

Titre: Traitement des discontinuités pour la simulation des réseaux
Title: électriques et des circuits d'électronique de puissance

Auteur: Willy Arnaud Nzale Mimbe
Author:

Date: 2021

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Nzale Mimbe, W. A. (2021). Traitement des discontinuités pour la simulation des réseaux électriques et des circuits d'électronique de puissance [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/9076/>
Citation:

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/9076/>
PolyPublie URL:

Directeurs de recherche: Jean Mahseredjian, & İlhan Kocar
Advisors:

Programme: Génie électrique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Traitement des discontinuités pour la simulation des réseaux électriques et
des circuits d'électronique de puissance**

WILLY ARNAUD NZALE MIMBE

Département de génie électrique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiae Doctor*

Génie électrique

Août 2021

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

**Traitement des discontinuités pour la simulation des réseaux électriques et des
circuits d'électronique de puissance**

présentée par **Willy Arnaud NZALE MIMBE**

en vue de l'obtention du diplôme de *Philosophiae Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Houshang KARIMI, président

Jean MAHSEREDJIAN, membre et directeur de recherche

Ilhan KOCAR, membre et codirecteur de recherche

Keyhan SHESHYEKANI, membre

Jeroen TANT, membre externe

DÉDICACE

À ma fille Sabrina...

REMERCIEMENTS

Mes premiers remerciements sont adressés aux partenaires industriels de la chaire de recherche qui ont soutenu financièrement mes travaux, ainsi qu'au gouvernement du Canada qui a facilité mon installation à Montréal avec ma petite famille.

Je voudrais marquer ma gratitude à certaines personnes pour le rôle qu'elles ont joué dans l'édification de ma personnalité et le soutien à mon parcours doctoral. Je commencerai par mon directeur de recherche Jean Mahseredjian qui m'a marqué par sa passion très visible pour son domaine d'activité, son esprit poussé de rigueur, son enthousiasme, sa polyvalence, sa disponibilité et l'énergie positive qu'il dégage. J'exprime également ma profonde reconnaissance à mon codirecteur Ilhan Koçar dont la simplicité, la flexibilité et l'expertise m'ont marqué depuis le jour de notre toute première rencontre. Je remercie Tarek Ould Bachir, Aboutaleb Haddadi et Haoyan Xue qui ont contribué à l'orientation de mon travail, qui m'ont permis de comprendre les exigences d'un parcours doctoral et qui ont mis leur expertise à ma disposition lorsque j'en avais besoin. Je remercie également Tshibain Tshibungu pour ses compétences en modélisation et simulation des réseaux. Un merci spécial à Sérigne Seye, ancien étudiant à la maîtrise qui a grandement contribué à ma compréhension du domaine de recherche dans lequel je me suis engagé à travers les résultats des travaux qu'il a menés avant moi.

Je remercie mes collègues de bureau Jesus, Anton, Nazack, Ming et les autres qui m'ont aidé à m'intégrer dans le laboratoire de simulation. J'exprime aussi ma gratitude à l'endroit de mes amis Adjiali Boukar, Gilles Either et Reine Vanessa qui n'ont cessé de me rappeler combien ils croient en mes capacités à mener à bien l'ambitieux projet doctoral.

Je suis reconnaissant envers le couple Fogue (Michel et Solange), Magloire Youdom, Benoit Ndzana, Bonaventure Obono, Hubert Feukam, Khaled Arfa, Yannick Foko et Sabrina Mancini qui m'ont apporté du soutien et qui ont directement ou indirectement contribué à bâtir ma personnalité.

Sur un plan plus personnel, je remercie mes parents Charles Mimbe et Kapche, mon épouse Cécile, mes sœurs Christelle et Cannella, ainsi que mon frère Teddy. Je souhaite être une source d'inspiration et de motivation pour chacun d'entre eux dans son domaine respectif.

RÉSUMÉ

Les systèmes électriques connaissent des évolutions perpétuelles dans le but de satisfaire une demande en énergie de plus en plus grandissante, ceci dans des conditions d'opération optimales. La simulation de tels systèmes fait face à de nombreux défis à cause de la complexité qui les caractérise. Cette thèse développe de nouvelles méthodes de simulation dans le domaine du temps des transitoires de réseaux électriques et des circuits d'électronique de puissance. Les méthodes présentées visent un traitement plus efficient des discontinuités. Elles permettent d'améliorer la précision et parfois de réduire les temps de calcul par rapport aux méthodes existantes. La relative simplicité d'implémentation dans des logiciels existants représente également une caractéristique essentielle des méthodes proposées. Ces dernières sont intégrées à des algorithmes qui utilisent l'approche MANA pour la formulation des équations et la méthode trapézoidale pour l'intégration numérique. Les contributions de la thèse peuvent être regroupées en trois grandes parties.

Dans la première partie, deux nouvelles méthodes utilisant un pas d'intégration numérique variable sont proposées. La première permet de simuler les transitoires des grands réseaux. Elle implémente une stratégie de changement du pas d'intégration selon le niveau de précision souhaité. La seconde permet de simuler les circuits d'électronique de puissance. C'est une méthode qui utilise l'interpolation linéaire pour détecter les discontinuités avec précision et un pas d'intégration très petit pour les traverser. Les résultats de simulation ont démontré que, comparées aux méthodes de simulation utilisant un pas d'intégration fixe, ces deux méthodes sont capables d'atteindre des précisions excellentes tout en réduisant de manière significative les temps de calcul.

La deuxième partie de cette thèse s'intéresse à la simulation des circuits d'électronique de puissance, plus particulièrement à la gestion des discontinuités. Il s'agit en effet de la principale source d'erreurs lors de la simulation de tels circuits. La détection précise des discontinuités par interpolation, la suppression des oscillations numériques et la réinitialisation des variables après chaque discontinuité sont les principales précautions à prendre afin de limiter les erreurs qui se créent pendant la simulation. De plus, il a été établi que l'erreur due aux techniques d'interpolation peut, dépendamment des méthodes, varier selon le moment précis où les variables sont interpolées entre deux points de temps consécutifs. À la lumière de ce résultat et d'analyses complémentaires qui ont été menées, trois nouvelles méthodes de simulation sont proposées. Testées sur de réels

circuits et par comparaison aux méthodes existantes, ces méthodes montrent une amélioration significative de la précision aux dépend d'une légère augmentation des temps de calcul.

Finalement, la troisième et dernière partie de cette thèse s'intéresse au sujet de la simulation des circuits contenant des composants non-linéaires. Quelques méthodes existantes sont présentées et leurs limites sont établies. Une nouvelle méthode est proposée. Elle permet à chaque instant de la simulation, d'atteindre le point d'opération de chaque composant non-linéaire en un nombre d'itérations nettement inférieur à celui requis par les méthodes existantes. La nouvelle méthode conduit à une réduction significative des temps de calcul tout en maintenant la précision dans les résultats.

En synthèse, les méthodes et techniques développées dans cette thèse conduisent à des améliorations significatives par rapport aux méthodes existantes. De plus, elles sont développées de manière à pouvoir s'intégrer aux logiciels de simulation existants, moyennant des ajustements conséquents.

ABSTRACT

Power systems and embedded power electronics converters are constantly evolving, aiming to satisfy a continuously increasing demand, in optimal operation conditions. The simulation of such systems faces several challenges due to related levels of complexity. This thesis develops new methods for time-domain simulation of transients in power networks and in power electronic circuits. The presented methods focus on discontinuities treatment. They improve accuracy and sometimes reduce computing times when compared to existing methods. The relative ease of implementation in actual software for the simulation of electromagnetic transients represents a key characteristic of the proposed methods. The contributions of the thesis are organised into three main parts.

In the first part, two variable time-step methods are proposed. The first one is designed for the simulation of transients in power networks. It implements a variable time-step strategy based on expected accuracy level. The second one is dedicated to the simulation of power electronic circuits. Basically, this method uses linear interpolation to accurately locate discontinuities and a much smaller time-step to cross them. Simulation results show that, when compared to fixed time-step methods, the proposed methods can reach excellent accuracy levels while significantly reducing computing times.

The second part of this thesis focuses on the simulation of power electronic circuits, specifically on the treatment of discontinuities. Discontinuities are the main sources of errors in the simulation of power electronic circuits. Accurate tracking of discontinuities through interpolation, suppression of numerical oscillations and re-initialisation of network variables after discontinuities are main features that must be implemented in simulation algorithms to reduce simulation induced errors. Furthermore, it has been established that errors due to interpolation techniques may, depending on the methods, vary with respect to the exact moment at which network variables are interpolated in between consecutive time-points. Regarding this observation and based on complementary analysis, three new simulation methods are proposed. Tested on practical circuits and compared to existing methods, the proposed methods show significant accuracy improvement without significant deterioration of computing times.

Finally, the third and last part of this thesis focuses on the simulation of nonlinear circuits. Some existing methods are presented, and their limits are analyzed. A new method is proposed. At each

simulation instant, it allows to reach the operating point in the nonlinear characteristic of each device with much less iterations (and computing times) than what is required with existing methods.

The methods and techniques developed in this thesis produce significant improvements when compared to existing simulation methods. Furthermore, they can be easily incorporated into actual simulation software.

TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS	IV
RÉSUMÉ.....	V
ABSTRACT	VII
TABLE DES MATIÈRES	IX
LISTE DES TABLEAUX.....	XV
LISTE DES FIGURES	XVI
LISTE DES SIGLES ET ABRÉVIATIONS	XIX
LISTE DES ANNEXES	XXI
CHAPITRE 1 INTRODUCTION.....	1
1.1 Contexte et motivation	1
1.2 Revue de la littérature	4
1.2.1 Simulation des grands réseaux électriques.....	4
1.2.2 Simulation des circuits d'électronique de puissance.....	6
1.3 Objectifs de la thèse	10
1.4 Contributions de la thèse	10
1.5 Organisation de la thèse	11
CHAPITRE 2 APPROCHE GÉNÉRALE DE MODÉLISATION ET DE SIMULATION DES CIRCUITS ÉLECTRIQUES.....	12
2.1 Les méthodes d'intégration numérique et la création des modèles.....	12
2.1.1 Les méthodes d'intégration numérique	12
2.1.1.1 Description générale.....	12

2.1.1.2	Critères d'évaluation des méthodes d'intégration numérique.....	13
2.1.1.3	Quelques méthodes d'intégration numérique.....	14
2.1.2	La création des modèles des composants électriques.....	17
2.1.2.1	Approche globale	17
2.1.2.2	Modèle de l'inductance	17
2.1.2.3	Modèle du condensateur.....	19
2.1.2.4	Modèle de l'inductance non-linéaire.....	19
2.1.2.5	Modèle d'une résistance non-linéaire	21
2.1.2.6	Modèle d'une ligne de transport monophasée.....	22
2.2	La formulation et la résolution des équations	23
2.2.1	La formulation nodale et la formulation MANA	23
2.2.1.1	Description générale.....	23
2.2.1.2	Participation des sources de tension idéales.....	24
2.2.1.3	Participation des sources de courant	25
2.2.1.4	Participation des transformateurs idéaux	25
2.2.1.5	Participation des interrupteurs idéaux	26
2.2.1.6	Exemple démonstratif	27
2.2.2	La résolution des équations	28
2.2.2.1	Approche générale.....	28
2.2.2.2	Apparition et traitement des discontinuités	29
2.2.2.3	Traitement des non-linéarités	30
CHAPITRE 3	SIMULATION À PAS D'INTÉGRATION NUMÉRIQUE VARIABLE	32
3.1	Description générale.....	32
3.1.1	Les avantages de l'utilisation d'un pas d'intégration variable	32

3.1.2	Les défis liés à l'utilisation d'un pas d'intégration numérique variable	33
3.1.3	Mode opératoire d'une simulation à pas variable	35
3.2	Algorithme utilisant un pas d'intégration numérique variable pour la simulation des transitoires des grands réseaux	37
3.2.1	Les fondements de l'algorithme proposé	37
3.2.2	Comportement de l'erreur de troncature	38
3.2.3	Application du comportement de LTE à l'algorithme à pas variable	39
3.2.4	Séquence des évènements de l'algorithme proposé	41
3.2.5	Application à la simulation d'un réseau, analyses et conclusions.....	43
3.3	Algorithme utilisant un pas d'intégration numérique variable pour la simulation des convertisseurs d'électronique de puissance.....	47
3.3.1	Description des interrupteurs électronique.....	47
3.3.2	Détection précise des discontinuités par l'interpolation	49
3.3.3	Les fondements de l'algorithme proposé	51
3.3.4	Séquence des évènements de l'algorithme proposé	52
3.3.5	Application à la simulation d'un convertisseur, analyses et conclusions	53
CHAPITRE 4	ÉTUDE DE LA PRÉCISION DANS LA SIMULATION DES CIRCUITS D'ÉLECTRONIQUE DE PUISSANCE	58
4.1	Les défis liés à la gestion des discontinuités	58
4.1.1	La détermination précise des instants d'apparition des discontinuités et le calcul des variables à ces instants	58
4.1.1.1	Formules d'interpolation pour le calcul des variables aux instants de discontinuités.....	59
4.1.1.2	Approche de détermination de l'instant de discontinuité.....	60
4.1.1.3	Utilisation de l'extrapolation comme technique alternative	64

4.1.2 La suppression des oscillations numériques.....	65
4.1.2.1 Origine des oscillations numériques	65
4.1.2.2 Techniques de suppression des oscillations numériques.....	66
4.1.2.3 Cas particuliers.....	67
4.1.3 La réinitialisation du circuit après une discontinuité	71
4.1.3.1 Réinitialisation des variables.....	71
4.1.3.2 Détection des commutations instantanées	72
4.2 Analyse du comportement de l'erreur selon les techniques d'interpolation	74
4.2.1 Comparaison entre les techniques d'interpolation et les techniques d'extrapolation	74
4.2.2 Impact de l'interpolation linéaire sur la précision des résultats pour différents algorithmes de simulation	77
4.2.2.1 Mise en contexte.....	77
4.2.2.2 Analyse du comportement de l'erreur	79
CHAPITRE 5 MÉTHODES PRÉCISES DE SIMULATION DES CIRCUITS D'ELECTRONIQUE DE PUISSANCE	88
5.1 Fondements des nouvelles méthodes de simulation.....	88
5.1.1 Les caractéristiques d'un algorithme évolué dédié à la simulation des circuits d'électronique de puissance	88
5.1.2 Analyse d'un algorithme combinant l'interpolation et la technique simultaneous switching	90
5.1.2.1 Description de l'algorithme.....	90
5.1.2.2 Les insuffisances de la méthode.....	92
5.2 Nouvelles méthodes	92
5.2.1 Méthode TR_BE_SS_I.....	93
5.2.2 Méthode TR_BE_I_SS_m.....	93

5.2.3	Méthode TR_I_SS_m_BE.....	94
5.3	Simulations et analyses	95
5.3.1	Simulation d'un simple circuit RL	96
5.3.2	Simulation d'un circuit back-boost	100
5.3.3	Simulation d'un circuit STATCOM simple	102
5.3.4	Simulation d'un convertisseur DC-AC-DC	104
CHAPITRE 6 METHODES DE SIMULATION DES CIRCUITS NON-LINÉAIRES		109
6.1	Généralités.....	109
6.1.1	Les composants linéaires et les composants non-linéaires.....	109
6.1.1.1	Description	109
6.1.1.2	Représentation	110
6.1.2	Approche de résolution d'un circuit contenant une non-linéarité	111
6.2	Méthodes existantes pour la simulation des circuits non-linéaires	112
6.2.1	La méthode Newton classique.....	113
6.2.2	La méthode de Katzenelson	116
6.2.3	La méthode de la modification de la tension itérative	119
6.2.4	Un schéma itératif robuste et efficient	120
6.2.5	La méthode arc length	120
6.2.6	Adaptation des méthodes présentées à la formulation MANA pour la résolution des circuits plus complexes	123
6.3	Adaptation de la méthode arc length à la formulation MANA	126
6.3.1	Détermination de l'équivalent Thevenin aux bornes de chaque composant à chaque itération.....	127
6.3.2	Détection et élimination des boucles infinies.....	128
6.4	Simulations et discussions.....	129

6.4.1	Simulation d'un simple circuit à diode	129
6.4.2	Simulation d'un convertisseur DC-AC-DC	131
6.4.3	Simulation d'un convertisseur triphasé	134
CHAPITRE 7	CONCLUSION ET RECOMMANDATIONS	138
7.1	Synthèse de la recherche	138
7.2	Perspectives de la recherche.....	140
LISTE DE RÉFÉRENCES.....		141
ANNEXES		148

LISTE DES TABLEAUX

Tableau 3.1: Paramètres de l'algorithme utilisant un pas d'intégration variable	43
Tableau 3.2: Évaluation des temps de calcul des deux algorithmes testés	46
Tableau 3.3: Conditions de conduction et de blocage de quelques interrupteurs électronique	47
Tableau 3.4: Performances des méthodes testées.....	56
Tableau 6.1: Points de cassure de la caractéristique linéaire par segments d'une diode (1/2)	117
Tableau 6.2: Points de cassure de la caractéristique linéaire par segments d'une diode (2/2)	117
Tableau 6.3: Nombre d'itérations selon la prédition initiale (modèle détaillé de la diode).....	130
Tableau 6.4: Nombre de solutions du système (et nombre moyen de solutions par point de temps) pour chaque méthode selon le pas d'intégration (modèle linéaire par segments)	132
Tableau 6.5: Nombre de solutions du système (et nombre moyen de solutions par point de temps) pour différentes méthodes et selon le pas d'intégration (modèle linéaire par segments)	136
Tableau 6.6: Tableau des écarts types des méthodes testées selon le pas d'intégration.....	137

LISTE DES FIGURES

Figure 1.1: Exemple de circuit équivalent Norton (modèle Norton)	2
Figure 1.2: Exemple de ligne de temps de la simulation (time-mesh).....	3
Figure 2.1: Modèle Norton d'une inductance idéale	18
Figure 2.2: Caractéristique courant-flux d'une inductance non-linéaire	20
Figure 2.3: Circuit équivalent d'une ligne monophasée sans pertes.....	22
Figure 2.4: Schéma d'une source de tension idéale	24
Figure 2.5: Source de courant	25
Figure 2.6: Schéma d'un transformateur idéal.....	25
Figure 2.7: Schéma d'un interrupteur idéal	26
Figure 2.8: Circuit de démonstration de la formulation MANA.....	27
Figure 3.1: Circuit simple RLC série	38
Figure 3.2: LTE sur la tension du condensateur et sur le courant de l'inductance	39
Figure 3.3: Exemple de circuit siège de transitoires électromagnétiques	43
Figure 3.4: Tension du condensateur C1	44
Figure 3.5: Courant dans la phase a du disjoncteur BR1	44
Figure 3.6: Zoom sur la transitoire 27 kHz du courant dans BR1	45
Figure 3.7: Modélisation d'un IGBT	48
Figure 3.8: Modélisation d'une diode.....	49
Figure 3.9: Exemple de time-mesh	50
Figure 3.10: Description de l'algorithme à pas variable proposé.	52
Figure 3.11: Convertisseur AC/DC	53
Figure 3.12: Courant dans l'impédance RL_a pour les méthodes testées	54
Figure 3.13: Tension aux bornes du condensateur C pour les méthodes testées	55

Figure 4.1: Exemple de time-mesh pour illustrer les techniques d'interpolation	59
Figure 4.2: Exemple de signaux issus d'un contrôleur	61
Figure 4.3: Exemple de time-mesh pour illustrer les techniques de suppression des oscillations .	66
Figure 4.4: Circuit RL simple pour illustrer la persistance des oscillations numériques	68
Figure 4.5: Oscillations numériques sur la tension de l'inductance L	70
Figure 4.6: Circuit servant à illustrer les situations de commutations simultanées	72
Figure 4.7: Formes d'onde du courant dans l'inductance L pour les méthodes testées	73
Figure 4.8: Circuit test pour comparer les techniques d'interpolation et d'extrapolation.....	75
Figure 4.9: Erreurs causées par l'interpolation (traits continus) et l'extrapolation (traits discontinus) en fonction du pas d'intégration	76
Figure 4.10: Diagramme illustratif de quelques techniques de gestion des discontinuités.....	79
Figure 4.11: Erreur locale des techniques testées en fonction de alpha	83
Figure 4.12: Circuit RLC simple servant à illustrer le comportement de l'erreur	84
Figure 4.13: Erreur relative RMS sur le courant dans l'inductance L_1 pour les techniques testées	85
Figure 4.14: Circuit simple à thyristors.....	86
Figure 4.15: Erreur dans le courant de l'inductance	87
Figure 5.1: Illustration des méthodes discutées	90
Figure 5.2: Oscillations numériques sur la tension de l'inductance avec les méthodes testées	98
Figure 5.3: Courant dans l'inductance L (allure générale au-dessus et zoom en dessous).....	101
Figure 5.4: Circuit STATCOM simple	103
Figure 5.5: Erreur relative moyenne (RRE) sur la tension de C selon le pas d'intégration.....	103
Figure 5.6: Convertisseur DC-AC-DC	105
Figure 5.7: Signaux de commande provenant du générateur PWM	105
Figure 5.8: Tension aux bornes de la charge R_{load} pour toutes les méthodes avec $\Delta t = 10\text{ns}$	106

Figure 5.9: Nombre de fois que le système est solutionné pour chaque méthode	107
Figure 6.1: Représentation détaillée et linéaire par segments d'un composant non-linéaire.....	110
Figure 6.2: Circuit générique comportant un composant non-linéaire.....	111
Figure 6.3: Illustration du processus itératif de la méthode Newton	114
Figure 6.4: Caractéristiques de la diode (en bleu) et de l'équivalent Thévenin du reste du circuit (en noir), à un point de temps t	115
Figure 6.5: Caractéristique linéaire par segments	127
Figure 6.6: Points explorés par les méthodes MTI et AL lors de la recherche de la solution.....	130
Figure 6.7: Convertisseur DC-AC-DC avec des interrupteurs non-linéaires.....	131
Figure 6.8: Tension aux bornes de la charge pour $\Delta t = 40\text{ ns}$ avec les méthodes testées	133
Figure 6.9: Schéma du circuit convertisseur triphasé	134
Figure 6.10: Schéma interne du convertisseur	134
Figure 6.11: Forme d'onde du courant dans la phase a de SW pour $\Delta t = 5\mu\text{s}$	135
Figure 6.12: Forme d'onde de la tension de phase a de la charge au BUS 2 pour $\Delta t = 5\mu\text{s}$	135

LISTE DES SIGLES ET ABRÉVIATIONS

AC	Alternating Current (Courant alternatif)
AL	Arc Length
ALA	Arc Length Adaptée
BDF	Backward Differentiation Formula
BE	Backward Euler
CDA	Critical Damping Adjustment
CP	Constant Parameter (modèle de ligne)
CRT	Chatter Removal Technique
DASSL	Differential/Algebraic System Solver
DC	Direct Current (Courant continu)
EMT	Electro Magnetic Transient (Transitoire Electro Magnétique)
EMTP	Electro Magnetic Transient Program
FE	Forward Euler
IGBT	Insulated Gate Bipolar Transistor (transistor bipolaire à grille isolée)
LTE	Local Truncation Error (Erreur de Troncature Locale)
MANA	Modified Augmented Nodal Analysis
MTI	Modification de la Tension Itérative
NR	Newton-Raphson standard
PWM	Pulse Width Modulation (modulation à largeur d'impulsion)
RMS	Root Mean Square
RRE	Relative RMS Error (erreur relative RMS)
SS	Simultaneous Switching (commutations simultanées)
TAM	Time Average Method

TR	Trapézoïdale
TR_BE	Méthode d'intégration trapézoïdale combinée à la méthode Backward Euler
TS	Transient Stability (Stabilité transitoire)

LISTE DES ANNEXES

Annexe A – NOMENCLATURE ET DESCRIPTION DES MÉTHODES ET TECHNIQUES DE SIMULATION DISCUÉES	148
Annexe B – DÉTERMINATION D'UNE VALEUR INCONNUE à PARTIR D'UNE TABLE DE VALEURS	150
Annexe C - CALCUL DE LA LONGUEUR D'ARC POUR LA MÉTHODE ARC LENGTH DANS LE CAS D'UNE REPRÉSENTATION LINÉAIRE PAR SEGMENTS.....	152

CHAPITRE 1 INTRODUCTION

1.1 Contexte et motivation

Les réseaux électriques et les circuits d'électronique de puissance évoluent continuellement. De nouveaux ouvrages sont sans cesse construits et raccordés aux réseaux existants. Ces ouvrages assurent la production (centrales électriques), le transport et la distribution (lignes, sous-stations), l'adaptation (transformateurs, convertisseurs) ou la consommation (charges) de l'énergie électrique. Afin de garantir le dimensionnement adéquat, la protection et l'opération dans des conditions convenables, fiables et optimales de ces ouvrages, il est nécessaire de faire des prédictions du comportement des systèmes électriques dans leur globalité. Ceci est rendu possible grâce à la simulation. Cette opération est réalisée par des logiciels dédiés qui exécutent des algorithmes informatiques. Un algorithme combine un ensemble de méthodes et de techniques dans un ordre précis afin de réaliser une simulation. Pour un circuit donné, l'algorithme construit les modèles des différents composants, assemble les équations du circuit, les solutionne en utilisant différentes techniques et méthodes, et enfin présente les résultats à l'utilisateur. Dépendamment des objectifs visés et du type d'étude, une simulation peut se faire en régime permanent ou en régime transitoire. L'outil de simulation peut être directement connecté au système électrique et être en synchronisme avec l'horloge temps réel (simulation temps-réel) [1]-[4] ou alors être totalement indépendant (physiquement) du système simulé (simulation temps différé) [5]-[7]. Les simulations en régimes transitoires et en temps différé, (principal domaine d'intérêt de la présente thèse) contribuent à prévoir le comportement dynamique des réseaux et circuits électriques à la suite de perturbations ou de discontinuités.

Dans les grands réseaux, les perturbations peuvent être provoquées par l'enclenchement d'un banc de condensateurs ou d'un transformateur, l'apparition d'un court-circuit, la perte ou la connexion de charge, etc... Des phénomènes dits transitoires, aux contenus fréquentiels variés peuvent apparaître [8]. Ils résultent de la réponse forcée et de la réponse naturelle du circuit simulé et créent en général des conditions de stress pouvant endommager l'appareillage et/ou causer l'interruption de l'alimentation électrique [9],[10]. Il est donc crucial de les étudier en détail afin de pouvoir soit les éviter, soit être capables de protéger l'appareillage électrique. On réalise pour cela des simulations dites dans le domaine du temps. Il existe plusieurs catégories de logiciels de simulation,

dépendamment de la gamme des fréquences des transitoires. Pour des variations lentes, des logiciels de type « stabilité transitoire » (TS) sont recommandés tandis que pour des variations plus rapides, des logiciels de type « transitoire électromagnétique » (EMT) sont nécessaires [11].

Dans les circuits d'électronique de puissance, les perturbations conduisent à des discontinuités. Cela peut être par exemple le changement d'état d'un interrupteur électronique, le déplacement du point d'opération d'un composant non-linéaire sur sa caractéristique, etc... Ces discontinuités créent des erreurs dans la simulation qui impactent sur la précision. Des logiciels dédiés [11] sont nécessaires pour un traitement adéquat des discontinuités et plus généralement pour la simulation des circuits d'électronique de puissance.

Un logiciel de simulation effectue dans l'ordre les 5 étapes suivantes [12] :

- Étape 1 : La saisie des données. C'est à cette étape que l'utilisateur fournit au logiciel les informations sur la topologie du réseau ou du circuit à simuler, ainsi que les types de composants et les valeurs des paramètres de ces composants.
- Étape 2 : La création des modèles. Un modèle est créé pour chaque composant. Il s'agit en général d'un modèle de type Norton qui se présente sous la forme d'un circuit équivalent. La Figure 1.1 présente un exemple de circuit équivalent Norton pour un composant situé entre les nœuds k et m d'un réseau. Le courant i qui traverse ce composant et la tension v à ses bornes sont calculés à partir des paramètres y_N et i_N du modèle.

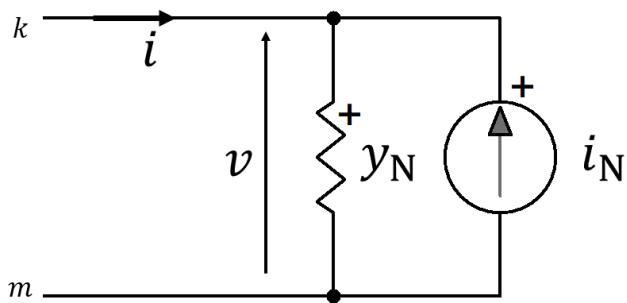


Figure 1.1: Exemple de circuit équivalent Norton (modèle Norton)

- Étape 3 : La formulation des équations. Des équations sont formulées et assemblées d'une façon particulière en respectant: la loi de Kirchhoff sur les courants (loi des nœuds), la loi de Kirchhoff sur les tensions (loi des mailles) et les équations issues des modèles des composants. La formulation des équations peut utiliser l'approche par équations d'état ou

alors l'approche nodale [12]. Cette dernière est généralement préférée pour sa relative simplicité. L'approche nodale a servi de base pour la création d'autres méthodes de formulation à l'instar de l'approche nodale modifiée [13], l'approche tableau [14] ou encore l'approche MANA [5],[6],[8],[15],[16]. Le système d'équations obtenu est de la forme :

$$\mathbf{Ax} = \mathbf{b} \quad (1.1)$$

Dans lequel \mathbf{A} est la matrice principale du réseau, \mathbf{x} est le vecteur des inconnues et \mathbf{b} est le vecteur des variables connues. Il est également possible de combiner plusieurs approches de formulation, selon les besoins [17],[18].

- Étape 4 : La résolution numérique. Le système d'équations obtenu à l'étape précédente est solutionné de façon discrète par intégration numérique à chaque point de temps jusqu'à ce que la durée totale de la simulation t_{sim} soit atteinte. L'intervalle de temps entre deux points de temps discrets et consécutifs est appelé pas d'intégration numérique, noté Δt . Il est généralement fixe dans la plupart des logiciels, mais peut être rendu variable, moyennant des transformations mathématiques. L'ensemble des points de temps de la simulation représente la ligne de temps, souvent désigné *time-mesh* ou encore *time-grid*. Un exemple est présenté à la Figure 1.2.

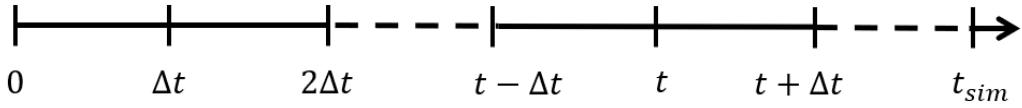


Figure 1.2: Exemple de ligne de temps de la simulation (time-mesh)

Des techniques et méthodes additionnelles (interpolation, réinitialisation, traitement des fonctions non-linéaires, adaptation du pas d'intégration, etc...) peuvent être rajoutées à cette étape de résolution afin d'améliorer la rapidité de la simulation et/ou la précision des résultats.

- Étape 5 : La présentation des résultats. Elle consiste simplement à mettre les résultats de la simulation à la disposition de l'utilisateur sur le format le mieux adapté selon ses besoins.

Avec le développement des systèmes électriques et l'intégration de composants de plus en plus complexes, les logiciels de simulation font face en permanence à l'exigence de produire des

résultats suffisamment proches de la réalité en des temps de calcul raisonnables (pas trop élevés). Ce double objectif a donné lieu à plusieurs contributions présentes dans la littérature.

1.2 Revue de la littérature

Les exigences et les problématiques auxquelles font face les logiciels ne sont pas nécessairement les mêmes, dépendamment de s'ils sont conçus pour simuler les grands réseaux ou alors les circuits d'électronique de puissance [11]. Les méthodes développées dans la littérature sont donc spécifiques à l'une ou à l'autre de ces deux catégories.

1.2.1 Simulation des grands réseaux électriques

Les logiciels dédiés à la simulation des grands réseaux tels que ATP [19], EMTDC [20],[21] et EMTP [5],[6] font majoritairement face à la taille et la complexité grandissante des ouvrages. Il en résulte des temps de calcul très élevés. Fort de ce constat, des méthodes de plus en plus évoluées sont nécessaires pour réaliser des simulations rapides et précises. Ces méthodes peuvent être classifiées selon quatre grands axes [22]. Le premier axe concerne le développement des équivalents de réseaux et consiste à développer des modèles simplifiés pour des parties plus ou moins larges des réseaux à simuler [23]-[28] afin de réduire la complexité globale. Le deuxième axe traite de la simulation hybride. Initialement proposée dans [29], puis reprise dans [30], la simulation hybride consiste à simuler un réseau en utilisant à la fois un logiciel de type EMT (pour les transitoires rapides), et un logiciel de type TS (pour les transitoires lentes ou le régime permanent) [31]-[34]. Le troisième axe s'intéresse à la résolution en parallèle. Avec cette approche, un large réseau est partitionné en plusieurs sous-réseaux [35] qui sont solutionnés simultanément. Il existe plusieurs techniques de partitionnement [36]-[38]. Elles permettent d'exploiter judicieusement les délais de propagation introduits par les modèles des longues lignes de transmission [38]-[40]. Les différents sous réseaux obtenus peuvent être simulés avec différents pas d'intégration numérique dépendamment de la dynamique de chaque sous-réseau [41]. Le quatrième axe concerne l'adaptation fréquentielle. Cette approche consiste à appliquer une transformation aux équations qui décrivent la dynamique d'un réseau de manière à obtenir des formes d'onde qui varient plus lentement. Ceci rend théoriquement possible l'utilisation de pas d'intégration numérique plus élevés [42]-[44].

Les méthodes développées dans chacun des axes présentés ci-dessus sont assez complexes et font face à de nombreuses difficultés d'implémentation qui les rendent difficiles à généraliser. Pour ces raisons, les travaux de recherche présentés dans cette thèse n'iront pas dans ces directions. La problématique de la simulation des grands réseaux sera traitée sous un angle lié à la gestion efficiente des discontinuités. En effet, une discontinuité peut résulter d'une perturbation (manœuvre de disjoncteur, etc...) et la représentation des comportements transitoires qui en découlent est la principale source de perte en précision et d'augmentation des temps de calcul dans les logiciels. Cela tient au fait que les discontinuités font apparaître des fréquences naturelles variant dans une gamme très large. La plupart des algorithmes utilisent un pas d'intégration constant tout au long de la simulation. Ce dernier doit donc nécessairement être choisi en fonction des fréquences transitoires les plus élevées (qui généralement ont une durée de vie très courte). Le pas d'intégration choisi (qui est en général très petit) est inutilement maintenu lorsque les valeurs des fréquences transitoires deviennent beaucoup plus faibles ou lorsque le régime permanent est atteint. Cela constitue une perte d'efficacité significative, puisque des valeurs de pas d'intégration beaucoup plus élevées auraient pu être utilisées pour obtenir le même niveau de précision, réduisant significativement les durées de simulation.

La simulation à pas d'intégration numérique adaptatif (variable) apparait théoriquement comme une excellente solution à ce problème de perte d'efficacité. Le principe de base consiste à adapter la valeur du pas d'intégration selon la valeur de la fréquence transitoire maximale présente sur le réseau. De cette façon, il est possible de conserver une précision très bonne tout au long de la simulation et éviter en même temps les pertes d'efficacité dues à l'utilisation de pas d'intégration inadaptés à certains moments de la simulation.

L'approche de simulation utilisant un pas d'intégration variable est très peu discutée dans la littérature, ceci à cause des difficultés pratiques d'implémentation. Par exemple, il n'est pas toujours facile de choisir la bonne valeur du pas d'intégration à utiliser selon le circuit simulé. En théorie il serait possible de sélectionner le pas d'intégration numérique en fonction des constantes de temps du circuit, mais ces constantes ne sont généralement pas connues en pratique. De plus, bien qu'il soit toujours mathématiquement possible de prévoir la dynamique du circuit à partir du calcul des valeurs propres et de sélectionner le pas d'intégration en conséquence, cette approche est très complexe et n'est parfois pas suffisante à cause notamment des non-linéarités et/ou de la présence des systèmes de control qui imposent des contraintes supplémentaires sur le choix du pas

d'intégration. Dans [45], le pas d'intégration et la méthode d'intégration numérique sont simultanément modifiés selon la valeur de l'erreur de troncature locale. Cette approche est complexe dans sa mise en œuvre et présente des risques à cause de la méthode d'intégration numérique qui peut devenir instable. Dans [46] il est proposé une méthode qui utilise un pas d'intégration petit seulement dans des zones de réseau siège de transitoires rapides. Un pas plus grand est utilisé pour le reste du réseau. Cette approche est non seulement complexe à mettre en œuvre, mais aussi ne peut être applicable que lorsque le réseau simulé contient des lignes de transport (puisque la méthode exploite le délai de propagation introduit par le modèle de la ligne). En outre, il n'est pas facile de garantir la précision dans chacune des zones du circuit utilisant des pas d'intégration différents. Utiliser plusieurs pas d'intégration pour simuler le même réseau est donc finalement très complexe. Les travaux de cette thèse seront concentrés sur la possibilité d'utiliser le même pas d'intégration pour tout le réseau simulé. Ce pas pourra changer de valeur en fonction du temps mais sa valeur restera toujours unique en un instant donné, sur tout le réseau. Cette façon de faire présente l'avantage de faciliter l'intégration aux logiciels de simulation existants, des méthodes développées suivant cette logique.

Bien que rejeter la possibilité d'utiliser des pas multiples à différents endroits d'un réseau à un instant donné d'une simulation permet une réduction relative de la complexité, il faut reconnaître que dans tous les cas, utiliser un pas d'intégration numérique variable reste une approche complexe à cause des modifications à apporter aux équations chaque fois que le pas d'intégration change. Pour cette raison, l'utilisation d'un pas variable dans la simulation est généralement limitée aux applications spécifiques et de faible envergure telles que l'optimisation de la commande des machines [47],[48].

1.2.2 Simulation des circuits d'électronique de puissance

Avec le développement de la technologie, les circuits d'électronique de puissance contiennent un nombre de plus en plus élevé d'interrupteurs électroniques qui commutent à des fréquences de plus en plus élevées. La plupart de ces circuits contiennent également des composants non-linéaires. Tout ceci contribue à augmenter significativement les contraintes liées à la simulation. Les logiciels de type SPICE [49] font en effet face à plusieurs défis.

Le premier défi est celui de la détection précise des discontinuités. Une discontinuité est causée par un changement d'état d'un interrupteur électronique, ou par un déplacement important du point

d'opération d'un composant non-linéaire. Durant une simulation, des changements d'état des interrupteurs peuvent survenir entre deux points de temps consécutifs. Par exemple, sur la Figure 1.2, un changement d'état peut se produire à un instant t_d compris entre t et $t + \Delta t$. Ce changement ne sera pris en compte qu'à l'instant $t + \Delta t$, créant ainsi une erreur dans la simulation. On peut de même envisager l'occurrence d'une seconde discontinuité à un instant $t_{d'} \neq t_d$ situé toujours entre t et $t + \Delta t$, ce qui contribuerait à augmenter l'erreur qui se crée. L'utilisation d'un pas d'intégration numérique Δt très petit permettrait de réduire l'erreur mais augmenterait significativement la durée de la simulation. Des techniques d'interpolation [50]-[54] sont proposées dans la littérature. Elles permettent à partir des solutions à t et à $t + \Delta t$, d'estimer la solution à l'instant de discontinuité t_d et de ramener ainsi la simulation à ce point-là afin de prendre en compte la discontinuité au moment précis où elle se produit. L'interpolation peut être également utilisée un pas plus tard pour resynchroniser la simulation sur son time-mesh. L'interpolation peut être linéaire ou quadratique [55]. L'interpolation est en général très efficace car elle permet de conserver la précision dans le traitement des discontinuités sans avoir besoin de baisser le pas d'intégration. Toutefois, dans certains cas, il a été constaté qu'un algorithme qui n'utilise pas l'interpolation peut donner dans certaines conditions un résultat plus précis qu'un algorithme qui interpole [56].

Le deuxième défi est celui de la réinitialisation. En effet, une discontinuité provoquant le changement d'état d'un ou de plusieurs interrupteurs peut conduire à une modification importante de la topologie du réseau. Cette modification peut à son tour produire des changements soudains des valeurs de certaines variables du réseau (courants et tensions). Ces changements doivent être effectifs dans la simulation au moment même où ils se produisent. Pour cela, la réinitialisation des variables est requise à la suite de chaque discontinuité. Cette opération impose un recalcul de la solution au point de temps d'apparition de la discontinuité. Dans [57], la technique proposée consiste à remplacer les condensateurs et les inductances respectivement par des sources de tension et de courant continu, puis de solutionner le nouveau circuit obtenu. Cette approche nécessite toutefois une reformulation complète des équations. De plus, il est évident qu'elle est inutilisable pour des circuits complexes pouvant contenir un nombre très élevé de composants parmi lesquels les non-linéarités. En revanche, dans [52], la technique proposée consiste à avancer dans le temps d'un pas (avec une intégration Backward Euler) puis ramener la valeur calculée un pas en arrière. Cette approche n'est toutefois précise que lorsque le pas d'intégration est très petit.

Les changements soudains des valeurs des variables à la suite d'une discontinuité peuvent créer des conditions de changement d'état d'autres interrupteurs. Ce phénomène porte le nom de commutations simultanées ou « simultaneous switching » (SS). Les cas de commutation simultanées sont identifiés et convenablement traités en solutionnant le circuit plusieurs fois au même point de temps et en vérifiant à chaque fois si de nouveaux changements doivent être effectués, comme proposé dans [5].

Afin de minimiser l'erreur créée à la traversée d'une discontinuité, la détection précise de la discontinuité (par interpolation) et la réinitialisation des variables doivent être exécutées. Combiner ces deux exigences peut devenir problématique et contribuer à l'introduction de nouvelles erreurs, dues au fait que le point d'interpolation t_d n'est pas situé sur le time-mesh. Dans [58], l'étape de réinitialisation par résolution de (1.1) crée une erreur additionnelle parce que le calcul ne correspond pas à un saut de Δt dans le temps. Une solution à ce problème serait d'obtenir la solution à $t_d - \Delta t$ (par interpolation) et l'utiliser pour la réinitialisation, tel que proposé dans [59]. Cette approche peut toutefois s'avérer non précise, en particulier dans le cas où d'autres discontinuités se sont produites entre $t_d - \Delta t$ et t_d .

Outre les défis de détection précise des discontinuités et de réinitialisation des variables qui sont directement liés à la précision, les logiciels de simulation des circuits d'électronique de puissance font également face au problème des temps de calcul élevés. Ce troisième défi est engendré par deux choses. Premièrement, les modifications que subit la matrice **A** de l'équation (1.1) chaque fois qu'une discontinuité se produit imposent sa refactorisation, ce qui consomme du temps. Deuxièmement, la nécessité de faire plusieurs itérations au même point de temps afin de déterminer le point d'opération de chaque composant non-linéaire conduit simultanément à des modifications de la matrice **A** et à des calculs multiples au même instant. Pour s'affranchir de ce problème global de temps de calculs élevés, certaines recherches ont développé des modèles équivalents permettant de simplifier les calculs. Dans la méthode TAM (Time Average Method) par exemple [60]-[62] des blocs d'interrupteurs sont remplacés par des sources de tension dont la valeur dépend de l'état de conduction des interrupteurs. Cette méthode simplifie significativement les calculs et est utilisée en simulation temps réel. Cependant, les simplifications dans les modèles ne permettent pas toujours de représenter fidèlement les formes d'onde. D'autres travaux présentent des approches alternatives de modélisation. Par exemple, [63] propose de représenter les interrupteurs par des

équivalents Norton comme à la Figure 1.1 avec la valeur de y_N qui reste constante malgré les changements d'état. Dans cette approche, ces changements se matérialisent uniquement par des modifications de i_N . Cela permet d'éviter les refactorisations de \mathbf{A} à la suite de chaque discontinuité, et représente un gain dans les temps de calcul. Cependant, non seulement cette méthode n'est précise que lorsque Δt est suffisamment petit, mais en plus, la détermination de y_N est hasardeuse et la valeur choisie affecte significativement la précision. Pour essayer d'éliminer cette limitation majeure, [64] propose une méthode optimale de détermination de y_N . Toutefois, des simulations montrent que les résultats obtenus ne sont toujours pas suffisamment précis, ce qui limite la portée et l'utilisation de cette méthode.

Une façon efficace de réduire les temps de calcul lors de la simulation des circuits (surtout ceux contenant des non-linéarités) consiste à limiter le nombre d'itérations effectuées pour la détermination des points d'opération des composants non-linéaires. Les itérations sont généralement effectuées par la méthode Newton [65] mais cette approche peut dans certains cas nécessiter un nombre important d'itérations ou rencontrer des problèmes de convergence. C'est généralement le cas lorsque les circuits simulés contiennent des composants fortement non-linéaires comme les diodes. Pour régler cela, [66] a proposé une approche visant globalement à améliorer la vitesse de convergence de la méthode Newton. Cette approche est reprise et légèrement modifiée dans [67]. Le logiciel SPICE [49] implémente une stratégie visant à contrôler la précision et à limiter le nombre d'itérations par variation du pas d'intégration numérique dans les simulations. Cette stratégie conduit malheureusement à des temps de calcul très élevés.

Il est important de souligner que la simulation des circuits d'électronique de puissance peut se heurter également à des problèmes de stabilité liés aux multiples commutations (plus précisément à la cadence des commutations) et liés à la présence de composants non-linéaires [68]. Cette problématique ne sera toutefois pas abordée dans cette thèse.

La revue de la littérature qui vient d'être proposée montre que les méthodes de simulation existantes présentent des insuffisances qui limitent leur efficacité. Pour la plupart d'entre elles, les gains en temps de calcul se font généralement au détriment de la précision. De la même manière, les méthodes qui améliorent la précision le font au détriment des temps de calcul ou de la complexité. De plus, l'impact réel de certaines approches tel que l'interpolation n'est pas

clairement et profondément compris. Il existe des zones d'ombres et des problèmes non ou partiellement résolus.

1.3 Objectifs de la thèse

Cette thèse se donne pour objectif général de proposer de nouvelles méthodes pour la simulation plus rapide et plus précise des transitoires de réseaux et des circuits d'électronique de puissance. Les méthodes à proposer seront axées sur la gestion des discontinuités et devront être capables de facilement s'intégrer aux logiciels de simulation existants. Cet objectif général se décline en objectifs spécifiques qui sont les suivants :

- Proposer une méthode de simulation des transitoires de réseaux et des circuits d'électronique de puissance qui utilise un pas d'intégration numérique variable (adaptatif) et qui permet de réduire les temps de calcul tout en améliorant la précision;
- Fournir une analyse plus détaillée du comportement de l'erreur à la traversée des discontinuités lors de la simulation des circuits d'électronique de puissance, plus clarifier l'impact sur la précision des techniques existantes telles que l'interpolation et la reinitialisation;
- Proposer de nouvelles techniques de simulation des circuits d'électronique de puissance permettant de garantir la précision à la traversée des discontinuités, sans significativement augmenter les temps de calcul;
- Proposer une nouvelle méthode plus efficace que la méthode Newton pour simuler les circuits contenant des composants non-linéaires.

1.4 Contributions de la thèse

Les contributions principales de cette thèse sont les suivantes :

- Conception d'un algorithme utilisant un pas d'intégration numérique variable pour la simulation des transitoires de réseaux électriques;
- Conception d'un algorithme utilisant un pas d'intégration numérique variable pour la simulation des circuits d'électronique de puissance;

- Analyse de l'impact des techniques d'interpolation sur le comportement de l'erreur à la traversée des discontinuités;
- Analyse de l'impact sur l'erreur de la réinitialisation, ainsi que de la resynchronisation ou non au time-mesh à la suite d'une discontinuité;
- Proposition de trois méthodes de simulation permettant de traverser les discontinuités sans perdre en précision;
- Adaptation de la méthode arc length à la formulation nodale modifiée augmentée pour la simulation des circuits non-linéaires.

1.5 Organisation de la thèse

Le présent chapitre est introductif et permet d'énoncer le contexte, les problèmes et les contributions existantes dans la littérature. Par la suite, le chapitre 2 présente l'approche générique de simulation dans le domaine du temps. La modélisation des composants électriques, la formulation des équations, l'intégration numérique et la résolution des équations sont tour à tour abordés. Le chapitre 3 présente deux nouveaux algorithmes utilisant un pas d'intégration numérique variable. Les transformations mathématiques des méthodes, ainsi que les gains en temps de calcul et en précision sont démontrés. Le chapitre 4 analyse l'impact des techniques utilisées lors de la traversée de discontinuités. Le chapitre 5 s'inspire des analyses faites au chapitre 4 pour proposer trois nouvelles méthodes pour la simulation des circuits d'électronique de puissance. Pour terminer, le chapitre 6 présente une nouvelle approche pour la simulation des circuits d'électronique de puissance contenant des composants non-linéaires.

CHAPITRE 2 APPROCHE GÉNÉRALE DE MODÉLISATION ET DE SIMULATION DES CIRCUITS ÉLECTRIQUES

L'objectif de cette thèse étant de développer de nouvelles méthodes de simulation dans le domaine du temps, il est important au préalable de rappeler l'approche usuelle telle qu'implémentée dans les logiciels de simulation existants. Ce chapitre présente une description des méthodes d'intégration numérique, explique l'approche de modélisation des composants, de formulation et de résolution des équations lors de la simulation dans le domaine du temps.

2.1 Les méthodes d'intégration numérique et la création des modèles

Les équations qui caractérisent la majorité de composants électriques sont algébriques et/ou différentielles. L'équation de type différentiel qui décrit le comportement d'une variable x d'un composant, exprimée en fonction du temps t se présente comme suit :

$$\dot{x} = f(x, t) \quad (2.1)$$

Dans cette équation, f est un opérateur. Les logiciels ne sont pas capables de manipuler directement des équations différentielles. Ces dernières doivent être transformées en équations algébriques par intégration numérique.

2.1.1 Les méthodes d'intégration numérique

2.1.1.1 Description générale

L'intégration numérique fournit un moyen de calculer une valeur discrète $x_{t+\Delta t}$ de la variable différentielle x de l'équation (2.1) à un instant $t + \Delta t$, à partir des valeurs x_t , $x_{t-\Delta t}$, ... aux instants discrets précédents t , $t - \Delta t$, ... La grandeur Δt représente le pas d'intégration numérique. Il existe théoriquement une infinité de méthodes d'intégration. Elles sont construites à partir du développement en séries de Taylor ou par approximation polynomiale [12]. Le calcul de $x_{t+\Delta t}$ nécessite l'évaluation de la fonction f aux instants discrets $t + \Delta t$, t , $t - \Delta t$, ... soit $f(x_{t+\Delta t}, t + \Delta t) = f_{t+\Delta t}$, $f(x_t, t) = f_t$, $f(x_{t-\Delta t}, t - \Delta t) = f_{t-\Delta t}$, ...

Lorsque le calcul de $x_{t+\Delta t}$ ne fait intervenir que f_t et les valeurs de f aux instants précédents, la méthode d'intégration est dite explicite. Si la valeur de $f_{t+\Delta t}$ est requise, la méthode est dite implicite. D'un autre côté, lorsque le calcul de $x_{t+\Delta t}$ ne nécessite la valeur de x qu'à l'instant $t(x_t)$, la méthode d'intégration est dite d'ordre 1. Cet ordre augmente de 1 chaque fois qu'une valeur précédente de x est requise pour l'intégration. Par exemple, une méthode qui utilise les valeurs de x_t , $x_{t-\Delta t}$ et $x_{t-2\Delta t}$ pour calculer $x_{t+\Delta t}$ est d'ordre 3. Lorsque l'ordre d'une méthode dépasse 1, elle est dite multi-pas. Pour certaines méthodes d'ordre 1, le calcul de $x_{t+\Delta t}$ fait intervenir x_t , mais aussi les valeurs de x à certains instants intermédiaires (précalculés) entre t et $t + \Delta t$. Ces méthodes sont dites multi étapes.

2.1.1.2 Critères d'évaluation des méthodes d'intégration numérique

Les méthodes d'intégration numériques qu'elles soient explicites ou implicites, d'ordre 1 ou d'ordre supérieur, peuvent être évaluées à l'aide de trois critères.

Le premier critère est la précision. C'est la capacité d'une méthode à produire une solution numérique suffisamment proche de la solution exacte. On l'évalue par le calcul de l'erreur dite de troncature locale. C'est l'erreur sur le calcul de la solution à un instant donné en assumant que toutes les valeurs obtenues aux instants précédents sont exactes. Cette erreur est fonction du pas d'intégration Δt , souvent désigné par h . On utilise alors la notation $O(h^{p+1})$ [56] pour signifier qu'une méthode a une précision d'ordre p . Plus p est élevé, plus la méthode est précise.

Le deuxième critère est la stabilité numérique. Dans le cas général, on dit qu'une méthode numérique est stable lorsque qu'elle atténue les erreurs de calcul quand on avance dans le temps. Fondamentalement, une méthode doit être capable de produire une solution numérique stable lorsque le circuit simulé est stable. De telles méthodes sont dites A - stables [69]. La stabilité des méthodes explicites varie selon la valeur du pas d'intégration numérique [12]. Cependant, les méthodes implicites sont généralement A - stables, mais peuvent parfois rencontrer des problèmes de stabilité lorsqu'elles simulent des circuits non-linéaires [70] ou lorsque l'ordre de la méthode dépasse 2 [71]. Aussi, certains circuits sont siège de variations tantôt très rapides, tantôt très lentes. Ils sont dits « stiff » [12],[72]. Pour certaines méthodes A - stables, des variations très rapides peuvent faire apparaître des oscillations numériques qui s'amortissent très faiblement ou pas du tout. Ce phénomène se produit en présence de circuits « stiff ». Certaines méthodes d'intégration

numériques sont capables d'amortir totalement ces oscillations en un pas de calcul. On dit qu'elles possèdent la propriété de stabilité « stiff-decay » [73]. On les qualifie alors de méthodes L-stables [72].

Le troisième critère est la rapidité. Cette notion renvoie à la vitesse avec laquelle les équations construites à partir de ces méthodes sont solutionnées. La rapidité d'une méthode est directement liée au degré de complexité dans sa formulation mathématique. Cette complexité est plus élevée pour les méthodes d'ordre élevé et pour les méthodes multi étapes.

2.1.1.3 Quelques méthodes d'intégration numérique

Il existe plusieurs méthodes d'intégration numérique, ayant des caractéristiques et des performances variées. Quelques-unes parmi les plus usuelles sont décrites ci-dessous. Les équations permettant de discréteriser (2.1) afin de calculer la valeur de la variable x à un instant $t + \Delta t$ sont présentées.

2.1.1.3.1 La méthode « Forward Euler »

C'est une méthode explicite, qui n'est pas A – stable [12]. Elle est très simple à implémenter et a une faible précision (ordre 1). Cette méthode est référée par FE. Sa formulation est la suivante :

$$x_{t+\Delta t} = x_t + \Delta t \cdot f_t \quad (2.2)$$

2.1.1.3.2 La méthode « Backward Euler »

C'est une méthode implicite, A – stable et L – stable [12]. Elle est simple à implémenter et a une précision d'ordre 1. Cette méthode est référée par BE. Sa formulation est la suivante :

$$x_{t+\Delta t} = x_t + \Delta t \cdot f_{t+\Delta t} \quad (2.3)$$

2.1.1.3.3 La méthode « trapézoidale »

C'est une méthode A – stable, relativement simple à implémenter. Elle a une précision d'ordre 2 [12] et est de ce fait plus précise que les méthodes FE et BE. Cette méthode est référée par TR. Sa formulation est la suivante :

$$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} [f_{t+\Delta t} + f_t] \quad (2.4)$$

Il existe une variante désignée « *trapézoidale avec amortissement* » (trapezoidal with damping) [74]. Sa formulation est la suivante :

$$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} [(1+\alpha)f_{t+\Delta t} + (1-\alpha)f_t] \quad (2.5)$$

Le paramètre α est appelé facteur d'amortissement. Pour $\alpha = 0$, cette méthode correspond à la méthode TR tandis que pour $\alpha = 1$, on retrouve la méthode BE.

2.1.1.3.4 Les méthodes « Backward Differentiation Formula » (BDF)

Ce sont des méthodes multi-pas encore appelées « formules de GEAR » [12],[72] qui s'expriment comme suit :

$$\sum_{k=0}^s a_k x_{t+k\Delta t} = \Delta t \cdot f(t + s \cdot \Delta t, x_{t+s \cdot \Delta t}) \quad (2.6)$$

Les coefficients (a_k) sont à déterminer selon l'ordre s de la formule. Pour $s = 1$, on peut construire par approximation polynomiale la méthode BE présentée plus haut. Pour $s = 2$, on peut construire la méthode suivante :

$$x_{t+2\Delta t} = \frac{4}{3}x_{t+\Delta t} - \frac{1}{3}x_t + \frac{2\Delta t}{3}f_{t+2\Delta t} \quad (2.7)$$

Il s'agit de la méthode « Gear 2 ». Elle est A – stable et L – stable. Sa précision est d'ordre 2. Elle est toutefois moins précise que la méthode TR, ceci à cause de son erreur de troncature locale qui est plus faible. La notion d'erreur de troncature sera discutée plus tard dans cette thèse. Les méthodes Gear vont jusqu'à l'ordre 6 et sont toutes utilisées dans le solveur DASSL [45]. Les méthodes Gear sont particulièrement efficaces pour simuler les circuits « stiff ». Toutefois, elles doivent être correctement initialisées. C'est cela leur principal inconvénient. En effet, une méthode Gear d'ordre s a besoin des valeurs des $s-1$ points de temps précédents. Ceci pose un problème au début de la simulation. Une mauvaise initialisation peut créer une erreur qui s'amplifie tout au long de la simulation.

2.1.1.3.5 La méthode 2S-DIRK

C'est une méthode multi-étapes [73],[75],[76]. Partant de x_t , la méthode calcule $x_{t+\Delta t}$ en 3 étapes.

La première étape est une intégration pour calculer une solution intermédiaire $\tilde{x}_{t+\Delta t}$ à un instant

$$\tilde{t} = t + a \cdot \Delta t \text{ avec } a = 1 - \frac{1}{\sqrt{2}} :$$

$$\tilde{x}_{t+\Delta t} = x_t + a \cdot \Delta t \cdot f_{\tilde{t}} \quad (2.8)$$

La deuxième étape est une conversion pour obtenir $\tilde{x}_{t+\Delta t}^*$ en utilisant $\alpha = -\sqrt{2}$ et $\beta = 1 + \sqrt{2}$:

$$\tilde{x}_{t+\Delta t}^* = \alpha \cdot x_t + \beta \cdot \tilde{x}_{t+\Delta t} \quad (2.9)$$

Enfin, la dernière étape permet d'obtenir $x_{t+\Delta t}$:

$$x_{t+\Delta t} = \tilde{x}_{t+\Delta t}^* + a \cdot \Delta t \cdot f_{t+\Delta t} \quad (2.10)$$

La méthode 2S-DIRK est A – stable, L – stable et a une précision d'ordre 2. Son principal inconvénient est la complexité de mise en œuvre. De plus, la précision de cette méthode se dégrade lorsque les variations de x deviennent très rapides, ce qui est le cas par exemple lorsqu'on simule un circuit contenant des hautes fréquences [77],[78].

2.1.1.3.6 La méthode TR-BDF2

Il est possible de combiner la méthode TR avec une méthode BDF d'ordre 2. Le résultat est TR-BDF2 qui est une méthode d'intégration multi étapes [56],[79],[80]. Le passage de x_t à $x_{t+\Delta t}$ commence par le calcul d'une solution intermédiaire x_γ au point $t + \gamma \cdot \Delta t$ avec $\gamma = 2 - \sqrt{2}$ par intégration trapézoïdale :

$$x_\gamma = x_t + \gamma \frac{\Delta t}{2} \left[f_t + f_{t+\gamma \cdot \Delta t} \right] \quad (2.11)$$

Ensuite, la solution finale $x_{t+\Delta t}$ est calculée par une méthode BDF d'ordre 2 :

$$x_{t+\Delta t} = (1 - \eta) x_t + \eta x_\gamma + \gamma \frac{\Delta t}{2} f_{t+\Delta t} \quad (2.12)$$

Avec $\eta = \frac{1+\sqrt{2}}{2}$. Cette méthode d'intégration numérique cumule les avantages de TR et ceux des méthodes BDF2. Mais, tout comme la méthode 2S-DIRK, elle est complexe à implémenter à cause du calcul de la solution intermédiaire.

2.1.2 La création des modèles des composants électriques

2.1.2.1 Approche globale

A partir d'une méthode d'intégration numérique, un modèle de type Norton (Figure 1.1) est créé pour chaque composant électrique décrit par une équation de la forme de (2.1). Le choix de la méthode d'intégration numérique est un exercice très complexe car chaque méthode présente des avantages et des inconvénients. La méthode TR fournit néanmoins un excellent compromis pour sa précision, sa simplicité et sa stabilité. C'est elle qui est utilisée dans la plupart des logiciels de simulation [11]. Puisqu'elle n'est pas L - stable, il est d'usage dans les algorithmes de combiner cette méthode avec une autre afin d'éviter les oscillations numériques. Pour cela, il est d'usage d'employer la méthode BE pour avancer la simulation les deux points de temps consécutifs à chaque discontinuité. Cette approche porte le nom de « Critical Damping Adjustment » (CDA) [81]. La méthode obtenue est référée par TR_BE. Les méthodes TR et BE seront utilisées ci-dessous pour démontrer le processus de création des modèles de certains composants électriques de base. Cette approche de modélisation a été initialement proposée dans [82].

2.1.2.2 Modèle de l'inductance

Une inductance idéale de valeur L , connectée entre deux nœuds k et m d'un réseau est caractérisée par l'équation suivante qui lit le courant i_{km} qui la traverse à la tension v_{km} à ses bornes :

$$\frac{di_{km}}{dt} = \frac{v_{km}}{L} \quad (2.13)$$

Puisque le courant apparaît dans la dérivée, on dit que c'est une variable d'état. Par application de la discrétisation trapézoidale (TR) décrite par (2.4), on obtient :

$$i_{km_{t+\Delta t}} = \frac{\Delta t}{2L} v_{km_{t+\Delta t}} + \frac{\Delta t}{2L} v_{km_t} + i_{km_t} \quad (2.14)$$

Dans cette équation, le courant à l'instant $t + \Delta t$ est la somme de deux courants : un courant obtenu en multipliant la tension à ce même instant par le facteur $\frac{\Delta t}{2L}$ (qui a la dimension d'une admittance), et un courant obtenu à partir des courants et tensions de l'instant précédent t . Cette composante est dite « historique ». L'équation (2.14) permet d'obtenir une représentation de l'inductance par le circuit équivalent (modèle) de la Figure 1.1, avec :

$$y_N = \frac{\Delta t}{2L} \quad (2.15)$$

$$i_N = -\left(\frac{\Delta t}{2L} v_{km_t} + i_{km_t} \right) \quad (2.16)$$

Cette correspondance permet de construire le modèle (circuit équivalent) Norton de l'inductance :

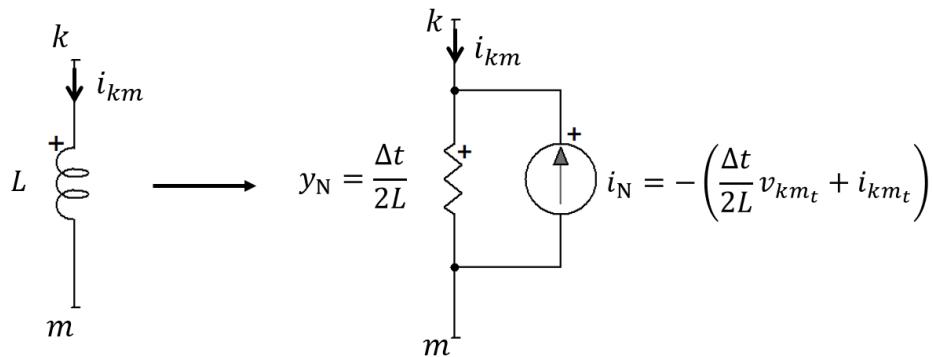


Figure 2.1: Modèle Norton d'une inductance idéale

On peut remarquer que pour la détermination de la solution à l'instant $t + \Delta t$, le courant i_N est obtenu à partir des valeurs de tension et courant à l'instant précédent t . Pour cette raison, i_N est appelé « courant historique ».

Si l'équation (2.13) est discrétisée avec BE décrite par (2.3), on obtient plutôt la relation :

$$i_{km_{t+\Delta t}} = \frac{\Delta t}{L} v_{km_{t+\Delta t}} + i_{km_t} \quad (2.17)$$

Ce qui permet de construire le même circuit équivalent, mais avec :

$$y_N = \frac{\Delta t}{L} \quad (2.18)$$

$$i_N = -i_{km} \quad (2.19)$$

2.1.2.3 Modèle du condensateur

De la même manière que précédemment, un condensateur idéal de valeur C , connecté entre deux nœuds k et m d'un réseau est caractérisée par l'équation :

$$\frac{dv_{km}}{dt} = \frac{i_{km}}{C} \quad (2.20)$$

Pour un condensateur, c'est la tension qui est une variable d'état. Par application de la discréétisation TR décrite par (2.4), on obtient :

$$i_{km_{t+\Delta t}} = \frac{2C}{\Delta t} v_{km_{t+\Delta t}} - \frac{2C}{\Delta t} v_{km_t} - i_{km_t} \quad (2.21)$$

Ce qui permet de construire le circuit équivalent de la Figure 1.1 avec :

$$y_N = \frac{2C}{\Delta t} \quad (2.22)$$

$$i_N = \frac{2C}{\Delta t} v_{km_t} + i_{km_t} \quad (2.23)$$

Si l'équation (2.20) est discréétisée avec BE décrite par (2.3), on obtient plutôt la relation :

$$i_{km_{t+\Delta t}} = \frac{C}{\Delta t} v_{km_{t+\Delta t}} - \frac{C}{\Delta t} v_{km_t} \quad (2.24)$$

Les valeurs du modèle Norton deviennent donc :

$$y_N = \frac{C}{\Delta t} \quad (2.25)$$

$$i_N = \frac{C}{\Delta t} v_{km_t} \quad (2.26)$$

2.1.2.4 Modèle de l'inductance non-linéaire

Une inductance non-linéaire peut être modélisée comme un composant linéaire par segment. Sa valeur dépend de l'intervalle dans lequel se situe le flux φ (ou le courant i) et sa caractéristique

est obtenue par la donnée d'un ensemble de N points (φ_j, i_j) avec j variant de 1 à N comme sur la Figure 2.2:

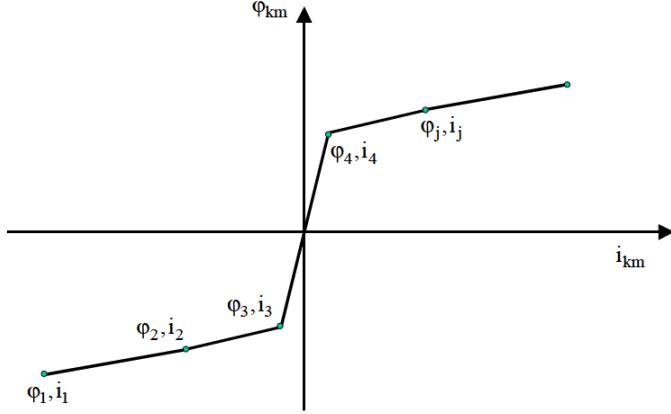


Figure 2.2: Caractéristique courant-flux d'une inductance non-linéaire

Ainsi, la relation de dépendance entre le flux φ_t et le courant i_t de l'inductance à tout instant t peut être décrite par une équation découlant de la représentation linéaire par segments :

$$\varphi_t = L_k^{jn} i_t + \varphi_k^{jn} \quad (2.27)$$

où L_k^{jn} (pente) et φ_k^{jn} (ordonnée à l'origine) dépendent du segment k sur lequel se trouve le point (φ_t, i_t) . Le segment k est situé entre les points frontières (φ_j, i_j) et (φ_n, i_n) .

D'un autre côté, le flux est lié à la tension par :

$$\frac{d\varphi_t}{dt} = v \quad (2.28)$$

La discrétisation TR de cette équation donne :

$$\varphi_{t+\Delta t} = \frac{\Delta t}{2} v_{t+\Delta t} + \frac{\Delta t}{2} v_t + \varphi_t \quad (2.29)$$

En écrivant (2.27) à l'instant $t + \Delta t$ et en égalant cette relation avec (2.29), on obtient :

$$i_{t+\Delta t} = \frac{\Delta t}{2L_k^{jn}} v_{t+\Delta t} + \frac{1}{L_k^{jn}} \left[\varphi_t - \varphi_k^{jn} + \frac{\Delta t}{2} v_t \right] \quad (2.30)$$

Ainsi, à l'instant $t + \Delta t$, connaissant la valeur de φ_t et les valeurs φ_k^{jn} et L_k^{jn} au point d'opération, le modèle Norton de l'inductance non-linéaire avec la méthode TR s'obtient en prenant :

$$y_N = \frac{\Delta t}{2L_k^{jn}} \quad (2.31)$$

$$i_N = -\frac{1}{L_k^{jn}} \left[\varphi_t - \varphi_k^{jn} + \frac{\Delta t}{2} v_t \right] \quad (2.32)$$

Avec BE, la discréétisation de (2.28) donne :

$$\varphi_{t+\Delta t} = \Delta t \cdot v_{t+\Delta t} + \varphi_t \quad (2.33)$$

La combinaison des relations (2.27) et (2.33) donne :

$$i_{t+\Delta t} = \frac{\Delta t}{L_k^{jn}} v_{t+\Delta t} + \frac{1}{L_k^{jn}} \left[\varphi_t - \varphi_k^{jn} \right] \quad (2.34)$$

Ainsi, le modèle Norton de l'inductance non-linéaire avec la méthode BE s'obtient en prenant :

$$y_N = \frac{\Delta t}{L_k^{jn}} \quad (2.35)$$

$$i_N = -\frac{1}{L_k^{jn}} \left[\varphi_t - \varphi_k^{jn} \right] \quad (2.36)$$

On peut remarquer que lorsque l'intégration BE est réalisée avec un pas d'intégration égal à $\frac{\Delta t}{2}$,

la valeur de y_N est exactement la même que celle obtenue avec TR pour les modèles de tous les composants. Cette heureuse coïncidence autorise des transitions entre la méthode TR et la méthode BE sans modification de la matrice principale (qui sera présentée plus bas).

2.1.2.5 Modèle d'une résistance non-linéaire

Certains composants non-linéaires peuvent être modélisés par des résistances non-linéaires dont la caractéristique tension-courant a l'allure de la Figure 2.2. Ces dernières sont décrites par un ensemble d'équations linéaires définies chacune sur un intervalle donné. Ceci permet d'obtenir une représentation dite linéaire par segments. A chaque instant t de la simulation, le point d'opération (i_t, v_t) est situé sur un segment s décrit par l'équation :

$$i_t = y_s^{jn} v_t + i_s^{jn} \quad (2.37)$$

Dans cette équation, y_s^{jn} est la pente et i_s^{jn} est l'ordonnée à l'origine du segment s , situé entre les points frontières (i_j, v_j) et (i_n, v_n) . Le modèle Norton d'un tel composant ne dépend pas de la méthode d'intégration numérique (il en est de même des modèles de tous les composants décrits par des équations algébriques). Ce modèle s'obtient à tout instant t en prenant simplement:

$$y_N = y_s^{jn} \quad (2.38)$$

$$i_N = -i_s^{jn} \quad (2.39)$$

2.1.2.6 Modèle d'une ligne de transport monophasée

Il existe plusieurs modèles pour les lignes de transport. Le plus simple d'entre eux est le modèle CP (constant parameters) dans lequel une ligne monophasée sans pertes de longueur l , d'inductance et de capacité linéaires L' et C' respectivement est représentée comme suit [83]:

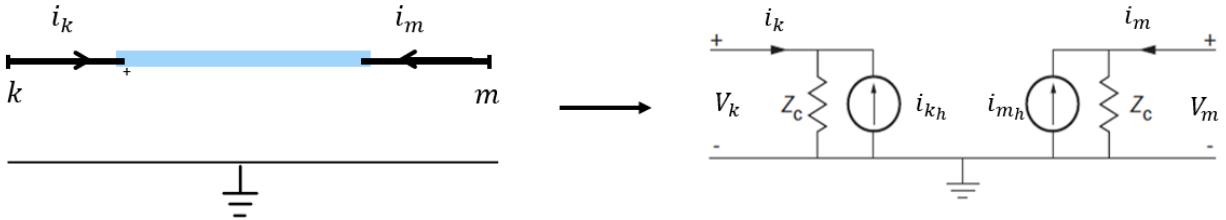


Figure 2.3: Circuit équivalent d'une ligne monophasée sans pertes

Cette figure est obtenue à partir des équations :

$$i_{k_h} = \frac{1}{Z_c} V_m(t - \tau) + i_m(t - \tau) \quad (2.40)$$

$$i_{m_h} = \frac{1}{Z_c} V_k(t - \tau) + i_k(t - \tau) \quad (2.41)$$

Avec $\tau = l\sqrt{L'C'}$ et $Z_c = \sqrt{\frac{L'}{C'}}$. La Figure 2.3 permet donc directement d'obtenir le modèle Norton

de la ligne monophasé. Les courants de ce modèle sont calculés à partir des valeurs antérieures (historiques) des courants et tensions. Ces valeurs doivent donc être stockées. Si τ n'est pas un multiple exact du pas d'intégration Δt , alors les valeurs à $t - \tau$ doivent être obtenues par

interpolation à partir des deux valeurs disponibles aux points de temps consécutifs situés de part et d'autre de $t - \tau$.

Avec le modèle CP, tout se passe donc comme si une modification au début de la ligne, à l'extrémité k prendrait un délai τ pour que ses effets soient perceptibles en bout de ligne, à l'extrémité m . Il y'a donc un découplage temporel entre le début et la fin de la ligne. Il est possible d'inclure les pertes à ce modèle ou alors de développer un équivalent triphasé. Les détails de calcul sont présentés dans [10] et [83].

Une fois que les modèles ont été créés pour tous les composants, les équations obtenues sont assemblées selon la topologie du réseau et en respectant les lois de Kirchhoff. Cette opération permet de constituer le système d'équations du réseau qui sera par la suite résolu.

2.2 La formulation et la résolution des équations

2.2.1 La formulation nodale et la formulation MANA

2.2.1.1 Description générale

A partir des circuits équivalents Norton de chaque composant, il est possible de constituer la matrice nodale [12]. Pour un réseau contenant n nœuds (numérotés de 1 à n), cette matrice est carrée d'ordre n et exprime les relations d'équilibre des courants à chaque nœud :

$$\mathbf{Y}_n \mathbf{v}_n = \mathbf{i}_n \quad (2.42)$$

La matrice \mathbf{Y}_n de taille $n \times n$ est la matrice d'admittance nodale, \mathbf{v}_n est le vecteur des tensions à chaque nœud (entre le nœud et la terre), et \mathbf{i}_n représente le vecteur des injections de courant à chaque nœud.

Soit un composant situé entre les nœuds k et m d'un circuit. Ce composant est modélisé par un équivalent Norton défini par y_N et i_N de la Figure 1.1. Sa participation au système d'équations nodales se fait par des affectations matricielles. Ainsi, son admittance y_N est ajoutée aux valeurs contenues dans les cellules $\mathbf{Y}_n(k,k)$ et de $\mathbf{Y}_n(m,m)$ tandis que la valeur opposée $-y_N$ est ajoutée aux valeurs des cellules $\mathbf{Y}_n(k,m)$ et $\mathbf{Y}_n(m,k)$. De plus, la source de courant i_N est ajoutée à la

valeur contenue dans la cellule $\mathbf{i}_n(k)$ tandis que son opposé $-i_N$ est ajouté à $\mathbf{i}_n(m)$. Il est à noter que si le nœud k représente la terre, alors $k=0$. Dans ce cas, seules les affectations matricielles impliquant $\mathbf{Y}_n(m,m)$ et $\mathbf{i}_n(m)$ seront effectuées. Dans le cas général, et pour tout composant contenant 2 ou plusieurs nœuds, lorsque l'un de ces nœuds représente la terre, les affectations matricielles qui font intervenir ce nœud sont tout simplement omises. Cette règle s'applique aussi pour tous les autres composants qui seront présentés plus bas.

La formulation nodale telle qu'exprimé par l'équation (2.42) ne prend en compte que les admittances et ne permet pas d'intégrer les sources de tension idéales, les transformateurs idéaux et de manière plus générale les fonctions de dépendance entre deux ou plusieurs nœuds et branches. Pour éliminer ces limitations, [5] propose une formulation modifiée et augmentée de (2.42) qui se présente comme suit :

$$\begin{bmatrix} \mathbf{Y}_n & \mathbf{V}_c & \mathbf{D}_c & \mathbf{S}_c \\ \mathbf{V}_r & \mathbf{V}_d & \mathbf{D}_{VD} & \mathbf{S}_{VS} \\ \mathbf{D}_r & \mathbf{D}_{DV} & \mathbf{D}_d & \mathbf{S}_{DS} \\ \mathbf{S}_r & \mathbf{S}_{SV} & \mathbf{S}_{SD} & \mathbf{S}_d \end{bmatrix} \begin{bmatrix} \mathbf{v}_n \\ \mathbf{i}_v \\ \mathbf{i}_D \\ \mathbf{i}_s \end{bmatrix} = \begin{bmatrix} \mathbf{i}_n \\ \mathbf{v}_b \\ \mathbf{d}_b \\ \mathbf{s}_b \end{bmatrix} \quad (2.43)$$

Cette formulation contient des blocs de sous-matrices permettant de représenter les relations de dépendance [16]. L'équation (2.43) peut être vue comme un système générique:

$$\mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t \quad (2.44)$$

Les différentes sous-matrices de (2.43) sont formulées à partir des composants du circuit simulé.

2.2.1.2 Participation des sources de tension idéales

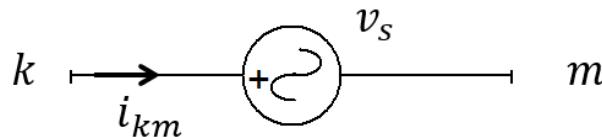


Figure 2.4: Schéma d'une source de tension idéale

Une source de tension v_s située entre les nœuds k et m est décrite par l'équation

$$v_k - v_m = v_s \quad (2.45)$$

Les équations de chaque source servent à construire les sous-matrices \mathbf{V}_r et \mathbf{v}_b de (2.43). En effet, lorsque (2.45) décrit la $q^{\text{ème}}$ source de tension, on a les affectations suivantes : $\mathbf{V}_r(q, k) = 1$, $\mathbf{V}_r(q, m) = -1$, $\mathbf{V}_c(k, q) = 1$, $\mathbf{V}_c(m, q) = -1$. Les deux dernières affectations permettent de prendre en compte les injections de courant de cette source aux nœuds k et m . La valeur de la source est considérée par l'affectation $\mathbf{v}_b(q) = v_s$. Le courant i_{km} qui circule de k vers m est une inconnue qui se retrouve dans la $q^{\text{ème}}$ ligne de \mathbf{i}_v . La matrice \mathbf{V}_d est carrée d'ordre égal au nombre total de sources de tensions idéales du réseau. Tous ses termes sont nuls, mais si la source est court-circuitée, alors ses termes diagonaux sont mis à 1 et $v_s = 0$.

2.2.1.3 Participation des sources de courant

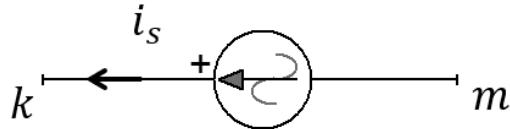


Figure 2.5: Source de courant

Une source de courant i_s située entre les nœuds k et m fournit un courant « entrant » dans le nœud k et « sortant » dans le nœud m . La contribution de cette source se fait par les deux affectations suivantes: $\mathbf{i}_n(k) = i_s$ et $\mathbf{i}_n(m) = -i_s$.

2.2.1.4 Participation des transformateurs idéaux

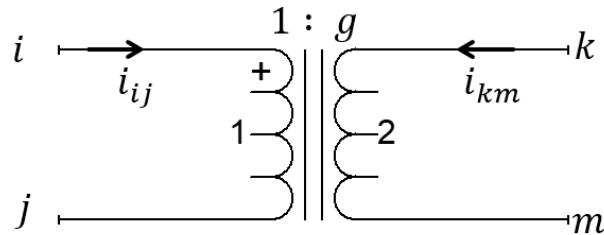


Figure 2.6: Schéma d'un transformateur idéal

Le transformateur idéal de la Figure 2.6 est décrit par l'équation :

$$v_k - v_m - gv_i + gv_j = 0 \quad (2.46)$$

Cette équation contribue dans la sous-matrice \mathbf{D}_r . Pour le $q^{\text{ème}}$ transformateur, on aura : $\mathbf{D}_r(q, k) = 1$, $\mathbf{D}_r(q, m) = -1$, $\mathbf{D}_r(q, i) = -g$ et $\mathbf{D}_r(q, j) = g$. On aura également $\mathbf{D}_c(k, q) = 1$, $\mathbf{D}_c(m, q) = -1$, $\mathbf{D}_c(i, q) = -g$ et $\mathbf{D}_c(j, q) = g$. Le courant i_{km} apparaît comme inconnue à la $q^{\text{ème}}$ ligne de \mathbf{i}_D .

2.2.1.5 Participation des interrupteurs idéaux

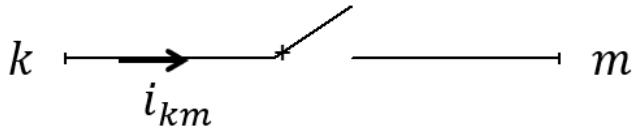


Figure 2.7: Schéma d'un interrupteur idéal

On assume que l'interrupteur idéal entre les nœuds k et m est le $q^{\text{ème}}$ interrupteur du réseau.

S'il est fermé, on a :

$$\nu_k - \nu_m = 0 \quad (2.47)$$

Cette équation contribue dans la sous-matrice \mathbf{S}_r via les affectations suivantes : $\mathbf{S}_r(q, k) = 1$ et $\mathbf{S}_r(q, m) = -1$. De plus, $\mathbf{S}_c(k, q) = 1$ et $\mathbf{S}_c(m, q) = -1$. La matrice \mathbf{S}_d est carrée d'ordre égal au nombre d'interrupteurs idéaux et à valeurs toutes nulles.

Si au contraire l'interrupteur est ouvert, on a :

$$i_{km} = 0 \quad (2.48)$$

Dans ce cas, $\mathbf{S}_r(q, k) = 0$ et $\mathbf{S}_r(q, m) = 0$. De plus, $\mathbf{S}_c(k, q) = 0$, $\mathbf{S}_c(m, q) = 0$ et on a $\mathbf{S}_d(q, q) = 1$.

Que l'interrupteur idéal soit ouvert ou fermé, $s_b(q) = 0$ et le courant i_{km} est l'inconnue situé à la $q^{\text{ème}}$ ligne de \mathbf{i}_s .

En procédant d'une façon similaire, il est possible d'écrire des équations pour tous les composants d'un réseau et de les intégrer au système général (2.43).

2.2.1.6 Exemple démonstratif

Le circuit de la Figure 2.8 est utilisé pour illustrer la formulation MANA. Il contient une source de tension v_s , une source de courant i_s , une inductance L , un condensateur C , deux résistances linéaires R et R_g , deux interrupteurs idéaux S_1 et S_2 , un transformateur idéal de ratio $1:g$ et une résistance non-linéaire pour laquelle on assumera que l'admittance et la source du courant du modèle Norton valent respectivement y_N and i_N (obtenues par (2.38) et (2.39)). L'interrupteur S_1 est ouvert tandis que S_2 est fermé.

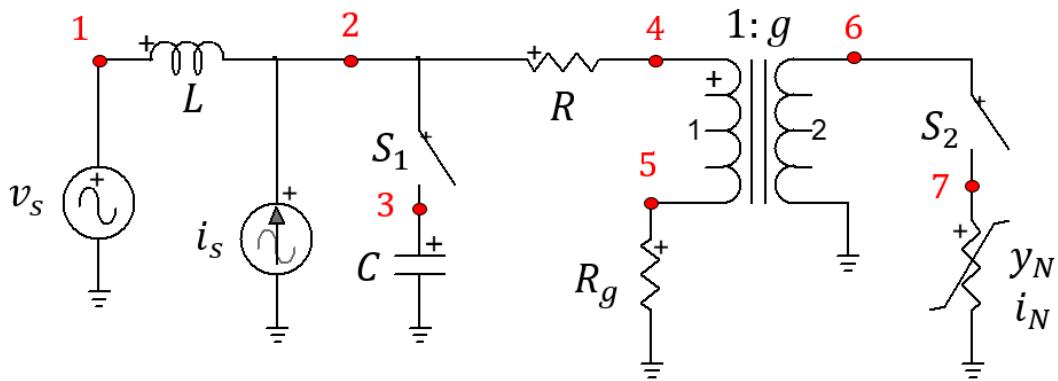


Figure 2.8: Circuit de démonstration de la formulation MANA

L'application de l'approche MANA à la formulation des équations de ce circuit, en considérant la méthode trapézoidale avec un pas d'intégration numérique Δt donne le résultat suivant :

$$\mathbf{A}_t = \begin{bmatrix} \frac{\Delta t}{2L} & -\frac{\Delta t}{2L} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{\Delta t}{2L} & \frac{\Delta t}{2L} + \frac{1}{R} & 0 & -\frac{1}{R} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2C}{\Delta t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{R} & 0 & \frac{1}{R} & 0 & 0 & 0 & 0 & -g & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{R_g} & 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_N & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g & g & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.49)$$

$$\mathbf{x}_t = [v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7 \ i_{vs} \ i_D \ i_{s1} \ i_{s2}]^T \quad (2.50)$$

$$\mathbf{b}_t = [i_{hL} \ i_s - i_{hL} \ i_{hC} \ 0 \ 0 \ 0 \ i_N \ v_s \ 0 \ 0 \ 0]^T \quad (2.51)$$

Les valeurs v_1, \dots, v_7 représentent les tensions par rapport à la terre des différents nœuds dont les numéros sont indiqués à la Figure 2.8; i_{vs} est le courant dans la source de tension, circulant du nœud 1 vers la terre, i_D est le courant dans le transformateur, du nœud 6 vers la terre et i_{s1} et i_{s2} sont les courants dans les interrupteurs S_1 et S_2 respectivement. Les valeurs i_{hL} et i_{hC} sont calculées à partir des relations (2.16) et (2.23) respectivement.

2.2.2 La résolution des équations

2.2.2.1 Approche générale

Après la numérotation des nœuds d'un circuit, le choix du pas d'intégration numérique Δt , la détermination des paramètres des modèles Norton des composants et la formulation des équations, la simulation proprement dite peut commencer.

La simulation dans le domaine du temps pour une durée totale de t_{sim} consiste à résoudre le système (2.44) à l'instant Δt , puis à tous les instants multiples consécutifs de Δt jusqu'à l'atteinte de t_{sim} .

Pendant la résolution, le vecteur \mathbf{b}_t est modifié continuellement à cause des changements de valeur des sources et à cause de la mise à jour des historiques de courants dans les modèles des composants du réseau. Pour les composants dont le modèle est construit à partir de la méthode d'intégration numérique, les valeurs des courants qui les traversent et des tensions à leurs bornes au temps $t = 0$ permettent de calculer les historiques de courants et constituer le vecteur \mathbf{b}_t au premier instant de simulation $t = \Delta t$. Les valeurs à $t = 0$ sont obtenues par initialisation [84] (à la suite d'un écoulement de puissance ou d'un calcul en régime établi). En l'absence d'initialisation, il est d'usage d'assumer que ces valeurs sont toutes nulles. Cette supposition peut dans certains cas créer une erreur initiale dans la simulation. En général, cette erreur s'amorti et disparaît après quelques points de simulation, étant donné que la méthode d'intégration numérique est stable.

La matrice \mathbf{A}_t peut varier durant les simulations. Des techniques de factorisation [38] permettent de l'écrire comme produit de deux matrices \mathbf{L} et \mathbf{U} afin de résoudre le système (2.44) de façon efficiente sans avoir besoin de calculer l'inverse de \mathbf{A}_t .

2.2.2.2 Apparition et traitement des discontinuités

Il peut arriver que certains termes de \mathbf{A}_t subissent des modifications à cause du changement d'état des interrupteurs, du changement du pas d'intégration numérique (si applicable) ou encore du changement de la valeur de l'admittance du modèle d'un ou de plusieurs composant(s) non-linéaire. Dans de pareils cas, on dit qu'il y'a discontinuité. La refactorisation de la matrice \mathbf{A}_t est nécessaire, ce qui ralenti la simulation. De plus, la méthode TR n'est pas L – stable. Aux points de temps consécutifs à l'instant d'apparition de la discontinuité, certaines variables peuvent prendre des valeurs erronées qui oscillent autour d'une valeur moyenne. Les formes d'onde de ces variables présentent donc des oscillations numériques qui faussent complètement les résultats. Pour éviter cette situation, il suffit d'utiliser la méthode BE pour la première intégration numérique à la suite d'une discontinuité. Si le pas d'intégration choisi pour appliquer la méthode BE est égal à $\Delta t/2$, la matrice de l'intégration BE est identique à \mathbf{A}_t . De ce fait, il suffit d'appliquer deux intégrations BE pour se retrouver à nouveau sur le time-mesh initial. Après cela, la méthode d'intégration peut

repasser à TR pour la suite de la simulation. Cette façon de passer de l'intégration TR avec un pas Δt à 2 intégrations BE avec un pas de $\Delta t/2$, puis de revenir à l'intégration TR avec un pas Δt a été présentée dans [81] et porte le nom de Critical Damping Adjustment (CDA). Elle est inspirée de l'approche initiée dans [85] qui consiste à exécuter une seule intégration BE pour réinitialiser la solution au point de discontinuité. La précision d'une telle approche n'est garantie que lorsque le pas d'intégration est très petit, raison pour laquelle l'approche a été abandonnée au profit de l'approche alternative CDA. La méthode obtenue à partir de CDA sera désignée par TR_BE dans la suite de cette thèse.

2.2.2.3 Traitement des non-linéarités

L'admittance y_N du circuit équivalent (modèle Norton) d'un composant non-linéaire dépend de son point d'opération. Par conséquence, lorsque la simulation évolue d'un point de temps à un autre, le point d'opération peut se déplacer significativement de manière à changer la valeur de y_N . La solution calculée en un point de temps ne peut être correcte que si l'admittance de chaque composant non-linéaire qui se trouve dans \mathbf{Y}_n est égale à l'admittance correspondante au point d'opération de ce composant à cet instant de la simulation. Pour garantir cela, il est parfois nécessaire d'effectuer des itérations. Les itérations sont des résolutions successives au même point de temps, du système (2.44). La première itération se fait avec une valeur de l'admittance basée sur un calcul en régime permanent ou sur la valeur finale obtenue au point de temps précédent. Cette valeur représente la prédition initiale. Après chaque itération, la valeur de l'admittance de chaque composant non-linéaire est corrigée jusqu'à ce qu'elle ne varie plus d'une itération à l'autre. A ce moment, on dit qu'il y'a convergence. La méthode utilisée pour déterminer la prochaine valeur de l'admittance à la suite de chaque itération influence le nombre total d'itérations nécessaires pour converger. La méthode Newton est la plus utilisée à cette fin. Il est important de préciser que pour des simulations en temps réel, les logiciels ne disposent pas d'assez de temps pour faire les itérations nécessaires. Il existe pour ces domaines d'application, des méthodes itératives rapides comme celle présentée dans [86].

La démarche adoptée dans le processus d'itérations peut créer des erreurs d'arrondi numérique et dégrader de manière significative la précision des résultats. Pour comprendre cela, on va considérer un composant non-linéaire dont l'admittance y_{N1} contribue au calcul de $\mathbf{Y}_n(i, j)$. Si cette

admittance doit être modifiée pour prendre la valeur y_{N2} , la façon évidente de réaliser cela consiste à affecter dans $\mathbf{Y}_n(i, j)$ le résultat obtenu en faisant $\mathbf{Y}_n(i, j) - y_{N1} + y_{N2}$. A l'échelle de l'ordinateur, chaque opération élémentaire qui est exécutée pour calculer $\mathbf{Y}_n(i, j)$ crée une erreur d'arrondi numérique. A cause des multiples opérations élémentaires qui vont être exécutées (retrancher et rajouter des valeurs), les erreurs numériques d'arrondi sur le calcul de $\mathbf{Y}_n(i, j)$ vont progressivement s'accumuler au fil des itérations et générer une erreur importante qui va impacter sur la précision des résultats. Pour pallier cela, il faut remplacer \mathbf{Y}_n par deux matrices. La première désignée \mathbf{Y}_{linear} (de même taille que \mathbf{Y}_n) ne va contenir que les admittances des composants linéaires du réseau. Elle sera donc constante tout au long de la simulation. La deuxième matrice désignée $\mathbf{Y}_{nonlinear}$ (de la même taille que \mathbf{Y}_n) ne contiendra que les valeurs d'admittances des composants non-linéaires. Chaque modification d'une admittance entraînera tout simplement le remplacement de la valeur correspondante dans $\mathbf{Y}_{nonlinear}$. Après cela, l'opération $\mathbf{Y}_{linear} + \mathbf{Y}_{nonlinear} = \mathbf{Y}_n$ est effectuée pour obtenir la matrice \mathbf{Y}_n qui est utilisée pour la résolution du système d'équations (2.44).

Après cette description de l'approche commune de modélisation, de formulation des équation et de résolution, il est à présent possible d'envisager de nouvelles façons de procéder qui pourraient conduire à des améliorations de la précision et de la rapidité dans les simulations.

CHAPITRE 3 SIMULATION À PAS D'INTÉGRATION NUMÉRIQUE VARIABLE

Ce chapitre démontre l'efficacité de l'utilisation d'un pas d'intégration numérique adaptatif ou variable pour la simulation des circuits électriques. Les avantages de cette approche, les défis liés à son implémentation ainsi que l'adaptation des équations mathématiques sont présentés. Deux algorithmes présentés par l'auteur cette thèse dans [87] et spécialement conçus pour la simulation des transitoires de grands réseaux et des circuits d'électronique de puissance sont proposés.

3.1 Description générale

3.1.1 Les avantages de l'utilisation d'un pas d'intégration variable

Lors d'une simulation dans le domaine du temps, le pas d'intégration numérique Δt est généralement fixe et défini par l'utilisateur au vu de sa connaissance à priori du circuit ainsi que du type de variations (rapides ou lentes) des grandeurs calculées. Des variations rapides nécessitent un pas d'intégration relativement petit pour être représentées le plus fidèlement possible tandis que des variations lentes peuvent être représentées avec une bonne précision, que le pas d'intégration soit petit ou relativement grand. Un pas d'intégration petit rallonge la durée totale de la simulation, tandis qu'un pas plus grand permet de compléter la simulation en un temps plus court. En effet, dans le premier cas, les points de temps de la simulation sont plus nombreux, ce qui demande plus d'effort de calcul de la part du logiciel. La sélection du pas d'intégration à utiliser résulte donc d'un compromis entre la rapidité de la simulation et le niveau de précision souhaité.

La dynamique d'un réseau peut devenir parfois très rapide, parfois lente. Par exemple, la simulation des transitoires d'un réseau électrique peut nécessiter de représenter des formes d'onde au contenu fréquentiel très varié [83]. Avec un pas d'intégration fixe, l'utilisateur n'a pas d'autre choix que de le choisir suffisamment petit pour simuler avec précision les variations rapides. Ce pas est ensuite conservé pour le reste de la simulation, même lorsque les variations deviennent beaucoup plus lentes. Ceci représente évidemment une perte en efficacité qui peut devenir significativement élevée. L'utilisation dans une simulation d'un pas d'intégration adaptatif, capable de prendre une valeur petite pour simuler les transitoires rapides et une valeur relativement élevée pour simuler les variations plus lentes offrirait un compromis idéal entre précision et rapidité.

D'un autre côté, le développement de convertisseurs à haute fréquence de commutation dans les circuits d'électronique de puissance requiert l'utilisation de pas d'intégration très petits pour la représentation exactes des formes d'onde. Ceci est principalement dû à la nécessité de détecter les discontinuités au moment précis où elles se produisent, mais aussi de réduire l'erreur qui se crée à la traversée de chaque discontinuité. Un pas d'intégration grand engendre des erreurs plus significatives que lorsque le pas est petit. Un pas d'intégration capable de devenir très petit pour traverser les discontinuités et redevenir grand en l'absence de discontinuité serait un excellent moyen d'accélérer la vitesse d'exécution des simulations des circuits d'électronique de puissance et de conserver la précision des résultats.

3.1.2 Les défis liés à l'utilisation d'un pas d'intégration numérique variable

Bien que la possibilité d'utiliser un pas d'intégration numérique variable soit théoriquement intéressante, son implémentation pratique se heurte à des défis importants qui doivent être convenablement adressés au risque de conduire à une perte totale de l'efficacité de cette approche, ou même de créer d'autres problèmes dans la simulation.

Le premier défi est celui de la refactorisation de la matrice du système. En effet, utiliser un Δt variable nécessite de recalculer les admittances des modèles de tous les composants décrits par une équation différentielle. Ces admittances sont déterminées par la méthode d'intégration numérique et dépendent de Δt . Lorsque Δt change, la matrice principale \mathbf{A}_t du circuit simulé est modifiée. Elle doit de ce fait être refactorisée à nouveau pour permettre la résolution du système d'équations du réseau (le système (2.44)). La refactorisation est une étape qui consomme du temps et ralentit la simulation. Il est donc important de s'assurer que le ralentissement causé par les multiples factorisations de la matrice tout au long de la simulation soit suffisamment compensé par le gain en temps de calcul engendré par l'algorithme utilisant un pas d'intégration adaptatif. Une façon de faire cela consiste à n'utiliser que des valeurs de Δt connues à l'avance. Ceci permet de précalculer et pré-factoriser (avant le début de la simulation) toutes les possibles matrices \mathbf{A}_t correspondantes aux différents pas d'intégration. L'efficacité de cette approche est toutefois compromise lorsque le circuit contient des composants non-linéaires car ceux-ci contribuent également à modifier \mathbf{A}_t .

Le deuxième défi est celui du traitement de certains composants particuliers. Par exemple, si le réseau simulé contient des lignes de transport représentées par le modèle à paramètres constants (CP), la prise en compte du délai de propagation τ pose un problème pour le calcul des courants historiques de lignes. En effet, pour le calcul à l'instant t , les valeurs requises pour le calcul des courants historiques des lignes sont obtenues à $t - \tau$ par interpolation. La variation du pas d'intégration peut engendrer des problèmes de précision des valeurs interpolées. Pour s'en prémunir, une solution consiste à utiliser des pas d'intégration multiples les uns des autres, puis de stocker les valeurs historiques de courants et tensions temporellement espacées du plus petit pas d'intégration. Par exemple, à la suite d'une intégration avec un pas 10 fois plus élevé que le plus petit pas, il faudra calculer (par interpolation) les 8 valeurs intermédiaires entre les deux points de temps consécutifs et les stocker. Aussi, il est important de préciser que le pas d'intégration lorsqu'il varie doit rester en tout temps inférieur au délai de propagation de la ligne la plus courte du réseau simulé. Cette contrainte complexifie l'utilisation du pas d'intégration variable.

Le troisième défi est celui du choix du pas d'intégration. En effet, modifier Δt permet d'améliorer la précision, mais à la condition d'utiliser un pas convenablement sélectionné. Puisqu'il n'existe pas de méthodologie rigoureuse de sélection du pas, la valeur retenue peut ne pas être convenable. Pour que l'approche de simulation à pas d'intégration variable produise les résultats avec la précision souhaitée, il faut s'assurer que les pas utilisés selon le moment de la simulation soient suffisamment petits.

Le quatrième défi est celui de l'identification du moment auquel le pas d'intégration doit être modifié. En effet, savoir quand changer de pas impacte significativement les performances de la simulation. Il faut autant de possible éviter d'utiliser un pas grand lorsque les variations dans le réseau sont rapides et requièrent un pas petit pour des besoins de précision. De même, il faut éviter de conserver un pas petit lorsque les variations deviennent plus lentes et qu'un pas plus grand permettrait de maintenir le niveau de précision souhaité tout en accélérant la simulation.

Le cinquième défi concerne la stabilité et la convergence. La valeur du pas d'intégration numérique peut affecter la stabilité de la méthode d'intégration numérique lorsque celle-ci n'est pas A – stable. Il est donc important d'en tenir compte dans toute stratégie de changement du pas d'intégration. Également, lors de la simulation de circuits contenant des composants non-linéaires, la vitesse de convergence du processus itératif exécuté à chaque point de temps peut être affecté par le pas

d'intégration choisi. C'est pour cette raison par exemple que la stratégie de variation du pas d'intégration appliquée dans le logiciel SPICE [49] prend en compte la précision souhaitée, mais aussi la vitesse de convergence à chaque point de temps.

Il ressort de ce qui précède qu'implémenter un algorithme utilisant un pas d'intégration variable nécessite de prendre en compte un grand nombre de facteurs qui pourraient influencer la performance des résultats. Il est temps à présent d'examiner le mode opératoire pour l'implémentation pratique de cette approche.

3.1.3 Mode opératoire d'une simulation à pas variable

Chaque changement du pas d'intégration numérique Δt durant une simulation entraîne des modifications dans le système d'équations. Ces modifications sont dues aux changements dans les valeurs des modèles de certains composants. Par exemple, pour une inductance, on peut voir à la Figure 2.1 que l'admittance y_N et la source de courant i_N de son circuit équivalent dépendent du pas d'intégration. Ces valeurs contribuent respectivement dans \mathbf{A}_t et dans \mathbf{b}_t de (2.44). Plus spécifiquement, elles contribuent dans \mathbf{Y}_n et \mathbf{i}_n de (2.43). De manière plus générale, tous les composants dont les modèles sont construits à partir d'une méthode d'intégration numérique voient leurs paramètres modifiés chaque fois que le pas d'intégration de la simulation change. Modifier le pas d'intégration demande donc de faire certains ajustements dans les équations. Pratiquement, si à l'instant t , le pas d'intégration passe de Δt à Δt_{new} , alors les actions suivantes doivent être exécutées pour le calcul de la solution au prochain point de temps $t + \Delta t_{new}$:

1. Recalculer les admittances de tous les composants impactés. Par exemple, l'admittance d'une inductance est recalculée par (2.15) en remplaçant Δt par Δt_{new} ;
2. Reformuler la sous-matrice \mathbf{Y}_n à partir des admittances calculées à l'étape ci-dessus;
3. Formuler \mathbf{i}_n de (2.43) en utilisant Δt_{new} . Par exemple, l'injection de courant (courant historique) d'une inductance se calculera avec la relation (2.16) dans laquelle Δt est remplacée par Δt_{new} . Il est important de préciser que dans la relation (2.16), les valeurs de

v_{km_t} et de i_{km_t} ne sont pas affectées par le changement de pas d'intégration, puisqu'elles correspondent aux solutions déjà calculées à l'instant précédent t ;

4. Reformuler le système (2.43) à partir des nouvelles matrices \mathbf{Y}_n et \mathbf{i}_n calculées ci-dessus.
Il est important de relever que les relations de dépendance introduites par les transformateurs et les interrupteurs sont indépendantes du changement du pas d'intégration. C'est pourquoi les sous-matrices traduisant ces dépendances restent intactes;
5. Résoudre le système (2.44) (qui est simplement la forme compacte de (2.43)) afin d'obtenir le vecteur solution \mathbf{x} à l'instant $t + \Delta t_{new}$.

Le pas d'intégration Δt_{new} peut être maintenu pour les calculs aux points de temps suivants ou alors être modifié de nouveau. Dans cette dernière éventualité, toutes les 5 étapes suscitées doivent être re-exécutées en considérant la nouvelle valeur de Δt . Il est à noter que si le réseau simulé ne contient que des composants linéaires et que les valeurs des différents pas d'intégration sont connues à l'avance, il est possible de précalculer les différentes variantes de \mathbf{Y}_n dans \mathbf{A}_t . Dans ce cas, l'étape 2 ci-dessus consistera donc simplement à sélectionner la matrice \mathbf{Y}_n correspondant au pas d'intégration sélectionné.

Une fois que la démarche opérationnelle du changement de pas d'intégration durant la simulation est connue, il faut définir une stratégie efficace de variation du pas qui permettra de bâtir un algorithme. Cette stratégie doit adresser une partie ou la totalité des enjeux liés à la simulation à pas variable. La stratégie à adopter dépendra également des problématiques inhérentes aux types de circuits auxquels elle s'applique. Les prochaines parties de ce chapitre présentent des stratégies à partir desquelles sont construites de nouveaux algorithmes de simulation à pas variable.

3.2 Algorithme utilisant un pas d'intégration numérique variable pour la simulation des transitoires des grands réseaux

3.2.1 Les fondements de l'algorithme proposé

Pour simuler les transitoires de grands réseaux, on peut développer un algorithme utilisant trois pas d'intégration Δt_1 , Δt_2 et Δt_3 classés dans cet ordre du plus petit au plus grand et sélectionnés par l'utilisateur. Le plus petit (Δt_1) est sélectionné pour simuler les transitoires les plus rapides qui peuvent apparaître sur le réseau tandis que le plus grand pas (Δt_3) est choisi pour simuler le régime permanent ou quasi permanent. Le pas Δt_2 est un pas intermédiaire. Dans l'algorithme, le plus petit pas Δt_1 est sélectionné chaque fois qu'une discontinuité (manœuvre de disjoncteur, etc...) se produit. En effet, dans la simulation des grands réseaux, les discontinuités engendrent des perturbations qui sont à l'origine de l'apparition des phénomènes transitoires. Utiliser le plus petit pas d'intégration à la suite de chaque discontinuité permet de représenter les transitoires qui apparaissent avec la meilleure précision. Par la suite, le pas d'intégration est progressivement augmenté et passe de Δt_1 à Δt_2 , puis de Δt_2 à Δt_3 dépendamment de la précision des résultats de la simulation.

Il est nécessaire de pouvoir évaluer la précision tout au long de la simulation afin de décider du moment de changement du pas d'intégration. L'erreur de troncature locale (*LTE*) fournit un moyen adéquat pour arriver à cette fin. Il s'agit de l'erreur numérique commise dans le calcul d'une variable d'état à un instant donné sous l'hypothèse que toutes les valeurs aux instants précédents sont exactes [49], [78]. Cette erreur est causée exclusivement par la méthode d'intégration numérique et ne dépend pas des erreurs d'arrondis numériques liées à l'ordinateur qui fait les calculs. Dans une simulation, lors du passage d'un instant t quelconque à l'instant suivant $t + \Delta t$, l'erreur de troncature résultante de l'intégration trapézoidale pour le calcul d'une variable x est donné par :

$$LTE_{t+\Delta t}(x) = -\frac{\Delta t^3}{12} x^{(3)}(\tau) \quad (3.1)$$

Avec $t < \tau < t + \Delta t$. Cette relation montre que l'erreur dépend du pas d'intégration et de la vitesse (taux) de variation de x . Le calcul de LTE requiert l'évaluation de la dérivée troisième de x . Cela est fait grâce à la formule des différences divisées de Newton (Newton divided differences) [49]. Cette formule permet d'évaluer la dérivée troisième d'une variable x en un instant t_{n+1} en utilisant la valeur à cet instant, désignée x_{n+1} , ainsi que les valeurs de la variable aux trois instants précédents que l'on désignera par x_n , x_{n-1} , x_{n-2} . La formule de calcul fait également intervenir les pas d'intégration entre les différents points de temps. En désignant par h_{n+1} la valeur numérique du pas d'intégration permettant de passer de x_n à x_{n+1} (soit $h_{n+1} = t_{n+1} - t_n$) et en posant $\langle h_{n+1} \rangle = \frac{h_{n+1} + h_n}{2}$, la formule des différences divisées de Newton s'exprime comme suit :

$$\frac{d^3x}{dt^3} = \frac{\frac{x_{n+1} - x_n}{h_{n+1}} - \frac{x_n - x_{n-1}}{h_n} - \frac{x_{n-1} - x_{n-2}}{h_{n-1}}}{\langle h_{n+1} \rangle - \langle h_n \rangle} \quad (3.2)$$

En résumé, cette formule commence par calculer le taux de variation de x à trois positions consécutives. On obtient donc trois dérivées premières. Ensuite, la formule évalue la variation des trois variations, ce qui permet d'obtenir deux dérivées secondes de x . Finalement, la variation entre ces deux dérivées seconde est calculée et le résultat représente la dérivée troisième. L'équation (3.2) est utilisée dans (3.1) pour évaluer $LTE(x)$. Toutefois, pour utiliser LTE comme outil d'appréciation de l'erreur dans l'algorithme proposé, il faut comprendre son comportement.

3.2.2 Comportement de l'erreur de troncature

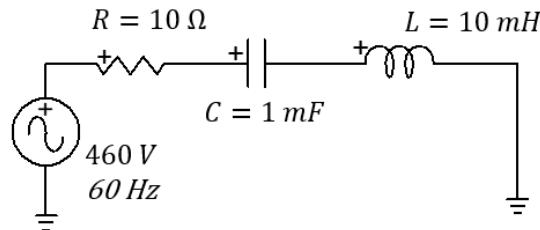


Figure 3.1: Circuit simple RLC série

La Figure 3.1 est un circuit RLC série simple qui va permettre d'étudier le comportement de l'erreur de troncature locale. Ce circuit contient deux variables d'état : la tension du condensateur C nommé v_C et le courant dans l'inductance L , désigné i_L . Le circuit est simulé avec un pas d'intégration $\Delta t = 50\mu s$ et les variables évoluent en régime permanent. Les erreurs de troncature calculées par (3.1) pour chaque variable d'état à chaque point de temps sont représentées à la Figure 3.2. Cette figure révèle que pour chaque variable, l'erreur de troncature n'est pas constante tout au long de la simulation, mais elle admet toutefois une valeur maximale. Aussi, on remarque que les erreurs ont des comportements similaires et des valeurs maximales identiques pour les deux variables. Ceci est tout à fait normal. En effet, v_C et i_L varient de façon sinusoïdale. Les dérivées de ces variables changent donc continuellement, tout en restant périodiques, ce qui justifie les formes d'onde observées. Si l'on refait la simulation avec un pas d'intégration plus grand, on peut constater que les courbes conservent la même allure, mais admettent des valeurs maximales plus élevées.

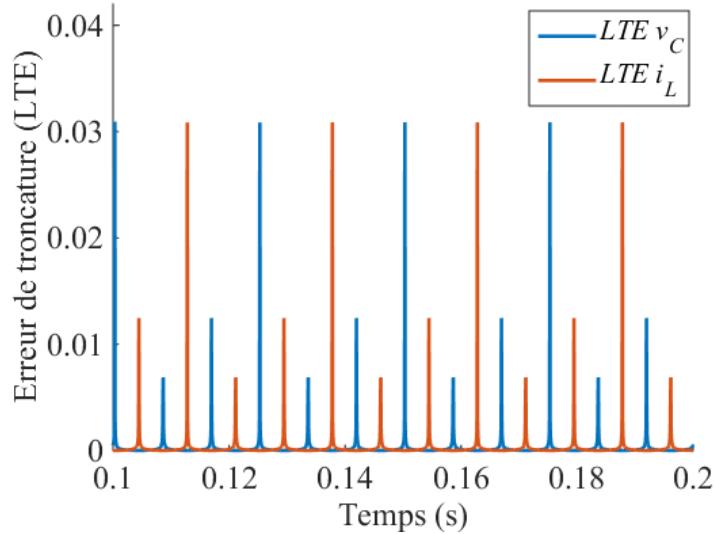


Figure 3.2: LTE sur la tension du condensateur et sur le courant de l'inductance

Cette analyse du comportement de l'erreur de troncature locale fournit des informations qui vont contribuer à bâtir l'algorithme proposé.

3.2.3 Application du comportement de LTE à l'algorithme à pas variable

Dans le cas le plus général, pour un circuit contenant N variables d'état, un total de N erreurs de troncature peuvent être évaluées à chaque instant de simulation. Chacune des N erreurs est associée

à une variable x_i pour $i \in \{1, 2, \dots, N\}$. Si on assume que les erreurs de toutes les variables ont la même valeur maximale, alors à la suite du calcul de la solution à l'instant $t + \Delta t$, on peut simplement calculer l'erreur de troncature d'une seule variable choisie de façon arbitraire. On peut aussi pour plus de rigueur calculer les erreurs de toutes les variables et retenir la valeur la plus élevée en valeur absolue. Cette erreur désignée $LTE_{t+\Delta t}$ est représentative du niveau de précision de la simulation à l'instant $t + \Delta t$. Elle s'obtient donc par la relation :

$$LTE_{t+\Delta t} = \max \{ |LTE_{t+\Delta t}(x_i)| \} \quad (3.3)$$

De plus, puisque la valeur numérique de l'erreur de troncature varie d'un point de temps à un autre, il serait plus prudent d'observer son comportement sur un intervalle de temps avant de conclure sur le niveau de précision de la simulation. Pour cette raison, il est nécessaire de définir un intervalle de temps d'observation de l'erreur. Une façon de faire cela consiste à créer un paramètre n_{count} défini par l'utilisateur. Ce paramètre représente le nombre de points de temps consécutifs qui seront parcourus par la simulation avant que l'erreur ne soit considérée pour quantifier la précision. Cela est équivalent à observer le comportement de l'erreur sur un intervalle de temps égal à $\Delta t \cdot n_{count}$. Donc, si le pas d'intégration a été baissé à l'instant $t + \Delta t$, il est logique d'utiliser la valeur maximale de l'erreur de troncature sur les n_{count} prochains instants de simulation pour quantifier la précision. Cette erreur désignée LTE_{max} s'obtient par :

$$LTE_{max} = \max \{ LTE_{t+k \cdot \Delta t} \} \quad (3.4)$$

avec $k \in \{1, 2, \dots, n_{count}\}$. Finalement, c'est la valeur de LTE_{max} qui servira de critère d'évaluation du niveau de précision dans la simulation pour décider si le pas d'intégration peut être augmenté ou pas. Il ne reste plus qu'à se donner un seuil de précision. Il s'agit d'un niveau d'erreur (ou tolérance) défini par l'utilisateur qui sera tel que si LTE_{max} est inférieure à ce seuil, on peut considérer que la simulation est très précise et le pas d'intégration peut être augmenté. Dans l'algorithme proposé, deux seuils de tolérance Tol_1 et Tol_2 seront définis pour permettre la transition de Δt_1 à Δt_2 , puis de Δt_2 à Δt_3 . On peut prendre $Tol_1 = Tol_2$. Toutefois, Δt_3 est supposé

être utilisé en régime permanent. De ce fait, la transition de Δt_2 à Δt_3 assume que le système simulé ait atteint le régime permanent ou quasi permanent. Si la transition se fait avant que le régime permanent ne soit réellement atteint, cela peut entraîner des erreurs significativement élevées. Afin de d'assurer donc que la transition ne se fasse pas trop tôt, il est préférable d'avoir $Tol_2 < Tol_1$. Lorsque le pas d'intégration prend la valeur Δt_3 , cette dernière est conservée jusqu'à ce qu'une nouvelle discontinuité se produise.

Il est à présent possible à la lumière des observations faites ci-dessus de formuler le mode opératoire ou la séquence d'exécution des évènements dans l'algorithme proposé.

3.2.4 Séquence des évènements de l'algorithme proposé

Avant le début de la simulation, l'utilisateur définit 3 catégories de paramètres :

- Les 3 valeurs de pas d'intégration qui seront utilisées pendant la simulation. Il s'agit de Δt_1 , Δt_2 et Δt_3 rangés dans cet ordre du plus petit au plus grand.
- Les 2 limites de tolérances Tol_1 et Tol_2 qui vont servir de seuils pour le passage de Δt_1 à Δt_2 , et de Δt_2 à Δt_3 respectivement.
- Le nombre de points de simulation n_{count} pendant lesquels l'erreur de troncature va être observée avant de prendre la décision d'augmenter le pas d'intégration ou pas.

Une fois que les paramètres susmentionnés aient été définis, le fonctionnement de l'algorithme est le suivant :

1. La simulation est normalement exécutée en régime permanent avec le plus grand pas Δt_3 et la méthode TR est utilisée pour l'intégration numérique. Au début de la simulation ou à tout instant t_E où se produit un évènement (manœuvre d'un disjoncteur, apparition d'un défaut, etc...), l'algorithme passe à l'étape 2.
2. Le pas d'intégration est réduit à sa valeur minimale Δt_1 .

3. Deux intégrations BE sont exécutées avec le pas $\Delta t_1/2$ (pour éviter les oscillations numériques). La simulation se trouve à présent au point $t_E + \Delta t_1$.
4. La simulation se poursuit avec la méthode TR et en utilisant Δt_1 . A chaque point de temps t (avec $t > t_E + \Delta t_1$), l'erreur de troncature est calculée pour chaque variable par la relation (3.1). Ensuite, la valeur LTE_t est obtenue par (3.3) et enregistrée.
5. Après n_{count} points de simulation, la valeur de LTE_{max} sur les n_{count} points est déterminée via (3.4) et sa valeur est comparée à Tol_1 . Si $LTE_{max} > Tol_1$, le pas d'intégration Δt_1 est maintenu et l'algorithme retourne à l'étape 4. Sinon, le pas d'intégration est augmenté et passe à Δt_2 , puis l'algorithme passe à l'étape 6.
6. La simulation se poursuit avec Δt_2 et une fois de plus, la valeur de LTE_t à chaque point de temps est calculée et enregistrée, comme à l'étape 4.
7. Après n_{count} points de simulation, la valeur de LTE_{max} sur ces n_{count} points est une fois de plus déterminée (comme à l'étape 5). Sa valeur est comparée à Tol_2 . Si $LTE_{max} > Tol_2$, le pas d'intégration Δt_2 est maintenu et l'algorithme retourne à l'étape 6. Sinon, le pas d'intégration est augmenté et passe à Δt_3 , puis l'algorithme se poursuit à l'étape 8.
8. La simulation se poursuit avec Δt_3 jusqu'à ce qu'un nouvel évènement se produit.

Il est important de préciser que chaque fois qu'un évènement se produit pendant la simulation, l'algorithme retourne à l'étape 2, qu'importe l'étape à laquelle il se trouve.

3.2.5 Application à la simulation d'un réseau, analyses et conclusions

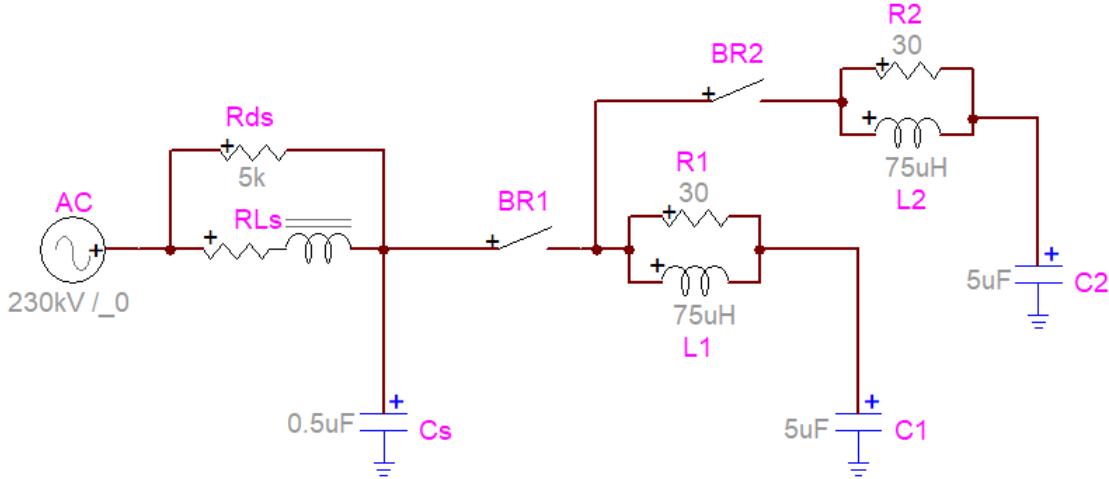


Figure 3.3: Exemple de circuit siège de transitoires électromagnétiques

L'algorithme proposé est testé sur le circuit de la Figure 3.3 afin de simuler un cas de transitoires de manœuvre dans les grands réseaux. La tension de la source est une tension efficace ligne-ligne. Le circuit est simulé pendant 400ms . Le disjoncteur BR1 ferme à $t = 20\text{ms}$, ouvre à 125ms et ensuite referme à 200ms . Le disjoncteur BR2 est initialement ouvert et ferme à 225ms . Toutes ces manœuvres font apparaître des fréquences transitoires qui varient de 340 Hz à 27 kHz , ce qui représente une plage assez large. Dans un algorithme de simulation utilisant un pas d'intégration constant, sa valeur doit être sélectionnée en fonction des transitoires les plus rapides. Une règle de pouce consiste à choisir un pas égal à l'inverse de 10 fois la fréquence maximale, ce qui donne $\Delta t = 1/(10 \times 27000) \approx 4\mu\text{s}$. Cette valeur sera utilisée pour exécuter une simulation à pas fixe et les résultats serviront de référence pour la comparaison avec l'algorithme pas variable. Les paramètres de l'algorithme pas variable sont présentés dans le Tableau 3.1.

Tableau 3.1: Paramètres de l'algorithme utilisant un pas d'intégration variable

Δt_1	Δt_2	Δt_3	Tol_1	Tol_2	n_{count}
$4\mu\text{s}$	$40\mu\text{s}$	$100\mu\text{s}$	2 %	1 %	500

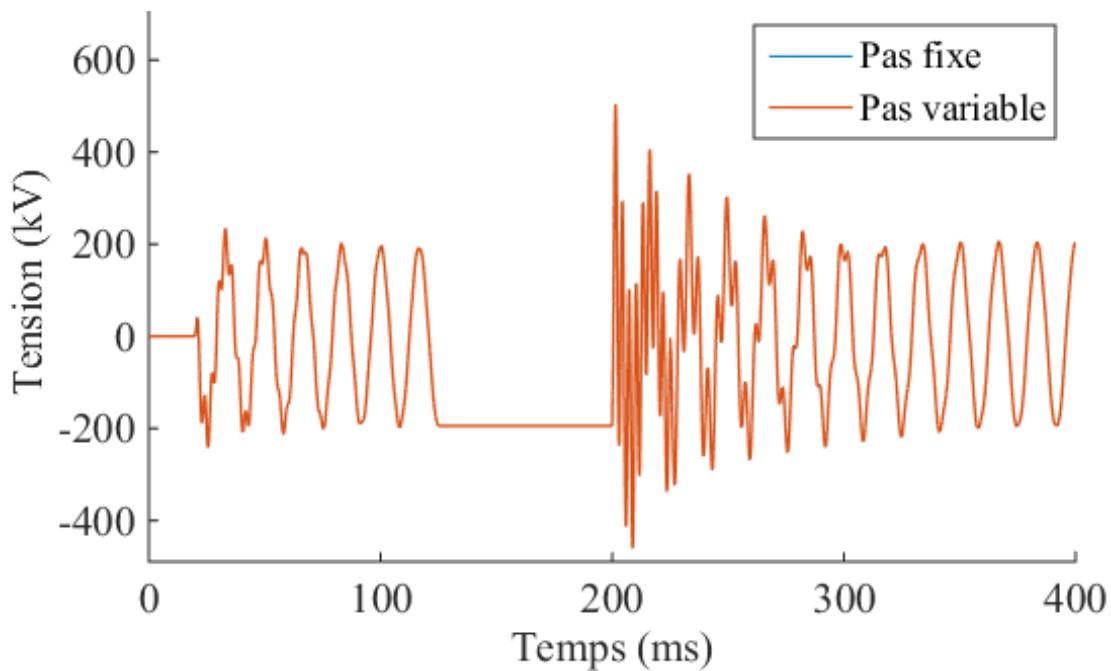


Figure 3.4: Tension du condensateur C1

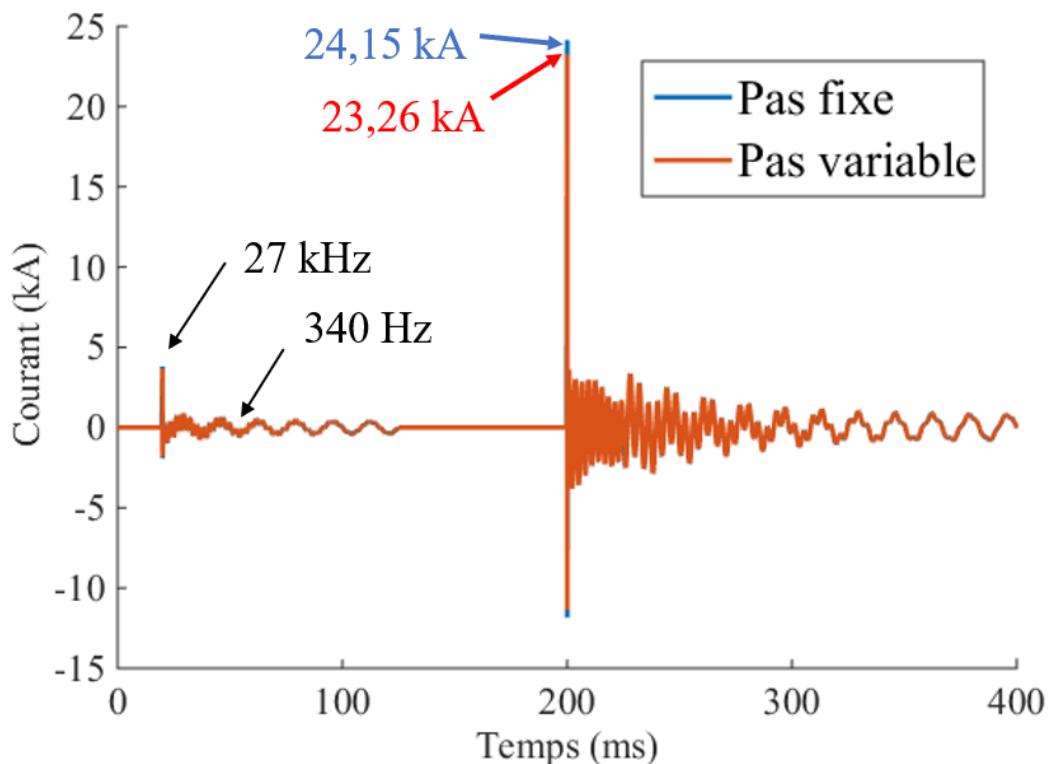


Figure 3.5: Courant dans la phase a du disjoncteur BR1

Les résultats de la simulation avec un pas d'intégration fixe $\Delta t = 4 \mu s$ (pas fixe) et avec un pas d'intégration variable (pas variable) sont présentés à la Figure 3.4 (tension aux bornes du condensateur C1) et à la Figure 3.5 (courant dans la phase a du disjoncteur BR1). Pour la simulation pas fixe, la valeur du pas d'intégration est choisie suffisamment petite pour obtenir un résultat très précis. C'est donc une simulation de référence en matière de précision.

Les formes d'onde obtenues après simulation contiennent des composantes transitoires de différentes fréquences et durées. On peut remarquer que la précision est pratiquement la même pour la simulation à pas fixe et la simulation à pas variable. En particulier, la transitoire de 27 kHz sur le courant du disjoncteur (transitoire la plus rapide donc la plus contraignante) est reproduite avec une excellent précision par l'algorithme à pas variable, comme le montre la Figure 3.6.

Les formes d'onde présentées ci-dessus permettent de conclure que l'algorithme proposé utilisant un pas d'intégration numérique variable est aussi précis que l'algorithme à pas fixe qui utilise un pas suffisamment petit. On peut en effet remarquer (par exemple) que le pic de courant obtenu à l'instant $t = 200 ms$ est égal à 24,15 kA avec l'algorithme de référence utilisant un pas fixe tandis qu'il est de 23,26 kA avec l'algorithme pas fixe. On note donc un léger écart relatif d'environ 3% entre ces valeurs, ce qui est négligeable.

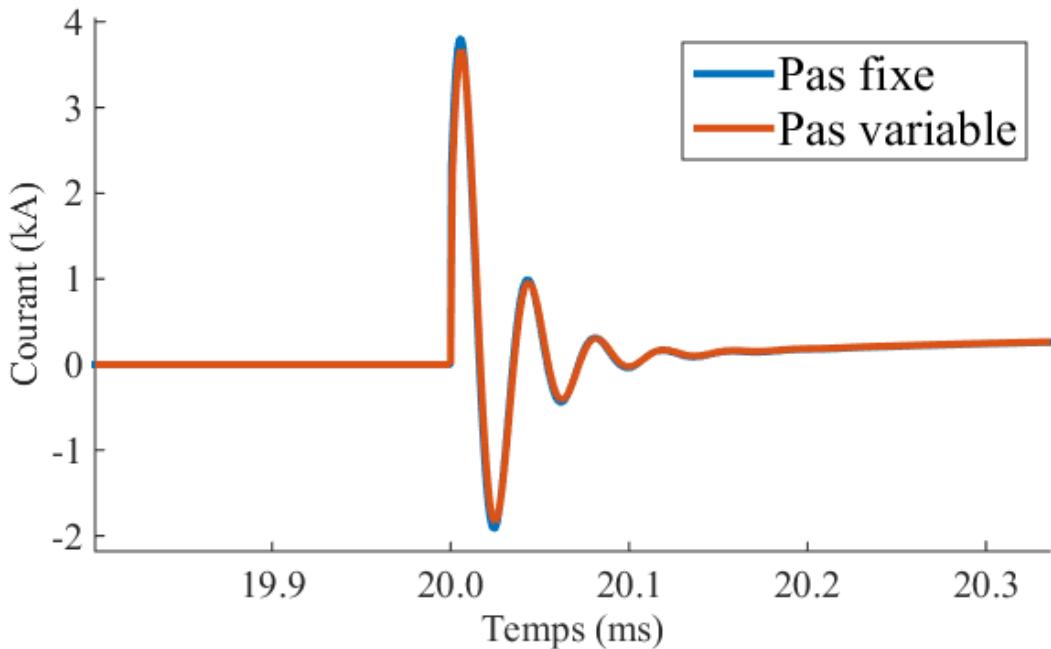


Figure 3.6: Zoom sur la transitoire 27 kHz du courant dans BR1

Les résultats de simulation présentées montrent que la stratégie de changement de pas proposée est capable de représenter les transitoires les plus rapides avec précision (grâce au plus petit pas d'intégration), et par la suite, représenter fidèlement les variations plus lente (grâce aux pas plus élevés) sans perdre en précision. En revanche, avec l'algorithme pas fixe, on est obligé d'utiliser un pas d'intégration suffisamment petit (pendant toute la simulation) pour obtenir une bonne précision, ce qui représente une perte énorme en temps de calcul comme on le verra plus bas. Si le pas d'intégration de l'algorithme à pas fixe est augmenté, on obtiendra évidemment une précision plus faible tandis que l'algorithme à pas variable restera toujours très précis, puisque sa stratégie de variation du pas utilise l'erreur de troncature qui est un indicateur du niveau de précision.

Il serait intéressant à présent d'analyser les performances des deux algorithmes en termes de temps de calcul. Ces performances sont représentées dans le Tableau 3.2.

Tableau 3.2: Évaluation des temps de calcul des deux algorithmes testés

Type de simulation	Simulation avec un pas d'intégration fixe	Simulation avec un pas d'intégration variable
Temps de calcul (s)	1 987	67

Ce tableau montre que l'algorithme à pas variable est beaucoup plus rapide (30 fois dans ce cas-ci) que l'algorithme à pas fixe. Le gain en temps de calcul est énorme. Il est à noter que ce gain peut devenir encore plus élevé si le circuit simulé passe plus de temps en régime permanent. En effet, l'algorithme à pas variable va utiliser le pas d'intégration le plus élevé alors que l'algorithme à pas fixe va conserver son petit pas tout au long de la simulation.

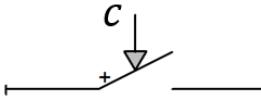
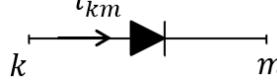
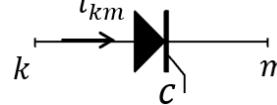
Les résultats obtenus ci-dessus permettent donc de valider l'efficacité (précision et rapidité) de la simulation à pas d'intégration numérique variable en général et de l'algorithme proposé en particulier. Il est important de souligner toutefois que le fait que les paramètres de l'algorithme soient définis par l'utilisateur pose un problème d'optimisation de la performance. Néanmoins, la logique de modification du pas d'intégration tout au long de la simulation permet de garantir une rapidité toujours supérieure à celle de l'algorithme à pas fixe et une précision au moins équivalente à celle de l'algorithme utilisant un pas fixe.

La partie suivante propose une seconde stratégie de variation du pas d'intégration qui est applicable à la simulation des convertisseurs d'électronique de puissance, étant donné les contraintes différentes rencontrées dans ce type de circuits.

3.3 Algorithme utilisant un pas d'intégration numérique variable pour la simulation des convertisseurs d'électronique de puissance

3.3.1 Description des interrupteurs électronique

Tableau 3.3: Conditions de conduction et de blocage de quelques interrupteurs électronique

Type d'interrupteur	Condition de conduction	Condition de blocage
Interrupteur commandé 	$c = 1$	$c = 0$
Diode 	$v_k - v_m \geq V_{seuil}$	$i_{km} < I_{seuil}$
Thyristor 	$(c=1) \text{ et } (v_k - v_m \geq V_{seuil})$	$i_{km} < I_{seuil}$

Les circuits d'électronique de puissance sont caractérisés par le nombre élevé d'interrupteurs électronique qu'ils contiennent. Ces interrupteurs peuvent être des diodes, des transistors, des thyristors, des « Insulated Gate Bipolar Transistors » (IGBT) ou de simples interrupteurs commandés. Ils sont généralement représentés par des modèles idéaux à deux états [88]: ouvert/fermé ou blocage/conduction. La transition d'un état à l'autre est une commutation. La commutation peut être naturelle (cas d'une diode) ou forcée (cas d'un IGBT par exemple). Une

commutation naturelle est provoquée par le dépassement d'un seuil de courant I_{seuil} ou de tension V_{seuil} (généralement tous nuls) tandis qu'une commutation forcée est provoquée par le passage à 1 ou à 0 d'un signal externe c commandant l'interrupteur et provenant d'un « contrôleur ». Il est appelé « signal de commande ». Le Tableau 3.3 est inspiré de [63] et présente les conditions de commutation (conduction ou blocage) de quelques interrupteurs électronique

L'IGBT n'est pas représenté dans ce tableau parce que généralement, il est modélisé par une association de deux diodes et d'un interrupteur contrôlé comme présenté à la Figure 3.7.

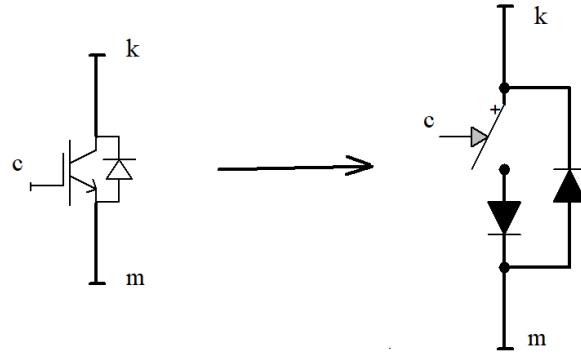


Figure 3.7: Modélisation d'un IGBT

La diode quant à elle représente le composant de base de tout interrupteur électronique. Il s'agit d'une jonction à semi-conducteurs dont le comportement peut être décrit par une équation non-linéaire entre le courant qui la traverse et la tension à ses bornes. Toutes, pour la plupart des applications dans les circuits réels, des modèle simplifiés ou idéalisés sont établis. Ainsi, il est d'usage de représenter une diode à l'état de conduction par une résistance R_{ON} de valeur très faible, en série avec une source de tension V_{DC} . A l'état bloqué, la diode est simplement représentée par une résistance très élevée R_{OFF} . Le circuit équivalent obtenu pour la diode, ainsi que les valeurs généralement admises pour les paramètres sont présentés à la Figure 3.8.

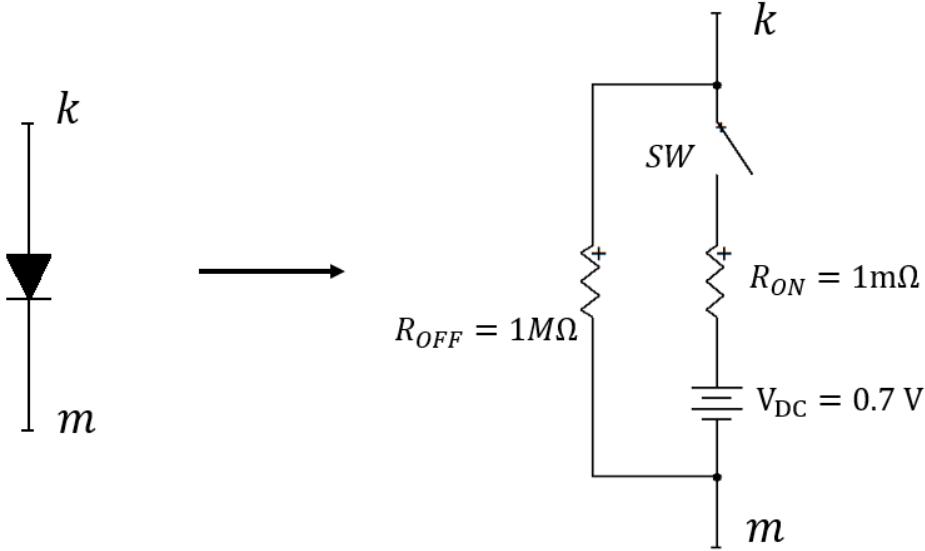


Figure 3.8: Modélisation d'une diode

Ainsi, les conditions de conduction et de blocage de la diode présentées au Tableau 3.3 agissent directement sur l'interrupteur SW de la Figure 3.8. Aussi, en général $V_{\text{seuil}} = V_{\text{DC}}$.

Chaque fois que la condition de conduction ou de blocage d'un interrupteur est satisfaite, une commutation a lieu, engendrant ainsi une discontinuité. De nombreuses discontinuités sont observées lors de la simulation des circuits d'électronique de puissance car ces circuits sont constitués d'un nombre très élevé d'interrupteurs commandés par des signaux à des fréquences qui peuvent atteindre des centaines de kHz. L'occurrence des discontinuités peut conduire à des problèmes de précision dans la simulation. En effet, avec un algorithme standard qui utilise un pas d'intégration fixe, il est très probable qu'une discontinuité se produise à un instant situé entre deux points de temps consécutifs. Cette discontinuité ne peut être détectée et prise en compte qu'au point de temps suivant, ce qui crée donc une erreur. Cette erreur contribue à baisser la précision de la simulation et peut dans certains cas conduire à l'apparition de fréquences harmoniques non caractéristiques dans les formes d'onde du circuit simulé [51].

3.3.2 Détection précise des discontinuités par l'interpolation

La littérature propose des techniques d'interpolation pour la détection précise des instants de discontinuité. Par exemple, l'interpolation linéaire [50],[51] permet de revenir en arrière dans la simulation et estimer les valeurs des variables du réseau simulé à l'instant de discontinuité.

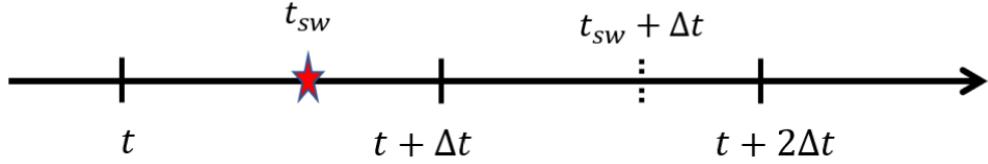


Figure 3.9: Exemple de time-mesh

Par exemple, sur le time-mesh de la Figure 3.9, le système d'équations du réseau est résolu à l'instant t , puis à l'instant $t + \Delta t$. A ce moment, l'algorithme détecte qu'un changement d'état d'un interrupteur s'est produit à l'instant t_{sw} . Dans un algorithme standard, la discontinuité est prise en compte à l'instant $t + \Delta t$ et la simulation se poursuit ensuite à $t + 2\Delta t$. Cependant, l'interpolation linéaire peut être exécutée pour estimer la valeur de n'importe quelle variable x à l'instant t_{sw} , soit $x_{t_{sw}}$, à partir de ses valeurs x_t à l'instant t et $x_{t+\Delta t}$ à l'instant $t + \Delta t$:

$$x_{t_{sw}} = (1 - \alpha)x_t + \alpha \cdot x_{t+\Delta t} \quad (3.5)$$

$$\alpha = (t_{sw} - t) / \Delta t \quad (3.6)$$

Lorsque la commutation est forcée, l'instant de son apparition t_{sw} est fourni par le contrôleur externe. En cas de commutation naturelle, l'interpolation linéaire peut être aussi utilisée pour déterminer t_{sw} . Par exemple, si le courant i_t dans une diode à l'instant t est $i_t > I_{seuil}$ et que sa valeur à l'instant $t + \Delta t$ est $i_{t+\Delta t} < I_{seuil}$, alors l'instant t_{sw} où le courant devient exactement égal à I_{seuil} (condition d'ouverture de la diode) est égal à :

$$t_{sw} = t + \frac{I_{seuil} - i_t}{i_{t+\Delta t} - i_t} \Delta t \quad (3.7)$$

Une formule similaire peut être établie pour le calcul de l'instant de franchissement du seuil de tension V_{seuil} (condition de conduction de la diode).

Après la détermination par interpolation linéaire de toutes les variables du réseau simulé à l'instant t_{sw} , la simulation peut évoluer au point de temps $t_{sw} + \Delta t$ par intégration numérique. A cet instant, une autre interpolation peut être appliquée pour recalculer les valeurs des variables du circuit à

l'instant $t + \Delta t$ qui est situé sur le time-mesh. L'interpolation permet en général d'améliorer la précision. Toutefois, cela reste une opération mathématique qui est elle-même une source d'erreur. L'utilisation d'un pas d'intégration plus petit lors de la détection et la traversée des discontinuités est une solution qui produit une erreur moins importante.

3.3.3 Les fondements de l'algorithme proposé

A la lumière de toutes les informations présentées ci-dessus, il est possible de développer un algorithme combinant l'utilisation d'un pas d'intégration variable et l'utilisation de l'interpolation linéaire. L'algorithme requiert deux valeurs de pas d'intégration (définies par l'utilisateur) : un pas très petit Δt_1 permettant de traverser les discontinuités et un pas normal Δt_2 permettant d'effectuer la simulation en l'absence de discontinuité. Afin de réduire au maximum les erreurs créées lors de la traversée des discontinuités, il est important de déterminer avec précision les valeurs des variables du réseau juste avant et juste après l'apparition de chaque discontinuité. Pour cela :

- D'une part, on peut envisager que l'interpolation ne ramène pas la simulation à l'instant de discontinuité, mais un pas de temps avant. De cette manière, les variables du circuit au point de discontinuité seront obtenues par intégration numérique et non par interpolation. L'erreur créée par l'interpolation va donc être amortie par l'intégration numérique exécutée pour se rendre au point de discontinuité. Ceci contribue à réduire l'erreur dans la détermination des variables à l'instant de discontinuité.
- D'autre part, le calcul des variables du réseau au point de temps suivant le point de discontinuité peut être exécuté avec un pas d'intégration plus petit. Ceci permet de calculer avec plus de précision les valeurs des variables juste après la discontinuité et réduire les erreurs dues à l'apparition de la discontinuité.

La prise en compte de ces deux spécificités dans l'algorithme de simulation à pas variable qui est proposé permet de traverser chaque discontinuité en minimisant les erreurs dans la détermination des variables, et d'améliorer de ce fait la précision dans la simulation.

3.3.4 Séquence des évènements de l'algorithme proposé

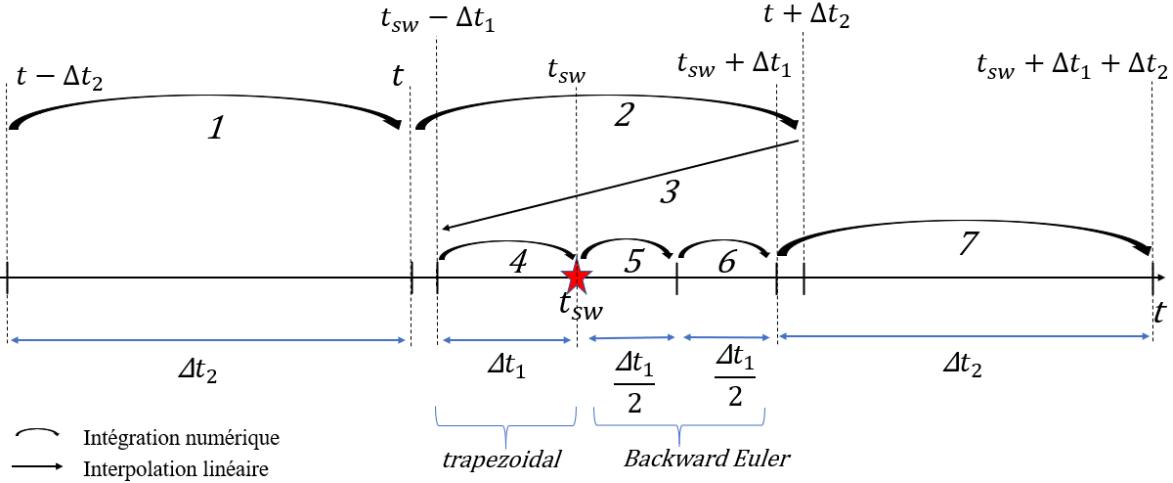


Figure 3.10: Description de l'algorithme à pas variable proposé.

La Figure 3.10 permet d'illustrer le fonctionnement de l'algorithme proposé, utilisant un pas d'intégration numérique variable pour la simulation des circuits d'électronique de puissance. Les différentes étapes sont les suivantes :

1. La simulation est normalement exécutée avec le pas Δt_2 . L'intégration TR est exécutée pour évoluer d'un point de temps au point suivant. Une discontinuité a lieu à l'instant t_{sw} . Elle est détectée à l'instant $t + \Delta t_2$ et l'algorithme passe à l'étape 2.
2. Le moment exact d'apparition de la discontinuité t_{sw} est déterminé et le pas d'intégration est réduit à la valeur Δt_1 .
3. A partir des solutions aux instants t et $t + \Delta t_2$, l'interpolation linéaire est appliquée pour calculer les variables et ramener la simulation à l'instant $t_{sw} - \Delta t_1$.
4. Une intégration trapézoïdale est exécutée pour évoluer de $t_{sw} - \Delta t_1$ à t_{sw} .
5. La discontinuité est prise en compte (modification de la matrice du circuit) et deux intégrations BE avec un pas $\Delta t_1/2$ sont exécutées pour passer de t_{sw} à $t_{sw} + \Delta t_1$.

6. Le pas d'intégration est ramené à la valeur Δt_2 et la simulation continue jusqu'à ce qu'une nouvelle discontinuité soit détectée. Lorsque c'est le cas, l'algorithme retourne à l'étape 2.

Il est important de mentionner que cet algorithme permet de gérer les cas de multiples commutations se produisant entre deux points de temps consécutifs.

3.3.5 Application à la simulation d'un convertisseur, analyses et conclusions

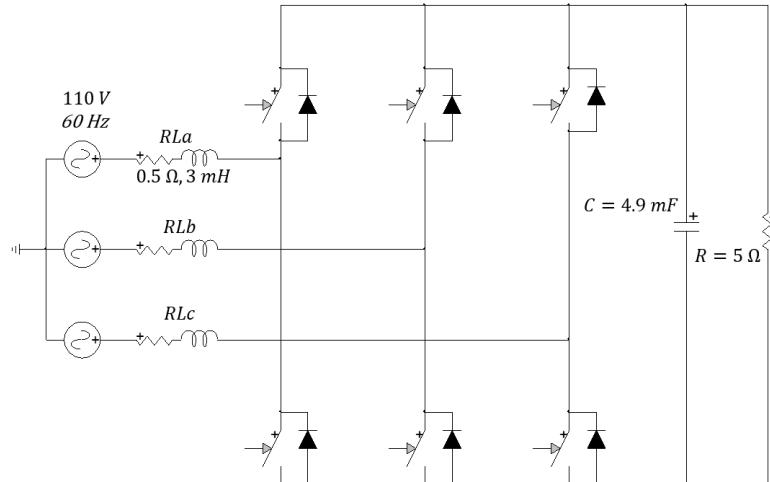


Figure 3.11: Convertisseur AC/DC

Le circuit de la Figure 3.11 est un convertisseur AC/DC constitué d'une source triphasée de 110 V RMS ligne-ligne. Ce circuit est simulé pendant 200 ms. Les six interrupteurs commandés reçoivent des signaux provenant d'un contrôleur à modulation de largeur d'impulsion (PWM). Ce contrôleur génère un signal triangulaire (appelée « porteuse ») à la fréquence de 1000 Hz. A chaque point de temps, la valeur de la porteuse est comparée à chacune des trois phases d'un système triphasé de tensions appelé « référence ». L'indice de modulation est égal à 0.8. Lorsque la référence pour une phase donnée est supérieure à la porteuse, un signal de commande égal à 1 est transmis à l'interrupteur supérieur connecté à la phase en question (signal de fermeture). Cet interrupteur doit donc entrer en conduction à cet instant s'il ne l'était pas déjà. L'interrupteur inférieur connecté à la même phase reçoit un signal complémentaire (égal à 0) qui correspond au signal d'ouverture. Il doit donc entrer en régime bloqué s'il ne l'était pas déjà. Inversement, lorsque la référence devient inférieure à la porteuse, l'interrupteur supérieur reçoit un signal de blocage tandis que l'interrupteur inférieur reçoit un signal de conduction.

Ce circuit est simulé avec l'algorithme proposé. Les résultats sont comparés à ceux obtenus avec deux autres algorithmes à pas d'intégration fixe : l'un utilisant l'interpolation linéaire pour la détection des discontinuités et l'autre ne l'utilisant pas. Les formes d'onde de référence sont obtenues par une simulation à pas d'intégration fixe avec un pas suffisamment petit pour garantir une excellent précision. Pour résumer, on aura donc les quatre simulations suivantes :

- Simulation à pas d'intégration fixe $\Delta t = 1\mu s$ (référence).
- Simulation à pas d'intégration variable avec $\Delta t_1 = 0.5\mu s$ et $\Delta t_2 = 100\mu s$ (pas variable).
- Simulation à pas d'intégration fixe $\Delta t = 100\mu s$ avec interpolation (pas fixe+interpolation).
- Simulation à pas d'intégration fixe $\Delta t = 100\mu s$ (pas fixe).

Les formes d'onde du courant dans l'impédance RLa (courant de la source) et de la tension aux bornes du condensateur C (tension de la charge) sont présentés aux Figure 3.12 et Figure 3.13.

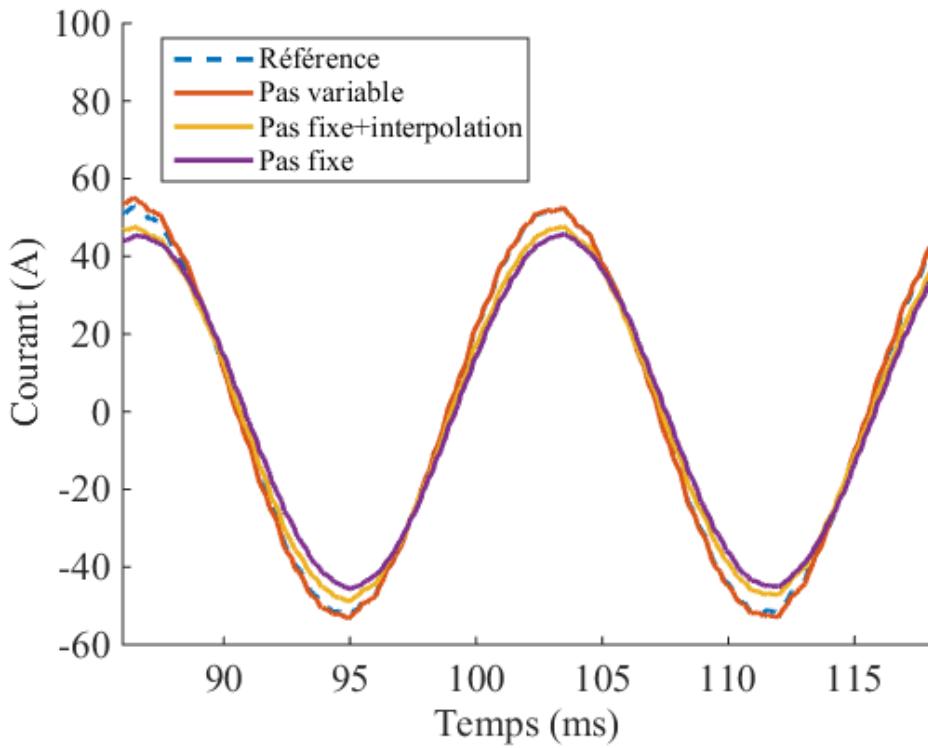


Figure 3.12: Courant dans l'impédance RLa pour les méthodes testées

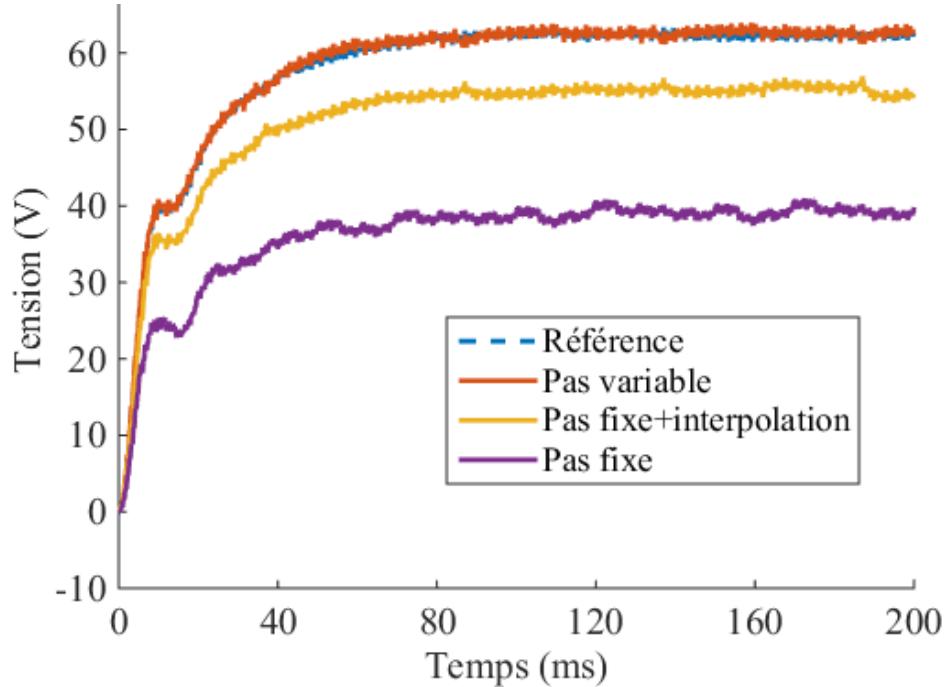


Figure 3.13: Tension aux bornes du condensateur C pour les méthodes testées

Ces résultats de la simulation montrent que l'algorithme utilisant un pas d'intégration fixe donne une mauvaise précision. Cela est principalement dû au pas d'intégration qui est élevé. L'ajout de l'interpolation linéaire pour la détection des discontinuités améliore la précision. Toutefois, les formes d'onde montrent que cette amélioration n'est pas suffisante. L'algorithme proposé, utilisant un pas d'intégration variable est la seule méthode qui donne une précision égale à celle de la référence. On peut finalement constater que les erreurs des méthodes testées sont plus significatives au niveau des formes d'onde de la tension de la charge.

Afin de quantifier la performance des différents algorithmes, il est possible de calculer l'erreur relative de chaque méthode par rapport à la référence. En choisissant les formes d'onde de la Figure 3.13 par exemple, il est possible de calculer à chaque instant t de la simulation, l'écart relatif entre la valeur de la tension de référence à cet instant v_{ref_t} et la valeur v_t de la tension mesurée avec l'une des méthodes testées au même instant. La formule de calcul est la suivante:

$$e_t(\%) = \left| \frac{v_{ref_t} - v_t}{v_{ref_t}} \right| \times 100 \quad (3.8)$$

En répétant ce calcul pour N instants différents $\{t_i\}_{i=1 \dots N}$, on peut calculer la valeur moyenne de l'erreur relative $e(\%)$ de chaque méthode :

$$e(\%) = \frac{\sum_{i=1}^N e_{t_i}(\%)}{N} \quad (3.9)$$

De la même façon, on peut quantifier la vitesse d'exécution (VE_k) de chaque algorithme k en comparant la durée totale T_k requise par cet algorithme pour compléter la simulation à la durée totale T_{ref} requise par la simulation de référence pour compléter également la simulation :

$$VE_k = T_{ref}/T_k \quad (3.10)$$

La comparaison des erreurs relatives et des vitesses d'exécution des méthodes testées par rapport à la référence permet de construire le Tableau 3.4.

Tableau 3.4: Performances des méthodes testées

Simulation	Erreur relative $e(\%)$	Vitesse d'exécution VE
Référence ($\Delta t = 1\mu s$)	0 %	$\times 1$
Pas variable $(\Delta t_1 = 0.5\mu s$ et $\Delta t_2 = 100\mu s$)	1.61 %	$\times 889$
Pas fixe+interpolation $(\Delta t = 100\mu s)$	10.54 %	$\times 1779$
Pas fixe ($\Delta t = 100\mu s$)	35.94 %	$\times 2668$

Conformément aux observations faites à partir des formes d'onde, l'algorithme proposé qui utilise un pas d'intégration variable est le plus précis, suivi de l'algorithme utilisant un pas d'intégration fixe avec interpolation. La relative différence d'erreur entre l'algorithme à pas variable et celui à pas fixe+interpolation est assez élevée. Ceci confirme l'efficacité de l'approche de simulation à

pas variable en général, et de la méthode proposée en particulier. Concernant les temps de calcul, on peut constater que l'algorithme à pas fixe est le plus rapide. Ceci est normal vu qu'il utilise un pas très grand (pour produire un résultat pas du tout précis). En ajoutant l'interpolation, la précision s'améliore, mais le temps de calcul augmente (1.5 fois plus élevé). Ceci est dû au fait que les retours en arrières engendrés par l'interpolation rallongent la durée totale de la simulation. Finalement, on peut observer que l'algorithme à pas variable est le moins rapide des 3 méthodes. En baissant le pas d'intégration pour réduire les erreurs chaque fois que la simulation rencontre une discontinuité, le nombre de points de temps requis pour compléter la simulation augmente significativement. Toutefois, l'excellente précision de l'algorithme à pas d'intégration variable en fait une solution prometteuse pour la simulation des circuits d'électronique de puissance, d'autant plus que lorsqu'on le compare à la référence, cet algorithme est beaucoup plus rapide.

CHAPITRE 4 ÉTUDE DE LA PRÉCISION DANS LA SIMULATION DES CIRCUITS D'ÉLECTRONIQUE DE PUISSANCE

Le niveau de précision dans le calcul des variables lors d'une simulation est fonction de la méthode d'intégration numérique utilisée. L'intégration TR est la plus courante, ceci grâce à sa stabilité et son excellente précision. De multiples changements d'état des interrupteurs sont observés lors de la simulation des circuits d'électronique de puissance. Ces changements entraînent des discontinuités qui doivent être correctement traitées au risque de créer des erreurs additionnelles dans la simulation. Le traitement des discontinuités soulève de ce fait plusieurs défis que les algorithmes existants tentent d'adresser à travers l'utilisation de diverses techniques telles que l'interpolation et la réinitialisation. Ces techniques sont en général efficaces [89], mais produisent dans certains cas des résultats contre-intuitifs. Les objectifs de ce chapitre sont d'identifier les principaux défis liés à la gestion des discontinuités lors de la simulation des circuits d'électronique de puissance, puis d'analyser comment les techniques existantes impactent sur la précision. Cette étude est présentée dans [90] par l'auteur de cette thèse. Elle constitue une étape indispensable vers le développement de nouvelles méthodes de simulation plus précises.

4.1 Les défis liés à la gestion des discontinuités

4.1.1 La détermination précise des instants d'apparition des discontinuités et le calcul des variables à ces instants

Lorsqu'une discontinuité survient à un instant situé entre deux points de temps consécutifs d'une simulation exécutée avec un pas d'intégration constant, il est important de déterminer l'instant exact d'apparition de la discontinuité, et de calculer les valeurs prises par les variables du circuit à cet instant. Ceci permet de réduire, voire d'éliminer l'erreur de détection de la discontinuité. Il est d'usage d'utiliser pour cela des techniques d'interpolation. L'interpolation peut être linéaire ou quadratique.

4.1.1.1 Formules d'interpolation pour le calcul des variables aux instants de discontinuités

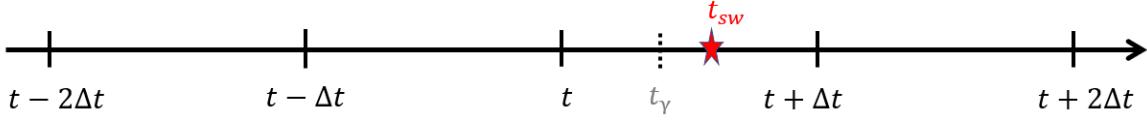


Figure 4.1: Exemple de time-mesh pour illustrer les techniques d'interpolation

Sur le time-mesh de la Figure 4.1, une discontinuité a lieu à l'instant t_{sw} situé entre t et $t + \Delta t$, deux points de temps consécutifs du time-mesh. La valeur $x_{t_{sw}}$ prise par une variable quelconque x du circuit simulé à l'instant t_{sw} peut se déterminer par interpolation linéaire [50] à partir des valeurs x_t et $x_{t+\Delta t}$ prises par cette variable aux instants t et $t + \Delta t$:

$$x_{t_{sw}} = (1 - \alpha)x_t + \alpha \cdot x_{t+\Delta t} \quad (4.1)$$

$$\alpha = \frac{t_{sw} - t}{(t + \Delta t) - t} \quad (4.2)$$

Il est également possible d'utiliser l'interpolation quadratique [55],[56] pour déterminer $x_{t_{sw}}$. L'interpolation quadratique nécessite trois points, contrairement à l'interpolation linéaire qui n'a besoin que de deux points. En utilisant la valeur disponible $x_{t-\Delta t}$ à l'instant $t - \Delta t$, la formule de calcul de $x_{t_{sw}}$ par interpolation quadratique est :

$$x_{t_{sw}} = p_1 x_{t-\Delta t} + p_2 x_t + p_3 x_{t+\Delta t} \quad (4.3)$$

$$\begin{aligned} p_1 &= \frac{(t_{sw} - t)(t_{sw} - (t + \Delta t))}{((t - \Delta t) - t)((t - \Delta t) - (t + \Delta t))} \\ p_2 &= \frac{(t_{sw} - (t - \Delta t))(t_{sw} - (t + \Delta t))}{(t - (t - \Delta t))(t - (t + \Delta t))} \\ p_3 &= \frac{(t_{sw} - (t - \Delta t))(t_{sw} - t)}{((t + \Delta t) - (t - \Delta t))((t + \Delta t) - t)} \end{aligned} \quad (4.4)$$

Si la méthode d'intégration numérique est multi étapes (à l'instar de 2S-DIRK ou TR-BDF2), alors une solution intermédiaire x_{t_γ} a été calculée à un point intermédiaire t_γ situé entre t et $t + \Delta t$. Cette solution intermédiaire peut être utilisée pour l'application de l'interpolation quadratique. Cela

consistera donc dans l'équation (4.3) à remplacer $x_{t-\Delta t}$, x_t et $x_{t+\Delta t}$ par x_t , x_{t_γ} et $x_{t+\Delta t}$ respectivement. Ensuite, dans (4.4), il faudra remplacer $t - \Delta t$, t et $t + \Delta t$ respectivement par t , t_γ et $t + \Delta t$, ce qui donne :

$$x_{t_{sw}} = p_1 x_t + p_2 x_{t_\gamma} + p_3 x_{t+\Delta t} \quad (4.5)$$

$$\begin{aligned} p_1 &= \frac{(t_{sw} - t_\gamma)(t_{sw} - (t + \Delta t))}{(t - t_\gamma)(t - (t + \Delta t))} \\ p_2 &= \frac{(t_{sw} - t)(t_{sw} - (t + \Delta t))}{(t_\gamma - t)(t_\gamma - (t + \Delta t))} \\ p_3 &= \frac{(t_{sw} - t)(t_{sw} - t_\gamma)}{((t + \Delta t) - t)((t + \Delta t) - t_\gamma)} \end{aligned} \quad (4.6)$$

Dans une simulation, lorsqu'une discontinuité se produit à un instant t_{sw} connu, l'équation (4.1), (4.3) ou (4.5) peut être utilisée pour déterminer la valeur prise par n'importe quelle variable du circuit à l'instant t_{sw} . Toutefois, ce calcul nécessite de connaître au préalable la valeur de t_{sw} .

4.1.1.2 Approche de détermination de l'instant de discontinuité

La façon de déterminer l'instant t_{sw} d'apparition d'une discontinuité n'est pas la même, dépendamment de s'il s'agit d'une discontinuité créée par une commutation forcée ou une commutation naturelle.

4.1.1.2.1 Cas d'une commutation forcée

Dans le cas d'une commutation forcée, t_{sw} est imposé par un contrôleur externe. La plupart des contrôleurs génèrent des signaux en comparant une porteuse triangulaire à un signal de référence.

Le signal de référence est généralement un signal co-sinusoidal d'amplitude A_r , de fréquence f_r et de phase initiale ϕ exprimé comme suit :

$$y_r = A_r \cos(2\pi f_r t + \phi) \quad (4.7)$$

La porteuse est quant à elle un signal triangulaire dont l'équation peut s'écrire :

$$y_p = \frac{2A_p}{\pi} \arcsin(\sin(2\pi f_p t)) \quad (4.8)$$

A_p et f_p représentent l'amplitude et la fréquence de la porteuse. Sa représentation graphique est un ensemble de « dents de scies » obtenues à partir de deux segments de droite définis dans des intervalles précis qui sont dupliqués avec une périodicité $T_p = 1/f_p$. Ainsi, sur l'intervalle $[-T_p/4, 3T_p/4]$ qui est équivalent à une période, l'équation (4.8) peut s'écrire :

$$\begin{cases} y_p = A_p \frac{4t_p}{T_p} & \text{pour } -\frac{T_p}{4} \leq t_p < \frac{T_p}{4} \\ y_p = A_p \frac{4(T_p/2 - t_p)}{T_p} & \text{pour } \frac{T_p}{4} \leq t_p < \frac{3T_p}{4} \end{cases} \quad (4.9)$$

Pour obtenir la valeur de la porteuse à tout instant t , il suffit au préalable de calculer t_p , le reste de la division de t par T_p , puis d'utiliser la relation (4.9). Lorsque la référence est supérieure à la porteuse, un signal égal à 1 est généré par le système de control. Dans le cas contraire, le signal généré est égal à 0.

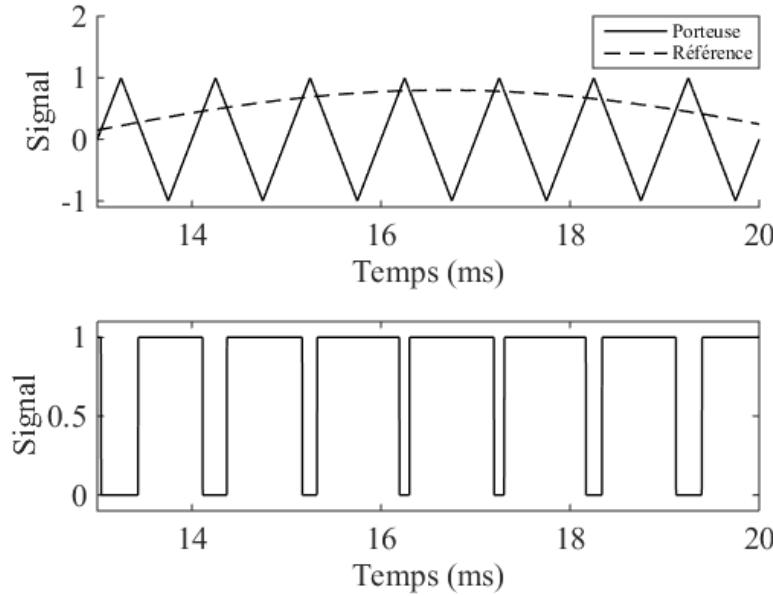


Figure 4.2: Exemple de signaux issus d'un contrôleur

Sur la Figure 4.2, lorsque la référence et la porteuse changent de positions relatives, l'instant de croisement des deux caractéristiques représente l'instant de discontinuité. A l'instant t , la référence et la porteuse prennent respectivement les valeurs y_{r_t} et y_{p_t} . De même, à l'instant $t + \Delta t$ ces valeurs deviennent $y_{r_{t+\Delta t}}$ et $y_{p_{t+\Delta t}}$. Lorsqu'une discontinuité a lieu à l'instant t_{sw} situé entre t et $t + \Delta t$, alors $y_{r_t} < y_{p_t}$ et $y_{r_{t+\Delta t}} > y_{p_{t+\Delta t}}$ ou alors $y_{r_t} > y_{p_t}$ et $y_{r_{t+\Delta t}} < y_{p_{t+\Delta t}}$. Une façon simple et efficace de calculer t_{sw} consiste à approximer la référence par une droite et d'en déterminer l'équation à partir des paires (t, y_{r_t}) et $(t + \Delta t, y_{r_{t+\Delta t}})$. Le résultat est une équation de la forme :

$$y = a_r t + b_r \quad (4.10)$$

Avec $a_r = (y_{r_{t+\Delta t}} - y_{r_t})/\Delta t$ et $b_r = y_{r_t} - a_r t$. La même démarche est appliquée pour obtenir l'équation de la porteuse à partir des paires (t, y_{p_t}) et $(t + \Delta t, y_{p_{t+\Delta t}})$, ce qui donne:

$$y = a_p t + b_p \quad (4.11)$$

Avec $a_p = (y_{p_{t+\Delta t}} - y_{p_t})/\Delta t$ et $b_p = y_{p_t} - a_p t$. La dernière étape est donc de résoudre le système d'équations construit à partir de (4.10) et (4.11). La solution représente l'instant de discontinuité:

$$t_{sw} = -\frac{b_p - b_r}{a_p - a_r} \quad (4.12)$$

Il est important de préciser que si les valeurs y_{p_t} et $y_{p_{t+\Delta t}}$ de la porteuse sont situées sur deux pentes différentes de sa caractéristique, alors il faudra faire intervenir les coordonnées du sommet de la « dent de scie » et formuler les équations des deux segments de part et d'autre de ce sommet avant de rechercher lequel des deux segments intercepte la caractéristique de la référence et obtenir t_{sw} .

4.1.1.2.2 Cas d'une commutation naturelle

Pour des commutations naturelles, t_{sw} ne dépend pas d'une source externe. Il correspond plutôt à l'instant auquel une variable d'un interrupteur (courant qui y circule ou tension à ses bornes) franchit un seuil X_{seuil} . Ainsi, lors du passage de l'instant de simulation t à l'instant suivant $t + \Delta t$,

une variable donnée x peut passer de $x_t < X_{\text{seuil}}$ à $x_{t+\Delta t} > X_{\text{seuil}}$ ou alors de $x_t > X_{\text{seuil}}$ à $x_{t+\Delta t} < X_{\text{seuil}}$. La valeur de X_{seuil} peut correspondre à un seuil de courant I_{seuil} ou de tension V_{seuil} . Dans l'un des cas comme dans l'autre, t_{sw} peut être déterminé par interpolation (linéaire ou quadratique).

Pour déterminer t_{sw} par interpolation linéaire, il suffit de reprendre les équations (4.1) et (4.2) puis d'interchanger les variables en x avec les variables de temps. Autrement dit, interchanger $x_{t_{sw}}$, x_t et $x_{t+\Delta t}$ avec t_{sw} , t et $t + \Delta t$ respectivement. Et puisque $x_{t_{sw}} = X_{\text{seuil}}$, on a donc :

$$t_{sw} = (1 - \alpha)t + \alpha(t + \Delta t) \quad (4.13)$$

$$\alpha = \frac{X_{\text{seuil}} - x_t}{x_{t+\Delta t} - x_t} \quad (4.14)$$

On peut obtenir les résultats suivants à partir de l'interpolation quadratique :

$$t_{sw} = p_1(t - \Delta t) + p_2t + p_3(t + \Delta t) \quad (4.15)$$

$$\begin{aligned} p_1 &= \frac{(X_{\text{seuil}} - x_t)(X_{\text{seuil}} - x_{t+\Delta t})}{(x_{t-\Delta t} - x_t)(x_{t-\Delta t} - x_{t+\Delta t})} \\ p_2 &= \frac{(X_{\text{seuil}} - x_{t-\Delta t})(X_{\text{seuil}} - x_{t+\Delta t})}{(x_t - x_{t-\Delta t})(x_t - x_{t+\Delta t})} \\ p_3 &= \frac{(X_{\text{seuil}} - x_{t-\Delta t})(X_{\text{seuil}} - x_t)}{(x_{t+\Delta t} - x_{t-\Delta t})(x_{t+\Delta t} - x_t)} \end{aligned} \quad (4.16)$$

De même, si la méthode d'intégration est multi étapes, les équations (4.15) et (4.16) deviennent :

$$t_{sw} = p_1t + p_2t_\gamma + p_3(t + \Delta t) \quad (4.17)$$

$$\begin{aligned} p_1 &= \frac{(X_{\text{seuil}} - x_{t_\gamma})(X_{\text{seuil}} - x_{t+\Delta t})}{(x_t - x_{t_\gamma})(x_t - x_{t+\Delta t})} \\ p_2 &= \frac{(X_{\text{seuil}} - x_t)(X_{\text{seuil}} - x_{t+\Delta t})}{(x_{t_\gamma} - x_t)(x_{t_\gamma} - x_{t+\Delta t})} \\ p_3 &= \frac{(X_{\text{seuil}} - x_t)(X_{\text{seuil}} - x_{t_\gamma})}{(x_{t+\Delta t} - x_t)(x_{t+\Delta t} - x_{t_\gamma})} \end{aligned} \quad (4.18)$$

4.1.1.3 Utilisation de l'extrapolation comme technique alternative

L'utilisation de l'interpolation (linéaire ou quadratique) pour la détection de l'instant de discontinuité t_{sw} nécessite le calcul de la solution à $t + \Delta t$ (par résolution du système d'équations du circuit). Il est possible de déterminer t_{sw} à partir de la solution à l'instant t , en utilisant une approche qui empêche de résoudre le système d'équations du circuit à l'instant $t + \Delta t$ avant de revenir en arrière.

Avec cette approche, les cas éventuels de commutation forcée provenant des contrôleurs sont directement détectables par le calcul des valeurs de la porteuse et de la référence à $t + \Delta t$ (en assumant que leurs équations soient directement disponibles). Les cas éventuels de commutation naturelle quant à eux nécessitent l'utilisation de techniques d'extrapolation [91] pour prédire les valeurs de certaines ou de la totalité des variables du système à l'instant $t + \Delta t$ à partir des valeurs connues à l'instant t et aux instants antérieurs. Les solutions obtenues permettent de déterminer si une discontinuité va se produire entre t et $t + \Delta t$, puis de calculer l'instant t_{sw} de son apparition. L'extrapolation peut être linéaire ou quadratique. Le principe de l'extrapolation linéaire consiste simplement à déterminer l'équation du polynôme de degré 1 qui passe par les paires $(t - \Delta t, x_{t-\Delta t})$ et (t, x_t) , puis d'utiliser cette équation pour calculer $x_{t+\Delta t}$, ce qui donne :

$$x_{t+\Delta t} = 2x_t - x_{t-\Delta t} \quad (4.19)$$

Le principe de l'extrapolation quadratique quant à lui consiste à déterminer l'équation d'un polynôme de degré 2 qui passe par $(t - 2\Delta t, x_{t-2\Delta t})$, $(t - \Delta t, x_{t-\Delta t})$ et (t, x_t) , puis d'utiliser cette équation pour calculer $x_{t+\Delta t}$, ce qui donne :

$$x_{t+\Delta t} = x_{t-2\Delta t} - 3x_{t-\Delta t} + 3x_t \quad (4.20)$$

La détermination de $x_{t+\Delta t}$ par extrapolation linéaire ou quadratique permet de détecter d'éventuels franchissements de seuils, indicateurs de commutations naturelles et de calculer les instants de leur apparition grâce aux relations (4.13), (4.15) ou (4.17). Il ressort donc de tout ce qui précède que les techniques d'interpolation et d'extrapolation permettent de détecter avec précision les discontinuités et d'adresser de ce fait le premier défi lié à la gestion des discontinuité.

4.1.2 La suppression des oscillations numériques

4.1.2.1 Origine des oscillations numériques

La plupart des algorithmes de simulation exécutent l'intégration TR. Cette méthode présente un inconvénient majeur. L'occurrence des discontinuités provoque l'apparition d'oscillations numériques sur certaines variables. Il est souvent coutume d'illustrer ce phénomène en considérant l'équation de discréttisation du courant dans une inductance L :

$$i_{t+\Delta t} = \frac{\Delta t}{2L} v_{t+\Delta t} + \frac{\Delta t}{2L} v_t + i_t \quad (4.21)$$

Cette équation peut être ré écrite comme suit :

$$v_{t+\Delta t} = \frac{2L}{\Delta t} (i_{t+\Delta t} - i_t) - v_t \quad (4.22)$$

Si cette inductance est reliée en série avec un interrupteur initialement fermé, et qu'à l'instant t ce dernier ouvre, alors à l'instant $t + \Delta t$, on doit nécessairement avoir $i_{t+\Delta t} = 0$, donc :

$$v_{t+\Delta t} = -\frac{2L}{\Delta t} i_t - v_t \quad (4.23)$$

En adaptant (4.22) pour le calcul au point de temps $t + 2\Delta t$, puisque $i_{t+2\Delta t} = i_{t+\Delta t} = 0$, on aura :

$$v_{t+2\Delta t} = -v_{t+\Delta t} \quad (4.24)$$

Ensuite, à chaque point de temps suivant, la tension aux bornes de l'inductance sera égale à l'opposé de la tension au point de temps précédent. Ce comportement résulte donc en l'apparition d'oscillations numériques sur la tension de l'inductance.

On peut procéder de manière similaire pour démontrer que des oscillations numériques apparaissent également sur le courant dans un condensateur lorsque la tension à ses bornes s'annule subitement. Ces oscillations ne traduisent pas un comportement réel du circuit, mais plutôt une caractéristique de la méthode d'intégration TR. Dans un algorithme, il est indispensable, d'empêcher l'apparition de ces oscillations chaque fois qu'une discontinuité se produit. Pour ce faire, il suffit de convenablement réinitialiser les variables du circuit après chaque discontinuité. En effet, avec la méthode TR, le calcul des historiques à chaque point de temps fait intervenir une variable qui n'est

pas la variable d'état. Par exemple, le terme historique dans (4.21) est $\frac{\Delta t}{2L}v_t + i_t$ et on peut bien constater qu'il contient v_t qui n'est pas la variable d'état pour l'inductance.

4.1.2.2 Techniques de suppression des oscillations numériques

Puisque l'objectif général pour la suppression des oscillations consiste à réinitialiser les variables du circuit, plusieurs approches peuvent être envisagées. Elles seront illustrées à partir du time-mesh de la Figure 4.3 dans lequel une discontinuité est détectée à l'instant t . La matrice du circuit est modifiée au même instant. L'exemple de l'ouverture d'un interrupteur en série avec une inductance sera repris à chaque fois pour démontrer comment chacune des approches qui seront présentées permet de supprimer les oscillations numériques.

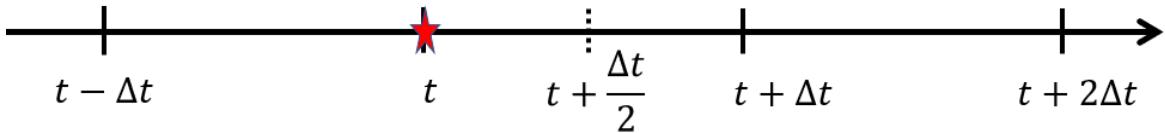


Figure 4.3: Exemple de time-mesh pour illustrer les techniques de suppression des oscillations

La première approche consiste à partir de l'instant t , à effectuer une intégration avec la méthode Backward Euler (BE) et un pas d'intégration égal à $\Delta t/2$. La formule de discréttisation BE correspondante est la suivante :

$$i_{t+\Delta t/2} = \frac{\Delta t}{2L} v_{t+\Delta t/2} + i_t \quad (4.25)$$

Puisque l'ouverture de l'interrupteur intervient à l'annulation du courant, on a $i_{t+\Delta t/2} = i_t = 0$. Ce qui conduit à $v_{t+\Delta t/2} = 0$ dans l'équation (4.25). Cette solution est par la suite ramenée à l'instant t avant de poursuivre la simulation par l'intégration TR pour se rendre de t à $t + \Delta t$. Ainsi aux points de temps suivants, la relation (4.24) ne produira pas d'oscillations numériques, puisque la tension est déjà nulle. Cette approche a été présentée dans [85]. Malheureusement, elle peut conduire à des erreurs dans les autres variables du circuit, puisque la solution à $t + \Delta t/2$ est ramenée à t pour toutes les variables. En d'autres termes, le fait d'avancer un demi pas en avant avec une intégration BE permet d'éviter les oscillations numériques. Toutefois, puisque la solution à $t + \Delta t/2$ est attribuée aux variables du réseau comme solution réinitialisée à t , il en résulte

naturellement une erreur numérique sur toutes les variables. Lorsque Δt est très petit, $t + \Delta t/2$ est très proche de t et l'erreur est plus faible. Mais dans le cas contraire, l'erreur est significativement élevée, ce qui peut être préjudiciable à la méthode.

Une autre version inspirée de l'approche précédemment décrite a été initialement proposée dans [81] et représente la deuxième approche présentée ici. Elle consiste à effectuer deux intégrations BE. La première pour se rendre de t à $t + \Delta t/2$ et la seconde pour évoluer de $t + \Delta t/2$ à $t + \Delta t$. Par la suite, l'intégration TR est exécutée pour poursuivre la simulation. Cette approche représente la méthode TR_BE et est désignée « Critical Damping Adjustment » (CDA). Elle est différente de la précédente et génère une erreur moins importante. La méthode BE utilisée dans cette approche peut être remplacée par une autre méthode comme dans [77] dans laquelle la méthode 2S-DIRK est utilisée. Le principal inconvénient de cette approche est qu'elle nécessite l'utilisation de deux méthodes d'intégration numérique, ce qui est fastidieux.

Les oscillations numériques peuvent être supprimées par l'utilisation de l'interpolation [92]: c'est la troisième approche présentée ici. En effet, après le calcul de la solution à l'instant $t + \Delta t$, une interpolation linéaire peut être appliquée pour déterminer la solution au point intermédiaire $t + \Delta t/2$. Conformément à l'équation (4.24), le résultat de cette interpolation donne la valeur 0 pour les variables susceptibles d'osciller numériquement. Ce résultat est ramené à l'instant $t + \Delta t$ et représente la solution à cet instant. La simulation peut donc se poursuivre avec l'intégration TR. Cette approche est désignée « Chatter Removal Technique » ou CRT [92].

D'autres approches de suppression des oscillations numériques combinent l'intégration TR avec une autre méthode L-stable pour obtenir une méthode d'intégration multi étapes. C'est le cas par exemple de la méthode TR-BDF2 [56],[79],[80],[93] qui a été présentée en 2.1.1.3.6. Cette technique est particulièrement complexe à mettre à œuvre (en comparaison avec les autres approches). Elle est de ce fait très peu utilisée.

4.1.2.3 Cas particuliers

Les techniques CDA et CRT présentées ci-dessus sont les plus usuelles dans les algorithmes de simulation utilisant l'intégration TR. Bien que ces techniques soient efficaces, il existe toutefois des configurations particulières de circuits pour lesquelles les discontinuités conduisent à des oscillations numériques qui persistent. En fait, cela se produit par exemple lorsque le courant dans

une inductance n'est pas nul au moment de la prise en compte de la discontinuité. Pour illustrer cela, le circuit de la Figure 4.4 sera utilisé.

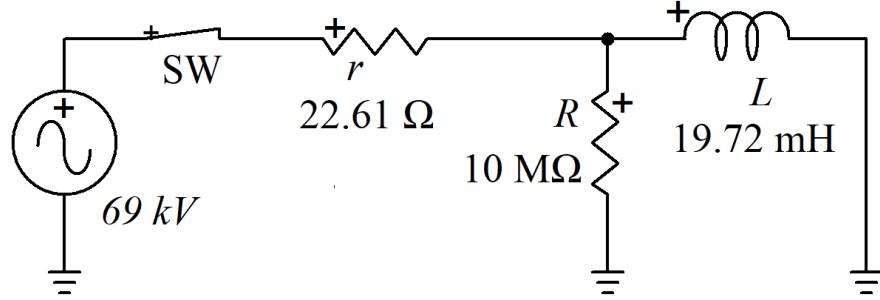


Figure 4.4: Circuit RL simple pour illustrer la persistance des oscillations numériques

Si on considère que l'interrupteur SW s'ouvre à l'instant t_d , avec la méthode TR_BE, les courants aux 2 points de temps suivant l'ouverture de SW se calculent par intégration BE comme suit :

$$i_{t_d + \Delta t/2} = \frac{\Delta t}{2L} v_{t_d + \Delta t/2} + i_{t_d} = -\frac{v_{t_d + \Delta t/2}}{R} \quad (4.26)$$

$$i_{t_d + \Delta t} = \frac{\Delta t}{2L} v_{t_d + \Delta t} - \frac{v_{t_d + \Delta t/2}}{R} = -\frac{v_{t_d + \Delta t}}{R} \quad (4.27)$$

$$v_{t_d + \Delta t} = -\frac{v_{t_d + \Delta t/2}}{R} \left/ \left[-\frac{\Delta t}{2L} - \frac{1}{R} \right] \right. \quad (4.28)$$

Lorsque l'interrupteur SW ouvre, son courant est nul, mais à cause de la présence de la résistance R , le courant i_{t_d} dans l'inductance ne peut pas être nul. Lorsque l'intégration continue avec la méthode TR, à tout instant t , on peut écrire l'expression du courant dans l'inductance :

$$i_t = \frac{\Delta t}{2L} v_t + \frac{\Delta t}{2L} v_{t-\Delta t} + i_{t-\Delta t} = -\frac{v_t}{R} \quad (4.29)$$

$$i_t = \frac{\Delta t}{2L} v_t + \frac{\Delta t}{2L} v_{t-\Delta t} - \frac{v_{t-\Delta t}}{R} = -\frac{v_t}{R} \quad (4.30)$$

$$v_t = v_{t-\Delta t} \left[-\frac{1}{R} + \frac{\Delta t}{2L} \right] \left/ \left[-\frac{\Delta t}{2L} - \frac{1}{R} \right] \right. \quad (4.31)$$

On peut donc constater que lorsque $-1/R + \Delta t/(2L) > 0$, la tension de l'inductance oscille numériquement avec un facteur d'amortissement (inférieur à 1). Ce phénomène est typique des circuits contenant des résistances très grandes.

Pour illustrer cela, le circuit de la Figure 4.4 est simulé en utilisant l'approche « Critical Damping Adjustment » et l'approche « Chatter Removal Technique ». Le pas d'intégration numérique pour chacune des 2 simulations est $\Delta t = 10\mu s$. L'interrupteur SW (initialement fermé) est ouvert à $t = 15 ms$. Les résultats de simulation (tension aux bornes de l'inductance) sont présentés à la Figure 4.5.

Les deux méthodes testées permettent de faire exactement les mêmes observations. A l'ouverture de SW, on a l'impression que la tension tombe à zéro (première sous-figure). Pourtant en effectuant un zoom, on peut constater des oscillations à très haute fréquence qui s'amortissent (deuxième sous-figure). Un zoom plus approfondi (troisième sous-figure) permet de constater que les deux méthodes produisent des oscillations dont les amplitudes ont des ordres de grandeur similaires.

Il est possible de baisser suffisamment le pas d'intégration et capturer exactement la transitoire. En effet, conformément à (4.31), lorsque $-1/R + \Delta t/(2L) = 0$ les oscillations sont évitées. Aussi, lorsque $-1/R + \Delta t/(2L) < 0$, la constante de temps de la branche RL cause la décroissance graduelle de la tension aux bornes de l'inductance et aucune oscillation n'est observée.

L'équation (4.31) montre que pour minimiser l'amplitude des oscillations, il faut minimiser $v_{t-\Delta t}$. Ceci revient à minimiser $v_{t_d+\Delta t/2}$ conformément à (4.26). De cette dernière équation, on peut exprimer :

$$v_{t_d+\Delta t/2} = -i_{t_d} \sqrt{\left[\frac{\Delta t}{2L} + \frac{1}{R} \right]} \quad (4.32)$$

La valeur i_{t_d} représente le courant dans l'inductance juste avant que l'intégration BE ne soit appliquée. Minimiser i_{t_d} permettra donc de minimiser l'amplitude des oscillations. Cela peut survenir lorsque l'interpolation est appliquée pour détecter l'instant précis d'ouverture de SW ou lorsque la valeur du courant est réinitialisée juste avant le passage à l'intégration BE.

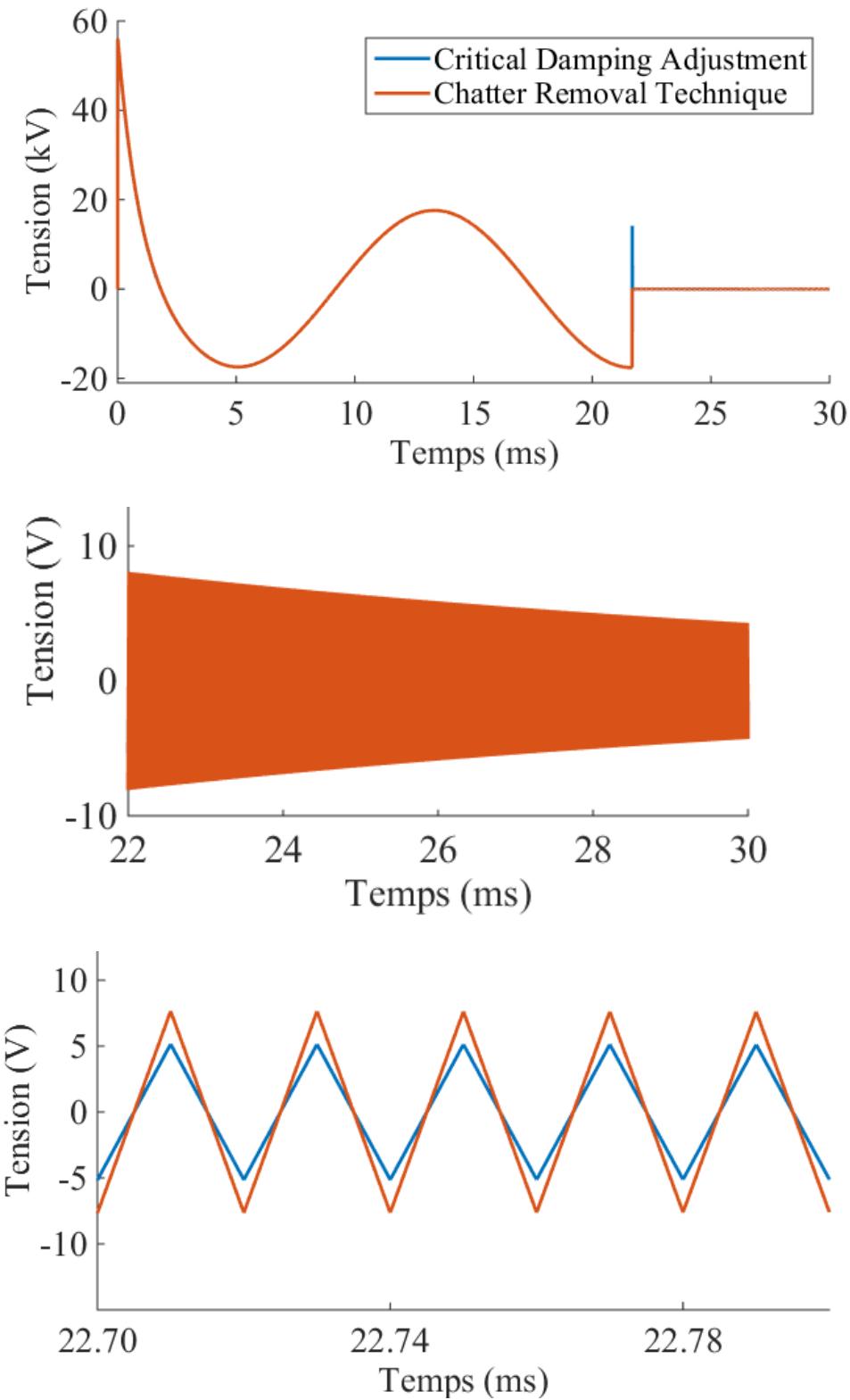


Figure 4.5: Oscillations numériques sur la tension de l'inductance L

Toute l'analyse présentée ci-dessus peut être refaite avec un circuit présentant un condensateur en série avec une résistance et comportant un interrupteur en amont. On peut montrer que des oscillations numériques apparaîtront sur le courant du condensateur à l'ouverture de l'interrupteur.

4.1.3 La réinitialisation du circuit après une discontinuité

4.1.3.1 Réinitialisation des variables

Chaque fois qu'une discontinuité se produit à une instant quelconque t_d d'une simulation, il existe en réalité deux solutions pour les variables du réseau à cet instant. La première solution est la solution juste avant la discontinuité. C'est la solution à l'instant t_d^- . La seconde solution est la solution juste après la discontinuité. C'est la solution à t_d^+ . Lorsque l'interpolation est utilisée pour calculer les variables du réseau à l'instant t_d , en réalité, c'est la solution à t_d^- qui est calculée. Afin d'obtenir la solution après la discontinuité (solution à t_d^+), il faut procéder à une réinitialisation. Il est commun d'admettre que les variables d'état ne peuvent pas subir des variations soudaines. C'est pourquoi l'exercice de réinitialisation ne concerne véritablement que les variables autres que les variables d'état. En l'absence de réinitialisation, la solution calculée au point de temps suivant l'instant t_d contiendra une erreur. Cette erreur se propagera par la suite aux points de temps suivants. Dans la plupart des cas, puisque les méthodes d'intégration numériques sont stables, cette erreur s'amortie progressivement aux pas suivant jusqu'à disparaître totalement. Le problème de la réinitialisation est assez important lorsque l'intégration TR est appliquée. Ceci parce que le calcul de la solution par exemple à l'instant $t_d + \Delta t$ requiert les valeurs des tensions et des courants à t_d , précisément à t_d^+ . Ceci peut être facilement vérifié en observant l'équation (4.21) dans laquelle les termes historiques sont des valeurs de courant et de tension. Cependant, avec l'intégration BE, les termes historiques pour le calcul de la solution à chaque point de temps ne sont liés qu'aux variables d'état. On peut en effet s'en convaincre en observant l'équation (4.25) dans laquelle le calcul du courant d'une inductance ne requiert que le courant au pas précédent comme grandeur historique (le courant est une variable d'état pour une inductance). Les variables d'état ne peuvent pas subir des variations soudaines. A partir de cette observation, il est donc possible de conclure que lorsque la méthode BE est utilisée juste après une discontinuité, la réinitialisation des variables

n'est pas nécessaire. Ceci justifie donc l'efficacité de la méthode TR_BE dans la simulation des circuits électriques.

4.1.3.2 Détection des commutations instantanées

La réinitialisation des variables permet de mettre en évidence un phénomène dont le traitement inadéquat peut contribuer à dégrader la précision de la simulation. Ce phénomène est la détection des commutations simultanées. Une situation de commutations simultanées se produit lorsque le changement d'état d'un interrupteur à commutation forcée provoque la réalisation d'une condition de fermeture ou d'ouverture d'un autre interrupteur à commutation naturelle du même circuit au même instant. Pour illustrer ce phénomène, considérons le circuit de la Figure 4.6.

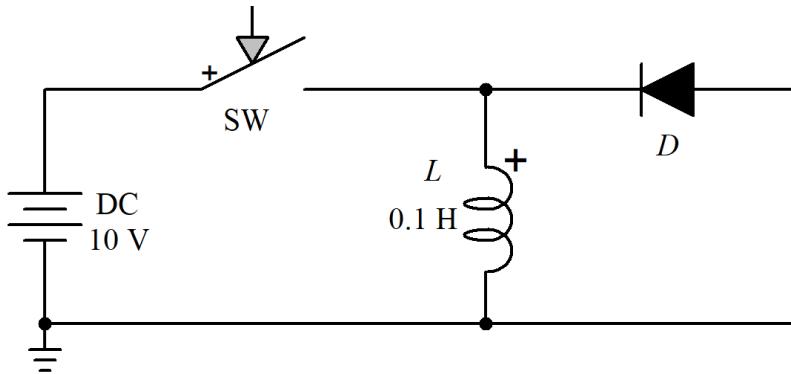


Figure 4.6: Circuit servant à illustrer les situations de commutations simultanées

Ce circuit contient un interrupteur SW commandé et une diode idéale D. Supposons que SW soit initialement en position fermée. La cathode de la diode se trouve au potentiel +10 V de la source (potentiel le plus élevé du circuit). La diode est donc bloquée. La tension de la source s'applique directement aux bornes de l'inductance L et le courant qui y circule croît linéairement en fonction du temps. Supposons qu'un signal de commande entraîne l'ouverture de SW. A cet instant précis, la forte diminution du courant dans la bobine fait apparaître à ses bornes une tension induite qui s'oppose à la variation. La polarité positive de cette tension induite se trouvant à la borne inférieure de l'inductance, la tension aux bornes de la diode devient donc subitement positive, ce qui entraîne son entrée en conduction immédiate. Le courant circulant dans la diode est directement égal à la valeur du courant dans la bobine au moment de l'ouverture de SW.

Reproduire fidèlement ce comportement par la simulation nécessite d'ouvrir SW et de fermer D au même instant. Pour rendre cela possible dans un algorithme de simulation, il faut intégrer une procédure qui exécute les étapes suivantes chaque fois qu'une commutation a lieu :

1. Modifier la matrice du circuit (prise en compte de la commutation);
2. Résoudre le système d'équations au même point de temps;
3. Vérifier à partir de la nouvelle solution si la condition de changement d'état d'un interrupteur à commutation naturelle est satisfaite. Dans l'affirmative, retourner à l'étape 1 et reprendre le processus.

Cette procédure décrit une technique nommée « Simultaneous Switching » ou SS [5]. Il est possible de l'intégrer à la méthode TR_BE, ce qui permet d'obtenir la méthode désignée TR_BE_SS.

Pour revenir à l'exemple de la Figure 4.6, si la technique SS n'est pas utilisée, les résultats de la simulation seront erronés. Pour s'en convaincre, il suffit de simuler le circuit avec la méthode TR_BE, puis avec la méthode TR_BE_SS. Le pas d'intégration sélectionné est $\Delta t = 100\mu s$. SW est initialement fermé et reçoit un ordre d'ouverture à l'instant $t_{sw} = 5.2 ms$. La forme d'onde du courant dans la bobine pour les deux méthodes testées est présentée à la Figure 4.7.

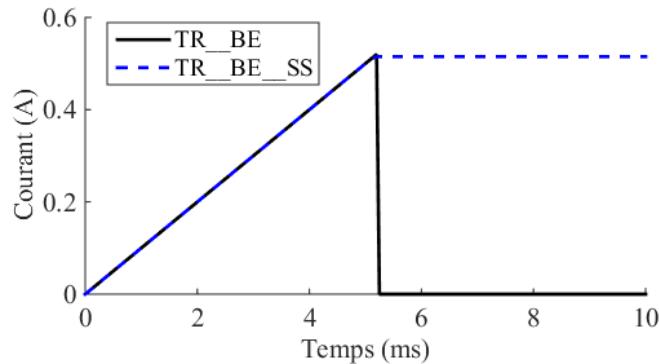


Figure 4.7: Formes d'onde du courant dans l'inductance L pour les méthodes testées

On peut constater que lorsque la technique SS n'est pas appliquée (méthode TR_BE), le courant dans l'inductance tombe à 0 dès l'ouverture de SW et conserve cette valeur, ce qui ne traduit pas le comportement réel du circuit. Cependant, l'application de la technique SS (méthode TR_BE_SS) permet d'obtenir une forme d'onde conforme au comportement réel. Cet exemple démontre

combien la prise en compte des commutations simultanées à travers la technique SS impacte fortement les résultats de la simulation.

En résumé, la localisation précise des instants de discontinuité et des valeurs des variables à ces instants, la suppression des oscillations numériques résultantes de l'intégration TR et la réinitialisation des variables sont des évènements qui impactent la précision dans une simulation. A cela, il est pertinent d'ajouter la problématique du retour ou non sur le time-mesh initial de la simulation (après la détection d'une discontinuité par interpolation). Cette étape accessoire peut se faire en variant le pas d'intégration [94] ou alors en exécutant une interpolation ou une extrapolation additionnelle [95]. Chacune de ces techniques peut impacter la précision. En particulier, les techniques d'interpolation/extrapolation jouent un rôle majeur dans la gestion des discontinuités. Il en existe plusieurs. La façon de les combiner à l'intégration TR est également source d'erreurs. A cette étape, une analyse du comportement de l'erreur dans la simulation dépendamment des techniques utilisées est nécessaire. C'est ce qui sera fait dans la prochaine partie.

4.2 Analyse du comportement de l'erreur selon les techniques d'interpolation

4.2.1 Comparaison entre les techniques d'interpolation et les techniques d'extrapolation

Comme il a été vu précédemment, les techniques d'interpolation et d'extrapolation permettent de détecter avec précision les instants d'apparition des discontinuités et/ou de calculer les variables du circuit à ces instants. Cette partie a pour but de comparer les niveaux de précision obtenus à partir de chacune de ces techniques. Pour cela, ces dernières seront appliquées à la simulation du circuit de la Figure 4.8. Ce circuit contient 4 diodes idéales qui sont des interrupteurs électroniques à commutation naturelle modélisés comme indiqué à la Figure 3.8. Chaque diode entre en conduction lorsque la tension à ses bornes devient supérieure à 0.7 V ($V_{seuil} = 0.7$) et bloque lorsque le courant qui la traverse devient inférieur ou égal à zéro ($I_{seuil} = 0$). Chacun de ces évènements conduit à une discontinuité qui peut être étudiée à partir du time-mesh de la Figure 4.1.

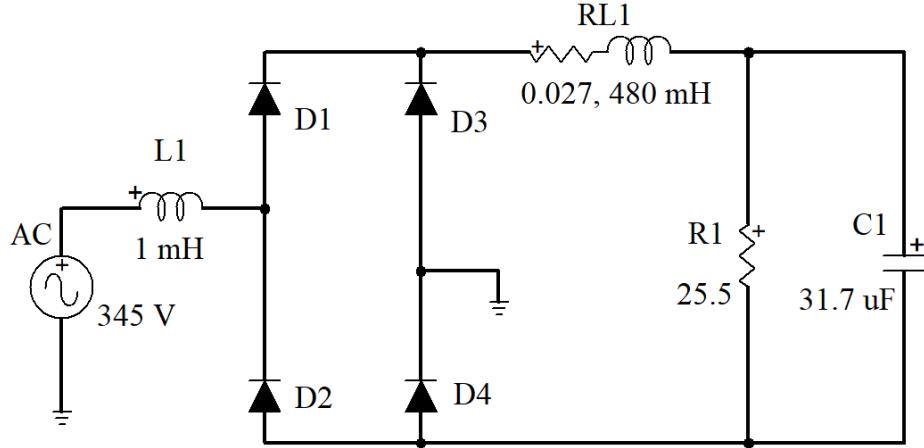


Figure 4.8: Circuit test pour comparer les techniques d'interpolation et d'extrapolation

Ainsi, pour la discontinuité qui a lieu à $t_{sw} \in [t, t + \Delta t]$, l'instant t_{sw} peut être déterminé par application de l'une des 4 techniques suivantes:

- L'interpolation linéaire à partir des solutions à t et à $t + \Delta t$, en utilisant (4.13);
- L'interpolation quadratique à partir des solutions à $t - \Delta t$, t et $t + \Delta t$ avec (4.15);
- L'extrapolation linéaire, qui consiste à prédire la solution à l'instant $t + \Delta t$ par (4.19), puis à calculer t_{sw} par (4.13);
- L'extrapolation quadratique, qui consiste à prédire la solution à l'instant $t + \Delta t$ par (4.20), puis à calculer t_{sw} par (4.15).

Dans un premier temps, le circuit de la Figure 4.8 est simulé avec la méthode TR_BE, en utilisant un pas d'intégration très petit et égal à $\Delta t = 100\text{ ns}$. Tout au long de cette simulation (qui sera la référence), chaque fois qu'une diode change d'état, l'instant de changement t_{swR} est enregistré (qu'importe la diode qui commute). On désigne par t_{swR}^i , le i -ème instant de changement d'état enregistré. De même, on désigne par n le nombre total de changements d'états enregistrés durant cette simulation de référence.

Dans un second temps, plusieurs simulations du même circuit sont effectuées avec la méthode TR_BE, en utilisant différents pas d'intégration numérique. Pendant chaque simulation, les différents instants de changement d'état des diodes sont déterminés en utilisant chacune des 4

techniques recensées plus haut. Pour une technique donnée, le i -ème instant trouvé est désigné t_{sw}^i . Ce dernier est comparé à t_{swR}^i , la valeur de même rang obtenue par la simulation de référence. Pour chacune des 4 techniques de calcul des instants de discontinuité, la valeur absolue de l'erreur relative moyenne sur la précision dans la détection est calculée via la formule :

$$e_{mean} = \frac{1}{n} \sum_{i=1}^n \frac{|t_{sw}^i - t_{swR}^i|}{t_{swR}^i} \times 100 \quad (4.33)$$

En synthèse, pour chaque pas d'intégration numérique sélectionné, on aura 4 résultats de e_{mean} : le premier correspond à l'interpolation linéaire, le deuxième correspond à l'interpolation quadratique, le troisième correspond à l'extrapolation linéaire et enfin le quatrième correspond à l'extrapolation quadratique. Ces informations sont représentées graphiquement sur la Figure 4.9 dans laquelle on retrouve les erreurs relatives de chaque technique en fonction du pas d'intégration.

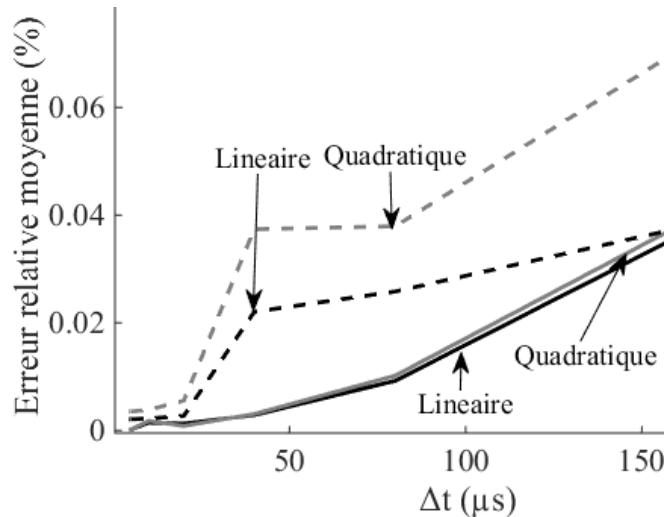


Figure 4.9: Erreurs causées par l'interpolation (traits continus) et l'extrapolation (traits discontinus) en fonction du pas d'intégration

Ces résultats montrent que l'interpolation linéaire est la technique la plus précise pour la prédiction des instants de discontinuité, quel que soit le pas d'intégration numérique. L'écart de précision avec la technique d'interpolation quadratique est très faible. De ce fait, on peut conclure que l'interpolation linéaire est une technique moins complexe qui fournit des résultats plus précis que ceux obtenus avec l'interpolation quadratique. Il faut préciser toutefois que l'interpolation quadratique discutée ici n'utilise pas de point intermédiaire entre deux solutions consécutives (pour la simple raison que ce point n'est pas disponible avec la méthode TR). Avec une méthode telle

que 2S-DIRK ou TR-BDF2, le point intermédiaire est disponible et l'application de l'interpolation quadratique en utilisant ce point milieu peut conduire à une conclusion différente lorsque la technique est comparée à l'interpolation linéaire.

La Figure 4.9 montre également que les techniques d'extrapolation (traits discontinus) ont une précision très mauvaise. Ceci est dû au fait que ces techniques effectuent une prédiction de la solution au point de temps suivant. En réalité, il est très difficile de prédire les valeurs d'une variable avec une bonne précision. Puisque la prédiction n'est donc pas précise, la valeur calculée de l'instant de discontinuité ne sera également pas précise.

Il ressort de cette analyse que l'interpolation linéaire est la meilleure technique de détermination de l'instant d'apparition d'une discontinuité. Toutefois, dans les algorithmes qui utilisent l'intégration TR, l'interpolation linéaire est combinée aux autres techniques de simulation de différentes façons. Les impacts sur la précision peuvent différer selon la façon employée, ce qui motive la prochaine partie.

4.2.2 Impact de l'interpolation linéaire sur la précision des résultats pour différents algorithmes de simulation

4.2.2.1 Mise en contexte

4.2.2.1.1 Motivation de l'étude

Il a été précédemment établi que l'interpolation linéaire est une technique simple et efficace pour la détection précise des discontinuités. Cette approche peut être implémentée de plusieurs façons différentes dans les algorithmes. Avec la méthode TR_BE, l'interpolation peut permettre de calculer les instants d'apparition des discontinuités et de calculer les valeurs des variables à ces instants. L'interpolation linéaire peut également être directement combinée à la méthode TR dans le but de supprimer les oscillations numériques, ce qui permet d'obtenir ce que l'on a appelé « chatter removal technique ». Par ailleurs, puisque l'exécution de l'interpolation amène la simulation sur un point de temps qui n'appartient pas au time-mesh, l'interpolation peut également être utilisée pour resynchroniser la simulation au time-mesh. Chacune de ces possibilités impacte différemment sur la précision des différents algorithmes qui l'utilisent. Une analyse profonde de

cet impact est donc nécessaire. Pour y parvenir, il faut évaluer pour différents algorithmes, l'erreur numérique créée après la traversée d'une discontinuité.

4.2.2.1.2 Quantification de l'erreur

Il existe plusieurs types d'erreur permettant d'apprécier la précision dans une simulation. L'erreur locale $e(t_k)$ sur une variable x à un instant donné t_k est égal à la différence entre sa valeur numérique x_{t_k} et sa valeur exacte $x(t_k)$ au même instant :

$$e(t_k) = x_{t_k} - x(t_k) \quad (4.34)$$

La valeur exacte $x(t_k)$ peut être obtenue via une simulation de référence ou alors elle peut être calculée analytiquement. La relation (4.34) permet d'exprimer l'erreur relative locale :

$$e_r(t_k) = e(t_k)/x(t_k) \quad (4.35)$$

L'erreur locale et l'erreur relative locale sont des valeurs ponctuelles. Elles ne sont généralement pas suffisantes pour évaluer la précision. Pour une simulation qui évolue entre les points t_{start} et t_{end} , l'erreur doit être évaluée à chaque instant $t_k \in [t_{start}, t_{end}]$. Pour le faire, on définit la fonction d'erreur pour chaque instant t_k et on quantifie l'erreur résultante par l'erreur relative RMS (RRE) :

$$e_{rms} = \frac{1}{x_{rms}} \sqrt{\frac{1}{n_s} \sum_{k=1}^{n_s} e(t_k)^2} \quad (4.36)$$

Dans cette relation, x_{rms} est la valeur RMS de la forme d'onde de la solution exacte et n_s est le nombre de points de simulation compris entre t_{start} et t_{end} .

4.2.2.2 Analyse du comportement de l'erreur

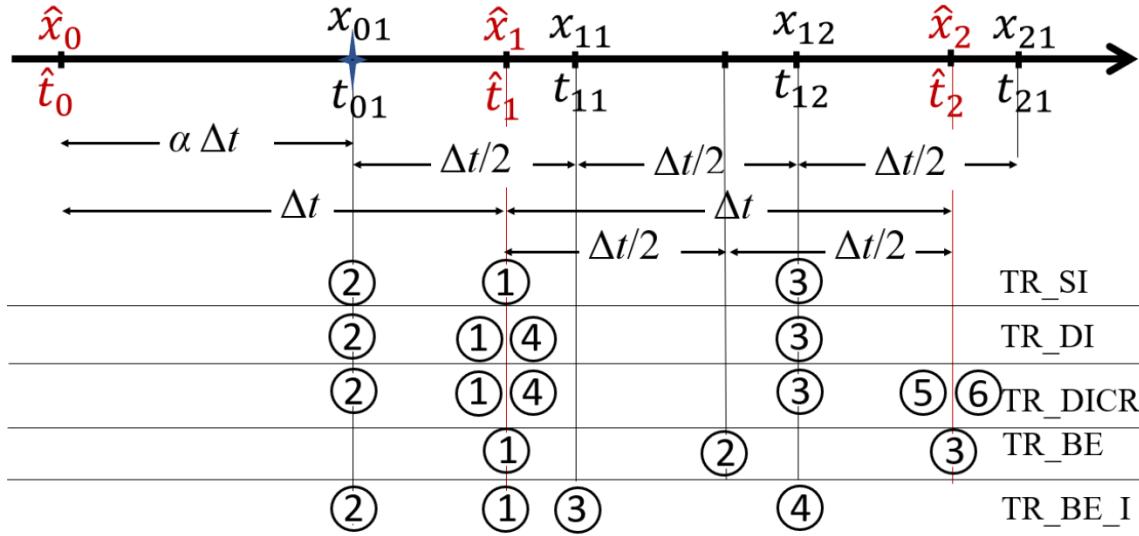


Figure 4.10: Diagramme illustratif de quelques techniques de gestion des discontinuités

La Figure 4.10 présente différentes techniques de gestion des discontinuités construites à partir de l'intégration TR. Chacune de ces techniques sera décrite plus bas et utilisée pour évaluer numériquement les valeurs à différents instants de la fonction x tirée de [12] et définie par :

$$\dot{x} = \lambda x \quad (4.37)$$

avec $\text{Re}(\lambda) < 0$. Si x_0 est la valeur initiale, alors la solution exacte de (4.37) est :

$$x(t) = x_0 e^{\lambda t} \quad (4.38)$$

Sur la Figure 4.10, les variables du time-mesh sont écrites en rouge et avec un chapeau. La simulation commence à partir de la valeur \hat{x}_0 à \hat{t}_0 . On peut supposer que $\hat{t}_0 = 0$. Une discontinuité a lieu à t_{01} , ce qui permet de déterminer $\alpha = (t_{01} - \hat{t}_0) / (\hat{t}_1 - \hat{t}_0) = (t_{01} - \hat{t}_0) / \Delta t$. La discréétisation TR décrite par (2.4) et appliquée à (4.37) permet de calculer la solution numérique à \hat{t}_1 :

$$\hat{x}_1 = K_0 x_0 = K_0 \hat{x}_0 \quad (4.39)$$

Avec $K_0 = \left(1 + \lambda \frac{\Delta t}{2}\right) / \left(1 - \lambda \frac{\Delta t}{2}\right)$. Ce premier calcul est commun à toutes les techniques de gestion des discontinuités qui seront analysées. Dans la suite, le fonctionnement de chacune des

techniques illustrées à la Figure 4.10 sera décrit. L'erreur sur le calcul de la variable au premier point de temps suivant la discontinuité sera évaluée.

4.2.2.2.1 Technique TR_SI

Cette technique est la plus basique. Elle consiste simplement à exécuter une interpolation linéaire pour déterminer la solution au point de discontinuité t_{01} , ce qui donne :

$$x_{01} = (1 - \alpha + \alpha K_0) \hat{x}_0 = K_1 \hat{x}_0 \quad (4.40)$$

Ensuite, la solution au prochain point de temps $t_{12} = t_{01} + \Delta t$ est calculée par intégration trapézoidale :

$$x_{12} = K_0 K_1 \hat{x}_0 = K_2 \hat{x}_0 \quad (4.41)$$

La simulation se poursuit ainsi par intégration trapézoidale. Le premier point de temps après application de la technique de gestion de la discontinuité est donc le point t_{12} . Cette méthode ne retourne pas sur le time-mesh initial. En combinant (4.41), (4.40) et (4.39), puis en utilisant (4.34) et (4.38), l'erreur locale de la technique TR_SI est donné par :

$$e_{TR_SI} = [(1 - \alpha)K_0 + \alpha K_0^2 - e^{\lambda(1+\alpha)\Delta t}] \hat{x}_0 \quad (4.42)$$

4.2.2.2.2 Technique TR_DI

Cette technique est semblable à la technique TR_SI à la seule différence qu'une dernière étape de resynchronisation au time-mesh en \hat{t}_1 est exécutée à partir des solutions à t_{12} et t_{01} :

$$\hat{x}_1 = [\alpha K_1 + (1 - \alpha) K_2] \hat{x}_0 = K_3 \hat{x}_0 \quad (4.43)$$

En combinant (4.43), (4.41), (4.40) et (4.39), puis en utilisant (4.34) et (4.38), l'erreur locale de la technique TR_DI, évaluée à \hat{t}_1 est donnée par :

$$e_{TR_DI} = \left[\alpha(1 - \alpha) + \left[(1 - \alpha)^2 + \alpha^2 \right] K_0 + \alpha(1 - \alpha) K_0^2 - e^{\lambda \Delta t} \right] \hat{x}_0 \quad (4.44)$$

4.2.2.2.3 Technique TR_DICR

Les techniques TR_SI et TR_DI ont un inconvénient commun. Elles ne permettent pas de supprimer les oscillations numériques causées par la méthode TR. La technique TR_DICR complète la technique TR_DI en y ajoutant la technique « chatter removal ». Elle consiste, à partir de la solution donnée par (4.43), à exécuter une intégration pour se rendre à \hat{t}_2 . La solution obtenue est $\hat{x}_2 = K_0 K_3 x_0$. Ensuite, une interpolation avec $\alpha = 0.5$ est exécutée à partir des solutions à \hat{t}_1 et \hat{t}_2 . Le résultat représentera la solution finale à \hat{t}_2 , soit :

$$\hat{x}_2 = 0.5(K_3 + K_0 K_3)\hat{x}_0 = K_4 \hat{x}_0 \quad (4.45)$$

Finalement, l'erreur sur le calcul de la solution à \hat{t}_2 est :

$$e_{TR_DICR} = \left[0.5 [1 - \alpha + \alpha K_0] [\alpha + (1 - \alpha) K_0] (1 + K_0) - e^{\lambda 2\Delta t} \right] \hat{x}_0 \quad (4.46)$$

Il est important de préciser qu'il existe une autre façon de combiner la technique « chatter removal » avec l'intégration TR [92]. Elle consiste à exécuter l'interpolation avec $\alpha = 0.5$ à partir des solutions à t_{01} et t_{12} calculées par (4.40) et (4.41) pour obtenir la solution à t_{11} . Ensuite, une intégration est effectuée pour calculer la solution à t_{21} . Finalement, l'interpolation linéaire est exécutée à partir des solutions à t_{11} et t_{21} pour obtenir la solution finale à \hat{t}_1 . Cette variante de la méthode TR_DICR contient les mêmes étapes mais exécutées dans un ordre différent. Les deux approches produisent des résultats similaires, leur objectif commun étant de supprimer les oscillations numériques.

4.2.2.2.4 Technique TR_BE

Cette technique n'utilise pas d'interpolation. La discontinuité qui se produit à t_{01} est détectée à l'instant \hat{t}_1 . Ensuite, deux intégrations Backward Euler (BE) avec un pas $\Delta t/2$ sont exécutées pour supprimer les oscillations numériques. L'erreur due à la détection imprécise de la discontinuité représente la différence entre \hat{x}_1 dans (4.39) et la solution exacte calculée à t_{01} :

$$e_{TR_BE} = \left[K_0 - e^{\lambda \alpha \Delta t} \right] \hat{x}_0 \quad (4.47)$$

4.2.2.2.5 Technique TR_BE_I

Cette technique est similaire à la technique TR_BE à la seule différence que l'interpolation linéaire est exécutée avant l'application des deux intégrations Backward Euler. Pour évaluer l'erreur juste après la traversée des discontinuités, il faut partir de (4.40) et exécuter une première intégration Backward Euler pour se déplacer de t_{01} à t_{11} . L'intégration BE est donnée par (2.3). Appliquée à (4.37), elle permet d'obtenir :

$$x_{11} = K_5 x_{01} \quad (4.48)$$

Avec $K_5 = \left(1 - \lambda \frac{\Delta t}{2}\right)^{-1}$. Ensuite, une seconde intégration BE est exécutée pour calculer la solution à t_{12} :

$$x_{12} = K_5 x_{11} = K_5^2 x_{01} \quad (4.49)$$

L'erreur de la technique TR_BE_I, évaluée à t_{12} est :

$$e_{TR_BE_I} = \left[K_5^2 (1 - \alpha + \alpha K_0) - e^{\lambda(1+\alpha)\Delta t} \right] \hat{x}_0 \quad (4.50)$$

4.2.2.2.6 Comparaison des techniques

Les équations (4.42), (4.44), (4.46), (4.47) et (4.50) indiquent que le moment précis auquel se produit une discontinuité (défini par α) influence la valeur de l'erreur locale juste après la traversée de la discontinuité. L'erreur est aussi dépendante de λ (via K_0) et de Δt . La valeur de λ est une caractéristique qui traduit la dynamique du circuit simulé. Les équations susmentionnées permettent de comparer les techniques de gestion des discontinuités. Ainsi, les formes d'onde des erreurs locales de chaque technique, calculées via (4.42), (4.44), (4.46), (4.47) et (4.50), pour différentes valeurs de α entre 0 et 1 sont représentées à la Figure 4.11. Ces formes d'onde ont été obtenues pour les valeurs arbitraires suivantes : $\hat{x}_0 = 1$, $\lambda = -10^4$ et $\Delta t = 50\mu s$.

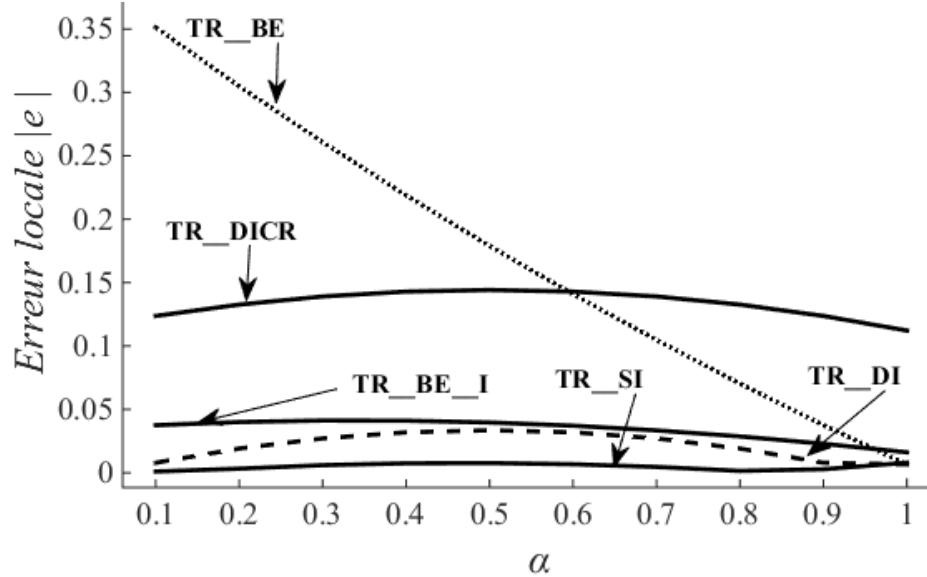


Figure 4.11: Erreur locale des techniques testées en fonction de alpha

Ces résultats permettent de tirer les conclusions suivantes :

- L’erreur créée par chaque technique n’est pas toujours la même. Elle peut varier dépendamment du moment précis où la discontinuité se produit entre deux points de temps consécutifs (défini par α) ;
- En comparant les courbes de TR_SI et de TR_DI on peut conclure que la resynchronisation au time-mesh (TR_DI) crée une erreur plus grande que l’erreur en l’absence de resynchronisation (TR_SI). C’est tout à fait normal puisque chaque fois qu’on interpole, on crée une erreur. TR_SI interpole une seule fois tandis que TR_DICR interpole deux fois. De ce fait, si dans une simulation il n’est pas requis de produire des solutions à intervalles de temps fixes (comme c’est le cas en simulation temps réel), alors la resynchronisation au time-mesh après chaque discontinuité devrait simplement être évitée. Cette remarque est valable particulièrement pour les logiciels de simulation en temps différé ;
- En comparant TR_BE_I et TR_DICR on peut observer que l’erreur avec TR_BE_I est plus faible qu’avec TR_DICR. On peut donc en conclure qu’utiliser la méthode BE pour supprimer les oscillations numériques est une meilleure approche que l’utilisation de chatter removal technique ;
- Avec TR_BE, l’erreur varie directement selon l’instant de discontinuité ;

- L'intersection entre la courbe de TR_BE et la courbe de TR_DICR révèle que la méthode TR_BE qui n'utilise pas l'interpolation pour la détection précise des discontinuités peut parfois produire une erreur plus faible que celle obtenue avec TR_DICR. Cette observation est contre intuitive, puisqu'on s'attendrait à ce qu'une méthode qui interpole soit en tout temps plus précise qu'une méthode qui n'interpole pas. Cette observation avait été faite dans [56] sans fournir d'explication. L'analyse présentée dans cette partie permet donc de comprendre pourquoi ;
- La méthode TR_BE_I est la technique la plus précise et donc la meilleure approche.

4.2.2.2.7 Validation à la simulation de deux circuits

Les techniques TR_BE, TR_DICR et TR_BE_I sont utilisées pour simuler deux circuits dans l'objectif de valider les résultats d'analyse présentés ci-dessus. Il faut préciser que les techniques TR_SI et TR_DI ne sont pas implémentables puisqu'elles ne suppriment pas les oscillations numériques. Elles ont été présentées juste à titre démonstratif pour comparer les erreurs qui se créent lorsque l'on resynchronise la simulation au time-mesh et lorsqu'on ne le fait pas. Le premier circuit simulé est un simple circuit RLC, représenté à la Figure 4.12.

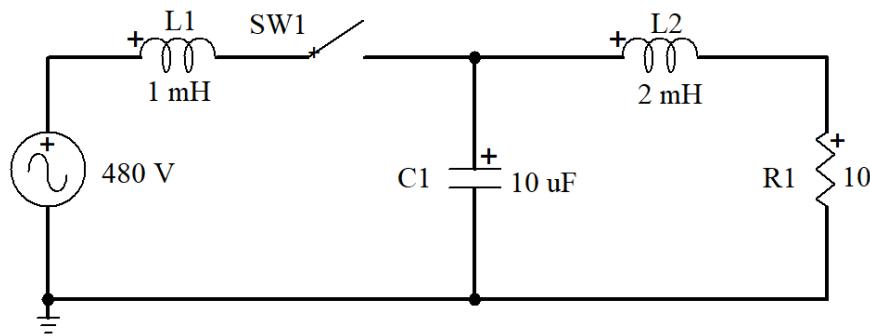


Figure 4.12: Circuit RLC simple servant à illustrer le comportement de l'erreur

Un ensemble de simulations sont exécutées en utilisant les techniques testées et un pas d'intégration $\Delta t = 10\mu s$. Pour chaque technique, plusieurs simulations sont effectuées. Dans chacune d'elles, l'interrupteur SW1 est fermé à un instant particulier $t_d = t_n + \alpha\Delta t$ situé entre l'instant $t_n = 30ms$ et l'instant $t_{n+1} = 30ms + \Delta t$. Le paramètre α prend une valeur comprise entre 0 à 1. Par exemple, pour l'une des simulations, on aura $\alpha = 0.3$, ce qui signifie que SW1 sera fermé à l'instant

$t_d = (30ms) + 0.3(10\mu s) = 30.003ms$. Pour une autre simulation, α prendra une autre valeur de façon à parcourir toutes les valeurs intermédiaires entre 0 et 1 par pas de 0.1.

A la suite de chaque simulation, la valeur de l'erreur relative RMS (RRE) sur le courant dans l'inductance L1, exprimée par (4.36) est calculée. Ce calcul est exécuté pour tous les points de simulation entre 30ms et 40ms (durée du comportement transitoire à la suite de la manœuvre de SW1). Pour chaque valeur de α , une simulation de référence avec TR_BE et un pas d'intégration suffisamment petit ($1\mu s$) est effectuée afin d'obtenir les valeurs de référence pour le calcul de RRE. Les résultats issus des différents calculs d'erreur sont représentés à la Figure 4.13.

On peut clairement constater que les profils d'erreur obtenus pour les méthodes testées sont parfaitement en accord avec les prédictions présentées à la Figure 4.11. En particulier, on peut bien constater que lorsque $\alpha > 0.5$, la méthode TR_BE devient plus précise que la méthode TR_DICR qui utilise pourtant l'interpolation pour la détection précise des discontinuités. Aussi, on peut constater que la méthode TR_BE_I est en tout temps la plus précise.

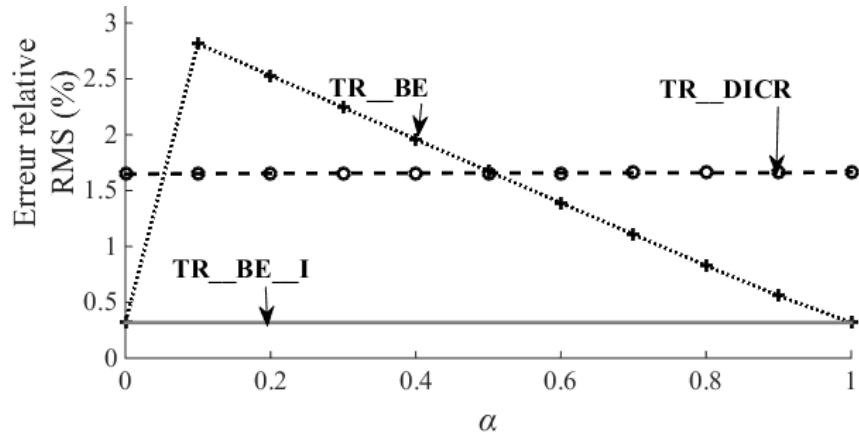


Figure 4.13: Erreur relative RMS sur le courant dans l'inductance L1 pour les techniques testées

Dans cet exemple, la valeur de α est définie par l'utilisateur qui peut choisir les instants de fermeture de SW1. Il serait intéressant de voir si des résultats similaires peuvent être obtenus en simulant un circuit pour lequel les instants de commutation (donc les valeurs de α) ne sont pas choisies par l'utilisateur. Pour cela, on va considérer le circuit de la Figure 4.14, tiré de [56]. Les instants de commutation seront déterminés par la procédure présentée à la section 4.1.1.2.

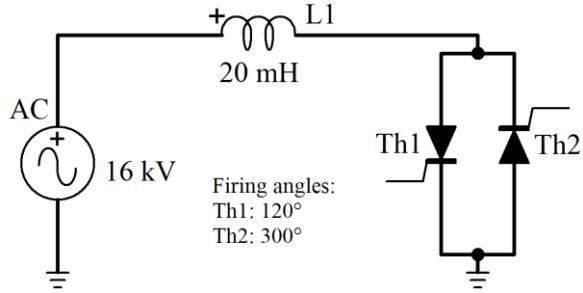


Figure 4.14: Circuit simple à thyristors

Chaque thyristor est modélisé comme à la Figure 3.8, à la seule différence que l'interrupteur SW est un interrupteur commandé en fermeture par un signal d'allumage. Ce signal (qui est une impulsion) provoque la fermeture de SW si la tension aux bornes du thyristor est supérieure à V_{DC} . De l'autre côté, l'ouverture de SW survient simplement lorsque le courant dans le thyristor devient négatif (même condition d'ouverture qu'une diode). Chaque thyristor reçoit un signal d'allumage qui est émis lorsque la phase d'une référence sinusoïdale atteint la valeur « *firing angle* » spécifiée à la Figure 4.14 (120° pour Th1 et 300° pour Th2).

Une simulation de référence est effectuée avec TR_BE et un pas d'intégration de $1\mu s$. Ensuite, le circuit est simulé avec TR_BE, TR_BE_I, TR_DICR et un pas d'intégration $\Delta t = 50\mu s$. La fonction d'erreur du courant dans l'inductance L1 est évaluée pour chaque méthode en calculant à chaque point de la simulation, la valeur absolue de l'erreur donnée par l'équation (4.34). Les formes d'onde obtenues pour chaque méthode sont présentées à la Figure 4.15.

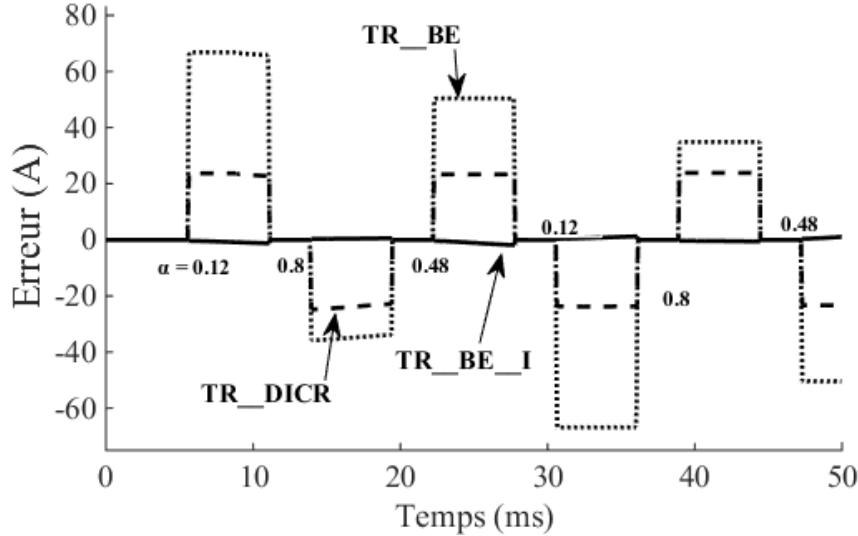


Figure 4.15: Erreur dans le courant de l'inductance

Sur la Figure 4.15, chaque fois qu'un thyristor passe de l'état « ouvert » à l'état « fermé », la valeur correspondante de α est calculée et affichée. Avec TR_BE, plus la valeur de α est petite, plus l'erreur après la fermeture du thyristor est élevée. Aussi, avec TR_DICR, l'erreur varie très peu en fonction de α . Finalement, on constate que la méthode TR_BE_I est celle qui produit l'erreur la plus petite (de l'ordre de 2A). C'est donc la méthode la plus précise. Toutes ces observations sont parfaitement en accord avec la prédition théorique de la Figure 4.11.

Il ressort de toute l'analyse présenté ci-dessus que, le moment auquel apparaît une discontinuité peut avoir un impact sur la précision dépendamment de la technique de gestion de la discontinuité. Dans des cas réels de simulation, les valeurs prises par α lors des discontinuités sont complètement imprédictibles. C'est pourquoi, la précision d'une méthode par rapport à une autre peut parfois être une question de probabilités. Les résultats des analyses présentées dans ce chapitre fournissent des éléments indispensables pour le développement de nouvelles méthodes de simulation.

CHAPITRE 5 MÉTHODES PRÉCISES DE SIMULATION DES CIRCUITS D'ELECTRONIQUE DE PUISSANCE

Le chapitre précédent a permis d'identifier les principales sources d'erreur qui peuvent apparaître lors de la simulation des circuits d'électronique de puissance. Des analyses de l'impact sur l'erreur des techniques employées dans les algorithmes existants ont été réalisées. Dans ce chapitre, il sera question d'abord de définir les caractéristiques essentielles qu'un algorithme évolué dédié à la simulation des circuits d'électronique de puissance doit nécessairement posséder, ceci au regard des observations et des conclusions énoncées au chapitre précédent. Il sera question par la suite de proposer trois nouvelles méthodes qui intègrent ces caractéristiques essentielles, puis les tester sur des circuits d'électronique de puissance. Ces méthodes sont présentées dans [96] par l'auteur de cette thèse.

5.1 Fondements des nouvelles méthodes de simulation

5.1.1 Les caractéristiques d'un algorithme évolué dédié à la simulation des circuits d'électronique de puissance

Dans cette section, les caractéristiques d'un algorithme de simulation dit « évolué » (selon notre appréciation) seront présentées. Le modèle d'algorithme intégrant ces caractéristiques servira « d'avatar » ou alors de modèle de base pour le développement des nouvelles méthodes de simulation qui seront présentées par la suite.

D'entrée de jeu, il faut préciser que l'intégration TR pour la simulation dans le domaine du temps est un excellent choix. Cette méthode d'intégration numérique est très populaire pour sa très bonne précision et sa simplicité d'implémentation. Il est donc normal d'assumer qu'un algorithme de simulation évolué exécutera l'intégration TR comme méthode de base. Cette considération présente l'avantage majeur de faciliter la fusion entre tout nouvel algorithme développé et un algorithme de simulation existant (puisque les algorithmes existants utilisent pour la majorité, la méthode TR). Le principal inconvénient de la méthode TR étant l'apparition d'oscillations numériques lorsque des discontinuités se produisent, il est obligatoire de compléter la méthode TR par une technique de suppression des oscillations numériques. Il a été observé au chapitre précédent que la technique « Critical Damping Adjustment » (CDA) qui consiste à utiliser l'intégration BE

donne de meilleurs résultats que la technique « Chatter Removal Technique » qui consiste à exécuter une interpolation à mi-chemin entre deux points de temps consécutifs. Pour la suppression des oscillations numériques, l'algorithme évolué utilisera la technique CDA. Finalement, la méthode TR_BE sera la méthode de simulation de base de notre algorithme évolué, à partir de laquelle seront bâties les nouvelles méthodes présentées plus bas.

Pour la détection des instants de discontinuités et le calcul des variables à ces instants, l'algorithme évolué devra utiliser la technique d'interpolation linéaire. Elle est simple et efficace, comme cela a été prouvé au chapitre précédent. L'algorithme évolué devra également intégrer une technique de réinitialisation des variables du réseau, de détection et de gestion des situations de commutations instantanées. Puisque l'intégration BE est appliquée après chaque discontinuité, la réinitialisation n'est donc pas nécessaire dans notre algorithme évolué. En effet, pour le calcul de la solution à un instant donné, la méthode BE n'utilise que les valeurs des variables d'état de l'instant précédent. Ces variables ne pouvant subir des variations soudaines, l'erreur causée par l'inexactitude des termes historiques sur le calcul de la solution au premier point de temps consécutif à une discontinuité sera donc évitée. À présent, pour la prise en compte des situations de commutations instantanées, l'algorithme évolué devra intégrer la technique « simultaneous switching » (SS) qui consiste après chaque discontinuité, à résoudre le système d'équations du circuit dans sa nouvelle configuration au point de temps de la discontinuité et vérifier si les valeurs des variables obtenues n'autorisent pas d'autres changements d'état d'autres interrupteurs du circuit. Dans l'affirmative, le système est de nouveau solutionné et la vérification est de nouveau faite, et ainsi de suite jusqu'à ce que l'algorithme ne détecte plus aucune discontinuité, ce qui lui permet d'avancer la simulation au point de temps suivant.

Rendu à ce niveau, il est important de préciser que la combinaison de l'interpolation et de la technique SS dans le même algorithme est une tâche délicate qui peut créer des erreurs additionnelles. La section suivante permet de s'en convaincre.

5.1.2 Analyse d'un algorithme combinant l'interpolation et la technique simultaneous switching

5.1.2.1 Description de l'algorithme

Pour la simulation des convertisseurs d'électronique de puissance, dans [58] il est proposé une méthode qui combine l'interpolation et le traitement des commutations simultanées dans un même algorithme afin d'améliorer la précision. Les effets de cette combinaison sur l'appréciation exacte des pertes dans les interrupteurs électroniques ont été démontrés dans [58]. Cette approche permet en effet d'obtenir des pertes plus réalistes, comparé aux résultats obtenus avec des algorithmes qui utilisent uniquement l'interpolation ou n'interpolent pas du tout. Toutefois, l'impact de la méthode proposée dans [58] sur la précision des formes d'onde de la simulation n'a pas été analysée en profondeur. En effet, un examen du comportement de cette méthode fait ressortir des sources d'erreur. Pour comprendre cela, il faut commencer par décrire le fonctionnement de cette méthode que l'on peut illustrer en utilisant le time-mesh de la Figure 5.1. Ce time-mesh servira également à illustrer les nouvelles méthodes qui seront proposées.

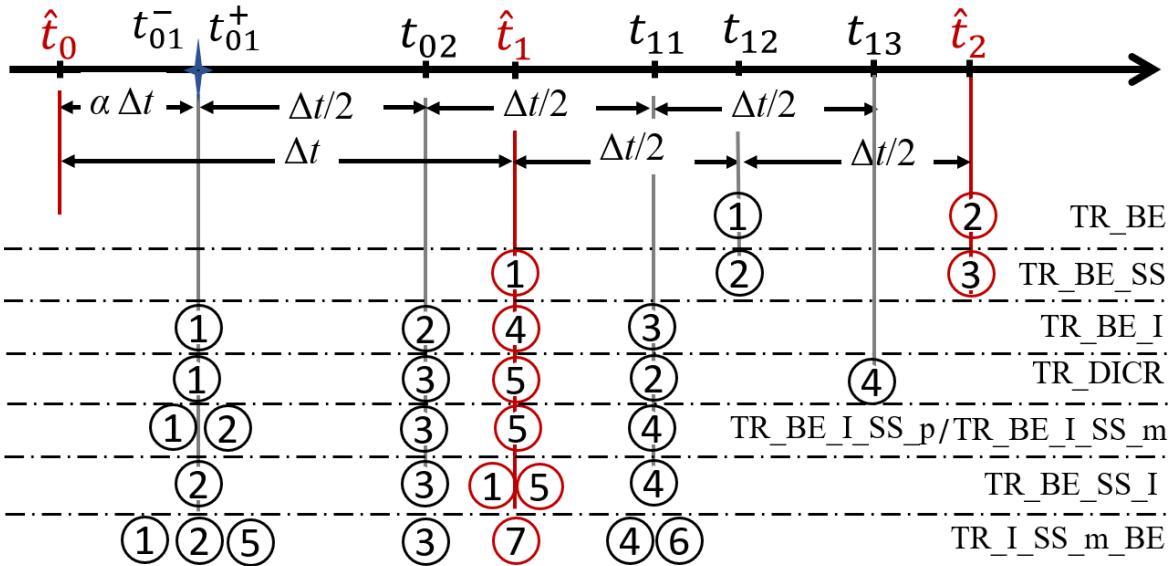


Figure 5.1: Illustration des méthodes discutées

Pour rappel, l'intégration numérique d'un point de temps à un autre se fait par résolution du système d'équations du réseau. Ce système est exprimé par (2.44) qui est reprise ici :

$$\mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t \quad (5.1)$$

L'indice t traduit le fait que chacune de ces matrices peut changer à chaque instant. Lorsque ce système est résolu à l'instant \hat{t}_1 , la solution obtenue pour une variable x est désignée $x_{\hat{t}_1}$. Les termes historiques nécessaires pour la formulation du vecteur $\mathbf{b}_{\hat{t}_1}$ sont obtenus à partir de la solution à \hat{t}_0 . Le pas d'intégration de la simulation est Δt . À \hat{t}_1 , une discontinuité (changement d'état d'un des interrupteur du circuit) est détectée, l'instant exact t_{01} de son apparition est calculé. A partir de ce moment, l'algorithme présenté dans [58] exécute dans l'ordre les étapes suivantes :

1. Les courants et les tensions sont linéairement interpolés à t_{01} (ceci permet d'obtenir la solution à t_{01}^-).
2. La matrice \mathbf{A}_t est modifiée (changement de l'état de l'interrupteur) et la technique SS est exécutée à l'instant t_{01} en utilisant le même vecteur $\mathbf{b}_{\hat{t}_1}$ que celui formulé pour calculer la solution initiale à \hat{t}_1 . Le résultat représente la solution à t_{01}^+ .
3. Une intégration TR est exécutée pour évoluer de t_{01} à t_{11} . Le vecteur $\mathbf{b}_{t_{11}}$ nécessaire à cette intégration est construit à partir de la solution à t_{01}^+ .
4. Les courants et les tensions sont linéairement interpolés afin de ramener la simulation sur le time-mesh à l'instant \hat{t}_1 .

Dans la pratique, il existe une cinquième étape qui consiste à appliquer la technique CRT. Elle consiste à exécuter une intégration pour se rendre à \hat{t}_2 puis une interpolation à mi-parcours entre \hat{t}_1 et \hat{t}_2 pour supprimer les oscillations numériques. Le résultat obtenu représente la solution à \hat{t}_2 . Cette méthode présentée dans [58] a été reproduite pour des besoins de comparaison avec les méthodes qui seront proposées plus bas. Lors de la reproduction, l'étape 3 a été remplacée par deux intégrations BE avec un pas d'intégration $\Delta t/2$. Ceci permet de supprimer les oscillations numériques et empêche d'exécuter CRT. La méthode obtenue est désignée TR_BE_I_SS_p.

5.1.2.2 Les insuffisances de la méthode

Bien que cette méthode utilise l'interpolation pour la détection précise des discontinuités et la technique SS pour l'identification et la prise en compte des situations de commutations simultanées, elle présente un inconvénient majeur. Il faut se rappeler que tel que la matrice \mathbf{A}_t a été formulée, toute résolution du système (5.1) doit correspondre à un saut dans le temps d'une longueur égale à Δt . Malheureusement, ce n'est pas le cas à l'étape 2 de cette méthode. En effet, puisque le vecteur \mathbf{b}_t utilisé est construit à partir de la solution à \hat{t}_1 , le saut dans le temps a plutôt une longueur égale à $\alpha \Delta t$. Le résultat obtenu, qui représente la solution à t_{01}^+ contient donc une erreur qui va se propager aux étapes de calcul suivants et diminuer le niveau de précision de la simulation.

Une solution pour résoudre ce problème consisterait à obtenir les valeurs des variables à l'instant $t_{01} - \Delta t$ et s'en servir pour construire le vecteur \mathbf{b}_t requis à l'étape 2. Toutefois, cette solution supposerait que la discontinuité ait eu lieu à $t_{01} - \Delta t$, ce qui n'est pas le cas. De plus, si d'autres discontinuités ont eu lieu entre $t_{01} - \Delta t$ et \hat{t}_0 , l'interpolation pour le calcul de la solution à $t_{01} - \Delta t$ produirait un résultat erroné. Tout ceci amène à comprendre la complexité qu'il y'a à combiner l'interpolation et la technique SS dans le même algorithme. La suite de ce chapitre présente trois nouvelles méthodes qui combinent ces deux techniques d'une façon plus efficiente.

5.2 Nouvelles méthodes

Les nouvelles méthodes qui seront présentées ci-dessous sont illustrées à la Figure 5.1. Chaque méthode porte un nom qui renseigne sur les techniques de simulation qu'elle exécute. Afin de définir le point de départ commun à partir duquel les étapes de chaque méthode seront présentées, on va considérer que le système (5.1) du circuit simulé est résolu à l'instant \hat{t}_1 et la solution obtenue est \hat{x}_1 . Les termes historiques nécessaires pour la formulation du vecteur \mathbf{b}_t proviennent de la solution à \hat{t}_0 . Le pas d'intégration de la simulation est Δt . À \hat{t}_1 , une discontinuité (changement d'état d'un interrupteur du circuit) est détectée et l'instant exact t_{01} de son apparition est calculé.

5.2.1 Méthode TR_BE_SS_I

Puisqu'il est difficile d'appliquer la technique SS après avoir interpolé (à cause du décalage de la valeur interpolée par rapport au time-mesh), on peut intuitivement envisager la combinaison de l'interpolation et de la technique SS de manière séquentielle. L'algorithme proposé ici applique la technique SS à \hat{t}_1 avant d'interpoler pour ramener la simulation au point t_{01} . A partir de l'instant \hat{t}_1 , point de détection de la discontinuité, la méthode TR_BE_SS_I exécute les étapes suivantes :

1. La matrice A_i est modifiée (changement d'état de l'interrupteur) et la technique SS est exécutée à l'instant \hat{t}_1 en utilisant $b_{\hat{t}_1}$.
2. Les tensions et les courants sont linéairement interpolés au point t_{01} .
3. Deux intégrations BE sont exécutées avec un pas d'intégration de $\Delta t/2$ pour faire évoluer la simulation de t_{01} à t_{11} .
4. Les tensions et les courants sont linéairement interpolés pour resynchroniser la simulation sur le time-mesh à l'instant \hat{t}_1 .

La véritable particularité de cette méthode est qu'elle exécute la technique SS avant d'interpoler. On peut se douter que cet ordre d'exécution des étapes ne permet pas véritablement un calcul suffisamment précis des variables du circuit aux instants exacts d'apparition des discontinuités.

5.2.2 Méthode TR_BE_I_SS_m

Cette méthode est inspirée de la méthode existante TR_BE_I_SS_p. Pour rappel, avec la méthode TR_BE_I_SS_p, l'application de la technique SS produit un résultat erroné à l'instant t_{01}^+ . Ce résultat erroné est utilisé pour constituer le vecteur b_i de la prochaine intégration. La méthode TR_BE_I_SS_m qui est proposée ici reprend les étapes de TR_BE_I_SS_p avec une seule variante : c'est la solution à t_{01}^- qui est utilisée pour constituer le vecteur b_i de la première intégration BE. En effet, la solution à t_{01}^- est obtenue par interpolation, donc avec une précision meilleure que celle obtenue après exécution de SS. De plus, bien que l'application de la technique

SS produise une solution finale erronée à t_{01}^+ , elle permet toutefois de déterminer les états réels (ouvert ou fermé) des interrupteurs du réseau. C'est pour cela que cette technique est maintenue dans la méthode proposée. Par ailleurs, puisque l'intégration BE (qui est l'étape suivante après l'application de SS) utilise les variables d'état pour constituer le vecteur \mathbf{b}_t , le fait d'utiliser la solution à t_{01}^- ne pose donc pas de problème de précision. Pour résumer, les étapes de la méthode proposée TR_BE_I_SS_m à partir de la détection de la discontinuité à \hat{t}_1 seront donc les suivantes :

1. Les courants et les tensions sont linéairement interpolés à t_{01} (ceci permet d'obtenir la solution à t_{01}^-).
2. La matrice \mathbf{A}_t est modifiée (changement de l'état de l'interrupteur) et la technique SS est exécutée à l'instant t_{01} en utilisant le même vecteur $\mathbf{b}_{\hat{t}_1}$ que celui formulé pour calculer la solution initiale à \hat{t}_1 . Le résultat représente la solution à t_{01}^+ .
3. Une intégration BE est exécutée pour évoluer de t_{01} à t_{02} . Le vecteur $\mathbf{b}_{t_{02}}$ nécessaire à cette intégration est construit à partir de la solution à t_{01}^- . Une seconde intégration BE est exécutée pour évoluer de t_{02} à t_{11} .
4. Les courants et les tensions sont linéairement interpolés afin de ramener la simulation sur le time-mesh à l'instant \hat{t}_1 .

5.2.3 Méthode TR_I_SS_m_BE

Cette méthode présente deux spécificités par rapport aux 2 précédentes méthodes proposées. Premièrement, elle propose une façon différente (plus précise) d'obtenir la solution finale à t_{01}^+ . Deuxièmement, elle n'utilise que l'intégration TR pour avancer dans le temps. L'intégration BE est utilisée uniquement pour réinitialiser la simulation à t_{01} . De cette manière on s'attend à ce que la faible précision de la méthode BE ne diminue pas la précision de la simulation comme c'est généralement le cas avec la méthode classique TR_BE. A partir de la détection de la discontinuité à \hat{t}_1 , la méthode TR_I_SS_m_BE fonctionne comme suit :

1. Les courants et les tensions sont linéairement interpolés à t_{01} (ceci permet d'obtenir la solution à t_{01}^-).
2. La matrice \mathbf{A}_t est modifiée (changement de l'état de l'interrupteur) et la technique SS est exécutée à l'instant t_{01} en utilisant le même vecteur $\mathbf{b}_{\hat{t}_1}$ que celui formulé pour calculer la solution initiale à \hat{t}_1 . Le résultat représente la solution à t_{01}^+ .
3. Une intégration BE est exécutée pour évoluer de t_{01} à t_{02} . Le vecteur $\mathbf{b}_{t_{02}}$ nécessaire à cette intégration est construit à partir de la solution à t_{01}^- . Une seconde intégration BE est exécutée pour évoluer de t_{02} à t_{11} .
4. Les tensions et les courants sont linéairement extrapolés en arrière à l'instant t_{01} à partir des solutions calculées à t_{02} et à t_{11} . Le résultat représente la solution du réseau à t_{01}^+ .
5. L'intégration trapézoidale est appliquée pour déplacer la simulation de t_{01} à t_{11} en utilisant le vecteur \mathbf{b}_t construit à partir de la solution à t_{01}^+ .
6. Les courants et les tensions sont linéairement interpolés afin de ramener la simulation sur le time-mesh à l'instant \hat{t}_1 .

5.3 Simulations et analyses

Dans cette partie, les performances des méthodes proposées et des méthodes existantes sont testées à travers la simulation de plusieurs circuits. Chaque circuit simulé est choisi pour mettre en exergue une ou plusieurs problématique(s) liée(s) à la simulation des circuits d'électronique de puissance. Pour rappel, voici une synthèse des méthodes discutées dans cette thèse qui seront testées dans les différents cas (cette synthèse est également présentée en annexe A) :

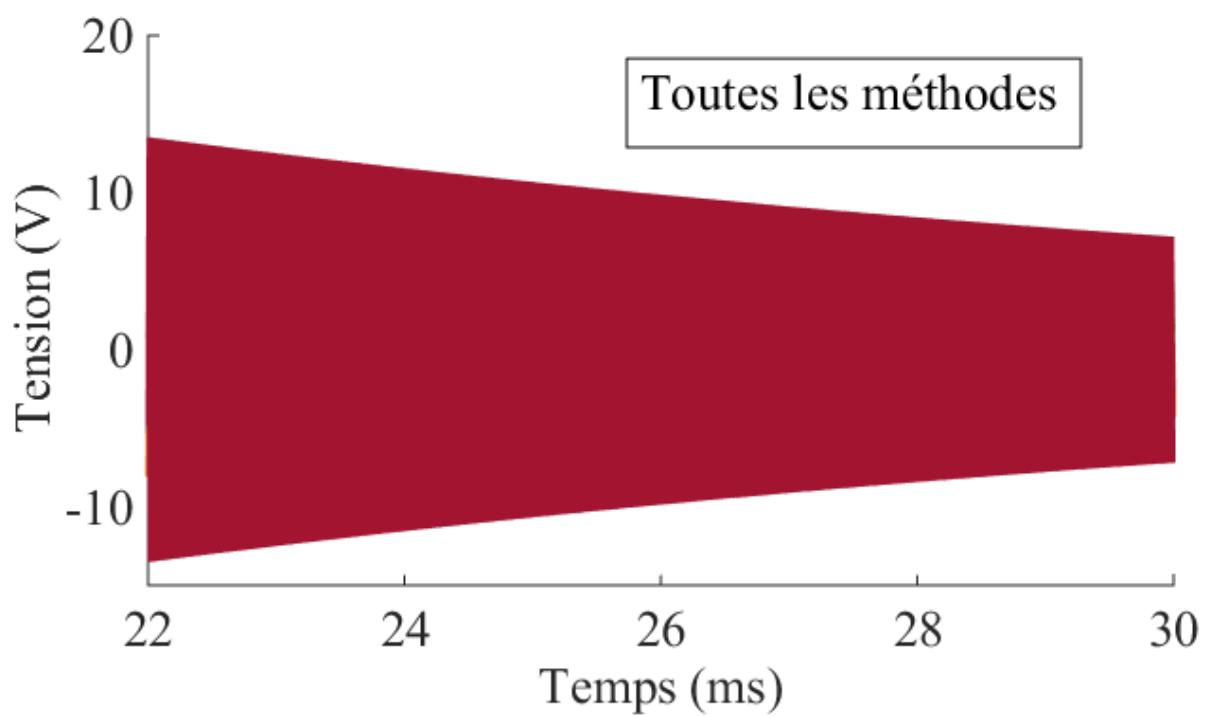
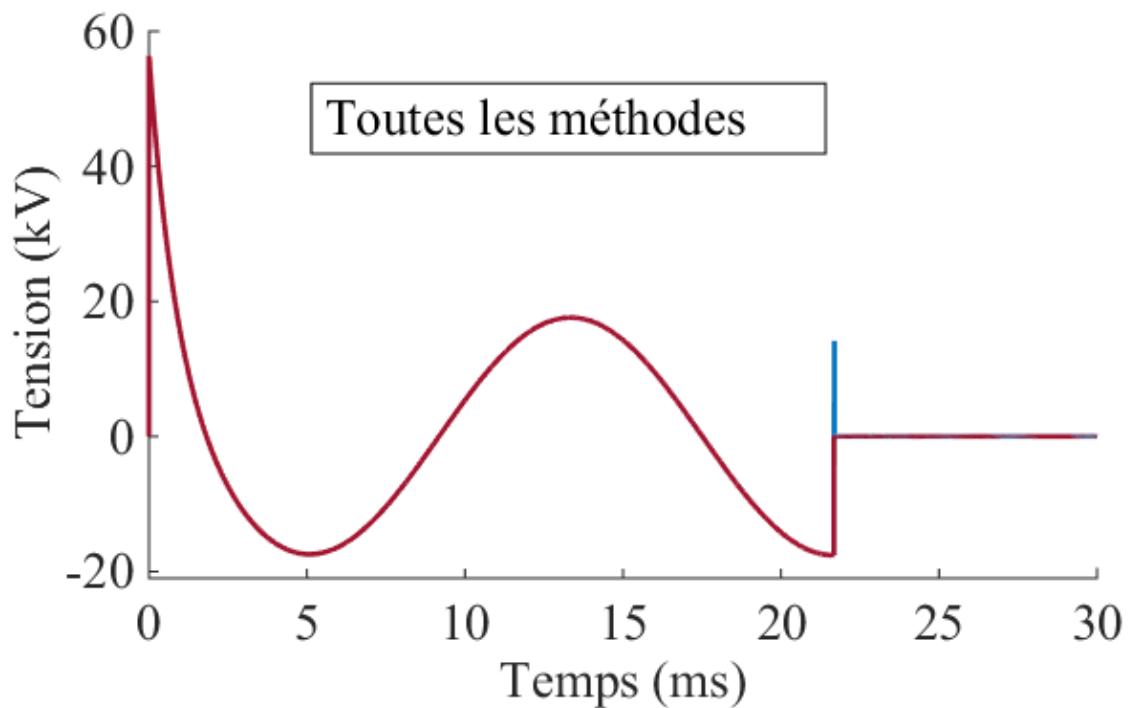
- TR_BE (intégration TR qui utilise la technique CDA). C'est la méthode d'intégration de base. Elle est décrite en 4.2.2.4.
- TR_DICR (intégration TR qui utilise *Chatter Removal Technique* en cas de discontinuités). Cette méthode est décrite en 4.2.2.3.

- TR_BE_I (méthode TR_BE qui utilise l'interpolation linéaire pour ramener la simulation au point de discontinuité). Elle est décrite en 4.2.2.2.5.
- TR_BE_SS (méthode TR_BE qui utilise la technique *Simultaneous Switching*). Cette méthode est décrite en 4.1.3.2.
- TR_BE_I_SS_p : c'est une variante de la méthode publiée dans [58] qui est décrite en 5.1.2.1.
- Les 3 nouvelles méthodes TR_BE_SS_I, TR_BE_I_SS_m et TR_I_SS_m_BE décrites dans la partie 5.2.

Chaque cas de simulation présenté ci-dessous vise un objectif bien précis. Le premier circuit simulé permet de mettre en évidence le comportement des méthodes testées pour la suppression des oscillations numériques. Le deuxième circuit permet d'observer le comportement des méthodes lorsque le circuit simulé est le siège de discontinuités se produisant entre deux points de temps consécutifs d'une part et le siège de situations de commutations simultanées d'autre part. Le troisième et le quatrième circuit sont des exemples réels de circuits d'électronique de puissance dont la simulation se heurte à tous les défis présentés en 4.1. Les performances des méthodes testées (surtout des méthodes proposées) en termes de précision et d'efforts de calculs sont évaluées.

5.3.1 Simulation d'un simple circuit RL

Le circuit de la Figure 4.4 est simulé afin d'étudier la capacité des méthodes testées et de certaines méthodes existantes à supprimer les oscillations numériques résultantes de l'intégration TR. Le pas d'intégration de la simulation est $\Delta t = 10\mu s$. L'interrupteur SW (initialement fermé) est ouvert à $t = 15ms$. Les formes d'onde de la tension aux bornes de l'inductance, (obtenues pour chacune des méthodes) sont présentés à la Figure 5.2.



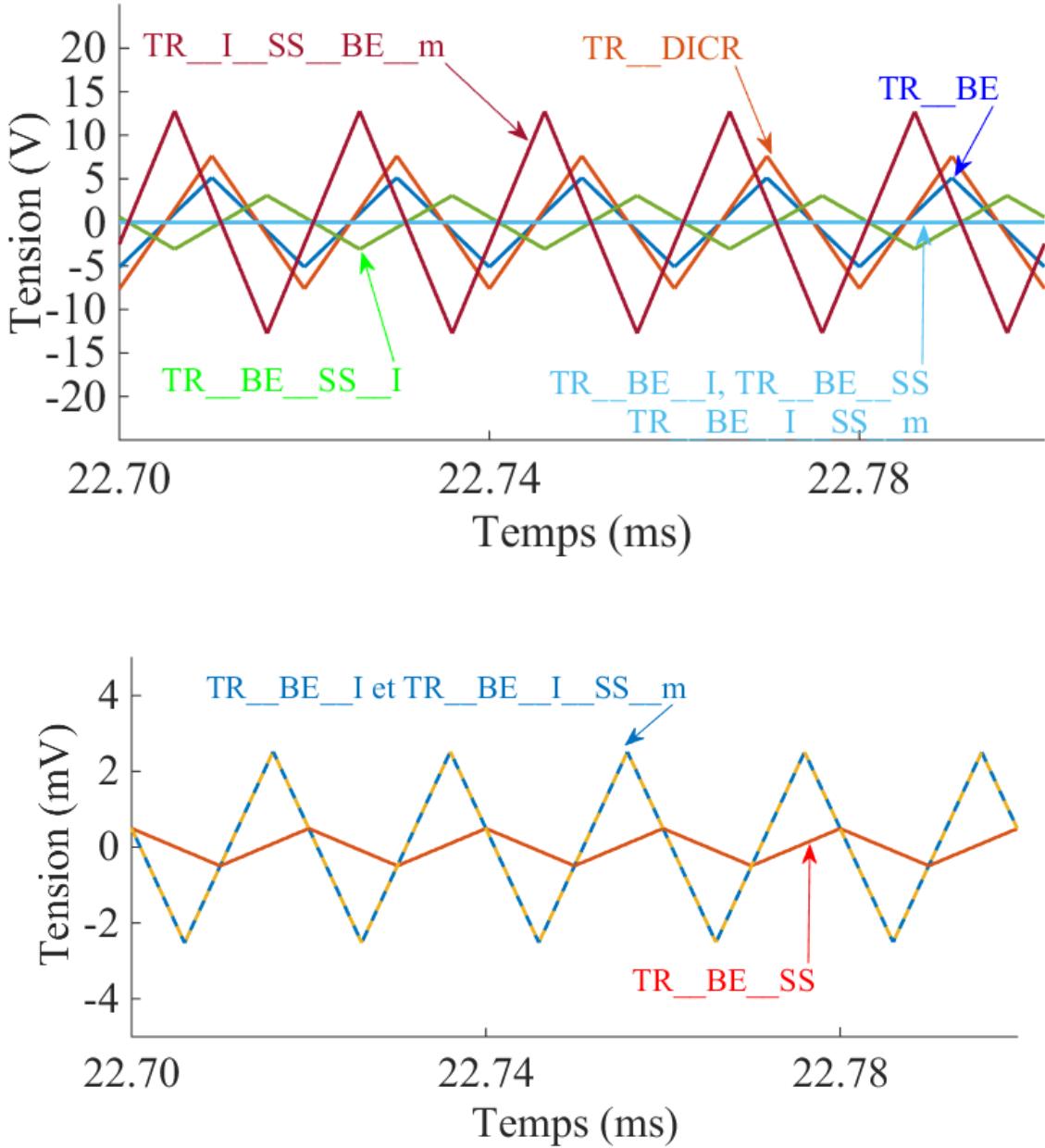


Figure 5.2: Oscillations numériques sur la tension de l'inductance avec les méthodes testées

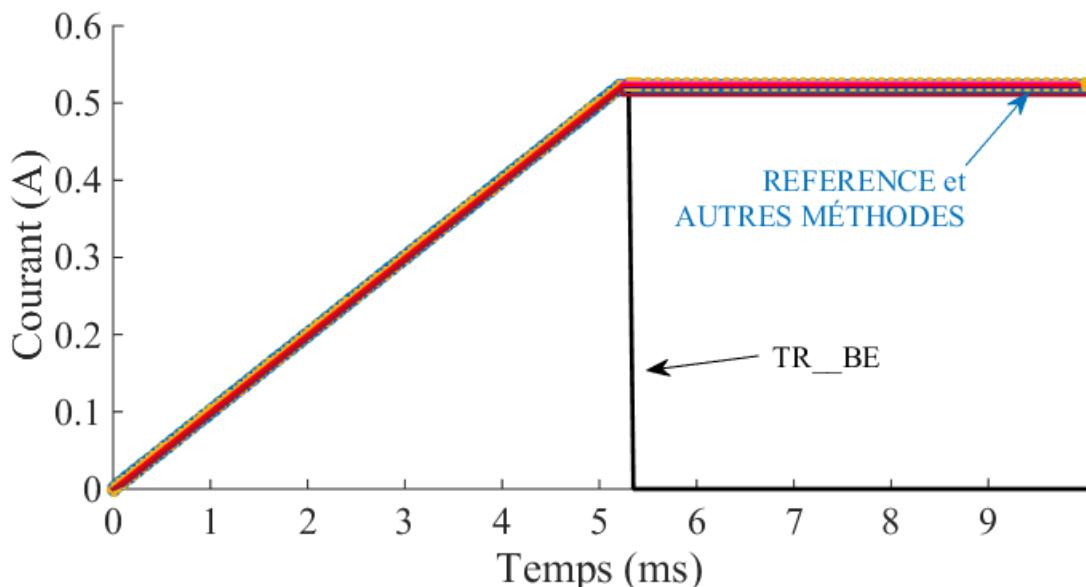
La première sous-figure (tout en haut) semble indiquer l'absence d'oscillations numériques, mais lorsque la figure est zoomée (deuxième sous-figure), on peut observer des oscillations numériques amorties. Un zoom plus accentué (troisième et quatrième sous-figures) permet d'observer les oscillations avec plus de détail. Le premier constat qui se dégage de ces résultats est le fait qu'aucune des techniques testées n'est en mesure d'éliminer complètement les oscillations numériques. Les oscillations observées présentent des amplitudes différentes selon la méthode. On

peut distinguer deux ordres de grandeur des amplitudes. D'une part, on a les méthodes TR_BE, TR_DICR TR_I_SS_m_BE et TR_BE_SS_I dont les amplitudes des oscillations sont de l'ordre du volt (figure en bas à gauche) et d'autre part on a les méthodes TR_BE_I, TR_BE_I_SS_m et TR_BE_SS qui présentent des amplitudes d'oscillations de l'ordre du millivolt (figure en bas à droite). On note un très grand écart entre ces ordres de grandeur. Cela s'explique par l'analyse faite à la partie 4.1.2.3, particulièrement à l'aide de l'équation (4.32) qui montre que pour les méthodes qui utilisent la technique CDA, l'amplitude des oscillations numériques dépend de la valeur du courant dans l'inductance à l'instant de la prise en compte de la discontinuité. Ainsi, on peut comprendre qu'avec la méthode TR_BE, la discontinuité est prise en compte au point de temps suivant. Le courant dans l'inductance au moment de la prise en compte de la discontinuité a donc la valeur la plus élevée possible parmi toutes les méthodes qui utilisent CDA. C'est pourquoi l'amplitude des oscillations avec TR_BE est la plus élevée parmi toutes les méthodes qui utilisent CDA. De même, avec TR_BE_I, (ou TR_BE_SS), l'utilisation de l'interpolation (ou de la technique SS) contribue à réduire significativement la valeur du courant dans l'inductance au moment de l'ouverture de SW. Pour cette raison, les oscillations qui résultent de ces méthodes ont des amplitudes très faibles. La même conclusion peut être obtenue pour TR_BE_I_SS_m. Cependant, avec TR_BE_SS_I, l'application de l'interpolation après le SS résulte en un courant dans l'inductance au moment de la discontinuité dont la valeur est inférieure à celle obtenue avec TR_BE, mais supérieur à celle de TR_BE_I, ce qui justifie l'ordre de grandeur de l'amplitude des oscillations de TR_BE_SS_I.

Lorsqu'on compare les courbes de TR_I_SS_m_BE, TR_DICR et TR_BE (trois techniques différentes de suppression des oscillations numériques), on peut conclure que l'utilisation de deux intégrations BE (technique CDA) est l'approche la plus efficace pour supprimer les oscillations car TR_BE présente des oscillations d'amplitude plus faible que les deux autres. La technique CDA est particulièrement plus efficace que la technique CRT, implémentée dans TR_DICR. D'un autre côté, avec la méthode TR_I_SS_m_BE, l'amplitude des oscillations est la plus élevée de toutes les méthodes testées. Pour rappel, cette technique est une variante de celle introduite dans [85] qui consiste à utiliser l'intégration BE pour réinitialiser les variables du circuit au point de discontinuité. Cette approche semble être la moins efficace des trois techniques, au vu des résultats obtenus. Finalement, de toutes les 3 méthodes proposées, c'est la méthode TR_BE_I_SS_m qui donne les meilleurs résultats.

5.3.2 Simulation d'un circuit back-boost

Le circuit de la Figure 4.6 est simulé dans le but d'étudier la problématique des commutations simultanées. L'interrupteur commandé SW est initialement fermé et le pas d'intégration de la simulation est $\Delta t = 100\mu s$. L'interrupteur SW est ouvert à l'instant $t_{sw} = 5.23 ms$, un instant choisi volontairement entre deux points de temps consécutifs afin de tester le comportement des méthodes qui utilisent l'interpolation (pour la détection précise des discontinuités) et le comportement de celles qui ne l'utilisent pas. Lorsque SW ouvre, la diode D doit entrer en conduction au même instant. Ceci requiert obligatoirement l'utilisation de la technique SS, autrement les formes d'onde obtenues présenteront des erreurs. Une simulation de référence exécutée avec la méthode TR_BE_SS et un pas d'intégration de $1\mu s$ permettra d'évaluer la capacité des méthodes testées à représenter fidèlement la forme d'onde du courant dans l'inductance L. Les différentes méthodes sont tour à tour utilisées pour simuler le circuit et représenter la forme d'onde du courant dans l'inductance L en fonction du temps. Les résultats obtenus sont présentés à la Figure 5.3.



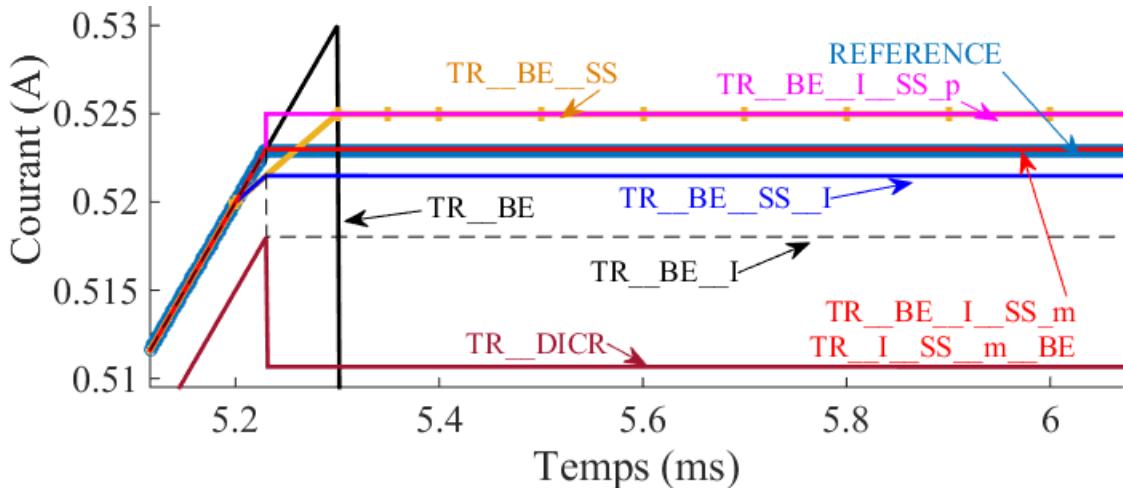


Figure 5.3: Courant dans l'inductance L (allure générale au-dessus et zoom en dessous)

Cette figure montre que seule la méthode TR_BE donne un résultat irréalistique dans lequel le courant dans l'inductance tombe à zéro dès l'ouverture de SW (voir allure générale sur la figure du dessus). Lorsqu'on effectue un zoom de la figure afin de distinguer les formes d'onde des différentes méthodes (figure en dessous), on observe que la méthode TR_DICR est la moins précise pour simuler ce circuit (sa courbe est la plus éloignée de la référence). En effet, avec TR_DICR, le courant tombe à zéro à $t_{sw} + \Delta t$, juste après l'ouverture de SW. A ce moment, la condition de conduction de la diode est satisfaite. L'instant précis t_D de début de conduction de la diode est déterminé par interpolation linéaire, en utilisant l'équation (4.13) et (4.14) dans laquelle $x_t = x_{t_{sw}}$ (la valeur du courant au moment de la première interpolation à t_{sw}) et $x_{t+\Delta t} = 0$. Le résultat donne une valeur très proche mais différente de t_{sw} ($t_D \approx t_{sw}$ mais $t_D > t_{sw}$). A cause de cela, l'interpolation à t_D donne donc un courant légèrement inférieur à sa vraie valeur à t_{sw} . Cette valeur se conserve aux points de temps suivants, tel qu'on peut le constater sur la figure. Un raisonnement analogue peut être fait pour expliquer la forme d'onde de TR_BE_I. Un autre constat important peut être fait à partir du résultat de TR_DICR. En effet, TR_DICR applique l'intégration TR au premier point de la simulation ($t = \Delta t$) avec une valeur non initialisée de la tension aux bornes de l'inductance. La première solution calculée présente donc une erreur due à cette non-initialisation. Cette erreur perdure dans la suite de la simulation. C'est pourquoi la courbe de TR_DICR est légèrement décalée des autres courbes.

Toutes les autres méthodes présentées à la Figure 5.3 appliquent la technique SS, mais produisent des résultats différents. Les courbes de TR_BE_I_SS_m et TR_I_SS_m_BE sont confondues avec la courbe de la référence. Ces deux méthodes proposées sont donc les plus précises pour la simulation de ce circuit. En effet, elles sont les seules qui réinitialisent correctement la solution à t_{sw} après la discontinuité. De son côté, TR_BE_SS applique la technique SS à t_{n+1} , le premier point de temps du time-mesh qui suit directement t_{sw} . Cette méthode assume donc que le changement d'état de SW et de la diode D apparaissent au même moment, mais à t_{n+1} (au lieu de t_{sw}). A cause de cela, la solution obtenue avec TR_BE_SS est légèrement décalée de la référence. De son côté, TR_BE_SS_I utilise la solution finale de TR_BE_SS à t_{n+1} pour linéairement interpoler la valeur du courant à t_{sw} . Le résultat de cette interpolation contiendra lui aussi une erreur, ce qui est la cause de l'écart entre la forme d'onde de TR_BE_SS_I et celle de la référence. Finalement, avec TR_BE_I_SS_p, la solution interpolée à t_{sw}^- est correcte mais la solution à t_{sw}^+ ne l'est pas, ceci à cause de la mauvaise matrice b_i utilisée pour calculer cette solution, comme expliqué en 5.1.2.2. C'est donc à cause de cela que la forme d'onde de TR_BE_I_SS_p présente elle aussi un écart par rapport à la référence. En résumé, seules les méthodes proposées TR_BE_I_SS_m et TR_I_SS_m_BE sont capables de représenter fidèlement les formes d'onde du courant pour ce circuit.

5.3.3 Simulation d'un circuit STATCOM simple

Le circuit de la Figure 5.4 est un circuit STATCOM simple, constitué de 6 IGBT commandés par un générateur PWM avec une fréquence de 1980 Hz. Chaque IGBT est modélisé par deux diodes et un interrupteur commandé comme indiqué à la Figure 3.7. De même, chaque diode est modélisée comme indiqué à la Figure 3.8.

Ce circuit est un convertisseur d'électronique de puissance qui sera simulé dans le but d'évaluer et de comparer la précision des méthodes testées, ceci pour différents pas d'intégration numérique. Des simulations de durée 100 ms chacune sont exécutées avec les pas d'intégration numérique suivant : $1\mu s, 2\mu s, 5\mu s$ et $10\mu s$. Une simulation est effectuée pour chacune des méthodes testées et avec chaque pas d'intégration proposé.

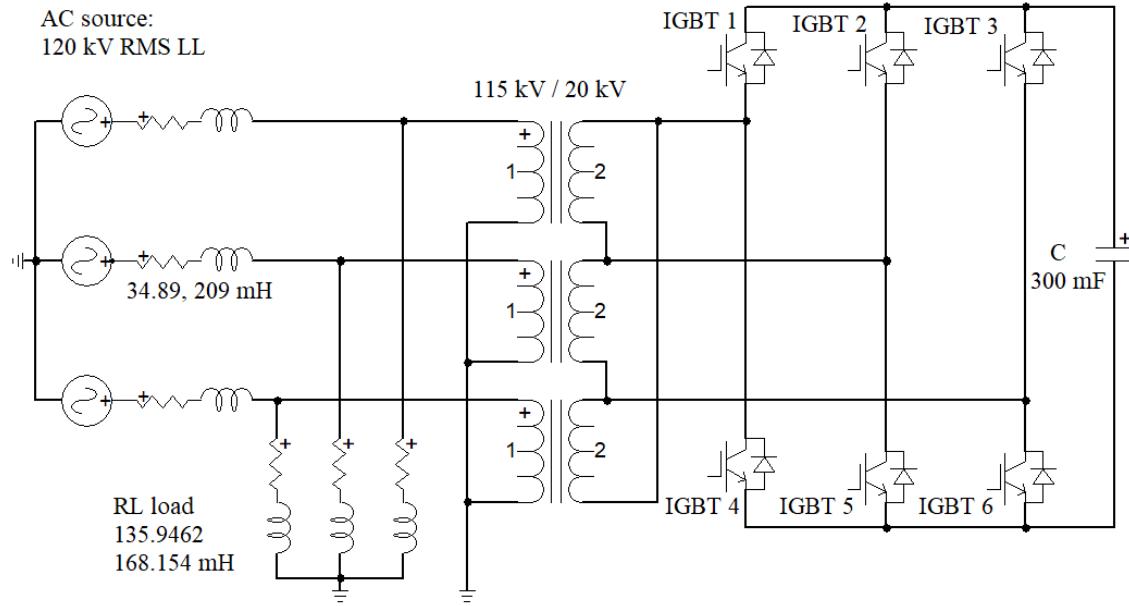


Figure 5.4: Circuit STATCOM simple

Une simulation utilisant la méthode TR_BE_SS avec un pas d'intégration $\Delta t = 500\text{ ns}$ servira de référence pour l'évaluation des erreurs des différentes méthodes. On s'intéresse à la tension aux bornes du condensateur C. La précision de chacune des méthodes pour chaque valeur de pas d'intégration de la simulation est évaluée en calculant l'erreur relative moyenne (RRE) définie par (4.36). Les résultats sont présentés à la Figure 5.5.

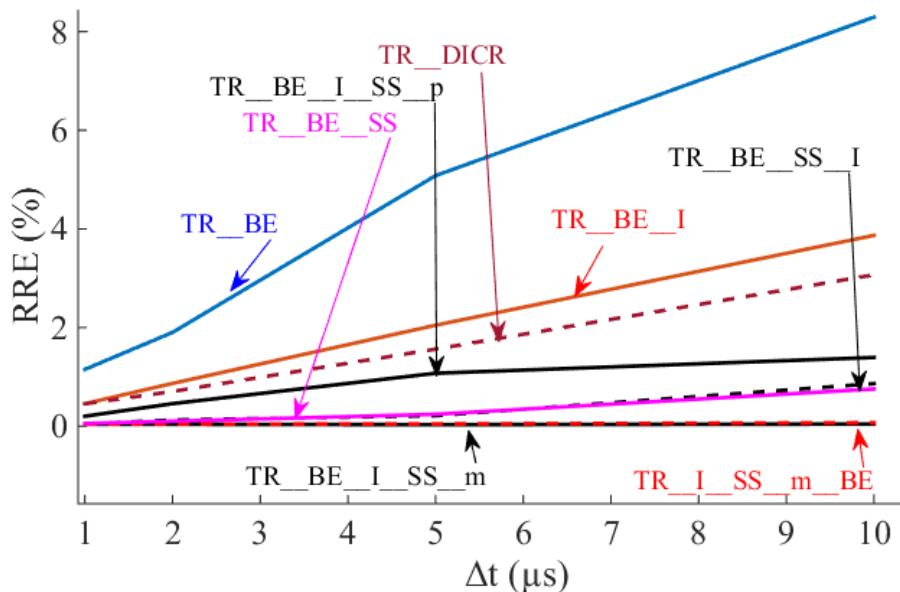


Figure 5.5: Erreur relative moyenne (RRE) sur la tension de C selon le pas d'intégration

De la Figure 5.5, il ressort que les comparaisons des erreurs des différentes méthodes conduisent aux mêmes conclusions, quel que soit la valeur du pas d'intégration. En observant les erreurs des méthodes construites à partir de TR_BE, on peut constater que chaque fois qu'une technique est rajoutée, la précision s'améliore. C'est ainsi que TR_BE est moins précise que TR_BE_I qui elle-même est moins précise que TR_BE_SS. En particulier, le fait que TR_BE_SS donne une erreur plus faible que TR_BE_I nous renseigne sur le fait que la technique SS a un impact plus important sur la réduction de l'erreur que la simple technique d'interpolation. On peut donc s'attendre à ce qu'un logiciel qui utilise la technique SS soit plus précis qu'un autre logiciel qui utilise uniquement l'interpolation.

La méthode existante TR_BE_I_SS_p qui utilise à la fois l'interpolation et la technique SS est plus précise que toutes les méthodes existantes qui utilisent uniquement l'interpolation (TR_BE_I et TR_DICR). Elle est toutefois moins précise que chacune des trois méthodes proposées dans cette thèse (TR_BE_SS_I, TR_BE_I_SS_m et TR_I_SS_m_BE). La méthode TR_BE_SS_I a une précision similaire à celle de TR_BE_SS. Finalement, on constate que pour la plage des pas d'intégrations testés, les méthodes proposées TR_BE_I_SS_m et TR_I_SS_m_BE ont non seulement la meilleure précision, mais aussi sont presque insensibles au pas d'intégration (l'erreur croît très peu lorsque Δt augmente). Ces méthodes peuvent donc garantir la précision des résultats lors des simulations avec des pas d'intégration grands. C'est exactement ce qui est recherché chez une méthode de simulation. Comme dernière observation importante, on peut constater que pour ce circuit, TR_DICR est plus précis que TR_BE_I. Ceci est dû au fait que l'utilisation de l'intégration BE dans la méthode TR_BE_I dégrade la précision de cette méthode, contrairement à TR_DICR qui n'utilise que l'intégration TR. Il faut aussi mentionner qu'aucune de ces 2 méthodes n'utilise la technique SS. Chaque fois que le changement d'état d'un interrupteur entraîne celui de plusieurs autres, chacune de ces méthodes estime par interpolation l'instant d'apparition des changements d'état des autres. De ce fait, les différences dans les valeurs obtenues de l'estimation de ces instants par les deux méthodes sont également sources d'écart de précision entre les résultats obtenus avec TR_BE_I et ceux obtenus avec TR_DICR.

5.3.4 Simulation d'un convertisseur DC-AC-DC

La simulation du circuit STATCOM a permis de valider les précisions des différentes méthodes, ceci pour différentes valeurs du pas d'intégration de la simulation. En plus de la précision, l'effort

de calcul de chaque méthode est également un critère qu'il est important d'évaluer pour conclure sur la performance d'une méthode. Il se mesure par le nombre d'opérations que l'ordinateur doit exécuter pour simuler un circuit en utilisant une méthode donnée.

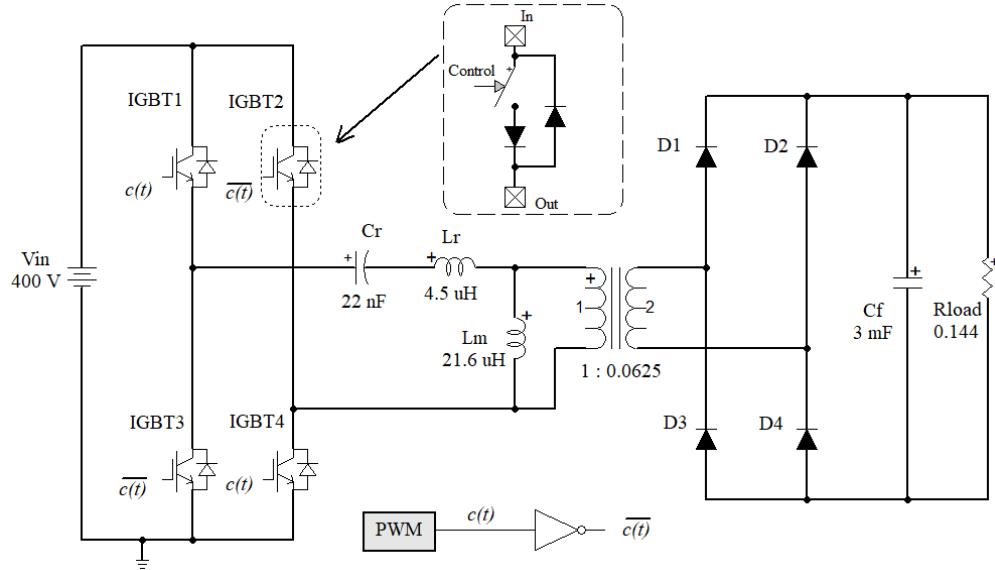


Figure 5.6: Convertisseur DC-AC-DC

Le circuit de la Figure 5.6 est un convertisseur DC-AC-DC constitué de 4 IGBT et de 4 diodes. Chaque IGBT est modélisé comme indiqué à la Figure 3.7 et chaque diode est modélisée comme à la Figure 3.8. Les interrupteurs commandés de chaque IGBT reçoivent des signaux d'un générateur « step » périodique, de largeur 50% et de fréquence 300 kHz. Ces signaux ont l'allure suivante (

$$T = \frac{1}{300 \text{ kHz}} :$$

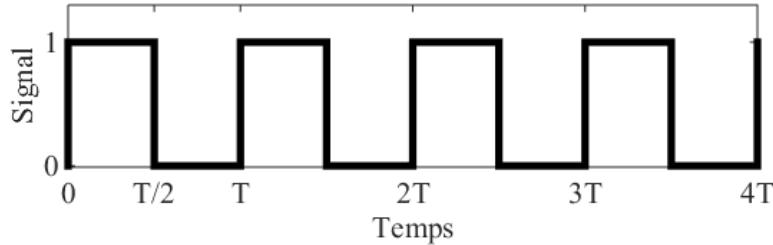


Figure 5.7: Signaux de commande provenant du générateur PWM

Le circuit de la Figure 5.6 est simulé avec les méthodes discutées dans cette thèse. L'objectif est d'évaluer l'effort de calcul de chaque méthode. Cet effort est quantifié par le nombre de fois que

le système d'équations du circuit simulé est solutionné. En effet, solutionner le système (2.44) représente l'opération la plus complexe, contraignante et consommatrice de temps et de ressources d'un ordinateur qui effectue une simulation. Par comparaison, réaliser une interpolation requiert un effort négligeable. L'effort de calcul d'une méthode est directement corrélé avec le temps de calcul.

Lorsque le circuit de la Figure 5.6 est simulé avec un pas d'intégration suffisamment petit (par exemple $\Delta t = 10\text{ns}$, toutes les méthodes ont la même précision. On peut le constater par exemple, sur la Figure 5.6 qui présente l'allure de la tension aux bornes de la charge R_{load} pour les méthodes testées.

Lorsque les simulations sont répétées avec des pas d'intégration plus grands, les formes d'onde obtenues montrent des différences de précisions entre les méthodes. Toutefois, la classification des méthodes selon leur niveau de précision demeure en accord avec les résultats de la Figure 5.5.

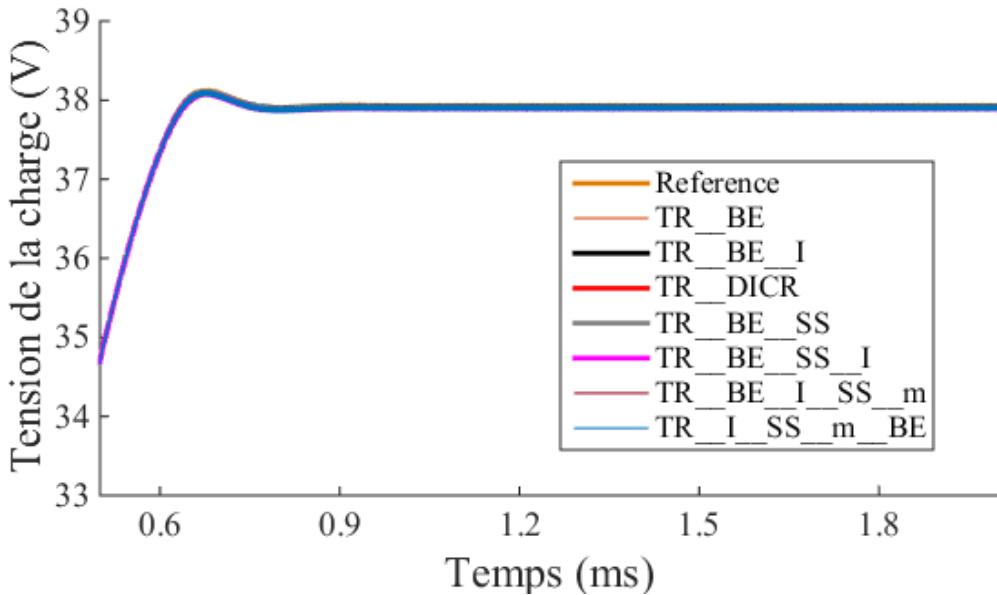


Figure 5.8: Tension aux bornes de la charge R_{load} pour toutes les méthodes avec $\Delta t = 10\text{ns}$

Afin d'évaluer l'effort de calcul des méthodes, ces dernières seront simulées pendant 1ms avec différents pas d'intégration : $30\text{ns}, 40\text{ns}, 50\text{ns}$, et 80ns . A la suite de chaque simulation, le nombre de fois que le système (2.44) aura été solutionné est enregistré. L'ensemble des résultats permettent pour chaque méthode de représenter l'évolution du nombre de fois que le système est solutionné en fonction du pas d'intégration. Cela permet d'obtenir la Figure 5.9.

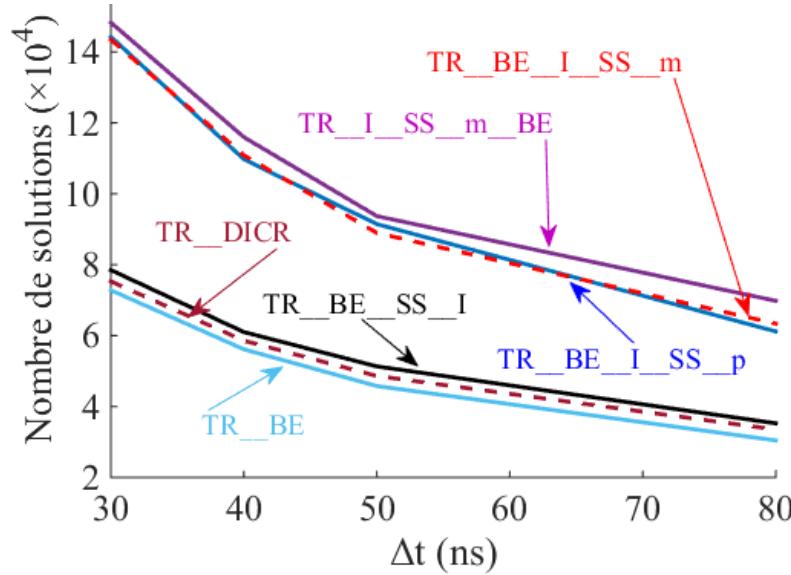


Figure 5.9: Nombre de fois que le système est solutionné pour chaque méthode

De base, plus le nombre de fois que le système est solutionné est élevé, plus l'effort de calcul (ou charge de calcul) de la méthode correspondante est important. La Figure 5.9 montre que plus le pas d'intégration est élevé, plus l'effort de calcul est faible. Ceci est tout à fait normal puisqu'avec un pas d'intégration plus grand, l'algorithme a besoin de solutionner le système d'équations moins de fois pour compléter la durée totale de la simulation. L'observation plus profonde de la figure révèle que les méthodes les moins sophistiquées (TR_BE, TR_DICR) ont les efforts de calcul les plus faibles.

Les méthodes TR_BE_I_SS_m, TR_I_SS_m_BE et TR_BE_I_SS_p ont à peu près la même charge de calcul. Une observation pareille est faite pour les méthodes TR_DICR, TR_BE et TR_BE_SS_I. De plus, il existe un écart non négligeable entre ces deux groupes de méthodes, preuve que la complexité d'une méthode augmente significativement lorsqu'on intègre simultanément l'interpolation et la technique SS. Il est toutefois observé que la charge de calcul de la méthode TR_BE_SS_I n'obéit pas à cette règle. Elle est plus faible. Ceci est dû aux simplifications introduites dans le mode de fonctionnement de la méthode. En effet, lorsqu'on a par exemple plusieurs commutations qui se produisent entre deux points de temps consécutifs, la méthode TR_BE_SS_I va les prendre en compte tous au même instant et interpoler une seule fois. Cependant, avec les autres méthodes qui utilisent l'interpolation et SS, chaque discontinuité sera traitée séquentiellement, l'une après l'autre, ce qui augmente l'effort de calcul.

La méthode TR_BE est la plus simple et est donc celle qui présente l'effort de calcul le plus faible. D'un autre côté, la méthode proposée TR_I_SS_m_BE est la plus complexe et est celle qui présente l'effort de calcul le plus important. Ceci est dû au processus de réinitialisation qui exécute deux intégrations BE (et qui revient en arrière au point de discontinuité par extrapolation) chaque fois qu'il y'a une discontinuité. Ce mécanisme augmente l'effort de calcul.

La conclusion la plus importante à retenir de cette figure est le fait que, de toutes les méthodes qui combinent l'interpolation et la technique SS, la méthode TR_BE_SS_I se démarque véritablement en termes de temps de calcul. Elle est beaucoup plus performante que les autres sur ce critère. Ce comportement excellent est toutefois affaibli par sa précision relativement plus faible, par comparaison avec les deux autres méthodes proposées dans cette thèse.

CHAPITRE 6 METHODES DE SIMULATION DES CIRCUITS NON-LINÉAIRES

Ce chapitre s'intéresse à la simulation des circuits contenant des composants non-linéaires. Simuler de tels circuits nécessite l'utilisation de méthodes particulières afin de calculer convenablement la solution à chaque point de temps, et déterminer simultanément les points d'opération des composants non-linéaires. Le but de ce chapitre est de préciser l'approche générale de simulation des circuits non-linéaires, de décrire les méthodes existantes et de présenter une nouvelle approche de simulation. Cette approche sera testée et comparée aux méthodes existantes.

6.1 Généralités

6.1.1 Les composants linéaires et les composants non-linéaires

6.1.1.1 Description

Chaque composant d'un circuit électrique est décrit par une relation liant au moins deux des variables qui le caractérisent. Ainsi, une résistance est décrite par la relation entre la tension à ses bornes et le courant qui la traverse. Une inductance est décrite par la relation entre le courant qui la traverse et le flux qu'elle génère. Un condensateur est décrit par la relation entre la tension à ses bornes et la charge électrique qu'il emmagasine. Dans chacun des cas suscités, la relation de dépendance peut être représentée graphiquement par une courbe. C'est la caractéristique du composant. Lorsque cette courbe est une droite, la relation de dépendance est linéaire et le composant concerné est dit linéaire. Dans le cas contraire, le composant est dit non-linéaire.

Un exemple très commun de composant non-linéaire est la diode. Cet interrupteur électronique de base est généralement représenté par un modèle idéalisé, comme à la Figure 3.8. Toutefois, son comportement réel est typique de celui d'une fonction non-linéaire. Le modèle idéal a l'avantage de réduire significativement les temps de calcul par rapport au modèle non-linéaire. Mais dans certains cas, les résultats obtenus ne représentent pas fidèlement la réalité ce qui justifie la nécessité d'utiliser des modèles non-linéaires.

6.1.1.2 Représentation

La représentation d'un composant non-linéaire peut être détaillée. Dans ce cas, la relation de dépendance entre deux variables de ce composant (généralement la tension v_ϕ et le courant i_ϕ) est décrite par une équation : c'est le modèle dit « détaillé ». La représentation graphique de cette dépendance constitue la caractéristique du composant. Il est possible d'approximer un modèle détaillé par un ensemble de segments linéaires obtenus en sélectionnant un ensemble de M points (v_m, i_m) de la caractéristique (avec m variant de 1 à M), et en les reliant par des segments de droite comme illustré à la Figure 6.1. Le résultat représente le modèle dit « linéaire par segments » [12],[97]. Les M points sélectionnés sur la caractéristique initiale qui servent à construire la caractéristique linéaire par segments sont appelés *points de cassure* (break-points). Ils sont représentés en rouge à la Figure 6.1.

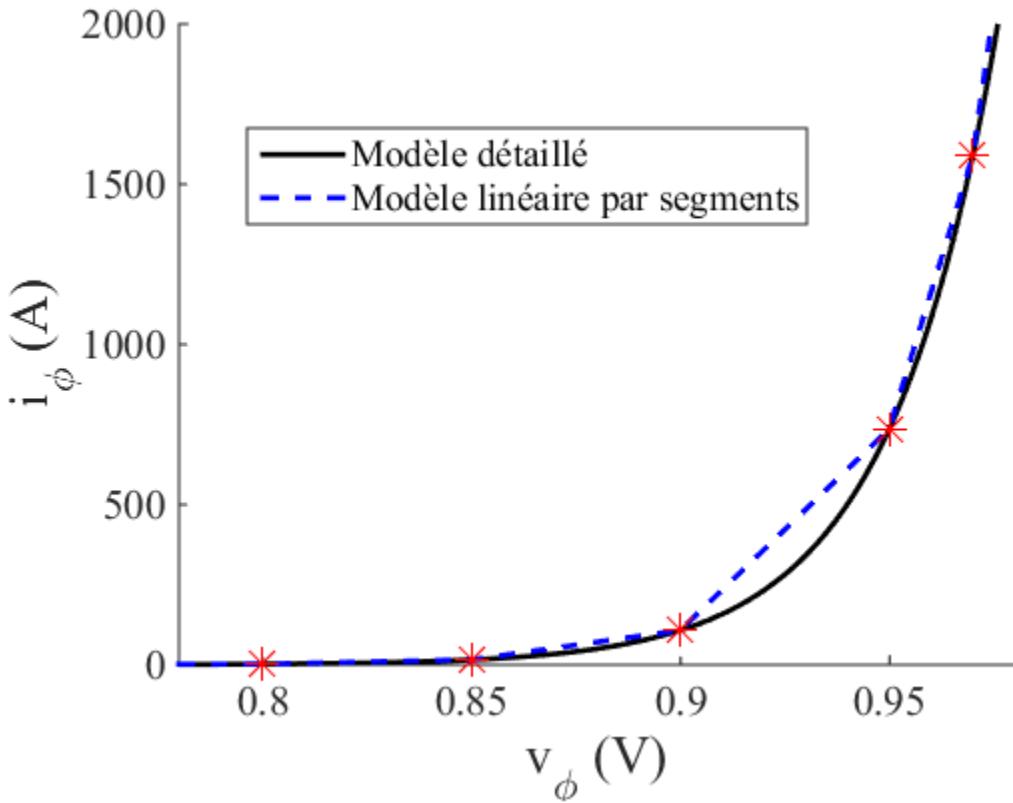


Figure 6.1: Représentation détaillée et linéaire par segments d'un composant non-linéaire

De ce fait, entre deux points de cassure consécutifs (v_j, i_j) et (v_{j+1}, i_{j+1}) , (j est un entier compris entre 1 et $M-1$), la non-linéarité est représentée par un segment de droite (identifié segment

numéro j). La relation de dépendance entre la tension et le courant (désignés respectivement par v_ϕ^j et i_ϕ^j) sur le segment j prend la forme linéarisée :

$$i_\phi^j = y_j v_\phi^j + i_j \quad (6.1)$$

où y_j (pente) et i_j (ordonnée à l'origine) dépendent du segment j . Ainsi, avec le modèle linéaire par segments, une équation de droite est établie pour représenter le composant sur chacun des $M - 1$ segments construits à partir des M points de cassure. Un exemple de calcul des paramètres y_j et i_j est présenté en annexe B.

Le modèle linéaire par segments est une simplification du modèle détaillé et permet d'obtenir une précision très bonne tout en simplifiant les calculs. C'est en général ce type de modèle qui est adopté pour représenter les composants non-linéaires.

6.1.2 Approche de résolution d'un circuit contenant une non-linéarité

Dans le cas le plus général, un composant non-linéaire connecté au reste d'un circuit peut être représenté à tout instant t par le circuit simplifié de la Figure 6.2.

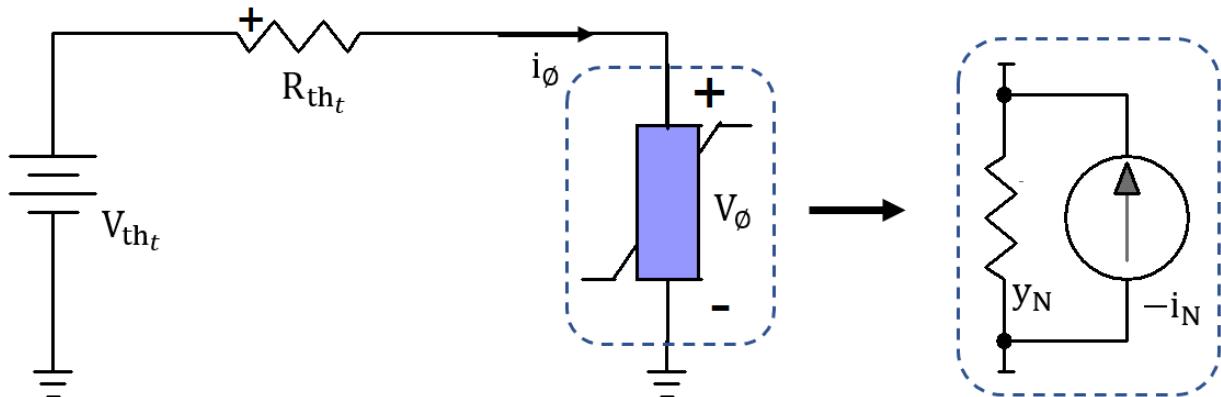


Figure 6.2: Circuit générique comportant un composant non-linéaire

Dans cette figure, V_{th_t} et R_{th_t} sont respectivement la tension et la résistance de l'équivalent Thévenin aux bornes du composant non-linéaire à l'instant t . On peut généraliser cette représentation dans le cas d'un circuit plus complexe, en regroupant tous les composants non-linéaires. L'équivalent Thévenin prendra dans ce cas une forme matricielle. Sur la figure, les

courant et tension de la branche non-linéaire portent l'indice ϕ . Dans le cas général, la non-linéarité est décrite par une équation de la forme :

$$f(v_\phi, i_\phi) = 0 \quad (6.2)$$

Dans cette équation, f est une fonction non-linéaire. A partir du circuit de la Figure 6.2, on peut écrire que :

$$v_{th_t} - R_{th_t} i_\phi = v_\phi \quad (6.3)$$

Si la branche non-linéaire est un composant extrêmement non-linéaire comme une diode, alors l'équation (6.2) correspond à l'équation du modèle détaillé de la diode :

$$i_\phi = I_s (e^{v_\phi/V_T} - 1) \quad (6.4)$$

$I_s = 10^{-13}$ et $V_T = 0.026$. L'équation du circuit prend la forme :

$$f(v_\phi, i_\phi) = v_{th_t} - R_{th_t} i_\phi - v_\phi = 0 \quad (6.5)$$

La résolution de cette équation permet de déterminer le point d'opération du composant non-linéaire à tout instant t , soit (v_{ϕ_t}, i_{ϕ_t}) . Pour ce faire, la relation (6.4) est intégrée dans l'équation

(6.5) pour ne faire apparaître qu'une seule variable inconnue (soit v_ϕ , soit i_ϕ), puis des solutions itératives de la variable en question sont calculées. Il s'agit en fait de partir d'une prédiction initiale, puis de corriger cette prédiction au fur et à mesure jusqu'à obtenir une valeur qui vérifie l'équation (6.5). Il existe plusieurs manières de calculer les différentes solutions itératives. Chacune d'elles correspond à une méthode de résolution.

6.2 Méthodes existantes pour la simulation des circuits non-linéaires

Pour la simulation des circuits comportant des non-linéarités, il existe des méthodes non itératives et des méthodes itératives. Certains logiciels n'utilisent pas des itérations. Les solutions numériques calculées à chaque point de temps présentent un écart (over shoot) par rapport à leurs vraies valeurs. Cet écart est corrigé au point de temps suivant, mais la précision est globalement mauvaise avec une telle approche. Les méthodes itératives quant à elles utilisent normalement la

méthode Newton (appelée Newton-Raphson (NR)) et permettent de déterminer le point d'opération de chaque composant non-linéaire à chaque point de temps de la simulation.

Dans cette section, les méthodes existantes les plus connues seront présentées et leur fonctionnement pour le calcul de la solution du circuit de la Figure 6.2 sera décrit en guise d'illustration. À la fin, l'intégration de ces méthodes à la formulation MANA dans le but de généraliser leur utilisation à la simulation de circuits plus complexes sera établie.

6.2.1 La méthode Newton classique

La méthode Newton (ou NR) [98] détermine le point d'opération de chaque composant non-linéaire en calculant des solutions itératives de la tension (ou du courant).

Par exemple, pour le cas simplifié décrit par l'équation (6.5), la solution itérative de la tension V_{ϕ_t} à l'instant t est donnée par :

$$V_{\phi_t}^{k+1} = V_{\phi_t}^k - f(V_{\phi_t}^k) \left[\frac{df}{dv_\phi} \Big|_{v_{\phi_t}^k} \right]^{-1} \quad (6.6)$$

Ainsi, après chaque itération k , la valeur connue $V_{\phi_t}^k$ est utilisée pour calculer la valeur $V_{\phi_t}^{k+1}$.

Cette dernière est à son tour utilisée pour le calcul de la valeur à l'itération $k+2$ et ainsi de suite. Le processus se répète jusqu'à ce que l'écart absolu entre deux solutions itératives consécutives devienne inférieur à une limite ε prédéfinie. En d'autres termes, le processus itératif va s'arrêter à l'itération $K+1$ tel que $|V_{\phi_t}^{K+1} - V_{\phi_t}^K| < \varepsilon$. La dernière solution obtenue $V_{\phi_t}^{K+1}$ représente la solution finale V_{ϕ_t} de la tension du composant non-linéaire à l'instant t . On dit que le calcul de la solution a convergé.

Le processus de la méthode NR peut être illustré graphiquement. Une façon de procéder consiste à représenter graphiquement les variations de la fonction f en fonction de v_ϕ tel que traduit par l'équation (6.5), puis rechercher le point où elle s'annule. Pour ce faire, à partir de la solution $V_{\phi_t}^k$

à l'itération k de l'instant de simulation t , la tangente en la courbe au point d'abscisse $V_{\phi_t}^k$ est tracée. L'abscisse du point où cette tangente s'annule représente $V_{\phi_t}^{k+1}$. De ce fait, la tangente en la courbe au point d'abscisse $V_{\phi_t}^{k+1}$ est tracée et le processus itératif se poursuit ainsi de suite jusqu'à la convergence. Un exemple illustratif est présenté à la Figure 6.3.

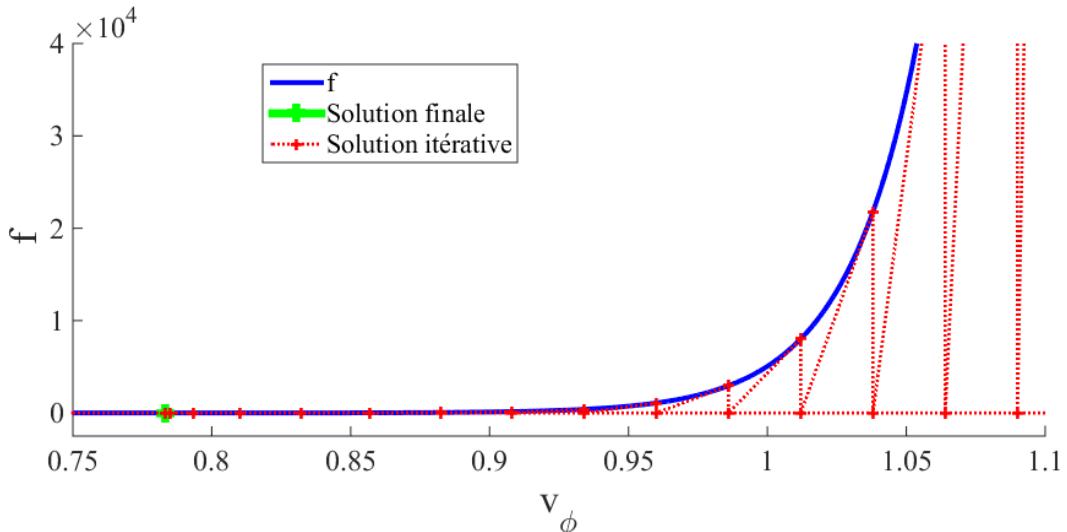


Figure 6.3: Illustration du processus itératif de la méthode Newton

Il est important de préciser que lorsque le composant est fortement non-linéaire, le nombre d'itérations requises pour converger peut devenir très élevé, surtout lorsque la prédition initiale n'est pas suffisamment proche de la solution finale. On peut le constater sur la Figure 6.3 en remarquant qu'avec la prédition initiale à 0 V, la tangente à la fonction non-linéaire est une droite pratiquement horizontale, ce qui fait que la première solution calculée se retrouve très loin à droite de la figure (hors du cadre). Plusieurs itérations sont par la suite nécessaires pour ramener la solution progressivement vers la valeur finale (en vert). Cet inconvénient majeur motive le développement de méthodes alternatives qui seront présentées plus bas.

Une autre façon d'illustrer graphiquement le fonctionnement de la méthode NR consiste à représenter la caractéristique non-linéaire de l'équation (6.4), puis représenter la caractéristique de l'équivalent Thévenin du reste du circuit à l'instant t pour enfin déterminer le point d'intersection P. D'après la Figure 6.2, la caractéristique de l'équivalent Thévenin s'exprime comme suit :

$$i_{\phi} = -\frac{1}{R_{th_t}} v_{\phi} + \frac{v_{th_t}}{R_{th_t}} \quad (6.7)$$

La Figure 6.4 présente les deux caractéristiques. Illustrons graphiquement les étapes d'une itération avec la méthode NR. On part du point M_k qui représente la solution $(v_{\phi_t}^k, i_{\phi_t}^k)$ à l'itération k (si $k=0$, cette solution correspond à la prédiction initiale à l'instant t). La tangente en M_k à la caractéristique non-linéaire est construite et le point d'intersection S de cette tangente avec la caractéristique Thévenin est déterminé. Le point S est ensuite déplacé verticalement pour retourner sur la caractéristique, ce qui donne le point $M_{k+1}(v_{\phi_t}^{k+1}, i_{\phi_t}^{k+1})$ qui représente la solution itérative suivante. Le processus est répété pour obtenir le point M_{k+2} et ainsi de suite jusqu'à aboutir à la solution finale P.

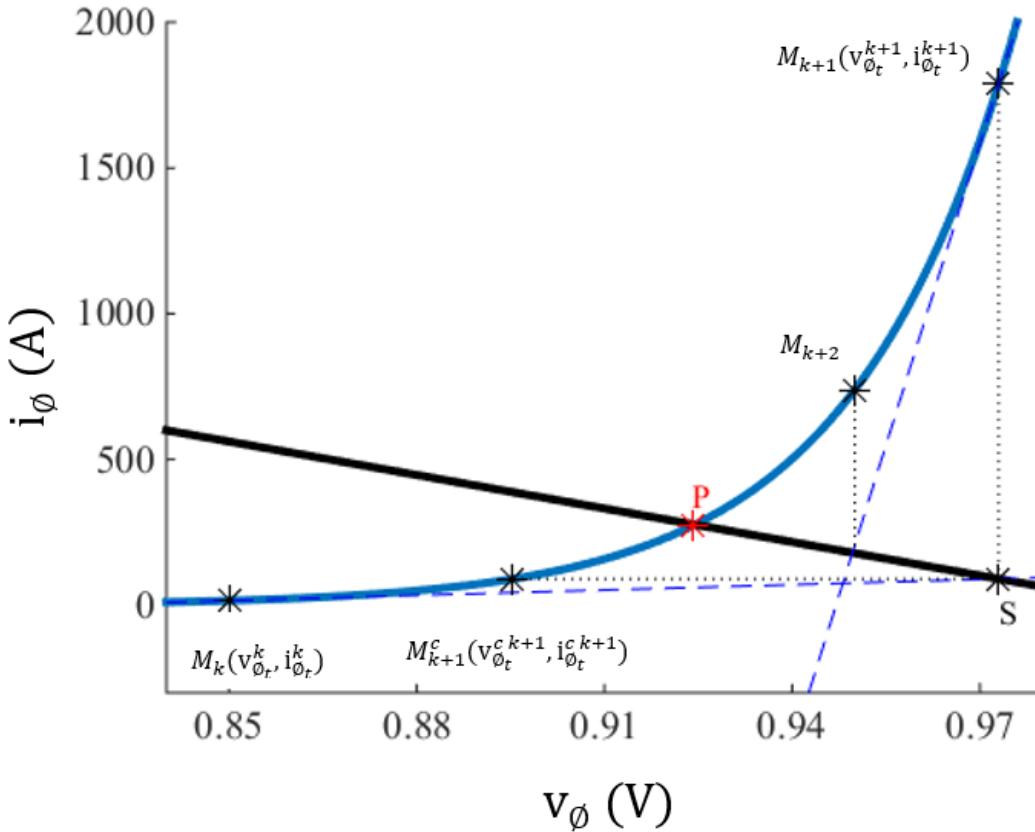


Figure 6.4: Caractéristiques de la diode (en bleu) et de l'équivalent Thévenin du reste du circuit (en noir), à un point de temps t

La méthode NR est une méthode efficace et applicable pour tout type de composant non-linéaire. Elle présente néanmoins deux inconvénients majeurs. D'une part, le nombre d'itérations nécessaires pour aboutir à la solution finale peut parfois être extrêmement élevé pour des fonctions très non-linéaires, comme l'équation d'une diode. D'autre part, la convergence de la méthode n'est pas garantie, sauf si la prédiction initiale est suffisamment proche de la solution finale telle que démontré par le théorème de Newton-Kantorovich [99]. Le processus itératif peut entrer dans une boucle infinie et ne jamais être capable d'atteindre la solution finale. Ce n'est pas de la divergence, mais du brouillage. Ces deux inconvénients majeurs ont motivé le développement de méthodes alternatives ou de variantes.

6.2.2 La méthode de Katzenelson

La méthode de Katzenelson [12],[100] est une variante de la méthode NR qui est supposé garantir la convergence lors du processus itératif. Elle est applicable uniquement lorsque les composants non-linéaires sont décrits par des modèles linéaires par segments. Pour rappel, cette représentation consiste à approximer la courbe d'une caractéristique non-linéaire par un ensemble fini de segments construits à partir de paires de points sélectionnés sur la caractéristique initiale. Par exemple, pour une diode, la caractéristique détaillée est obtenue en présentant graphiquement la dépendance courant-tension décrite par l'équation (6.4). La caractéristique linéaire par segments quant à elle est obtenue en sélectionnant un échantillon de 15 points de cette caractéristique et en les reliant par des segments de droite. Le résultat est la courbe en bleue à la Figure 6.1. Les points sélectionnés, qui sont appelés points de cassure de la caractéristique (break-points) correspondent généralement à ceux présentés dans les Tableau 6.1 et Tableau 6.2.

Les différences dans le mode opératoire entre la méthode de Katzenelson et la méthode NR peuvent être décrites à partir des points représentés sur la Figure 6.4, en remplaçant la caractéristique non-linéaire par une caractéristique linéaire par segments, construite à partir des points des Tableau 6.1 et Tableau 6.2.

Tableau 6.1: Points de cassure de la caractéristique linéaire par segments d'une diode (1/2)

Point	1	2	3	4	5	6	7	8
$v_\phi(V)$	-10^5	-50	0	0.7	0.8	0.85	0.8999	0.9499
$i_\phi(A)$	-10^{-7}	-10^{-8}	0	0.049	2.306	15.779	107.547	735.837

Tableau 6.2: Points de cassure de la caractéristique linéaire par segments d'une diode (2/2)

Point	9	10	11	12	13	14	15
$v_\phi(V)$	0.9699	0.9899	1	1.1	1.2	1.4	1.7
$i_\phi(A)$	1588	3427	5054	2.37×10^5	1.108×10^7	2.427×10^{10}	2.490×10^{15}

À tout instant t et à chaque itération k , la valeur connue $v_{\phi_t}^k$ de la solution est utilisée pour calculer la solution $v_{\phi_t}^{k+1}$ de la prochaine itération. Le fonctionnement de la méthode Katzenelson commence par le calcul d'une valeur potentielle de $v_{\phi_t}^{k+1}$ via (6.6) comme c'est fait avec la méthode NR. Il est important de préciser que puisque la méthode Katzenelson utilise le modèle linéaire par segments, alors dans la relation (6.5) (qui est utilisé dans (6.6)), le courant i_ϕ est obtenu non pas par (6.4), mais plutôt par (6.1) et en utilisant les paramètres y_j et i_j du segment j auquel appartient $v_{\phi_t}^k$. Une fois que la solution $v_{\phi_t}^{k+1}$ a été obtenue par (6.6), la méthode Katzenelson procède comme suit :

1. Le précédent point d'opération $M_k(v_{\phi_t}^k, i_{\phi_t}^k)$ est déplacé dans la direction déterminée par $v_{\phi_t}^{k+1}$ tout en restant sur le segment linéaire auquel il appartient. En d'autres termes, si $v_{\phi_t}^{k+1} > v_{\phi_t}^k$, le point M_k est déplacé vers la droite tandis que si $v_{\phi_t}^{k+1} < v_{\phi_t}^k$, le point est déplacé vers la gauche.

2. Si lors du déplacement, le point de tension $V_{\phi_t}^{k+1}$ est atteint sans croiser de point de cassure, alors la solution finale de la tension est $V_{\phi_t}^{k+1}$ et le processus itératif s'arrête. Dans le cas contraire, le déplacement stoppe dès la rencontre du premier point de cassure.
3. La solution M_{k+1} est positionnée à ce point de cassure et le segment qui est situé juste après le point de cassure est sélectionné pour le calcul de la prochaine itération. Ainsi, les paramètres de ce segment seront utilisés pour le calcul de la solution à la prochaine itération (via l'équation (6.1)).

En guise d'illustration graphique, on va considérer la Figure 6.4 et assumer que la courbe en bleu est remplacée par une courbe linéaire par segments. De ce fait, le point de départ M_k appartient au segment construit à partir des points 6 et 7 du Tableau 6.1. Ce segment a été utilisé pour calculer la solution S. Puisque S est situé à droite de M_k , l'application de la méthode Katzenelson consiste à déplacer le point d'opération (initialement M_k) sur son segment et vers la droite jusqu'à la rencontre du point dont les coordonnées correspondent à ceux du point numéro 7 du Tableau 6.1. A ce moment, le segment sélectionné pour la prochaine itération est le segment construit à partir des points 7 et 8 et le point d'opération M_{k+1} est positionné au point de cassure dont les coordonnées sont données par le point 7 du Tableau 6.1.

La méthode Katzenelson permet de parcourir les segments de la caractéristique un à un sans interruption jusqu'à tomber sur le segment de la solution finale. Dans la littérature, il est dit que pour cette raison, cette méthode est capable de toujours atteindre la solution finale en un nombre fini d'itérations et ce, qu'importe le point de départ [100],[101],[102]. La convergence serait donc garantie. Cette affirmation peut toutefois être remise en question puisque des simulations ont montré qu'avec cette méthode, le point d'opération peut osciller indéfiniment entre deux segments consécutifs. Aussi, avec la méthode Katzenelson, le nombre d'itérations requises peut être très élevé, puisque la méthode parcourt les segments de la caractéristique un à un de manière consécutive.

6.2.3 La méthode de la modification de la tension itérative

Lors de la simulation de circuits contenant des composants fortement non-linéaires comme des diodes, lorsque la prédiction initiale (solution à l'itération $k=0$ et désignée $V_{\phi_t}^0$) n'est pas assez proche de la solution finale et que la prochaine solution calculée $V_{\phi_t}^1$ est telle que $V_{\phi_t}^1 > V_{\phi_t}^0$, le prochain point d'opération sur la caractéristique, $(V_{\phi_t}^1, i_{\phi_t}^1)$ peut se retrouver très loin de son prédecesseur $(V_{\phi_t}^0, i_{\phi_t}^0)$ et très loin également de la solution finale. Ceci est causé par le fait que la pente de la caractéristique de la diode augmente exponentiellement avec la tension. Dans une pareille situation, la méthode NR nécessite un nombre très élevé d'itérations pour converger. Cette problématique est expliquée dans [66] et illustrée à la Figure 6.3.

Il est possible de modifier la tension itérative de la méthode NR afin de forcer un mouvement contrôlé du point d'opération sur la caractéristique non-linéaire. La méthode qui en découle est présentée dans [66] et implémentée dans EMTP. Elle sera désignée « méthode de la modification de la tension itérative » (méthode MTI). Elle permet lorsque $V_{\phi_t}^{k+1} > V_{\phi_t}^k$, de corriger la tension par un mécanisme simple qui peut être illustré à partir de la Figure 6.4. La tension corrigée $V_{\phi_t}^{c k+1}$ est déterminée à partir de la solution calculée $V_{\phi_t}^{k+1}$ en ramenant le point S sur la caractéristique non-linéaire par un déplacement horizontal (et non vertical comme le requiert la méthode NR). Le point d'opération obtenu est le point M_{k+1}^c , identifié par la paire $(V_{\phi_t}^{c k+1}, i_{\phi_t}^{c k+1})$. Il est important de préciser que $i_{\phi_t}^{c k+1}$ est calculé à partir de l'équation de la droite tangente à la caractéristique non-linéaire en M_k . Le résultat donne :

$$i_{\phi_t}^{c k+1} = \frac{I_s}{V_T} e^{V_{\phi_t}^k/V_T} \left(V_{\phi_t}^{k+1} - V_{\phi_t}^k \right) + I_s \left(e^{V_{\phi_t}^k/V_T} - 1 \right) \quad (6.8)$$

$i_{\phi_t}^{c k+1}$ et $V_{\phi_t}^{c k+1}$ doivent satisfaire (6.4). Donc :

$$v_{\phi_t}^{c,k+1} = v_{\phi_t}^k + V_T \ln \left(1 + \frac{v_{\phi_t}^{k+1} - v_{\phi_t}^k}{V_T} \right) \quad (6.9)$$

Le point M_{k+1}^c issu de la correction de la tension représente le prochain point d'opération du processus itératif de recherche de la solution finale. Cette approche qui consiste à modifier la tension lors du processus itératif peut significativement accélérer la convergence.

6.2.4 Un schéma itératif robuste et efficient

Dans la référence [67], il est proposé une approche itérative pour la résolution des circuits contenant des composants non-linéaires représentés par le modèle linéaire par segment. Cette approche utilise deux valeurs entières prédéfinies N_B et N_K . A chaque point de temps de la simulation, la méthode NR est utilisée pour exécuter le processus itératif. Après N_B itérations, si la solution n'a pas convergé, le schéma itératif modifie la méthode. Cette dernière passe de la méthode NR à la méthode désignée NR biaxiale (*biaxial NR method*). La méthode NR biaxiale est identique à la méthode de la modification de la tension itérative (MTI). Notons que la méthode MTI [66] existait avant [67] et était accessible et présentée aux auteurs de [67] par Jean Mahseredjian en 1995. Dans le schéma itératif proposé dans [67], la méthode NR biaxiale est exécutée pour un maximum de N_K itérations. Si la solution ne converge toujours pas, le schéma itératif modifie de nouveau la méthode. Cette dernière passe de NR biaxiale à la méthode Katzenelson. Au regard du fonctionnement de ce schéma itératif, il ressort qu'il n'améliore que très peu la vitesse de convergence de la méthode NR (puisque il utilise de base la méthode NR). De plus, ses performances peuvent être significativement influencées par les valeurs choisies de N_B et N_K .

6.2.5 La méthode arc length

La méthode dite « arc length » ou méthode AL est une méthode qui s'inspire du concept de la longueur de l'arc (*arc length*). Ce dernier avait été proposée la première fois dans [103] pour solutionner les équations aux dérivées partielles avec la méthodes des éléments finis. Le concept d'arc avait été proposé pour calculer de façon plus précise les valeurs des variables aux points intermédiaires du système décomposé (ce calcul se faisait jusqu'ici par interpolation). Des versions améliorées de l'approche ont été proposées dans [104]-[106]. Elles permettaient entre autres

d'améliorer les représentations paramétriques d'arcs qui décrivent le comportement des variables des systèmes modélisés. Le concept d'arc peut être appliqué à la simulation des circuits contenant des non-linéarités. Une façon de procéder a été présentée dans [66] et le travail présenté dans [107] par l'auteur de cette thèse permet de généraliser la méthode présentée dans [66] et qui est décrite ci-après.

Il a été établi qu'avec la méthode NR, la solution v_{ϕ_t} de la tension du circuit de la Figure 6.2 à un instant t de la simulation est calculée via le processus itératif décrit par l'équation (6.6). Il est possible d'accélérer la vitesse de convergence et contrôler les valeurs itératives prises par v_{ϕ_t} en appliquant un changement de variable. Considérons le changement de variable suivant :

$$f(\rho) = v_{th_t} - R_{th_t} i_\phi(\rho) - v_\phi(\rho) = 0 \quad (6.10)$$

La variable ρ représente la longueur de l'arc de la caractéristique. Elle est définie par :

$$(d\rho)^2 = (dv_\phi)^2 + (di_\phi)^2 \quad (6.11)$$

Cette équation peut être modifiée pour devenir :

$$\left(\frac{d\rho}{dv_\phi} \right)^2 = 1 + \left(\frac{di_\phi}{dv_\phi} \right)^2 \quad (6.12)$$

Ensuite, la variable ρ peut être déterminée comme suit :

$$\rho = \int_0^{v_\phi} \sqrt{1 + (di_\phi/dv_\phi)^2} dv_\phi \quad (6.13)$$

Et finalement :

$$\rho = V_T \left\{ \ln \left[\frac{x}{1 + \sqrt{1+x^2}} \right] + \sqrt{1+x^2} \right\} - V_T \left\{ \ln \left[\frac{\theta}{1 + \sqrt{1+\theta^2}} \right] + \sqrt{1+\theta^2} \right\} \quad (6.14)$$

Avec $\theta = \frac{I_s}{V_T}$ et :

$$x = \frac{di_\phi}{dv_\phi} = \theta e^{(v_\phi/V_T)} \quad (6.15)$$

L'équation (6.15) est modifiée pour calculer v_ϕ en fonction de x :

$$v_\phi = V_T \ln\left(\frac{x}{\theta}\right) \quad (6.16)$$

Et combinée avec (6.4) pour calculer i_ϕ en fonction de x :

$$i_\phi = V_T x - I_s \quad (6.17)$$

Ensuite, la formule itérative de la méthode NR peut être appliquée pour solutionner (6.10) à tout instant t et :

$$\rho_t^{k+1} = \rho_t^k - f(\rho_t^k) \left[\frac{df}{d\rho} \Big|_{\rho_t^k} \right]^{-1} \quad (6.18)$$

A partir de l'équation (6.10), on a :

$$\frac{df}{d\rho} = -R_{th,t} \frac{di_\phi}{d\rho} - \frac{dv_\phi}{d\rho} \quad (6.19)$$

Les quantités $\frac{di_\phi}{d\rho}$ et $\frac{dv_\phi}{d\rho}$ sont calculées en utilisant (6.12) et (6.15) :

$$\frac{di_\phi}{d\rho} = \frac{x}{\sqrt{1+x^2}} \quad (6.20)$$

$$\frac{dv_\phi}{d\rho} = \frac{1}{\sqrt{1+x^2}} \quad (6.21)$$

Fondamentalement, la méthode AL détermine le point d'opération de chaque composant non-linéaire en itérant sur la longueur de l'arc qui va de l'origine de la caractéristique jusqu'au point d'opération final. Cette méthode est applicable, que le composant non-linéaire soit représenté par un modèle détaillé ou alors par un modèle linéaire par segments. Dans ce dernier cas, des tables de valeurs peuvent être construites pour exprimer les relations de dépendance décrites par les équations (6.13) à (6.17). Ainsi, pour chacune des valeurs des tensions aux points de cassure de la caractéristique linéaire par segments présentées aux Tableau 6.1 et Tableau 6.2, il est possible de calculer les valeurs correspondantes de x et de ρ , via (6.15) et (6.14). Par la suite il est possible de

construire des tables de valeurs permettant de calculer n'importe laquelle de ces variables, connaissant une autre. Le mode opératoire est présenté en annexe B.

Exécuter une itération en un instant t en appliquant la méthode AL tel que présentée dans [66] consiste à accomplir dans l'ordre les étapes suivantes, en partant du point d'opération actuel défini par la paire $(v_{\phi_t}^k, i_{\phi_t}^k)$:

1. Calculer x en utilisant (6.15) ou en utilisant la table de valeurs de $v_{\phi_t}^k$ et x .
2. Calculer (si nécessaire) ρ de l'itération k , (désigné ρ_t^k) à partir de (6.14) ou en utilisant la table des valeurs entre ρ et x .
3. Calculer ρ_t^{k+1} en résolvant (6.18). Dans (6.18), $f(\rho_t^k)$ est calculée à partir de (6.10) en prenant $v_{\phi}(\rho) = v_{\phi_t}^k$ et $i_{\phi}(\rho) = i_{\phi_t}^k$. Dans (6.18), $\frac{df}{d\rho}$ est calculé par (6.19). Dans (6.19), $\frac{di_{\phi}}{d\rho}$ et $\frac{dv_{\phi}}{d\rho}$ sont déterminés par (6.20) et (6.21) respectivement.
4. Recalculer x à partir de ρ_t^{k+1} (en se servant de la table de valeurs entre x et ρ).
5. Déterminer $v_{\phi_t}^{k+1}$ et $i_{\phi_t}^{k+1}$ à partir de (6.16) et (6.17), ce qui permet d'obtenir le prochain point d'opération $(v_{\phi_t}^{k+1}, i_{\phi_t}^{k+1})$ à partir duquel le processus itératif peut se poursuivre.

La méthode AL permet d'accélérer de manière significative la vitesse de convergence de la méthode NR. Toutefois, elle présente des limites. En effet, dans sa version actuelle, elle ne permet pas de simuler des circuits complexes, tel qu'il sera établi ci-dessous.

6.2.6 Adaptation des méthodes présentées à la formulation MANA pour la résolution des circuits plus complexes

Afin de permettre la simulation de circuits de grande taille et pouvant contenir plusieurs composants non-linéaires, les méthodes présentées ci-haut doivent être capables de s'intégrer à la

formulation MANA du système global d'équations d'un circuit électrique. En guise de rappel, ce système d'équations est présenté en (2.44) et repris ici :

$$\mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t \quad (6.22)$$

Chaque composant non-linéaire participe dans (6.22) pour la détermination de la solution globale du circuit à chaque point de temps t de la simulation. Comme il l'a été présenté au chapitre 2, la participation de chaque composant non-linéaire au système (6.22) nécessite la création de son modèle Norton. C'est donc précisément ce qui doit être fait pour chaque composant non-linéaire. Avec le circuit simple de la Figure 6.2, il a été établi que le calcul de la solution via le processus itératif de la méthode NR consiste à résoudre l'équation (6.6). Cela est équivalent à itérer avec un équivalent Norton du composant non-linéaire construit à partir de la solution précédente $(v_{\phi_t}^k, i_{\phi_t}^k)$.

Cet équivalent Norton est déterminé à partir de la caractéristique non-linéaire au point $(v_{\phi_t}^k, i_{\phi_t}^k)$.

Ainsi, au voisinage de ce point, la caractéristique non-linéaire est remplacée par l'équation linéarisée :

$$i_{\phi} = y_N v_{\phi} + i_N \quad (6.23)$$

Lorsque la caractéristique est détaillée (et que le composant est une diode), les valeurs de y_N et i_N sont calculées à chaque itération k en linéarisant (6.4) en $v_{\phi_t}^k$, ce qui donne :

$$y_N = \left. \frac{di_{\phi}}{dv_{\phi}} \right|_{v_{\phi_t}^k} = \frac{I_S}{V_T} e^{v_{\phi_t}^k / V_T} \quad (6.24)$$

$$i_N = i_{\phi_t}^k - y_N v_{\phi_t}^k \quad (6.25)$$

Si cependant la caractéristique est linéaire par segments, les valeurs de y_N et i_N du modèle Norton sont directement accessibles à partir de l'équation du segment sur lequel se trouve le point $(v_{\phi_t}^k, i_{\phi_t}^k)$. Cette équation a la forme de (6.1).

Par la suite, les valeurs de y_N et i_N participent dans \mathbf{A}_t et \mathbf{b}_t respectivement (comme expliqué au chapitre 2) pour la détermination de la solution à l'itération $k+1$ par résolution de (6.22). Cette

Résolution pour un circuit complexe est l'équivalent du calcul effectué dans (6.6). Ainsi, que ce soit par application de la méthode NR ou des autres méthodes présentées, l'algorithme de résolution des circuits non-linéaires peut être synthétisé comme suit :

1. A l'instant $t = 0$, la solution initiale du réseau est calculée (cela peut se faire par écoulement de puissance conduisant à l'initialisation de toutes les variables). Cette initialisation permet également de déterminer le point d'opération initial de chaque composant non-linéaire. En l'absence d'initialisation, toutes les valeurs sont considérées nulles.
2. La simulation avance à l'instant $t = \Delta t$ (ou $t = \Delta t/2$ si l'intégration BE est utilisée). A partir de la solution à $t = 0$, le vecteur \mathbf{b}_t est formulé. Le point d'opération obtenu au point de temps précédent pour chaque composant non-linéaire représente la prédiction initiale ($k = 0$) de la solution au point de temps présent. A partir de cette prédiction, les paramètres y_N et i_N de chaque composant non-linéaire sont déterminés et intégrés dans \mathbf{A}_t et \mathbf{b}_t respectivement. Il est utile de rappeler qu'il est d'usage de précalculer \mathbf{A}_t avant le début de la simulation et corriger par la suite certains de ses termes tout au long de la simulation (voir la section 2.2.2.3).
3. Le système (6.22) est solutionné pour obtenir la solution à l'itération $k = 1$. Avec la méthode NR, la solution trouvée représente directement la solution de l'itération $k = 1$. Cependant, avec la méthode Katzenelson ou la méthode MTI, la solution est corrigée afin de déterminer ce qui représentera la solution de l'itération $k = 1$.
4. Si la solution calculée à l'itération $k = 1$ est différente de la solution de l'itération $k = 0$, la solution de l'itération $k = 1$ est utilisée comme nouveau point d'opération pour le calcul de la solution à l'itération $k = 2$ et ainsi de suite jusqu'à ce que la convergence soit atteinte.
5. Après la convergence, l'algorithme avance au point de temps suivant $t = 2\Delta t$ (ou $t = \Delta t$ si BE est utilisée). La solution finale obtenue au point de temps précédent représente la prédiction initiale au point de temps présent et le processus itératif est de nouveau exécuté pour la détermination de la solution finale au point précédent.

6. Le même processus est reproduit aux points de temps suivant jusqu'à l'atteinte de la durée totale de la simulation.

Il est important de rappeler que seule la méthode arc length n'est pas capable d'effectuer les étapes présentées ci-dessus. En effet, pour appliquer la méthode AL, il faut connaître au point de temps t et à chaque itération, les valeurs des tension et courant de l'équivalent Thévenin aux bornes de chaque composant non-linéaire. Il s'agit des valeurs de V_{th_t} et R_{th_t} du circuit de la Figure 6.2. Si l'équivalent Thévenin aux bornes d'un composant non-linéaire n'est constitué que de sources constantes dans le temps et de composants linéaires, alors V_{th_t} et R_{th_t} sont constants tout au long de la simulation. Dans le cas contraire, ces paramètres varient à chaque itération parce que le circuit simulé peut contenir plusieurs composants non-linéaires qui changent de valeur à chaque itération. La méthode AL (comme on le verra plus bas) permet d'accélérer véritablement la vitesse de convergence de la méthode NR. Mais dans sa version actuelle, elle ne s'intègre pas à la formulation MANA et n'est donc pas applicable pour la simulation des circuits complexes. Pour résoudre ce problème, l'auteur de cette thèse propose une approche qui est présentée dans [107] et qui constituera la prochaine section de cette thèse.

6.3 Adaptation de la méthode arc length à la formulation MANA

Le principal défi à relever afin de permettre l'utilisation de la méthode arc length lorsque les équations du circuit à simuler sont formulées avec l'approche MANA consiste à déterminer à chaque point de temps et à chaque itération l'équivalent Thévenin (tension et résistance) aux bornes de chacun des composants non-linéaires qui constituent le circuit. Une façon de procéder est présentée ci-dessous. Aussi, afin d'augmenter les chances de convergence de la méthode obtenue, un mécanisme additionnel (qui est également présenté) est rajouté. L'ensemble constitue un schéma itératif très efficient pour la simulation des circuits contenant des non-linéarités. Ce schéma sera désigné « *méthode arc length adaptée* » ou méthode ALA.

6.3.1 Détermination de l'équivalent Thevenin aux bornes de chaque composant à chaque itération

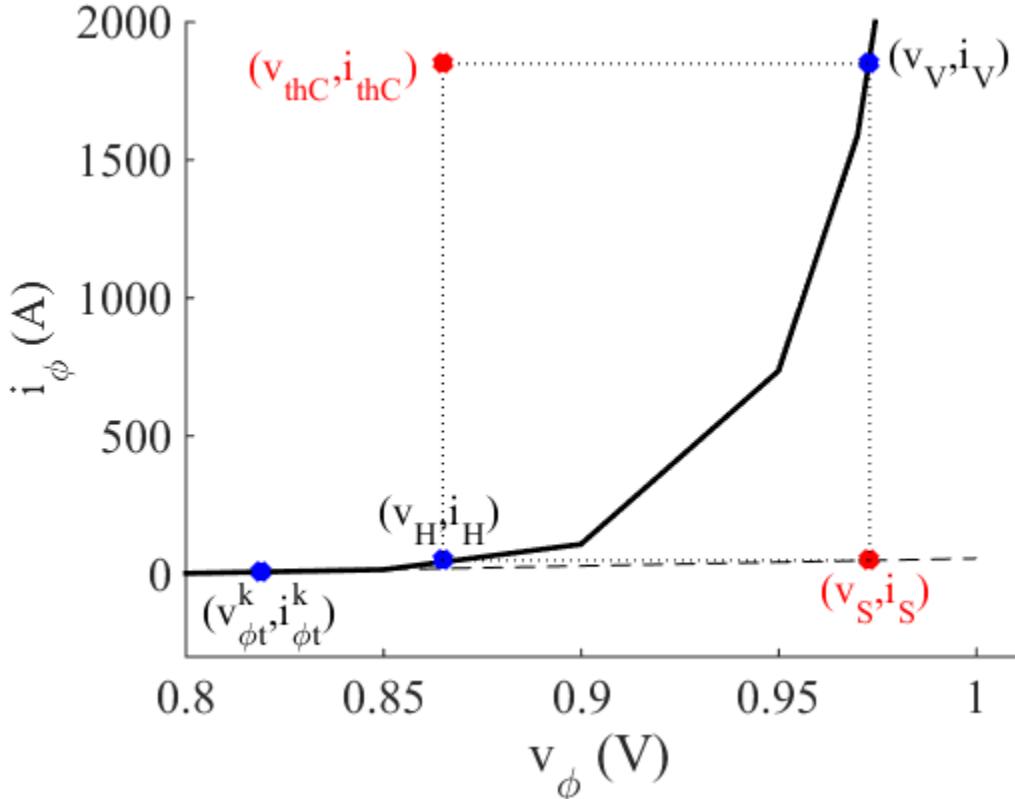


Figure 6.5: Caractéristique linéaire par segments

La Figure 6.5 représente la caractéristique linéaire par segments d'un composant (ici une diode).

À un instant de simulation donné, à partir du point d'opération $(v_{\phi_t}^k, i_{\phi_t}^k)$, la résolution du système

(6.22) permet d'obtenir la solution (v_s, i_s) . Le prochain possible point d'opération $(v_{\phi_t}^{k+1}, i_{\phi_t}^{k+1})$ doit

être égal à (v_V, i_V) si la méthode NR est utilisée ou alors (v_H, i_H) si la méthode MTI est appliquée.

En réalité, la paire (v_s, i_s) représente l'intersection de la caractéristique linéarisée du composant non-linéaire et de celle de l'équivalent Thevenin aux bornes du composant. Cette dernière est décrite par une équation de la forme de (6.7). Il est clair que la paire (v_s, i_s) est un point de la caractéristique de cet équivalent Thevenin. Tout ce qui reste à faire, c'est trouver un autre point (v_{thC}, i_{thC}) de cette caractéristique. Ainsi, V_{th_t} et R_{th_t} de la formule (6.7) seront déterminés par :

$$R_{th_t} = -\frac{v_{thC} - v_s}{i_{thC} - i_s} \quad (6.26)$$

$$v_{th_t} = R_{th_t} i_s + v_s \quad (6.27)$$

En pratique, il n'existe pas d'information disponible pour déterminer le second point (v_{thC}, i_{thC}) . Il est toutefois possible d'en faire une prédition réaliste. En effet, puisque l'intersection entre les caractéristiques du Thevenin et du composant non-linéaire doit se produire quelque part entre (v_H, i_H) et (v_V, i_V) , une prédition réaliste consiste à considérer le dernier sommet du rectangle construit à partir de (v_s, i_s) , (v_H, i_H) et (v_V, i_V) . Le résultat est le point (v_{thC}, i_{thC}) sur la Figure 6.5. Cette prédition permet de calculer R_{th} et v_{th} en utilisant (6.26) et (6.27). Le même processus est répété à chaque itération pour mettre à jour les valeurs de v_{th_t} et R_{th_t} . Ces valeurs sont utilisées à chaque itération lors de l'exécution des étapes de la méthode ALA.

6.3.2 Détection et élimination des boucles infinies

Dans la simulation des circuits non-linéaires, la principale source de non-convergence dans le processus itératif à chaque point de temps est l'apparition de boucles infinies. Il s'agit d'un ensemble fini de points explorés de manière répétitive dans le processus de recherche du point d'opération final. Il est important de pouvoir détecter l'apparition des boucles infinies et les éliminer, autrement la simulation va tourner indéfiniment au même point de temps sans jamais avancer. Il est possible à la suite de la détection, d'envisager un changement de méthode ou alors l'arrêt des itérations et le passage au point de temps suivant. Dans cette thèse, il est proposé de compléter la méthode ALA par une technique capable de détecter et d'éliminer les boucles. Cette technique est applicable lorsque la représentation linéaire par segments est adoptée. La technique nécessite l'identification préalable des différents segments de la caractéristique. Elle fonctionne comme suit : Après l'itération k , le segment S_k qui est le segment de la caractéristique contenant la paire $(v_{\phi_t}^k, i_{\phi_t}^k)$ est sauvegardé en position k d'un vecteur qui sera désigné \mathbf{V}_{seg} . A la suite d'une itération quelconque n , si la solution calculée se trouve sur le segment nommé S , alors le vecteur \mathbf{V}_{seg} est parcouru dans le but de rechercher la position m de la plus récente itération pour laquelle

la solution obtenue était S . Si m existe, alors $S = S_n = S_m$. La prochaine étape consiste à vérifier si $S_{n-1} = S_{m-1}$. Si oui, l'algorithme considère qu'il s'agit du début d'une boucle infini. À partir de ce moment, le processus itératif bascule vers la méthode Katzenelson et ce, jusqu'à ce que le point d'opération final soit trouvé.

Cette technique de détection et d'élimination des boucles, combinée à la méthode ALA constitue un puissant schéma itératif qui garantit et accélère la convergence à chaque point de temps de la simulation comme il sera établi dans la prochaine partie.

6.4 Simulations et discussions

Dans cette partie, des circuits contenant des composants fortement non-linéaires (diodes et autres interrupteurs d'électronique de puissance) sont simulés avec la méthode ALA proposée. Des comparaisons sont faites avec la méthode Newton Raphson (NR) et la méthode de la modification de la tension itérative (MTI). Les critères de comparaison incluent la précision des résultats et le nombre total d'itérations requises par chaque méthode pour compléter chacune des différentes simulations. Afin d'établir des critères équitables de comparaison (notamment prévoir et garantir le même nombre de points de temps de la simulation pour chacune des méthodes), l'intégration numérique BE sera utilisée pour toutes les méthodes qui seront testées.

6.4.1 Simulation d'un simple circuit à diode

Ce premier cas de simulation consiste simplement à calculer la tension v_ϕ dans le circuit de la Figure 6.2 dans sa version monophasée. Le composant non-linéaire est une diode décrite par son modèle détaillé de l'équation (6.4). On prendra $R_{th} = 0.1\Omega$ et $v_{th} = 10V$. Toutes les trois méthodes testées aboutissent à la même valeur finale $v_\phi = 0.89557V$. En modifiant les valeurs de la prédiction initiale, on a mesuré le nombre d'itérations requises par chaque méthode pour atteindre la solution. Les résultats sont présentés dans le Tableau 6.3.

Ces résultats montrent que la méthode NR est très sensible à la prédiction initiale. Sa performance est meilleure lorsque cette prédiction est relativement proche de la solution finale. Cette exigence est difficile à satisfaire lors de la simulation de réels circuits car il n'est pas possible de connaître à l'avance la solution finale et de sélectionner la prédiction initiale en conséquence. Pour cette

raison, on peut donc s'attendre à ce que la méthode NR requiert dans certaines situations, un nombre très élevé d'itérations pour converger. La méthode MTI quant à elle nécessite beaucoup moins d'itérations que la méthode NR. De plus, elle semble moins sensible à la prédiction initiale. Finalement, pour ce circuit, on voit clairement que la méthode AL est complètement insensible à la prédiction initiale et est particulièrement efficace pour la simulation de ce circuit.

Tableau 6.3: Nombre d'itérations selon la prédiction initiale (modèle détaillé de la diode)

Prédiction initiale (V)	Méthode NR	Méthode MTI	Méthode AL
0	356	9	5
0.7	299	4	5
0.8	37	4	5
1.0	9	9	5

Afin de démontrer l'efficience de la méthode AL, les différents points de la caractéristique explorés par les méthodes MTI et AL lors des itérations successives menant vers la solution finale sont représentés à la Figure 6.6. La prédiction initiale est prise à l'origine (tension égale à zéro).

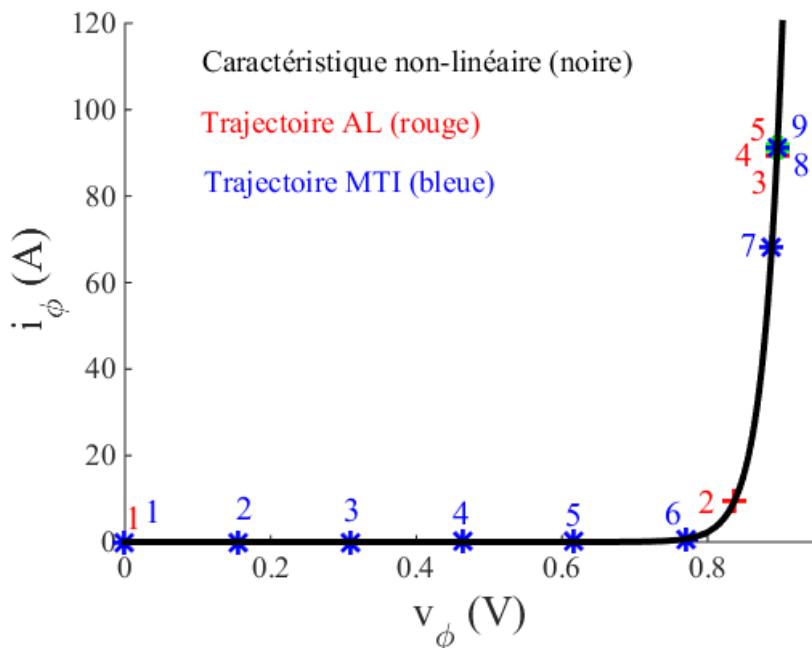


Figure 6.6: Points explorés par les méthodes MTI et AL lors de la recherche de la solution

Il ressort de cette figure que la méthode AL converge très vite. A la 3^{ième} itération, le point exploré est déjà très proche de la solution finale tandis qu'avec la méthode MTI, ce n'est qu'à la 8^{ième} itération que le point exploré devient suffisamment proche de la solution finale. Ceci démontre l'efficacité de la méthode AL.

Des résultats de simulation non présentés ici montrent que lorsque l'on utilise la méthode ALA proposée pour simuler ce circuit, le nombre d'itérations nécessaires en partant de la même prédiction initiale est égal à 8. On voit bien que le fait que la méthode ALA fasse elle-même sa propre prédiction de la valeur de l'équivalent Thévenin aux bornes de la diode fait que le nombre d'itérations nécessaires devient supérieur à celui qu'on observe avec AL. Toutefois, ce nombre d'itérations reste inférieur à celui requis par la méthode MTI. De plus, vu que l'utilisation de la méthode AL n'est possible que pour un circuit très simple contenant un seul composant non-linéaire, alors on peut en conclure que la méthode ALA demeure une méthode très intéressante pour la simulation des circuits non-linéaires comme on le verra dans les prochains cas.

6.4.2 Simulation d'un convertisseur DC-AC-DC

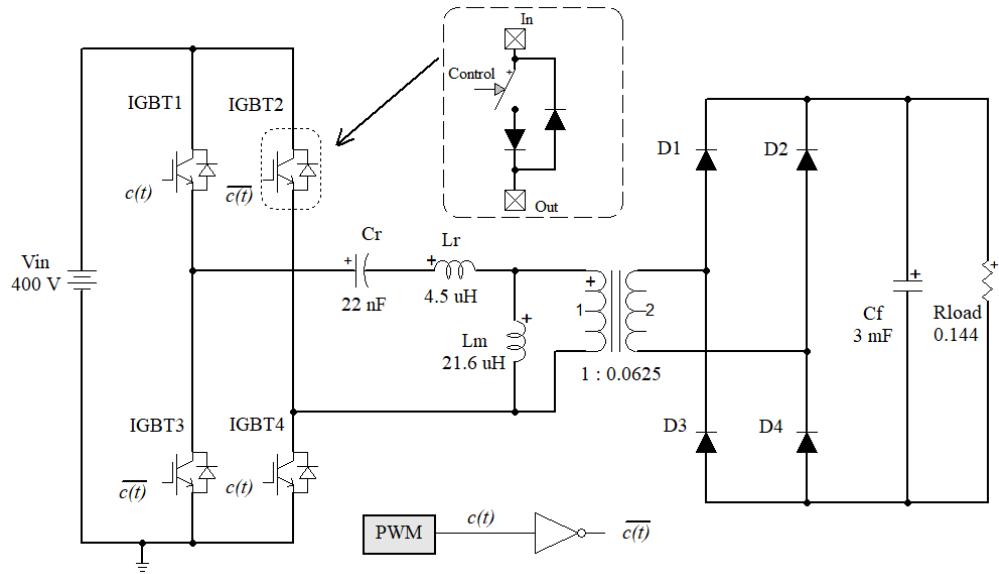


Figure 6.7: Convertisseur DC-AC-DC avec des interrupteurs non-linéaires

Le circuit testé dans cette partie est celui de la Figure 5.6 (repris à la Figure 6.7) dans lequel tous les interrupteurs sont représentés par des modèles non-linéaires. Ainsi, les IGBT sont toujours modélisés comme indiqué à la Figure 3.7 tandis que les diodes sont représentées par un modèle

linéaire par segments. Les données de ce modèle sont présentées aux Tableau 6.1 et Tableau 6.2. L'un des objectifs de ce cas de simulation est la comparaison de la précision des résultats entre une simulation utilisant un modèle idéal et une simulation utilisant un modèle non-linéaire.

Le circuit est simulé pendant 1ms en utilisant différents pas d'intégration Δt . Le Tableau 6.4 présente le nombre de fois que le système (6.22) est solutionné avec chaque méthode pour compléter la simulation. Entre parenthèses à coté de chaque résultat, il est indiqué le nombre moyen de résolution du système d'équations par point de temps. Par exemple, pour $\Delta t = 20\text{ ns}$, le nombre de points de temps de la simulation est $N_s = 1\text{ ms}/20\text{ ns} = 50\,000$. Puisque la méthode ALA exécute un total de 69 686 résolutions du système, le nombre moyen de solutions par point de temps sera donc de $69\,686/50\,000 = 1.39$.

Tableau 6.4: Nombre de solutions du système (et nombre moyen de solutions par point de temps) pour chaque méthode selon le pas d'intégration (modèle linéaire par segments)

Méthode	$\Delta t = 20\text{ ns}$	$\Delta t = 40\text{ ns}$	$\Delta t = 80\text{ ns}$
Méthode NR	98 154 (1.96)	56 750 (2.27)	31 875 (2.55)
Méthode MTI	74 167 (1.48)	42 992 (1.72)	25 130 (2.01)
Méthode ALA	69 686 (1.39)	40 541 (1.62)	22 729 (1.82)

Il est important de préciser que le nombre de solutions du système s'obtient en additionnant les solutions normales à chaque point de temps et les solutions additionnelles dues aux itérations à chaque point de temps. Les résultats du Tableau 6.4 montrent que la méthode MTI est significativement plus efficace que la méthode NR. Toutefois la méthode ALA proposée est la plus efficace des trois, puisqu'elle est capable de compléter la simulation en un nombre minimum d'itérations, quel que soit le pas d'intégration. Par ailleurs, on peut remarquer que, pour toutes les méthodes, le nombre moyen de résolutions par point de temps augmente avec le pas d'intégration. Aussi, il faut préciser que l'écart entre les performances de MTI et ALA n'est pas très élevé tandis que l'écart de performance entre chacune de ces 2 méthodes et la méthode NR est significatif.

Des résultats de simulation non représentés montrent que toutes les méthodes testées produisent le même niveau de précision. Pour l'illustrer, on a représenté à la Figure 6.8 la forme d'onde de la

tension aux bornes de la charge R_{load} pour chacune des trois méthodes testées avec $\Delta t = 40 \text{ ns}$. Pour des besoins de comparaison, les résultats d'une simulation additionnelle dans laquelle les diodes sont représentées par le modèle idéal de la Figure 3.8 sont rajoutés.

Cette figure montre que les méthodes NR, MTI et ALA appliquées au modèle non-linéaire conduisent exactement à la même précision. En revanche, lorsque le modèle idéal est utilisé pour représenter les diodes, les formes d'onde obtenues présentent une erreur. On peut en effet observer une erreur absolue en régime permanent de l'ordre de 1.0 V. Le modèle idéal est populaire parce qu'il permet d'accélérer les simulations. Toutefois il est important de remarquer comme on vient de le voir que dans certains cas, ce modèle peut dégrader la précision.

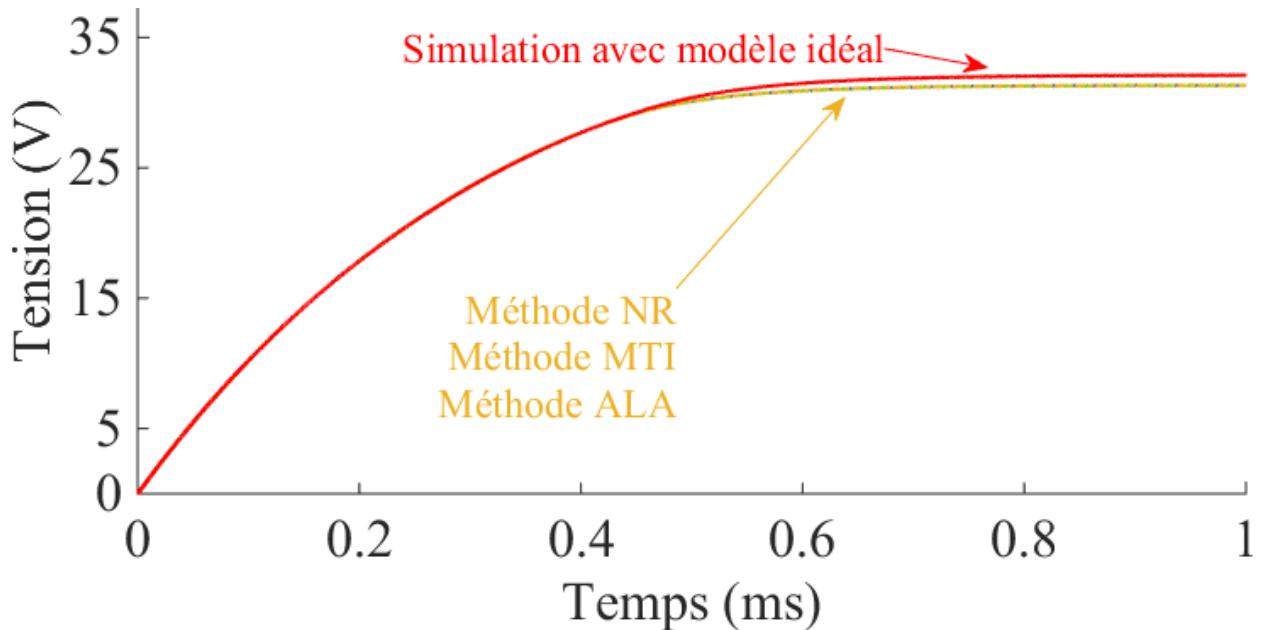


Figure 6.8: Tension aux bornes de la charge pour $\Delta t = 40 \text{ ns}$ avec les méthodes testées

En conclusion, ce cas de simulation permet de confirmer que la méthode ALA (méthode proposée) est la plus rapide et conduit exactement à la même précision que les deux autres méthodes.

6.4.3 Simulation d'un convertisseur triphasé

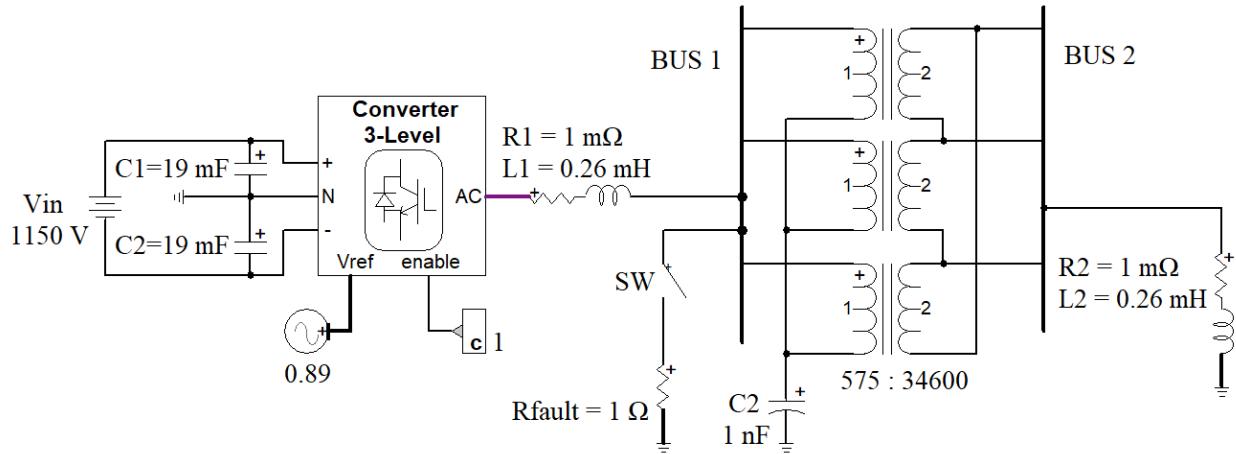


Figure 6.9: Schéma du circuit convertisseur triphasé

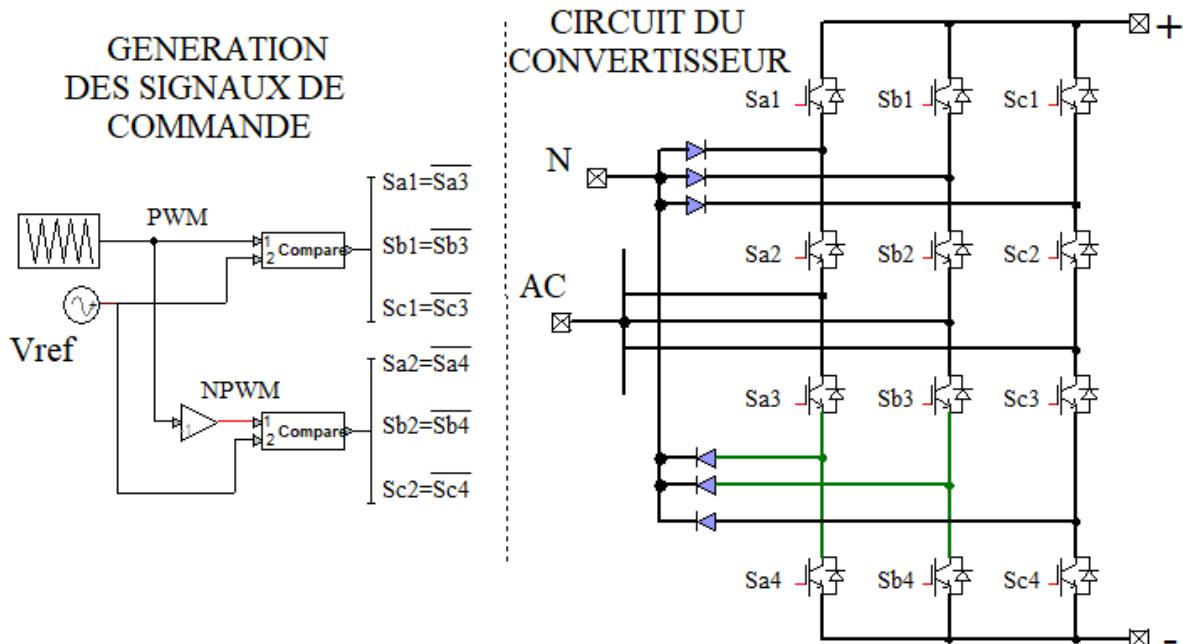


Figure 6.10: Schéma interne du convertisseur

Dans cet exemple, le circuit de la Figure 6.9 est simulé dans le but d'évaluer l'efficacité de la méthode proposée lorsque le nombre de composants non-linéaires du circuit simulé est relativement élevé. Ce convertisseur, dont le schéma interne est présenté à la Figure 6.10, est constitué de $4 \times 3 = 12$ IGBT. Chaque IGBT est modélisé comme indiqué à la Figure 3.7 et chaque diode est représentée par le modèle linéaire par segment dont les données se trouvent aux Tableau

6.1 et Tableau 6.2. Avec les 6 diodes additionnelles, le nombre total de composants non-linéaires sur cette figure est $12 \times 2 + 6 = 30$. Les signaux de commande des interrupteurs contrôlés sont générés par une source PWM. La porteuse triangulaire génère un signal d'amplitude 0.5, de fréquence 2500 Hz avec un offset de 0.5. La tension de référence provient d'un système triphasé direct, d'amplitude 0.89, de fréquence 60 Hz et de phase égal à 10 degrés pour la phase a.

Un défaut triphasé est appliqué au BUS 1 à 10ms et est éliminé à 20 ms. Le circuit est simulé pendant 30 ms, avec trois valeurs de pas d'intégration : $2\mu s$, $5\mu s$ et $10\mu s$.

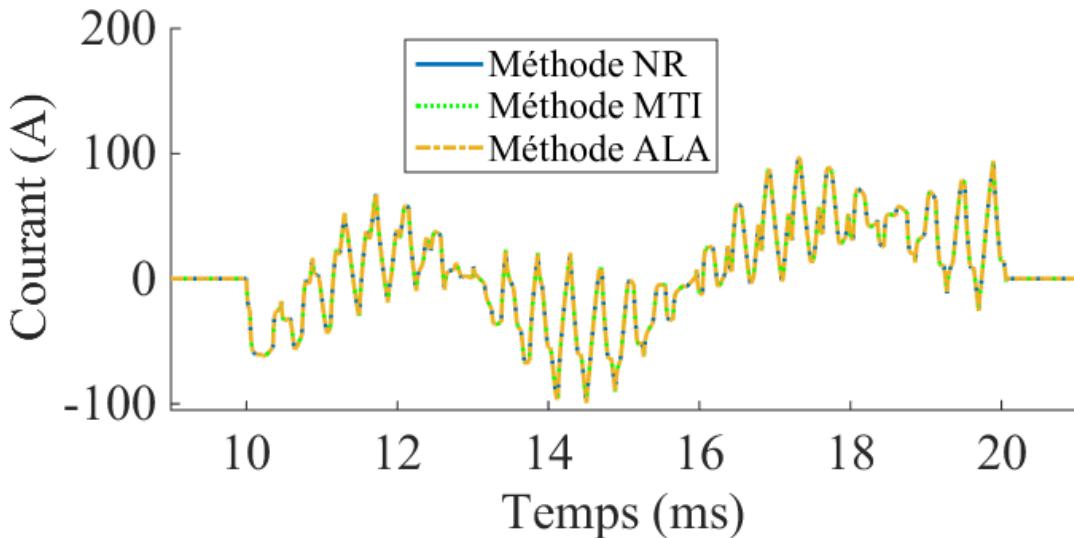


Figure 6.11: Forme d'onde du courant dans la phase a de SW pour $\Delta t=5\mu s$

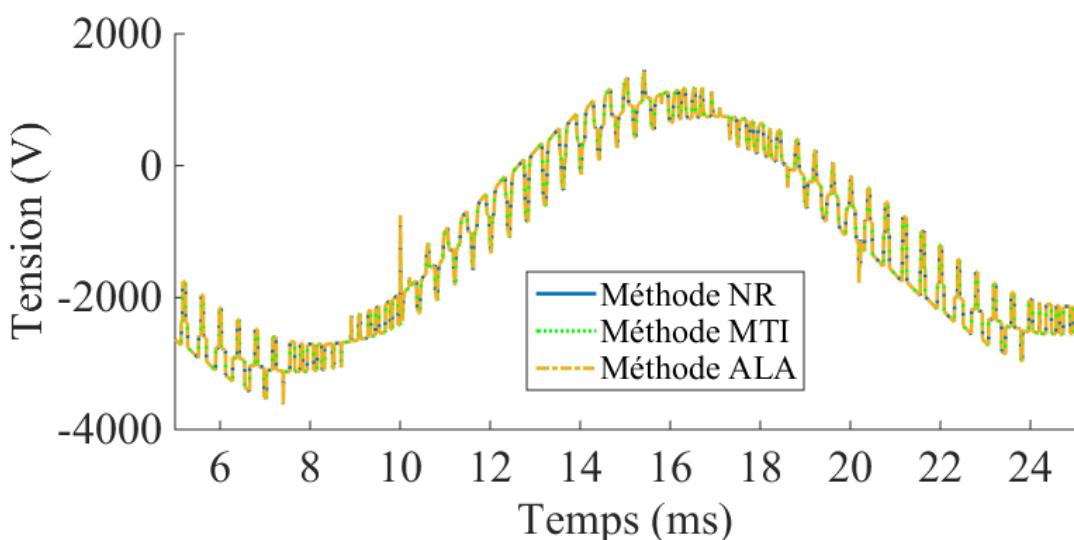


Figure 6.12: Forme d'onde de la tension de phase a de la charge au BUS 2 pour $\Delta t=5\mu s$

Les résultats de simulation montrent que pour chaque valeur du pas d'intégration, toutes les méthodes produisent exactement le même niveau de précision. En guise d'illustration, la Figure 6.11 et la Figure 6.12 présentent pour chacune des méthodes testées, le courant de défaut dans la phase a de SW et la tension phase a de la charge connectée au BUS 2 pour $\Delta t = 5\mu s$.

Le Tableau 6.5 présente le nombre de solutions du système requises par chaque méthode pour compléter la simulation, dépendamment du pas d'intégration. Comme pour le cas de simulation précédent, il est représenté à la suite de chaque valeur et entre parenthèses, le nombre de résolutions du système d'équations par point de temps de la simulation.

Tableau 6.5: Nombre de solutions du système (et nombre moyen de solutions par point de temps) pour différentes méthodes et selon le pas d'intégration (modèle linéaire par segments)

Méthode	$\Delta t = 2\mu s$	$\Delta t = 5\mu s$	$\Delta t = 10\mu s$
Méthode NR	165 547 (11.04)	75 628 (12.6)	33 947 (11.32)
Méthode MTI	26 397 (1.76)	12 404 (2.07)	7 843 (2.61)
Méthode ALA	25 462 (1.70)	11 143 (1.86)	6 704 (2.23)

Tout comme on a pu le constater avec la simulation du circuit de la Figure 6.7, la méthode MTI est bien plus efficace que la méthode NR. Toutefois, la méthode ALA proposée est la plus efficace car elle requiert le moins d'itérations, quelle que soit la valeur du pas d'intégration. Par exemple, pour $\Delta t = 5\mu s$, la méthode ALA proposée est près de 7 fois plus rapide que la méthode NR tandis que la méthode MTI est environ 6 fois plus rapide que NR. Ces résultats démontrent l'efficacité de l'approche arc length pour la simulation des circuits fortement non-linéaires.

Dans le processus itératif de recherche de la solution finale à chaque point de temps, le nombre d'itérations nécessaires varie d'un point de temps à un autre. Ceci est valable pour toutes les méthodes testées et est dû au fait que, pour chaque composant non-linéaire du circuit, le point d'opération de départ du processus itératif (prédition initiale) est plus ou moins éloigné du point d'opération final. Pour une méthode donnée, de larges variations entre les nombres d'itérations effectuées à différents points de temps traduisent une forte sensibilité de la méthode aux conditions initiales. L'évaluation de la sensibilité des méthodes a été réalisée en 6.4.1 pour un petit circuit

dans lequel il était possible de définir volontairement la valeur de la prédiction initiale. Cela n'est pas possible avec le cas de la Figure 6.9, puisque qu'à chaque point de temps, la prédiction initiale correspond simplement à la solution du point de temps précédent. Une façon de quantifier la sensibilité d'une méthode consiste lors d'une simulation, à calculer l'écart type (standard deviation) de la distribution des données correspondante au nombre total d'itérations effectuées à chaque point de temps. Pour cela, il suffit d'enregistrer le nombre d'itérations effectuées à chaque point de temps de la simulation, puis de calculer la moyenne et l'écart type des valeurs obtenues. Les résultats sont présentés au Tableau 6.6.

Tableau 6.6: Tableau des écarts types des méthodes testées selon le pas d'intégration

Méthode	$\Delta t = 2\mu s$	$\Delta t = 5\mu s$	$\Delta t = 10\mu s$
Méthode NR	5.96	6.45	7.30
Méthode MTI	1.33	1.95	2.61
Méthode ALA	0.92	1.30	1.69

Le Tableau 6.6 montre que, quel que soit le pas d'intégration, l'écart type de la méthode NR est très élevé, ce qui prouve que cette méthode est très sensible aux conditions initiales. Avec MTI, on note une sensibilité significativement plus faible qu'avec NR. La méthode ALA est clairement la moins sensible. Toutes ces observations sont parfaitement en accord avec celles établies en 6.4.1.

Toutes les simulations et analyses présentées dans ce chapitre démontrent que la méthode arc length adaptée proposée est la plus efficace pour la simulation des circuits non-linéaires.

CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS

7.1 Synthèse de la recherche

L'objectif visé dans cette thèse est le développement de nouvelles méthodes pour le traitement des discontinuités lors de la simulation dans le domaine du temps des transitoires de réseaux et des circuits d'électronique de puissance. Ce document commence par préciser le contexte, les motivations ainsi que l'état des travaux existants. Il en ressort que la simulation dans le domaine du temps fait face à de nombreux défis. Dans la simulation des réseaux de transport, les défis majeurs sont liés à la taille grandissante des réseaux et à la complexité des phénomènes transitoires qui y apparaissent à la suite de discontinuités. Ces éléments imposent des temps de calculs très élevés et des risques de pertes en précision dans les résultats. Dans le domaine de la simulation des circuits d'électronique de puissance, le maintien de la précision à la traversée des discontinuités représente le défi majeur identifié. En effet, les discontinuités sont sources d'erreurs et compromettent la précision. Les résultats des travaux dans la littérature comportent des limites qui motivent la présente recherche.

Dans cette thèse, la simulation à pas d'intégration numérique variable comme solution permettant à la fois d'accélérer les simulations mais aussi d'améliorer la précision des résultats est examinée. Deux algorithmes sont présentés. Le premier est dédié à la simulation des grands réseaux tandis que le second est dédié à la simulation des circuits d'électronique de puissance. Les choix des pas d'intégration et les logiques proposées pour l'adaptation du pas d'intégration garantissent l'amélioration de la précision et de la vitesse d'exécution des simulations. Toutefois, les algorithmes proposés ne tiennent pas compte des composants complexes tels que les machines, ce qui limite leur champ d'application. Les algorithmes proposés fournissent néanmoins des moyens de réduire significativement les temps de calcul tout en conservant la précision. Leurs performances très intéressantes sont dues à la pertinence des stratégies de changement du pas d'intégration qu'ils implémentent. Aucune approche similaire avec une efficacité avérée n'a été jusqu'ici présentée dans la littérature, ce qui démontre l'impact et l'importance des algorithmes développés dans cette thèse. Cela démontre aussi et surtout leur utilité dans tout travail à venir de conception d'algorithmes à pas variables pour la simulation de circuits plus complexes.

Cette thèse contribue également à améliorer les méthodes de simulation des circuits d'électronique de puissance. Les principales sources d'erreur dans la simulation de tels circuits, ainsi que les techniques permettant de les minimiser ou les éviter sont présentées. L'impact de ces techniques sur la précision est également analysé. Il est établi que, la détection des discontinuités par interpolation linéaire, la suppression des oscillations numérique par l'intégration Backward Euler et la réinitialisation des variables par l'approche *simultaneous switching* représentent les principales techniques qui permettent une gestion optimale des discontinuités. De plus, l'instant exact auquel une discontinuité apparaît entre deux points de temps consécutifs peut impacter différemment sur la précision, dépendamment de la technique de gestion des discontinuités qui est utilisée.

Sur la base des observations et analyses établies, les exigences d'une méthode de simulation permettant d'adresser convenablement les problématiques liées à la gestion des discontinuités sont énumérées dans cette thèse. Trois nouvelles méthodes de simulation sont présentées et testées. Toutes conduisent à des résultats plus précis que ceux obtenus avec les méthodes existantes. En particulier, nous avons proposé une méthode meilleure que toutes celles qui existent actuellement dans les logiciels de simulation. Cette méthode utilise l'intégration trapézoidale pour avancer dans le temps, l'interpolation linéaire pour la détection précise des discontinuités, ainsi que la réinitialisation des variables et la suppression des possibles oscillations numériques par la méthode Backward Euler. Il s'agit de la méthode TR_BE_I_SS_m proposée. Toutefois, il est constaté une légère augmentation de l'effort de calcul chez les nouvelles méthodes proposées (par comparaison aux méthodes existantes). Mais le gain considérable en précision justifie largement la faible complexité supplémentaire liée à l'utilisation de ces nouvelles méthodes.

Cette thèse s'intéresse enfin au problème de la simulation des circuits contenant des composants non-linéaires. Les limites des méthodes de simulation existantes sont exposées. Une formulation adaptée de la méthode arc length est proposée. Elle permet d'intégrer la méthode arc length à la formulation MANA pour la simulation des circuits contenant plusieurs non-linéarités. Les simulations ont montré que l'approche proposée accélère la convergence de la simulation à chaque point de temps tout en maintenant la précision dans les résultats. Elle est beaucoup plus rapide que la méthode Newton standard et présente les mêmes garanties de convergence. La contribution de cette thèse dans le domaine de la simulation des circuits non-linéaires est significative dans la mesure où il n'existe dans la littérature aucun travail qui a réussi à adapter l'utilisation de la

méthode arc length à la formulation nodale modifiée augmentée (MANA) des équations d'un circuit électrique.

Les résultats présentés dans cette thèse apportent donc des améliorations aux méthodes et techniques de simulation existantes. Toutefois des problèmes non résolus persistent, ouvrant la voie vers d'autres pistes de recherche.

7.2 Perspectives de la recherche

Les limites des méthodes proposées et l'existence de problèmes non solutionnés dégagent de nouvelles pistes de recherche telles que :

- Le développement d'algorithmes de simulations à pas d'intégration numérique variable pour la simulation des circuits contenant des composants plus complexes tels que les machines et les lignes de transport. Ces algorithmes devront être capables de s'adapter aux contraintes imposées par ces composants.
- Le développement de méthodes qui utilisent l'interpolation et/ou la variation du pas d'intégration, et qui sont capables de réduire davantage l'erreur due à l'interpolation à la traversée des discontinuités. Ces méthodes idéalement devront éviter le retour en arrière dans la simulation chaque fois qu'une discontinuité apparaît.
- Le développement de méthodes plus précises de simulation des circuits d'électronique de puissance qui ont un niveau de complexité de mise en œuvre moins élevé que celui des méthodes proposées.
- Le développement de nouvelles méthodes plus rapides et avec garantie de convergence pour la simulation des circuits non-linéaires.

LISTE DE RÉFÉRENCES

- [1] D. Paré, G. Turmel, J.-C. Soumagne, V. Q. Do, S. Casoria, M. Bissonnette, B. Marcoux and D. McNabb, "Validation Tests of the Hypersim Digital Real Time Simulator with a Large AC-DC Network," *International Conference on Power Systems Transients*, New Orleans, USA, Sept. 28 - Oct. 2, 2003.
- [2] R. Kuffel, J. Giesbrecht, T. Maguire, R. P. Wierckx, P. McLaren, "RTDS – A fully digital Power System Simulator Operating in Real Time," Proc. of *IEEE WESCANEX 95. Communications, Power and Computing Conf.*, Vol. 2, May 1995, pp. 300-305.
- [3] J. A. Hollman, J. R. Marti, "Real time network simulation with PC-cluster," *IEEE Trans. on Power Systems*, Vol. 18, No.2, May 2003, pp. 563-569.
- [4] J. Bélanger, V. Lapointe, C. Dufour, L. Schoen, "eMEGASim: An Open High-performance Distributed Real-Time Power Grid Simulator. Architecture and Specification," *International Conf. on Power Systems (ICPS'07)*, Bangalore, India, December 12-14, 2007.
- [5] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian and L. Gérin-Lajoie, "On a new approach for the simulation of transients in power systems," *Electric Power Systems Research*, vol. 77, no. 11, pp. 1514-1520, 2007.
- [6] J. Mahseredjian, U. Karaagac, S. Dennetière and H. Saad, "Simulation of electromagnetic transients with EMTP-RV," in *Numerical Analysis of Power System Transients and Dynamics*, London, U.K.:IET, pp. 103-134, Jan. 2015.
- [7] O. Anaya-Lara and E. Acha, "Modeling and analysis of custom power systems by PSCAD/EMTDC," *IEEE Trans. on Power Delivery*, Vol. 17, No. 1, Jan. 2002, pp. 266-272.
- [8] J. Mahseredjian, "Simulation des transitoires électromagnétiques dans les réseaux électriques," Édition 'Les Techniques de l'Ingénieur', February 10, 2008, Dossier D4130, 12 pages.
- [9] J. Mahseredjian, Notes du cours ELE8457, Génie Électrique, École polytechnique de Montréal, 2017.
- [10] J. Mahseredjian, A. Xémard, B. Khodabakhchian, "Régimes transitoires dans les réseaux électriques," Édition 'Les Techniques de l'Ingénieur', Novembre, 2007, Dossier D82, 18 pages.
- [11] J. Mahseredjian, V. Dinavahi and J. A. Martinez, "Simulation Tools for Electromagnetic Transients in Power Systems: Overview and Challenges," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1657 - 1669, 2009.
- [12] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis Of Electronic Circuits: Algorithms And Computational Techniques*, New Jersey: Prentice-hall, INC., 1975.
- [13] C.W. Ho, R. Ruehli and P.A. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol.-CAS-22, no. 6, pp. 504-509, 1975.
- [14] G. Hachtel, R. Brayton and F. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," in *IEEE Transactions on Circuit Theory*, vol. 18, no. 1, pp. 101-113, January 1971.
- [15] J. Mahseredjian, I. Kocar and U. Karaagac, *Solution Techniques for Electromagnetic Transients in Power Systems. Power System Transients*, Wiley Online Library, 2014.
- [16] A. Ametani, "Numerical Analysis of Power System Transients and Dynamics," IET (The Institution of Engineering and Technology), 2015.

- [17] C. Dufour, J. Mahseredjian and J. Bélanger, "A Combined State-Space Nodal Method for the Simulation of Power System Transients," in *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 928-935, April 2011.
- [18] J. Mahseredjian, C. Dufour, U. Karaagac, J. Bélanger: "Simulation of power system transients using state-space grouping through nodal analysis", Proc. of *International Conference on Power Systems Transients*, IPST 2011 in Delft, Netherlands, 14-17 June, 2011.
- [19] <https://www.atpdraw.net/>
- [20] <https://www.pscad.com/>
- [21] D. A. Woodford, A. M. Gole, and R. Z. Menzies, "Digital simulation of dc links and ac machines," *IEEE Trans. Power App. Syst.*, vol. 102, no. 6, pp. 1616–1623, Jun. 1983.
- [22] S. Fan and H. Ding, "Time Domain Transformation Method for Accelerating EMTP Simulation of Power System Dynamics," in *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1778-1787, Nov. 2012.
- [23] A. S. Morched and V. Brandwajn, "Transmission network equivalents for electromagnetic transients studies," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 9, pp. 2385–2994, Sep. 1983.
- [24] J. Morales, J. Mahseredjian, A. Ramirez, K. Sheshyekani, I. Kocar, "A Loewner/MPM – VF Combined Rational Fitting Approach, " *IEEE Trans. on Power Delivery*, Vol. 35, No. 2, pp. 802-808, 2019.
- [25] J. Morales, J. Mahseredjian, K. Sheshyekani, A. Ramirez, E. Medina, "Pole-Selective Residue Perturbation Technique for Passivity Enforcement of FDNEs, " *IEEE Transactions on Power Delivery*, Vol. 33, No. 6, pp. 2746-2754, 2018.
- [26] U. D. Annakkage, N.-K. C. Nair, A. M. Gole, V. Dinavahi, T. Noda, G. Hassan, and A. Monti, "Dynamic system equivalents: A survey of available techniques, " in Proc. *IEEE Power & Energy Society General Meeting (PES'09)*, Calgary, AB, Canada, pp. 1–5, Jul. 2009.
- [27] J. Morales Rodriguez, E. Medina, J. Mahseredjian, A. Ramirez, K. Sheshyekani and I. Kocar, "Frequency-Domain Fitting Techniques: A Review," in *IEEE Transactions on Power Delivery*, vol. 35, no. 3, pp. 1102-1110, June 2020.
- [28] J. Morales, Computation of frequency dependent network equivalents using vector fitting, matrix pencil method and loewner matrix, these de doctorat, École Polytechnique de Montréal, 2019.
- [29] M. D. Heffernan, K. S. Turner, J. Arrillaga, and C. P. Arnold, "Computation of A.C.-D.C. system disturbances: Part I, II, and III," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 11, pp. 4341–4363, Nov. 1981.
- [30] H. T. Su, L. A. Snider, T. S. Chung, and D. Z. Fang, "Recent advancements in electromagnetic and electromechanical hybrid simulation," in Proc. Int. Conf. Power Syst. Technology, Singapore, Nov. 2004, pp. 1479–1484.
- [31] A. van der Meer, M. Gibescu, M. van der Meijden, W. Kling, and J. Ferreira, "Advanced hybrid transient stability and EMT simulation for VSC-HVDC systems," *IEEE Trans. Power Del.*, vol. 30, no. 3, pp. 1057-1066, Jun. 2015.
- [32] A. Haddadi, J. Mahseredjian and C. Dufour, "Time Warping method for accelerated EMT simulations," 2016 Power Systems Computation Conference (PSCC), pp. 1-7, Genoa, Italy, 2016.

- [33] F. Plumier, e. Geuzaine, and T. Van Cutsem, "On the convergence of relaxation schemes to couple phasor-mode and electromagnetic transients simulations," in Proc. of IEEE PES General Meeting, 2014.
- [34] F. Plumier, P. Aristidou, E. Geuzaine, and T. Van Cutsem, "A relaxation scheme to combine phasor-mode and electromagnetic transients simulations," in Proceedings of the J 8th Power System Computation Conference, Wroclaw, Poland, Aug. 2014.
- [35] D. M. Falcao, E. Kaszkurewicz, and H. L. S. Almeida, "Application of parallel processing techniques to the simulation of power system electromagnetic transients, " *IEEE Trans. Power Syst.*, vol. 8, no. 1, pp. 90–96, Feb. 1993.
- [36] C. Yue, X. Zhou, and R. Li, "Node-splitting approach used for network partition and parallel processing in electromagnetic transient simulation," in Proc. *Int. Conf. Power Syst. Technology*, Singapore, pp. 425–430, Nov. 2004.
- [37] M. Armstrong, J. R. Marti, L. R. Linares, and P. Kundur, "Multilevel MATE for efficient simultaneous solution of control systems and nonlinearities in the OVNI simulator," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1250–1259, Aug. 2006.
- [38] J. Mahseredjian: Compensation, Diakoptics, MATE and Bordered-Block-Diagonal methods for network solution parallelization, Polytechnique Montreal, March 18, 2019.
- [39] A. Abusalah, Accelerated Simulation of Large-Scale Power System Transients, these de doctorat, École Polytechnique de Montréal, 2019.
- [40] A. Abusalah, O. Saad, J. Mahseredjian, U. Karaagac, I. Kocar, "Computation of Electromagnetic Transients Through Parallel Solution of Sparse Matrices," *IEEE Power and Energy Technology Systems Journal*, Vol. 7, pp. 13-21, 2019.
- [41] A. Semlyen and F. De Leon, "Computation of electro-magnetic transients using dual or multiple time steps," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1274–1281, Aug. 1993.
- [42] P. Zhang, J. R. Marti, and H. W. Domme, "Shifted-frequency analysis for EMTP simulation of power-system dynamics," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 57, no. 9, pp. 2564–2574, Sep. 2010.
- [43] S. Henschel, "Analysis of Electromagnetic and Electromechanical Power System Transient with Dynamic phasors, " Ph.D. dissertation, University of British Columbia, 1999.
- [44] T. H. Demiray, "Simulation of Power System Dynamics using Dynamic Phasor Models," Ph.D. dissertation, ETH, Zurich, Switzerland, 2008.
- [45] L. R. Petzold, "A Description of DASSL: A Differential/Algebraic System Solver," presented at the IMACS World Congress, Montreal, August 8-13, 1982.
- [46] A. Semlyen, F. de Leon, "Computation Of Electro-Magnetic Transients Using Dual Or Multiple Time Steps," *IEEE Transactions on power Systems*, vol. 8, no. 3, pp. 1274-1281, August 1993.
- [47] J. Park, S. Jung and J.-I. Ha, "Variable Time Step Control for Six-Step Operation in Surface-Mounted Permanent Magnet Machine Drives," *IEEE Transactions On Power Electronics*, vol. 33, no. 2, pp. 1501-1513, Feb. 2018.
- [48] S. Jung, J. Park, E. Chung and J.-I. Ha, "Variable time step control with synchronous PWM in low frequency modulation index for AC machine drive," presented at the IEEE Energy Conversion Congress and Exposition (ECCE), Milwaukee, USA, 2016.
- [49] L. W. Nagel, SPICE2: A Computer Program to Simulate Semiconductor Circuits Memorandum No. UCB/ERL M520, May 9, 1975.
- [50] P. Kuffel, K. Kent and G. D. Irwin, "The Implementation and Effectiveness of Linear Interpolation Within Digital Simulation," in *International Conference on Power System Transients*, Lisbon, 1995.

- [51] G. Sybille, H. Le-Huy, R. Gagnon and P. Brunelle, "Analysis and Implementation of an Interpolation Algorithm for Fixed Time-Step Digital Simulation of PWM Converters," in *IEEE International Symposium on Industrial Electronics*, Vigo, 2007
- [52] M. Zou, J. Mahseredjian, G. Joos, B. Delourme and L. Gérin-Lajoie, "Interpolation and reinitialization in time-domain simulation of power electronic circuits," *Electric Power Systems Research*, vol. 76, no. 8, pp. 688-694, 2006.
- [53] B. De Kelper, L.-A. Dessaint, V.-Q. Do and S. J. -C, "An algorithm for accurate switching representation in fixed-step simulation of power electronics," in *IEEE Power Engineering Society Winter Meeting*, Singapore, 2000.
- [54] O. M. Faruque, V. Dinavahi and W. Xu, "Algorithms for the accounting of multiple switching events in digital simulation of power-electronic systems," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1157-1167, 2005.
- [55] P. Li, Z. Meng, X. Fu, H. Yu, C Wang, "Interpolation for power electronic circuit simulation revisited with matrix exponential and dense outputs," *Electric Power Systems Research*, vol 189, 2020.
- [56] J. Tant and J. Driesen, "On the Numerical Accuracy of Electromagnetic Transient Simulation With Power Electronics," *IEEE Transactions on Power Delivery*, vol. 33, no. 5, pp. 2492-2501, 2018.
- [57] B. De Kelper, L. A. Dessaint, K. Al-Haddad and H. Nakra, "A comprehensive approach to fixed-step simulation of switched circuits," in *IEEE Transactions on Power Electronics*, vol. 17, no. 2, pp. 216-224, March 2002.
- [58] G. Irwin, D. Woodford and A. Gole, "Precision simulation of PWM controller," in *International Conference on Power System Transients*, Rio de Janeiro, 2001.
- [59] H. Peng, M. Liu, Q. Mu, L. Sun and X. Zhang, "Interpolation algorithm considering simultaneous solution and instantaneous solution for power electronics electromagnetic transient simulation," in *The Institution of Engineering and Technology*, 2018.
- [60] K. L. Lian, and P. W. Lehn, "Real-time simulation of voltage source converters based on time average method," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 110–118 Feb. 2005.
- [61] J. Allmeling and N. Felderer, "Sub-cycle average models with integrated diodes for real-time simulation of power converters," in *Proc. IEEE SPEC*, pp. 1–6, Dec. 2017.
- [62] S. Horiuchi, K. Sano and T. Noda, "An Inverter Model Simulating Accurate Harmonics With Low Computational Burden for Electromagnetic Transient Simulations," in *IEEE Transactions on Power Electronics*, vol. 36, no. 5, pp. 5389-5397, May 2021.
- [63] P. Pejovic and D. Maksimovic, "A method for fast time-domain simulation of networks with switches," in *IEEE Transactions on Power Electronics*, vol. 9, no. 4, pp. 449-456, July 1994.
- [64] R. Reza, F. Chrysanthi, M. Paolone and R. Farhad, "A novel method for the optimal parameter selection of discrete-time switch model," Proceedings of the 10th International Conference on Power Systems Transients, 2013.
- [65] P. Deuflhard, *Newton Methods for Nonlinear Problems*. Springer, New York, 2004.
- [66] J. Mahseredjian, "Nonlinear devices, ", École Polytechnique de Montréal, 2006.
- [67] T. Noda and T. Kikuma, "A Robust and Efficient Iterative Scheme for the EMT Simulations of Nonlinear Circuits," in *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1030-1038, April 2011.
- [68] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a short survey of recent results, " *proceedings of the 2005 IEEE International Symposium on Intelligent Control Limassol*, Cyprus, June 27-29, 2005.

- [69] E. Gad, M. Nakhla, R. Achar and Y. Zhou, "A-Stable and L-Stable High-Order Integration Methods for Solving Stiff Differential Equations," *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, vol. 28, no. 9, pp 1359-1372, Sept. 2009.
- [70] E. V. Solodovnik, G. J. Cokkinides and A. P. S. Meliopoulos, "On stability of implicit numerical methods in nonlinear dynamical systems simulation," presented at the Proceedings of Thirtieth Southeastern Symposium on System Theory, Morgantown, WV, USA, pp. 27-31, 1998.
- [71] G. Dahlquist, "A special stability problem for linear multistep methods," *BIT*, vol. 3, no. 1, pp. 27–43, Mar. 1963.
- [72] E. Hairer, G. Wanner, "Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems," *Springer Heidelberg Dordrecht London New York*, Second revised edition, 2010
- [73] T. Noda, K. Takenaka and T. Inoue, "Numerical Integration by the 2-Stage Diagonally Implicit Runge-Kutta Method for Electromagnetic Transient Simulations," *IEEE Transactions on Power Delivery*, vol. 24, 2009.
- [74] F. L. Alvarado, R. H. Lasseter and J. J. Sanchez, "Testing of Trapezoidal Integration With Damping for the Solution of Power Transient Problems," *IEEE Power Engineering Review*, vol. 3, no. 12, pp. 31-32, 1983.
- [75] A. Roger, "Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 1006-1021, 1977.
- [76] J. Liu, T. Wei, L. Jiayu, Z. Wei, J. Hou and Z. Xiang, "Suppression of numerical oscillations in power system electromagnetic transient simulation via 2S-DIRK method," in *IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Xi'an, 2016.
- [77] X. Fu, S. M. Seye, J. Mahseredjian, M. Cai and C. Dufour, "A Comparison of Numerical Integration methods and Discontinuity Treatment for EMT simulations," presented at the *Power Systems Computation Conference (PSCC)*, Dublin, Ireland, pp. 11-15, June 2018.
- [78] S. M. Seye, "Plateforme de tests pour les méthodes d'intégration numériques appliquées aux réseaux électriques," *Mémoire de maîtrise*, Génie Electrique, Polytechnique Montréal, 2018.
- [79] M. Hosea and L. Shampine, "Analysis and implementation of TR-BDF2," *Applied Numerical Mathematics*, vol. 20, no. 1-2, pp. 21-37, 1996.
- [80] S. Dharmaraja, "An Analysis of the TR-BDF2 integration scheme," *Master of Science in Computation for Design and Optimization*, Massachusetts Institute of Technology, 2007.
- [81] J. Marti and J. Lin, "Suppression of numerical oscillations in the EMTP," *IEEE Transactions on Power Systems*, vol. 4, no. 2, pp. 739-746, 1989.
- [82] F. H. Branin: "Computer methods of network analysis", Proc. *IEEE*, Vol. 55, pp. 1787-1801, Nov 1967.
- [83] H. W. Dommel, "EMTP Theory Book," Microtran Power System Analysis Corporation, Vancouver, British Columbia, 1996.
- [84] K. W. Louie, A. Wang, P. Wilson and P. Buchanan, "Discussion on the initialization of the EMTP-type programs," *Canadian Conference on Electrical and Computer Engineering*, pp. 1962-1965, Saskatoon, SK, Canada, 2005.
- [85] B. Kulicke, "Simulationsprogramm NETOMAC: Differenzleitwertverfahren bei kontinuierlichen und diskontinuierlichen Systemen (Simulation program NETOMAC: Difference conductance method for continuous and discontinuous systems)," *Siemens Forshungs- und Entwicklungsberichte*, vol. 10, no. 5, pp. 299-302, 1981.

- [86] O. Tremblay, "Precise Algorithm for Nonlinear Elements in Large-Scale Real-Time Simulator," Conference: 2012 CIGRÉ Canada Conference, Montréal, 2012.
- [87] W. Nzale, J. Mahseredjian, I. Kocar, X. Fu and C. Dufour, "Two Variable Time-Step Algorithms for Simulation of Transients," *IEEE Milan PowerTech*, pp. 1-6, Milan, Italy, 2019.
- [88] V. Litovski, M. Savic and Z. Mrčarica, "Ideal switch model cuts simulation time," *IEEE circuits and devices magazine*, 2006.
- [89] A. M. Gole, I. T. Fernando, G. D. Irwin and O. B. Nayack, "Modeling of Power Apparatus: Additional Interpolation Issues," in *International Conference on Power Systems Transients*, pp. 23-28, Seattle, 1997.
- [90] W. Nzale, J. Mahseredjian, X. Fu, I. Kocar and C. Dufour, "Improving numerical accuracy in time-domain simulation for power electronics circuits," in *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 157-165, 2021.
- [91] 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," in *International Conference on Power Systems Transients*, Rio de Janeiro, 2001.
- [92] N. Watson, J. Arrillaga, Power system electromagnetic transients simulation, Institute of Elect. Eng., London, UK, 2003.
- [93] R. E. Bank, W. M. Coughran, W. Fichtner, E. H. Grosse, D. J. Rose and R. K. Smith, "Transient simulation of silicon devices and circuits," *IEEE Transactions on Electron Devices*, vol. 32, no. 10, pp. 1992 - 2007, 1985.
- [94] K. Strunz, "Flexible numerical integration for efficient representation of switching in real time electromagnetic transients simulation," *IEEE Transactions on Power Delivery*, vol. 19, no. 3, July 2004.
- [95] C. Dufour and J. Belanger, "Discrete time compensation of switching events for accurate real-time simulation of power systems," *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society*, vol.2, pp. 1533-1538, Denver, CO, USA, 2001.
- [96] W. Nzale, J. Mahseredjian, X. Fue, I. Kocar and C. Dufour, "Accurate time-domain simulation of power electronic circuits," *Electric Power Systems Research*, vol. 195, 2021.
- [97] E. S. Kuh and I. N. Hajj, "Nonlinear circuit theory: Resistive networks," in Proceedings of the *IEEE*, vol. 59, no. 3, pp. 340-355, March 1971
- [98] P. Deuflhard, Newton Methods for Nonlinear Problems. Springer, New York, 2004.
- [99] L. V. Kantorovich, "On Newton's method for functional equations," *Doklady Akademii Nauk SSSR*, vol. 59, pp. 1237–1240, 1948.
- [100] J. Katzenelson, "An algorithm for solving nonlinear resistor networks," *Bell Syst. Tech. J.*, pp. 1605–1620, Oct. 1965.
- [101] T. Fujisawa and E. S. Kuh, "Piecewise-linear theory of nonlinear networks," *SIAM J. Appl. Math.*, vol. 22, no. 2, pp. 307–328, Mar. 1972.
- [102] T. Ohtsuki, T. Fujisawa, and S. Kumagai, "Existence theorems and a solution algorithm for piecewise-linear resistor networks," *SIAM J. Math. Anal.*, vol. 8, no. 1, pp. 69–99, Feb. 1977.
- [103] G. A. Wempner, "Discrete approximations related to nonlinear theories of solids," *International Journal of Solids and Structures*, vol. 7, no. 11, pp. 1581-1599, 1971.
- [104] M. A. Crisfield, "A fast incremental/iterative solution procedure that handles snap-through", *Computers and structures*, vol. 13, no. 1-3, pp. 55-62, 1981.

- [105] B. Guenter and R. Parent, "Computing the arc length of parametric curves," in *IEEE Computer Graphics and Applications*, vol. 10, no. 3, pp. 72-78, May 1990.
- [106] B.-A. Memon and X. SU, "Arc-length technique for nonlinear finite element analysis", Journal of Zhejiang University SCIENCE, vol. 5, no. 5, pp. 618-628, 2004.
- [107] W. Nzale, J. Mahseredjian, X. Fu, I. Kocar and C. Dufour, "Adapted Arc Length Method for Time domain Simulation of Nonlinear Circuits, " to be submitted.

ANNEXES

Annexe A – NOMENCLATURE ET DESCRIPTION DES MÉTHODES ET TECHNIQUES DE SIMULATION DISCUSÉES

TR_BE Méthode d'intégration trapézoïdale qui, à la suite d'une discontinuité, utilise deux intégrations Backward Euler avec un pas d'intégration réduit de moitié.

TR_BE_I Méthode d'intégration TR_BE qui applique l'interpolation linéaire pour la détection précise d'une discontinuité juste avant d'exécuter les deux intégrations Backward Euler.

TR_BE_I_SS_m Méthode d'intégration trapézoïdale qui, à la suite d'une discontinuité, exécute l'interpolation linéaire comme TR_BE_I, puis applique la technique *simultaneous switching*. Ensuite, elle exécute deux intégrations Backward Euler. C'est la solution obtenue par interpolation au point de discontinuité (solution avant la discontinuité) qui est utilisée pour calculer les termes historiques de la première intégration Backward Euler (d'où la présence de la lettre *m* pour « moins » dans le nom de la méthode). C'est une méthode qui est proposée dans cette thèse.

TR_BE_I_SS_p Méthode d'intégration trapézoïdale qui, à la suite d'une discontinuité, exécute l'interpolation linéaire comme TR_BE_I, puis applique la technique *simultaneous switching*. Ensuite, elle exécute deux intégrations Backward Euler. C'est la solution obtenue après l'application de la technique *simultaneous switching* au point de discontinuité (solution après la discontinuité) qui est utilisée pour calculer les termes historiques de la première intégration Backward Euler (d'où la présence de la lettre *p* pour « plus » dans le nom de la méthode).

TR_BE_SS Méthode d'intégration TR_BE qui exécute la technique *simultaneous switching* lorsqu'une discontinuité est détectée, avant d'exécuter les deux intégrations Backward Euler.

TR_BE_SS_I Méthode TR_BE_SS qui applique l'interpolation juste après avoir exécuté la technique *simultaneous switching*. C'est une méthode qui est proposée dans cette thèse.

TR_DI Méthode TR qui interpole pour détecter les discontinuités, exécute une intégration TR et interpole de nouveau pour resynchroniser sur le time-mesh.

TR_DICR Méthode TR_DI qui utilise *Chatter Removal Technique*. Ainsi, à la suite de la resynchronisation, la méthode TR_DICR exécute une intégration TR et remplace la solution

obtenue par le résultat d'une interpolation linéaire à mi-chemin entre les 2 dernières solutions calculées.

TR_I_SS_m_BE Méthode TR qui applique l'interpolation lorsqu'une discontinuité est détectée, puis exécute la technique SS, ensuite fait deux intégrations BE pour finalement extrapoler la solution au point de discontinuité. C'est une méthode qui est proposée dans cette thèse.

Annexe B – DÉTERMINATION D'UNE VALEUR INCONNUE à PARTIR D'UNE TABLE DE VALEURS

Lorsque la relation de dépendance entre deux variables est exprimée sous la forme d'une table de valeurs, il est toujours possible de déterminer la valeur prise par l'une des variables lorsque l'autre variable est connue. Pour illustrer comment, on va considérer le Tableau 6.1 qui exprime un exemple de relation de dépendance, notamment entre la tension v_ϕ et le courant i_ϕ d'un composant non-linéaire. Évidemment, ce tableau est construit à partir de l'équation analytique (6.4) mais nous allons supposer que cette équation n'est pas connue. La seule information disponible est ce tableau (que l'on appelle aussi table des valeurs). Supposons que l'on souhaite déterminer la valeur v_ϕ de la tension lorsque le courant est égal par exemple à 10.1 A.

Dans un premier temps, on peut remarquer que $10.1 \in [2.306, 15.779]$. En d'autres termes, la valeur du courant connue se trouve entre les points 5 et 6 du Tableau 6.1. De ce fait, pour déterminer la valeur de la tension lorsque le courant vaut 10.1 A, il suffit de déterminer l'équation de la fonction affine qui lie les points 5 et 6 et qui exprime le courant en fonction de la tension, puis utiliser cette équation pour calculer la tension lorsque le courant est égal à 10.1 A.

La dépendance entre la tension et le courant entre les points 5 et 6 peut s'écrire sous la forme $i_5^5 = y_5 v_5 + i_5$. C'est l'équation (6.1). Puisque les points 5 et 6 correspondent aux paires (v, i)

données par : $(0.8, 2.306)$ et $(0.85, 15.779)$, alors on a : $y_5 = \frac{15.779 - 2.306}{0.85 - 0.8} = 269.54$ et

$i_5 = 15.779 - y_5 \times 0.85 = -213.33$. Ainsi, on peut écrire que $10.1 = 269.54 v_\phi - 213.33$, ce qui

permet d'obtenir $v_\phi = \frac{10.1 + 213.33}{269.54} = 0.829 \text{ V}$.

En procédant de la même manière, on peut déterminer n'importe quelle tension, connaissant le courant correspondant, et inversement. Il est à noter que si la valeur connue est inférieure à la première valeur (ou supérieure à la dernière valeur) de la table des valeurs, alors l'équation de la fonction affine liant les deux premières points (ou reliant les deux derniers points) est celle qui sera considérer pour trouver la valeur inconnue.

Une table de valeurs peut être construite à partir d'une équation analytique en apparence complexe afin de calculer facilement une variable, connaissant l'autre. Ceci permet d'éviter des

manipulations mathématiques sur l'équation analytique. Par exemple, une table de valeurs peut être construite pour représenter la relation (6.14). En prenant un échantillon de plusieurs valeurs de x régulièrement espacées (par exemple 1000 valeurs équidistances entre 0 et 1), on peut calculer la valeur correspondante de ρ pour chaque x et construire une table de valeurs. Ainsi, on pourra dès lors déterminer aisément x connaissant ρ (par la démarche présentée ci-dessus) sans avoir besoin d'exprimer analytiquement x en fonction de ρ à partir de (6.14).

Pour terminer, il est également possible à partir d'une table de valeurs, de déterminer la valeur de la dérivée d'une variable selon l'autre variable à un point donné. Par exemple, avec la table de dépendance du Tableau 6.1, si on a besoin d'évaluer $\frac{di}{dv}$ au point de valeur de courant 10.1, on peut

simplement remarquer que ce point est situé entre les points 5 et 6 de la table des valeurs et calculer

$$\text{la dérivée comme suit : } \left. \frac{di}{dv} \right|_{i=10.1} = \frac{15.779 - 2.306}{0.85 - 0.8} = 269.46.$$

Annexe C - CALCUL DE LA LONGUEUR D'ARC POUR LA MÉTHODE ARC LENGTH DANS LE CAS D'UNE REPRÉSENTATION LINÉAIRE PAR SEGMENTS

La longueur de l'arc de la caractéristique d'un composant non-linéaire à une position donnée représente la distance algébrique entre l'origine et la position en question, comme le traduit l'équation (6.13) qui est applicable quel que soit le type d'équation liant la tension et le courant. Dans le cas particulier d'une dépendance linéaire par segments, il est possible de calculer cette longueur différemment. Pour illustrer, on va considérer la représentation non-linéaire du Tableau 6.1 dans lequel on souhaite connaître la valeur de la longueur d'arc associée à chaque point.

Pour cela, il suffit de remarquer que, dans une représentation linéaire par segments, deux points sont reliés par un segment de droite dont la longueur peut se déterminer aisément en utilisant la propriété de Pythagore. Ainsi, la longueur du segment compris entre les points 1 et 2 est égale à $\sqrt{(-50+10^5)^2 + (-10^{-8}+10^{-7})^2} = 99950$. De même, la longueur du segment compris entre les points 2 et 3 est égale à $\sqrt{(0+50)^2 + (0+10^{-8})^2} = 50$. On peut par le même calcul, évaluer toutes les longueurs. Puisque la longueur de l'arc représente la longueur entre l'origine (le point 3) et chacun des autres points, alors la longueur d'arc correspondant au point 1 sera simplement la somme des longueurs des segments allant de 3 à 2 puis de 2 à 1, ce qui permet d'écrire $99950 + 50 = 100000$. Puisque ces segments sont dans le côté négatif des tensions, les longueurs d'arc associées seront comptées négativement. En répétant cette démarche de calcul à l'ensemble des points de la caractéristique, on obtient le tableau suivant, donnant les valeurs des longueurs d'arc selon les tensions :

Point	1	2	3	4	5	6	7	8
$v_\phi(V)$	-10^5	-50	0	0.7	0.8	0.85	0.8999	0.9499
ρ	-10^{-5}	-50	0	0.70	2.96	16.43	108.20	736.49