état e .
3. Les processeurs les moins occupés dans la répartition de e sont priorisés légèrement.

Pour $h(e)$, on essaie d'évaluer ce qui reste à faire pour avoir une répartition de toutes les tâches depuis l'état e . Cela va dépendre du nombre de tâches restantes à allouer ainsi que du nombre de tâches qui ont des liens de communication avec des tâches déjà allouées de e . Le calcul de h ne prend pas en compte directement le load balancing. Cependant, il favorise l'allocation groupée des tâches qui sont liées, ce qui peut optimiser les coûts de communication si cela se fait sur le même processeur.

Pour les règles M de génération des successeurs de l'état courant qui alimentent la liste des états à traiter ($ListeTrieEtats$), les stratégies suivantes peuvent être mises en place :

- Priorité donnée aux tâches importantes pour faciliter le load balancing.
- Sélectionner uniquement les tâches voisines (liens de communication avec des tâches déjà allouées).
- Trouver les processeurs candidats (qui respectent la contrainte temps réel) pour former les affectations possibles avec les tâches présélectionnées juste avant.

Afin d'implémenter la mesure d'utilité f , il suffit d'avoir une *ListeEtats* triée par ordre décroissant de f . Choisir l'état d'utilité minimale revient à retirer le premier élément de cette liste triée. Ci-dessous l'algorithme de la recherche A* est détaillé.

Algorithm 2 Recherche A*

```

1: ListeTrieEtats  $\leftarrow \{\emptyset\}$ 
2:  $e \leftarrow e_0$ 
3: while  $e \notin F$  do
4:   for  $succ \in Succ(e)$  do
5:     if  $succ$  satisfait  $M$  then
6:        $val \leftarrow g(succ) + h(succ)$ 
7:       ListeTrieEtats  $\leftarrow Inserter(succ, val)$ 
8:     end if
9:   end for
10:   $e \leftarrow Head(ListeTrieEtats)$ 
11: end while

```

Afin de mieux comprendre la stratégie de recherche A*, appliquons-la à l'exemple très simple ci-dessous qui comporte 4 postes, 2 lignes doubles (en rouge) et 2 lignes simples (en bleu). Les dénominations de « poste » et de « ligne » sont génériques ici pour être le plus général possible vis-à-vis de la définition des tâches de réseau pour la parallélisation.



Figure 5-1 : Exemple simple d'un réseau à quatre tâches à répartir.

On souhaite réaliser la répartition des tâches sur 4 processeurs avec un pas de temps de 20 μ s. Si on estime que la propagation de chaque ligne est supérieure au pas de temps, on peut séparer ce réseau en 4 tâches (S1, S2, S3 et S4), une pour chaque poste. Ci-dessous (Figure 5-2) sont présentés le graphe des tâches (temps d'exécution estimés aux sommets en μ s, communication entre les tâches aux arêtes) ainsi que celui des processeurs (communications entre processeurs sur les arêtes).

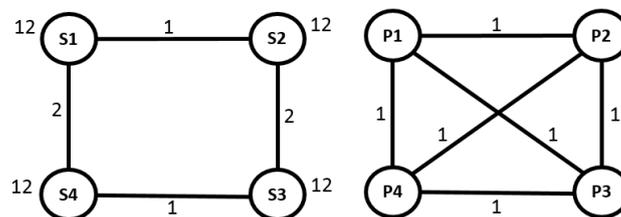


Figure 5-2 : Graphes des tâches et de processeurs associés au réseau.

On commence la recherche A* à partir d'une assignation tâche-processeur vide. Puis on génère les assignations possibles. A ce stade c'est l'ensemble des assignations possibles qui est considéré. Après avoir trié les assignations candidates selon la mesure utilité, on sélectionne celle qui minimise cette mesure (S1, P1). On affecte la tâche S1 au processeur P1. On génère à nouveau les assignations candidates. Cette fois-ci, S1 étant déjà affectée on ne considère plus cette tâche pour les assignations possibles. De même P1 est exclu car y affecter une tâche supplémentaire reviendrait à ne plus

respecter le pas de temps (charge estimée alors à $24 \mu s > 20 \mu s$). La mesure d'utilité fait choisir l'assignation (S2, P2). On réitère le processus jusqu'à ce que la dernière tâche poste S4 soit assignée.

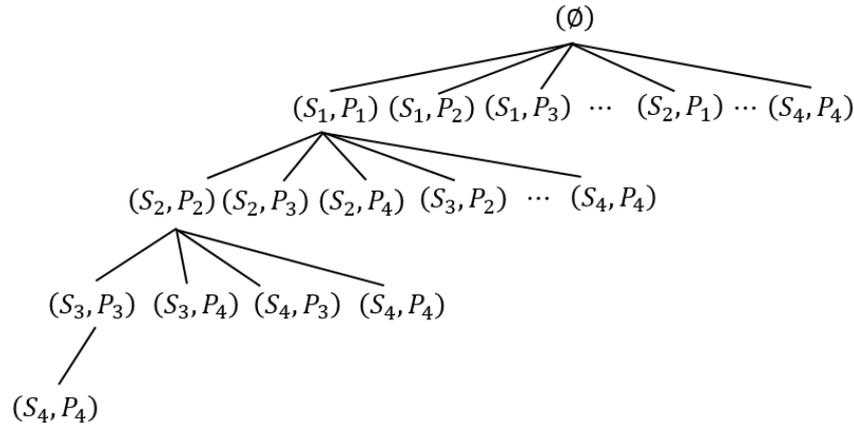


Figure 5-3 : Déroulé de la recherche A* sur le réseau à 4 tâches.

En pratique, l'algorithme A* fournit des résultats mitigés au niveau des performances ainsi qu'au niveau de la qualité des solutions proposées. En effet, il est difficile à régler comme l'a montré le détail du calcul de la mesure d'utilité si l'on veut à la fois minimiser le nombre de communications et favoriser le load balancing. L'algorithme aura tendance à étaler les tâches sur un maximum de processeurs disponibles. On peut se retrouver avec des processeurs peu occupés ce qui remet en cause le load balancing. Cependant, la solution proposée a le mérite d'être toujours admissible vis-à-vis des contraintes temporelles.

Par ailleurs, les performances se dégradent avec des instances importantes. Cela peut provenir de backtraps (retour en arrière) dans l'algorithme. En effet, si $Card(e) > Card(Hail(ListeTrieEtats))$ (le nombre de tâches allouées à diminuer avec le nouvel état courant), l'algorithme est allé vers une solution sous-optimale ou infaisable. Il retourne en arrière en diminuant le nombre de tâches allouées dans l'état courant. De plus, à chaque itération la taille de *ListeTrieEtats* ne fait qu'augmenter en ajoutant les successeurs admissibles de l'état courant ce qui dégrade les performances.

5.2.2.2 Partitionnement de graphe

Un problème de répartition de tâches peut se ramener à un problème de partitionnement de graphe [37]. L'objectif est de mapper un graphe source S (graphe des tâches) sur un graphe cible C (graphe des processeurs). Le graphe source est d'abord partitionné en sous-parties, équivalentes chacune à un processeur ; puis ces sous-ensembles de sommets sont mappés à un sommet de C (allocation sur un processeur). On note $V(S)$ et $V(C)$, les ensembles des sommets respectifs de S et de C , $E(S)$ et $E(C)$ les ensembles des arcs. S un graphe valué en ses arcs ($c: E(S) \rightarrow \mathbb{R}$) et ses sommets ($w: V(S) \rightarrow \mathbb{R}$). Le graphe C est considéré comme non valué (architecture homogène). Cette approche peut se généraliser aux architectures hétérogènes (graphe C valué). Mapper S sur C consiste à définir deux applications d'allocation, une pour associer les sommets de S à ceux de C , $\varphi: V(S) \rightarrow V(C)$, et une autre $\rho: E(S) \rightarrow \{E(C)\}$, $\rho(e_S) = \{e_C^1, \dots, e_C^n\}$ où $\{e_C^1, \dots, e_C^n\}$ est le chemin minimal sur C pour réaliser la communication e_S . L'objectif du partitionnement est de minimiser les coûts de communication à travers la fonction suivante:

$$\min \sum_{e_S \in E(S)} c_S(e_S) |\rho(e_S)| \quad (5-12)$$

Où $|\cdot|$ est l'opérateur de cardinalité. La minimisation de ce critère a pour conséquence d'agglomérer les sommets d'adjacents de S qui ont des coûts de communication élevés en les allouant au même sommet de C ou à des sommets proches. Ainsi, cela réduit la taille des chemins dans C qui ont été mappés aux arcs à fort coût de communication de S . En complément de ce critère, la répartition de la charge entre les processeurs (sommets de C) est prise en compte. Un Load Imbalance Ratio (LIR) [37] $0 \leq \delta \leq 1$ est défini comme suit

$$\delta = \frac{\sum_{v_c \in V(C)} \left| \sum_{\substack{v_s \in V(S) \\ \varphi(v_s) = v_c}} (w(v_s)) - \mu \right|}{\sum_{v_s \in V(S)} (w(v_s))} \quad (5-13)$$

Avec,

$$\mu = \frac{\sum_{v_s \in V(S)} (w(v_s))}{|V(C)|} \quad (5-14)$$

Où μ est la répartition idéale moyenne par processeur, δ est l'écart normalisé par la charge de toutes les tâches vis-à-vis de la répartition idéale μ . Généralement, δ est considéré comme une contrainte faible, le coût des communications (5-12) reste la fonction objectif.

Les méthodes de partitionnement de graphe [100] peuvent être classifiées comme suit : les méthodes exactes basées sur une formulation en programmation mathématique comme celles de (5-8)(5-9), les techniques spectrales, géométriques et multi-niveaux. Les méthodes exactes [96] sont généralement utilisées sur de petites instances du problème pour vérifier la qualité des autres heuristiques (voir 5.3.4). Les méthodes géométriques [101] sont basées sur des coordonnées spatiales, ce qui ne correspond pas au formalisme d'un TAP. Les techniques spectrales [102] calculent les vecteurs propres dits de Fielder [103] de la matrice Laplacienne du graphe source. Obtenir ces vecteurs est très coûteux en temps de calcul. Ainsi, ce type de méthodes a été dépassé par les algorithmes de partitionnement multi-niveaux. En effet, ces méthodes sont très rapides en utilisant l'approche multi-niveaux. La Bipartition Récursive (RB) [37] multi-niveaux a été testée sur des exemples de simulations EMT [104]. Cette méthode est utilisée dans [1]. Elle sera détaillée par la suite pour bien comprendre les avantages en termes de performances et les limites d'un partitionnement multi-niveaux. Récemment, les techniques multi-niveaux ont été optimisées [105] dans la recherche locale et l'itération sur la graine aléatoire (seed). Mais, la tendance actuelle est aux méthodes multi-niveaux évolutionnaires [106] qui combinent les principes d'un algorithme génétique avec l'approche multi-niveaux pour améliorer la qualité des solutions obtenues.

Une des limites du partitionnement de graphe pour le temps réel est de ne pas prendre en compte la contrainte temporelle sur le pas de temps. Le LIR reste une contrainte faible. Ceci a été très peu étudié dans la littérature des simulations EMT. Cela est analysé en détail dans le paragraphe 5.3.

5.3 Partitionnement multi-niveaux de graphe

5.3.1.1 Principe de l'algorithme

5.3.2 Un algorithme rapide de partitionnement

L'algorithme de partitionnement de graphe implémenté dans Scotch [37][83] est composé de plusieurs sous-routines. La principale est le **Bipartition Récursive (RB)**. Cet algorithme utilise l'approche diviser pour régner en appliquant récursivement des bipartitions d'un sous-ensemble de tâches sur deux sous-ensembles de processeurs.

Entrée : ensemble des processeurs P , ensemble des tâches T .

- Si $P = 1$ (un seul processeur), terminaison de l'algorithme

- P est divisé en deux ensembles P0, P1 via un **algorithme de bipartition** qui minimise les communications entre les deux ensembles de processeurs.
- T est bipartitionné sur P0 et P1 en deux sous-ensembles T0 et T1 via un **algorithme de bipartition** qui minimise les communications entre tâches et qui respecte la contrainte faible du load balancing.
- Réappliquer l'algorithme pour chaque paire: (P0, T0) et (P1, T1).

Reprenons l'exemple simple de la Figure 5-1 du réseau à quatre tâches à répartir sur quatre processeurs. En déroulant l'algorithme, on bipartitionne le graphe des processeurs en deux sous-ensembles (P1, P4) et (P2, P3) tout en veillant à minimiser la communication entre ces deux groupes. Puis, on fait de même avec le graphe des tâches où deux sous-ensembles sont obtenus (S1, S4) et (S2, S3) et affectés sur un groupe de processeurs en tenant compte des critères de Load Balancing et de minimisation des communications inter-tâches : (S1, S4) -> (P1, P4), (S2, S3) -> (P2, P3). Récursivement on réitère la bipartition. On subdivise (P1, P4) en deux sous-groupes de processeurs (P1) et (P4). On bipartitionne également le groupe de tâches associé (S1, S4) en (S1) et (S4) que l'on affecte à un groupe de processeurs : (S1) -> (P1), (S4) -> (P4). Pour l'autre sous-groupe de tâches (S2, S3) -> (P2, P3) on obtient de même une affectation finale : (S2) -> (P2), (S3) -> (P3).

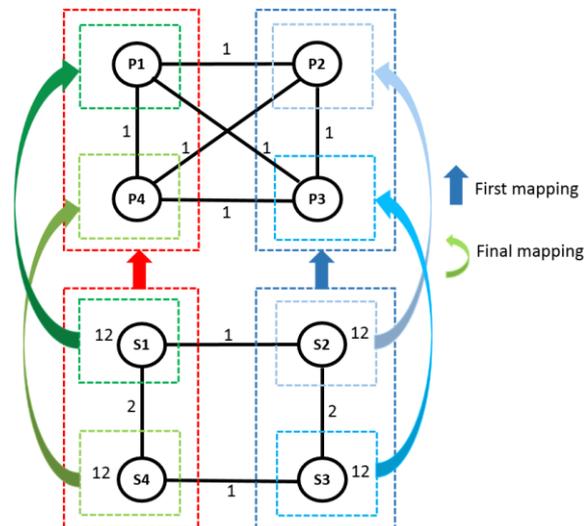


Figure 5-4 : Déroulé de l'algorithme de Bipartition récursive sur un exemple simple à quatre tâches.

Il reste à définir l'**algorithme de bipartition** à utiliser. Dans [37][83], un algorithme glouton de Bipartition (**GPA**) est employé. En commençant sur un sommet sélectionné au hasard, il utilise des heuristiques pour diviser l'ensemble des sommets du graphe en deux parties égales. Il itère jusqu'à garder la meilleure partition. L'avantage d'un algorithme glouton est qu'il apporte une bonne approximation de la solution optimale. Son inconvénient est qu'il est lent pour de grandes instances. Afin d'accélérer l'algorithme, le **Multi-niveaux (ML)** a été mis en place pour réduire la taille du graphe à partitionner via le GPA. Dans une première étape (opération de coarsening), la méthode ML réduit le graphe en un équivalent plus petit. Cette réduction du graphe est stoppée quand une limite inférieure sur sa taille est atteinte. Un algorithme **RB** avec le **GPA** pour la bipartition peut être employé sur l'équivalent réduit du graphe. Une fois la partition faite, le graphe va recroître pour revenir à sa taille d'origine (phase de uncoarsening). Cette méthode permet une accélération de l'algorithme de partitionnement pour de grands graphes qui n'est finalement appliqué qu'à un équivalent réduit. On retrouve cette même intuition dans le répartiteur de tâches historiques de [1] basé sur la recherche A*. Une possibilité était offerte pour regrouper les tâches en une super tâche pour accélérer l'algorithme de répartition.

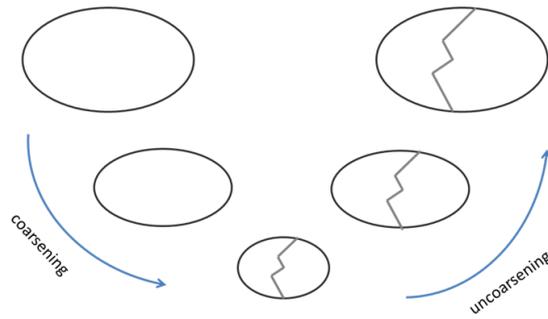


Figure 5-5 : Principe de la méthode multi-niveaux.

Afin de respecter la contrainte du load balancing, une heuristique de post-traitement, appelée **Exactifier (EX)**, est lancée à la fin du multi-niveaux. Elle consiste à échanger des sommets entre partitions pour mieux les équilibrer au niveau de la charge de calcul. Afin d'éviter une détérioration de la qualité de la solution, les échanges de sommets qui impactent le moins le coût des communications sont privilégiés. En mettant bout à bout ces différentes sous-routines, la Figure 5-6 illustre les différentes étapes de l'algorithme de partitionnement employé par Scotch.

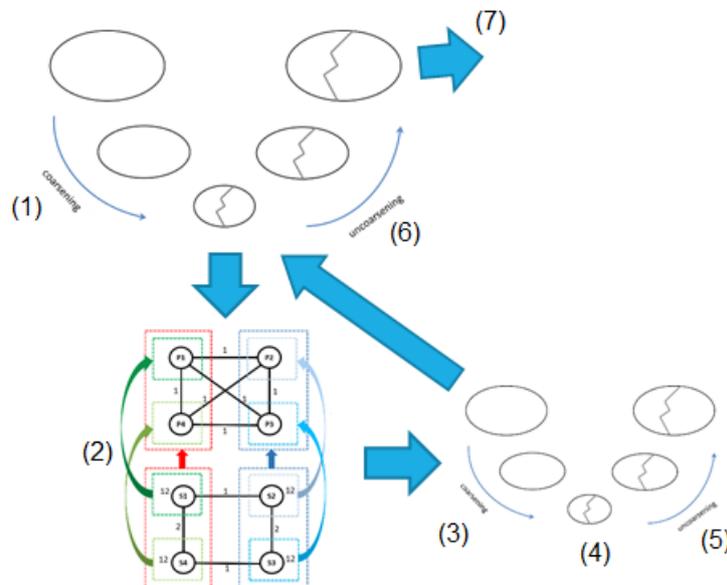


Figure 5-6 : Etapes de l'algorithme de partitionnement multi-niveaux implémenté dans Scotch.

Tout d'abord, un ML externe pour les graphes de grande taille (supérieure à 5000 sommets) réduit la taille du graphe durant la première phase de coarsening (étape 1). Puis, un algorithme RB (étape 2) utilise un ML interne pour réduire la taille de chaque partition en un graphe de petite taille contenant au plus 120 sommets (étape 3). Le GPA est utilisé comme algorithme de bipartition sur le graphe réduit. Puis, les phases d'uncoarsening internes et externes, permettent de revenir à la taille du graphe d'origine (étapes 5-6). A la fin de l'algorithme (étape 7), la méthode EX va améliorer le load balancing. Deux stratégies sont proposées par Scotch : Quality et Balance. Elles permettent d'avoir des stratégies différentes en privilégiant un critère plus qu'un autre, le coût des communications pour Quality, le load balancing pour la stratégie Balance. Par exemple, la méthode EX n'est utilisée que pour la stratégie Balance et évitée pour Quality pour ne pas détériorer le critère sur le coût des communications.

5.3.2.1 Limites

Dans l'algorithme de partitionnement de graphe à aucun moment il n'y a de restriction sur la charge de calcul sur chaque processeur. La contrainte temporelle doit être vérifiée après coup sur le résultat

de la répartition. Si la charge de calcul dépasse le pas de temps, un processeur supplémentaire est rajouté à la répartition. Cette vérification est répétée jusqu'à ce que la répartition trouvée soit admissible vis-à-vis de la contrainte temporelle ou qu'il n'y ait plus de processeurs disponibles pour la répartition. Une conséquence de cette méthode de validation est l'utilisation d'un grand nombre de processeurs sous-optimal pour faire tourner la simulation. Par ailleurs, comme une stratégie est composée de plusieurs heuristiques, elle peut rater une solution admissible pour un nombre de processeurs donné et augmenter le nombre de processeurs. L'utilisation de différentes stratégies peut mener à des solutions très différentes, de celles avec moins de processeurs utilisés à celles qui proposent une solution infaisable.

5.3.3 Recherche d'un paramétrage optimal de l'algorithme

Pour pallier aux limites de l'algorithme au niveau de la contrainte temps réel, la recherche d'un paramétrage optimal a été entreprise.

Tout d'abord, le **Load Imbalance Ratio** permet de fixer la contrainte faible sur le load balancing. En le fixant proche de 0, il va prioriser le load balancing sur la minimisation des coûts de communication. Il y aura moins de risque qu'un processeur soit surchargé et donc d'obtenir une solution non admissible vis-à-vis de la contrainte temps réel.

A l'image des stratégies Balance et Quality, on peut mettre en place **une stratégie spécifique (Specific)** qui soit plus efficace pour la répartition des tâches d'un réseau électrique. Généralement, le graphe des tâches issu de la parallélisation des grands réseaux électriques reste pour sa taille de l'ordre du millier de tâches. Ce sont des tailles relativement petites avec les grandes instances traitées par les algorithmes de partitionnement de graphe où c'est de l'ordre du milliard de tâches. On peut dans une stratégie spécifique passer plus de temps sur certaines routines. Ainsi, le socle de la stratégie Specific est basé sur la stratégie Quality où est rajouté le post-traitement de la méthode EX pour avoir un meilleur load balancing. La limite inférieure du ML externe a été abaissée pour pouvoir être appliquée aux graphes des tâches des réseaux électriques. Ceci est pour accélérer l'algorithme de partitionnement de graphe. Le changement le plus impactant a été l'augmentation du nombre d'itérations du GPA pour la bipartition. En effet, avec davantage d'itérations, on peut espérer que le GPA trouve une meilleure solution.

Le dernier paramétrage est de jouer sur la **variabilité de la graine aléatoire (seed)**. Pour les algorithmes aléatoires, il est d'usage de rendre déterministe leur résultat en introduisant un seed qui va émuler un tirage aléatoire qui est toujours le même. Le seed est principalement utilisé par le GPA qui commence sur un choix de sommet fait au hasard. En faisant varier la valeur du seed, un plus grand nombre de solutions sont explorées pour le GPA ce qui augmente les chances de trouver une meilleure solution. Cette idée d'itération sur la valeur du seed a également été utilisée par les algorithmes Itératifs multi-niveaux [105].

5.3.4 Validation de l'optimalité avec les méthodes exactes

L'utilisation des méthodes exactes permet de vérifier la qualité des solutions obtenues via des heuristiques. Vérifions cela en réutilisant la formulation en programmation quadratique de (5-8). Afin d'accélérer les calculs, on utilise plutôt une formulation linéarisée pour le calcul des coûts de communication et l'écart à la moyenne au lieu de la variance pour le load balancing. Pour la linéarisation, on introduit une variable supplémentaire de dilution ρ_{ik}^{jl} . Elle vaut 1 si la tâche i est allouée sur le processeur k et la tâche j est allouée sur le processeur l . La architecture des processeurs est considérée comme complète et homogène : quel que soit les processeurs k et l , $c_{kl} = 1$. On obtient alors la formulation de Programmation Linéaire (LP) suivante :

$$\min \frac{1}{M} \sum_{k=1}^M \left| \sum_{i=1}^n w_i x_{ik} - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^n w_i x_{ij} \right| + \sum_{i=1}^n \sum_{j>i}^M \sum_{k=1}^M \sum_{l \neq k}^M d_{ij} \rho_{ik}^{jl} \quad (5-15)$$

Soumis aux contraintes suivantes :

$$\forall (i, j), (k, l) \in ([1, n] \times [1, m])^2, \rho_{ik}^{jl} \geq x_{ij} + x_{kl} - 1 \quad (5-16)$$

$$\forall i \in [1, n], \sum_{j=1}^M x_{ij} = 1 \quad (5-17)$$

$$\forall j \in [1, M], \sum_{i=1}^n w_i x_{ij} \leq w \quad (5-18)$$

On peut diviser la fonction objectif en deux parties suivant le critère de load balancing (*LIR*) et celui des coûts de communication (*CC*).

$$LIR = \frac{1}{M} \sum_{k=1}^M \left| \sum_{i=1}^n w_i x_{ik} - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^n w_i x_{ij} \right| \quad (5-19)$$

$$CC = \sum_{i=1}^n \sum_{j>i}^M \sum_{k=1}^M \sum_{l \neq k}^M d_{ij} \rho_{ik}^{jl} \quad (5-20)$$

Pour étudier les solutions optimales de cette optimisation multi-objectifs, on trace le front de Pareto qui permet de varier l'importance relative de chaque critère. Il permet d'avoir une limite inférieure et d'apprécier la qualité d'une solution approchée vis-à-vis d'un optimum. Une méthode pour tracer le front de Pareto est de pondérer, via des coefficients α et β , l'importance de chaque critère dans la fonction objectif. Elle se réécrit sous la forme :

$$\min_{\alpha+\beta=1} \alpha \times LIR + \beta \times CC \quad (5-21)$$

Ainsi, il suffit de résoudre le problème linéaire en faisant varier α de 0 à 1 pour obtenir l'ensemble des points du front de Pareto.

5.4 Etudes des Performances

5.4.1 Etudes comparatives des heuristiques de partitionnement de graphe sur de grandes instances

Pour mesurer les performances, les différentes heuristiques de partitionnement de graphes sont testées sur des cas de parallélisation de grands réseaux réalistes : recherche A*, méthode spectrale de Chaco [107], multi-niveaux RB de Scotch [83], multi-niveaux amélioré de KaFFPa et le multi-niveaux évolutionnaire de KaFFPaE [108]. Les cas de test sont issus de réseaux de transport d'électricité nationaux. Par exemple dans la Figure 5-7, l'ensemble du réseau de transport français (400 kV + 225 KV) est divisé en plusieurs milliers de tâches. Ce type d'exemple est directement importé des outils de planification de réseau [55].

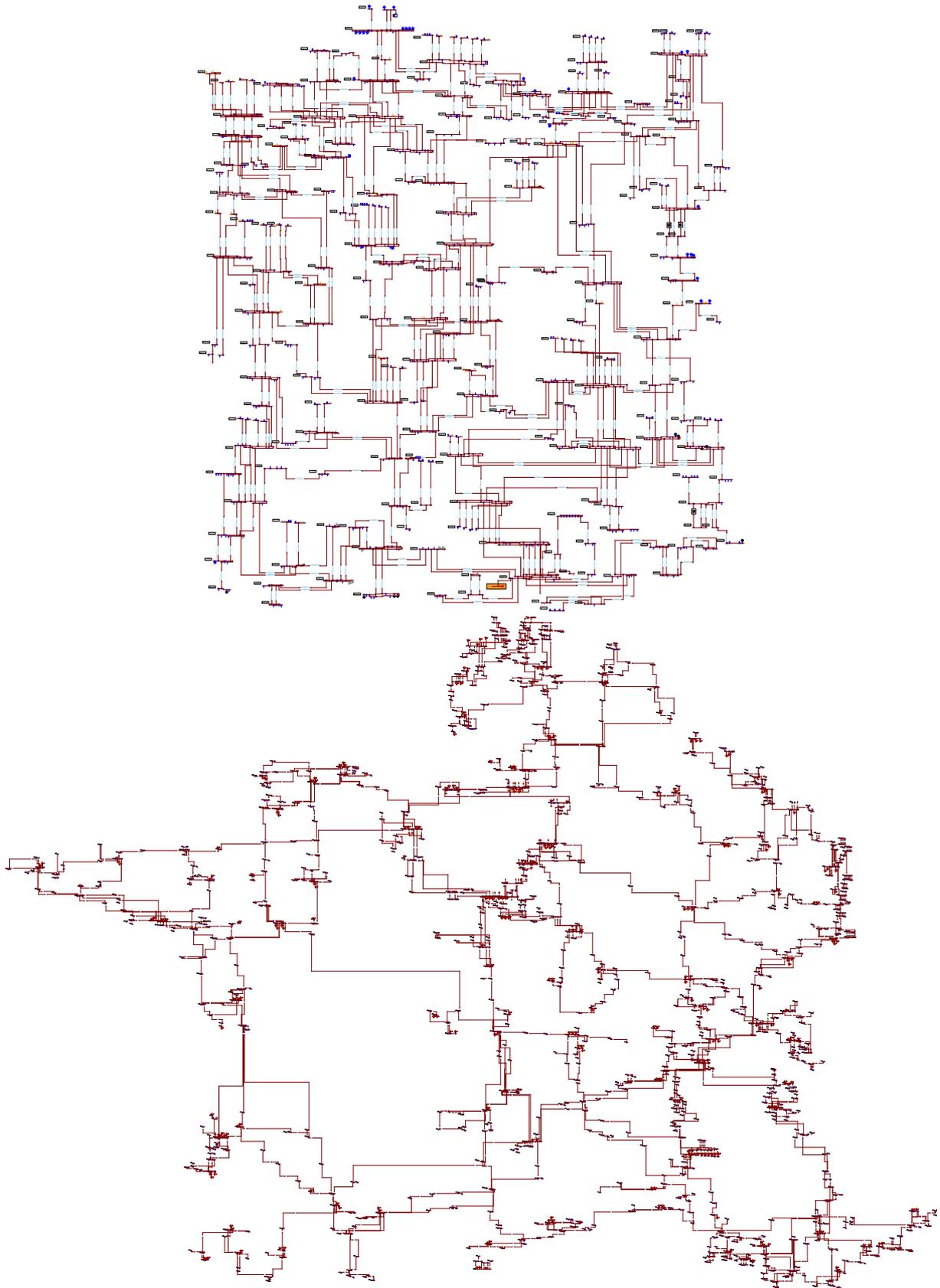


Figure 5-7 : Réseaux français 400 KV (haut) et 225 KV (bas).

Au niveau des architectures simulation, si la simulation tourne sur un seul ordinateur (pas de clusters de machines), le graphe des processeurs est complet : un lien de communication existe entre chaque processeur. En revanche, il n'est pas toujours homogène (même coût de communication entre

chaque processeur). L'architecture SGI UV1000 [109] (96 processeurs au total) est un simulateur temps réel utilisé dans le laboratoire temps réel de RTE. Il contient quatre châssis qui ont alternativement une à deux blades chacun. Une blade (serveur lame) contient deux sockets/CPU avec huit cœurs/processeurs chacun. Les communications inter-châssis seront plus importantes que les communications inter blade. Puis, les communications inter-blades le seront plus que les communications inter-sockets, idem pour les communications inter-cœurs. Cette architecture est hétérogène.

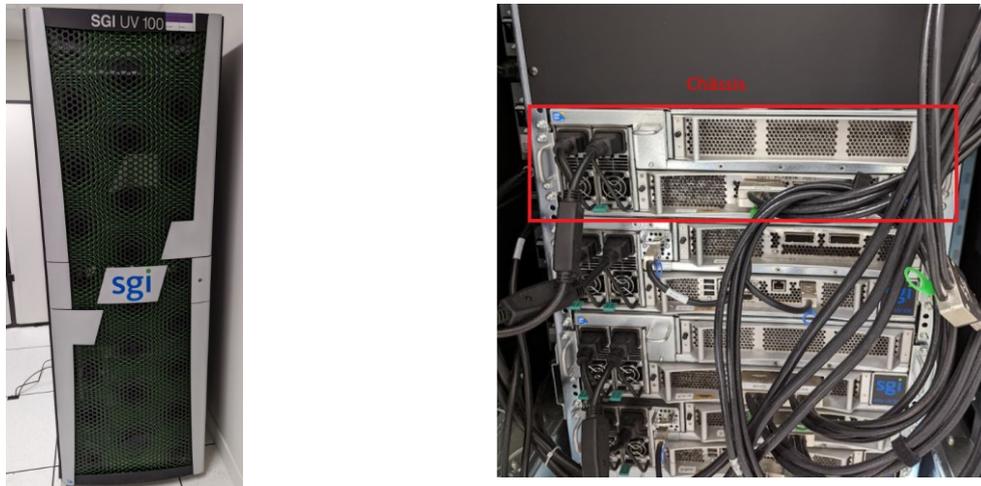


Figure 5-8 : Aperçu d'une architecture SGI UV100 dans le laboratoire temps réel de RTE.

Le Tableau 5-1 présente les performances obtenues de la répartition des tâches de deux grands réseaux pour la recherche A* et le multi-niveaux RB (Scotch avec la stratégie balance, $\delta = 0.01$) sur l'architecture UV100 avec un pas de temps fixé à 40 μ s. Pour quantifier les performances, les métriques suivantes sont utilisées : **NbTask**, le nombre de tâches ; **NbProc**, le nombre de processeurs utilisés ; **Comm**, le nombre total des communications entre tâches, qui correspond au nombre de signaux échangés multiplié par le coût de communication inter-processeur (respectivement 10, 16, 43 et 65 pour les communications inter-cœurs, sockets, blades et châssis) ; **Var**, variance de la répartition de la charge de calcul sur les processeurs ; **Time**, le temps d'exécution en secondes pour répartir les tâches (performances mesurées sur un PC avec un CPU Intel i7-4910MQ @ 2.90GHz).

Tableau 5-1 : Performances pour de grands réseaux pour A* et le RB multi-niveaux de Scotch.

Cas	NbTask	Mapping Strategy	NbProc	Commm	Var	Time
Réseau français 400 KV	460	A*	16	15096	4.93	0.98
		Scotch	16	6960	0.14	0.06
Réseau français 400 kV + 225kV	1510	A*	79	148974	14.35	81.1
		Scotch	79	123216	0.34	0.17

Sur de grandes instances, la technique de partitionnement de graphe (multi-niveaux RB) est beaucoup plus rapide que la recherche A*. On voit l'efficacité de la méthode multi-niveaux qui permet de réduire la taille du graphe à considérer pour la bipartition. Le temps d'exécution est très rapide en n'excédant pas les quelques secondes. Par ailleurs, la solution est bien meilleure sur les deux critères d'optimisation (load balancing et coût des communications).

Le Tableau 5-2 ci-dessous montre les résultats de la répartition des tâches sur une architecture de simulation homogène pour plusieurs heuristiques de partitionnement de graphe pour le cas du réseau français 400 kV à 460 tâches. Toutes les méthodes donnent rapidement une solution. Cependant, la méthode spectrale donne une solution de moins bonne qualité, surtout pour la partie communication. Le multi-niveaux RB est performant pour optimiser le load balancing tout en ayant

un coût limité sur les communications. L'algorithme évolutionnaire est le plus performant pour ce qui concerne l'optimisation des coûts de communication.

Tableau 5-2 : Performances des méthodes de partitionnement de graphe pour la répartition des tâches du réseau français 400 KV.

Méthodes	Solveur	Configuration	NbProc	Comm	Var	Time
Multi-niveaux RB	Scotch	Stratégie Balance + $\delta = 0.01$	16	610	0.07	0.13
Spectral	Chaco	Méthode de Lanczos + Bisection	16	1256	3.04	0.001
Multi-niveaux Amélioré	KaFFa	Strong configuration	16	510	1.30	0.20
Multi-niveaux Evolutionnaire	KaFFaE	Strong configuration	16	510	1.15	0.36

5.4.2 Recherche du paramétrage optimal pour le partitionnement de graphe multi-niveaux

Les différents paramétrages (Imbalance Load Ratio, stratégie Specific et itération sur le seed) ont été testés sur un benchmark issu de la duplication d'un réseau à 35 bus (Figure 5-9). Au total, le graphe des tâches comptabilise 802 sommets et 2052 arcs (332 bus, 513 lignes et 42 transformateurs). Les tâches sont réparties sur une architecture complète et homogène de 19 processeurs. Le pas de temps est fixé à 40 μ s. Le nombre minimum de processeurs requis est de 16. Il correspond au temps total estimé d'exécution de l'ensemble des tâches sur le pas de temps.

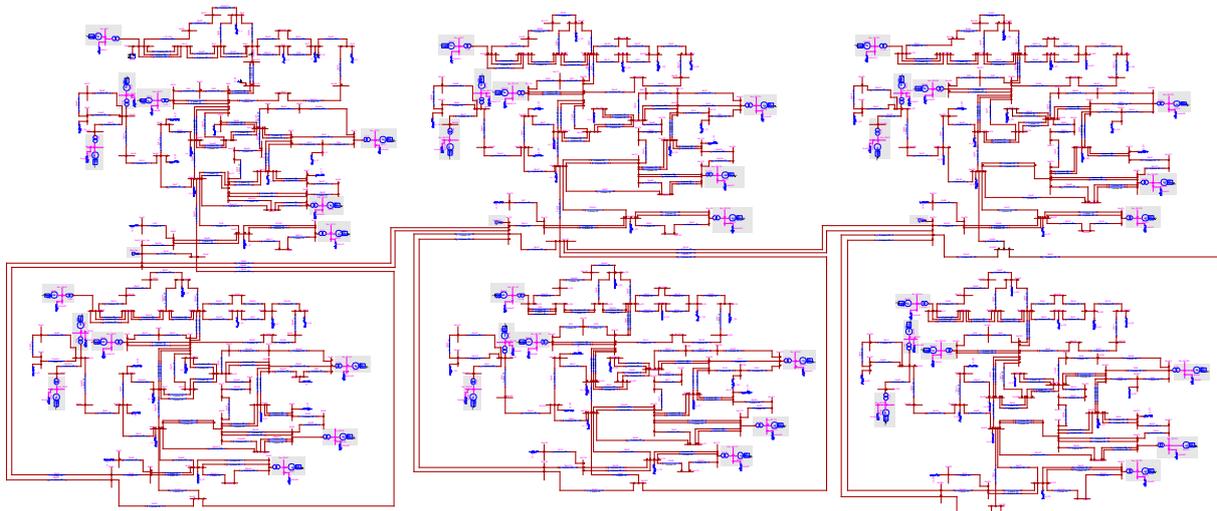


Figure 5-9 : Benchmark d'un réseau à 802 tâches à répartir.

Le Tableau 5-3 montre les résultats en faisant varier la contrainte de load balancing δ et les stratégies en incluant la nouvelle stratégie définie (Specific). Le symbole – signifie qu'aucune solution admissible à la contrainte temporelle n'a été trouvée pour le nombre de processeurs fixé. On constate que le nombre de processeurs utilisés diminue avec δ . En revanche, le nombre de communications augmente. En effet, on a plutôt privilégié le critère du load balancing. De plus, on voit que pour ce cas précis la stratégie Specific a amélioré la qualité des solutions trouvées sur les deux critères d'optimisation, que ce soit pour les communications ou le load balancing. L'utilisation de la méthode EX a bien aidé pour répartir la charge sur les processeurs tout comme l'augmentation des itérations du GPA qui optimise davantage le coût des communications.

Tableau 5-3 : Résultats de la variation du δ et de l'utilisation de la stratégie Specific.

δ	Stratégie	NbProc	Comm	Var	Time
0.25	Quality	19	390	8.874	0.036
	Balance	-	-	-	-
	Specific	18	396	4.979	0.029
0.10	Quality	18	390	4.644	0.036
	Balance	18	402	4.004	0.033
	Specific	18	384	3.647	0.036
0.01	Quality	17	450	0.463	0.023
	Balance	17	480	0.032	0.018
	Specific	17	444	0.040	0.026

Les résultats des itérations sur le seed sont affichés sur le Tableau 5-4 en fixant $\delta = 0.25$. **NbIter** est le nombre d'itérations avant de trouver une répartition admissible. On a fixé le nombre maximal d'itérations sur le seed à 10 avant d'augmenter le nombre de processeurs pour trouver une solution admissible. Quand **NbIter** vaut \emptyset , cela correspond au cas sans itération avec le seed que l'on retrouve sur le tableau précédent. **NbIter** = 26 pour la deuxième ligne signifie que la solution a été trouvée à la sixième itération du seed après avoir utilisé un processeur supplémentaire. Pour ce cas, les itérations sur le seed ont amélioré l'ensemble des critères de l'optimisation, **NbProc**, **Comm** et **Var** pour les stratégies Balance et Quality. On trouve sensiblement la même solution pour la stratégie Specific. Ainsi, l'itération sur le seed est une possibilité pour améliorer la solution mais cela ne garantit pas toujours une meilleure solution.

Tableau 5-4 : Résultats des itérations sur le seed.

Stratégie	NbProc	NbIter	Comm	Var	Time
Quality	19	\emptyset	390	8.874	0.036
Quality	18	26	372	6.435	0.257
Balance	-	\emptyset	-	-	-
Balance	19	32	396	12.13	0.202
Specific	18	\emptyset	396	4.979	0.029
Specific	18	21	396	5.417	0.167

5.4.3 Validation de la qualité des solutions obtenues

Un réseau simple de 35 bus (Figure 5-10), disponible dans [1], est utilisé pour obtenir assez rapidement les solutions optimales de la méthode exacte formulée en LP afin de tracer le front de Pareto. Il est divisé en 103 tâches qui doivent être réparties sur 3 processeurs ce qui fait tout de même 10^{49} allocations possibles.

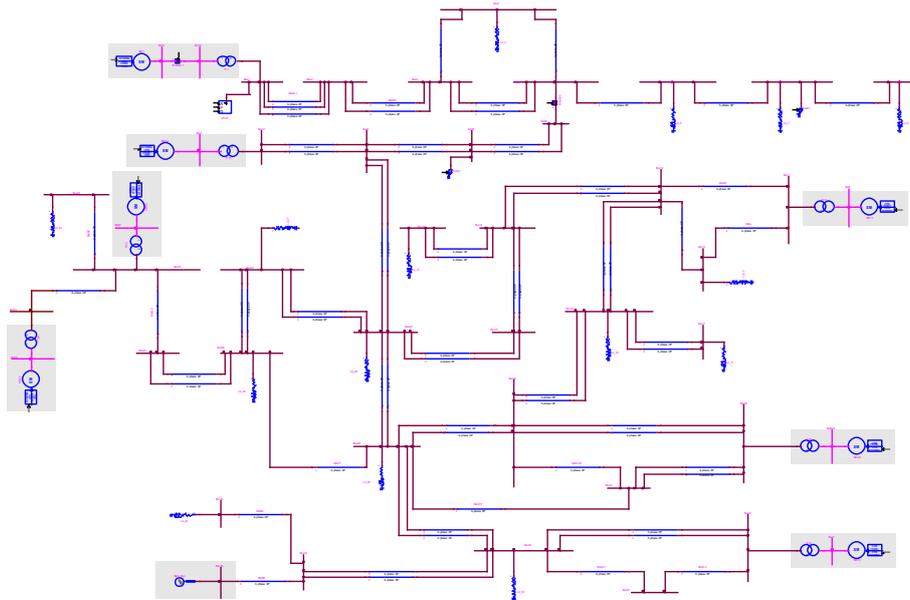


Figure 5-10 : Cas test d'un réseau à 35 bus.

Pour résoudre le problème linéaire, le langage de modélisation mathématique AMPL [110] est choisi combiné avec le solveur parallèle FICO™ Xpress [111]. Un cluster de 190 cœurs est utilisé pour la résolution. La Figure 5-11 montre la comparaison entre le front de Pareto et les résultats de l'heuristique de partitionnement (multi-niveaux RB de Scotch). Les points en bleu sont les solutions exactes de la formulation LP. Comme le problème d'allocation est discret, le front de Pareto n'est composé que de quatre points. Les valeurs entre parenthèse sont les poids (α, β) utilisés dans l'expression (6-45). Quand $\alpha \leq 0,999$, l'ensemble des solutions exactes sont confondues au même point bleu en bas à droite. Les différentes tailles pour les marqueurs représentent des valeurs différentes pour δ , de la plus grande $\delta = 0.25$ à la plus petite $\delta = 0.01$ en passant par les tailles intermédiaires $\delta = 0.1$ et $\delta = 0.05$. Chaque forme de marqueur est associée à une stratégie utilisée : le diamant pour Balance, le cercle pour Specific et la croix pour Quality. Lorsqu'on compare les solutions des heuristiques avec celles exactes du front de Pareto, on remarque que ces dernières ont tendance à prioriser la contrainte LIR. Le coût de communication est légèrement détérioré pour se rapprocher d'une répartition de la charge proche de l'optimum.

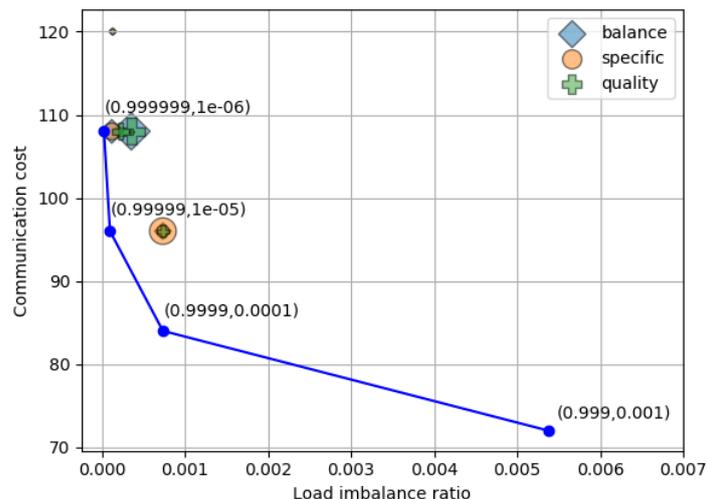


Figure 5-11 : Front de Pareto pour le réseau à 35 bus.

Pour mesurer la distance entre le front de Pareto et les solutions approchées, la métrique de la plus petite distance euclidienne (distance classique entre un point et une courbe) est utilisée :

$$\min_{p \in PF} \|p - v\|_2 \quad (5-22)$$

Où p est un point de PF , le front de Pareto, et v une solution approchée d'une heuristique. Le Tableau 5-5 ci-dessous montre les distances calculées de l'ordre de 10^{-4} . Tant que la contrainte de load balancing est faisable (δ pas trop petit), les solutions sont très proches d'une solution optimale.

Tableau 5-5 : Distance au Front de Pareto pour les solutions des heuristiques pour le réseau à 35 bus.

δ	Balance	Specific	Quality
0.25	3.30	6.37	3.30
0.10	0.95	0.95	2.01
0.05	6.37	6.37	6.37
0.01	120000	120000	3.30

Cette comparaison avec les solutions exactes a confirmé la bonne qualité des solutions issues de l'heuristique RB multi-niveaux qui met l'accent sur une bonne répartition de la charge sur les processeurs. Ce critère reste primordial pour diminuer le temps de simulation et in fine respecter la contrainte temps réel.

5.4.4 Performances temps réel

Les études précédentes s'attardent à montrer l'efficacité des heuristiques de partitionnement de graphe sur la qualité des solutions. Cependant, le temps d'exécution de chaque tâche est une valeur souvent estimée qui n'est pas forcément systématiquement vérifiée lors de la simulation temps réel. Il va falloir vérifier les performances effectives de la répartition lors de simulation temps réel. Deux exemples HVDC vont permettre de valider les performances et montrer l'intérêt pratique d'avoir un répartiteur automatique des tâches.

5.4.4.1 Cas réseau HVDC CIGRE

Au vu du développement des réseaux DC avec la multiplication des dispositifs d'électronique de puissance, le Comité d'Etude B4 du CIGRE a lancé en 2011 la mise en place d'un cas test de référence pour étudier la dynamique des réseaux DC maillés, le benchmark réseau DC CIGRE B4 [112]. Ce réseau (Figure 5-12) est à base de technologie VSC-MMC pour les convertisseurs AC/DC et XLPE [113] (polyéthylène réticulé) pour l'isolation des câbles DC. Au total, il est composé de :

- 11 Stations de conversion AC/DC VSC (6 en configuration monopôle, indice m , 5 en bipôle respectivement, indice b),
- 5 lignes aériennes DC (1500 km),
- 8 lignes souterraines DC (1600 km),
- 6 lignes aériennes AC (1200 km),
- 1 ligne sous-marine AC (50 km).

Ce réseau a été initialement monté dans [2] et simulé en temps différé [112][114]. L'objectif est d'appliquer les algorithmes de répartition de tâches pour pouvoir le simuler en temps réel avec un pas de temps de 20 μ s.

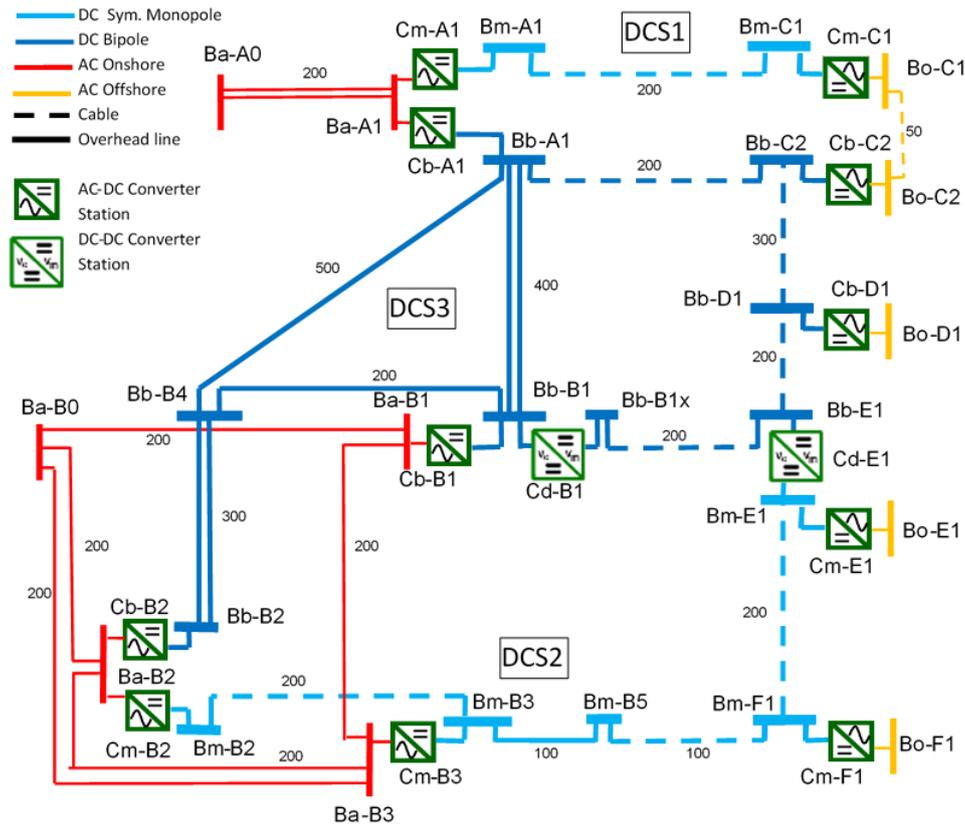


Figure 5-12 : Vue d'ensemble du réseau DC CIGRE B4.

Pour la modélisation des stations de conversion, le Model#3 [115], entièrement simulé sur CPU, est utilisé ici. Le Model#2 [115], simulé sur CPU et sur FPGA, peut également être employé. Des modèles fréquentiels [27] sont utilisés pour les lignes AC et DC. Un contrôleur générique, importé depuis Simulink [7], contrôle les stations de conversion. Par la suite, selon les besoins d'étude temps réel, ce contrôleur peut être remplacé par un contrôleur physique (réplique) pour gagner en précision [21]. La parallélisation selon les lignes de transmission sépare le réseau en 53 tâches. Voici une description des tâches principales :

- 5 pour les stations de conversion en configuration bipôle,
- 6 pour les stations de conversion en configuration monopôle,
- 16 pour les contrôles des stations de conversion,
- 15 pour les lignes DC,
- 6 pour les lignes AC,
- 5 pour les autres postes électriques ; 2 postes AC, 1 pour la synchronisation des signaux et deux postes DC (Bm-B5 et Bb-B1x).

Les méthodes de répartition des tâches suivantes sont testées : la recherche A*, le RB Multi-niveaux de Scotch avec les stratégies Balance/Quality et une répartition manuelle. Pour la répartition manuelle, les règles suivantes ont été utilisées pour essayer de bien répartir la charge de calcul sur les processeurs :

- (1 station de conversion en bipôle + 1 station de conversion en monopôle + ligne AC ou DC) /processeur
- 3 contrôles de convertisseur / processeur
- 4 lignes DC / processeur

- Regroupement des contrôles des bipôles, par exemple, Cb-C2 Control réunit les contrôles de chaque pôle du bipôle Cb-C2, Cb-C2_1 Control et Cb-C2_2 Control.

Le Tableau 5-6 ci-dessous présente les résultats des méthodes testées pour la répartition des tâches. L'architecture utilisée pour la simulation temps réel, sur laquelle sont réparties les tâches, est une OP5031 [59] avec 32 cœurs (2 CPU Intel Xeon E5-2667 v3 @ 3.2GHz – 16 cœurs). Cette architecture est complète et homogène. On retrouve les mêmes colonnes que précédemment (NbProc, Var...) Le temps d'exécution (Time) des algorithmes a été mesuré sur un PC avec un CPU Intel i7-4910MQ @ 2.90 GHz. On constate que la recherche A* privilégie la minimisation de coût de communication au détriment du Load balancing qui est le pire par rapport aux autres méthodes. Ce dernier critère ne permet pas de tenir la contrainte temps réel pour la recherche A*. On privilégie le Load Balancing avec la méthode RB Multi-niveaux Balance qui est choisie par la suite. Même si la répartition manuelle présente de bons résultats sur les deux critères d'optimisation, elle reste moins flexible par rapport à la méthode choisie, qui permet d'ajuster le nombre de processeurs vis-à-vis des erreurs d'estimation de la charge de calcul des tâches. Le Tableau 5-9 présente la répartition des tâches pour ces deux méthodes (répartition manuelle et RB Multi-niveaux Balance).

Tableau 5-6 : Résultats des différentes méthodes de répartition des tâches.

Méthodes	NbProc	Time	Var	Comm
Recherche A*	13	0.0293	10.43	1191
RB Multi-niveaux Balance	15	0.00414	0.28	1378
RB Multi-niveaux Quality	15	0.00388	1.40	1215
Répartition Manuelle	15	∅	0.87	1382

Tableau 5-7 : Aperçus de la répartition des tâches pour la méthode manuelle et la méthode automatique retenue.

Répartition Manuelle			RB Multi-niveaux Balance		
Proc 1 Cb-C2	Proc 6 Cb-C2 Control Cm-C1 Control	Proc 11 Cm-B3 Control 2 DC lines	Proc 1 Cb-B1 Cm-B3 1 AC line	Proc 6 Cb-C2	Proc 11 Cm-F1 Control Cb-A1_1 Control Cb-A1_2 Control 1 synchronization substation
Proc 2 Cb-B1 Cm-E1 2 DC lines	Proc 7 Cb-D1 Control Cm-B2 Control	Proc 12 Cm-B3 Cm-C1 2 AC substations 2 DC substations 1 AC line 1 synchronization substation	Proc 2 Cb-A1 1 DC line	Proc 7 Cm-A1 Cm-B2 Control 2 AC lines 1 DC line	Proc 12 Cb-D1 2 DC lines
Proc 3 Cb-A1 Cm-A1 1 AC line	Proc 8 Cb-B1 Control Cm-E1 Control	Proc 13 3 DC lines	Proc 3 Cb-B2_1 Cm-E1 1 AC line	Proc 8 Cm-C1 Control Cb-C2_2 Control Cb-D1_1 Control 1 DC line	Proc 13 Cm-C1 1 AC substation 2 DC lines 1 AC line
Proc 4 Cb-B2 Cm-B2 1 AC line	Proc 9 Cb-B2 Control Cm-F1 Control	Proc 14 3 DC lines	Proc 4 Cb-B2 control Cm-B3 control Cm-A1 control 1 DC substation	Proc 9 Cb-C2_1 control Cb-D1_2 control Cm-E1 control 1 DC line	Proc 14 1 DC substation 2 DC lines 1 AC lines
Proc 5 Cb-D1 Cm-F1 1 AC line	Proc 10 Cb-A1 Control Cm-A1 Control	Proc 15 5 DC Lines 2 AC lines	Proc 5 Cm-F1 Cm-B1 1 AC substation 1 DC line	Proc 10 Cb-B1_1 control Cb-B1_2 control Cb-B2_2 control 1 DC line	Proc 15 3 DC lines

Avec la répartition de RB Multi-niveaux Balance, le Tableau 5-8 montre que la contrainte temps réel est tenue en régime permanent pour l'ensemble des processeurs.

Tableau 5-8 : Temps d'exécution temps réel en régime permanent lors d'un pas de temps pour chaque processeur.

Processors	Exec Time (μ s)	Exec Time Max (μ s)
1	12.43	18.24
2	10.62	12.33
3	11.65	16.17
4	11.68	12.87
5	10.13	10.93
6	13.98	14.92
7	10.48	11.29
8	12.55	13.31
9	12.41	13.39
10	11.92	13.0
11	11.47	12.49
12	11.26	15.27
13	8.79	9.41
14	9.26	9.87
15	10.86	11.46

Pour valider le modèle lors des transitoires, le scénario suivant est simulé en temps réel :

- t = 1.5 s, défaut fugitif pôle-terre à la station Bb-A1 pendant 100 ms,
- t = 1.503 s, ouverture de la ligne en défaut,
- t = 1.7 s, fermeture de la ligne avec le défaut éliminé,
- t = 3 s, trip permanent du pôle négatif de Cb-C2 (blocage de la station et ouverture des disjoncteurs AC),
- t = 5 s, défaut pôle-terre pour le pôle positif de Bm-A1 ce qui provoque le trip permanent de Cm-A1 et de Cm-C1,
- t = 6 s, défaut triphasé avec la terre à Ba-B2 de 150 ms.

La Figure 5-13 montre que, lors de la simulation durant 7 s du scénario précédent, la contrainte temps réel est tenue. Le temps d'exécution ne dépasse pas les 20 μ s du pas de temps pour les cinq processeurs les plus chargés en temps de calcul, contenant des tâches de stations de conversion en configuration bipôle (Cb-A1, Cb-C2, Cb-D1, Cb-B1 et Cb-B2).

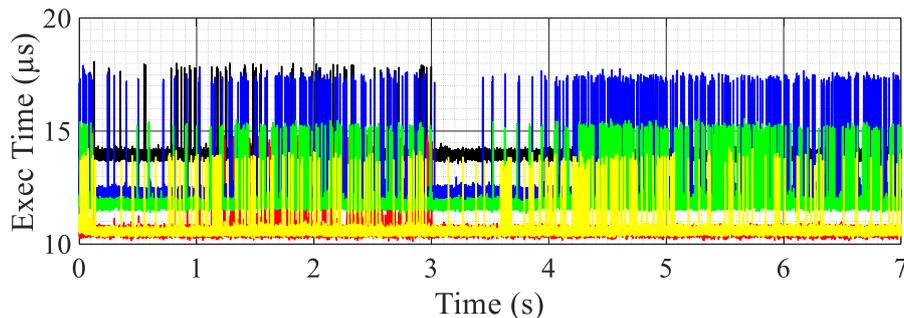


Figure 5-13 : Temps d'exécution pour les processeurs des stations de conversion en configuration bipôle.

Dans la validation des différents événements transitoires du scénario de simulation, on compare la réponse du temps différé (offline, trait plein) avec celle du temps réel (real-time, trait en pointillé). HYPERSIM est utilisé pour les deux environnements de simulation. Le défaut de la station Bb-A1 à t = 1.5 s est bien éliminé comme le montre le courant DC de Cb-A1 (Figure 5-14). Ce dernier se stabilise à sa valeur initiale.

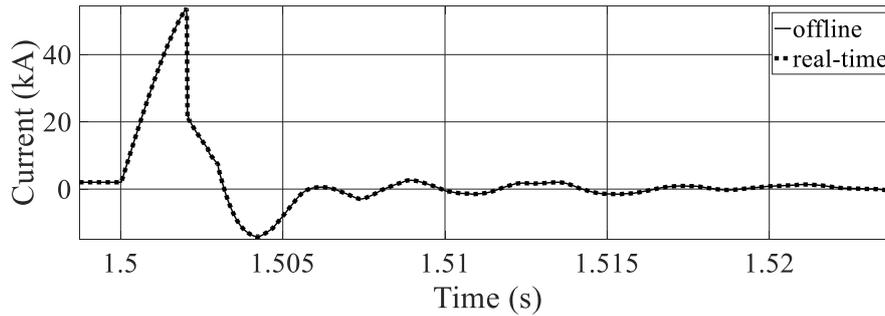


Figure 5-14 : Courant DC dans le pôle positif de Cb-A1.

Le trip permanent de Cb-C2 à $t = 3$ s va être compensé au niveau du transit de la puissance par la configuration en bipôle des stations de conversion Cb-C2 et CB-C1 (Figure 5-15). Le pôle négatif de Cb-C2 est hors d'usage en ne transitant aucune puissance. Le report de charge s'effectue sur le pôle négatif de Cb-A1 qui augmente sa puissance transitée.

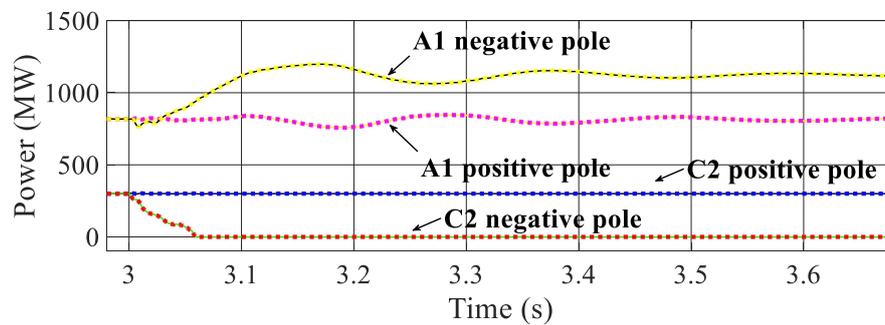


Figure 5-15 : Transits de puissance aux pôles de Cb-A1 et Cb-C2 après le trip permanent de pôle négatif de Cb-C2.

Lors du trip permanent de Cm-A1, suite au défaut de Bm-A1 à $t = 5$ s, la surtension a bien été limitée à 1.8 pu ($V_{dc_ref} = \pm 200$ kV) par les parafoudres de liaison, comme en témoigne la tension DC du pôle négatif de Cm-A1 sur la Figure 5-16. La liaison HVDC Cm-A1 – Cm-C1 est déconnectée du réseau DC via les lignes AC afin de limiter l'impact sur le reste du réseau.

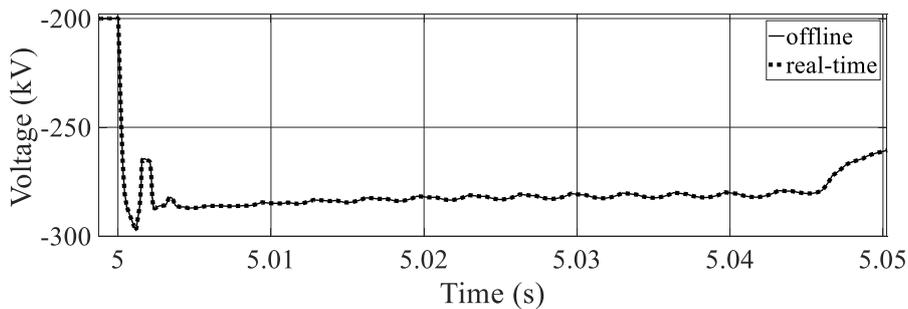


Figure 5-16 : Tension DC du pôle négatif de Cm-A1.

Le défaut AC triphasé de Ba-B2 à $t = 6$ s bloque les stations Cb-B2 et Cm-B2 mais sans trip : le transit de puissance revient à son état initial. La Figure 5-17 montre les tensions AC à Ba-B2 juste après l'élimination du défaut.

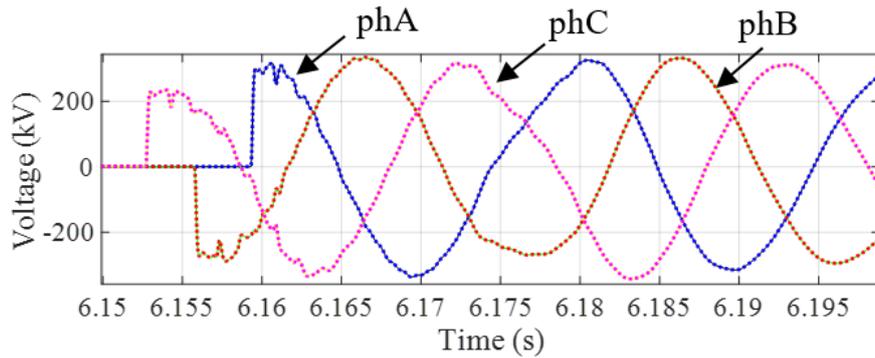


Figure 5-17 : Tensions AC à Ba-B2.

Au final, pour l'ensemble des évènements transitoires, on obtient une bonne superposition des réponses en temps réel et en temps différé. Cela valide la répartition des tâches choisie, RB Multi-niveaux Balance, qui permet de faire tourner en temps réel ce réseau DC de référence.

5.4.4.2 Cas réseau HVDC projet Best Paths

Un autre cas industriel de simulation HIL va permettre de valider l'efficacité du répartiteur des tâches basé sur le RB multi-niveaux de Scotch. Ce cas a été monté pour la DEMO#2 du projet de recherche Européen Best Paths [116]. Il contient un réseau HVDC à trois terminaux comportant des disjoncteurs DC (DCCBs) [117] comme le montre la Figure 5-18. Les stations de conversion AC/DC sont représentées par les Model#2 et Model#3 de MMC [115]. Les Model#3 sont simulés sur CPU alors que le Model#2 est partiellement simulé sur CPU et sur FPGA [93]. Deux stations MMC (Station 2 and 3) sont contrôlées par un contrôleur générique développé sous Simulink et importé dans [4] via l'interface EMT-Simulink [7]. Le dernier convertisseur (Station 1) est contrôlé par un contrôleur industriel fournis par ABB [118]. Les lignes DC sont composées de deux sections de câble de 32 km et de 100 km et d'une partie aérienne de 40 km. Un modèle dépendant de la fréquence est utilisé pour les représenter [27].

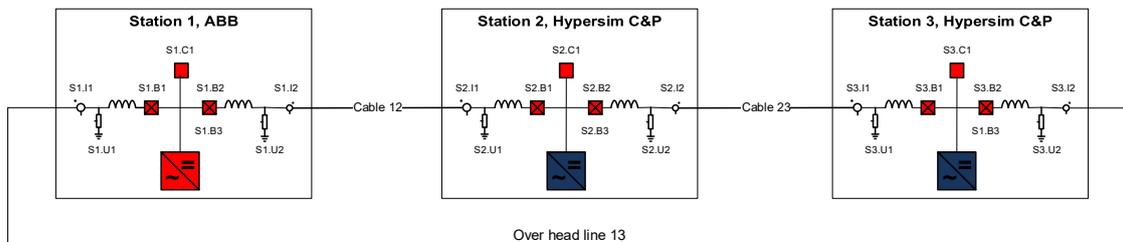


Figure 5-18 : Réseau DC à trois terminaux.

L'objectif de la plateforme HIL mise en place est de tester l'efficacité des algorithmes de protection DC, ainsi que le contrôle des DCCBs dans un réseau DC suite à différents défauts. Les résultats détaillés de ces tests sont présentés dans [119].

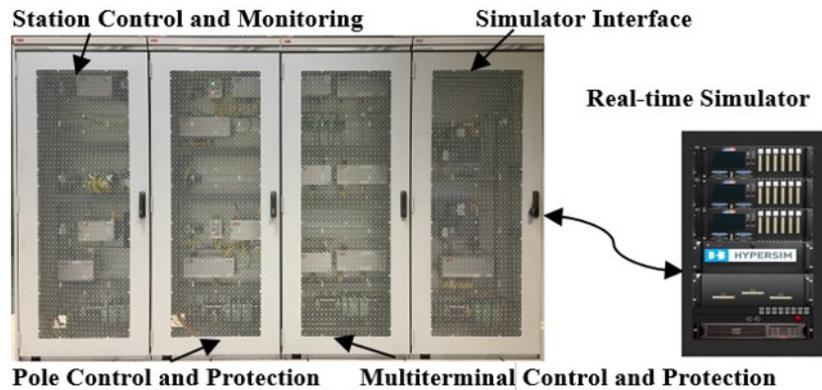


Figure 5-19 : Plateforme de simulation temps réel HIL avec le contrôleur physique (réplique) fourni par ABB.

La parallélisation du réseau DC le divise en 85 tâches à répartir sur les processeurs. Ci-dessous une liste par ordre d'importance au niveau du temps d'exécution des tâches :

- 3 tâches pour les stations de conversion avec le disjoncteur DC,
- 1 tâche pour le système de contrôle de deux convertisseurs,
- 6 tâches pour les lignes DC (deux sections pour chaque ligne DC),
- Le reste des tâches est dédié à la communication des IOs pour la réplique (disjoncteur DC et contrôle du VSC) pour le modèle des valves MMC sur FPGA.

Les trois stratégies de Scotch, présentées dans 5.3.3, ont été testées avec un imbalance ratio $\delta = 0.01$. Le pas de temps est fixé 30 μ s. La dernière colonne du Tableau 5-9 contient le temps d'exécution en régime permanent du processus le plus chargé en temps de calcul. L'architecture d'un OP5031 [59] avec 32 cœurs (2 CPU Xeon E5-2697A v4 @ 2.60GHz - 16 cœurs) a été utilisée pour les simulations temps réel. Seules les stratégies Balance et Specific arrivent à respecter la contrainte temps réel. Il semble que favoriser le load balancing sur le coût des communications permet de mieux gérer les erreurs d'estimation sur le temps d'exécution des tâches. La répartition de la stratégie Balance est gardée par la suite pour simuler les transitoires afin de valider le bon fonctionnement du DCCB.

Tableau 5-9 : Résultats de la répartition des tâches pour chaque stratégie de Scotch.

Strategy	NbProc	Comm	Var	Time	Max RT ExecTime
Quality	6	375	18.47	0.0028	31.5
Balance	6	524	0.09	0.0029	24.0
Specific	6	522	0.27	0.0113	24.0

Le scenario de test consiste en un défaut permanent pôle-terre sur le câble DC le plus court. Le défaut apparaît à $t = 200$ ms. La Figure 5-20 montre que, pendant les transitoires, la contrainte du pas de temps de 30 μ s est respectée pour les trois processeurs les plus chargés en calcul (chaque processeur exécute une tâche de station de conversion). On voit que la stratégie Balance a de meilleures performances que la stratégie Quality comme cela a déjà été constaté en régime permanent.

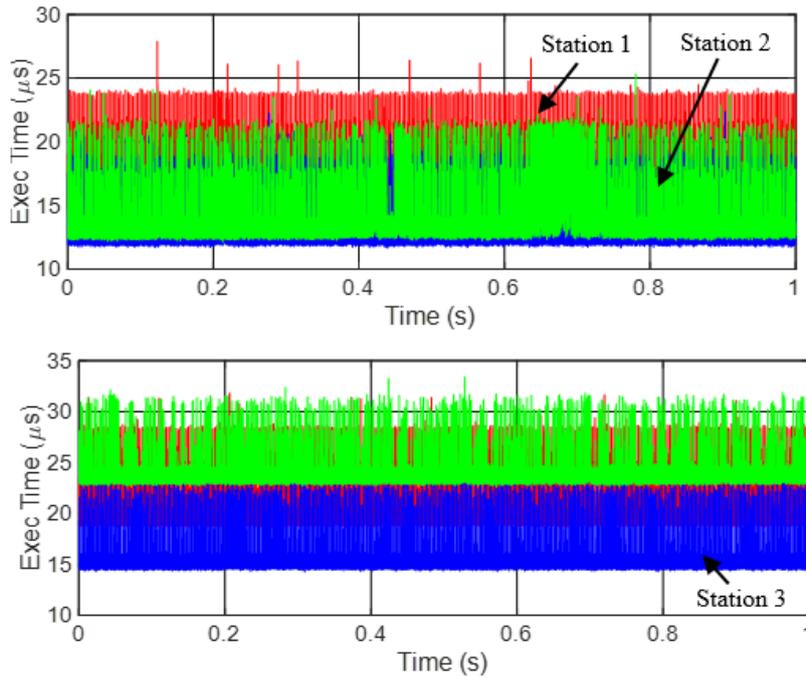


Figure 5-20 : Temps d'exécution pour les stations de conversion durant les transitoires (haut : stratégie Balance, bas : stratégie Quality).

La Figure 5-21 montre que les communications inter-processeur sont négligeables. Cela confirme que, pour ce cas, une bonne répartition de la charge des processeurs est plus importante que la minimisation des communications inter-processeurs. Ceci justifie l'emploi de la stratégie Balance. Cette conclusion ne sera pas forcément valable pour les architectures hétérogènes où les délais de communication peuvent être plus importants.

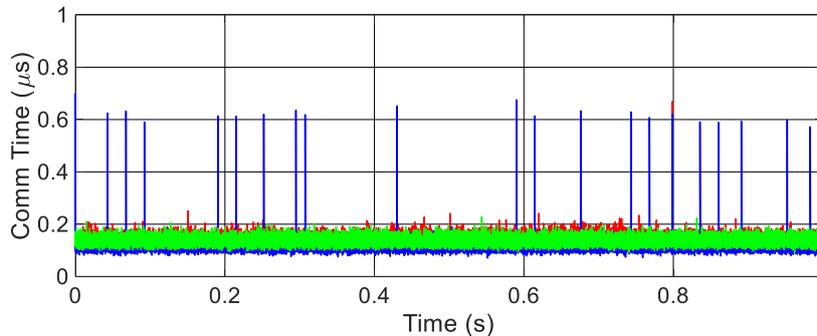


Figure 5-21 : Temps de communications inter-processeur pour chaque station de conversion.

Grâce à l'algorithme de protection DC implémenté dans le contrôleur, le câble en défaut est isolé du reste du réseau DC avec les DCCBs. Après quelques transitoires dus au défaut et aux manœuvres du DCCB, chaque station retourne à son point de fonctionnement initial autour de +/- 320 kV comme le montre la Figure 5-22.

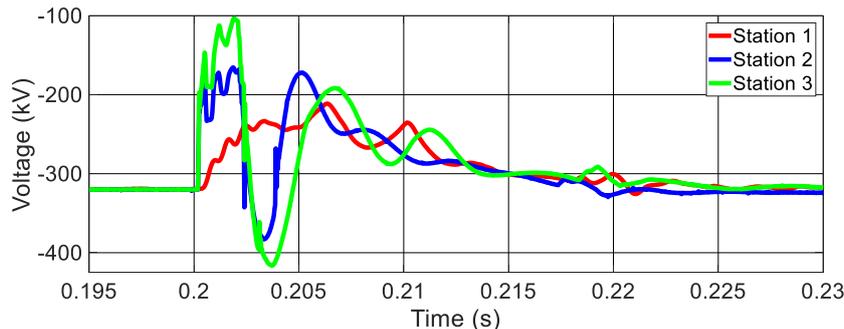


Figure 5-22 : Tensions du pôle négatif de chaque convertisseur qui sont impactées par l'isolation du défaut câble DC.

5.5 Conclusion

Ce chapitre a présenté le problème de répartition des tâches sur les processeurs, élément essentiel de la parallélisation des simulations EMT. Une première contribution est l'identification des méthodes de répartition les plus efficaces, afin de fournir automatiquement une répartition qui répond aux deux critères d'optimisation : une répartition de la charge de calcul la plus équi-répartie possible (load balancing) sur les processeurs et la minimisation des coûts de communication. Les méthodes de partitionnement de graphe multi-niveaux se sont révélées être les plus performantes du fait de leur temps de calcul et de la qualité des solutions obtenues en privilégiant le load balancing. En effet, ce critère est le plus important pour accélérer la simulation et in fine respecter la contrainte temps réel. Il prévaut sur les coûts de communications qui sont de moindre importance sur les architectures modernes multi-cœurs de simulation. Cependant, ces méthodes se basent sur un relâchement de la contrainte temps réel, ce qui peut mener dans certains cas à utiliser davantage de processeurs. La recherche d'un paramétrage optimal de la méthode de partitionnement est une seconde contribution pour améliorer la qualité de la solution. La comparaison avec les méthodes exactes, une première tentative dans le cadre de la répartition des tâches d'une simulation EMT, a montré que, malgré ce relâchement de contraintes, les solutions obtenues restent proches de l'optimum. Des cas pratiques de simulation temps réel de réseaux HVDC, réalistes ou tirés de cas pratiques, ont validé le choix de la méthode retenue pour la répartition des tâches, le RB multi-niveaux. Que ce soit pour le réseau DC benchmark CIGRE ou le cas HIL du projet Best Paths, le RB multi-niveaux offre une méthode de répartition des tâches performantes qui permet de tenir la contrainte temps réel.

Dans sa genèse, ce projet était plutôt porté sur le développement. L'objectif au départ était de mettre en place une répartition manuelle pour donner plus de latitude à l'utilisateur vis-à-vis du répartiteur historique de [1]. Progressivement, le projet a pris une orientation recherche sur l'analyse des différentes techniques de répartition et l'identification de la meilleure méthode. Ce projet a illustré qu'en recherche, on travaille rarement seul en étant nourri par plusieurs échanges et collaborations. Les spécialistes de la simulation temps réel d'Hydro-Québec de [1] ont apporté un éclairage sur les méthodes de répartition et les architectures utilisées pour la simulation. Le pôle d'expertise d'optimisation mathématique de la R&D de RTE a fourni les outils et méthodes pour pouvoir obtenir le front de Pareto. Grâce au projet Best Paths, une plateforme de simulation HIL temps réel a permis de valider sur un cas industriel concret la répartition des tâches. Ce projet a fait l'objet de l'encadrement d'un master recherche. Tout ceci m'a appris à mener un projet de recherche de sa formulation, à sa réalisation, en trouvant les bonnes ressources par le biais d'échanges/collaborations ou de l'encadrement d'un stage.

Chapitre 6 Méthodes d'accélération des résolutions de systèmes de contrôle

6.1 Introduction

La simulation des phénomènes transitoires électromagnétiques sur les réseaux électriques ne serait être précise sans la prise en compte des systèmes de contrôle [40][120]. Régulant les grandeurs électriques (courant, tension...) des équipements (dispositifs d'électronique de puissance, postes électriques...), ils ont un impact important sur la dynamique des réseaux étudiés. La Figure 6-1 ci-dessous illustre le principe de fonctionnement d'une boucle fermée simple de régulation d'un équipement électrique. Le système de contrôle reçoit les mesures des grandeurs électriques du composant et les compare à une consigne (écart). Une commande est actionnée pour amener la mesure de la grandeur de sortie de l'Équipement à la valeur de la consigne. Ce chapitre va s'intéresser à la modélisation et à la simulation numérique des systèmes de contrôle dans un environnement de simulation EMT.

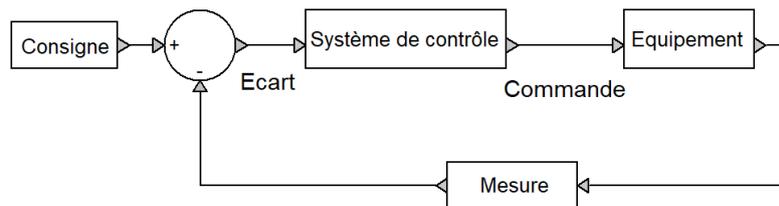


Figure 6-1: Schéma-bloc d'une boucle générale de régulation fermée pour un équipement électrique.

Afin d'avoir une modélisation la plus précise possible, des solutions ont été mises en place. La première consiste à utiliser la génération de code [6][7][53][54] pour avoir une bonne correspondance avec un contrôleur physique. La seconde solution est d'intégrer directement le contrôleur physique dans la boucle de simulation via la simulation temps réel HIL. Les chapitres précédents (Chapitre 4 et Chapitre 5) ont largement illustré ce cas d'application pour les contrôleurs des liaisons HVDC. En complément de ces deux solutions, une modélisation détaillée et précise des systèmes de contrôle reste nécessaire dans les outils EMT. Elle permet de créer des modèles ouverts de contrôle [2][121] et de modéliser des phénomènes physiques (modèle d'arc électrique [122]).

Pour ce faire, la plupart des outils, qui simulent des systèmes de contrôle dans un environnement EMT [1][2] ou hors EMT [6], ont adopté l'approche entrée/sortie schéma-bloc. Ainsi, ce chapitre explicitera les techniques de résolution des équations d'un système de contrôle qui ont été représentées en détail par des schémas-blocs. L'accent sera mis sur les possibilités d'accélération de ces méthodes de résolution. Plus précisément, via l'approche schéma-bloc, un système de contrôle est représenté par un graphe orienté. Si le graphe est acyclique (pas de boucle de rétroaction), la théorie des graphes [39][124] rend la résolution facile : l'ordre topologique du graphe permet de mettre à jour séquentiellement les schémas-blocs qui composent le système de contrôle. Lorsque le graphe est cyclique (présence de boucles de rétroaction), l'ordre topologique va introduire des délais artificiels dans la résolution. Pour le temps réel [1][6][123], cet ordre reste utilisé par la génération de code afin d'avoir une résolution rapide. Cependant, avec l'introduction de délais, la résolution

n'est plus simultanée et peut mener à des instabilités numériques. Afin d'avoir une bonne précision dans la résolution des boucles de rétroaction et des non-linéarités, une formulation Jacobienne (JM) [40] est nécessaire pour garder la solution simultanée via une méthode de Newton itérative.

Une méthode de Jacobienne réduite (RJM) [41][42] a été introduite pour accélérer la formulation Jacobienne initiale en réduisant la taille du système linéaire à résoudre. Les autres variables sont mises à jour via des séquences de boucle. Ce chapitre va présenter les améliorations qui ont été apportées à RJM. Un premier point est de proposer une formulation générique de l'obtention des séquences de boucles en vue d'implémenter la méthode dans un outil de simulation EMT. Un deuxième point a été d'explorer le problème de minimisation du rang de la Jacobienne qui est équivalent à résoudre un problème du Feedback Arc Set minimal (FAS) : trouver l'ensemble des arcs de taille minimale, qui retirés d'un graphe, le rendent acyclique. Les états de l'art existants [43] permettent d'identifier les heuristiques les plus efficaces. La nouvelle méthode de Jacobienne réduite optimisée a été implémentée et testée dans l'environnement en temps différé de [2] principalement sur des réseaux de production d'énergies renouvelables [125].

6.2 Principe de résolution des équations de contrôle

Dans les outils de simulation EMT, la résolution des équations de contrôle est découplée de celle des équations de réseaux. Un délai d'un pas de temps est inséré entre les deux résolutions. La partie contrôle récupère les mesures (courants, tensions...) issues de la résolution du réseau pour faire les calculs des commandes à appliquer au pas de temps suivant. Dans la résolution des équations de contrôle, l'approche entrée/sortie schéma-bloc est utilisée [39]. Le système de contrôle est représenté par un graphe orienté $G = (V(G), E(G))$. $V(G)$ est l'ensemble des sommets qui correspond à l'ensemble des sorties des schémas-blocs. L'ensemble des arcs, $E(G)$, correspond à l'ensemble des schémas-blocs. Ces connections entre les sorties résultent de leur interdépendance dans le calcul de leur valeur. En effet, la valeur d'une sortie dépend des valeurs d'autres sorties qu'utilise en entrée pour son calcul la fonction du schéma-bloc associé. Pour chaque sortie i est associée la variable x_i . Chaque arc de sommet destination i est labélisé par la fonction f_i qui permet de calculer la sortie x_i . La Figure 6-2 ci-dessous montre la représentation d'une boucle algébrique en graphe de contrôle.

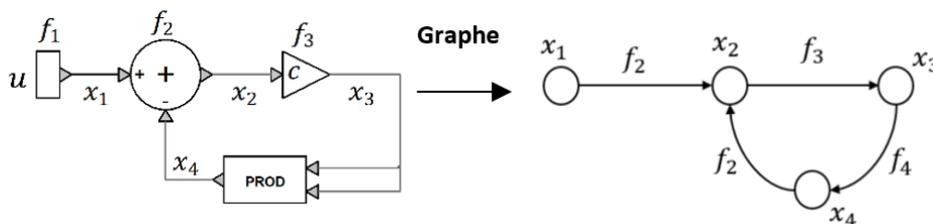


Figure 6-2 : Boucle algébrique simple modélisée par des schémas-blocs et son graphe associé.

Cette boucle correspond à la résolution d'une équation du second degré. Pour chaque bloc, on a les fonctions suivantes :

$$\forall x, y \in \mathbb{R}, f_1(x) = u, f_2(x, y) = x - y, f_3(x) = cx, f_4(x, y) = xy \quad (6-1)$$

A partir de ces fonctions, on calcule les sorties associées :

$$x_1 = u, x_2 = x_1 - x_4, x_3 = cx_2, x_4 = x_3^2 \quad (6-2)$$

En substituant x_2 dans l'ensemble des équations, on obtient l'équation du second degré à résoudre :

$$u - x_2 - c^2 x_2^2 = 0 \quad (6-3)$$

Pour tout sommet v de G , on introduit l'ensemble de ses successeurs $Succ(v) \subset V(G)$ et l'ensemble de ses prédécesseurs $Pred(v) \subset V(G)$ définis comme suit :

$$\begin{aligned} \forall v \in V(G), succ \in Succ(v) &\Leftrightarrow (v, succ) \in E(G) \\ pred \in Pred(v) &\Leftrightarrow (pred, v) \in E(G) \end{aligned} \quad (6-4)$$

On introduit $S(G)$, l'ensemble des sommets source de G , tel que :

$$\forall s \in S(G), Pred(s) = \{\emptyset\} \quad (6-5)$$

On définit également $P(G)$, l'ensemble des sommets puits de G , tel que :

$$\forall s \in P(G), Succ(s) = \{\emptyset\} \quad (6-6)$$

Pour résoudre un système de contrôle, on cherche, si possible, à ordonner les sommets de son graphe G tel que tout sommet se trouve avant son successeur. Soit la fonction $Ordre: V(G) \rightarrow \mathbb{N}$ qui définit cet ordonnancement recherché des sommets. Les sommets de G vérifient :

$$\forall v \in V(G), \forall succ \in V(G) \quad Ordre(v) < Ordre(succ) \quad (6-7)$$

Cela permet d'avoir une séquence d'exécution **Seq** des schémas-blocs pour pouvoir calculer dans le bon ordre l'ensemble des sorties : lorsqu'on exécute un schéma-bloc l'ensemble de ses entrées ont été calculées préalablement. Si G n'a pas de cycle, cela correspond à l'ordre topologique. Dans le cas cyclique, certains sommets ne vont pas respecter (6-7). Cela aura pour conséquence d'introduire un délai dans les calculs.

L'algorithme du Parcours en Profondeur (DFS) [126], décrit ci-dessous dans sa version récursive, est utilisé pour obtenir la séquence **Seq** des fonctions des schémas-blocs. Cette séquence donne leur ordre d'exécution à chaque pas de temps pour résoudre séquentiellement les équations de contrôle. On distingue deux types d'ordonnements : l'ordre pré-visite (*Visites*) et l'ordre post-visite (*OrdrePostVisite*). L'ordre pré-visite est l'ordre de parcours des sommets. Il ne garantit pas (6-7). Dans l'ordre post-visite, chaque sommet est ordonné après ses successeurs. En prenant l'ordre post-visite inversé, on garantit (6-7) pour les graphes acycliques. A noter que dans le DFS, on parcourt d'abord les sommets source non-visités du graphe dans le respect de (6-7).

Algorithm 1 DFS (Parcours en profondeur)

```

1: Graphe  $G$ 
2:  $Visites \leftarrow \{\emptyset\}$ 
3:  $OrdrePostVisite \leftarrow \{\emptyset\}$ 
4: for  $v \in S(G)$  do
5:   if  $v \notin Visites$  then
6:      $Explorer(v)$ 
7:   end if
8: end for
9: for  $v \in V(G)$  do
10:  if  $v \notin Visites$  then
11:     $Explorer(v)$ 
12:  end if
13: end for
14: function  $Explorer(v)$ 
15:    $Visites \leftarrow Inserir(v)$ 
16:   for  $succ \in Succ(v)$  do
17:    if  $succ \notin Visites$  then
18:       $Explorer(succ)$ 
19:    end if
20:   end for
21:    $OrdrePostVisite \leftarrow Inserir(v)$ 
22: end function

```

Au niveau de l'implémentation, deux représentations du graphe peuvent être choisies pour construire $Succ(v)$ pour tout sommet v . La plus naturelle est la représentation en listes d'adjacence. Ainsi, pour chaque sommet v , on définit une liste de successeurs $ListeSucc(v)$ qui représente

Succ(v). Une deuxième représentation est la matrice d'adjacence A_a de G avec $|V(G)| = n$. Soit une numérotation arbitraire des sommets de G telle que $V(G) = \{v_i, i \in \llbracket 1, n \rrbracket\}$. Les éléments a_{ij} de A_a sont définis par :

$$a_{ij} = \begin{cases} 1 & \text{si } (v_i, v_j) \in E(G) \\ 0 & \text{sinon} \end{cases} \quad (6-8)$$

La matrice d'adjacence pour un graphe de contrôle est souvent creuse et peut être de grande taille. Ainsi, il est plus efficace de choisir la représentation en listes d'adjacence qui est plus performante qu'une représentation matricielle. Néanmoins, la matrice d'adjacence est utilisée pour la formulation de la Jacobiennne réduite dans 6.4 car les matrices considérées sont de taille plus réduite.

Reprenons l'exemple de la boucle algébrique simple de la Figure 6-2. En utilisant l'algorithme du DFS, on obtient les ordres suivants,

- Ordre pré-visite : $\{x_1, x_2, x_3, x_4\}$
- Ordre post-visite : $\{x_4, x_3, x_2, x_1\}$
- Ordre post-visite inversé : $\{x_1, x_2, x_3, x_4\}$

Ici l'ordre pré-visite correspond à l'ordre post-visite inversé. Ce n'est pas toujours le cas. A partir de l'ordre post-visite inversé, on remplace chaque sortie par la fonction du schéma-bloc associée pour obtenir la séquence d'exécution **Seq** pour le calcul des sorties :

$$\mathbf{Seq} = \{f_1, f_2, f_3, f_4\} \quad (6-9)$$

Pour ce graphe cyclique, l'ordre pos-visite inversé est équivalent à l'ordre topologique du graphe acyclique $G_{acy} = (V(G), E(G) \setminus (x_2, x_4))$ qui le graphe G auquel on a cassé le cycle en retirant l'arc (x_2, x_4) . Cela revient à ajouter un délai artificiel pour le calcul de x_4 à partir de f_2 . Ainsi, la résolution séquentielle (SEQ), à partir de l'ordre post-visite inversé du DFS, est la méthode la plus utilisée dans les outils de simulations. On la retrouve pour le temps réel de [1] et dans Matlab/Simulink [6] où elle est privilégiée dans la génération de code. Néanmoins, la solution n'est plus simultanée par l'ajout de délais artificiels ce qui peut mener à une dégradation de la précision et rendre instable la simulation.

6.3 Formulation Jacobiennne pour résoudre les boucles de rétroaction

Afin d'avoir une solution simultanée [40], on écrit les équations de contrôle de chaque schéma-bloc composant le système de contrôle. Soit n le nombre de schémas-blocs. Pour $i \in \llbracket 1, n \rrbracket$, il faut résoudre les équations différentielles algébriques (DAE) suivantes :

$$\tilde{f}_i(\dot{\mathbf{x}}, \mathbf{x}, t) = x_i \quad (6-10)$$

Où \mathbf{x} contient les variables de sorties x_i des schémas-blocs, t la dépendance temporelle et \tilde{f}_i est la fonction de bloc associée pour calculer x_i . Comme pour la partie réseau, on discrétise le temps de simulation suivant un pas de temps Δt . On intègre les équations différentielles en utilisant la méthode du trapèze pour approximer le calcul de l'intégrale. Ainsi, à chaque pas de temps, on se ramène à la résolution des équations algébriques suivantes :

$$f_i(\mathbf{x}) = x_i \quad (6-11)$$

Où f_i est la fonction discrétisée du bloc i . Par la suite, f_i est dénommée comme la fonction du schéma-bloc i comme dans le paragraphe 6.2. Cela revient à résoudre l'équation,

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) - x_1 \\ \vdots \\ f_n(\mathbf{x}) - x_n \end{pmatrix} = 0 \quad (6-12)$$

Comme \mathbf{F} peut être non-linéaire, un schéma itératif de Newton est utilisé pour résoudre cette équation. A l'itération $k + 1$, on a :

$$\mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{J}^{(k)}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \mathbf{0} \quad (6-13)$$

Où \mathbf{J} est la matrice Jacobienne de \mathbf{F} . On réécrit cette équation sous une forme linéarisée :

$$\widehat{\mathbf{A}}^{(k)} \mathbf{x}^{(k+1)} = \mathbf{B}^{(k)} \quad (6-14)$$

Où,

$$\begin{aligned} \widehat{\mathbf{A}}^{(k)} &= \mathbf{J}^{(k)} \\ \mathbf{B}^{(k)} &= \mathbf{J}^{(k)} \mathbf{x}^{(k)} - \mathbf{F}(\mathbf{x}^{(k)}) \end{aligned} \quad (6-15)$$

$\widehat{\mathbf{A}}^{(k)}$ va contenir les tangentes de $\mathbf{F}(\mathbf{x})$ et $\mathbf{B}^{(k)}$ les ordonnées à l'origine.

Le critère de convergence suivant est utilisé pour la méthode de Newton :

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\|} \leq tol \quad (6-16)$$

Où la valeur de tol est définie par l'utilisateur.

Prenons l'exemple simple d'un schéma-bloc de contrôle avec pour équation $y = x^2$. Dans le cadre de la méthode de Newton, on recherche une approximation linéaire ($m^{(k)}$ et $b^{(k)}$) au voisinage $x^{(k)}$ de y tel que :

$$y = m^{(k)}x + b^{(k)} \quad (6-17)$$

Par définition, $m^{(k)}$ est égale à la tangente en $x^{(k)}$ de y et on obtient $b^{(k)}$ en prenant $x = x^{(k)}$ après le calcul de la pente. D'où, on en déduit que :

$$m^{(k)} = \left. \frac{\partial y}{\partial x} \right|_{x=x^{(k)}} = 2x^{(k)} \quad (6-18)$$

Et,

$$\begin{aligned} b^{(k)} &= -m^{(k)}x^{(k)} + (x^{(k)})^2 \\ b^{(k)} &= -(x^{(k)})^2 \end{aligned} \quad (6-19)$$

On utilise cette approximation linéaire pour l'étape $k + 1$,

$$\begin{aligned} y^{(k+1)} &= m^{(k)}x^{(k+1)} + b^{(k)} \\ [-m^{(k)} \quad 1] \begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} &= b^{(k)} \end{aligned} \quad (6-20)$$

Ainsi, $[-m^{(k)} \quad 1]$ et $b^{(k)}$ vont remplir respectivement les lignes correspondantes à y dans $\widehat{\mathbf{A}}^{(k)}$ et $\mathbf{B}^{(k)}$.

Reprenons la boucle algébrique de la Figure 6-2. Calculons la matrice Jacobienne $\mathbf{J}(\mathbf{x})$ associée ainsi que les ordonnées à l'origine $\mathbf{B}(\mathbf{x})$.

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} u - x_1 \\ x_1 - x_4 - x_2 \\ cx_2 - x_3 \\ x_3^2 - x_4 \end{pmatrix} = \mathbf{0} \quad (6-21)$$

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial F_1(\mathbf{x})}{\partial x_1} & \frac{\partial F_1(\mathbf{x})}{\partial x_2} & \frac{\partial F_1(\mathbf{x})}{\partial x_3} & \frac{\partial F_1(\mathbf{x})}{\partial x_4} \\ \frac{\partial F_2(\mathbf{x})}{\partial x_1} & \frac{\partial F_2(\mathbf{x})}{\partial x_2} & \frac{\partial F_2(\mathbf{x})}{\partial x_3} & \frac{\partial F_2(\mathbf{x})}{\partial x_4} \\ \frac{\partial F_3(\mathbf{x})}{\partial x_1} & \frac{\partial F_3(\mathbf{x})}{\partial x_2} & \frac{\partial F_3(\mathbf{x})}{\partial x_3} & \frac{\partial F_3(\mathbf{x})}{\partial x_4} \\ \frac{\partial F_4(\mathbf{x})}{\partial x_1} & \frac{\partial F_4(\mathbf{x})}{\partial x_2} & \frac{\partial F_4(\mathbf{x})}{\partial x_3} & \frac{\partial F_4(\mathbf{x})}{\partial x_4} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 \\ 0 & c & -1 & 0 \\ 0 & 0 & 2x_3 & -1 \end{bmatrix} \quad (6-22)$$

On en déduit les ordonnées à l'origine $\mathbf{B}(\mathbf{x})$:

$$\mathbf{B}(\mathbf{x}) = \mathbf{J}(\mathbf{x})\mathbf{x} - \mathbf{F}(\mathbf{x}) \quad (6-23)$$

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} -x_1 \\ x_1 - x_4 - x_2 \\ cx_2 - x_3 \\ 2x_3^2 - x_4 \end{pmatrix} - \begin{pmatrix} u - x_1 \\ x_1 - x_4 - x_2 \\ cx_2 - x_3 \\ x_3^2 - x_4 \end{pmatrix} = \begin{pmatrix} -u \\ 0 \\ 0 \\ x_3^2 \end{pmatrix} \quad (6-24)$$

Dans l'annexe A.1, la formulation Jacobienne est détaillée pour le cas du PLL (Phase-locked loop, boucle à verrouillage de phase) [40][127] qui permet à un contrôleur de se synchroniser sur la fréquence du réseau.

Dans les exemples précédents, l'expression analytique des dérivées des fonctions blocs sont bien connues pour le calcul de la Jacobienne. Ce n'est pas le cas pour les boîtes noires de contrôle (code de contrôle sous forme de DLL) ou de blocs définis par l'utilisateur via un éditeur d'équations mathématiques générique dans [2]. La méthode des perturbations est utilisée pour calculer les dérivées. Pour une fonction f ayant pour variable scalaire x , dont on ne connaît pas une expression analytique de $\frac{\partial f}{\partial x}$, cela donne pour l'itération k :

$$\left. \frac{\partial f}{\partial x} \right|_{x=x^{(k)}} \approx \frac{1}{\delta} [f(x^{(k)} + \delta) - f(x^{(k)})] \quad (6-25)$$

Avec $\delta = \max(10^{-12}, 10^{-8}|x^{(k)}|)$.

La résolution du système linéaire se fait comme pour les équations de réseaux du paragraphe 3.2 avec un solveur de décomposition LU adapté aux matrices creuses. Cette résolution peut être assez lente. En effet, à chaque fois, il faut recalculer la matrice Jacobienne ce qui nécessite de la refactoriser et de relancer la résolution. Généralement, on ne forme pas la Jacobienne pour l'ensemble du graphe du contrôleur. On détermine les composantes fortement connexes (SCC) du graphe [124]. Les SCCs, qui ont plus d'un sommet, contiennent au moins un cycle. Une formulation Jacobienne est nécessaire pour leur sous-graphe associé. Les SCCs, qui n'ont pas de cycles, ne comportent qu'un sommet. Une résolution séquentielle suivant la séquence des SCCs suffit pour calculer les sorties. L'algorithme de Kosaraju, détaillé dans l'annexe B, est utilisé pour déterminer les SCCs d'un graphe \mathbf{G} . Il consiste en deux DFS. Le premier permet de déterminer l'ordre post-visite inversé des sommets. A partir de cet ordre, le second DFS va parcourir le graphe transposé \mathbf{G}^T et construire les SCCs correspondantes. Par définition,

$$\begin{aligned} \mathbf{V}(\mathbf{G}^T) &= \mathbf{V}(\mathbf{G}) \\ (i, j) \in \mathbf{E}(\mathbf{G}^T) &\Leftrightarrow (j, i) \in \mathbf{E}(\mathbf{G}) \end{aligned} \quad (6-26)$$

Ainsi, $\mathbf{Succ}(\mathbf{G})$ est utilisé pour le DFS de \mathbf{G} et $\mathbf{Pred}(\mathbf{G})$ pour le DFS de \mathbf{G}^T .

Si l'on reprend le cas simple de la boucle algébrique, on obtient deux composantes fortement connexes comme l'illustre la Figure 6-3.

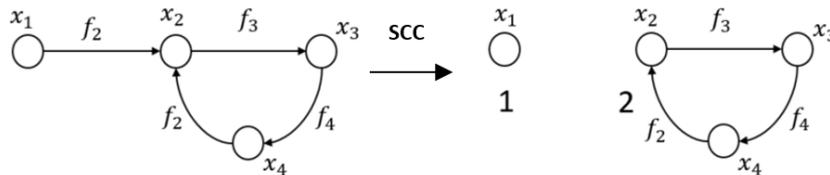


Figure 6-3: Décomposition en composantes fortement connexes (SCC) du graphe de la boucle algébrique.

Cela a pour conséquence de réduire la taille de la Jacobienne à considérer d'une dimension.

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} -1 & 0 & -1 \\ c & -1 & 0 \\ 0 & 2x_3 & -1 \end{bmatrix}, \mathbf{B}(\mathbf{x}) = \begin{pmatrix} -u \\ 0 \\ x_3^2 \end{pmatrix} \quad (6-27)$$

6.4 Formulation optimisée de la Jacobienne réduite

6.4.1 Principe

Le principe de la Jacobienne réduite (RJM), introduite dans [41][42], est de diminuer la taille du système linéaire à résoudre à chaque itération de Newton, afin d'accélérer la résolution. Une première intuition pour améliorer les performances serait de réduire le nombre de variables prises en compte dans la méthode JM. Par exemple, deux blocs de contrôle en série peuvent être regroupés en un seul bloc, ce qui enlève une variable. Ainsi, RJM applique ce principe de contraction des blocs afin de réduire le nombre de variables. Cela s'inspire de la représentation classique du « general feedback interconnection » pour étudier la stabilité des boucles de rétroaction [39] comme illustré par la Figure 6-4 ci-dessous.

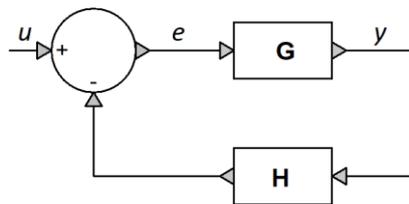


Figure 6-4 : Schémas-blocs contractés pour l'étude de la stabilité des boucles de rétroaction.

Les schémas-blocs de la Figure 6-4 représentent les équations génériques d'une boucle de rétroaction :

$$\begin{aligned} e &= u - H(y) \\ y &= G(e) \end{aligned} \quad (6-28)$$

Où les vecteurs u , y et e contiennent respectivement les entrées, les sorties et les erreurs de la boucle de rétroaction. G et H sont les fonctions issues de la contraction des blocs de contrôle.

La Figure 6-5 montre le formalisme général utilisé par RJM pour la résolution de chaque SCC cyclique. L'ensemble des schémas-blocs ont été contractés en un super bloc et un bloc de feedback (de rétroaction) auxquels on a associé respectivement les fonctions G (fonction de la séquence de boucle) et ϕ (fonction du bloc de feedback). Les variables de coupes ou de feedback contenues dans e , une fois retirées, rendent le graphe acyclique. L'entrée u correspond aux variables qui ont été calculées dans les résolutions des précédents SCCs, dans l'ordre topologique du graphe des SCCs. La sortie \hat{e} sera utilisée pour le calcul des SCCs suivantes.

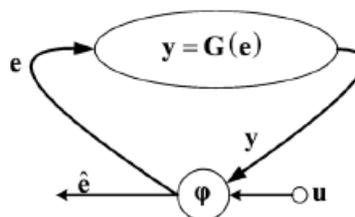


Figure 6-5 : Schéma-bloc générique de la Jacobienne réduite.

On obtient alors les équations du cycle pour la variable e :

$$y = G(e) \quad (6-29)$$

$$e = \phi(u, G(e)) \quad (6-30)$$

$$\phi(e) = e - \phi(u, G(e)) = 0$$

Il ne reste plus qu'à appliquer la méthode de Newton à ϕ pour calculer les variables de feedback.

$$\begin{aligned} \phi(e^{(k)}) + J^{(k)} \Delta e^{(k)} &= 0 \\ e^{(k+1)} &= e^{(k)} + \Delta e^{(k)} \end{aligned} \quad (6-31)$$

Où J est la matrice jacobienne de ϕ . Comme aucune forme analytique n'est disponible, elle est calculée par la méthode des perturbations directement sur ϕ sans le faire pour chaque bloc. Cela évite de cumuler les erreurs d'approximation de chaque bloc. En appliquant l'équation (6-25) à ϕ , on obtient :

$$\forall i, j \in \llbracket 1, N_f \rrbracket, J_{ij}^{(k)} = \frac{1}{\delta_j} \left[\phi_i(e_j^{(k)} + \delta_j) - \phi_i(e_j^{(k)}) \right] \quad (6-32)$$

$$\delta_j = \max(10^{-12}, 10^{-8} |e_j^{(k)}|)$$

Avec N_f le nombre de variables de coupe ou de feedback. Une fois que e a été calculé, il suffit d'utiliser l'ordre de composition des fonctions qui constitue G pour mettre à jour les autres variables du SCC. Pour cette formulation, il reste à déterminer de manière automatique G et ϕ , mais également les variables de feedback e , dont leur nombre doit être minimisé pour réduire le rang de la Jacobienne.

On reprend le même critère de convergence de la méthode JM. D'après l'équation (6-16), on vérifie à chaque itération :

$$\frac{\|e^{(k+1)} - e^{(k)}\|}{\|e^{(k)}\|} \leq tol \quad (6-33)$$

Remarque

Le critère de convergence de la méthode de Newton n'est vérifié que sur les variables de feedback (équation (6-33)) et plus sur l'ensemble des variables (équation (6-16)). Cela a des conséquences sur la propagation des erreurs d'approximation de la méthode de Newton à travers la séquence des blocs et sur les propriétés de convergence. La convergence de ϕ ne sera pas forcément identique à celle de F qui comporte les fonctions de l'ensemble des schémas-blocs comme déjà constaté dans [41].

6.4.2 Calcul des séquences de boucle

Considérons sur la Figure 6-6 la variable de coupe/feedback e_i .

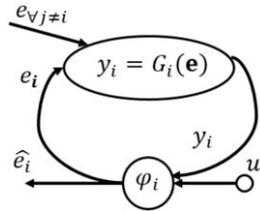


Figure 6-6 : Schéma-bloc contracté de la Jacobienne réduite associé à la variable de feedback e_i .

La fonction de séquence de boucle associée à e_i est notée G_i . Elle permet de déterminer la composante ϕ_i de ϕ .

$$\phi_i(e) = e_i - \phi_i(u, G_i(e)) \quad (6-34)$$

Après avoir fixé les variables de coupes/feedback, le graphe acyclique G_{acy} du graphe de contrôle de la SCC est obtenu en retirant l'ensemble des arcs, qui ont pour origine un sommet associé à une variable de feedback. L'absence de cycle est garantie dans la définition des variables de coupes/feedback e . Les sommets des variables de feedback sont labélisés comme des « sommets puits » et les sommets destination des arcs retirés comme des « sommets sources » (ensemble $Src(G_{acy})$). La Figure 6-7 ci-dessous montre l'identification de la variable de feedback en gras ($e = x_4$) de la SCC cyclique de la boucle algébrique (Figure 6-2), ainsi que la labélisation en source/puits des sommets dans le graphe acyclique obtenu. On déduit également la fonction du bloc du feedback pour calculer e : $\phi = f_4$.

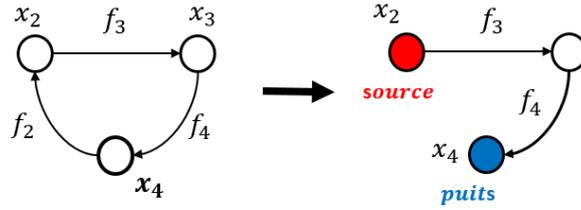


Figure 6-7 : Obtention du graphe acyclique avec la labélisation source/puit après identification de la variable de feedback (en gras).

Selon [41], la séquence de boucle \mathcal{S}_{e_i} associée à G_i (variable de feedback/sommet puit e_i) est composée de l'ensemble des sommets qui appartiennent à un chemin allant d'un sommet labélisé en source $s \in \mathbf{Src}(G_{acy})$ vers e_i . En étant ordonnée suivant l'ordre topologique de G_{acy} , \mathcal{S}_{e_i} permet de déduire l'ordre \mathcal{O}_{e_i} de composition des fonctions des schémas-blocs qui constitue G_i à l'exclusion de φ_i . Il suffit de remplacer chaque variable de sortie par la fonction du schéma-bloc correspondante pour son calcul.

$$\mathcal{S}_{e_i} = \{x_{k_1}, \dots, x_{k_{card(L_i)-1}}, e_i\} \quad (6-35)$$

$$\mathcal{O}_{e_i} = \{f_{k_1}, \dots, f_{k_{card(L_i)-1}}, \varphi_i\} \quad (6-36)$$

$$G_i = f_{k_{card(L_i)-1}} \circ \dots \circ f_{k_1} \quad (6-37)$$

On définit $\mathbf{P}(G)$ l'ensemble des chemins de G . Un chemin est défini comme une suite d'arcs consécutifs. $\pi[i, j]$ est un chemin allant du sommet i au sommet j . On note pour $s, v \in \mathbf{V}(G)$, \mathcal{S}_{sv} l'ensemble des sommets appartenant à un chemin de s à v .

$$\mathcal{S}_{sv} = \{c \in \mathbf{V}(G) \setminus \pi(s, c), \pi(c, v) \in \mathbf{P}(G)\} \quad (6-38)$$

$$\mathcal{S}_{e_i} = \bigcup_{s \in \mathbf{Src}(G_{acy})} \mathcal{S}_{se_i} \quad (6-39)$$

Pour déterminer \mathcal{S}_{e_i} , on privilégie une représentation matricielle de G_{acy} . Généralement, G_{acy} n'est pas de grande taille compte tenu de la décomposition en SCCs. Les calculs matriciels devraient être assez performants. On définit pour tout ensemble $V \subset \mathbf{V}(G)$, la représentation en vecteur colonne \tilde{V} tel que :

$$\tilde{V}_{k_v} = 1 \text{ si } v \in V, 0 \text{ sinon} \quad (6-40)$$

Avec $k_v \in \llbracket 1, Card(\mathbf{V}(G)) \rrbracket$ la numérotation/indice de v issu d'un ordonnancement de $\mathbf{V}(G)$. Généralement, on prend l'ordre post-visite inversé d'un DFS [126] ou l'ordre topologique pour un graphe acyclique tel que G_{acy} .

On introduit les opérateurs, élément par élément, \oplus et \otimes sur des matrices binaires \mathbf{A} et \mathbf{B} de même taille:

$$(\mathbf{A} \oplus \mathbf{B})_{ij} = a_{ij} \oplus b_{ij} \quad (6-41)$$

$$(\mathbf{A} \otimes \mathbf{B})_{ij} = a_{ij} \otimes b_{ij} \quad (6-42)$$

Où \oplus et \otimes correspondent respectivement aux opérateurs logiques ET et OU. Les matrices binaires \mathbf{A} et \mathbf{B} peuvent être assimilées à des matrices d'adjacence de graphes.

Les extracteurs de colonnes et de lignes sont définis comme suit :

$$\text{Ligne } \mathbf{A}(i) = [a_{i1} \quad \dots \quad a_{ij} \quad \dots] \quad (6-43)$$

$$\text{Colonne } \mathbf{A}(j) = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{ij} \\ \vdots \end{bmatrix} \quad (6-44)$$

Pour un graphe G orienté, la détermination de l'ensemble \mathcal{S}_{sv} pour $s, v \in \mathbf{V}(G)$ s'appuie sur la fermeture transitive de G notée $\mathbf{C}(G)$. Elle permet de connaître l'accessibilité entre deux sommets via l'existence d'un chemin. $\mathbf{C}(G)$ est défini comme suit,

$$\begin{aligned} V(\mathcal{C}(G)) &= V(G) \\ (i, j) \in E(\mathcal{C}(G)) &\Leftrightarrow \exists \pi[i, j] \in P(G) \end{aligned} \quad (6-45)$$

La fermeture transitive peut être obtenue via un DFS. L'algorithme ci-dessous permet de construire la matrice d'accessibilité \mathbf{R} qui est la matrice d'adjacence de $\mathcal{C}(G)$.

Algorithm 2 Fermeture Transitive via un DFS

```

1: Graphe  $G$ 
2:  $R \leftarrow 0$ 
3: for  $i \in V(G)$  do
4:    $DFS\_tc(i, i)$ 
5: end for
6: function  $DFS\_tc(i, j)$ 
7:    $R_{ij} \leftarrow 1$ 
8:   for  $succ \in Succ(j)$  do
9:     if  $R_{isucc} = 0$  then
10:       $DFS\_tc(j, succ)$ 
11:     end if
12:   end for
13: end function

```

D'après la définition de $\mathcal{C}(G)$,

$$\begin{aligned} \pi(s, c) \in P(G) &\Leftrightarrow (s, c) \in E(\mathcal{C}(G)) \\ \pi(c, v) \in P(G) &\Leftrightarrow (c, v) \in E(\mathcal{C}(G)) \end{aligned} \quad (6-46)$$

On en déduit pour S_{sv} que :

$$c \in S_{sv} \Leftrightarrow (s, c) \in E(\mathcal{C}(G)) \text{ et } (c, v) \in E(\mathcal{C}(G)) \quad (6-47)$$

Dans une représentation matricielle, en utilisant \mathbf{R} et les opérateurs précédemment, on obtient :

$$(\tilde{S}_{sv})_i = r_{k_s i} \otimes r_{i k_v} \quad (6-48)$$

Avec les indices k_s et k_v issus de l'ordre topologique de G_{acy} .

On en déduit que :

$$\tilde{S}_{sv} = \text{Ligne } \mathbf{R}(k_s) \otimes \text{Colonne } \mathbf{R}(k_v)^T \quad (6-49)$$

En combinant sur l'ensemble des éléments de $\text{Src}(G_{acy})$, on obtient le calcul matriciel de \tilde{S}_{e_i} .

$$\tilde{S}_{e_i} = \bigoplus_{s \in \text{Src}(G_{acy})} (\text{Ligne } \mathbf{R}(k_s) \otimes \text{Colonne } \mathbf{R}(k_{e_i})^T) \quad (6-50)$$

Pour l'exemple de la boucle algébrique, on a :

$$\text{Ordre Topologique} = (x_2 \quad x_3 \quad x_4) \quad (6-51)$$

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (6-52)$$

$$\begin{aligned} \tilde{S} &= \text{Ligne } \mathbf{R}(1) \otimes \text{Colonne } \mathbf{R}(3)^T \\ \tilde{S} &= (1 \quad 1 \quad 1) \otimes \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^T \end{aligned} \quad (6-53)$$

$$\tilde{S} = (1 \quad 1 \quad 1)$$

On en déduit que :

$$\mathbf{S} = \{x_2, x_3, x_4\} \quad (6-54)$$

$$\mathbf{O} = \{f_2, f_3, f_4\} \quad (6-55)$$

$$G = f_3 \circ f_2 \quad (6-56)$$

Le calcul des séquences de boucle est détaillé dans l'annexe A.2 pour le cas de la PLL.

Cette formulation générique pour obtenir les séquences de boucle a simplifié l'approche originale de [41]. Il n'est pas nécessaire d'ajouter des sommets virtuels. Après avoir éliminé les cycles, une labélisation en source et en puits des sommets est suffisante pour calculer les séquences de boucle. Il a fallu également adapter cette formulation aux graphes de contrôle dont les sommets sont les

sorties de chaque bloc afin de faciliter l'intégration dans [2]. Dans [41], les sommets du graphe de contrôle sont associés aux fonctions des schémas-blocs.

6.4.3 Détection des coupes

L'objectif de la détection des coupes est de réduire au maximum la dimension de la matrice Jacobienne, c'est-à-dire trouver le nombre minimal de coupes qui rend acyclique le graphe associé au système de contrôle. Cela revient à résoudre le problème du Feedback Arc Set minimal (FAS) qui cherche un ensemble minimal d'arcs qui, une fois retirés, rendent le graphe acyclique. Ce type de problème est NP-difficile [43]. Généralement, comme le calcul de la solution exacte est trop long, des heuristiques sont utilisées pour trouver rapidement une solution approchée [43][129].

Dans [41][42], une méthode basée sur les valeurs propres de la matrice d'adjacence est utilisée pour vérifier que le graphe est acyclique. En effet, si toutes les valeurs propres sont nulles cela signifie que le graphe est sans cycle. Cependant, il n'est pas nécessaire de calculer les valeurs propres (assez coûteux en temps de calcul) du graphe pour détecter et supprimer l'ensemble des cycles. Les heuristiques définies dans [43][129] seront plus performantes et permettent de trouver des solutions de meilleure qualité. Aucune d'entre elles n'est basée sur un calcul de valeurs propres. Cela induit que l'on peut couper n'importe où les cycles du graphe de contrôle. Mathématiquement rien n'interdit de couper n'importe où mais une coupe différente peut impacter la précision du résultat. En effet, les propriétés de convergence de la méthode de Newton ne sont pas forcément les mêmes suivant l'ordre de composition des fonctions des schémas-blocs issu des séquences de boucle.

L'heuristique la plus simple pour résoudre le problème FAS est le DFS. Suite à un DFS, les sommets du graphe G d'un SCC cyclique sont ordonnés suivant l'ordre post-visite inversé. On note k_s le rang dans cet ordre pour $s \in V(G)$. Un arc $(s, v) \in E(G)$ est un back arc si et seulement si $k_s > k_v$. En retirant l'ensemble des back arcs de G , on obtient un graphe acyclique G_{acy} . Pour un back arc (s, v) , s est labélisé en sommet puit et la variable associée est une variable de coupe/feedback ; v est labélisé comme un sommet source ($v \in Src(G_{acy})$).

La Figure 6-8 montre l'identification du back arc suivant l'ordre du DFS qui rend acyclique le graphe de la boucle algébrique. Cela permet d'identifier la variable de feedback ($e = x_4$) et la fonction du schéma-bloc de feedback associé ($\varphi = f_4$). La labélisation en source/puits est utilisée pour le calcul de la séquence de boucle du paragraphe 6.4.2.

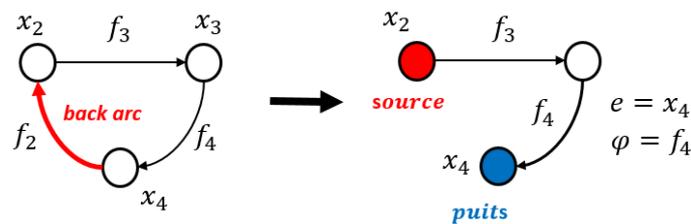


Figure 6-8 : Graphe acyclique à partir d'un DFS pour la boucle algébrique.

Cependant, le DFS ne propose aucune stratégie pour minimiser le nombre de coupes afin de rendre le graphe acyclique. La Figure 6-9 ci-dessous présente un graphe acyclique où le DFS est sous-optimal pour résoudre le problème FAS.

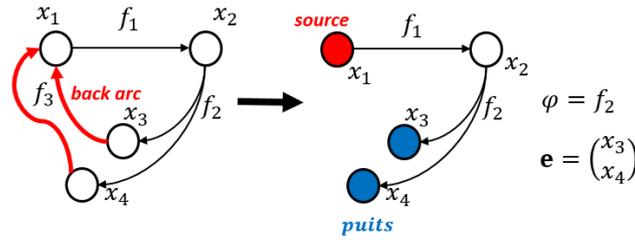


Figure 6-9 : Solution sous-optimale du DFS au problème du FAS.

Deux back arcs ont été retirés avec le choix de deux variables de feedback alors que la solution optimale (Figure 6-10) est d'enlever l'arc (x_1, x_2) pour rendre le graphe acyclique en utilisant qu'une seule variable de feedback ($e = x_1$).

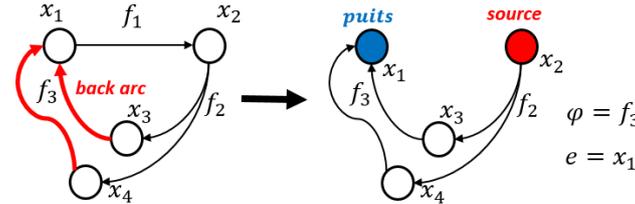


Figure 6-10 : Solution optimale du FAS obtenue par DFSOpt.

Généralement, dans les systèmes de contrôle, le schéma-bloc associé à la fonction d'un back arc, appelé aussi le bloc de feedback, aura plus d'entrées que de sorties. Comme cela avait été déjà proposé par [41][42], au lieu de couper les back arcs du DFS, il serait plus judicieux d'enlever l'ensemble des arcs qui partent d'un sommet destination d'un back arc. L'algorithme de cette heuristique, DFSOpt, est décrit ci-dessous. Après la suppression des arcs (*PostVisite* \rightarrow ordre post-visite inversé), un deuxième DFS est nécessaire pour ordonner, selon l'ordre topologique, les sommets du graphe acyclique. La solution optimale de la Figure 6-10 est obtenue en appliquant DFSOpt.

Algorithm 3 DFSOpt

```

1: Graphe  $G$ 
2:  $PostVisite \leftarrow \{\emptyset\}$ 
3:  $DFS(G, PostVisite)$ 
4: for  $(s, v) \in E(G)$  do
5:   if  $Pos(PostVisite, s) > Pos(PostVisite, v)$  then
6:     for  $succ \in Succ(v)$  do
7:        $E(G) \leftarrow E(G) \setminus (v, succ)$ 
8:     end for
9:   end if
10: end for
11:  $DFS(G, PostVisite)$ 

```

D'autres travaux ont proposé des heuristiques efficaces pour réduire le nombre de coupes. Deux d'entre elles se détachent en faisant un bon compromis entre performance et qualité de la solution obtenue. La première GreedyFAS, initialement introduite par [129], est performante pour des graphes de grande taille. Sa stratégie consiste à minimiser le nombre d'arcs orientés de la gauche vers la droite. Successivement les sommets puits sont mis à droite et les sommets source à gauche pour éviter la création de back arc. La mesure δ est définie comme suit pour $u \in V(G)$,

$$\delta(u) = d^+(u) - d^-(u) \quad (6-57)$$

Où $d^+(u)$ est le degré des arcs entrant dans u et $d^-(u)$ pour les arcs sortants.

$$d^+(u) = Card(\{v \in V(G) \setminus (v, u) \in E(G)\}) \quad (6-58)$$

$$d^-(u) = Card(\{v \in V(G) \setminus (u, v) \in E(G)\}) \quad (6-59)$$

Afin de minimiser la création potentielle de back arc, les sommets u à plus grande valeur de $\delta(u)$ sont placés à gauche. Il ne reste plus qu'à retirer les back arcs selon l'ordre séquentiel **Seq** des sommets obtenus.

Algorithm 4 GreedyFAS

```

1: Graphe  $G$ 
2:  $Seq \leftarrow \{\emptyset\}, s_1 \leftarrow \{\emptyset\}, s_2 \leftarrow \{\emptyset\}$ 
3: while  $G \neq \{\emptyset\}$  do
4:   while  $P(G) \neq \{\emptyset\}$  do
5:     Choisir  $u \in P(G)$ 
6:      $s_2 \leftarrow us_2$ 
7:      $G \leftarrow G \setminus u$ 
8:   end while
9:   while  $S(G) \neq \{\emptyset\}$  do
10:    Choisir  $u \in S(G)$ 
11:     $s_1 \leftarrow s_1u$ 
12:     $G \leftarrow G \setminus u$ 
13:   end while
14:   Choisir  $u \in V(G)/\delta(u)$  minimale
15:    $s_1 \leftarrow s_1u$ 
16:    $G \leftarrow G \setminus u$ 
17: end while
18:  $Seq \leftarrow s_1s_2$ 

```

Les heuristiques de Tri telle que le Tri par Insertion (SortFAS) [130], donnent de bonnes solutions approchées. Le principe est d'insérer progressivement les sommets du graphe dans la position de la liste triée (**Seq**) qui génère le moins de back arcs (compteur *val*). Cet algorithme est monotone. Chaque itération sur la séquence obtenue **Seq** ne peut qu'améliorer la solution obtenue. On note SortFAS* la version itérative de SortFAS. On initialise **Seq** avec l'ordre post-visite inversé du DFS. Comme il a été constaté dans [43], les méthodes de Tri comme SortFAS sont moins performantes pour des graphes de grande taille. Mais comme la décomposition en SCC a réduit la taille des graphes à considérer par rapport à celle du graphe d'origine, il semble intéressant de retenir cette méthode.

Algorithm 5 SortFAS (Initialisation OrdrePostVisite du DFS)

```

1: Graphe  $G$ 
2:  $Seq \leftarrow \{\emptyset\}$ 
3:  $DFS(G, Seq)$ 
4: for  $v \in Seq$  do
5:    $val \leftarrow 0, min \leftarrow 0, loc \leftarrow Pos(Seq, v)$ 
6:   for  $j \in [loc - 1, 0]$  do
7:      $u \leftarrow Seq(j)$ 
8:     if  $(v, u) \in E(G)$  then
9:        $val \leftarrow val - 1$ 
10:    else if  $(u, v) \in E(G)$  then
11:       $val \leftarrow val + 1$ 
12:    end if
13:    if  $val \leq min$  then
14:       $min \leftarrow val, loc \leftarrow j$ 
15:    end if
16:  end for
17:   $Seq \leftarrow Insérer(v, loc)$ 
18: end for

```

Au total cinq heuristiques sont considérées :

- DFS [126], ordre post-visite inversé,
- DFSOpt,
- GreedyFAS [129] en utilisant l'implémentation *ArrayFAS* de [43],

- SortFAS [130] initialisée à partir du DFS, implémentée dans la version *InsertionSortFAS* de [43],
- SortFAS*, version itérative de SortFAS.

Tout en apportant une solution au problème FAS, ces heuristiques peuvent être utilisées pour une résolution séquentielle des contrôles. En effet, il suffit d'exécuter les fonctions des schémas-blocs pour calculer dans l'ordre de *Seq* les sorties. La minimisation du nombre de back arcs revient à minimiser le nombre de délais introduits dans le graphe de contrôle pour éliminer les cycles. Intuitivement, l'insertion moindre de délais devrait moins dégrader la précision. On retrouve cette idée dans [131] où l'heuristique pour minimiser l'insertion de délais est assez proche du principe de GreedyFAS. Ainsi, on peut considérer cinq méthodes (SEQ+nom de l'heuristique du FAS) à partir des heuristiques du FAS, pour obtenir une séquence d'exécution des fonctions des schémas-blocs : SEQ+DFS, SEQ+DFSOpt, SEQ+GreedyFAS, SEQ+SortFAS et SEQ+SortFAS*.

6.5 Validation et Performances

Le Tableau 6-1 présente les différentes méthodes qui sont testées pour la résolution des équations de contrôle. SEQ est la résolution séquentielle qui introduit des délais. Différents ordonnancements sont utilisés à partir des heuristiques du FAS. Pour une résolution simultanée, les méthodes sont JM (formulation de la Jacobienne) et JM (Jacobienne réduite) avec l'heuristique du FAS sélectionnée pour réduire le rang de la Jacobienne. Lorsque * suit le nom de la méthode, cela indique que l'on itère sur la méthode.

Tableau 6-1 : Méthodes de résolution testées.

Méthode	Version Itérative	Description
SEQ+DFS	-	Solution séquentielle basée sur l'ordre post-visite inversé du DFS.
SEQ+DFSOpt	-	Solution séquentielle basée sur l'ordre de DFSOpt.
SEQ+GreedyFAS	-	Solution séquentielle basée sur l'ordre de GreedyFAS.
SEQ+SortFAS*	-	Solution séquentielle basée sur l'ordre de SortFAS*.
JM	JM*	Méthode de la Matrice Jacobienne.
RJM+DFS	RJM*+DFS	Méthode de la Jacobienne Réduite avec DFS pour le problème FAS.
RJM+DFSOpt	RJM*+DFSOpt	Méthode de la Jacobienne Réduite avec DFSOpt pour le problème FAS.
RJM+GreedyFAS	RJM*+GreedyFAS	Méthode de la Jacobienne Réduite avec GreedyFAS pour le problème FAS.
RJM+SortFAS*	RJM*+SortFAS*	Méthode de la Jacobienne Réduite avec SortFAS* pour le problème FAS.

Les cas pratiques suivants de réseaux avec des contrôles modélisés sous forme de schémas-blocs sont testés (disponibles dans [40][125]) :

- Une boucle à verrouillage de phase (PLL) [40][127].
- La modélisation de l'arc secondaire d'un disjoncteur via des schémas-blocs de contrôle [128].
- Un modèle de parc éolien agrégé de 45 éoliennes. Il est testé avec deux types de technologies, un parc à convertisseur pleine échelle (Full Scale Converter Wind Park, FSC WP) et un autre à générateur à induction à double alimentation [132] (Doubly Fed Induction Generator, DFIG WP). Un modèle moyen est utilisé pour les convertisseurs AC/DC.
- Le cas IEEE39 [125] contenant deux parcs éoliens en DFIG et en FSC, 8 machines synchrones (SM).
- Le réseau T0 [125] avec 72 SMs et 10 parcs éoliens en DFIG.

On reprend les mêmes métriques que celles du paragraphe 4.6 pour mesurer la précision des méthodes, à savoir l'erreur relative en scalaire $e\%$ et en série temporelle e_t . Pour mesurer les performances, on utilise le speedup ratio s défini ci-dessous :

$$s = T_{\bar{f}}/T_f \quad (6-60)$$

Où $T_{\tilde{f}}$ est le temps d'exécution de la solution de référence \tilde{f} et T_f pour la solution f étudiée.

6.5.1 Analyse des graphes de contrôle

L'analyse des graphes de contrôle permet de se rendre compte de la complexité de chaque cas test et de l'efficacité des heuristiques du FAS dans la réduction du rang de la matrice Jacobienne. Tout d'abord, le Tableau 6-2 montre la complexité de chaque contrôle où le nombre de signaux est le nombre de sommets du graphe associé. On remarque l'efficacité de la décomposition en SCCs qui réduit la taille des graphes à résoudre, de plusieurs milliers de sommets pour le graphe original à moins d'une centaine pour les SCCs cycliques. Cela réduit d'autant l'ordre du système linéaire à résoudre issu de la formulation de la matrice Jacobienne.

Tableau 6-2: Complexité des systèmes de contrôle pour chaque cas test.

Cas test	Nb Signaux	Nb SCCs cycliques	Taille Max des SCCs
PLL	22	1	14
Arc Secondaire	29	1	5
FSC WP	136	6	16
DFIG WP	1608	8	57
IEEE39	4524	40	57
Réseau T0	17007	135	57

Puis, les différentes heuristiques pour résoudre le problème du FAS sont testées sur les cas de contrôle. Le Tableau 6-3 montre le nombre total de variables de coupes/feedback trouvées par chaque heuristique pour l'ensemble des SCCs cycliques. Le Tableau 6-4 détaille pour le réseau IEEE39 le nombre de coupes de feedback pour chaque SCC cyclique. N est la taille de chaque SCC et N_r le ratio de réduction de l'ordre de la matrice Jacobienne à considérer.

Tableau 6-3 : Nombre de variables de feedback pour chaque heuristique du FAS.

Cas Test	DFS	DFSOpt	GreedyFAS	SortFAS	SortFAS*
PLL	2	2	2	2	2
Arc Secondaire	1	2	1	1	1
FSC WP	14	8	7	6	6
DFIG WP	22	11	11	10	9
IEEE39	116	99	60	86	57
Réseau T0	275	165	165	155	145

Tableau 6-4 : Détail du nombre de variables de feedback pour le réseau IEEE39 pour chaque type de contrôle.

Systèmes de Contrôle	ID du SCC	DFS	DFSOpt	SortFAS*	N	N_r
SM	1	1	1	1	5	5
	2	1	1	1	4	4
FSC WP	1	11	9	3	47	15.67
	2,3,4	4	4	2	20	10
	5	1	1	1	4	4
	6	3	1	1	9	9
	7	3	1	1	16	16
	8	2	2	1	53	53
	9	2	2	1	11	11
DFIG WP	1,2	11	9	3	47	15.67
	3,4,5,6,7,8,9	4	4	2	20	10
	10	1	1	1	4	4
	11,12	3	1	1	9	9
	13	3	1	1	16	16
	14	4	3	2	57	28.5
	15	2	1	1	15	15

Si l'on prend le SCC ID = 8 pour le contrôle du FSC WP, la matrice Jacobienne 53×53 ($N_r = 53$) est réduite en un scalaire pour SortFAS* ce qui simplifie grandement la résolution du système linéaire associé. Pour l'ensemble des SCCs cycliques des cas test, le nombre optimal de coupes de feedback

excède rarement 3. Pour l'ensemble des cas, le temps d'exécution des heuristiques est très rapide (pas plus d'une seconde). En effet, les tailles des SCC cycliques ne dépassent pas la centaine de sommets. Au niveau de la qualité des solutions obtenues (nombre de coupes de feedback le plus faible possible), on obtient les mêmes conclusions que [43]. Avec une stratégie de minimisation du nombre de back arcs, GreedyFAS et SortFAS* surpassent le DFS et son variant DFSOpt jusqu'en diminuant de 50% le nombre de variables de feedback à considérer. SortFAS* est légèrement mieux que GreedyFAS. Il n'est pas pénalisé par le temps d'exécution car les tailles des SCCs sont extrêmement petites, en comparaison aux graphes des données du Web (réseaux sociaux, sites, emails...) étudiés dans [43] (ordre de la centaine de milliers de sommets). Ces gains sur l'ordre de la Jacobienne doivent être confirmés pour la simulation temporelle.

6.5.2 Validation

6.5.2.1 Cas de la simulation d'une PLL

Le cas de la PLL est illustré par la Figure 6-11 avec les coupes de feedback pour chaque heuristique du FAS, DFS, DFSOpt et GreedyFAS. SortFAS n'améliore pas la séquence initiale issue du DFS. Ce contrôle est simulé pendant 300 ms avec échelon de 1000% sur l'amplitude du cosinus de l'entrée u de 500 à 5000 à $t = 125$ ms. Différents pas de temps pour l'intégration numérique des équations du contrôle sont testés de $1 \mu\text{s}$ à $35 \mu\text{s}$. Les figures 6-12 et 6-13 montrent l'amplitude synthétisée par la et l'erreur relative vis-à-vis de la méthode Jacobienne itérative (RJM*). Une tolérance de 10^{-4} est utilisée pour la convergence des itérations. Ainsi, on constate que pour un pas de temps de $7 \mu\text{s}$, l'ensemble des méthodes ont convergé. Une superposition parfaite est obtenue sur les sorties de la PLL.

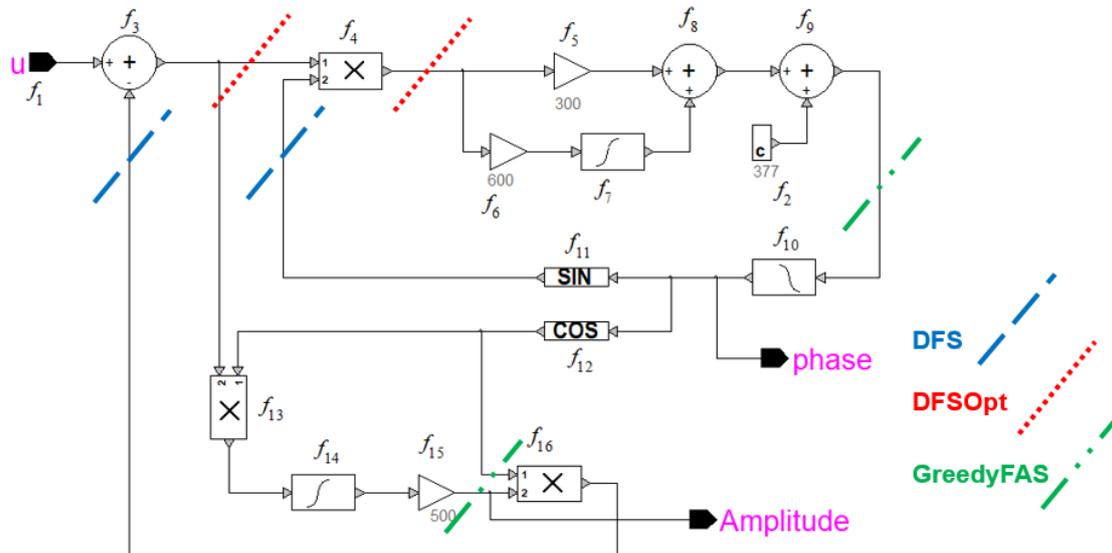


Figure 6-11 : Coupes FAS du PLL pour différentes heuristiques.

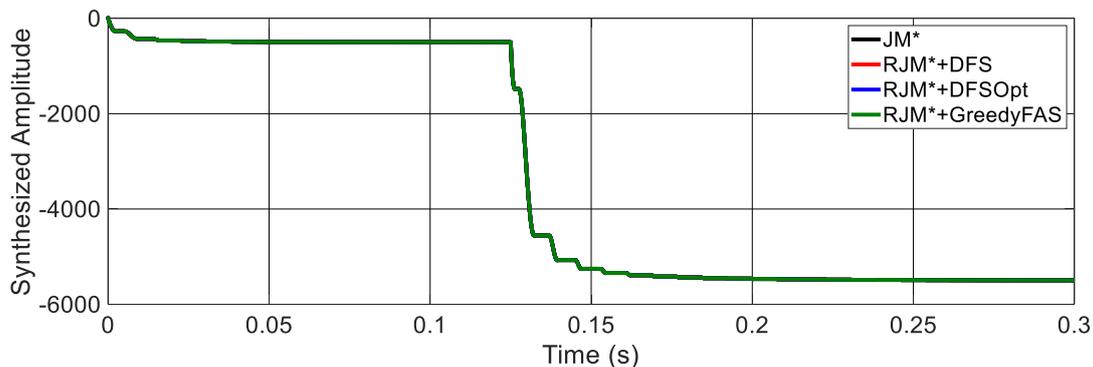


Figure 6-12 : Amplitude synthétisée pour chaque heuristique du FAS (Pas de temps de $7 \mu\text{s}$).

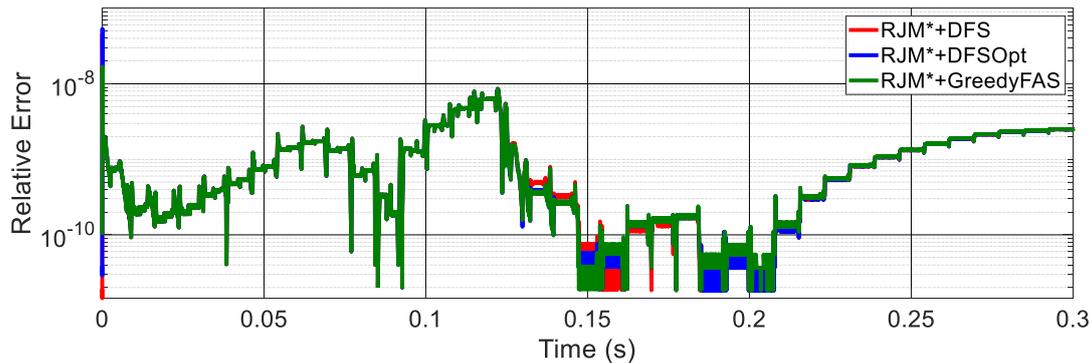


Figure 6-13 : Erreur relative par rapport à JM* sur l'amplitude synthétisée pour chaque heuristique du FAS.

Surtout, ce cas démontre bien l'intérêt des méthodes simultanées (JM et RJM) sur la méthode séquentielle du DFS (SEQ+DFS). Les mêmes conclusions peuvent être tirées avec les autres ordonnancements des heuristiques du FAS pour SEQ. En effet, avec un pas de temps supérieur à 1 μ s, SEQ+DFS est numériquement instable. Le Tableau 6-5 montre les gains de performance pour JM* et RJM* vis-à-vis de SEQ+DFS (à 1 μ s) pour 10 PLL de la Figure 6-11. Les erreurs relatives de l'amplitude synthétisée sont affichées dans le Tableau 6-6. Les cellules grisées concernent des problèmes de convergence et/ou d'instabilité numérique ($e\% > 10\%$) avec le symbole (-) ou d'une précision dégradée ($e\% > 1\%$). Même avec des itérations, l'utilisation d'un pas temps plus élevé pour JM* et RJM* permet d'accélérer la simulation par rapport à SEQ+DFS tout en gardant la même précision. De plus la réduction du rang de la matrice Jacobienne de RJM* améliore les performances vis-à-vis de JM*. Par ailleurs, pour un pas de temps supérieur à 7 μ s, on peut noter que la précision n'est pas exactement la même pour les différentes heuristiques du FAS utilisées par RJM*. Des variations dans le calcul de la matrice Jacobienne peuvent modifier la convergence et la précision de la méthode itérative de Newton.

Tableau 6-5 : Speed up ratio des méthodes itératives Jacobiennes par rapport à SEQ+DFS (pas de temps à 1 μ s).

Méthodes de Résolution	Pas de Temps (μ s)								
	1	5	7	10	15	20	25	30	35
SEQ+DFS	1.0	-	-	-	-	-	-	-	-
JM*	0.2	0.9	1.3	1.7	2.4	3.1	3.5	4.6	4.5
RJM*+DFS	0.3	1.5	2.0	3.0	4.2	-	-	-	9.2
RJM*+DFSOpt	0,3	1.5	2.0	-	4.3	5.9	-	9.4	-
RJM*+GreedyFAS	0.3	1.5	2.0	2.9	4.1	5.3	7.1	-	-

Tableau 6-6 : Erreur relative des méthodes de résolution par rapport à JM* (pas de temps à 1 μ s).

Méthodes de Résolution	Pas de Temps (μ s)						
	1	5	7	10	15	20	25
SEQ	5e-3	19.6	36.9	57.6	83.3	100	116
JM*	0	0.0058	0.0088	0.8	0.96	0.81	1.2
RJM*+DFS	1.4e-7	0.0058	0.0088	0.8	0.96	4.0e5	2.9e7
RJM*+DFSOpt	1.4e-7	0.0058	0.0088	-	1.16	1.5	-
RJM*+GreedyFAS	1.4e-7	0.0058	0.0088	0.014	0.0131	0.036	0.657

6.5.2.2 Cas de la simulation de l'arc secondaire d'un disjoncteur

Lors d'un défaut phase-terre sur un réseau électrique (par exemple sur une ligne), le défaut est localisé et éliminé de manière automatique via des systèmes de protection. La phase en défaut est

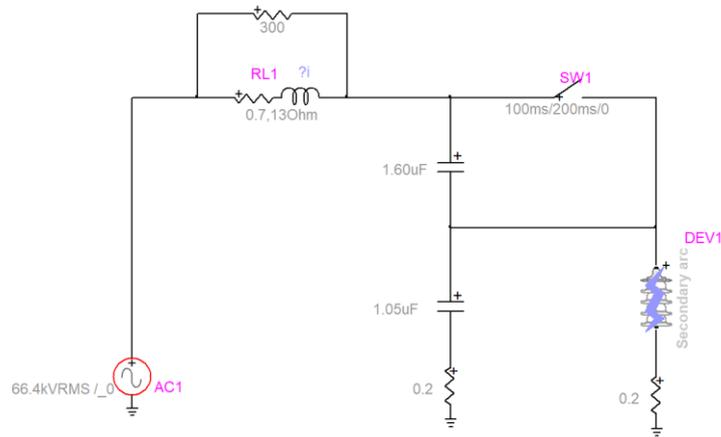


Figure 6-15 : Cas de simulation d'un arc secondaire pour un réseau 66 kV.

Les figures 6-16 et 6-17, ci-dessous, montrent le courant et la tension de l'arc pour les trois méthodes JM* (pas de temps de 10 μ s, solution de référence), SEQ+DFS (20 μ s) et RJM+DFS (50 μ s). On remarque que RJM offre une bonne précision sans itération et avec un pas de temps élevé par rapport à la solution de référence. En revanche, la résolution séquentielle échoue à bien simuler le phénomène à partir d'un pas de temps de 20 μ s. L'ajout d'un pas de temps artificiel induit une erreur du contrôle sur l'état de l'arc. On retrouve cette erreur quelle que soit l'heuristique du FAS adoptée pour l'ordonnancement de SEQ. L'arc est éliminé à $t = 0.4$ s (tension de rétablissement atteinte et courant de l'arc qui s'annule) alors que dans la solution de référence (ou avec des pas de temps plus petits), il perdure tout le long de la simulation. Une résolution simultanée (JM ou RJM) est nécessaire si l'on veut modéliser des phénomènes physiques avec des blocs de contrôle.

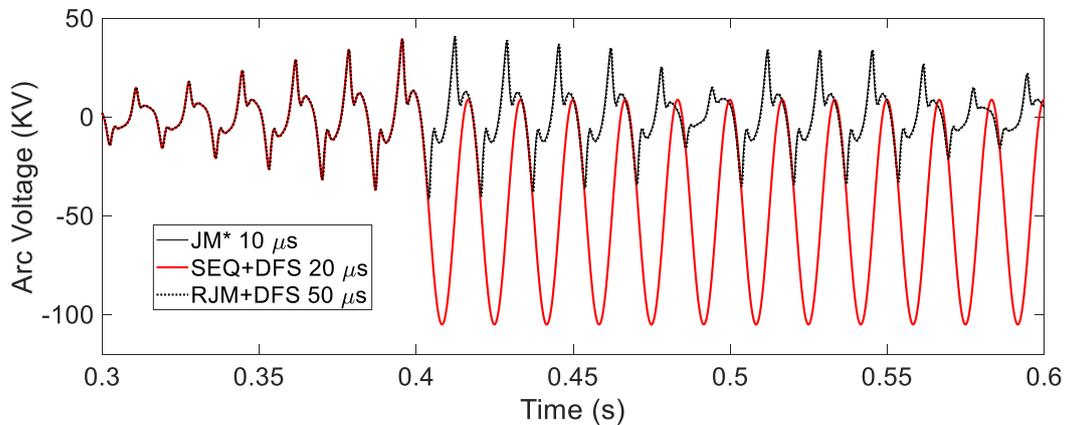


Figure 6-16 : Tension de l'arc secondaire pour JM* (10 μ s), SEQ+DFS (20 μ s) et RJM+DFS (50 μ s).

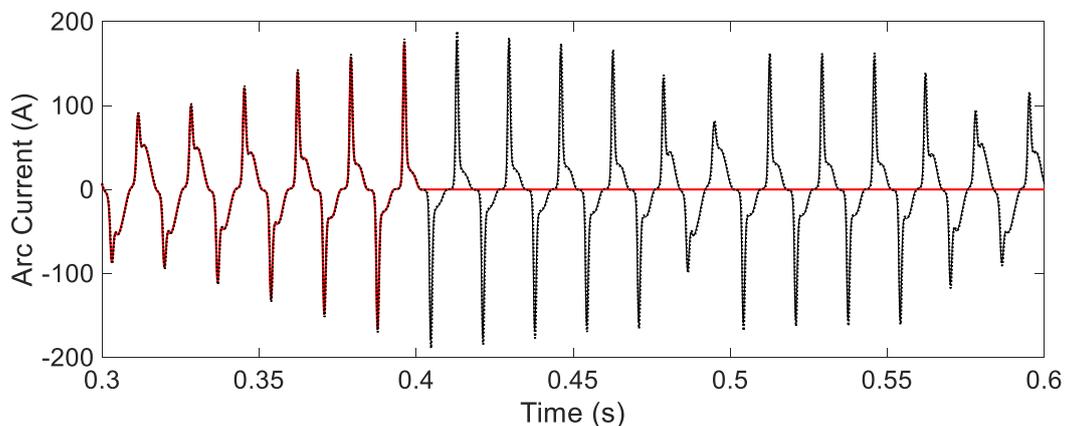


Figure 6-17 : Courant de l'arc secondaire pour JM* (10 μ s), SEQ+DFS (20 μ s) et RJM+DFS (50 μ s).

Au niveau des performances, RJM+DFS présente un speed up ratio de 1.19 vis-à-vis de la solution de référence JM. Cela confirme les gains de performance issus de la réduction de la matrice Jacobienne qui de 5x5 devient un scalaire pour ce cas. Par ailleurs, pour une solution séquentielle plus précise à 15 μ s (sans élimination de l'arc), le speed ratio est de 1.52 pour RJM à 50 μ s. La simultanéité de la solution permet d'augmenter le pas de temps pour accélérer la simulation tout en gardant une bonne précision.

6.5.2.3 Cas de la simulation de parcs éoliens

Les cas tests de parcs éoliens utilisent le même réseau que dans [132]. Ils consistent en un parc éolien connecté à un petit réseau de 120 kV à 60 Hz (Figure 6-18). Un modèle moyen est utilisé pour représenter les convertisseurs AC/DC de chaque éolienne de 1.5 MW du parc (45 au total). Deux types d'éolienne sont testés : un parc avec des éoliennes en FSC (cas FSC WP) et un autre avec des éoliennes en DFIG (cas DFIG WP). Pendant une simulation de 3 s avec un pas de temps de 40 μ s, un défaut double phase-terre arrive sur le BUS4 à $t = 1$ s. Comme les itérations peuvent ralentir la simulation, elles ne sont pas activées pour JM et RJM. Presque toutes les heuristiques couplées avec RJM (DFS, GreedyFAS et SortFAS*) donnent des résultats similaires pour chaque type de turbine dans les figures 6-19 et 6-19. Seule RJM+DFSOpt (en bleu), pour les turbines en DFIG, a une précision dégradée lors des transitoires. En effet, cela montre bien que des séquences de boucle différentes changent le chemin de perturbation de Φ pour le calcul de la matrice Jacobienne. Cela peut avoir un impact sur la précision des résultats. Cependant, plus le nombre de variables de feedback est faible (RJM+GreedyFAS et RJM+SortFAS*), plus la solution est précise. Un nombre minimal de variables de feedback limite la propagation d'erreurs numériques de la méthode des perturbations.

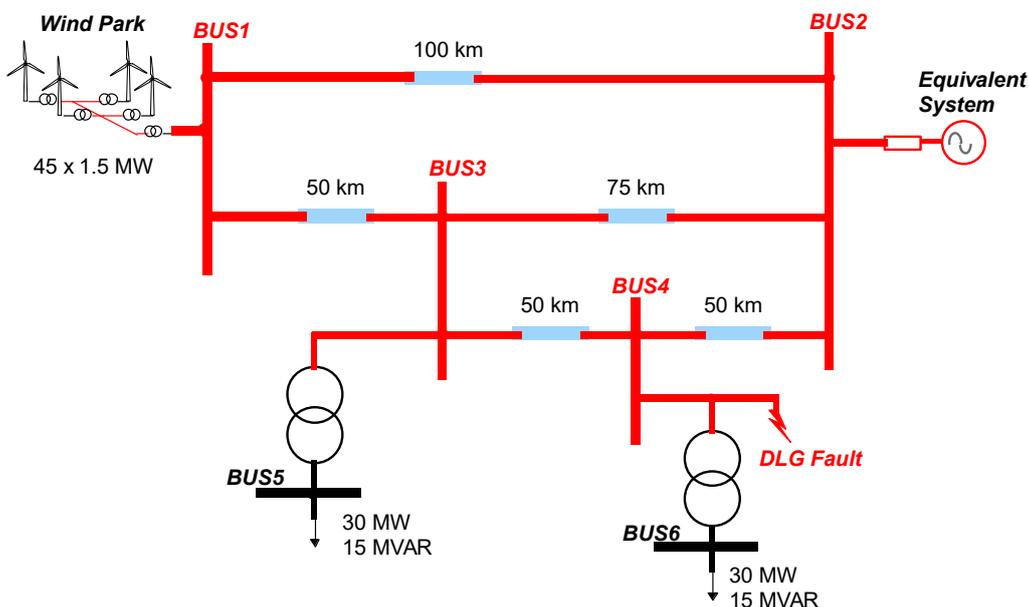


Figure 6-18 : Réseau pour les cas test des parcs éoliens.

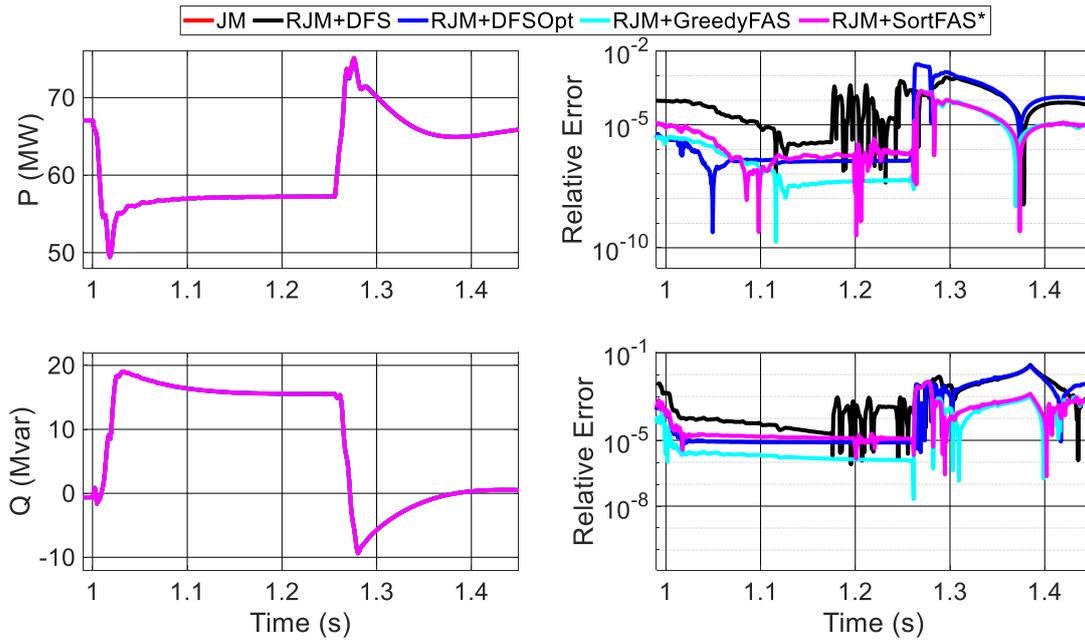


Figure 6-19 : Résultats de simulation (JM, RJM + heuristique du FAS) pour le cas FSC WP, puissances active et réactive du parc avec l'erreur relative associée (par rapport à JM).

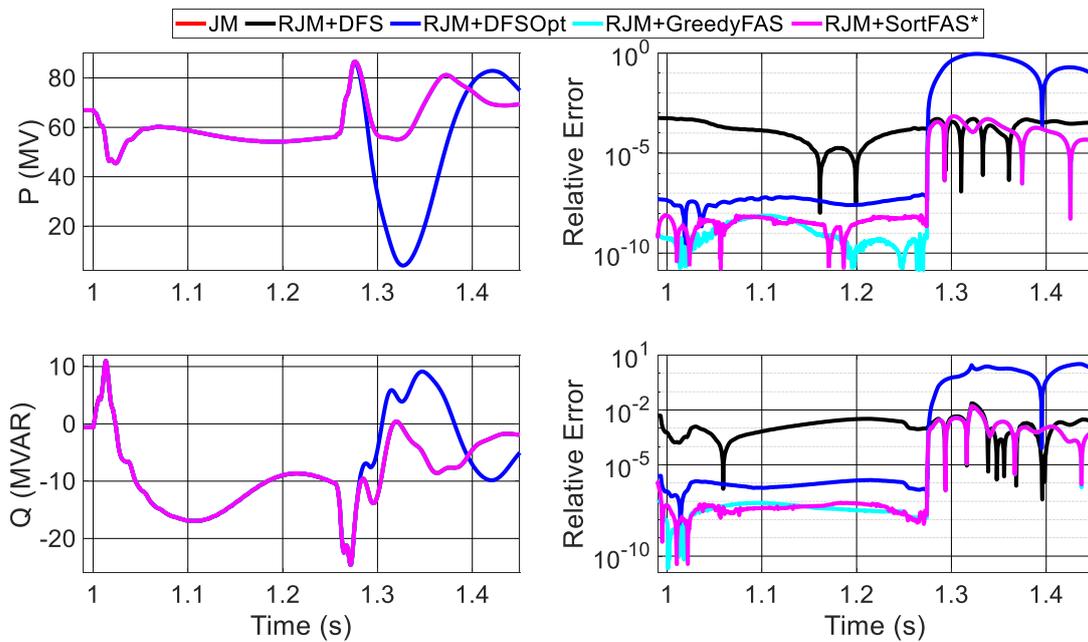


Figure 6-20 : Résultats de simulation (JM, RJM + heuristique du FAS) pour le cas DFIG WP, puissances active et réactive du parc avec l'erreur relative associée (par rapport à JM).

Par ailleurs, on teste également la solution séquentielle avec les différentes heuristiques du FAS. La Figure 6-21 montre les résultats de simulation comparés à la solution de référence JM. La plupart des ordonnancements testés dégradent la précision par rapport à une solution simultanée lorsque l'événement transitoire survient. L'ordonnancement selon DFS, le plus intuitif, se révèle le plus imprécis, la réponse se met à osciller après le défaut. Seule SEQ+DFSOpt présente une précision équivalente à une solution sans délai. Minimiser le nombre de délais ne semble pas suffisant pour garantir une bonne précision. Il faut prendre également en compte la localisation où l'on insère le délai.

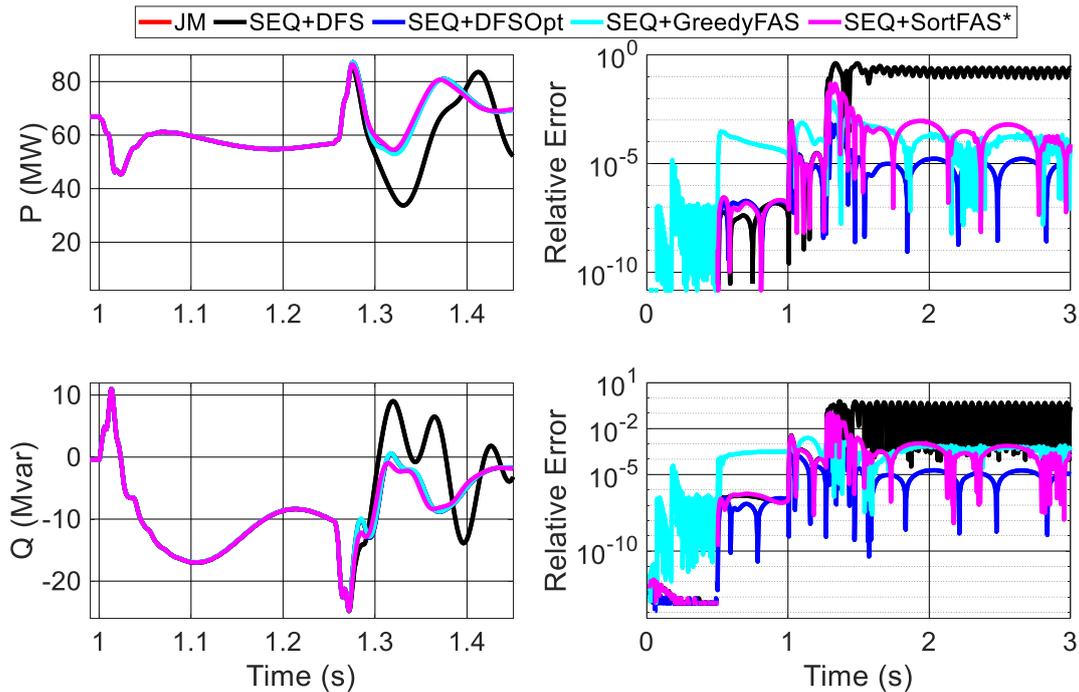


Figure 6-21 : Résultats de simulation (JM, SEQ + heuristique du FAS) pour le cas DFIG WP, puissances active et réactive du parc avec l'erreur relative associée (par rapport à JM).

6.5.3 Analyse des Performances

Les cas tests suivant sont considérés pour mesurer les performances de RJM avec les différentes heuristiques du FAS (DFS, DFSOpt, GreedyFAS et SortFAS*) ainsi que la méthode SEQ :

- Cas FSC WP et DFIG WP, simulation identique à celle étudiée au paragraphe 6.5.2.3.
- Réseau IEEE39 (Figure 6-22) avec une simulation de 4 s et un pas de temps de 20 μ s. Un défaut phase-terre se produit à $t = 2$ s. La ligne en défaut est isolée à $t = 2.1$ s.
- Réseau T0 (Figure 6-23) avec une simulation de 3 s et un pas de temps de 50 μ s. Un défaut phase-terre se produit à $t = 1$ s. La ligne en défaut est isolée à $t = 1.58$ s.

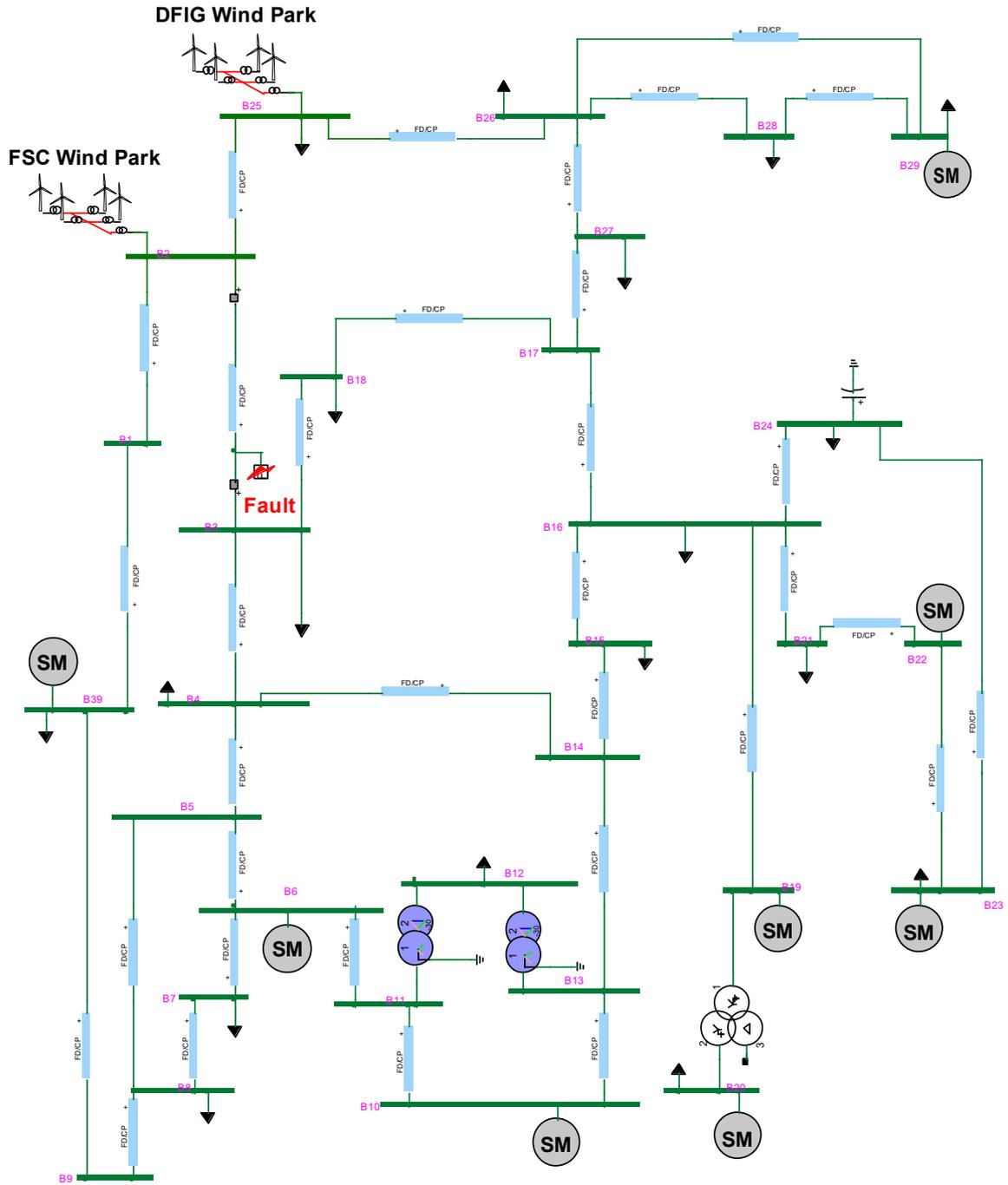


Figure 6-22 : Cas test IEEE39 avec de la production éolienne.

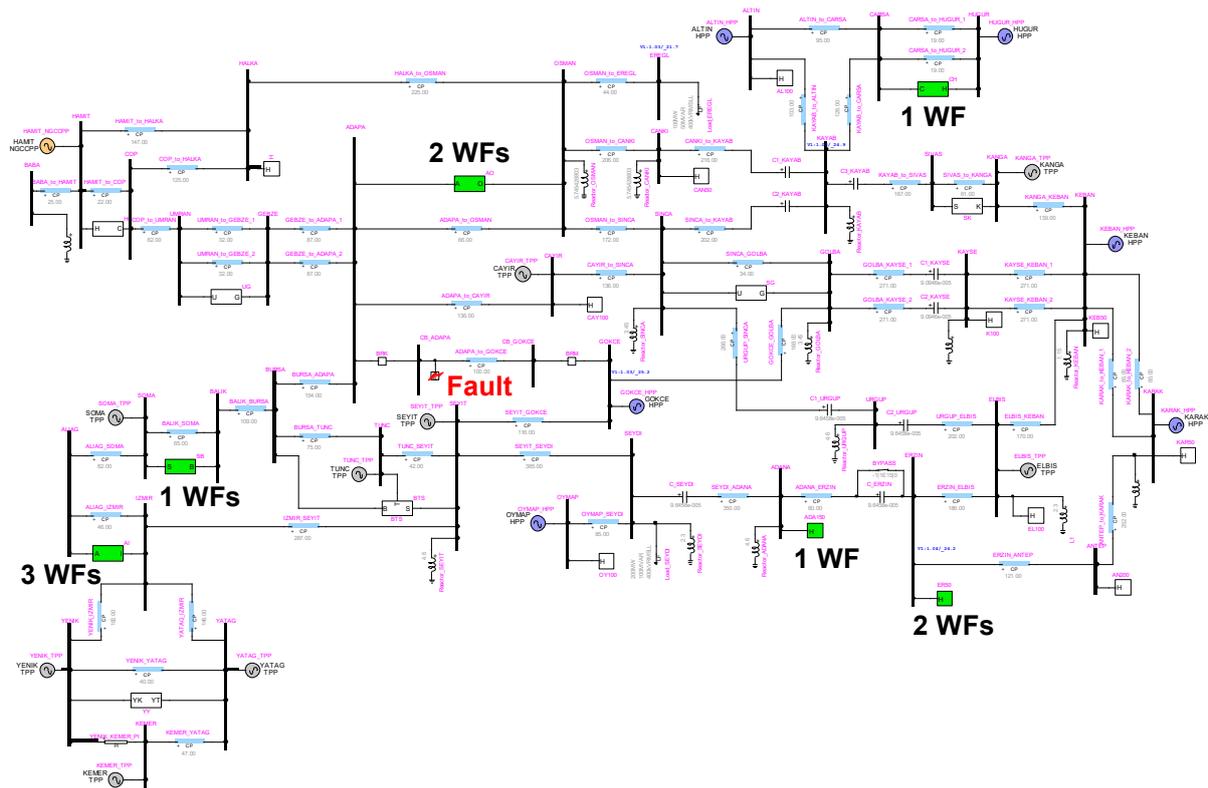


Figure 6-23 : Vue générale du réseau T0 avec de la production éolienne.

Comme précédemment, les itérations ne sont pas activées pour JM et RJM afin d’avoir de meilleures performances. Le Tableau 6-7 montre le temps d’exécution total (temps CPU) de la résolution du réseau et de leurs systèmes de contrôle en utilisant JM. Les simulations ont été lancées sur un PC portable muni d’un processeur Intel i7-1165G7-CPU @ 2.80 GHz. Le speed up ratio est calculé pour RJM et SEQ avec différentes heuristiques du FAS. Les cellules grisées indiquent une divergence (-) de la solution de réseau ou des problèmes numériques. Dans le Tableau 6-8, l’erreur relative $e\%$ est calculée sur la puissance réactive du parc éolien le plus proche de la zone en défaut. Comme attendu, les gains de performances de RJM dépendent du nombre de variables de feedback. Moins il y en a, plus la simulation temporelle est rapide. RJM+SortFAS* peut être choisie comme la méthode RJM la plus efficace en termes de performance et de stabilité numérique de la solution obtenue. Les méthodes séquentielles SEQ sont plus rapides que JM et RJM car elles n’utilisent pas de formulations matricielles. Pour la plupart des séquences choisies, issues des heuristiques du FAS, on constate une précision dégradée. Seul SEQ+DFSOpt a une précision équivalente aux méthodes simultanées. Pour ces simulations, SEQ+DFSOpt peut être utilisé afin d’accélérer la simulation. Cependant, l’ajout de délais artificiels ne garantit pas, pour d’autres scénarios de simulation, que la précision soit préservée.

Tableau 6-7 : Speed up ratio de RJM et SEQ sur JM pour chaque heuristique du FAS.

Cas test	JM CPU Time	RJM+ DFS	RJM+ DFSOpt	RJM+ GreedyFAS	RJM+ SortFAS*	SEQ
FSC WP	12.1 s	1.03	1.15	1.20	1.25	1.43
DFIG WP	14.4 s	0.95	1.11	1.08	1.18	1.46
IEEE39	182.3 s	-	0.94	1.18	1.19	1.66
T0-grid	300.4 s	1.07	1.16	1.16	1.20	1.32

Tableau 6-8 : Erreur relative e% des méthodes RJM et de SEQ par rapport à JM.

Cas test	RJM+ DFS	RJM+ DFSOpt	RJM+ GreedyFAS	RJM+ SortFAS*	SEQ+ DFS	SEQ+ DFSOpt	SEQ+ GreedyFAS	SEQ+ SortStar
FSC WP	0.14	0.12	0.22	0.14	9.2	0.14	0.78	3.0
DFIG WP	0.21	47.4	0.11	0.11	127.4	0.11	0.68	6.2
IEEE39	-	4.2	0.6	0.75	20.2	0.72	2.2	2.2
T0-grid	0.29	26.7	0.23	0.23	108	0.24	0.41	0.41

6.6 Conclusion

Ce chapitre a identifié les méthodes pour accélérer la résolution des équations de contrôle lors d'une simulation EMT. Suivant la rapidité et/ou la précision demandée, on a trois options pour simuler les contrôles : RAPIDE avec l'utilisation de la méthode séquentielle SEQ où l'ordonnancement des fonctions blocs est issu d'une heuristique du FAS, DFSOpt semble la plus stable; PRECIS avec la méthode de la matrice Jacobienne JM qui garantit une simultanéité de la solution ; PRECIS et RAPIDE avec la Jacobienne réduite qui grâce aux meilleures heuristiques du FAS (GreedyFAS et SortFAS*) arrive à réduire le nombre de variables à considérer pour la méthode de Newton. L'optimisation de la Jacobienne réduite est la contribution principale de ce chapitre. Une formulation plus simple du calcul des séquences de boucle via une labélisation Source/Puits a facilité l'intégration de la méthode dans un outil de simulation EMT [2]. L'investigation du problème du Feedback Arc Set minimal a permis de sélectionner les heuristiques les plus performantes pour réduire la taille de la matrice Jacobienne et *in fine* d'accélérer la simulation. La Jacobienne réduite a été validée sur des cas pratiques de contrôle de grands réseaux comportant des productions d'énergies renouvelables. En étant jusqu'à 25% plus rapide, elle a une précision équivalente à JM. La méthode séquentielle a également été explorée. Si pour certains cas (boucles algébriques, modélisation de phénomènes physiques), l'ajout de délais artificiels est réhibitoire pour la précision, pour d'autres, la précision est préservée jusqu'en augmentant de 66% la vitesse de simulation. Elle doit être utilisée avec parcimonie car la précision sur l'ensemble des scénarios ne sera pas autant garantie qu'avec une méthode de résolution simultanée.

C'est dans cet axe de recherche que l'on trouve les apports les plus théoriques de ce mémoire. Tout d'abord, il a fallu bien comprendre la méthode originelle qui mêlait différents domaines : analyse numérique via la résolution de Newton, analyse des graphes pour obtenir les séquences de boucle et minimiser le nombre de variables et la simulation de systèmes de contrôle de réseaux électriques. Puis, en faisant le lien entre ces différents domaines, cela a permis d'identifier un problème connexe tel que le FAS minimal. Utilisé à l'origine pour un réseau social afin, par exemple, de retirer des fausses informations, il a été appliqué à la réduction du rang de la matrice Jacobienne pour accélérer la simulation de systèmes de contrôle de réseaux électriques. Ainsi, j'ai développé une capacité de transposition. Je m'inspire de méthodes de résolution pour des cas d'applications en dehors de mon domaine, et je les applique à mes propres thématiques de recherche.

Chapitre 7 Conclusion sur les contributions

Les chapitres 3-6 ont détaillé les contributions qui ont été apportées à l'accélération de la vitesse de simulation des transitoires électromagnétiques d'un réseau électrique, tout en préservant la précision. Différentes techniques ont été explorées : l'utilisation d'un équivalent fréquentiel de réseau (FNDE), la parallélisation des équations de réseaux et la Jacobienne réduite pour résoudre les boucles de rétroaction des équations de contrôle. A chaque fois, des cas industriels (HVDC et production d'énergie renouvelable), reflètent des mutations du réseau du fait de la transition énergétique, ont permis de valider les gains de performance de ces contributions. La plupart des conclusions obtenues sont valables pour les deux environnements de simulation, temps différé d'EMTP® et temps réel d'HYPERSIM.

Ainsi, dans le Chapitre 3, la mise en place d'un modèle d'équations d'état, en utilisant une représentation creuse des matrices, a interfacé un équivalent fréquentiel de réseau (FDNE) à la simulation temps réel. La modélisation en un FDNE des différentes sections d'un câble d'une liaison HVDC a permis d'accélérer les performances pour pouvoir simuler en temps réel.

C'est dans les deux chapitres suivants que l'on trouve les apports les plus conséquents. Ils concernent la parallélisation de la résolution des équations de réseaux dans leur séparation en tâches. La méthode de Compensation (Chapitre 4) permet de paralléliser n'importe où la résolution des équations de réseaux sans se baser sur le délai de propagation des lignes, tout en conservant une bonne précision. Ceci est bénéfique pour les réseaux d'électronique de puissance (cas HVDC) et les réseaux de distribution où l'insertion de délais artificiels pour paralléliser peut dégrader la précision de la simulation. L'implémentation de la méthode de Compensation dans un outil de simulation temps réel tel qu'HYPERSIM a permis de tester pour la première fois cette méthode sur un large éventail de cas industriels de simulation temps réel. Ceci a validé une accélération des performances par cette méthode tout en préservant la précision via une résolution simultanée des équations de réseaux. Le recoupement théorique de la méthode de Compensation, avec d'autres techniques de parallélisation de la résolution d'un système linéaire, laisse entrevoir des travaux futurs. Il serait intéressant d'étudier les performances des méthodes BBD générales évoquées au paragraphe 4.4.3.2. Tout comme ce qui a été fait pour la méthode de Compensation, dans l'idéal, une implémentation de ces différentes méthodes doit être faite dans un outil de simulation EMT tels que EMTP® ou HYPERSIM pour avoir accès à de nombreux cas de test les plus réalistes possibles.

Par ailleurs, au Chapitre 5, les techniques les plus adéquates ont été identifiées pour la répartition automatique des tâches, issues de la séparation précédente via le délai des lignes ou la méthode de Compensation, sur les processeurs de simulation. Les méthodes de partitionnement de graphes multi-niveaux, comme la Bipartition Réursive multi-niveaux, se sont révélées être les plus efficaces pour à la fois obtenir une charge équilibrée des processeurs et minimiser le temps de communication entre les processeurs. Même si la contrainte du temps réel est relâchée, la mise en place d'un paramétrage de l'algorithme plus optimisé ainsi que la comparaison aux méthodes exactes montrent que la solution obtenue reste de bonne qualité. Là aussi, une validation sur des cas concrets de simulation temps réel a permis de conforter cette conclusion. Dans un axe de recherche

futur, ces méthodes de partitionnement de graphes peuvent être appliquées également à l'automatisation de la séparation en tâches comme cela a été évoqué dans le paragraphe 4.4.4.

Dans le Chapitre 6, des apports théoriques ont amélioré la Jacobienne réduite qui a été introduite pour accélérer la résolution des boucles de rétroaction. Une formulation plus simple et plus générique permet d'obtenir les séquences des boucles, primordiales pour le calcul de la Jacobienne réduite via la méthode des perturbations. Une formulation en un problème du FAS (Feedback Arc Set, ensemble des arcs de rétroaction) minimal a permis de minimiser le rang de la Jacobienne en utilisant les heuristiques les plus performantes pour résoudre ce type de problème, comme SortFAS* ou GreedyFAS. La méthode séquentielle a également été investiguée et enrichie en explorant les ordonnancements des heuristiques du FAS. Il a été identifié les cas où une résolution simultanée est nécessaire. La Jacobienne réduite y accélère la simulation. Mais pour d'autres cas, la résolution séquentielle, via l'insertion de délais, permet d'avoir une simulation plus rapide tout en conservant la même précision. Des travaux futurs pourraient consister à étudier les caractéristiques des boucles de rétroaction afin de savoir s'il faut construire la matrice Jacobienne ou s'il faut rester sur une solution séquentielle. Egalement, il serait approprié d'identifier les points de coupe les plus pénalisants sur la précision de la solution pour une méthode séquentielle ou pour la Jacobienne réduite.

Conclusion

Tout d'abord, cette VAE doctorale a montré les différentes compétences d'un chercheur que j'ai acquises au cours de mon activité professionnelle. Sept années en tant qu'ingénieur R&D à RTE m'ont progressivement fait passer du développement informatique à la recherche dans le domaine de la simulation des transitoires électromagnétiques d'un réseau électrique. Ainsi, le développement informatique, à partir d'un besoin utilisateur, a cédé la place à une conception et une élaboration d'une démarche de recherche et de développement. Comme en témoigne la description de mon parcours, j'ai réalisé lors des cinq dernières années une production scientifique significative qui a été validée par mes pairs : huit papiers scientifiques publiés, un en prépublication et trois rapports internes. Mes compétences en développement informatique m'ont permis de transférer ces travaux vers des applications industrielles dans les outils de simulation de transitoires électromagnétiques. J'ai également diffusé ma culture scientifique par des interventions dans des colloques internationaux ou via l'encadrement d'un mastère recherche. Ma curiosité en recherche se nourrit également des nombreuses relectures que j'effectue pour des revues du domaine.

Mais avant tout, ce sont mes contributions scientifiques qui donnent toute son sens à cette thèse. Elles s'inscrivent dans les grandes mutations des enjeux environnementaux et industriels du XXI^e siècle que porte la Transition Energétique. Le réseau électrique est en train d'incorporer un foisonnement de dispositifs d'électronique de puissance afin d'accueillir au mieux les énergies renouvelables. La simulation des transitoires électromagnétiques, en temps différé ou en temps réel, se retrouve au cœur de cette transition en étant un outil d'étude qui facilite ces intégrations. Mes contributions se sont focalisées sur l'accélération de la vitesse de calcul et sur la précision de la simulation. Un panorama assez complet des techniques d'accélération de la simulation temporelle a été exploré : équivalent fréquentiel de réseau, techniques de parallélisation et Jacobienne réduite pour les équations de contrôle. C'est dans les techniques de parallélisation que l'on trouve les contributions qui ont un plus grand impact pratique. Cela a permis d'avoir à disposition des modèles encore plus précis en pouvant les simuler en temps réel. Des apports théoriques dans l'amélioration de la Jacobienne réduite ont permis d'accélérer la résolution des équations de contrôle. En définitive, ce mémoire de VAE doctorale constitue un document qui capitalise l'ensemble de mes projets de recherche de ces cinq dernières années sur les techniques d'accélération de la simulation de transitoires électromagnétiques.

Si cette démarche de VAE doctorale touche à sa fin, il n'en est rien pour mes recherches. De nombreuses pistes restent à explorer non seulement pour les techniques de parallélisation mais encore pour la résolution des équations de contrôle. Indéniablement, les compétences acquises durant cette démarche de VAE sont des plus utiles pour la poursuite de ces travaux.

Références

- [1] V. Q. Do, J.-C. Soumagne, G. Sybille, G. Turmel, P. Giroux, G. Cloutier and S. Poulin, "Hypersim, an integrated real-time simulator for power networks and control systems," in *Proc. ICDS'99*, Vasteras, Sweden, May 1999, pp. 1-6.
- [2] J. Mahseredjian, S. Denetière, L. Dubé, B. Khodabakhchian, and L. Gérin-Lajoie, "On a new approach for the simulation of transients in power systems," *Elect. Power Syst. Res.*, vol. 77, no. 11, pp. 1514–1520, Sep. 2007.
- [3] *Futurs Energétiques 2050 – Principaux Résultats*, RTE, Oct. 2021, pp. 1-64.
- [4] *Bilan Electrique 2020*, RTE, Jan. 2021, pp 1-185.
- [5] *Trans-European Infrastructure Energy – Project of Common Interest*, European Commission, 2014, pp. 1-264.
- [6] Simulink. The MathWorks Inc, Natick, MA, USA. [Online]. Available: <http://www.mathworks.com>
- [7] B. Bruned, C. Martin, S. Denetière and Y. Vernay, "Implementation of a unified modelling between EMT tools for Network Studies," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, 2017, pp. 1-7.
- [8] S. Denetière, B. Bruned, H. Saad and E. Lémieux, "Task separation for real-time simulation of the CIGRE DC grid benchmark," in *Proc. Power Syst. Comput. Conf. (PSCC)*, Dublin, Ireland, 2018, pp. 1-7.
- [9] B. Bruned et S. Denetière, "Simulation temps réel de grands réseaux électriques : solutions de parallélisation," *RTE Tech*, N°16, Janvier 2018, pp. 30-31.
- [10] B. Bruned, I. M. Martins, P. Rault and S. Denetière, "Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Perpignan, France, 2019, pp. 1-6.
- [11] B. Bruned, I. M. Martins, P. Rault and S. Denetière, "Use of Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," *Elect. Power Syst. Res.*, vol. 189, Dec. 2020.
- [12] B. Bruned, S. Denetière, J. Michel, M. Schudel, J. Mahseredjian and N. Bracikowski, "Compensation Method for parallel real-time EMT studies," *Elect. Power Syst. Res.*, vol. 198, Sep. 2021.
- [13] B. Bruned, J. Mahseredjian, S. Denetière, J. Michel, M. Schudel, N. Bracikowski, "Compensation Method for parallel and iterative real-time simulation of electromagnetic transients," accepted to *IEEE Trans. Power Del.*, 2023.
- [14] B. Bruned, "Compensation Method to speedup offline and real-time Electromagnetic Transients simulation of inverter-based transmission networks," in *Contributions CIGRE Session*, Paris, 2022, pp 1-2.

- [15] B. Bruned, S. Denetière, Y. Vernay, V. Oiring De Castro Cesar and H. Saad, "Optimized real-time setup for Interaction study between VSC-HVDC and SVC on the French Network," in *Proc. Conf. IEEE Indus. Electron. Soc. (IECON)*, Bruxelles, 2022, pp 1-6.
- [16] B. Bruned, J. Mahseredjian, S. Denetière and N. Bracikowski, "Optimized reduced Jacobian formulation for simultaneous solution of control systems in Electromagnetic Transients simulations," to be revised in *IEEE Trans. Power Del.*, 2023.
- [17] B. Bruned, X. M. Viel, "Simulations paramétriques de grands réseaux électriques : solutions de parallélisation," *RTE Tech*, N°22, Juillet 2019, pp. 16-17.
- [18] B. Bruned et S. Denetière, "Méthode de Compensation pour paralléliser la simulation temps réel de réseaux électriques," *RTE Tech*, N°27, Avril 2021, pp. 10-11.
- [19] *Stratégie nationale bas-carbone*, Ministère de la Transition Ecologique et Solidaire, Mar. 2020, pp. 1-192.
- [20] R. Ryndzionek, and Sienkiewicz L., "Evolution of the HVDC Link Connecting Offshore Wind Farms to Onshore Power Systems," *Energies* 13, no. 8: 1914, 2020.
- [21] S. Denetière, H. Saad, Y. Vernay, P. Rault, C. Martin and B. Clerc, "Supporting Energy Transition in Transmission Systems: An Operator's Experience Using Electromagnetic Transient Simulation," *IEEE Power Energy Magazine*, vol. 17, no. 3, pp. 48-60, May-June 2019.
- [22] H. Saad, Y. Vernay, S. Denetière, P. Rault and B. Clerc, "System Dynamic Studies of Power Electronics Devices with Real-Time Simulation - A TSO operational experience," in *Proc. CIGRE Session*, Paris, 2018, pp 1-7.
- [23] Y. Vernay, A. Drouet D'Aubigny, Z. Benalla and S. Denetière, "New HDVC LCC replica platform to improve the study and maintenance of the IFA2000 link," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, June 2017, pp 1-7.
- [24] Y. Vernay and B. Gustavsen, "Application of Frequency-Dependent Network Equivalents for EMTP Simulation of Transformer Inrush Current in Large Networks," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Vancouver, Canada, 2013.
- [25] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052-1061, July 1999.
- [26] J. Morales, J. Mahseredjian, A. Ramirez, K. Sheshyekani and I. Kocar, "A Loewner/MPM—VF Combined Rational Fitting Approach," *IEEE Trans. Power Del.*, vol. 35, no. 2, pp. 802-808, Apr. 2020.
- [27] B. Clerc, C. Martin and S. Denetière, "Implementation of accelerated models for EMT tools," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Cavtat, Croatia, 2015.
- [28] W. F. Tinney, "Compensation methods for network solutions by optimally ordered triangular factorization," *IEEE Trans. Power App. Syst.*, vol. PAS-91, no. 1, pp. 123_127, Jan. 1972.

- [29] L. Chua and Li-Kuan Chen, "Diakoptic and generalized hybrid analysis," in *IEEE Transactions on Circuits and Systems*, vol. 23, no. 12, pp. 694-705, December 1976.
- [30] J. Mahseredjian, S. Lefebvre and D. Mukhedkar, "Power Converter simulation module connected to the EMTP," *IEEE Trans. Power Syst.*, vol. 6, no. 2, pp. 501-510, May 1991.
- [31] J. Mahseredjian, S. Lefebvre and X.D. Do, "A New Method for Time-Domain Modelling of Nonlinear Circuits in Large Linear Networks", in *Proc. Power Syst. Comput. Conf. (PSCC)*, Avignon, France, 1993, pp 916-922.
- [32] J. Mahseredjian , C. Dufour, U. Karaagac and J. Belanger, "Simulation of power system transient using state-space grouping through nodal analysis," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Delft, Netherlands, 2011, pp 1-7.
- [33] C. Dufour, J. Mahseredjian, J. Belanger, "A Combined State-Space Nodal Method for the Simulation of Power System Transients," *IEEE Trans. Power Del.*, vol. 26, no, 2, pp. 928-935, 2011.
- [34] D. P. Koester, "Parallel block-diagonal-bordered sparse linear solvers for power systems applications," Ph.D. dissertation, *Elec. Eng. and Comput. Sc.*, Syracuse Univ., Syracuse, NY, 1995.
- [35] J. Aguilar and E. Gelenbe, "Task Assignment and Transaction Clustering Heuristics for Distributed Systems," *Information Sciences* 199-219, March 1997.
- [36] T. Wong, "Automatic parallel task mapping: Application in the simulation of real-time power systems," PhD Thesis, Polytechnique Montréal, 1999.
- [37] F. Pellegrini, J. Roman, "SCOTCH: A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs," in *Proc. HPCN'96*, Brussels, Belgium, April 15-19, 1996.
- [38] P. Zhang, J. R. Marti, H. W. Dommel, "Network Partitioning for Real-Time Power System Simulation," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Montréal, Canada, June 2005.
- [39] M. Vidyasagar, "Input-output analysis of large-scale interconnected systems: decomposition, well-posedness, and stability," *Lecture notes in control and information sciences*, Berlin, New York: Springer-Verlag, 1981, ISBN: 0387105018 (U.S.), 3540105018 (Germany).
- [40] J. Mahseredjian, L. Dubé, Ming Zhou, S. Denetière, G. Joos, "Simultaneous solution of control system equations in EMTP," *IEEE Trans. Power Syst.*, Vol. 21, No. 1, Feb. 2006, pp. 117 – 124.
- [41] C.F. Mugombozi, "Solution of large dimensional control systems based on feedback loops in the simulation of electrical networks," PhD Thesis, Polytechnique Montréal, 2013.
- [42] C.F. Mugombozi, J. Mahseredjian, O. Saad, "Efficient Computation of FeedBack-Based Control System Equations for Electromagnetic Transients," *IEEE Trans. Power Syst.*, Vol. 30, No. 6, Dec. 2015, pp. 2501 – 2509.

- [43] M. Simpson, V. Srinivasan, and A. Thomo, "Efficient computation of feedback arc set at web-scale," in *Proc. VLDB Endo.*, vol. 10, no. 3, Nov. 2016, pp. 133–144.
- [44] J. Mahseredjian, A. Xémard and B. Khodabakhchian, "Régimes transitoires dans les réseaux électriques," *Tech. Ing. Conver. Energie Elec.*, Nov. 2017, pp 1-18.
- [45] A. Ametani, N. Nagaoka, Y. Baba, T. Ohno and K. Yamabuki. *Power System Transients: Theory and Applications, Second Edition (2nd ed.)*. CRC Press, 2016.
- [46] J. A. Martinez Velasco, *Transient Analysis of Power Systems: Solutions, Techniques, Tools and Applications*. Ed. Wiley-IEEE Press, 2015.
- [47] Q. Piraud, X. M. Viel and J. Michel, "Harmonic studies performed by RTE for wind farm connection," in *Proc. CIGRE Session*, Paris, 2022, pp. 1-9.
- [48] I. S. Duff, A. M. Erisman and J. K. Reid, *Direct Methods for Sparse Matrices Second Edition*. Ed. Oxford University Press, New York, 2017.
- [49] T. A. Davis, "*Direct Methods for Sparse Linear Systems*. Ed. SIAM, Philadelphia, 2006.
- [50] V.Q Do, D. McCallum, P. Giroux and B. De Kelper, "A Backward-Forward Interpolation Technique for a Precise Modelling of Power Electronics in HYPERSIM," *Proc. Int. Conf. Power Syst. Transients (IPST)*, Rio de Janeiro, Brazil, 2001, pp 337-342.
- [51] J. Mahseredjian, U. Karaagac, S. Denetiere, and H. Saad, "Simulation of electromagnetic transients with EMTP-RV," in *Numerical Analysis of Power System Transients and Dynamics*, A. Ametani, Ed. Institution of Engineering and Technology (IET), 2015, pp.103-134.
- [52] C. Dufour, S. Cense, V. Jalili-Marandi and J. Bélanger, "Review of state-of-the-art solver solutions for HIL simulation of power systems, power electronic and motor drives," in *Proc. Europ. Conf. Power Elect. Appli. (EPE)*, Lille, France, 2013, pp. 1-12.
- [53] F. Guay, P.-A. Chiasson, N. Verville, S. Tremblay and P. Askvid, "New Hydro-Québec Real-Time Simulation Interface for HVDC Commissioning Studies," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, 2017, pp. 1-8.
- [54] J. Hernandez, E. Kilander, S. Auddy, K. A. Mitsch and M. Meisingset, "Virtual MACH – A powerful simulation platform for HVDC and FACTS in present and future grids," in *Proc. CIGRE Session*, Paris, 2022, pp. 1-13.
- [55] C. Martin and Y. Fillion, "Automation of model exchange between planning and EMT tools," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, 2017, pp. 1-7.
- [56] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions," *IEEE Trans. Power Syst.*, vol. 16, pp. 97-104, Feb. 2001.

- [57] J. Morales, J. Mahseredjian, K. Sheshyekani, A. Ramirez, E. Medina and I. Kocar, "Pole-Selective Residue Perturbation Technique for Passivity Enforcement of FDNEs," in *IEEE Trans. Power Del.*, vol. 33, no. 6, pp. 2746-2754, Dec. 2018.
- [58] J. Mahseredjian, *State-space equations*, EMTP Help.
- [59] Data sheet of Opal-RT real-time simulators OPxxxx [online]. Available: <https://www.opal-rt.com/simulator-platform-op5031/>
- [60] O. Ramos-Leanos, J. L. Naredo, J. Mahseredjian, C. Dufour, J. A. Gutierrez-Robles and I. Kocar, "A Wideband Line/Cable Model for Real-Time Simulations of Power System Transients," in *IEEE Trans. Power Del.*, vol. 27, no. 4, pp. 2211-2218, Oct. 2012.
- [61] Piedmont – Savoy HVDC Link [online]. Available: <http://www.think-grid.org/piedmont-savoy-hvdc-link>.
- [62] S. Y. R. Hui, K. K. Fung and C. Christopoulos, "Decoupled simulation of DC-linked power electronic systems using transmission-line links," *IEEE Trans. Power Electron.*, vol. 9, no. 1, pp. 85-91, Jan. 1994.
- [63] A. Abusalah, O. Saad, J. Mahseredjian, U. Karaagac and I. Kocar, "Accelerated Sparse Matrix-Based Computation of Electromagnetic Transients," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 13-21, 2020.
- [64] H. Hooshyar, L. Vanfretti and C. Dufour, "Delay-free parallelization for real-time simulation of a large active distribution grid model," in Proc. *Conf. IEEE Indus. Electron. Soc. (IECON)*, Florence, Italy, 2016, pp. 6278-6284.
- [65] I. S. Duff and J. K. Reid, "Algorithm 529: Permutations to block triangular form [F1]," *ACM Trans. Math. Softw.*, vol. 4, no. 2, pp. 189–192, Jun. 1978.
- [66] T. A. Davis and E. P. Natarajan, "Algorithm 907: KLU, A Direct Sparse Solver for Circuit Simulation Problems," *ACM Trans. Math. Softw.*, vol. 37, no. 3, pp. 1–17, Sep 2010.
- [67] O. Alsac, B. Stott and W. F. Tinney, "Sparsity-Oriented Compensation Methods for Modified Network Solutions," *IEEE Trans. Power App. Syst.*, Vol. PAS-102, no. 5, pp. 1050-1060, May 1983.
- [68] J. Mahseredjian, "Compensation, Diakoptics, MATE and Bordered-Block-Diagonal methods for network solution parallelization," Polytechnique Montreal, March 18, 2019.
- [69] F. A. Moreira and J. R. Martí, "Latency techniques for time-domain power system transients' simulation," in *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 246-253, Feb. 2005.
- [70] M. A. Tomim, J. R. Martí, T. De Rybel, L. Wang and M. Yao, "MATE network tearing techniques for multiprocessor solution of large power system networks," in Proc. *IEEE PES General Meeting*, Providence, RI, 2010, pp. 1-6.

- [71] T. Wu, "Solution of large-scale networks by tearing," *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 12, pp. 706–713, Dec. 1976.
- [72] F. M. Uriarte, "On Kron's diakoptics," *Elect. Power Syst. Res.*, vol. 88, pp. 146-150, Jul. 2012.
- [73] A. Sangiovanni-Vincentelli, L.-K. Chen and L.O. Chua, "Node-tearing Nodal Analysis," Electronics Research Laboratory, University of California, Berkeley, Memo. No. ERL-M582, Sept. 1976.
- [74] P. W. Aitchison and A. Klos, "Network tearing - an alternative method," *Inter. Journal Elect. Power & Energy Syst.*, vol. 13, pp. 308-320, Dec 1991.
- [75] S. Fan, H. Ding, A. Kariyawasam, and A. M. Gole, "Parallel electromagnetic transients simulation with shared memory architecture computers," *IEEE Trans. Power Del.*, vol. 33, no. 1, pp. 239_247, Feb. 2018.
- [76] G. Meurant, "Computer Solution of Large Linear Systems," *Studies in Mathematics and its Applications 28*. North-Holland, Amsterdam, 1999.
- [77] Y. Saad. *Iterative Methods for Sparse Linear Systems*, Second edition. SIAM, Philadelphia, PA, 2003.
- [78] A. Quarteroni and A. Valli, "Domain Decomposition Methods for Partial Differential Equations," *Oxford University Press*, New York, 1999.
- [79] T. F. Chan and T. P. Mathew. *Domain decomposition algorithms*. Cambridge University Press, 1994.
- [80] A. Sangiovanni-Vincentelli, Li-Kuan Chen and L. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks," in *IEEE Trans. Circuits Syst.*, vol. 24, no. 12, pp. 709-717, Dec. 1977.
- [81] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *J. Parallel Distrib. Comput.*, vol. 48, no. 1, pp. 96–129, 1998.
- [82] G. Karypis, METIS—A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, 2011. [Online]. Available: <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>
- [83] F. Pellegrini, "SCOTCH and LIBSCOTCH 6.0 User's Guide," LaBRL, University of Bordeaux, Sep. 2014.
- [84] F. M. Uriarte, R. E. Hebner, and A. L. Gattozzi, "Accelerating the simulation of shipboard power systems," in *Proc. Grand Challenges Modeling Simulation Conf.*, The Hague, Netherlands, Jun. 2011, pp. 22_29.

- [85] R. Salcedo et al., "Banshee distribution network benchmark and prototyping platform for hardware-in-the-loop integration of microgrid and device controllers," in *The Journal of Engineering*, vol. 2019, no. 8, pp. 5365-5373, Aug. 2019.
- [86] T. Kauffmann; U. Karaagac; I. Kocar; S. Jensen; J. Mahseredjian and E. Farantatos, "An Accurate Type III Wind Turbine Generator Short Circuit Model for Protection Applications," in *IEEE Trans. Power Del.*, vol. 32, no. 6, pp. 2370-2379, Dec. 2017.
- [87] S. Kovacevic, D. Jovcic, O. Despouys and P. Rault, "Comparative stability analysis of LCC, MMC and dual-infeed HVDC operating with weak AC network," in *Proc. 2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada, 2020, pp. 1-5.
- [88] S. Denetière, H. Saad, B. Clerc and J. Mahseredjian, "Setup and performances of the real-time simulation platform connected to the INELFE control system," *Elec. Power Syst. Res.*, vol. 138, pp. 180-187, Sep. 2016.
- [89] H. Saad, S. Denetière and B. Clerc, "Interactions investigations between power electronics devices embedded in HVAC network," in *Proc. IET Int. Conf. AC and DC Power Trans. (ACDC)*, Manchester, UK, 2017, pp. 1-7.
- [90] H. Saad, P. Rault, S. Denetière, M. Schudel, C. Wikstrom and K. Sharifabadi, "HIL Simulation to Assess Interaction Risks of HVDC Systems for Upcoming Grid Development," in *Proc. Conf. IEEE Indus. Electron. Soc. (IECON)*, Singapore, 2020, pp. 5041-5048.
- [91] A. Persico, M. Spiranelli, M. Hyttinen and D. Barron, "IFA2 - HVDC goes compact with Hybrid PASS420 modules and GIB connections," in *Proc. AEIT HVDC Int. Conf. (AEIT HVDC)*, Florence, Italy, 2019, pp. 1-6.
- [92] Y. Vernay, C. Martin, D. Petesch and S. Denetière, "Hardware in the loop simulation to test SVC performances on the French Grid," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Caviat, Croatia, 2015, pp. 1-8.
- [93] W. Li and J. Bélanger, "An Equivalent Circuit Method for Modelling and Simulation of Modular Multilevel Converters in Real-Time HIL Test Bench," *IEEE Trans. Power Del.*, vol. 31, no. 5, pp. 2401-2409, Oct. 2016.
- [94] A. Petit, J. Michel, H. Saad, G. Torresan, C. Faure-Llorens and Y. Vernay "System studies on the French network including HVDC stations and using real-time simulation," in *Proc. CIGRE Session*, Paris, 2020, pp. 1-16.
- [95] Rikido Yonezawa and Taku Noda, "A Study of Solution Process Parallelization for an EMT Analysis Program Using OpenMP," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, 2017, pp. 1-7.
- [96] Andreas Ernst, Houyuan Jiang and Mohan Krishnamoorthy, *A new Lagrangian Heuristic for the Task Allocation Problem*, Industrial Mathematics, pp. 137-158, 2006.

- [97] R. Blikberg and T. Sorevik, "Load balancing and OpenMP implementation of nested parallelism," *Parallel Comp.*, vol. 31, pp. 984-998, 2005.
- [98] A. C. F. Alvim and C. C. Ribeiro, *A hybrid improvement heuristic for the bin packing problem and its application to the problem of task scheduling*, Madero, Mexico 2007.
- [99] P. Habermehl et D. Kesner, *Algorithmes de recherche*, Cours Programmation Logique et Intelligence Artificielle, IUP 3^{ème} année, Jussieu 2003/2004.
- [100] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders and C. Schulz, "Recent Advances in Graph Partitioning," *Algorithm Engineering*, Springer, pp. 117-158, 2016.
- [101] R. D. Williams, "Performance of Dynamic Load Balancing Algorithms for Unstructured Mesh Calculations," *Concurrency: Practice and experience*, vol. 3, no. 5, pp. 457-481, Oct. 1991.
- [102] B. Hendrickson and R. Leland, "An improved spectral graph partitioning algorithm for mapping parallel computations," *SIAM J. Sci. Comput.*, vol. 16, no. 2, pp 452-469, Mar. 1995.
- [103] B. Parlett and D. Scott, "The Lanczos algorithm with selective orthogonalization," *Math. Comp.*, vol. 33, no. 145, pp. 217-238, Jan. 1979.
- [104] P. Le-Huy, M. Woodacre, S. Guérette and E. Lemieux, "Massively Parallel Real-Time Simulation of Very-Large-Scale Power Systems," in *Proc. Int. Conf. Power Syst. Transients (IPST)*, Seoul, Republic of Korea, 2017, pp. 1-7.
- [105] P. Sanders and C. Schulz, "Engineering Multilevel Graph Partitioning Algorithms," in *Proc. Europ. Symp. Algorith. (ESA)*, vol. 6942 of LNCS, pp. 469-480, Springer, 2011.
- [106] P. Sanders and C. Schulz, "Distributed Evolutionary Graph Partitioning," in *Proc. Workshop Algorith. Engin. Experi. (ALENEX)*, pp. 16-29, 2012.
- [107] B. Hendrickson and R. Leland, "The Chaco user's guide version 2.0," Tech. Rep. SAND95-2344, Sandia National Laboratories, Albuquerque, NM, July, 1995.
- [108] P. Sanders and C. Schultz, "KaHIP v2.1 Karlsruhe High Quality Partitioning User Guide," 2013.
- [109] SGI® Altix® UV 100 System User's Guide.
- [110] R. Fourer, D. Gay and B. Kernighan, "AMPL—A Modelling Language for Mathematical Programming," Thomson Brooks/Cole, Pacific Grove, CA, 2003.
- [111] FICO™ Xpress Optimization Suite (2017) Xpress-Optimizer Reference manual Release 31.10.
- [112] "Guide for the development of models for HVDC converters in a HVDC grid," CIGRE Brochure 604, Working Group B4.57, Dec. 2014.

- [113] ABB catalog “XLPE submarine cable systems (attached to XLPE land cable systems User’s Guide)” rev. 5.
- [114] S. Denetière, H. Saad, “EMT simulation of the CIGRE B4 DC Grid test system,” in *Proc. CIGRE Canada Conf.*, Toronto, Canada, Sep. 2014.
- [115] H. Saad, S. Denetière and J. Mahseredjian, “On modelling of MMC in EMT-type program,” in *Proc. IEEE Workshop Control Model. Power Electron. (COMPEL)*, Trondheim, Norway, June 2016.
- [116] Best Paths project online. <http://www.bestpaths-project.eu/>
- [117] J. Häfner and B. Jacobson, “Proactive Hybrid HVDC Breakers - A key innovation for reliable HVDC grids,” in *Proc. CIGRE Bologna Conf.*, Paper 0264, 2011, pp. 1-8.
- [118] N. Johannesson, S. Norrga and C. Wikström, “Selective Wave-Front Based Protection Algorithm for MTDC Systems,” in *Proc. Int. Conf. Dev. Power Syst. Protec. (DPSP)*, Edinburgh, UK, 2016, pp. 1-6.
- [119] P. Rault, M. Yazdani, S. Denetière, C. Wikstrom and H. Saad, “Industrial Application of DCCB in HVDC grids – Modelings and Testing with Physical Controls,” *Elect. Power Syst. Res.*, vol. 189, Dec. 2020.
- [120] J. J. DiSteffano, A. R. Stubberud and I. J. Williams, *Feedback and Control Systems*. Schaum’s outline series, 2nd Edition, McGraw-Hill, 1990.
- [121] H. Saad, P. Rault and S. Denetière, “Investigation on parallel operation of two MMC-HVDC links in grid forming connected to an existing network,” in *Proc. European Conf. Power Electron. Applications (EPE)*, 2020, pp. 1-10.
- [122] B. Khodabakhchian, J. Mahseredjian, M.-R. Sehati and M. Mir-Hosseini, “Potential Risk of Failures in Switching EHV Shunt Reactors in Some One-and-a-half Breaker Scheme Substations,” in *Proc. Int. Conf. Power Syst. Transients (IPST)*, New Orleans, USA, 2003, pp. 1-6.
- [123] R. Kuffel, J. E. Giesbrecht, T. Maguire and R. Wierckx and P. G. McLaren, “RTDS-a fully digital power system simulator operating in real time,” in *Proc. Int. Conf. Energy Manag. Power Del. (EMPD)*, Singapore, 1995, pp 498-503.
- [124] S. Baase and A. Van Gelder, *Computer Algorithms. Introduction to Design and Analysis*. Addison Wesley, 1999.
- [125] A. Haddadi and J. Mahseredjian, “Power system test cases for EMT-type simulation studies,” CIGRE, Paris, France, Tech. Rep. CIGRE WG C4.503, 2018, pp. 1-142.
- [126] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, Jun. 1972.

- [127] M. Karimi-Ghartemani and M. R. Iravani, "A new phase-locked loop (PLL) system," in *Proc. IEEE 44th Midwest Symp. Circuits Syst.*, Aug. 2001, vol. 1, pp. 421–424.
- [128] M. Kizilcay, G. Ban, L. Prikler and P. Handl, "Interaction of the secondary arc with the transmission system during single-phase autoreclosure," in *Proc. IEEE Power Tech Conf.*, Bologna, Italy, 2003, pp. 1-7.
- [129] P. Eades, X. Lin, and W. F. Smyth, "A fast and effective heuristic for the feedback arc set problem," *Inf. Process. Lett.*, 1993.
- [130] S. Chanas and P. Kobyłański, "A new heuristic algorithm solving the linear ordering problem," *Comput. Optim. Applic.*, vol. 6, pp. 191–205, Sep. 1996.
- [131] R. H Lasseter and J. Zhou, "TACS Enhancements for the Electromagnetic Transient Program," *IEEE Trans. Power Syst.*, vol. 9, no. 2, May 1994.
- [132] U. Karaagac, J. Mahseredjian, H. Gras, H. Saad, J. Peralta, and L. D. Bellomo, "Doubly-fed induction generator based wind park models in EMTP," *Polytechnique Montréal*, Internal report, March 2017.
- [133] A. M. Cassie, "Théorie Nouvelle des Arcs de Rupture et de la Rigidité des Circuits," *CIGRE*, Vol. 102, 1939, pp. 588-608.

Annexes

A. Résolution d'un PLL

La Figure A- 1 ci-dessous montre la représentation en schémas-blocs des boucles de contrôle du PLL de [127].

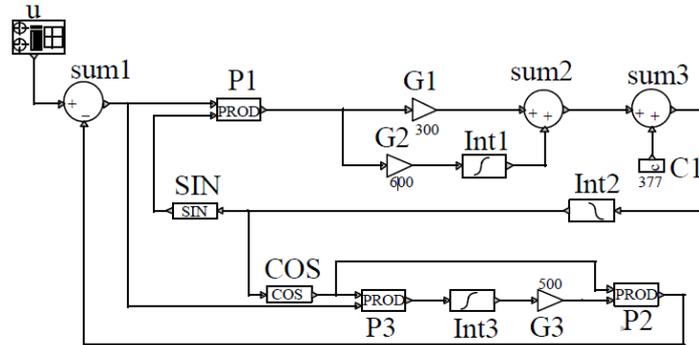


Figure A- 1 : Aperçu de la modélisation d'un PLL via des schémas-blocs.

Via un DFS, on obtient l'ordre post-visite inversé qui permet d'avoir la séquence Seq d'exécution des fonctions de chaque bloc. La Figure A- 2 montre la numérotation des fonctions f_i bloc dans cet ordre.

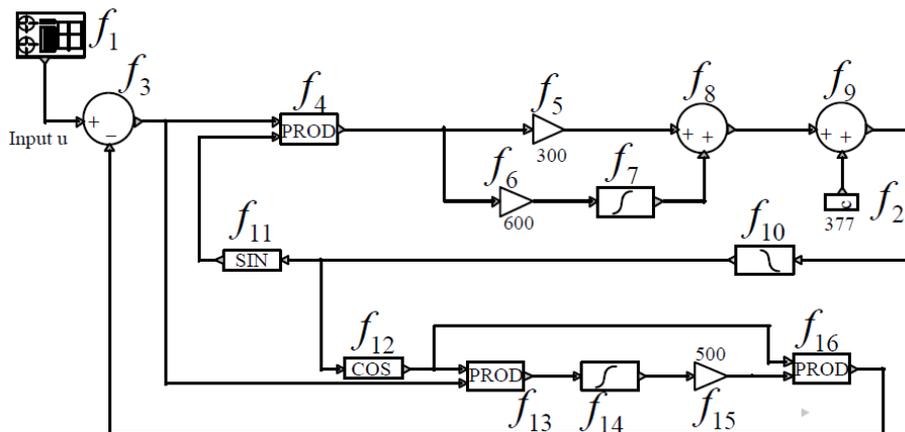


Figure A- 2 : Ordonnement des fonctions des schémas-blocs pour un DLL suivant l'ordre post-visite du DFS.

On associe au PLL le graphe de contrôle suivant (Figure A- 3). Les variables x_i à calculer labélisent les sommets et les fonctions bloc f_i les arêtes.

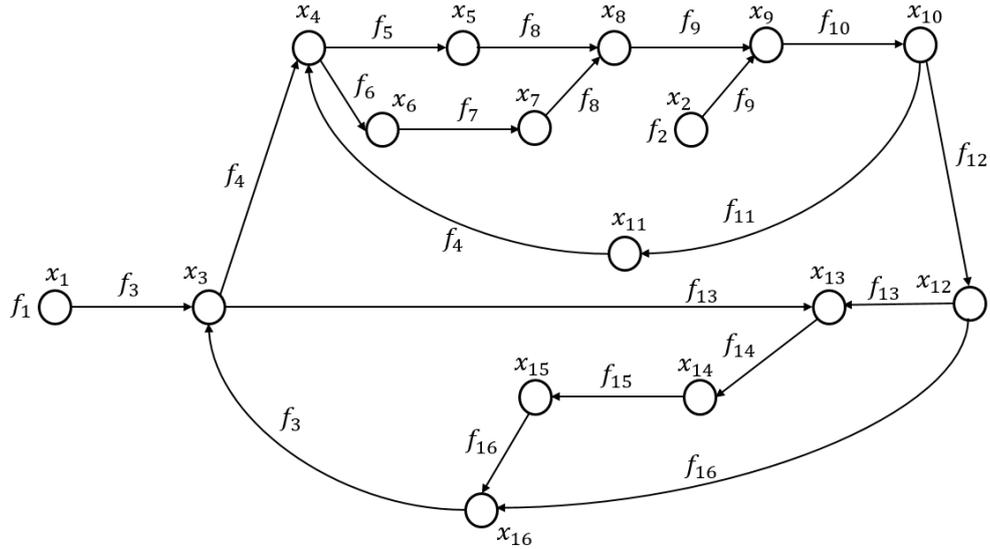


Figure A-3 : Graphe de contrôle associé au PLL.

A.1 Formulation de la méthode Jacobienne

Si la plupart des f_i sont des équations algébriques, cela n'est pas le cas pour f_7, f_{10} et f_{14} qui sont des intégrateurs avec pour équation $y = \int x$. Comme une capacité, c'est équivalent à $\frac{dy}{dt} = x$.

De même que pour la résolution des équations de réseaux de 3.2, on intègre cette équation différentielle sur la durée d'un pas de temps de simulation Δt en utilisant la méthode trapézoïdale. On se ramène à une formulation algébrique discrétisée. Pour l'instant t ,

$$\int_{t-\Delta t}^t \frac{dy(t)}{dt} dt = \int_{t-\Delta t}^t x(t) dt \quad (A-1)$$

$$y(t) - y(t - \Delta t) = \frac{\Delta t}{2} (x(t) + x(t - \Delta t)) \quad (A-2)$$

On définit un terme d'historique $hist$ comme pour la partie réseau :

$$y = \frac{\Delta t}{2} x + hist \quad (A-3)$$

$$hist(t) = \frac{\Delta t}{2} x(t - \Delta t) + y(t - \Delta t)$$

Entre deux itérations de Newton, $hist$ est constant. Il est recalculé à chaque pas de temps.

Ecrivons les équations de chaque bloc à partir de l'expression des f_i (dans l'ordre de Seq). On introduit k_5, k_6 les gains respectifs de f_5 et f_6 . On associe également des valeurs historiques issues de la discrétisation des intégrateurs $hist_7, hist_{10}, hist_{14}$.

$$\begin{aligned} x_1 - u &= 0 \\ x_2 - c &= 0 \\ -x_1 + x_3 + x_{16} &= 0 \\ x_3 x_{11} - x_4 &= 0 \\ k_5 x_4 - x_5 &= 0 \\ k_6 x_4 - x_6 &= 0 \\ -(\Delta t/2)x_6 + x_7 - hist_7 &= 0 \\ -x_5 - x_7 + x_8 &= 0 \\ -x_2 - x_8 + x_9 &= 0 \\ -(\Delta t/2)x_9 + x_{10} - hist_{10} &= 0 \\ \sin x_{10} - x_{11} &= 0 \\ \cos x_{10} - x_{12} &= 0 \end{aligned} \quad (A-4)$$

$$\begin{aligned}
x_3 x_{12} - x_{13} &= 0 \\
-(\Delta t/2)x_{13} + x_{14} - hist_{14} &= 0 \\
k_{15} x_{14} - x_{15} &= 0 \\
x_{12} x_{15} - x_{16} &= 0
\end{aligned}$$

On peut en déduire $F(x)$:

$$F(x) = \begin{pmatrix} x_1 - u \\ x_2 - c \\ -x_1 + x_3 + x_{16} \\ -x_3 x_{11} + x_4 \\ -k_5 x_4 + x_5 \\ -k_6 x_4 + x_6 \\ -(\Delta t/2)x_6 + x_7 - hist_7 \\ -x_5 - x_7 + x_8 \\ -x_2 - x_8 + x_9 \\ -(\Delta t/2)x_9 + x_{10} - hist_{10} \\ -\sin x_{10} + x_{11} \\ -\cos x_{10} + x_{12} \\ -x_3 x_{12} + x_{13} \\ -(\Delta t/2)x_{13} + x_{14} - hist_{14} \\ -k_{15} x_{14} + x_{15} \\ -x_{12} x_{15} + x_{16} \end{pmatrix} \quad (A-5)$$

A partir de ces équations, on calcule la matrice Jacobienne J de F en x

$$J(x) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -x_{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -x_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_6 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\cos x_{10} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sin x_{10} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -x_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{15} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x_{15} & 0 & 0 & -x_{12} & 1 & 0 \end{pmatrix} \quad (A-6)$$

On en déduit $J^{(k)}$ à l'itération k ,

$$J^{(k)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -x_{11}^{(k)} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -x_3^{(k)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_6 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\cos x_{10}^{(k)} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sin x_{10}^{(k)} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -x_{12}^{(k)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x_3^{(k)} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\Delta t/2) & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{15} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x_{15}^{(k)} & 0 & 0 & -x_{12}^{(k)} & 1 & 0 \end{pmatrix} \quad (A-7)$$

On calcule l'ordonnée à l'origine $B^{(k)}$.

$$B^{(k)} = J^{(k)} x^{(k)} - F(x^{(k)}) \quad (A-8)$$

$$\mathbf{B}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ -x_1^{(k)} + x_3^{(k)} + x_{16}^{(k)} \\ -2x_3^{(k)} x_{11}^{(k)} + x_4^{(k)} \\ -k_5 x_4^{(k)} + x_5^{(k)} \\ -k_6 x_4^{(k)} + x_6^{(k)} \\ -(\Delta t/2)x_6^{(k)} + x_7^{(k)} \\ -x_5^{(k)} - x_7^{(k)} + x_8^{(k)} \\ -x_2^{(k)} - x_8^{(k)} + x_9^{(k)} \\ (\Delta t/2)x_9^{(k)} + x_{10}^{(k)} \\ -x_{10}^{(k)} \cos x_{10}^{(k)} + x_{11}^{(k)} \\ x_{10}^{(k)} \sin x_{10}^{(k)} + x_{12}^{(k)} \\ -2x_3^{(k)} x_{12}^{(k)} + x_{13}^{(k)} \\ -(\Delta t/2)x_{13}^{(k)} + x_{14}^{(k)} \\ -k_{15} x_{14}^{(k)} + x_{15}^{(k)} \\ -2x_{12}^{(k)} x_{15}^{(k)} + x_{16}^{(k)} \end{pmatrix} - \begin{pmatrix} x_1^{(k)} - u \\ x_2^{(k)} - c \\ -x_1^{(k)} + x_3^{(k)} + x_{16}^{(k)} \\ -x_3^{(k)} x_{11}^{(k)} + x_4^{(k)} \\ -k_5 x_4^{(k)} + x_5^{(k)} \\ -k_6 x_4^{(k)} + x_6^{(k)} \\ -(\Delta t/2)x_6^{(k)} + x_7^{(k)} - hist_7 \\ -x_5^{(k)} - x_7^{(k)} + x_8^{(k)} \\ -x_2^{(k)} - x_8^{(k)} + x_9^{(k)} \\ -(\Delta t/2)x_9^{(k)} + x_{10}^{(k)} - hist_{10} \\ -\sin x_{10}^{(k)} + x_{11}^{(k)} \\ -\cos x_{10}^{(k)} + x_{12}^{(k)} \\ -x_3^{(k)} x_{12}^{(k)} + x_{13}^{(k)} \\ -(\Delta t/2)x_{13}^{(k)} + x_{14}^{(k)} - hist_{14} \\ -k_{15} x_{14}^{(k)} + x_{15}^{(k)} \\ -x_{12}^{(k)} x_{15}^{(k)} + x_{16}^{(k)} \end{pmatrix} - \begin{pmatrix} u \\ c \\ 0 \\ -x_3^{(k)} x_{11}^{(k)} \\ 0 \\ 0 \\ hist_7 \\ 0 \\ 0 \\ hist_{10} \\ \sin x_{10}^{(k)} - x_{10}^{(k)} \cos x_{10}^{(k)} \\ \cos x_{10}^{(k)} + x_{10}^{(k)} \sin x_{10}^{(k)} \\ -x_3^{(k)} x_{12}^{(k)} \\ hist_{14} \\ 0 \\ -x_{12}^{(k)} x_{15}^{(k)} \end{pmatrix} \quad (\text{A-9})$$

Avec $\mathbf{B}^{(k)}$ et $\mathbf{J}^{(k)}$, on résout le système linéaire ci-dessous pour passer à l'itération suivante.

$$\mathbf{J}^{(k)} \mathbf{x}^{(k+1)} = \mathbf{B}^{(k)} \quad (\text{A-10})$$

Une fois la convergence de la méthode de Newton obtenue et avant de passer au pas de temps suivant, on met à jour les historiques pour le prochain pas de temps $t + \Delta t$:

$$\begin{aligned}
hist_7(t + \Delta t) &= (\Delta t/2)x_6(t) + x_7(t) \\
hist_{10}(t + \Delta t) &= (\Delta t/2)x_9(t) + x_{10}(t) \\
hist_{14}(t + \Delta t) &= (\Delta t/2)x_{13}(t) + x_{14}(t)
\end{aligned} \quad (\text{A-11})$$

A.2 Formulation de la Jacobienne réduite

A partir du graphe de contrôle du PLL, on peut le décomposer en trois composantes fortement connexes (Figure A- 4). On s'intéresse à la résolution de la 2^{ème} composante qui comporte des cycles. En rouge, ont été détectés les back arcs via le DFS, qui une fois enlevés, rendent le graphe acyclique.

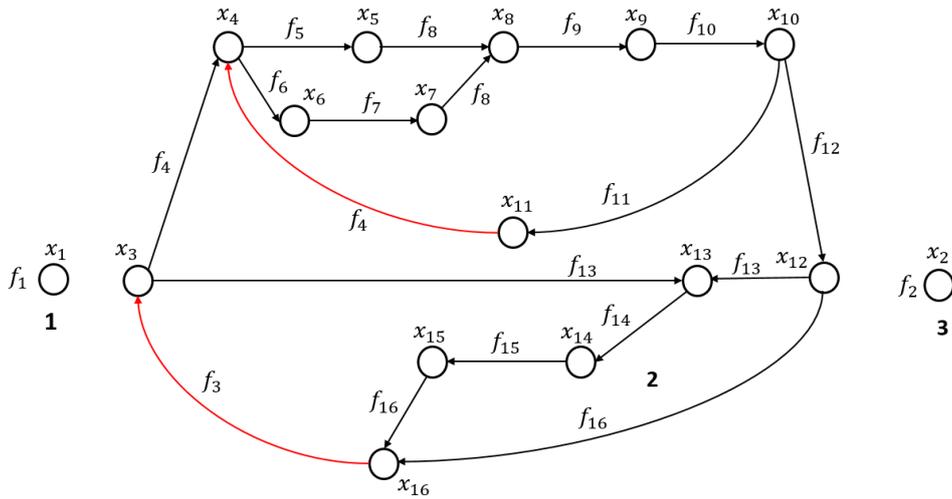


Figure A- 4 : Décomposition du graphe du PLL en composantes fortement connexes et détection des back arcs en rouge.

On en déduit φ ainsi que le vecteur des variables de coupe/Feedback e ,

$$\varphi = \begin{pmatrix} f_{11} \\ f_{16} \end{pmatrix}, e = \begin{pmatrix} x_{11} \\ x_{16} \end{pmatrix} \quad (\text{A-12})$$

Pour le calcul des séquences de boucle, on labélise dans la Figure A- 5 les sommets destination des arcs retirés comme sources (en rouge) et les sommets origine comme des puits (en bleu) sur le graphe acyclique.

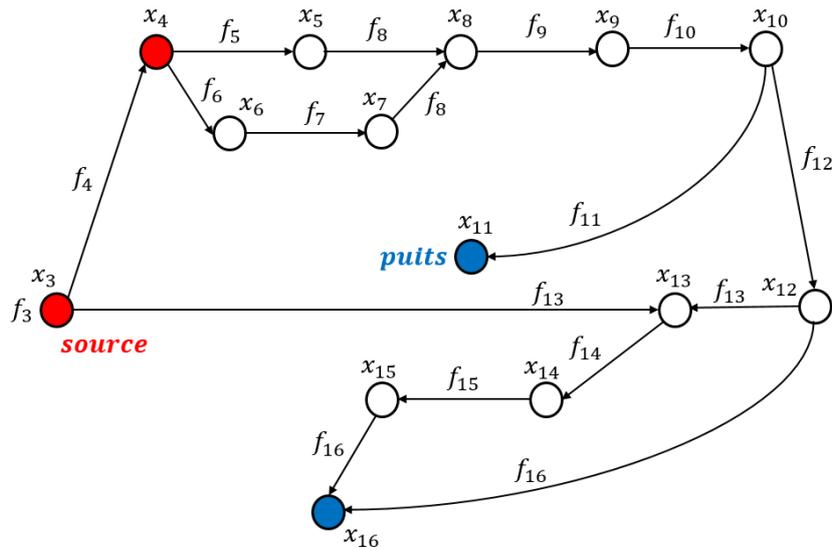


Figure A- 5 : Graphe acyclique du PLL avec les labels des sommets sources et puits.

On en déduit l'ordre topologique du graphe sans cycle ainsi que la séquence **Seq** des fonctions des blocs pour calculer l'ensemble des variables à partir des variables de coupe e .

$$\begin{aligned} \text{Ordre Topologique} &= (x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}) \\ \text{Seq} &= (f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}) \end{aligned} \quad (\text{A-13})$$

La matrice R d'accessibilité du graphe est définie comme suit en reprenant l'ordre topologique pour la numérotation des sommets.

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A-14})$$

Pour la coupe/puits x_{11} , on combine les séquences provenant de l'ensemble des sources (x_3, x_4) pour calculer la séquence de boucle $\mathbf{S}_{x_{11}}$ et l'ordre des fonctions bloc associées $\mathbf{O}_{x_{11}}$.

$$\tilde{\mathbf{S}}_{x_3 x_{11}} = \text{Ligne } \mathbf{R}(1) \otimes \text{Colonne } \mathbf{R}(9)^T \quad (\text{A-15})$$

$$\tilde{\mathbf{S}}_{x_3 x_{11}} = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (\text{A-16})$$

$$\tilde{\mathbf{S}}_{x_4 x_{11}} = \text{Ligne } \mathbf{R}(2) \otimes \text{Colonne } \mathbf{R}(9)^T \quad (\text{A-17})$$

$$\tilde{\mathbf{S}}_{x_4 x_{11}} = (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (\text{A-18})$$

$$\begin{aligned} \tilde{\mathbf{S}}_{x_3 x_{11}} &= (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0) \\ \tilde{\mathbf{S}}_{x_4 x_{11}} &= (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0) \end{aligned} \quad (\text{A-19})$$

$$\tilde{\mathbf{S}}_{x_{11}} = \tilde{\mathbf{S}}_{x_3 x_{11}} \oplus \tilde{\mathbf{S}}_{x_4 x_{11}} \quad (\text{A-20})$$

$$\mathbf{S}_{x_{11}} = (x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}) \quad (\text{A-21})$$

$$\mathbf{O}_{x_{11}} = (f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}) \quad (\text{A-22})$$

On fait de même pour la coupe/puits x_{16} sur l'ensemble des sources (x_3, x_4) pour obtenir la séquence de boucle $\mathbf{S}_{x_{16}}$ et l'ordre des fonctions bloc associées $\mathbf{O}_{x_{16}}$.

$$\tilde{\mathbf{S}}_{x_3 x_{16}} = \text{Ligne } \mathbf{R}(1) \otimes \text{Colonne } \mathbf{R}(14)^T \quad (\text{A-23})$$

$$\tilde{\mathbf{S}}_{x_3 x_{16}} = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (\text{A-24})$$

$$\tilde{\mathbf{S}}_{x_4 x_{16}} = \text{Ligne } \mathbf{R}(2) \otimes \text{Colonne } \mathbf{R}(14)^T \quad (\text{A-25})$$

$$\tilde{\mathbf{S}}_{x_4 x_{16}} = (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (\text{A-26})$$

$$\begin{aligned} \tilde{\mathbf{S}}_{x_3 x_{16}} &= (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1) \\ \tilde{\mathbf{S}}_{x_4 x_{16}} &= (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1) \end{aligned} \quad (\text{A-27})$$

$$\begin{aligned} \tilde{\mathbf{S}}_{x_{16}} &= \tilde{\mathbf{S}}_{x_3 x_{16}} \oplus \tilde{\mathbf{S}}_{x_4 x_{16}} \\ \tilde{\mathbf{S}}_{x_{16}} &= (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1) \end{aligned} \quad (\text{A-28})$$

$$\mathbf{S}_{x_{16}} = (x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}) \quad (\text{A-29})$$

$$\mathbf{O}_{x_{16}} = (f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}) \quad (\text{A-30})$$

Ainsi, pour chaque coupe on a la paire variable et la séquence des fonctions des blocs : $\{e_1, (f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11})\}, \{e_2, (f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16})\}$.

Au niveau des équations de boucle,

$$\mathbf{e} = (e_1, e_1) \quad (\text{A-30})$$

$$\begin{aligned} G_1 &= f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3, u_1 = 0, \varphi_1 = f_{11} \\ y_1 &= G_1(\mathbf{e}) = f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \end{aligned} \quad (\text{A-31})$$

$$\begin{aligned} \phi_1(\mathbf{e}) &= e_1 - \varphi_1(u_1, y_1) \\ \phi_1(\mathbf{e}) &= e_1 - f_{11} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \end{aligned}$$

De même, pour l'autre coupe :

$$\begin{aligned} G_2 &= f_{15} \circ f_{14} \circ f_{13} \circ f_{12} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3 \\ u_2 &= 0, \varphi_2 = f_{16} \end{aligned} \quad (\text{A-32})$$

$$\begin{aligned} y_2 &= G_2(\mathbf{e}) = f_{15} \circ f_{14} \circ f_{13} \circ f_{12} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \\ \phi_2(\mathbf{e}) &= e_2 - \varphi_2(u_2, y_2) \\ \phi_2(\mathbf{e}) &= e_2 - f_{16} \circ f_{15} \circ f_{14} \circ f_{13} \circ f_{12} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \end{aligned}$$

$$\boldsymbol{\phi}(\mathbf{e}) = \begin{pmatrix} \phi_1(\mathbf{e}) \\ \phi_2(\mathbf{e}) \end{pmatrix} \quad (\text{A-33})$$

$$\boldsymbol{\phi}(\mathbf{e}) = \begin{pmatrix} e_1 - f_{11} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \\ e_2 - f_{16} \circ f_{15} \circ f_{14} \circ f_{13} \circ f_{12} \circ f_{10} \circ f_9 \circ f_8 \circ f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3(\mathbf{e}) \end{pmatrix} \quad (\text{A-34})$$

Après, il n'y a plus qu'à calculer la matrice Jacobienne 2x2 de $\boldsymbol{\phi}$ via la méthode des perturbations et à appliquer la méthode de Newton. Ainsi, on a bien eu une réduction significative du rang de la Jacobienne de 14x14 à 2x2.

B. Algorithme de Kosaraju pour les composantes fortement connexes

Algorithm 1 Kosaraju (Composantes Fortement Connexes)

```
1: Graphe  $G$ 
2:  $Visites, OrdrePostVisite, SCCs, SCC \leftarrow \{\emptyset\}$ 
3:  $i \leftarrow 0$ 
4:
5: // Premier DFS
6: for  $v \in S(G)$  do
7:   if  $v \notin Visites$  then
8:      $DFS\_1(v)$ 
9:   end if
10: end for
11: for  $v \in V(G)$  do
12:   if  $v \notin Visites$  then
13:      $DFS\_1(v)$ 
14:   end if
15: end for
16:
17: // Deuxieme DFS
18:  $Visites \leftarrow \{\emptyset\}$ 
19: for  $v \in Inverse(OrdrePostVisite)$  do
20:   if  $v \notin Visites$  then
21:      $i \leftarrow i + 1$ 
22:      $SCC \leftarrow SCCs(i)$ 
23:      $DFS\_2(v)$ 
24:   end if
25: end for
26:
27: function  $DFS\_1(v)$ 
28:    $Visites \leftarrow Inserer(v)$ 
29:   for  $succ \in Succ(v)$  do
30:     if  $succ \notin Visites$  then
31:        $DFS\_1(succ)$ 
32:     end if
33:   end for
34:    $OrdrePostVisite \leftarrow Inserer(v)$ 
35: end function
36:
37: function  $DFS\_2(v)$ 
38:    $Visites \leftarrow Inserer(v)$ 
39:    $SCC \leftarrow Inserer(pred)$ 
40:   for  $pred \in Pred(v)$  do
41:     if  $pred \notin Visites$  then
42:        $DFS\_2(pred)$ 
43:     end if
44:   end for
45: end function
```

C. Récapitulatif de la demande de validation des acquis de l'expérience

VAE DOCTORALE	
GENIE ELECTRIQUE	
COMPETENCES ET ACTIVITES DU REFERENTIEL	COMPETENCES MOBILISEES DURANT VOTRE PARCOURS
Conception et élaboration d'une démarche de recherche et développement, d'études et prospective.	<p>Méthodes pour améliorer la précision et la vitesse de calcul des outils de simulation de phénomènes transitoires électromagnétiques sur le réseau électrique.</p> <p>Recherche transverse sur différents domaines :</p> <p>Génie Electrique comprendre les équations physiques de réseaux, les phénomènes physique et leur modélisation.</p> <p>Mathématiques appliquées pour les méthodes de résolution numériques.</p> <p>Informatique : techniques de parallélisation, algorithme de répartition des tâches.</p>
Valorisation et transfert des résultats d'une démarche R&D	<p>Valorisation d'une démarche R&D à travers des publications scientifiques lors de conférence ou de journaux.</p> <p>3 articles de journaux, 5 de conférences.</p> <p>1 article de journaux en révision.</p> <p>Transfert des résultats d'une démarche de R&D dans des outils existants de simulation EMT (de transitoires électromagnétique).</p>
Veille scientifique et technologique à l'échelle internationale	<p>Veille scientifique lors des conférences sur le domaine de la simulation de transitoires électromagnétiques.</p> <p>Travail d'arbitre pour des revues scientifiques (IEEE Transactions on Power Delivery, Electric Power Systems Research).</p> <p>Outstanding Reviewer 2020 pour IEEE Transactions on Power Delivery pour la qualité et la quantité de reviews fournies au journal.</p>

Formation et diffusion de la culture scientifique et technique et formation	Trois grands niveaux de diffusion de sa culture scientifique : <ul style="list-style-type: none"> - Grand public, conférencier Showroom pour présenter à des publics variés les activités de R&D de RTE, - Interne RTE, rédaction d'article vulgarisée dans la revue technique <i>RTE Tech</i>, - Public spécialisé du domaine de recherche, présentations lors de conférences internationales (IPST, PSCC, RT18), organisation de conférence (IPST2017).
Encadrement d'équipes dédiées à des activités de recherche et développement, d'études et prospective.	Encadrement de stagiaires et d'un alternant sur des sujets techniques. Stage M2 recherche qui a mené à deux articles (IPST et EPSR).

D. Annexes « pédagogiques » illustratives et liées à la description et à l'analyse des expériences

▪ ***Publications Scientifiques***

- [1] B. Bruned, J. Mahseredjian, S. Denetière, J. Michel, M. Schudel, N. Bracikowski, "Compensation Method for parallel and iterative real-time simulation of electromagnetic transients," **accepted to IEEE Transactions on Power Delivery**, 2023. [doi:10.1109/TPWRD.2023.3238422](https://doi.org/10.1109/TPWRD.2023.3238422).
- [2] B. Bruned, S. Denetière, Y. Vernay, V. Oiring De Castro Cesar, H. Saad, "Optimized real-time setup for Interaction study between VSC-HVDC and SVC on the French Network," **Proceedings of the IEEE Industrial Electronics Society Conference (IECON)**, Bruxelles, Belgique, 2022. [doi:10.1109/IECON49645.2022.9969119](https://doi.org/10.1109/IECON49645.2022.9969119).
- [3] B. Bruned, "Compensation Method to speedup offline and real-time Electromagnetic Transients simulation of inverter-based transmission networks," **Contributions of CIGRE Session**, Paris, France, 2022.
- [4] B. Bruned, S. Denetière, J. Michel, M. Schudel, J. Mahseredjian, N. Bracikowski, "Compensation Method for parallel real-time EMT studies," **Electric Power Systems Research**, Volume 198, 2021. doi.org/10.1016/j.epsr.2021.107341.
- [5] B. Bruned, I. M. Martins, P. Rault, S. Denetière, "Use of Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," **Electric Power Systems Research**, Volume 189, 2020. [doi:10.1016/j.epsr.2020.106604](https://doi.org/10.1016/j.epsr.2020.106604).

- [6] B. Bruned, I. M. Martins, P. Rault, S. Denetière, "Efficient Task Allocation Algorithm for Parallel Real-Time EMT Simulation," **Proceedings of the International Conference on Power Systems Transients 2019 (IPST 2019)**, Perpignan, France, 2019. https://www.ipstconf.org/papers/Proc_IPST2019/19IPST035.pdf
- [7] S. Denetière, B. Bruned, H. Saad, E. Lémieux, "Task separation for real-time simulation of the CIGRE DC grid benchmark," **Proceedings of the Power Systems Computation Conference 2018 (PSCC 2018)**, Dublin, Ireland, 2018. [doi:10.23919/PSCC.2018.8442556](https://doi.org/10.23919/PSCC.2018.8442556).
- [8] B. Bruned, C. Martin, S. Denetière, Y. Vernay, "Implementation of a unified modelling between EMT tools for Network Studies," **Proceedings of the International Conference on Power Systems Transients 2017 (IPST 2017)**, Seoul, Republic of Korea, 2017. https://www.ipstconf.org/papers/Proc_IPST2017/17IPST128.pdf
- **Rapports Internes**
- [9] B. Bruned, S. Denetière, "Méthode de Compensation pour paralléliser la simulation temps réel de réseaux électriques," **RTE Tech**, N°27, April, 2021.
- [10] B. Bruned, X. M. Viel, "Simulations paramétriques de grands réseaux électriques: solutions de parallélisation," **RTE Tech**, N°22, Juillet, 2019.
- [11] B. Bruned, S. Denetière, "Simulation temps réel de grands réseaux électriques : solutions de parallélisation," **RTE Tech**, N°16, Janvier, 2018.
- **Prépublications Scientifiques**
- [12] B. Bruned, J. Mahseredjian, S. Denetière and N. Bracikowski, "Optimized reduced Jacobian formulation for simultaneous solution of control systems in Electromagnetic Transients simulations," to be revised in **IEEE Transactions on Power Delivery**, 2023.

Titre : Amélioration de la vitesse de calcul et de la précision des outils de simulation des phénomènes transitoires électromagnétiques sur les réseaux électriques

Mots clés : Validation des Acquis de l'Expérience, Réseaux Électriques, Dispositifs d'Électronique de Puissance, Simulation, Transitoires Électromagnétiques, Temps réel, Équations de réseaux, Équivalent fréquentiel de réseau, Parallélisation, Équations de Contrôle, Boucles de rétroaction

Résumé : Ce mémoire de doctorat rédigé dans le cadre de la Validation des Acquis de l'Expérience (VAE) résulte de sept années de travail en tant qu'ingénieur R&D au sein de l'entreprise RTE (Réseau de Transport d'Électricité) sur les outils de simulation de Transitoires ElectroMagnétiques (EMT) des réseaux électriques. Des compétences de chercheur ont été développées autour de la thématique de la simulation temps réel des réseaux électriques incluant de l'électronique de puissance. En effet, au vu des objectifs de la Transition Énergétique, un nombre croissant de dispositifs d'électronique de puissance est installé sur le réseau pour l'intégration des énergies renouvelables. La simulation EMT est au cœur de ces enjeux car elle est utilisée pour valider le bon fonctionnement de ces dispositifs.

L'axe principal de mes recherches est d'accélérer la vitesse de calcul et d'améliorer la précision des simulations EMT d'un réseau électrique. Des équivalents de réseaux fréquents ont été mis en œuvre pour le temps réel afin de réduire le nombre de variables pris en compte par la résolution des équations de réseaux. Les techniques de parallélisation des équations de réseaux ont été étudiées et implémentées en temps réel pour accélérer le temps de simulation. Enfin, le nombre de variables associées aux boucles de rétroaction des équations des contrôles d'un réseau électrique, ont été réduites pour augmenter la vitesse de calcul.

Title : Improvement of speed up computation and accuracy for the simulation of electromagnetic transient phenomena in electrical networks

Keywords : Validation of prior Learning and Experience, Electrical Networks, Power Electronics Equipment, Simulation, Electromagnetic Transients, Real-Time, Network Equations, Network Frequency Equivalent, Parallelization, Control Equations, Feedback loop.

Abstract : This doctoral dissertation, written under the framework of the Validation of prior Learning and Experience (VAE) results from seven years of work as an R&D engineer within RTE (Electricity Transport Network) on tools for simulating ElectroMagnetic Transients (EMT) of electrical networks. Researcher skills have been acquired in the field of real-time simulation of electrical networks including power electronics. Indeed, given the objectives of the Energy Transition, an increasing number of power electronic equipment are installed on the network to integrate renewable energies. EMT simulation is at the heart of these challenges since they are used to validate the proper functioning of these devices.

The main topic of my research is to accelerate the computation speed and to improve the accuracy of EMT simulations of an electrical network. Frequency network equivalents have been implemented in real-time to reduce the number of network equations to solve. Network equation parallelization techniques have been studied and implemented in real-time to speed up simulation time. Finally, the number of variables related to feedback control loop equations of an electrical network has been reduced to increase the computation speed