



**HAL**  
open science

# CONTRIBUTION A UNE METHODOLOGIE DE DIMENSIONNEMENT DES CONVERTISSEURS STATIQUES

Stéphane Bergeon

► **To cite this version:**

Stéphane Bergeon. CONTRIBUTION A UNE METHODOLOGIE DE DIMENSIONNEMENT DES CONVERTISSEURS STATIQUES. Energie électrique. Institut National Polytechnique de Grenoble - INPG, 1998. Français. NNT: . tel-00819760

**HAL Id: tel-00819760**

**<https://theses.hal.science/tel-00819760>**

Submitted on 2 May 2013

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

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

THESE  
présentée par

Stéphane BERGEON  
Maître ès Sciences Physiques  
de l'Université F. Rabelais de Tours

pour obtenir le grade de DOCTEUR  
de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE  
(Arrêté ministériel du 30 mars 1992)

Spécialité Génie Electrique

---

---

**CONTRIBUTION  
A UNE METHODOLOGIE DE DIMENSIONNEMENT  
DES CONVERTISSEURS STATIQUES**

---

---

Le 6 janvier 1998

Composition du jury :

Messieurs :	C. ROMBAUT	Président et Rapporteur
	F. FOREST	Rapporteur
	R. MILLY	
	J.-P. FERRIEUX	
	L. GERBAUD	
	J. BIGEON	

Thèse préparée au sein du Laboratoire d'Electrotechnique de Grenoble



*A Julie*



Je remercie les membres du jury :

Monsieur Christian Rombaut, professeur à l'Ecole Centrale de Lille, qui m'a fait l'honneur de présider le comité ;

Monsieur François Forest, professeur à l'Ecole Nationale Supérieure de Cachan, qui a accepté d'être rapporteur de cette thèse ;

Monsieur Roger Milly, directeur de la société Enertronic, pour l'intérêt qu'il porte d'emblée aux travaux de recherche initiés dans cette thèse ;

Monsieur Jean-Paul Ferrieux, professeur à l'Ecole Nationale Supérieure d'Ingénierie en Electrotechnique de Grenoble, pour les intéressantes discussions sur le problème de la conception des convertisseurs statiques.

Je tiens particulièrement à remercier :

Monsieur Jean Bigeon, directeur de recherche au CNRS, et directeur de l'équipe Conception et Diagnostic Intégrés du LEG. Parmi ses nombreuses qualités scientifiques communément admises, j'évoquerai seulement ici son esprit de synthèse suffisamment pointu pour tirer un portrait flatteur en trois mots. Par contre je m'attarderai davantage sur ses qualités humaines peut-être moins connues, dans le désordre : proximité, écoute, confiance, franchise, et exigence. Je le remercie de m'avoir accueilli dans son équipe, et guidé pendant ces années.

Monsieur Laurent Gerbaud, chargé de recherche au CNRS. Si je devais m'essayer au périlleux exercice de synthèse pour décrire Laurent, j'utiliserais certainement les termes optimisme, enthousiasme, rigueur, entrain, et disponibilité. Bien plus qu'un encadrement, c'est un soutien sans faille que j'ai trouvé auprès de lui. Sources de motivation, les nombreuses discussions que nous avons partagées ont permis chacune de passer une nouvelle étape dans ma thèse.

Dans ce générique, je ne veux pas oublier toutes les personnes qui ont eu un rôle important dans cette aventure de trois ans :

Monsieur Jean-Louis Coulomb qui m'a fait confiance et m'a ouvert les portes du LEG en m'accueillant en DEA.

James Roudet, qui a toujours montré un grand intérêt pour mes travaux, et m'a fait l'honneur de co-signer une publication.

Frédéric Wurtz. Chaque entretien scientifique entrepris avec lui (ce n'ai jamais une entreprise de tout repos) enrichit forcément votre propre réflexion. Véritable catalyseur d'idées, et modèle d'encadrement (il a été mon mentor pendant mon DEA), souhaitons que Frédéric puisse renforcer les rangs du CNRS dès cette année ;

Les administrateurs réseaux pour leur disponibilité : Marie-Thérèse Loubinoux, Etienne Callegher, Bruno Ferrari, Patrick Guillot, avec une mention spéciale à Florence François qui, avec sa gentillesse légendaire, a toujours résolu mes ennuis informatiques dans la bonne humeur.

Les chercheurs du Laboratoire de Modélisation et de Calcul : Gilles Villard, Moulay Barkatou, et David Lanier. J'espère que la collaboration que nous avons initiée pendant cette thèse a été aussi fructueuse pour eux qu'elle le fut pour moi.

Christophe Lechevalier sans peur et sans reproche. Derrière la tranquille bonhomie du *Tof* se cachent un grand maître du punch, un wonder-DJ et un as de l'informatique. Jamais pris au dépourvu, il sort toujours l'idée ou la formule qui vous cueille. Alors, affûtez vos arguments lorsque vous engagez une polémique avec lui, et puis après, invitez-le à la cafète pour une pause.

Patrick Podczaski dont je me fais un plaisir d'écrire le nom (l'occasion de placer un *c* entre un *d* et un *z* sans faire une coquille est tellement rare). Il est d'ores et déjà prêt pour le vingt et unième siècle.

Les membres de la dynamique équipe CDI que je n'ai pas encore cités : Alain Bolopion, Basma Belhabib, Christophe Sauvey, Coralie Coutel, Jaime Fandino, et Jean-Marc Bouché dont les mémorables représentations improvisées ont plusieurs fois déclenché le fou rire sur la plateforme.

Eric Escande pour la confiance, la franchise et la convivialité qui ont animées nos nombreuses discussions.

Patrick Eustache et Pascal Pétin avec qui j'eus le privilège de former le trio des *SIC-users* à la grande époque...

Claire Divoux, autrement dit *la femme du tonnerre* ou *l'Anapurna sans bouteilles*. Clairette la starlette est un formidable remède contre la neurasthénie.

Les thésards du LEG avec qui j'ai eu le plaisir de partager déjeuners au ru, pauses café, parties de tennis, cinéma, soirées, et/ou polémiques scientifiques : Alain Groud, Ambroise Schellmanns, Charlotte Gillot, Djelloul Moussaoui, Fabrice Le Dorze prince du Mail, Gwénaelle Le Coat, Hassan Benqassmi, Jean Schutz, Jean-Christophe Crebier, Joël Conrad, Nadège Piette, Nicolas Retière, Rachid Laouamer, Raphaël Peugeot, Réza Zolghadri, Séverine Guffon, Son Pete S. Duong...

Mes charmantes colocataires de bureau dans leur ordre d'apparition : Edith Clavel puis Rafika Berrouche.

Toutes les personnes qui m'ont soutenu pendant ces trois années à Grenoble : Ando, Caroline, Estelle, Jean-Paul, Jérôme, Lydia, Marie, Mehdi, Pierre, Rémy.Blache, Sandra, et Sonia<sup>2</sup>... ; mes cyber-cousins : Cécile, Claire, Isabelle et Jean.

Tous ceux qui ne sont pas cités ici par inadvertance et que je n'ometts pas à dessein (ils se reconnaîtront).

Ma princesse préférée.

Et ma famille.

Et aussi un merci spécial à Pégase et à Péricles.

# TABLE DES MATIERES

<b>TABLE DES MATIÈRES</b>	<b>i</b>
<b>GLOSSAIRE</b>	<b>vii</b>
<b>INTRODUCTION</b>	<b>1</b>
<b>CHAPITRE I : LA PROBLÉMATIQUE DE CONCEPTION DES CONVERTISSEURS STATIQUES</b>	<b>3</b>
<b>1 CADRE DE L'ÉTUDE</b>	<b>3</b>
1.1 GÉNÉRALITÉS SUR LES CONVERTISSEURS STATIQUES	3
1.2 LES DIFFÉRENTES PHASES DE LA CONCEPTION DES CONVERTISSEURS STATIQUES	4
1.2.1 Les différentes phases de conception des convertisseurs statiques	5
1.2.2 L'interaction entre les phases de conception	5
<b>2 DÉFINITION D'UN OUTIL D'ANALYSE EN VUE DU DIMENSIONNEMENT</b>	<b>6</b>
2.1 GÉNÉRICITÉ	6
2.1.1 Un problème méthodologique	6
2.1.2 Un problème de modélisation	6
2.2 EVOLUTIVITÉ	7
2.3 INTÉGRATION	7
2.4 RAPIDITÉ	7
2.5 FINESSE	8
2.6 PRISE EN CHARGE DES IMPERFECTIONS DES COMPOSANTS :	8
<b>3 PRINCIPE DE CONCEPTION : REFORMULATION DU PROBLÈME DE DIMENSIONNEMENT EN UN PROBLÈME D'OPTIMISATION CONTRAINTE</b>	<b>8</b>
3.1 PROBLÈME DIRECT (OU PROBLÈME D'ANALYSE)	8
3.1.1 Définition	8
3.1.2 Méthodes	9
3.2 PROBLÈME INVERSE	10
3.3 PROBLÈME INVERSE RELAXÉ	11
3.4 PROBLÈME INVERSE SOUS CONTRAINTES	11
3.5 PROBLÈME D'OPTIMISATION CONTRAINTE	12



3.5.1	Définition	12
3.5.2	Méthodes de résolution d'un problème d'optimisation contraint	14
<b>4</b>	<b>POSITIONNEMENT DES MÉTHODES D'OPTIMISATION PAR RAPPORT À LA CONCEPTION DES CONVERTISSEURS DE PUISSANCE</b>	<b>17</b>
4.1	CONCEPTION DE STRUCTURES DES CONVERTISSEURS STATIQUES	17
4.2	DIMENSIONNEMENT DES COMPOSANTS DES CONVERTISSEURS STATIQUES	17
4.2.1	Traduction du problème de dimensionnement contraint comme un problème d'optimisation contrainte	17
4.2.2	Problèmes particuliers du dimensionnement des convertisseurs statiques	19
4.2.3	Méthodes utilisées pour le dimensionnement des convertisseurs statiques	19
<b>5</b>	<b>DÉMARCHE PROPOSÉE</b>	<b>22</b>
5.1	PRÉSENTATION	22
5.2	HYPOTHÈSES DE TRAVAIL	22
5.2.1	Dimensionnement continu	22
5.2.2	Mode de fonctionnement figé	23
5.2.3	Domaine temporel	23
5.2.4	Point de départ connu	23
5.2.5	Modèles simplifiés	23
5.3	APPORTS	24
<b>6</b>	<b>CONCLUSION</b>	<b>26</b>

## **CHAPITRE II : LA FORMULATION GÉNÉRIQUE DU MODÈLE DE DIMENSIONNEMENT** 27

<b>1</b>	<b>DÉFINITIONS</b>	<b>27</b>
1.1	FORMULATION DU MODÈLE DE DIMENSIONNEMENT	27
1.2	GÉNÉRICITÉ	27
<b>2</b>	<b>PROBLÈMES DE FORMULATION DU MODÈLE DE DIMENSIONNEMENT</b>	<b>29</b>
<b>3</b>	<b>LES CHOIX DE FORMULATION</b>	<b>30</b>
3.1	MODE DE FONCTIONNEMENT FIGÉ	30
3.2	AIDE À LA FORMULATION DES CRITÈRES GÉNÉRIQUES	30
3.3	NORMALISATION	31
<b>4</b>	<b>LES ÉLÉMENTS CONSTITUTIFS DU MODÈLE DE DIMENSIONNEMENT</b>	<b>31</b>
4.1	CLASSIFICATION DES DONNÉES	31
4.2	LES CRITÈRES	33
4.2.1	Critères propres au fonctionnement du convertisseur statique : les critères physiques	33
4.2.2	Critères de dimensionnement	41
4.3	LA FONCTION OBJECTIF	47
4.4	LES ENTRÉES DU DIMENSIONNEMENT	50

4.5	FORMULATION DES GRADIENTS	51
4.6	FORMULATION DU CAHIER DES CHARGES	52
4.6.1	Définition	52
4.6.2	Les différentes contraintes	52
4.6.3	L'affectation des contraintes	52
<b>5</b>	<b>CRITIQUE DU MODÈLE DE DIMENSIONNEMENT</b>	<b>53</b>
5.1	MODÈLE UTILISÉ POUR FORMULER LES CRITÈRES	53
5.1.1	Equations de base	53
5.1.2	Discussion sur la pertinence du modèle à topologie variable pour fonder un modèle de dimensionnement des convertisseurs statiques.	54
5.2	PROBLÈMES DE CONSTRUCTION DU MODÈLE DE DIMENSIONNEMENT	55
5.2.1	Problème de l'intégration symbolique des systèmes d'état	55
5.2.2	Problème de la traduction en un problème d'optimisation	59
5.3	LIMITES INTRINSÈQUES DU MODÈLE	59
<b>6</b>	<b>CONCLUSION</b>	<b>61</b>

---

### **CHAPITRE III : NOTRE MÉTHODOLOGIE DE DIMENSIONNEMENT DES CONVERTISSEURS STATIQUES**

---

<b>1</b>	<b>ENCHAÎNEMENT DES ÉTAPES (FIGURE III.1)</b>	<b>64</b>
<b>2</b>	<b>PRÉSENTATION DES 6 ÉTAPES</b>	<b>64</b>
2.1	ETAPE 1 : MODÉLISATION	64
2.1.1	Rôle	64
2.1.2	Entrées	65
2.1.3	Sorties	65
2.1.4	Méthode	66
2.2	ETAPE 2 : SIMULATION	66
2.2.1	Rôle	66
2.2.2	Entrées	66
2.2.3	Sorties	67
2.2.4	Méthode	67
2.3	ETAPE 3 : ANALYSE DE LA SIMULATION	69
2.3.1	Rôle	69
2.3.2	Entrées	70
2.3.3	Sorties	70
2.3.4	Méthode	70
2.4	ETAPE 4 : GÉNÉRATION DU MODÈLE DE DIMENSIONNEMENT	71
2.4.1	Rôle	71

2.4.2	Entrées	71
2.4.3	Sorties	71
2.4.4	Méthode	72
2.5	ETAPE 5 : TRADUCTION EN UN PROBLÈME D'OPTIMISATION	73
2.5.1	Rôle	73
2.5.2	Entrées	73
2.5.3	Méthode	74
2.6	ETAPE 6 : DIMENSIONNEMENT	76
<b>3</b>	<b>CONCLUSION</b>	<b>78</b>

---

**CHAPITRE IV : IMPLANTATIONS** **80**

<b>1</b>	<b>CHOIX TECHNIQUES GÉNÉRAUX</b>	<b>80</b>
1.1	OUTILS EXISTANTS UTILISÉS	80
1.1.1	GENTIANE	80
1.1.2	PASCOMA	81
1.1.3	Mathcad	84
1.2	CALCUL SYMBOLIQUE	84
1.2.1	Macsyma	84
1.2.2	Maple	86
1.3	MÉTHODES NUMÉRIQUES	88
<b>2</b>	<b>ETAPES DÉJÀ RÉALISÉES PAR AILLEURS</b>	<b>89</b>
<b>3</b>	<b>ETAPES EN COURS DE RÉALISATION</b>	<b>89</b>
3.1.1	Analyse de la simulation	89
3.1.2	Formulation du modèle de dimensionnement et traduction du problème d'optimisation	91
3.1.3	Intégration / Couplage informatique	101
<b>4</b>	<b>CONCLUSION</b>	<b>103</b>

---

**CHAPITRE V : APPLICATIONS** **103**

<b>1</b>	<b>INTRODUCTION</b>	<b>103</b>
<b>2</b>	<b>MISE EN GARDE</b>	<b>103</b>
<b>3</b>	<b>UNE SESSION UTILISATEUR, APPLICATION AU DIMENSIONNEMENT DU CONVERTISSEUR MULTIRÉSONANT EN THYRISTOR DUAL FORWARD</b>	<b>104</b>
3.1	PRÉSENTATION DU ZVS-MRC	104
3.2	UNE SESSION UTILISATEUR POUR DIMENSIONNER LE ZVS-MRC AVEC L'APPROCHE QUE NOUS AVONS MISE EN ŒUVRE.	106
3.2.1	MODELISATION	106

---

3.2.2	Génération de l'outil de dimensionnement dédié	120
3.2.3	Session de dimensionnement	121
<b>4</b>	<b>SECONDE APPLICATION : LE CAS D'ÉCOLE DU HACHEUR SÉRIE</b>	<b>132</b>
4.1	STRUCTURE ET MODE DE FONCTIONNEMENT	132
4.2	PROBLÈME SPÉCIFIQUE DE DIMENSIONNEMENT	133
4.2.1	Cahier des charges	133
4.2.2	Session de dimensionnement	134
<b>5</b>	<b>CONCLUSION</b>	<b>137</b>
<b><u>AVANTAGES ET INCONVÉNIENTS DE LA DÉMARCHE PROPOSÉE</u></b>		<b><u>139</u></b>
1	AVANTAGES	139
2	INCONVÉNIENTS	140
<b><u>CONCLUSION ET PERSPECTIVES</u></b>		<b><u>141</u></b>
<b><u>RÉFÉRENCES BIBLIOGRAPHIQUES</u></b>		<b><u>143</u></b>
<b><u>ANNEXE A LES RETRAITEMENTS SYMBOLIQUES SUR LES BLOCS-FONCTIONS</u></b>		<b><u>151</u></b>
<b><u>ANNEXE B FICHER MATHCAD D'ANALYSE DE LA SIMULATION GENTIANE-ORO</u></b>		<b><u>155</u></b>
<b><u>ANNEXE C FICHER DE DESCRIPTION DU MODÈLE DE DIMENSIONNEMENT DU ZVS-MRC (VERSION NON AUTOMATISÉE)</u></b>		<b><u>159</u></b>
<b><u>ANNEXE D TABLEAU DE CORRESPONDANCE ENTRE LES PARAMÈTRES PASCOSMA ET CEUX PRÉSENTÉS AU CHAPITRE II POUR L'APPLICATION DU ZVS-MRC</u></b>		<b><u>166</u></b>



## GLOSSAIRE

**Contrainte :** Une contrainte est un ensemble de valeurs que l'on autorise pour un paramètre : en continu il existe trois types de contraintes : les contraintes fixes à une valeur, les contraintes sur un intervalle de variation, et les contraintes ignorées, c'est-à-dire sur un intervalle infini.

**Critère :** Un critère permet d'exprimer une relation entre plusieurs paramètres de dimensionnement. Nous formulons symboliquement ces relations, en les stockant dans des variables qui deviennent donc des nouveaux paramètres du dimensionnement, pouvant alors servir pour formuler d'autres critères ... ce qui explique l'architecture imbriquée du modèle de dimensionnement.

$$C = \{\text{critères}\}$$

**Entrée :** Les entrées sont les paramètres du dimensionnement que l'on choisit indépendants pour pouvoir orienter le graphe de dépendance. Le choix de ces entrées est fait en fonction de leur accessibilité (peut-on obtenir facilement leur valeur par simulation par exemple ?), et en fonction de leur généricité (peut-on formuler symboliquement aisément tous les critères à partir de ces paramètres ?) Cette orientation arbitraire correspond à un choix de formulation du modèle de dimensionnement.

$$E' = \{\text{entrées primaires}\}$$

$$E'' = \{\text{entrées secondaires}\}$$

$$E''_{\text{auto}} = \{\text{entrées secondaires automatiques}\}$$

$$E''_{\text{uti}} = \{\text{entrées secondaires utilisateur}\}$$

$$E = \{\text{entrées}\}$$

**Générer :** Générer, ou engendrer un modèle, consiste à construire le modèle dédié à une application donnée en utilisant le moule d'un modèle générique.

**Générique :** Qui permet d'engendrer (de générer) un genre (une classe) de problèmes donné.

**Genre :** Un genre, ou une classe de problèmes, est un ensemble de problèmes définis par un sujet et un objectif communs. Une classe de problème peut être résolue par différentes méthodes, en utilisant différents modèles.

**GENTIANE :** Environnement de simulation des ensembles machine – convertisseur-statique – commande permettant notamment la simulation à topologie variable des convertisseurs statiques.

**Gradient :** Lorsque l'orientation du graphe de dépendance est faite, les entrées et critères sont définis, et on peut calculer les dérivées des critères en fonction des entrées : ce sont les gradients du modèle de dimensionnement.

$$G = \{\text{gradients}\}$$

$DP = \frac{\partial C}{\partial P_c}$  est l'ensemble des dérivées partielles.

**Macsyma :** Logiciel de calcul formel utilisé pour ses capacités de manipulation des expressions évoluées, et sa faculté de pouvoir traduire de façon optimisée ses expressions symboliques en Fortran.

**Maple :** Logiciel de calcul formel utilisé pour sa capacité de manipuler les rootofs

**Mathcad :** Logiciel interactif et convivial idéal pour développer des outils d'analyse nécessitant des visualisations graphiques, des programmations d'algorithmes itératifs, et des calculs symboliques simples.

**Modèle de dimensionnement :** Ensemble des équations de dimensionnement. Dans l'optique de sa future traduction en un problème d'optimisation contrainte, notre modèle de dimensionnement formulé symboliquement contient plus précisément : les expressions symboliques des critères, et des gradients permettant ainsi de générer le programme d'analyse et le programme de calcul de sensibilité tous deux nécessaires pour résoudre le problème d'optimisation dual.

**Paramètre :** Les paramètres du dimensionnement sont tous les paramètres apparaissant dans la formulation des critères.

$$P = \{\text{paramètres}\}$$

$P_C$ , l'ensemble des paramètres dont dépend directement le critère  $C$ ,

**Paramètre interne :** Les paramètres internes du modèle de dimensionnement sont les paramètres indispensables pour formuler le modèle, mais qui sont complètement transparents pour l'utilisateur. Ils ne sont ni initialisés, ni contraints.

**PASCOSMA :** Logiciel de génération automatique du programme d'optimisation dédié à un problème de dimensionnement dont les équations sont formulées symboliquement.

**Rootof :** Le mot *rootof* est issu de la terminologie de Maple. C'est le vecteur des racines d'un polynôme non factorisable sous forme radicale par les méthodes classiques. N'étant pas exprimables symboliquement en fonction des coefficients de ce polynôme, nous les évaluons numériquement.





# *Introduction*



# INTRODUCTION

Le concepteur de convertisseurs statiques est confronté dans son activité au problème de dimensionnement de composants. Cette tâche constitue une partie délicate du processus de conception, où à partir du cahier des charges, le concepteur n'opère en général que par l'analyse, par exemple avec des simulations numériques.

Dans ce cadre, différents outils et méthodes qui ne résolvent que ce problème d'analyse sont disponibles, et leur fiabilité est reconnue. Le travail de dimensionnement à proprement dit est donc en général laissé à la charge du concepteur.

Afin de l'aider à ce niveau, apparaît l'idée de considérer ces problèmes comme des problèmes d'optimisation. Dans ce cadre, des méthodes sont :

- soit développées autour des outils d'analyse (par exemple SABER, SIMPLORER) en proposant un algorithme général d'optimisation, mais souvent cela est lourd et difficile d'emploi, et donc difficilement intégrable dans une procédure générale de conception des convertisseurs statiques ;
- soit cela est fait au coup par coup en développant un outil dédié. Ceci requiert de l'analyse, c'est souvent efficace (temps de calcul, résultats) mais délicat à mettre en œuvre et non systématique.

L'approche que nous proposons tente de résoudre les problèmes d'intégration des outils généraux tout en permettant d'élaborer automatiquement des outils dédiés et en offrant une analyse de sensibilité due au choix de l'algorithme d'optimisation.

Après un positionnement du cadre de notre problématique de dimensionnement dans celle de conception, le chapitre I présente différentes façons d'aborder le problème d'analyse et de dimensionnement des convertisseurs statiques. En particulier, il précise différentes approches d'optimisation. A partir de là, il présente notre approche qui tente de résoudre le problème de dimensionnement en proposant une méthodologie qui permet au maximum, à partir de la description de la structure d'un convertisseur statique, d'obtenir un outil de dimensionnement dédié reposant sur un algorithme d'optimisation des gradients conjugués. Puis il se termine par les hypothèses et apports de cette approche.

Le chapitre II propose une formulation générique du modèle de dimensionnement à partir de la description de son circuit, en ne faisant intervenir l'expert que pour caractériser les composants

spécifiques (par exemple un modèle de dimensionnement d'un transformateur). Cette modélisation repose sur des manipulations de modèles exprimés sous forme d'équations. Le chapitre se termine en spécifiant les limites intrinsèques de l'approche, et en proposant certaines perspectives pour les traiter. Enfin, le choix étant fait de construire ces modèles dans un domaine continu de valeurs pour les dimensions des composants, la prise en compte de l'aspect discret du dimensionnement est laissée à la charge du concepteur, ou d'un futur autre acteur de la conception.

Le chapitre III propose une méthodologie de dimensionnement intégrant la modélisation évoquée au chapitre II, ainsi que des outils d'analyse, de calcul et de génération de modèles et de programmes. Cette méthodologie est fondée sur six étapes qui font intervenir des outils préexistants ou développés au chapitre IV.

Le chapitre IV expose l'implantation de la méthodologie décrite précédemment. Dans ce cadre, les techniques utilisées sont d'abord exposées. Puis sont présentés les outils qui ont été récupérés, ceux qui ont été complétés, et ceux qui ont été mis en œuvre ; ainsi que la façon de les faire cohabiter afin d'obtenir un environnement intégré de dimensionnement.

Le chapitre V présente deux exemples d'application de l'approche. L'objectif de ce chapitre est d'illustrer la démarche proposée, et en particulier la façon d'utiliser les outils. Seul l'aspect génération automatique du modèle de dimensionnement étant en cours de test actuellement n'est pas illustrée ici, mais les formulations correspondantes ont été testées au mieux "manuellement" (sans processus automatisé) dans ces exemples.

Enfin, les avantages et inconvénients de l'approche sont présentés avant de conclure et d'ouvrir sur différentes perspectives.

## *Chapitre I :*

# *La problématique de conception des convertisseurs statiques*



# CHAPITRE I : LA PROBLEMATIQUE DE CONCEPTION DES CONVERTISSEURS STATIQUES

## 1 Cadre de l'étude

### 1.1 Généralités sur les convertisseurs statiques

Un convertisseur de puissance réalise le transfert d'énergie entre deux sources d'énergie électrique. Convertir l'énergie électrique consiste à modifier les signaux de la tension et de l'intensité délivrés. Longtemps réalisée mécaniquement avec l'utilisation de pièces tournantes (convertisseurs dynamiques), la conversion d'énergie électrique, aussi bien pour les faibles que pour les fortes puissances, est aujourd'hui presque exclusivement réalisée par des convertisseurs statiques : la conversion ne passe pas par une phase de transformation de l'énergie électrique en énergie mécanique, mais procède par stockages et restitutions de l'énergie électrique, judicieusement gérés par le contrôle des commutations des semi-conducteurs de la structure du convertisseur. Un tel fonctionnement des convertisseurs statiques implique l'emploi de deux sortes de composants électriques :

- les composants passifs qui servent à emmagasiner et filtrer l'énergie électrique – condensateurs et bobines inductrices -, ou à la dissiper – résistances -, et
- les composants dynamiques qui permettent de réaliser les commutations : il s'agit des semi-conducteurs commandés tels que des thyristors ou des transistors, et les diodes qui sont des interrupteurs non commandés.

Différents types de conversion existent (cf. Tableau I.1) :

Source CONTINUE – Source CONTINUE
Source CONTINUE – Source ALTERNATIVE
Source ALTERNATIVE - Source CONTINUE
Source ALTERNATIVE – Source ALTERNATIVE

**Tableau I.1 : Les différents types de conversion**



Tous les convertisseurs statiques sont caractérisés par un fonctionnement discret qui peut être représenté par une succession de configurations temporaires du circuit provoquée par l'ouverture et la fermeture des semi-conducteurs. En fonctionnement normal, la succession de ces configurations suit un cycle périodique, et définit le mode de fonctionnement du convertisseur statique.

## 1.2 Les différentes phases de la conception des convertisseurs statiques

La conception d'un convertisseur statique pose trois problèmes liés (cf. Figure I.1) :

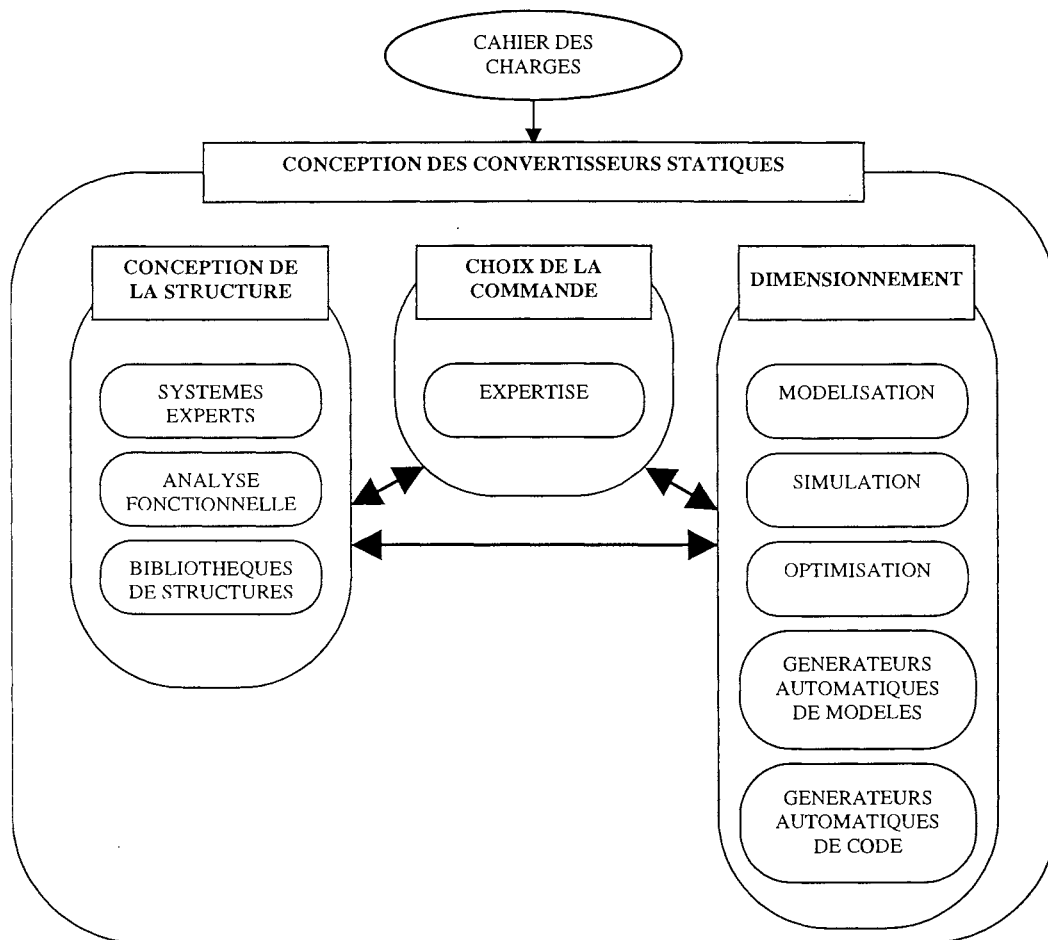


Figure I.1 : La vision modulaire d'une méthodologie globale de conception des convertisseurs statiques

## 1.2.1 Les différentes phases de conception des convertisseurs statiques

### 1.2.1.1 *Le choix de la structure*

A partir de la définition du cahier des charges, il est nécessaire de déterminer l'assemblage des composants qui constituent le circuit du convertisseur et qui permettent de réaliser la fonction de conversion demandée. La résolution de ce problème fait appel à l'expertise du concepteur qui utilise expérience et savoir-faire en procédant le plus souvent par similitude avec d'autres problèmes plus ou moins proches déjà étudiés ou aperçus dans la littérature.

### 1.2.1.2 *Le choix de la commande*

Afin de réaliser la conversion souhaitée, il faut faire fonctionner la structure dans un mode défini. Le problème consiste alors à déterminer l'ordre d'ouverture et de fermeture des interrupteurs du circuit pour mettre en place ce fonctionnement.

### 1.2.1.3 *Le dimensionnement et le choix des composants*

Le problème est maintenant de choisir les composants, (condensateurs, inductances, résistances, interrupteurs ...), et les ensembles de composants (transformateurs), qui vont former la structure finale du convertisseur. Ce dimensionnement obéit à des critères issus des besoins en termes de performances et de coût, et repose sur les caractéristiques de chaque composant.

## 1.2.2 L'interaction entre les phases de conception

Concevoir de façon globale un convertisseur statique ne consiste pas à résoudre séquentiellement chaque problème pris successivement. En effet, ces trois problèmes sont relativement imbriqués, ainsi par exemple le choix de la commande influe sur le mode de fonctionnement, et sur le type d'interrupteurs à utiliser dans la structure. Cependant, compte tenu de la difficulté à résoudre l'ensemble du problème de conception de façon globale, nous proposons de résoudre chacun de ces problèmes de façon indépendante, d'autant plus que nous verrons que leur résolution fait intervenir des techniques fort différentes. Nous envisageons de traiter l'ensemble du problème de conception des convertisseurs par une approche globale modulaire, où chaque module s'attaquerait à un des trois problèmes, et avec une gestion assistée par ordinateur des rebouclages nécessaires sur chaque module pour prendre en compte les interactions entre les problèmes. Aussi, dans cette optique, nos

travaux se sont intéressés exclusivement au problème du dimensionnement des convertisseurs statiques, le choix de la structure et de la commande pouvant alors être considéré comme figé.

## **2 Définition d'un outil d'analyse en vue du dimensionnement**

Nous allons tenter de dégager les points caractéristiques d'un outil efficace d'analyse en vue du dimensionnement des convertisseurs statiques : efficace afin d'être cohérent avec notre vision modulaire du problème de conception des convertisseurs statiques, et efficace dans le sens de l'aide apportée au concepteur.

Six points nous semblent importants à prendre en compte pour le développement d'un tel outil :

### **2.1 Généricité**

On entend par généricité la capacité d'adaptation de l'outil en termes de structures des convertisseurs à étudier, et en termes de requêtes de l'utilisateur / concepteur.

Proposer une manière générique de dimensionner les convertisseurs statiques pose un double problème :

#### **2.1.1 Un problème méthodologique**

Il faut être apte à proposer une méthode (une méthodologie, une approche) applicable pour le dimensionnement de toute structure de convertisseur, ou du moins d'un certain nombre de types donnés.

#### **2.1.2 Un problème de modélisation**

On doit avoir le souci de proposer un modèle de dimensionnement que l'on puisse formuler pour toute structure. Ceci est aussi un problème double :

- celui de la construction de ce modèle c'est à dire du choix des critères génériques,
- celui de la formulation qui s'apparente à un problème de structuration de la connaissance, et qui consiste à déterminer l'expression générique de chaque critère de manière à ce qu'elle soit toujours la même quelle que soit l'application considérée ; cette formulation générique permet alors de construire les objets informatiques correspondants qui pourront être manipulés aisément, rendant ainsi possible l'automatisation de la méthodologie.

Dans cette optique, la généricité est la condition nécessaire pour permettre l'automatisation d'une partie de la modélisation.

## 2.2 Evolutivité

Ce souci d'évolutivité est lié au besoin de généricité évoqué précédemment : l'outil d'analyse ou de dimensionnement doit être modifiable aisément de manière à pouvoir y introduire de nouvelles connaissances en matière de modèles de composants et sur le dimensionnement - nouveaux critères, nouveaux paramètres, nouveaux objectifs.

## 2.3 Intégration

L'intégration d'un outil est sa capacité d'ouverture vers d'autres outils.

On parlera d'outil intégré

- s'il utilise plusieurs outils différents en prenant en charge les liaisons entre ces outils, minimisant ainsi l'intervention de l'utilisateur,
- et s'il est pilotable par d'autres outils, ceci étant rendu possible par la minimisation de l'information nécessaire pour faire fonctionner cet outil, ainsi que la structuration de cette information.

$$\text{Intégré} = \text{intégrateur} + \text{intégrable}$$

En généralisant le concept d'outil à l'acteur (dans notre cas le concepteur), d'après cette définition, un outil bien intégré sera un outil où l'action, ou les actions, de l'utilisateur seront bien spécifiées et codifiées.

La notion d'intégration s'inscrit dans une vision modulaire du problème global de conception, imbriquant les trois niveaux de problème de la conception : choix de la structure, choix de la commande, dimensionnement (cf. Figure I.1).

## 2.4 Rapidité

Un outil d'analyse ou de dimensionnement est susceptible d'être utilisé dans un processus itératif. Donc sa capacité à fournir rapidement un résultat est primordiale. Par contre, la durée de construction de cet outil n'est pas critique, dans la mesure où, une fois construit, il offre les fonctionnalités suffisantes pour réaliser certains ajustements de façon aisée et rapide.

Ainsi, la rapidité de fonctionnement de l'outil doit être privilégiée même si cela se fait au détriment d'une durée notable de construction de l'outil.

Cette promotion de la rapidité de fonctionnement au détriment de la vitesse de construction de l'outil, participe au besoin d'intégration, dans le sens où, plus l'outil a un temps de réponse rapide, plus il est intéressant pour être intégré dans une méthodologie susceptible de faire de multiples appels à cet outil, et en particulier dans une approche globale de conception modulaire que nous envisageons.

## 2.5 Finesse

Le modèle sur lequel repose le développement de l'outil doit proposer un niveau de finesse compatible avec les phénomènes étudiés.

## 2.6 Prise en charge des imperfections des composants :

- Un outil de dimensionnement des convertisseurs statiques doit permettre de prendre en compte
- l'aspect discret du choix des composants dans un catalogue, et
  - les tolérances sur les paramètres du dimensionnement.

## 3 *Principe de conception : reformulation du problème de dimensionnement en un problème d'optimisation contrainte*

Le principe sur lequel repose l'ensemble de la méthodologie que nous proposons, consiste à traduire le problème de dimensionnement dans les termes d'un problème d'optimisation contrainte, ce problème d'optimisation dual pouvant être vu comme un cas particulier d'un problème inverse.

### 3.1 Problème direct (ou problème d'analyse)

#### 3.1.1 Définition

Un problème direct consiste à déterminer les performances d'un système - réel ou décrit par un modèle - en fonction de ses dimensions caractéristiques appelées paramètres.

Pour un jeu de paramètres, on peut déterminer un ensemble de performances et un seul (Figure I.2) :

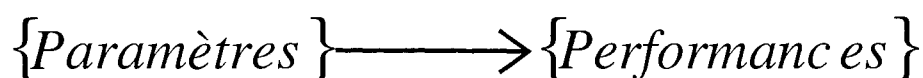


Figure I.2 : Problème direct

Pour un convertisseur statique, les paramètres sont typiquement : les valeurs des inductances, des condensateurs, des résistances, des sources de courant et de tension, des données caractéristiques des interrupteurs, et de la fréquence ... et on voudra déterminer des performances sur les courants moyens, la puissance, les pertes ...

### 3.1.2 Méthodes

Deux méthodes existent pour résoudre un problème direct :

#### **3.1.2.1 *Mesurer les performances sur une maquette prototype du système à analyser***

La méthode expérimentale est bien sûr la plus fiable, et les outils de mesure actuels sont d'une grande précision.

Cependant la mise au point d'un prototype peut être coûteuse en temps et en argent. De plus, chaque modification de la structure représente un surcoût, et pour cette raison, on cherchera à minimiser le nombre d'interventions sur la maquette expérimentale.

#### **3.1.2.2 *Calculer ces performances à partir d'un modèle informatique***

Les méthodes utilisant une modélisation informatique sont à présent éprouvées et fiables. Elles consistent à construire les équations d'état du système, et à en déduire toutes les grandeurs à déterminer. Leur précision est bonne, et dépend du niveau de finesse du modèle et de la méthode de calcul employée. Nous pouvons aujourd'hui commencer à scinder ces méthodes de calcul en deux groupes : les méthodes très usitées par simulations numériques, et les méthodes plus récentes procédant par calculs symboliques.

##### **3.1.2.2.1 Simulations numériques**

La simulation numérique consiste à calculer les performances en utilisant des méthodes d'intégration numériques (Runge Kutta, exponentielles de matrices, trapèzes, etc.) pour la résolution des systèmes d'état.

L'intérêt est que ces méthodes ne sont pas limitées par la taille des systèmes d'état, et donc, pouvant traiter tout type de problèmes, elles sont très générales.

Cependant l'utilisation de telles simulations peut poser des problèmes numériques, dus au choix de la finesse des pas d'intégration et à la méthode, qui affectent la précision et les temps de calcul.

### 3.1.2.2 Méthodes symboliques

La méthode symbolique consiste à intégrer formellement les équations d'état du modèle de façon à écrire toutes les expressions symboliques des performances en fonction des paramètres, et ensuite à évaluer ces formules pour des jeux différents de valeurs d'entrée.

Ces méthodes tendent à se développer [SUN97], [RAJAGOPALAN92], [WEBSTER95], néanmoins elles sont encore appliquées à des problèmes petits ou simplifiés (par exemple des modèles moyens pour les convertisseurs statiques), car, d'une part elles sont limitées aux cas où les systèmes d'état sont intégrables formellement, et d'autre part les expressions symboliques formulées doivent rester de taille réduite de manière à être "*humainement manipulables*", car le développement informatique de ces méthodes reste encore entièrement à la charge du concepteur : il s'agit de traitements manuels utilisant des outils de calculs formels adaptés tels que Mathcad, Maple, Macsyma, ou Mathematica. La mise au point d'un tel modèle d'analyse symbolique peut être longue, mais les calculs des performances ainsi formulées sont très rapides : il s'agit de simples évaluations des formules.

En matière d'intégration, les deux méthodes d'analyse utilisant une modélisation informatique sont plus intéressantes que les prototypes réels : plusieurs appels à ces méthodes peuvent être faits sans augmentation de coût, et avec une fréquence beaucoup plus élevée que celle permise par des retours successifs sur une maquette expérimentale. Les méthodes symboliques de modélisation ont un niveau d'intégration encore plus important du fait de leur réponse rapide, mais aussi grâce à leur formulation :

*« L'avantage de l'analyse symbolique des circuits par rapport à une simulation numérique est qu'elle fournit une vue interne du comportement du circuit, et qu'elle peut engendrer un modèle analytique pour l'optimisation du circuit ainsi que la conception de sa commande » [SUN97].*

## 3.2 Problème inverse

Un problème inverse consiste à retourner le problème direct de façon à déterminer les paramètres caractéristiques d'un système connaissant ses performances. Pour ce type de problème, on préférera le terme de critères plutôt que performances, et entrées plutôt que paramètres. Théoriquement, plusieurs jeux de paramètres (un nombre fini) peuvent amener des performances identiques (il n'y a pas bijection entre l'ensemble des critères et l'ensemble des entrées d'un système).

Donc la donnée d'un jeu de critères peut aboutir à  $n$  jeux différents de valeurs pour les entrées solutions (Figure I.3) :

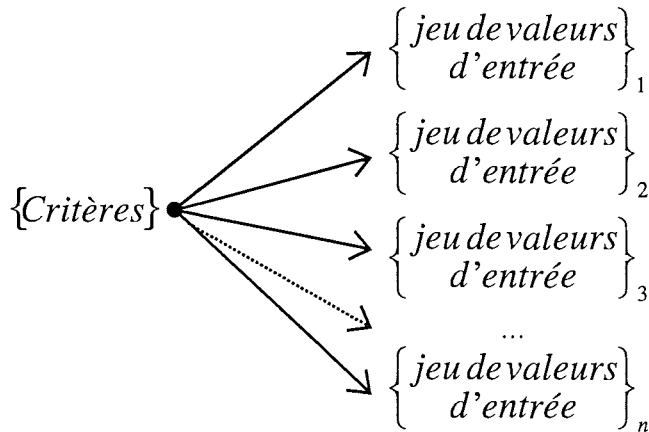


Figure I.3 : Problème inverse

### 3.3 Problème inverse relaxé

Le problème inverse relaxé part d'un ensemble de critères dont les valeurs ne sont plus fixées, mais peuvent varier sur un intervalle continu.

$$\{\text{Critères}\} = \{C_f\} \cup \{C_r\}$$

L'ensemble  $\{C_f\}$  des critères fixes vérifient des relations  $F(C_f) = 0$ .

L'ensemble  $\{C_r\}$  des critères relaxés vérifient des relations  $R(C_r) \leq 0$ .

L'ensemble des jeux d'entrées solutions d'un problème inverse relaxé peut avoir une dimension infinie (Figure I.4), et pour cette raison entre autres, la résolution de ces problèmes est très difficile.

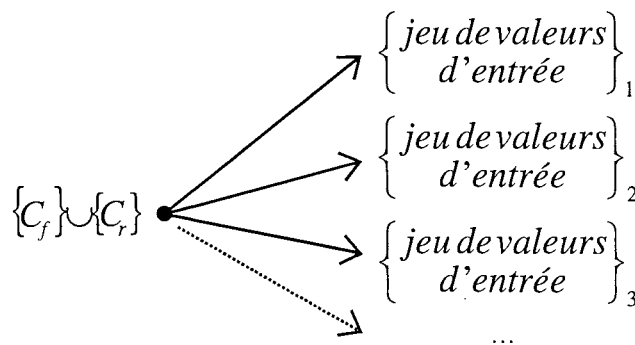


Figure I.4 : Problème inverse relaxé

### 3.4 Problème inverse sous contraintes

Le problème inverse sous contraintes est un problème inverse pour lequel on ne garde que les jeux de solutions dont les valeurs des entrées ( $E$ ) respectent des contraintes pour leurs variations :



ces contraintes peuvent être des inégalités  $c_I$  qui imposent des restrictions lâches sur les solutions, et qui s'écrivent :

$$c_I(E) \leq 0$$

Ces contraintes peuvent aussi être des égalités  $c_E$ . Dans ce cas, elles s'écrivent :

$$c_E(E) = 0$$

et elles imposent des restrictions sévères sur les solutions.

Par conséquent, l'espace des entrées étant rétréci par ces contraintes, on obtient un nombre fini  $m$  de jeux d'entrées solutions pour un problème inverse sous contraintes (Figure I.5).

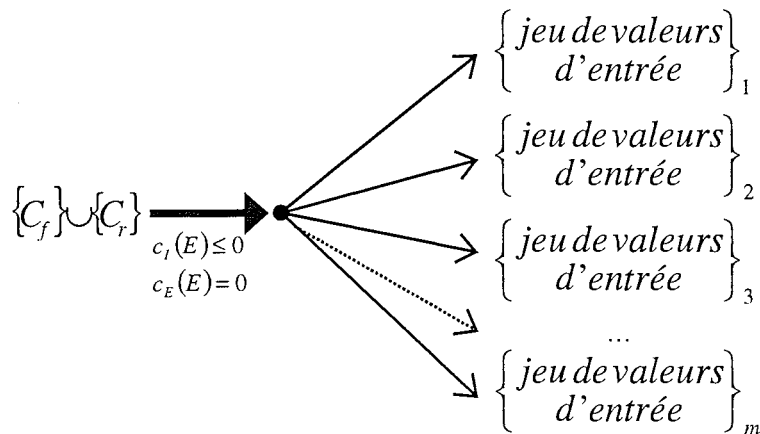


Figure I.5 : Problème inverse relaxé sous contraintes

Et on a  $m \leq n$ ,  $n$  étant le nombre de jeux d'entrées solutions du problème inverse non contraint. Si les contraintes sont trop sévères,  $m$  peut devenir nul, cela veut dire qu'un problème inverse sous contraintes peut être sans solution.

### 3.5 Problème d'optimisation contrainte

#### 3.5.1 Définition

Un problème d'optimisation contrainte peut être défini comme un problème inverse relaxé et sous contraintes auquel on ajoute un objectif en terme de minimisation d'une fonction coût (ou fonction objectif) qui dépend des critères et des entrées. Parmi l'ensemble des solutions d'un problème inverse sous contraintes relaxé, une seule fonction permet de minimiser la fonction objectif (Figure I.6).

$$\begin{array}{c}
 \{C_f\} \cup \{C_r\} \xrightarrow[c_i(E) \leq 0]{F(C)=0} \{Jeu\ de\ valeurs\ d'entrée\} \\
 R(C) \leq 0 \quad c_E(E) = 0 \\
 f(C, E) \text{ minimum}
 \end{array}$$

Figure I.6 : Problème d'optimisation contrainte

Le problème d'optimisation contrainte peut se formuler comme suit :

Minimiser  $f(E, C)$

$$F(C_f) = 0$$

Avec  $R(C_r) \leq 0$

$$c_E(E) = 0$$

$$c_i(E) \leq 0$$

Et on peut généraliser cette formulation en considérant les critères  $C_f$  et  $C_r$  et les entrées  $E$  comme des paramètres  $p$  indifférenciés du problème d'optimisation, et en définissant deux types de contraintes sur ces paramètres :

. l'ensemble des contraintes d'égalité  $E(p) = F(C_f) \cup c_E(E)$ ,

. l'ensemble des contraintes d'inégalité  $I(p) = R(C_r) \cup c_i(E)$ .

Alors, le problème d'optimisation contrainte se formule :

Minimiser $f(p)$ Avec $I(p) \leq 0$ $E(p) = 0$
--

Ainsi, sont traités indifféremment les contraintes sur les entrées et les critères, comme des contraintes sur les paramètres du problème d'optimisation.

Finalement, critères et entrées sont des paramètres indifférenciés pour le problème d'optimisation contrainte :

- ils peuvent être contraints de la même façon à des valeurs fixes ou sur des intervalles, et
- ils seront amenés à évoluer pendant le processus d'optimisation

Ces paramètres (critères et entrées) engendrent l'espace de recherche du problème d'optimisation, qui est limité par leurs contraintes.

Résoudre un problème d'optimisation contraint revient donc à explorer cet espace de recherche pour déterminer le point de cet espace qui minimise la valeur de la fonction objectif.

L'espace de recherche du problème d'optimisation est engendré par l'ensemble des paramètres (entrées et critères). Or, les critères dépendent des entrées : la relation  $C(E)$  entre l'ensemble des jeux d'entrées et l'ensemble des jeux de sorties peut être matérialisée par un prototype réel ou explicitée dans un modèle informatique. Donc la dimension de l'espace de recherche est donnée par le nombre des entrées qui sont les variables indépendantes du système. D'autre part, cet espace est limité par les contraintes sur les paramètres. Ainsi, résoudre le problème d'optimisation contraint revient à explorer l'espace de recherche limité par les contraintes de façon à déterminer le point de cet espace qui minimise la valeur de la fonction objectif.

### 3.5.2 Méthodes de résolution d'un problème d'optimisation contraint

Nous n'avons à notre disposition que le modèle d'analyse (réel ou informatique) du système. La résolution du problème d'optimisation contraint consiste donc à procéder par essais successifs des paramètres d'entrée extraits dans l'espace de recherche en itérant sur le modèle d'analyse. Deux types de méthodes existent pour réaliser ce travail [NIQUIL97].

#### 3.5.2.1 *La méthode d'énumération complète*

Cette méthode dite *méthode du British Muséum* consiste à essayer toutes les solutions, et donc elle n'est applicable que très rarement lorsque l'espace de recherche est très petit (ce qui ne sera pas le cas pour nos problèmes).

#### 3.5.2.2 *Des méthodes "intelligentes" d'exploration de l'espace de recherche*

Ces méthodes offertes par la programmation non linéaire évitent l'énumération complète. On peut les classer en deux groupes [KONE93].

##### 3.5.2.2.1 Méthodes primitives

Les méthodes primitives permettent de résoudre directement le problème contraint sans transformation.

Parmi celles-ci, on trouve :

#### 3.5.2.2.1.1 Des algorithmes stochastiques :

- algorithme de Monte-Carlo [MOSES91], [CAMERON92], [SRIDHAR95],
- algorithme génétique [LEVITIN95], [NAITOH95], [SALUDJAN97].

Bien adaptés aux espaces de recherches discrets, leurs avantages sont :

- la possibilité de partir d'un point de départ quelconque, et
- le fait qu'ils ne sont pas piégés par des extremums locaux, néanmoins,
- ils sont beaucoup plus lents que les algorithmes convergents [SALUDJAN97], et
- leur mise en œuvre n'est pas générique car elle demande de mettre au point certains coefficients inhérents aux algorithmes, variables d'un problème à l'autre, et surtout dont on ne sait pas encore régler les valeurs d'après les données du problème (pour un algorithme génétique : taille de la population, coefficients de mutation, cross-over ..., pour un algorithme du Monte-Carlo : nombre d'itérations, accélérateurs ...).

#### 3.5.2.2.1.2 Des algorithmes d'énumération implicite

En considérant l'espace de recherche comme un arbre, ces algorithmes accélèrent l'exploration en élaguant le plus de branches possibles.

Des techniques "d'intelligence artificielle" permettent de réaliser cette tâche :

- Les algorithmes de séparation et évaluation (branch and bound) et les systèmes experts utilisent des méthodes heuristiques pour parcourir judicieusement l'arbre de recherche. Ils sont génériques, mais la gestion de chaque itération du processus de recherche est lourde et complexe (justement pour assurer la généralité) et donc ces méthodes sont relativement lentes.
- Les réseaux de neurones présentent des temps de mise en œuvre et des problèmes d'apprentissage qui affectent grandement la généralité de ces techniques.

#### 3.5.2.2.2 Méthodes convergentes

Les méthodes convergentes procèdent par transformation du problème contraint en problème sans contrainte de manière à pouvoir effectuer une exploration partielle et heuristique de l'espace de recherche [KONE93]. De nombreuses méthodes de transformation existent [ERTAN93], [KUROE95], [VIAROUGE93].

Parmi les méthodes convergentes, on trouve la famille des algorithmes du gradient (algorithme du gradient, algorithme du recuit simulé, algorithme tabou) qui utilise l'information donnée par les

dérivées des contraintes par rapport aux paramètres d'entrée, pour parcourir l'espace de recherche dans des directions préférentielles, et ainsi accélérer la convergence [KONE93], [SINGH94]. Leur efficacité dépend de la précision obtenue pour l'évaluation des dérivées, ce qui préconise l'utilisation du calcul formel. Mais l'inconvénient majeur de ces méthodes est le risque d'être piégé par un extremum local. Cependant leur généralité et leur efficacité répondent aux critères de sélection des méthodes pour leur intégration dans notre approche de dimensionnement :

*« De part l'importance de la généralité dans les problèmes de dimensionnement (problèmes avec ou sans contraintes égalités ou inégalités essentiellement non linéaires) les méthodes de transformation qui utilisent des algorithmes d'optimisation sans contrainte, semblent être naturellement les mieux adaptées (...). Les algorithmes disponibles sont sûrs et efficaces et utilisés avec succès en électrotechnique ». [KONE93]*

Plusieurs travaux dans le domaine de l'électrotechnique, pour le dimensionnement des machines [WURTZ96a,b], [SINGH94] et des convertisseurs [TOURKHANI95,97], utilisent les méthodes convergentes pour résoudre leurs problèmes d'optimisation.

Les résultats obtenus par des méthodes convergentes sont sensibles à leur initialisation : ils peuvent fortement varier suivant le point de départ choisi dans l'espace de recherche, et ils ne sont pas très distants de ce point (la recherche s'arrête au premier extremum). Cela altère la capacité d'innovation d'une méthodologie de dimensionnement qui utilise ce type de méthodes pour son optimisation : il s'agit de partir du dimensionnement d'un convertisseur déjà existant pour aller vers de nouvelles dimensions répondant à de nouveaux critères d'utilisation (fréquence, puissance, ...). Cette problématique de RE-DIMENSIONNEMENT est la plus courante dans le domaine de l'électrotechnique où, d'une part peu de nouvelles structures de convertisseurs sont créées, et d'autre part un premier dimensionnement grossier, issu de l'expérience du concepteur, accompagne toujours ces nouvelles structures.

## **4 Positionnement des méthodes d'optimisation par rapport à la conception des convertisseurs de puissance**

### **4.1 Conception de structures des convertisseurs statiques**

Les méthodologies de conception structurelle des convertisseurs statiques concernent deux problèmes bien précis.

- Une problématique de choix de la nature des composants pour une structure donnée (qui est davantage un problème topologique), qui consiste à placer les semiconducteurs adéquats dans le squelette de la structure du convertisseur à partir des ondulations en entrée et sortie données par le concepteur (logiciel CAGECT [SAAD93], [FEZZANI93,96,97]). Ceci peut être étendu à la conception fonctionnelle [LECHEVALIER97].
- Une problématique de choix de structure dans une bibliothèque prédéfinie à partir des données de conception saisies par le concepteur ([HSIEH92], [DEBEBE93]).

Pour ces deux problématiques, la méthodologie de résolution employée s'appuie sur une base de connaissance construite avec des règles qui modélisent le raisonnement de l'expert (règles statiques décrivant les interrupteurs, règles de commutation, règles traduisant les lois de Kirshoff, règles s'appuyant sur la notion de cellule de commutation). C'est une approche système expert qui incorpore l'heuristique des concepteurs expérimentés, et peut la restituer lisiblement à des fins pédagogiques par exemple [DEBEBE93], [FEZZANI93], [FEZZANI97].

### **4.2 Dimensionnement des composants des convertisseurs statiques**

#### **4.2.1 Traduction du problème de dimensionnement contraint comme un problème d'optimisation contrainte**

[BOUCHE95] [KONE93] [VIAROUGE93] [FRON94], [WURTZ96a,b]

Voici la formulation du problème de dimensionnement telle qu'elle soit traduisible en un problème d'optimisation contrainte :

- un ensemble de dimensions indépendantes du système à dimensionner,
- un ensemble de critères de performances pour ce système, exprimables en fonction des dimensions,
- une fonction de certains de ces critères et certaines dimensions, appelée fonction objectif

- un ensemble de variations acceptables pour les valeurs des dimensions et critères, appelé cahier des charges.

Le dimensionnement consiste à déterminer le couple (ensemble des valeurs des dimensions, ensemble des valeurs des critères) répondant au cahier des charges et qui minimise la fonction coût, cette dernière étant choisie et construite pour exprimer les objectifs de dimensionnement.

On peut identifier chaque élément du problème de dimensionnement par un élément du problème d'optimisation (cf. Tableau I.1 ci-dessous).

<b>Problème de dimensionnement</b>	<b>Problème d'optimisation dual</b>
Dimensions	Paramètres
Critères	
Cahier des charges	Contraintes
point fixe	égalités
intervalle	inégalités
Objectifs à atteindre	Fonction coût multicritère à minimiser

**Tableau I.2 : La traduction d'un problème de dimensionnement en un problème d'optimisation**

Les objectifs à atteindre peuvent être formulés en terme de critères à minimiser ou à maximiser:

Une optimisation vectorielle consiste à optimiser chaque critère-objectif pris un par un de façon indépendante. Or les méthodes d'optimisation fonctionnent sur une fonction objectif scalaire. Donc pour traiter ces problèmes d'optimisation multicritère, la fonction objectif vectorielle est transformée en fonction objectif scalaire par l'introduction de facteurs de pondération.

Cette traduction du problème multicritère à l'aide des facteurs de pondération pose un problème de validité de l'optimum calculé, et de sa corrélation avec les poids affectés. C'est l'effet horizon ([AMBROSIO94], [HAWORTH93]), expression imagée pour décrire le fait que le procédé qui consiste à assigner arbitrairement des poids aux critères-objectifs ne garantit pas que l'optimisation voulue sera réalisée.

## 4.2.2 Problèmes particuliers du dimensionnement des convertisseurs statiques

Le fonctionnement par commutation des convertisseurs statiques pose d'une part un problème de discontinuité à gérer dans le modèle de dimensionnement [GERBAUD96] ; et d'autre part, le problème du choix du mode de fonctionnement dans lequel sera réalisé le dimensionnement : régimes transitoires ou régimes permanents ?

Ce dernier se pose en termes de modèle : comment déterminer le mode de fonctionnement dans le modèle de dimensionnement (cf. chap. II) ; et en termes de temps de calcul : l'obtention d'un état permanent par des méthodes de simulation peut être gourmande en temps de calcul [TOURKHANI97].

## 4.2.3 Méthodes utilisées pour le dimensionnement des convertisseurs statiques

En considérant le problème de dimensionnement comme un problème d'optimisation contrainte, lui-même vu comme un cas particulier des problèmes inverses, il est possible de classer les méthodes de dimensionnement en deux groupes selon leur niveau de prise en charge de la résolution du problème inverse associé :

- *les méthodes d'analyse orientée conception* [HAMILL93] laissent le problème du parcours dans l'espace de recherche au soin de l'utilisateur, en lui offrant l'information et les outils nécessaires pour réaliser cette tâche au mieux.
- *les méthodes de conception* véritables prennent en charge le problème de parcours de l'espace de recherche et cherche à automatiser l'ensemble du processus de résolution du problème inverse.

### 4.2.3.1 **Les méthodes d'analyse orientée conception.**

Ces méthodes d'assistance à la conception consistent - telles des fils d'Ariane dans un dédale - à offrir une cartographie plus ou moins précise de l'espace de recherche afin d'aider le concepteur à le parcourir efficacement. Elles mettent en œuvre des outils de simulation (SPICE, CIRCUIT, SUCCESS, SABER, SIMPLORER...), des outils de calculs symboliques (Mathcad), des maquettes expérimentales, et cherchent à offrir un modèle d'analyse conçu de telle façon qu'il donne l'information nécessaire pour diminuer le nombre d'allers et retours, itérations nécessaires pour résoudre le problème inverse. L'information ainsi fournie est constituée de visualisations



graphiques, de constructions d'abaques, de procédures itératives automatisées, afin qu'il soit possible de réaliser les études de sensibilité de certains critères en fonction de certains paramètres d'entrée du dimensionnement. Ces méthodes sont très populaires, et ont fait preuve de leur efficacité sur un nombre considérable d'applications variées [TOURKHANI91], [HAMILL93], [REATTI94], [BHAT95], [HAMMA95], [SPIAZ96]... . Cependant, chacune d'elle a été mise en œuvre au coup par coup de façon spécifique. Si des travaux de généralisation ont été entrepris (systèmes experts [CHHAYA92,93], approche orientée objet pour la simulation [MENDEZ95], logiciel d'aide pour l'analyse [MARANESI92,94]), les considérations de conception sur lesquelles ces méthodes d'analyse orientée conception sont fondées restent néanmoins limitées par leur nécessaire exploitation humaine : elles doivent mettre en jeu un nombre restreint de paramètres et de relations afin de pouvoir être gérées par le concepteur dans son travail de résolution du problème inverse. Cela sous-entend que de nombreuses hypothèses simplificatrices doivent être faites : on parlera de « pire des cas » [KUO-PENG91], de « frontières de validité », « toutes choses égales par ailleurs », autant d'expressions pour cacher l'incapacité à gérer humainement l'ensemble des interactions qui lient tous les paramètres de dimensionnement d'un convertisseur statique. Ces simplifications, qui, induisant un parcours parcellaire de l'espace de recherche, négligent l'exploration de certaines zones de cet espace, peuvent nuire à la qualité du dimensionnement [SRIDHAR95]. En pratique, ces méthodes se satisfont d'un dimensionnement lorsqu'il est obtenu, sans chercher à l'optimiser.

#### **4.2.3.2 Les méthodes de conception.**

Les méthodes de conception tentent de résoudre automatiquement le problème inverse du dimensionnement dans son ensemble, en prenant en charge les itérations à effectuer sur le modèle analytique avec une méthode d'optimisation. Elles peuvent être divisées en deux catégories suivant la formulation employée pour le modèle d'analyse sur lequel elles sont fondées :

- une optimisation sur un modèle d'analyse numérique, ou
- une optimisation sur un modèle d'analyse symbolique.

##### **4.2.3.2.1 Optimisation sur un modèle d'analyse numérique.**

Pour cette méthode de conception, la partie analyse est réalisée sur la base de simulations numériques, et la résolution du problème inverse par un algorithme d'optimisation.

Certains outils d'analyse des circuits électriques offrent des fonctionnalités d'optimisation (Matlab : moindres carrés ; Saber et Simplorer : Monte-Carlo), mais leur mise en œuvre est complexe et coûteuse en temps, et demande une analyse approfondie du circuit étudié (pour Matlab) ainsi qu'une bonne maîtrise de ces outils de simulation.

Cependant, ces méthodes numériques ont l'avantage de pouvoir s'appliquer à des modèles fins sans limitation due à la complexité de leurs systèmes d'état, ces systèmes étant intégrés numériquement (par des méthodes de type Runge-Kutta ou trapèze par exemple). Néanmoins, cet avantage est contrebalancé par les problèmes de précision et de temps de calcul induits par les méthodes d'intégration numérique employées. Ces problèmes se répercutent sur le choix du pas d'intégration : un pas d'intégration trop grand dégrade la précision tandis qu'un pas trop fin augmente très considérablement les temps de calcul. L'ajustement de ce pas peut être optimisé dynamiquement par des méthodes d'intégration à pas variable, cependant il doit être paramétré de façon spécifique et non aisée pour chaque application traitée, ce qui altère fortement la généralité de telles méthodes de conception. De plus, les temps de calcul sont toujours liés à l'obtention d'un état permanent. Même si des méthodes existent pour accélérer ce processus [TOURKHANI95,96,97], ce calcul réalisé à chaque nouvelle itération de l'algorithme d'optimisation peut être gourmand en temps. Enfin, souvent, ces méthodes de dimensionnement sont mises en œuvre au cas par cas, et des travaux ont été entrepris pour les généraliser [TOURKHANI96].

### 4.2.3.2.2 Optimisation sur un modèle d'analyse symbolique.

Cette méthode de dimensionnement tend à se développer dans le fil de l'utilisation des méthodes d'analyse orientée conception, l'idée étant de coupler l'outil d'analyse avec un algorithme d'optimisation [BOUCHE95]. L'essor de cette méthode est due à l'apparition des logiciels de calculs symboliques tels Maple, Macsyma, Mathematica, et Mathcad, qui offrent des interfaces de développement conviviales. Toutefois, là encore, la méthode est mise en œuvre au cas par cas, pour une structure de convertisseur donnée, et un problème de dimensionnement bien spécifique. Cependant, le développement de cette méthode de dimensionnement est limitée parce qu'elle est toujours mise en œuvre manuellement par le concepteur, ce qui impose des tailles raisonnables pour les expressions symboliques utilisées, et un nombre limité de paramètres, ce qui implique des modèles très simplifiés, moins riches que ceux utilisables dans les méthodes fondées sur des modèles numériques. Néanmoins, les travaux de J.M. Bouché [BOUCHE95] ont montré la possibilité de traiter un nombre plus grand de paramètres de dimensionnement, et ils ont aussi

montré que l'avenir de ce type de méthodes de dimensionnement, fondé sur de l'optimisation et sur un modèle d'analyse symbolique, dépendait de la mise en œuvre d'outils capables de prendre en charge la complexité des expressions constitutives des modèles d'analyse symboliques.

## **5 Démarche proposée**

La démarche de dimensionnement des convertisseurs statiques que nous proposons pour répondre au plus grand nombre des critères évoqués précédemment (cf. §2), est issue de l'analyse de l'état de l'art dans le domaine.

### **5.1 Présentation**

Pour des raisons de précision, de rapidité et de généralité, nous avons décidé de fonder notre démarche sur une méthode de conception symbolique avec optimisation, autour de l'outil PASCOSMA développé dans l'équipe C.D.I<sup>1</sup> du L.E.G<sup>2</sup> [WURTZ96a]. Cette démarche s'attache à formuler génériquement et automatiquement le maximum des contraintes du dimensionnement à partir de la simple description du circuit du convertisseur statique, et à décharger l'utilisateur des retraitements symboliques nécessaires pour formuler le problème de conception contrainte spécifique au convertisseur étudié.

La démarche prend en charge la traduction de ce problème de conception contrainte en problème d'optimisation. Le but de cette méthodologie est de tendre vers un environnement intégré de dimensionnement des convertisseurs statiques incluant des outils de modélisation, d'analyse, de simulation, d'optimisation et de post-traitement.

### **5.2 Hypothèses de travail**

Nous rappelons les hypothèses de travail dues aux outils et méthodes employés pour la mise en œuvre de la démarche.

#### **5.2.1 Dimensionnement continu**

Notre type de problème ne peut pas se classer ni dans l'ensemble des problèmes complètement continus, ni dans l'ensemble des problèmes entièrement discrets. C'est un problème hybride dont

---

<sup>1</sup> Conception et Diagnostic Intégrés

<sup>2</sup> Laboratoire d'Electrotechnique de Grenoble

une partie des données varie continûment (cela concerne les variables d'état du circuit, tous les signaux électriques à travers chaque composant pendant chacune des configurations du mode de fonctionnement), et une autre partie prend des valeurs discrètes (cela concerne les caractéristiques de la plupart des composants). Théoriquement, les méthodes d'optimisation stochastiques semblent être les seules envisageables pour résoudre ce type de problèmes, dans la mesure où elles sont applicables dans des domaines continus, alors que les méthodes convergentes sont inutilisables en discret. Toutefois, dans notre souci de généricité et d'intégration qui impose une réponse rapide, nous décidons de considérer notre problème de dimensionnement comme un problème continu au niveau du modèle de dimensionnement, afin de pouvoir utiliser un algorithme du gradient générique et rapide, mais qui ne peut explorer qu'un espace de recherche continu ; et nous proposons de prendre en charge hors modèle les discontinuités, dans la méthodologie, par une adaptation a posteriori des résultats obtenus aux valeurs discrètes acceptables.

### 5.2.2 Mode de fonctionnement figé

Afin d'assurer la validité physique des équations de fonctionnement qui fondent l'ensemble du dimensionnement, nous devons figer le mode de fonctionnement dans lequel nous dimensionnons le convertisseur.

### 5.2.3 Domaine temporel

Les critères de dimensionnement sont formulés à partir des expressions symboliques temporelles des variables d'état. Cependant il est possible d'exploiter les transformées de Laplace de ces expressions (que l'on peut toujours obtenir) pour formuler d'autres critères (sur la stabilité par exemple).

### 5.2.4 Point de départ connu

Ce point est nécessaire pour initialiser l'algorithme d'optimisation du gradient, nous nous plaçons donc d'emblée dans une problématique de redimensionnement.

### 5.2.5 Modèles simplifiés

Il est parfois nécessaire d'utiliser des modèles simplifiés pour permettre l'intégration symbolique des systèmes d'état. Cependant nous verrons que nous pouvons envisager d'étendre la méthodologie à des modèles plus complexes, en employant une technique d'intégration faisant intervenir un peu de calculs numériques (chap. IV).

### 5.3 Apports

La problématique de dimensionnement des convertisseurs statiques est actuellement résolue par des méthodes lourdes et difficiles à mettre en place, parce que réalisées le plus souvent au coup par coup, le travail réalisé pour une application étant très peu exploitable pour la suivante : l'implantation du problème direct est entièrement réalisée par le concepteur, en prenant en compte tant bien que mal les contraintes intrinsèques liées au fonctionnement, par des restrictions rigides de l'espace de recherche, issues d'une analyse approfondie de la structure. Jusqu'à présent, les seuls travaux entrepris dans le sens d'une automatisation de ce travail concernent des méthodes numériques qui s'avèrent relativement lentes et dont la mise en œuvre est délicate du fait de l'ajustement nécessaire des paramètres de calcul. De plus, lorsque des méthodes d'optimisation sont entreprises pour résoudre le problème inverse attaché, cela est souvent réalisé de façon spécifique pour l'application donnée, et pour un nombre limité de critères de dimensionnement.

L'apport de la démarche que nous proposons se fait à deux niveaux :

- Au niveau du modèle, nous proposons de construire ce que nous avons appelé un modèle de dimensionnement symbolique fondé sur deux principes :
  - **Il prend en charge automatiquement la formulation du problème physique du fonctionnement du convertisseur**, et permet au concepteur de se concentrer sur le véritable problème de dimensionnement, un cadre générique de formulation de ses propres critères de dimensionnement lui étant offert par ailleurs. La généralité du modèle de dimensionnement fondé sur les expressions temporelles symboliques des variables de la structure présente deux inconvénients :
    - d'une part, elle est rapidement limitée par la nécessaire intégration symbolique des systèmes d'état qui régissent le fonctionnement, intégration formelle qui n'est plus générique au delà d'une dimension quatre pour les systèmes différentiels ;

Cependant, nous allons voir que la généralité du modèle fondé sur l'intégration symbolique des systèmes d'état peut être étendue au delà de la dimension quatre par des techniques mixant traitement formel et calcul numérique (cf. Chap. IV). Ceci qui élargit encore le champ d'application de l'approche de dimensionnement, à des structures de convertisseurs plus complexes, et sur lesquelles il sera alors possible de réaliser un *dimensionnement intégré*, et non plus un dimensionnement par parties. Cette technique qui nécessite de découper le problème en morceaux dimensionnés séparément et de les coupler a posteriori en moyennant

les liaisons entre chaque morceau, est la seule méthode envisageable jusqu'à présent, mais n'est pas satisfaisante.

- d'autre part, cette généricité impose une forte complexité du modèle, qui, ainsi défini, ne serait plus que très difficilement formulable manuellement.

Toutefois, une telle base générique permet d'ores et déjà une construction automatique du modèle de dimensionnement, et ainsi permet de résoudre tous les problèmes jusqu'alors traités manuellement, et limités par le même problème mathématique de l'intégration des systèmes d'équations différentielles. A ce stade, l'apport se fait déjà sur le nombre d'opérations et la complexité des expressions que peut traiter l'ordinateur, cette gestion de la complexité permettant ainsi d'éviter des simplifications de modèle jusqu'alors nécessaires pour une manipulation humaine.

- **Le modèle de dimensionnement** que nous proposons **prend aussi en charge le couplage analyse – optimisation**, en intégrant l'utilisation du logiciel PASCOSMA développé dans l'équipe C.D.I du L.E.G, permettant ainsi de construire un outil dédié au dimensionnement de l'application étudiée, qui offre une interface conviviale pour l'utilisateur. Ainsi couplé avec PASCOSMA, le modèle générique de dimensionnement que nous proposons permet de prendre en compte un grand nombre de paramètres pour le dimensionnement.

- Au niveau méthodologique, l'apport que nous faisons par cette démarche de construction d'un outil de dimensionnement dédié à une structure et à un fonctionnement s'exprime en terme d'**intégration** :

- Nous proposons **une méthodologie intégrée** de construction de l'outil de dimensionnement qui modélise tout un processus de développement (Chap. III), en mixant différents outils (générateurs automatiques de modèles et de programmes, logiciels de calculs formels, tableurs...), et différentes méthodes (simulation, retraitements formels...).
- Cette méthodologie, qui fournit plusieurs outils très utiles dans le domaine de la conception : simulation, analyse de sensibilité des critères par rapport aux entrées, et dimensionnement ; peut être vue comme **un agent intégrable** dans une méthodologie globale de conception (cf. Figure I.7)

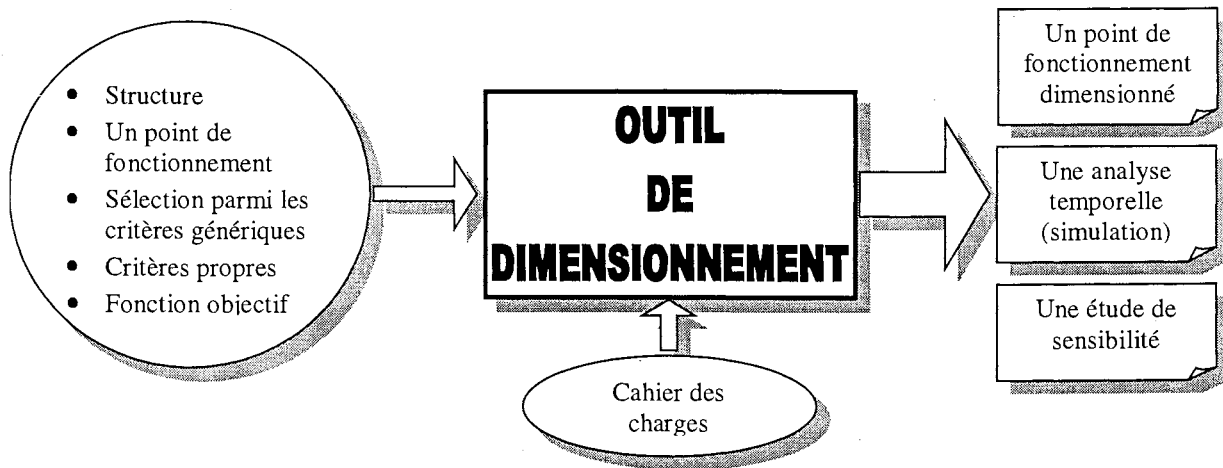


Figure I.7 : L'outil de dimensionnement vu comme un agent d'une méthodologie globale de conception

## 6 Conclusion

Dans ce chapitre, nous présentons notre façon d'appréhender le problème de dimensionnement comme un problème inverse, et nous faisons un état de l'art des méthodes utilisées pour résoudre cette problématique de conception en électronique de puissance. Nous proposons de traiter le problème de dimensionnement comme un problème d'optimisation contrainte. Aussi, il nous est apparu nécessaire de présenter les différentes techniques existantes pour résoudre un tel problème d'optimisation contrainte afin de choisir la plus pertinente pour notre problème particulier du dimensionnement des convertisseurs statiques qui fait intervenir des paramètres discrets et continus (problème mixte). L'ensemble de la démarche que nous proposons ensuite repose sur le choix de cette méthode d'optimisation : nous choisissons d'exploiter au maximum l'aspect continu du problème du dimensionnement des convertisseurs statiques, et ainsi de pouvoir employer une méthode convergente (un algorithme du gradient) qui nous offre rapidité, précision et généralité.

A partir de ces considérations, nous positionnons une démarche de dimensionnement des convertisseurs statiques, en termes de choix techniques généraux, hypothèses de travail et apports.

Les trois chapitres suivants présentent plus en détail la méthodologie que nous proposons dans ce cadre : le chapitre II détaille le modèle de dimensionnement générique que nous proposons de formuler symboliquement, modèle qui induit le choix de la méthodologie de dimensionnement intégrant différentes phases de modélisation, simulation et optimisation, comme nous le montrons dans le chapitre III. Le chapitre IV présente dans quelles mesures l'ensemble la méthodologie est automatisable, et comment est réalisée son implantation informatique. Enfin, le chapitre V présente deux applications dimensionnées en utilisant cette méthodologie.

## *Chapitre II :*

# *La formulation générique du modèle de dimensionnement*





# CHAPITRE II : LA FORMULATION GÉNÉRIQUE DU MODELE DE DIMENSIONNEMENT

Dans ce chapitre, nous présentons la formulation du modèle de dimensionnement que nous proposons, avec, à tous les niveaux, le souci de généralité évoqué précédemment et que nous allons définir plus en détails dans le paragraphe suivant : généralité au niveau du fond - de la définition - et au niveau de la forme - de la formulation -, pour les critères, les paramètres, la fonction objectif, et le cahier des charges.

## 1 Définitions

### 1.1 Formulation du modèle de dimensionnement

En vue de la traduction du problème de dimensionnement en un problème d'optimisation que nous proposons de traiter avec un algorithme du gradient, nous proposons de construire ce que nous avons appelé *un modèle de dimensionnement* qui contiendra toutes les équations de dimensionnement de la structure. Ces équations, que nous formulons symboliquement, régissent les relations entre les entrées, les critères, et leurs gradients. C'est la formulation de toutes ces équations que nous réalisons symboliquement, que nous avons appelée un peu improprement la *formulation du modèle de dimensionnement*.

### 1.2 Généralité

Nous allons tenter de définir le concept de généralité tel que nous l'appréhendons dans cet exposé, en le déclinant sous les différentes formes qu'il peut prendre.

- **Genre :** Un genre, ou une classe de problèmes et un ensemble de problèmes définis par un sujet et un objectif communs. Une classe de problèmes peut être résolue par différentes méthodes, en utilisant différents modèles.

Par exemple, nous étudions dans cette thèse la classe des problèmes de dimensionnement des convertisseurs statiques.

- **Générique** : C'est ce qui permet d'engendrer (de générer) un genre (une classe) de problèmes donnés.

Par exemple : un modèle générique de dimensionnement des convertisseurs statiques

- **Générer** : Générer, ou engendrer un modèle, consiste à construire le modèle dédié à une application donnée en utilisant le moule d'un modèle générique.

Ainsi, la méthode de résolution du problème de dimensionnement des convertisseurs statiques proposée, consiste à mettre au point une méthodologie permettant de générer un modèle de dimensionnement pour une application donnée. Ce modèle de dimensionnement sera alors transformé en problème d'optimisation qui pourra être résolu par une méthode numérique.

- **Données génériques** : On dira d'un ensemble de données permettant d'engendrer (ou de générer) un modèle qu'elles sont génériques si elles sont nécessaires et suffisantes pour construire ce modèle.

Pour le modèle de dimensionnement que nous proposons, les seules données que nous possédons sont issues du modèle de fonctionnement (les systèmes d'état, les tests de commutation, et les relations entre les autres variables du circuit et les variables d'état), et d'une éventuelle simulation d'un point de fonctionnement initial. Ce sont donc parmi elles que se trouvent les candidates pour faire partie de l'ensemble des données génériques, qui elles-mêmes peuvent fournir d'autres données génériques (valeurs efficaces, valeurs moyennes...).

- **Formulation générique** : Une formulation (d'une équation, d'une règle, ou d'un modèle) est dite générique :
  - si elle est fondée sur des données génériques, et si l'agencement de ces données peut être reproduit sans modification d'une application à l'autre
  - et si cette formulation permet de prendre en compte tous les aspects de la classe de problèmes considérée.

Ainsi, formuler génériquement un modèle permet de construire un modèle générique de description d'un genre de problèmes donné.

- **Généricité de notre point de vue**

Pour notre modèle de dimensionnement, la généricité de la formulation est conditionnée par

- la liste des données génériques : [mode, systèmes d'états, règles de commutations, expressions des signaux, point de fonctionnement initial, cahier des charges] ;
- les deux aspects du problème de dimensionnement des convertisseurs statiques :
  - l'aspect physique lié au fonctionnement par commutations du convertisseur,

- et l'aspect spécifique lié au dimensionnement que veut mettre en œuvre le concepteur (et qui fait intervenir les valeurs efficaces, valeurs moyennes...)

## **2 Problèmes de formulation du modèle de dimensionnement**

Tous les problèmes qui apparaissent sont induits de notre volonté d'être générique dans la construction du modèle de dimensionnement : la difficulté consiste en fait à définir les paramètres qui, quelle que soit la structure de convertisseur étudiée, vérifieront les points suivants :

- Dans la mesure du possible, ils devront être formulables génériquement : leur expression symbolique doit être exprimable dès lors que l'on possède l'expression temporelle des variables d'état, et doit faire intervenir des paramètres identifiables, c'est-à-dire des paramètres dont il est possible d'obtenir une valeur initiale de façon relativement aisée, à partir du point de fonctionnement de départ et d'une simulation de ce point.
- Ils devront permettre d'assurer la cohérence du fonctionnement. L'ensemble de la formulation symbolique du modèle de dimensionnement repose sur les expressions temporelles formelles des variables d'état durant un cycle de fonctionnement complet du convertisseur étudié. Cependant, du fait du fonctionnement discret des convertisseurs, l'expression temporelle de chaque variable d'état est définie par morceaux : elle varie d'une configuration à l'autre. Nous verrons quels retraitements sont opérés sur ces variables d'état afin de formuler le modèle de dimensionnement, et assurer la viabilité physique du modèle ainsi défini.
- Ils devront servir de base pour donner la possibilité à l'utilisateur d'exprimer ses requêtes spécifiques, en termes de critères de dimensionnement des composants autres que ceux cités précédemment : sur les pertes, les dimensions géométriques ...

Les données génériques que nous fournit le modèle de fonctionnement sont :

- l'enchaînement des configurations qui caractérisent le mode de fonctionnement temporel,
- les expressions symboliques temporelles des signaux dans tous les composants pour chaque configuration,
- et les règles de commutation des interrupteurs.

A cela s'ajoutent les données calculables à partir d'un point de fonctionnement initial que l'on peut simuler.

Nous allons voir dans ce chapitre comment formuler le modèle de dimensionnement à partir de cette information, de manière à ce qu'il soit possible de le générer automatiquement pour toute application de convertisseur statique.

### **3 Les choix de formulation**

La réponse que nous proposons à ce besoin de généricité, repose sur trois choix :

- figer le mode de fonctionnement,
- apporter une aide à la formulation des critères non génériques,
- normaliser le modèle dans la mesure du possible.

#### **3.1 Mode de fonctionnement figé**

En nous restreignant à l'hypothèse de départ du mode de fonctionnement figé, nous choisissons de contraindre de façon automatique les configurations à s'enchaîner dans un ordre donné, celui du point de fonctionnement initial. De cette façon, la cohérence du fonctionnement est assurée pendant le dimensionnement, c'est à dire qu'il n'y a pas de changement de mode intempestif.

La difficulté à ce niveau consiste à déterminer quels critères permettent de contraindre le mode de fonctionnement : les *critères physiques*. Nous verrons que les critères qui caractérisent ces règles physiques peuvent être formulés à partir d'un nombre donné de retraitements opérés sur les variables d'état, assurant ainsi la généricité de leur formulation, telle que nous l'avons définie.

#### **3.2 Aide à la formulation des critères génériques**

Pour l'ensemble des équations de dimensionnement spécifiques, nous choisissons de fournir toute l'information électrique disponible et nécessaire pour que le concepteur puisse formuler n'importe quel *critère de dimensionnement*.

La difficulté à ce niveau consiste, d'une part à déterminer les critères qui forment l'information électrique minimale indispensable pour le dimensionnement de tout composant, et d'autre part à formuler cette information sur la base d'un nombre déterminé de critères, formulables à partir des variables d'état du circuit.

### 3.3 Normalisation

La normalisation consiste à diviser un critère par la valeur maximale qu'il est susceptible d'atteindre.

- Cette normalisation répond au besoin de généralité. En effet, les valeurs possibles des critères sont ramenées sur une plage unitaire  $[0 ; 1]$ , et ainsi, quel que soit le critère considéré, cette plage de valeur est la contrainte la plus relaxée admissible. De plus, nous allons voir que pour beaucoup des critères physiques, cet intervalle  $[0 ; 1]$  est la seule contrainte à imposer.
- Cette normalisation répond aussi à un souci d'homogénéisation qui sous-tend l'ensemble du modèle de dimensionnement : l'idée est que tout critère est susceptible de pouvoir être implanté dans la fonction objectif avec la méthode des pondérations (cf. §4.3). dans ce cadre, avoir des critères normalisés permet de régler efficacement les poids entre eux, et ainsi d'ajuster les influences relatives de chaque critère sur l'objectif.
- Cela permet d'améliorer la convergence du calcul de dimensionnement (optimisation).

#### Notation

Dans la suite de l'exposé, les variables normalisées seront surmontées d'un  $\sim$ .

## 4 Les éléments constitutifs du modèle de dimensionnement

### 4.1 Classification des données

Avant de poursuivre, classifions les données manipulées.

- $E' = \{\text{entrées primaires}\}$

est l'ensemble des paramètres qui émaillent les systèmes d'état du modèle de fonctionnement, par exemple les condensateurs, les inductances.

- $E'' = \{\text{entrées secondaires}\}$

est l'ensemble des paramètres d'entrée du problème de dimensionnement qui ne sont pas des entrées primaires,

cet ensemble est constitué de deux sous-ensembles :

- $E''_{\text{auto}} = \{\text{entrées secondaires automatiques}\}$

$$E''_{\text{auto}} = \{\text{Conditions Initiales}\} \cup \{\text{durées réduites des configurations}\} \cup \{\text{dates réduites des crêtes}\}$$

C'est l'ensemble des paramètres secondaires créés automatiquement pendant la phase de formulation du modèle de dimensionnement,

- $E''_{uti} = \{\text{entrées secondaires utilisateur}\}$

est l'ensemble des paramètres issus des expressions des critères spécifiquement formulés par l'utilisateur.

Ainsi,

$$E'' = E''_{auto} \cup E''_{uti} \quad (1)$$

- $E = \{\text{entrées}\}$

est l'ensemble des paramètres d'entrée du problème de dimensionnement,

$$E = E' \cup E'' \quad (2)$$

- $C = \{\text{critères}\}$

est l'ensemble des critères du problème de dimensionnement

- $P = \{\text{paramètres}\}$

est l'ensemble des paramètres (entrées et critères) du problème de dimensionnement :

$$P = E \cup C \quad (3)$$

Ces paramètres sont traités indifféremment par l'algorithme d'optimisation, ils peuvent être contraints, et leurs valeurs sont susceptibles d'évoluer pendant le dimensionnement. Cependant, le choix de l'algorithme du gradient, en imposant une initialisation, donne un statut particulier aux paramètres d'entrée qui doivent donc être choisis de façon à pouvoir leur affecter une valeur initiale relativement facilement. Cette considération oriente l'ensemble de la formulation du modèle de dimensionnement, c'est ce que nous allons montrer dans la suite de ce chapitre.

- $G = \{\text{gradients}\}$

est l'ensemble des gradients qui sont nécessaires à l'algorithme d'optimisation que nous employons [WURTZ96a]. Le gradient d'un critère étant défini par le vecteur de ses dérivées par rapport à toutes les entrées du problème de dimensionnement, on peut donc définir l'ensemble G par :

$$G = \frac{dC}{dE} \quad (4)$$

et le nombre d'éléments de cet ensemble est donné par :

$$\text{card}(G) = \text{card}(C) \times \text{card}(E) \quad (5)$$

On peut aussi définir

- $P_C$ , l'ensemble des paramètres dont dépend directement le critère C,  $C = C(P_C)$ ,  $P_C$  peut contenir des entrées primaires, des entrées secondaires, et des critères :

$$P_c \subset P \quad (6)$$

- $DP = \frac{\partial C}{\partial P_c}$  est l'ensemble des dérivées partielles.

## 4.2 Les critères

### 4.2.1 Critères propres au fonctionnement du convertisseur statique : les critères physiques

Afin de contraindre le mode de fonctionnement à rester identique pendant tout le dimensionnement, on se propose de formuler quatre types de critères physiques.

- Les critères permettant d'assurer la continuité des variables d'état.
- Les critères sur les durées des configurations d'un mode.
- Les critères qui assurent le respect des règles de commutation des interrupteurs.
- Les critères sur les pulsations.

#### 4.2.1.1 Continuité des variables d'état

##### 4.2.1.1.1 Définition

Pour les convertisseurs statiques dont le fonctionnement est discret, par définition, les lois de variations des variables d'état sont implicitement continues sur chaque configuration, car obtenues par intégration d'un système différentiel. La continuité doit être aussi vérifiée au moment des commutations d'une configuration à la suivante dans un même mode.

Pour la grandeur d'état  $X_i$ , en considérant le passage de la configuration ( $a$ ) (de durée  $\tau_a$ ) à la configuration ( $b$ ), ceci s'écrit :

$$X_i^{(a)}(\tau_a) = X_i^{(b)}(0) \quad (7)$$

##### 4.2.1.1.2 Notation

Les configurations sont notées par une lettre minuscule entre parenthèse, la première configuration du mode est notée ( $a$ ), la dernière ( $z$ ), de telle façon que le mode est notée [ $a,b,c,\dots,z$ ].

Une configuration quelconque est notée par une lettre grecque minuscule ( $\alpha$ ), ( $\beta$ )...



Les  $n_{VE}$  variables d'état sont différenciées entre elles par un numéro noté en indice  $X_1, X_2, X_i$ . De cette manière, on note l'expression symbolique temporelle de la variable d'état  $X_i$  pendant la configuration ( $\alpha$ ) par :  $X_i^{(\alpha)}(t)$ .

Notons que pour toute variable d'état  $X_i^{(\alpha)}(t)$  l'origine des temps est définie au début de la configuration  $\alpha$ , et la durée est notée  $\tau_\alpha$ .

#### 4.2.1.1.3 Formulation

Les critères qui découleraient intuitivement de ces relations seraient les suivants par exemple :

$$\text{cont}X_i^{(a \rightarrow b)} = X_i^{(a)}(\tau_a) - X_i^{(b)}(0) \quad (8)$$

En contraignant chaque critère  $\text{cont}X_i^{(a \rightarrow b)}, \text{cont}X_i^{(b \rightarrow c)}, \dots, \text{cont}X_i^{(z \rightarrow a)}$  à zéro, pour chaque transition  $(a \rightarrow b), (b \rightarrow c), \dots, (z \rightarrow a)$  du mode  $[a-b-c-\dots-z]$  et chaque variable d'état  $X_i$ , on assurerait ainsi la continuité du fonctionnement. Cependant,  $n_C$  étant le nombre de configurations du mode, et  $n_{VE}$  étant le nombre de variables d'état, une telle formulation ferait intervenir  $n_C \times n_{VE}$  critères, et  $n_C \times (n_{VE} + 1)$  entrées.

Aussi, afin de diminuer ce nombre de critères, nous allons adopter la formulation suivante :

$$\begin{aligned} \text{cont}X_i &= X_i^{(z)}(\tau_z, X0_i^{(z)}) - X0_i^{(a)} \\ X0_i^{(b)} &= X_i^{(a)}(\tau_a, X0_i^{(a)}) \\ X0_i^{(c)} &= X_i^{(b)}(\tau_b, X0_i^{(b)}) \\ &\dots \\ X0_i^{(z)} &= X_i^{(y)}(\tau_y, X0_i^{(y)}) \end{aligned} \quad (9)$$

Cette formulation ne fait plus intervenir que  $n_{VE}$  critères  $\text{cont}X_i$  et  $(n_C + n_{VE})$  entrées : les  $n_C$  durées des configurations et  $n_{VE}$  conditions initiales sur chaque variable d'état, car on n'a plus besoin que d'une entrée  $X0_i^{(z)}$  pour chaque variable d'état, les autres variables  $X0_i$  deviennent des paramètres internes qui sont complètement transparents pour l'utilisateur (pas d'initialisation, pas de contrainte).

Afin d'assurer la continuité des variables d'état, les  $n_{VE}$  critères  $\text{cont}X_i$  seront contraints à zéro.

#### 4.2.1.1.4 Exemple

Illustrons ceci dans l'exemple du convertisseur multirésonant en thyristor dual forward, application présentée plus en détail dans le chapitre V.

Les variables d'état qui le caractérisent sont :  $i_{L1}$  ,  $v_{C1}$  ,  $v_{C2}$  ,

Le mode de fonctionnement choisi est le mode 3, qui présente quatre configurations dans l'ordre C4-C3-C4-C2 (cf. chap. V),

Les équations générées s'écrivent, pour la continuité de la tension  $v_{C1}$  aux bornes du condensateur C1

$$\begin{aligned} covca &= [v_{C12}(\tau_d, vc14) - vc11] / vc1max \\ vc12 &= v_{C14}(\tau_a, vc11) \\ vc13 &= v_{C13}(\tau_b, vc12) \\ vc14 &= v_{C14}(\tau_c, vc13) \end{aligned} \tag{10}$$

où le critère de continuité formulé pour cette variable d'état  $v_{C1}$  est noté *covca*.

Cinq entrées sont apparues dans cette formulation :  $\tau_a$  ,  $\tau_b$  ,  $\tau_c$  ,  $\tau_d$  ,  $vc11$

De la même façon, pour la continuité des deux autres variables d'état  $i_{L1}$  et  $v_{C2}$ , on construit les critères *coila*, et *covcb*, et deux entrées supplémentaires apparaissent : *il11*, *vc21*, ce qui fait au total (cf. Tableau II.1) : trois critères *coila*, *covca*, *covcb* à contraindre à 0, et sept entrées ( $\tau_a$  ,  $\tau_b$  ,  $\tau_c$  ,  $\tau_d$  , *il11*, *vc11*, *vc21*) qui devront être initialisées, et pourront être contraintes comme tous les paramètres du dimensionnement.

Continuité de $i_{L1}$	Continuité de $v_{C1}$	Continuité de $v_{C2}$
$coila = [i_{L12}(\tau_d, il14) - il11] / il1max$	$covca = [v_{C12}(\tau_d, vc14) - vc11] / vc1max$	$covcb = [v_{C22}(\tau_d, vc24) - vc21] / vc2max$
$il12 = i_{L14}(\tau_a, il11)$	$vc12 = v_{C14}(\tau_a, vc11)$	$vc22 = v_{C24}(\tau_a, vc21)$
$il13 = i_{L13}(\tau_b, il12)$	$vc13 = v_{C13}(\tau_b, vc12)$	$vc23 = v_{C23}(\tau_b, vc22)$
$il14 = i_{L14}(\tau_c, il13)$	$vc14 = v_{C14}(\tau_c, vc13)$	$vc24 = v_{C24}(\tau_c, vc23)$

**Tableau II.1 : La formulation des trois critères de continuité des variables d'état pour l'application du convertisseur multirésonant en thyristor dual forward**

#### 4.2.1.1.5 Normalisation

La normalisation du critère de continuité consiste à diviser l'expression du critère *contX<sub>i</sub>* par la valeur maximale atteignable par la variable  $X_i$ , notée  $Xmax_i$  - et qui devra donc être exprimée par ailleurs -. Les critères de continuité normalisés s'expriment alors par la relation suivante :

$$\text{cont}\tilde{X}_i = \left( \frac{1}{X_{\max,i}} \times \left[ X_i^{(z)}(\tau_z, X0_i^{(z)}) - X0_i^{(a)} \right] \right)^2 \quad (11)$$

qui ramène leur variation sur un intervalle [0 ; 1].

L'intérêt de cette normalisation apparaît au moment de l'affectation des contraintes sur ces critères. Il s'agit d'imposer ce critère à rester nul. La méthode que nous employons consiste à autoriser une tolérance sur cette contrainte, tout en introduisant le critère dans la fonction objectif de manière à ce qu'il soit minimisé (c'est une sorte de "relaxation lagrangienne" au sens de Niquil [NIQUIL97]). L'avantage de cette formulation est qu'elle améliore la convergence de l'algorithme (les contraintes fixes sur les critères sont fortement à éviter), et de plus, l'affectation de la tolérance est aisément ajustable par l'utilisateur, et identique pour toutes les variables d'état. Par exemple, affecter la contrainte sur l'intervalle [0 ; 0,01] donne une tolérance de 1% pour tous les critères de continuité.

Il est parfois possible d'obtenir une expression symbolique des  $X0_i^{(a)}$  : pour cela, il faut résoudre symboliquement les équations (12)

$$X_i^{(z)}(\tau_z, X0_i^{(z)}) - X0_i^{(a)} = 0 \quad (12)$$

qui, après substitutions de tous les  $X0_i^{(\alpha)}$ , peuvent s'exprimer en fonction des seuls  $X0_i^{(a)}$  :

$$X_i'(\tau_a, \tau_b, \tau_c, \dots, \tau_z, X0_i^{(a)}) - X0_i^{(a)} = 0 \quad (13)$$

Cependant, le souci de généralité qui sous-tend l'ensemble de l'approche impose la formulation (9). En effet, du fait de la complexité des expressions  $X_i'(\tau_a, \tau_b, \tau_c, \dots, \tau_z, X0_i^{(a)})$  - souvent émaillées de fonctions trigonométriques -, la résolution formelle des relations (13) n'est pas générale. Par contre la résolution numérique proposée dans la formulation (9) est toujours réalisable.

#### 4.2.1.2 Critères sur les durées des configurations

La formulation des critères afin de contraindre la continuité des variables d'état fait intervenir les durées des configurations qui deviennent par conséquent de nouveaux paramètres d'entrée du problème de dimensionnement.

Ces paramètres sont reliés par l'expression suivante qui définit la période de fonctionnement du convertisseur :

$$T = \sum_i \tau_i \quad (14)$$

Chaque durée de configuration est inférieure à la période. On peut donc normaliser ces durées en ramenant leur valeur à une fraction de la période. On note ici  $\tilde{\tau}_i$  les durées des configurations normalisées :

$$\tilde{\tau}_i = \frac{\tau_i}{T} \quad (15)$$

Ce sont ces paramètres qui seront considérés comme entrées du modèle de dimensionnement, et non les durées naturelles  $\tau_i$ . Ils sont facilement contraignables :

$$0 \leq \tilde{\tau}_i \leq 1 \quad (16)$$

et on peut obtenir une valeur initiale pour ces paramètres (à partir d'une simulation, cf. chap. III).

Les  $\tau_i$ , qui sont utilisés par ailleurs dans le modèle contraint pour formuler d'autres critères (cf. plus loin), et qui s'avèreraient trop difficiles à contraindre tels quels, sont définis à partir des durées normalisées :

$$\tau_i = \tilde{\tau}_i \times T \quad (17)$$

et ainsi, ils deviennent des paramètres internes transparents pour l'utilisateur.

Enfin, on préfère manipuler la fréquence, qui est davantage parlante pour le concepteur, plutôt que la période de fonctionnement, et donc on formule une relation supplémentaire :

$$T = \frac{1}{Freq} \quad (18)$$

A partir des entrées  $\tilde{\tau}_i$ , on formule le critère période normalisée suivant :

$$\tilde{T} = \sum_i \tilde{\tau}_i \quad (19)$$

qui doit être contraint à la valeur fixe 1, quel que soit le convertisseur étudié, ce qui assure sa généralité.

Les entrées apparues lors de cette formulation sont au nombre de  $n_c+1$  :

- les  $n_c$  durées normalisées  $\tilde{\tau}_i$  qui sont bien évidemment aussi nombreuses que les configurations du mode choisi pour le dimensionnement,
- la fréquence de fonctionnement du convertisseur  $Freq$ .

Les deux critères précédents (critères de continuité et critères de durée) permettent de garantir la cohérence des paramètres de dimensionnement internes lorsque le mode de fonctionnement est celui désiré. Cependant ces deux premiers critères n'influencent pas l'ordre d'enchaînement des configurations. C'est le rôle des deux types de critères suivants.

### 4.2.1.3 Critères sur les règles de commutation

Afin d'assurer l'enchaînement des configurations d'un mode dans un ordre donné de façon générique, on propose de contraindre les tests de commutation des interrupteurs à être obligatoirement vérifiés au moment des transitions.

Pour cela, on formule trois types de critères :

- *les critères de validation* qui vérifient que la commutation n'a pas eu lieu avant la transition de phase,
- *les critères de commutation* qui imposent la commutation à avoir lieu à la date précise de la transition,
- *les critères de maintien* qui assurent le maintien des conditions de commutation un temps suffisant après la transition (lorsque cela est nécessaire).

Ces critères sont formulés pour chaque interrupteur, qu'il change ou non d'état :

- *le blocage* : l'interrupteur passe de l'état passant à l'état ouvert,
- *l'amorçage* : l'interrupteur passe de l'état ouvert à l'état passant,
- *le non blocage* : l'interrupteur reste dans l'état passant,
- *le non amorçage* : l'interrupteur reste dans l'état ouvert,

et cela pour chaque interrupteur, à chaque transition de configuration.

On peut donc définir de façon générique les trois types de critères pour les quatre types de commutations, et cela pour chaque type d'interrupteur.

Alors, l'information générique pour déclarer ces critères est de deux ordres :

- une expression symbolique du test,
- la date à laquelle le test doit être valide.

On peut décomposer de façon générique l'information nécessaire pour décrire le test de commutation (cf. Figure II.1) : il s'agit d'un signal (courant ou tension) que l'on compare à une valeur seuil.

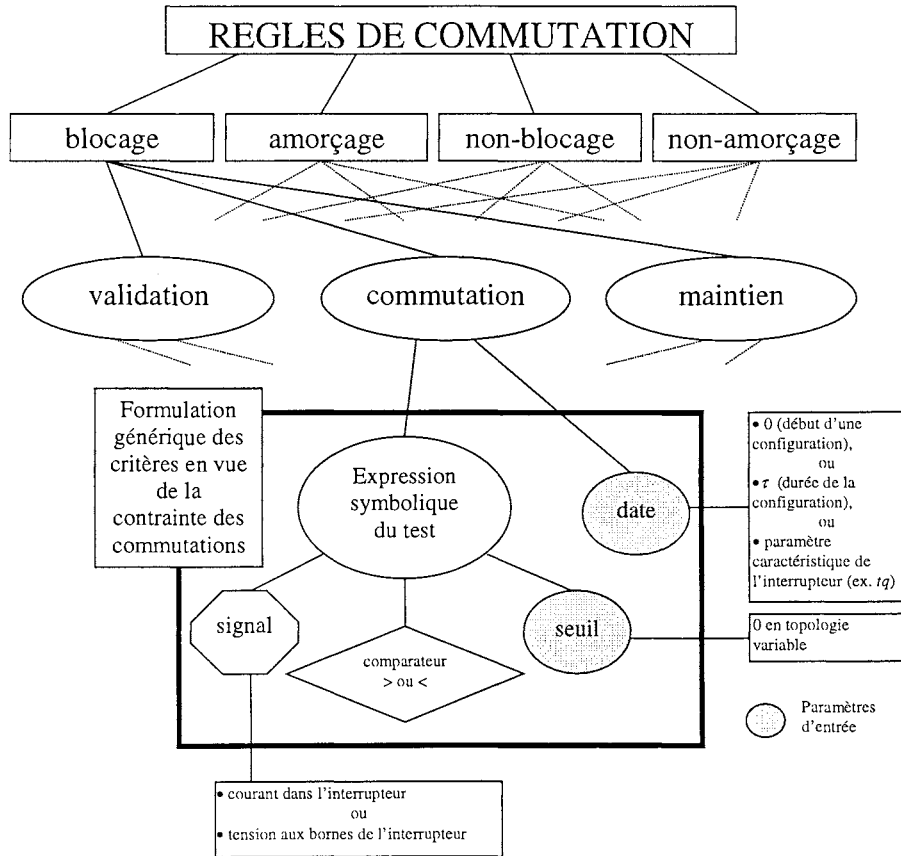


Figure II.1 : La formulation générique des critères sur les tests de commutation

Par exemple, le test de blocage d'une diode de la configuration C1 à la configuration C2 s'exprime par :

- le critère de validation  $i_D > 0$  à la fin de la configuration C1,
- le critère de commutation  $v_D < 0$  au début de la configuration C2,
- et le critère de maintien  $v_D < 0$  à la date  $tp$ .

En conclusion, nous avons proposé une formulation générique :

- des règles de commutation dans un cadre contraint,
- les critères définis à partir de ces règles sont formalisables génériquement sur la base des courants et tensions dans les interrupteurs, et des durées des configurations.

Afin de formuler les critères de commutation, on propose d'exprimer les grandeurs génériques suivantes :

$$\text{Test} = \left( \frac{\text{signal}(\text{date\_critere}) - \text{seuil}}{\text{valeur\_maximale\_du\_signal}} \right) \times \text{signe} \quad (20)$$

avec  $\text{signe} = +1$ , (respectivement  $-1$ ), lorsque le comparateur est  $\geq$ , (respectivement  $<$ ), de cette façon, quel que soit le critère considéré, sa grandeur associée est contrainte sur une plage  $[0 ; 1]$ .

#### 4.2.1.4 Critères faisant intervenir les pulsations

Parmi les critères formulés afin de contraindre les commutations des interrupteurs, on a défini le critère de validation. Son rôle est de vérifier que lors de la phase précédant la commutation, les conditions de déclenchement des interrupteurs n'ont pas lieu.

Le principe de formulation de ce critère consiste à contraindre le test de déclenchement à être non valide, cela à la fin de la configuration. Cette formulation n'est valable que si le signe du signal de déclenchement n'évolue pas pendant la durée de la configuration.

En fait, la formulation de la validation du test de commutation doit faire intervenir deux critères :

$$signal(t) < seuil \text{ ou } signal(t) > seuil \quad (21)$$

$$signe(signal(t)) = signe(signal(0)) \quad \forall t \in [0, \tau] \quad (22)$$

Afin de formuler le second critère sur le signe du signal pendant toute la durée  $\tau$  de la configuration, on exploite la forme particulière des signaux dans un convertisseur statique : ce sont des signaux périodiques. Donc possédant les expressions symboliques de ces signaux  $S$ , et moyennant un traitement adéquat de ces expressions, il est possible d'en extraire les pulsations  $\omega_s$  et donc les périodes  $T_s$ .

$$T_s = \frac{2\pi}{\omega_s} \quad (23)$$

Ensuite, nous proposons de formuler le critère normalisé  $\tilde{T}_s$  par rapport à la durée de la configuration : En général, les durées de configuration sont plus courtes que les périodes des signaux, alors nous ferons la normalisation inverse suivante :

$$\tilde{T}_s = \frac{\tau}{T_s} = \frac{\omega_s \cdot \tau}{2\pi} \quad (24)$$

Ainsi l'intervalle de variation de  $\tilde{T}_s$  est  $[0 ; 1]$ .

Il est alors possible de contraindre le signal à ne se propager que pendant une fraction de sa période pendant la durée d'une configuration, et ainsi s'assurer du signe de ce signal pendant toute la configuration.

Exemple : Dans le cas suivant (Figure II.2), on veut que le signal reste positif pendant toute la configuration (en grisé) de durée  $\tau$  :

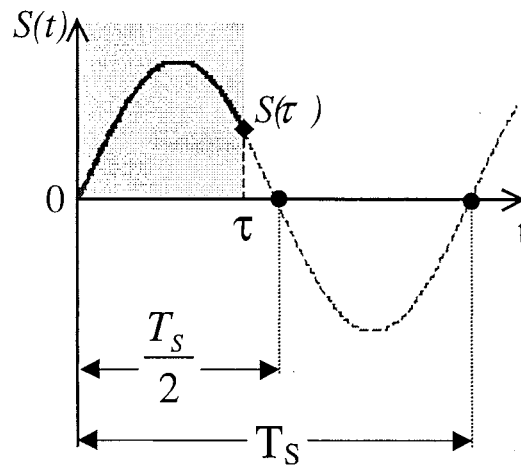


Figure II.2 : Un exemple de contrainte faisant intervenir la pulsation du signal

Ici, on a :

$$S(t) \geq 0 \forall t \in [0, \tau] \Leftrightarrow \begin{cases} S(\tau) \geq 0 \\ 0 \leq \tilde{T}_s \leq \frac{1}{2} \end{cases} \quad (25)$$

Donc pour contraindre le signe de ce signal pendant toute la durée de la configuration, il faudra contraindre deux critères normalisés sur deux intervalles génériques :

$\tilde{S}(\tau)$  sur l'intervalle  $[0 ; 1]$ , et

$\tilde{T}_s$  sur l'intervalle  $[0 ; \frac{1}{2}]$ .

Avec cette formulation du critère, il n'y a que la fraction de la période (ici  $\frac{1}{2}$ ) à déterminer pour ajuster la contrainte.

#### 4.2.2 Critères de dimensionnement

La viabilité de notre modèle de dimensionnement étant maintenant assurée par la formulation des critères physiques, nous pouvons maintenant nous focaliser sur la formulation des critères dont la vocation est de caractériser les dimensions de la structure : nous appellerons ces critères les critères de dimensionnement par opposition aux critères physiques.

Un grand nombre de critères peuvent servir à dimensionner un composant en électronique de puissance. Le choix de ces critères est réalisé au cas par cas par le concepteur en se fondant sur une pré-analyse du circuit (fonctionnement, stresses particuliers sur certains éléments).

Cependant, beaucoup de ces critères reposent sur des informations d'ordre électrique : à partir d'un nombre limité de caractéristiques électriques, toutes les dimensions des composants d'un convertisseur statique peuvent être calculées.

Donc connaissant les expressions symboliques temporelles des caractéristiques électriques fondamentales, toutes les dimensions peuvent être exprimées symboliquement. Bien sûr, il est hors



de question de formuler tous les critères de dimensionnement de tous les composants (pour des raisons de taille mémoire et de temps de calcul).

Donc on propose de ne formuler que les données électriques fondamentales, et de donner la possibilité à l'utilisateur/concepteur de formuler dans un environnement générique les critères particuliers qui l'intéressent pour chaque composant. Dans cette optique, nous avons d'abord défini neuf grandeurs qui représentent cette information électrique minimum :

- $Freq$  la fréquence de fonctionnement, qui intervient dans les effets de peau dans les conducteurs et l'effet de proximité, joue un grand rôle dans le calcul des pertes dans les composants passifs ;
- $(V_{max}, I_{max})$  les valeurs maximales des courants et tensions qui provoquent en particulier d'importants stresses sur les composants ;
- $(V_{moy}, I_{moy})$  les valeurs moyennes qui déterminent la gamme de puissance des composants ;
- $(V_{eff}, I_{eff})$  les valeurs efficaces qui déterminent la gamme de puissance des composants , et qui participent aux calculs des pertes dans les composants ;
- les  $\frac{dV}{dt}$  et  $\frac{dI}{dt}$  qui caractérisent le pouvoir de commutation et le pouvoir de coupure des interrupteurs.

Ensuite, nous avons formalisé de façon symbolique les critères correspondants du modèle de dimensionnement.

#### **4.2.2.1 Formulation du critère de fréquence**

On considère la fréquence de fonctionnement du circuit comme un paramètre d'entrée du problème de dimensionnement (cf. § 4.2.1.2).

Cependant du point de vue de l'optimisation, ce paramètre, dans l'outil final de dimensionnement, peut être contraint de la même façon qu'un critère.

#### **4.2.2.2 Formulation générale des critères de valeurs maximales**

Il faut garder à l'esprit que pendant le dimensionnement, l'ensemble des paramètres est amené à varier plus ou moins fortement, ce qui a une influence sur le fonctionnement du convertisseur et par conséquent sur l'allure des courbes représentatives des courants et tensions dans les composants.

En particulier, la valeur maximale d'un signal, initialement observée dans une configuration, peut apparaître, après le dimensionnement, dans une autre configuration (cf. Figure II.3 et Figure II.4).

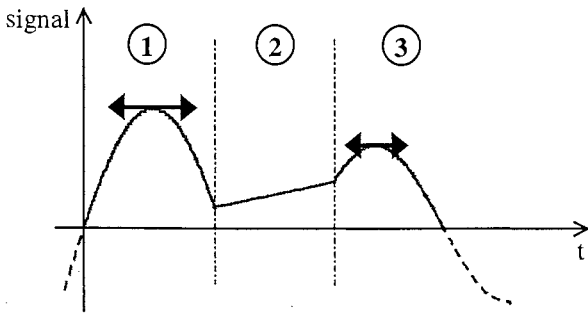


Figure II.3 : Point de fonctionnement initial, le maximum a lieu pendant la phase 1

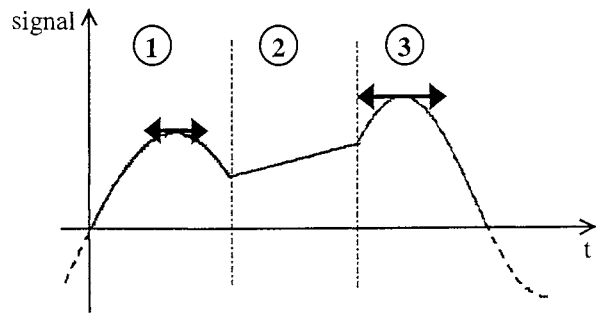


Figure II.4 : Fonctionnement après dimensionnement, le maximum a lieu pendant la phase 3

Par conséquent, il est impossible de formuler la valeur maximale dans un mode de fonctionnement de façon figée sur une configuration donnée.

Il est nécessaire de calculer les extremums partiels  $S_{ext\ i}$  de chaque signal  $S$  sur chaque configuration  $i$ , puis de formuler la valeur maximale d'ensemble  $S_{max}$  comme le maximum de tous les extremums partiels.

#### 4.2.2.1 Formulation des valeurs maximales d'ensemble

$$S_{max} = \max(S_{ext\ i}) \quad (26)$$

Pour formuler ce critère, nous avons utilisé l'expression (27) définissant analytiquement le maximum de deux nombres :

$$\max(a, b) = \frac{1}{2} (|a + b| + |a - b|) \quad (27)$$

que nous avons généralisée en programmant une fonction récursive afin d'obtenir l'expression symbolique du maximum dans une liste de variables quelconques.

Exemple : Dans cet exemple, on cherche l'expression symbolique du maximum de trois variables  $a$ ,  $b$  et  $c$ . On écrit la récursivité :

$$\begin{aligned} \max(a, b, c) &= \max(a, \max(b, c)) \\ &= \max\left(a, \frac{1}{2} (|b + c| + |b - c|)\right) \\ &= \frac{1}{2} \left( \left| a + \frac{1}{2} (|b + c| + |b - c|) \right| + \left| a - \frac{1}{2} (|b + c| + |b - c|) \right| \right) \end{aligned}$$

On peut de cette façon obtenir l'expression symbolique du maximum d'une liste quelconque de variables.

#### 4.2.2.2 Formulation des extremums partiels

Formuler un extremum partiel consiste en fait à exprimer l'expression symbolique du signal  $S$  (tension ou courant dans un composant) à la date où se produit l'extremum.

Donc la formulation d'un extremum partiel repose sur l'expression de deux grandeurs :

$t_{ext}$  la date de l'extremum,

$S(t_{ext})$  la valeur du signal  $S$  à la date de l'extremum.

Comme le reste des paramètres de dimensionnement, la date  $t_{ext}$  est susceptible d'évoluer pendant le dimensionnement.

D'ailleurs la contrainte physique sur ce paramètre est triviale :

$$0 \leq t_{ext} \leq \tau \quad (28)$$

$\tau$  étant la durée de la configuration.

On préférera manipuler le paramètre normalisé que l'on peut contraindre sur un intervalle unitaire :

$$0 \leq \tilde{t}_{ext} \leq 1 \quad (29)$$

avec

$$\tilde{t}_{ext} = \frac{t_{ext}}{\tau} \quad (30)$$

La relation pour déterminer le paramètre  $t_{ext}$  fait intervenir la dérivée du signal à cette date :

$$\frac{dS}{dt}(t_{ext}) = 0 \quad (31)$$

Cette relation n'est pas toujours directement résoluble de façon symbolique à cause des fonctions trigonométriques qui émaillent les expressions des signaux.

Donc, dans un souci de généralité, on propose de formuler le critère

$$\hat{S} = \frac{dS}{dt}(t_{ext}) \quad (32)$$

que l'on normalise de la façon suivante :

$$\tilde{\hat{S}} = \frac{dS}{dt}(t_{ext}) \times t_{ext} \quad (33)$$

Théoriquement, ces critères devraient être contraints à zéro pendant le processus de dimensionnement, contraignant ainsi indirectement la date  $t_{ext}$  à pointer sur la date de l'extremum partiel. Cependant, un trop grand nombre de contraintes sur les critères de sortie imposées à des valeurs fixes, altère fortement la convergence de l'algorithme d'optimisation. C'est même une cause de l'échec de l'optimisation (problème sur-contraint [BAKKER93]). Or les critères que nous venons de formuler peuvent être nombreux. Il faut donc contraindre ces critères sur un intervalle de

variations admissibles, qui n'altère pas significativement la valeur de l'extremum correspondant. Cette tolérance dépend de la variation du signal : plus cette variation est rapide, plus la tolérance doit être restreinte. C'est ici que la normalisation est utile, car elle permet d'homogénéiser l'amplitude des variations des signaux, et ainsi de contraindre les critères dans un même intervalle de tolérance générique.

En fait, cette formulation consiste à résoudre numériquement la relation (31) p.44 par l'algorithme d'optimisation pendant le processus de dimensionnement.

Cette formulation reste valable lorsque la date de l'extremum est figée à une valeur connue. Par exemple dans le cas d'une évolution monotone du signal, la date du maximum est 0 pour une décroissance,  $\tau$  pour une croissance du signal. Cependant la contrainte sur  $\tilde{S}$  ne doit pas être imposée.

La limite de cette formulation concerne les cas où plusieurs dates d'extremums locaux coexistent pour une même configuration. Nous proposons d'étendre la notion de maximum des extremums à l'intérieur des configurations et de contraindre automatiquement le début et la fin de la configuration comme date possible d'un extremum, sans le critère sur la pente du signal.

Ce domaine de la formulation des critères sur les extremums des signaux est encore un vaste champ d'investigation. Nos réflexions portent maintenant sur le moyen de maîtriser les cas d'apparitions ou de disparitions intempestives de certains des extremums pendant le dimensionnement. Ce problème que nous traitons actuellement par l'analyse ressemble beaucoup à un problème d'optimisation, alors pourquoi ne pas le traiter avec l'algorithme mis à notre disposition...

#### 4.2.2.3 Formulation des critères des valeurs moyennes

Nous formulons le critère de valeur moyenne d'un signal en partant de sa définition

$$S_{moy} = \frac{1}{T} \sum_{config} \int_0^{\tau} S(t) dt \quad (34)$$

en posant  $S(t) = \sum_i \alpha_i \cdot X_i(t) + B(t)$ , dans le cas où le signal n'est pas une variable d'état,

$$\int_0^{\tau} S(t) dt = \alpha_i \sum_i \int_0^{\tau} X_i(t) dt + \int_0^{\tau} B(t) dt \quad (35)$$

De cette façon, il est possible de formuler le critère de valeur moyenne uniquement en fonction des variables d'état.

#### 4.2.2.4 Formulation des critères des valeurs efficaces

Comme précédemment, nous formulons le critère de valeur efficace d'un signal en partant de sa définition :

$$S_{eff} = \sqrt{\frac{1}{T} \sum_{config} \int_0^{\tau} S^2(t) dt} \quad (36)$$

en posant  $S(t) = \sum_i \alpha_i \cdot X_i(t) + B(t)$ , dans le cas où le signal n'est pas une variable d'état,

$$\int_0^{\tau} S^2(t) dt = \sum_i \left[ \alpha_i^2 \int_0^{\tau} X_i(t)^2 dt + 2\alpha_i \int_0^{\tau} B(t) \cdot X_i(t) dt + 2\alpha_i \sum_{j < i} \alpha_j \int_0^{\tau} X_i(t) \cdot X_j(t) dt \right] + \int_0^{\tau} B(t)^2 dt \quad (37)$$

De cette façon, il est possible de formuler le critère de valeur efficace uniquement en fonction des variables d'état.

#### 4.2.2.5 Formulation des critères $\frac{dV}{dt}$ et $\frac{dI}{dt}$

A partir des expressions symboliques temporelles des signaux, on peut réaliser leur dérivation par rapport au temps. Cette information est utile pour le dimensionnement de certains composants.

On formule donc les critères, qui sont la pente du signal à une date  $t_{critique}$  jugée critique par le concepteur :

$$penteS = \frac{dS}{dt} (t_{critique}) \quad (38)$$

$t_{critique}$  est un paramètre d'entrée du problème de dimensionnement, ainsi ce critère  $penteS$  peut être calculé pour n'importe quelle date.

En pratique, on utilisera le paramètre normalisé  $\tilde{t}_{critique}$  afin d'être plus générique

$$\tilde{t}_{critique} = \frac{t_{critique}}{\tau_{config}} \quad (39)$$

Ces derniers critères ne sont pas formulés automatiquement dans le modèle actuel, mais toute l'information nécessaire pour une formulation manuelle est d'ores et déjà accessible.

#### 4.2.2.6 Formulation des critères définis par l'utilisateur

A partir des critères précédemment définis, il est maintenant possible de contraindre toutes les dimensions des composants.

Pour cela, il faut formuler de nouveaux critères spécifiques à la structure étudiée et aux choix du concepteur.

**EX1 : Dimensions géométriques d'une inductance**

Par exemple, dans le cadre du dimensionnement d'une inductance, il est possible d'obtenir le nombre de spires et la section des conducteurs pour un noyau magnétique donné [TOURKHANI96] :

$$N = \frac{L \cdot I_{\max}}{B_{\max} \cdot A_e} \tag{40}$$

$$s_c = \frac{\beta \cdot S_b}{N} \tag{41}$$

Avec les paramètres géométriques suivants :

- |   |   |
|---|---|
| $\beta$ : facteur de remplissage de la fenêtre du noyau | $B_{\max}$ : induction maximale dans le noyau |
| $s_c$ : section des conducteurs                         | $S_b$ : section de la fenêtre                 |
| $N$ : nombre de spires                                  | $A_e$ : section effective du noyau            |

**EX2 : Pertes fer / Pertes cuivre**

Une expression empirique des pertes Fer et Cuivre fait apparaître *Freq* la fréquence de fonctionnement du convertisseur à dimensionner :

$$P_{Fer} = 11 \cdot \left( \frac{Freq}{10 \cdot K} \right)^{1,3} \cdot \left( \frac{B_m}{0,1} \right)^2 \tag{42}$$

$$P_{Cu} = R_a \cdot I_L^2 \tag{43}$$

Où  $R_a$  est la résistance en courant alternatif.

Par rapport à sa valeur constante en continu  $R_c = \frac{\rho \cdot l}{S}$ ,  $R_a$  augmente avec la fréquence suivant l'effet de peau et l'effet de proximité. S'il existe des formules d'approximation pour évaluer l'effet de peau (cf. 43a et 43b), l'effet de proximité reste plus difficile à quantifier.

Formule de Levasseur pour les fils cylindriques :

$$\frac{R_a}{R_c} = 0,25 + \sqrt[6]{0,18 + \left( \frac{S}{p_e \cdot \delta_p} \right)^6} \tag{43a}$$

Avec  $S$  section du conducteur,  $p_e$  périmètre du conducteur, et  $\delta_p$  épaisseur de peau.

Formule de Levasseur pour les conducteurs méplats :

$$\frac{R_a}{R_c} = \sqrt[8]{1 + \left( \frac{a}{2 \cdot \delta_p} \right)^8} \tag{43b}$$

Avec  $a$  petit côté du méplat,  $b$  grand côté,  $a \ll b$ .

C'est ici que les composants écartés initialement par les simplifications effectuées au début du processus pourront être réintégrés au problème de dimensionnement, en formulant les critères externes adéquats (fondés sur des besoins en terme de filtrage par exemple).

### 4.3 La fonction objectif

La fonction objectif du problème de dimensionnement des convertisseurs statiques telle que nous l'avons appréhendée doit prendre en compte la coexistence des aspects présentés plus haut :

- l'aspect physique de cohérence entre le modèle de fonctionnement et le modèle de dimensionnement
- l'aspect dimensionnement proprement dit, où l'utilisateur formule ses propres critères. C'est à ce niveau que la notion de coût financier peut être introduite. Si actuellement elle est à la charge de l'utilisateur à partir de formules empiriques, cette information pourrait être formulée génériquement, en utilisant une bibliothèque de fonctions par exemple.

Pour cela, on propose de formuler la fonction objectif multicritère comme la somme de deux fonctions auxquelles on affecte un facteur de pondération.

$$fobj = w_{phys} \cdot f_{phys} + w_{dim} \cdot f_{dim} \quad (44)$$

La partie physique s'applique à assurer la convergence des critères de continuité. Elle est donc formulable automatiquement elle-même comme fonction pondérée.

$$f_{phys} = \frac{1}{n_{CC}} \sum_{i=1}^{n_{CC}} contX_i \quad (45)$$

$n_{CC}$  étant le nombre de critères de continuité  $contX_i$  formulés automatiquement dans le modèle de dimensionnement. Cette formulation simple est possible grâce à la normalisation des critères. On peut ainsi affecter, dans cette fonction pondérée, un poids identique à chaque critère  $contX_i$ .

La partie dimensionnement est laissée à la liberté de l'utilisateur/concepteur. Là aussi plusieurs critères peuvent entrer en jeu, et une formulation comme fonction pondérée semble requise.

Deux types de critères sont susceptibles d'être implantés dans "la fonction objectif de dimensionnement" : des critères à minimiser, et des critères à maximiser.

Les premiers seront implantés tels quels dans la fonction objectif de dimensionnement avec le poids correspondant  $poids_i$  :

$$poids_i \times critere_i \quad (46)$$

Pour les seconds critères à maximiser, on implantera leur inverse, car la fonction objectif finale est minimisée par l'algorithme d'optimisation

$$poids_j / critere_j \quad (47)$$

Les facteurs de pondérations  $poids_i$ ,  $poids_j$ ,  $w_{phys}$  et  $w_{dim}$  seront des entrées du problème de dimensionnement d'une part pour donner plus ou moins d'importance à l'aspect physique ou à l'aspect dimensionnement, et d'autre part pour insister sur certains critères plutôt que sur d'autres.

La mise au point des facteurs de pondération dans une fonction multicritère est toujours très délicate :

### Effet horizon

L'effet horizon (cf. chap. I § 4.2.1, [AMBROSIO94], [HAWORTH93]) décrit le problème de validité de l'optimum calculé et de sa corrélation avec les poids affectés lorsque la traduction du problème multicritère est faite à l'aide de facteurs de pondération. La solution préconisée par les deux auteurs référencés ci-dessus (preference-directed-design) est de ne pas figer les valeurs de ces poids, mais de donner leur ordre d'importance relative (leur préférence) que l'on peut exprimer sous forme de contraintes inégalité. Pour l'instant nous n'avons pas mis en œuvre ce procédé qui complexifierait considérablement le modèle.

### Problème d'hétérogénéité

Les valeurs des critères peuvent être d'ordre très différents, ils ne sont pas tous normalisés - les critères utilisateur - et donc leur influence naturelle sur la fonction objectif reproduit ces écarts. La première difficulté consiste donc à harmoniser la fonction objectif en jouant sur les poids. Il est donc fortement conseillé à l'utilisateur, dans la mesure du possible, de normaliser ses critères.

### Problème d'incohérence

Certains des critères implantés dans la fonction objectif peuvent s'avérer antinomiques. Leurs influences respectives sur le dimensionnement peuvent s'opposer (ainsi par exemple augmenter la puissance et minimiser les pertes).

Dans ce cas-là, le dimensionnement final est un compromis entre tous ces critères, et si cela est acceptable lorsqu'il s'agit de critères de dimensionnement entre eux, cela devient catastrophique lorsque l'on touche à l'intégrité des critères physiques : ceux-ci doivent toujours être vérifiés sous peine de mise en défaut du modèle. Toute la difficulté à ce niveau consiste donc à garder un poids suffisant aux critères physiques de manière à assurer la pertinence du modèle de dimensionnement, d'une part, et toutefois de ne pas surévaluer l'influence de ces critères pour pouvoir tenir compte des besoins de l'utilisateur, d'autre part.

On atteint ici les limites de la formulation de la fonction multicritère, qui peut se traduire par une non convergence de l'algorithme. Tant que l'on contraindra le problème de cohérence



fonctionnement/dimensionnement au même niveau que le reste des critères de dimensionnement, ce problème subsistera à moins de pouvoir minimiser indépendamment les deux fonctions objectif physique et dimensionnement. A ce jour, à notre connaissance, il n'existe pas d'algorithme d'optimisation pouvant réaliser ce travail.

Une solution plus envisageable consiste à ne plus contraindre de la même façon le problème de dimensionnement et le problème de cohérence physique.

Dans ce but, des travaux sont en cours pour ajouter à l'outil d'optimisation PASCOSMA un étage *pré-optimisation* de résolution numérique d'équations.

C'est à cet étage que les équations constituant la partie physique du problème pourront être résolues numériquement : il s'agit du calcul des zéros des fonctions.

Ainsi, les critères physiques n'apparaîtront plus dans le problème de dimensionnement et la fonction objectif ne fera plus intervenir que les critères utilisateur autorisant un compromis.

#### **4.4 Les entrées du dimensionnement**

Les paramètres d'entrée du dimensionnement sont les variables indépendantes du modèle contraint. Le nombre de ces paramètres donne la dimension de l'espace de recherche du problème d'optimisation dual. Donc, afin d'aider la convergence de l'algorithme d'optimisation, on cherche à minimiser le nombre des entrées dans la formulation du modèle de dimensionnement.

Nous avons divisé l'ensemble des paramètres d'entrée du dimensionnement en deux groupes distincts :

- le groupe des entrées primaires,
- le groupe des entrées secondaires.

Les entrées primaires sont les paramètres qui émaillent les équations de fonctionnement du convertisseur statique. Elles apparaissent dans les systèmes d'état du mode de fonctionnement. Ce sont typiquement les inductances  $L$ , les capacités  $C$ , les résistances  $R$ , les coefficients de transformation  $N$ , les sources constantes  $E$ ,  $I$ ...

Les entrées secondaires sont les paramètres engendrés par la formulation des critères.

Elles comportent :

- la fréquence,
- les durées des configurations,

- les dates des crêtes,
- les durées caractéristiques des semi-conducteurs,
- les paramètres des critères utilisateur,
- les facteurs de pondération de la fonction objectif.

C'est le nombre des entrées secondaires qui est susceptible d'être minimisé, puisque le nombre des entrées primaires est donné dès le choix de la structure et du fonctionnement. Cette minimisation est réalisée pour les conditions initiales (cf. §4.2.1.1.3).

Beaucoup des entrées secondaires ne sont pas imposées directement par l'utilisateur pour définir le point de fonctionnement de départ. Ces grandeurs ne peuvent qu'être estimées indirectement à partir de la simulation du point de fonctionnement, (par exemple les durées des configurations et les dates des crêtes). Leur domaine de validité et leurs influences sur les autres paramètres du dimensionnement sont difficilement contrôlables, aussi est-il très difficile de contraindre ces paramètres tels quels. Nous avons donc cherché à construire pour ces grandeurs non parlantes pour l'utilisateur des paramètres d'entrée normalisés de façon à ce qu'ils aient une plage de variation limitée à  $[0 ; 1]$ .

Du fait de la normalisation, certains des paramètres secondaires peuvent être contraints automatiquement sur  $[0 ; 1]$ , c'est le cas pour les durées normalisées des configurations et les dates normalisées des crêtes.

Les paramètres primaires et les paramètres secondaires doivent être initialisés (par un jeu de valeurs physiquement cohérent) avant le lancement d'une optimisation. Cependant, un point de fonctionnement ne permet d'initialiser que les entrées primaires du problème de dimensionnement. Les entrées secondaires ne peuvent être connues qu'après l'analyse du fonctionnement du convertisseur pour le point de fonctionnement donné. Aussi nous proposons de réaliser l'initialisation des entrées secondaires à partir d'une simulation (cf. chap. III).

#### **4.5 Formulation des gradients**

Dans notre hypothèse du choix d'un algorithme du gradient pour résoudre notre problème d'optimisation issu du problème de dimensionnement, il est indispensable de pouvoir calculer les

gradients, c'est à dire les dérivées de tous les critères en fonction de chaque paramètre d'entrée. Nous avons décidé de ne pas formuler les expressions symboliques directes des gradients, obtenues par substitution [WURTZ96], mais de procéder par composition (cf. Chap. III). Pour matérialiser l'information nécessaire pour évaluer les gradients calculés de cette façon, deux éléments sont implantés dans le modèle de dimensionnement : l'arbre de dépendance des paramètres (des critères, des entrées, et des paramètres intermédiaires), et les dérivées partielles de toutes les expressions symboliques en fonction des paramètres dont elles dépendent directement (cf. Chap. III).

## **4.6 Formulation du cahier des charges**

### **4.6.1 Définition**

Le cahier des charges est formulé vis-à-vis des dimensions des composants. Dans le cadre du problème d'optimisation du problème de conception, on appellera cahier des charges l'ensemble des contraintes sur les critères et les paramètres du modèle de dimensionnement.

Formuler le cahier des charges du problème d'optimisation consiste donc en pratique à donner les bornes de la plage de variation acceptable pour les valeurs de chaque critère et chaque paramètres d'entrée.

### **4.6.2 Les différentes contraintes**

Deux sortes de contraintes peuvent être définies :

- une plage continue, définie par ses bornes supérieure et inférieure,
- une valeur fixe.

Tout paramètre - indifféremment critère ou paramètre d'entrée - peut être contraint de cette façon.

### **4.6.3 L'affectation des contraintes**

L'affectation des contraintes est faite de manière interactive par l'utilisateur au moyen de l'interface que propose l'outil de dimensionnement final (PASCOMA). De cette manière la modification du cahier des charges pendant la session de dimensionnement est aisée :

- contrainte sur une plage,
- contrainte fixe,
- libération de contrainte (uniquement pour les critères).

## 5 Critique du modèle de dimensionnement

### 5.1 Modèle utilisé pour formuler les critères

#### 5.1.1 Equations de base

La formulation des critères de dimensionnement d'un convertisseur statique repose sur des retraitements symboliques appropriés sur les équations de fonctionnement de ce convertisseur statique.

Le choix du modèle de fonctionnement sous-tend donc l'ensemble du modèle de dimensionnement proposé. Ainsi, en particulier, le niveau de finesse atteint pour le dimensionnement est conditionné par celui du modèle de fonctionnement.

Dans notre gamme de problèmes, on n'envisage pas un dimensionnement microscopique des composants qui irait par exemple pour un interrupteur jusqu'aux paramètres micro-électroniques (comme la largeur de canal, ...).

Aussi un modèle macroscopique du comportement des interrupteurs nous satisfait à ce niveau.

Nous avons choisi d'utiliser un modèle discret idéal pour représenter les semi-conducteurs (topologie variable). Dans ce type de modèles, les interrupteurs sont représentés par un court-circuit à l'état passant, et un circuit ouvert à l'état bloqué. Chacune des  $2^n$  configurations possibles provoquées par l'ouverture et la fermeture des  $n$  interrupteurs est décrite par un système d'état différent qui définit le comportement des variables d'état du circuit pendant cette configuration. Ces systèmes d'état sont des systèmes d'équations différentielles linéaires du premier ordre de la forme :

$$\begin{aligned} u(P,t) \\ \frac{dX^{(i)}}{dt} = A^{(i)}(P) \cdot X^{(i)}(P,t) + B^{(i)}(u,t) = A^{(i)}(P) \cdot X^{(i)}(P,t) + B^{(i)}(P,t) \\ Y^{(i)} = C^{(i)}(P) \cdot X^{(i)}(P,t) + C^{(i)}(u,t) = C^{(i)}(P) \cdot X^{(i)}(P,t) + D^{(i)}(P,t) \end{aligned} \quad (48)$$

où  $0 \leq t \leq \tau_i$ , chaque  $\tau_i$  étant la durée de la  $i^{\text{ème}}$  configuration. On définit l'origine des temps  $t = 0$  au début de chaque configuration.

Les variables d'état  $X^{(i)}$  pour chaque configuration  $i$  sont les courants dans les inductances et les tensions aux bornes des condensateurs du circuit.

Les paramètres  $P$  du système incluent :

- les valeurs des composants passifs du circuit électrique du convertisseur (inductances  $L$ , condensateurs  $C$ , résistances  $R$ , coefficients des transformateurs parfaits  $N \dots$ ),
- les éléments caractéristiques des sources de courant et tension (valeur maximale et pulsation par exemple),
- les conditions initiales  $X_0$  pour les variables d'état  $X$ .

Les variables  $Y^{(i)}$  sont les courants et tensions autres que variables d'état (tensions aux bornes des inductances, courants dans les condensateurs, courant et tension dans les interrupteurs, dans les résistances, ...).

Les caractéristiques des systèmes d'état issus des modèles à topologie variable (linéarité, ordre 1, simplicité) font que ce genre de modèles est utilisé classiquement dans les approches symboliques. En particulier, la linéarité des systèmes d'état permet d'utiliser les méthodes symboliques d'intégration des systèmes d'équations différentielles de type Laplace et exponentielles de matrices. De plus, les modèles à topologie variable ne font pas intervenir de paramètres "non physiques" comme dans les modèles à topologie fixe – avec résistance de faible valeur à l'état passant et résistance élevée à l'état ouvert –. Ces paramètres sont difficiles à estimer sans connaissance précise des composants, mais surtout, du fait de leur caractère grossier, ils n'apportent rien en regard des autres phénomènes à considérer d'autre part.

### 5.1.2 Discussion sur la pertinence du modèle à topologie variable pour fonder un modèle de dimensionnement des convertisseurs statiques.

Les modèles à topologie variable sont des modèles comportementaux grossiers des interrupteurs, et occultent de nombreux phénomènes physiques tels que la chute de tension à l'état passant, le seuil d'amorçage. Si l'influence de ces phénomènes physiques sur le fonctionnement des convertisseurs statiques peut être négligée dans un état permanent, elle peut par contre s'avérer prépondérante dans les états transitoires où ils provoquent des pics de tensions et d'intensité. Comme après toute modélisation, la vigilance doit accompagner l'emploi de notre méthodologie afin de s'assurer que le niveau de finesse du modèle de fonctionnement est suffisant pour fonder les critères de dimensionnement que nous proposons.

Quelle que soit leur sévérité, les critères qui forceront le mode seront toujours issus du modèle de base à topologie variable, et donc sujets à caution malgré toutes les précautions prises. Là encore, la vigilance et l'analyse des résultats obtenus sont nécessaires. Une "validation" du niveau de modélisation par rapport à une simulation fine est donc requise afin de se conforter dans les choix faits.

## 5.2 Problèmes de construction du modèle de dimensionnement

Deux problèmes viennent contrarier la construction du modèle de dimensionnement tel que nous l'avons décrit précédemment :

- l'intégration symbolique des systèmes d'état,
- la traduction en un problème d'optimisation

### 5.2.1 Problème de l'intégration symbolique des systèmes d'état

Toute la formulation du modèle de dimensionnement repose sur les expressions temporelles symboliques des variables d'état. Et pour obtenir ces expressions, il faut intégrer symboliquement les systèmes d'état des configurations du mode.

Du fait de la linéarité de ces systèmes différentiels, on peut utiliser des méthodes classiques d'intégration formelle du type exponentielles de matrice ou transformation de Laplace mais celles-ci ne sont pas infaillibles.

#### 5.2.1.1 *Les raisons de l'échec de l'intégration symbolique des systèmes d'état*

L'intégration d'un système d'équations différentielles, quelle que soit la méthode employée (exponentielles de matrices ou transformation de Laplace) revient à la détermination des racines d'un polynôme (valeurs propres du polynôme caractéristique ou pôles du dénominateur des transformées de Laplace). Le degré de ce polynôme est donné par la dimension du système différentiel.

Notre problème physique d'intégration des systèmes d'état bute sur un problème mathématique très étudié : la factorisation des polynômes d'ordre élevé.

Il a été démontré que les racines d'un polynôme de degré inférieur ou égal à quatre peuvent toujours être calculées symboliquement sous forme radicale, donc un système différentiel d'ordre inférieur ou égal à quatre peut théoriquement être intégré symboliquement. Par contre, on démontre aussi que

pour un ordre supérieur à quatre, un polynôme ne peut pas être factorisé sous une forme radicale générique. On ne sait donc pas intégrer systématiquement de manière classique un système différentiel d'ordre supérieur à quatre, à moins d'en factoriser le polynôme caractéristique en  $p$  polynômes de degré inférieur ou égal à quatre. Cette factorisation est parfois naturelle lorsque les variables d'état du circuit sont physiquement "découplées".

$$\begin{cases} \frac{dX_i}{dt} = A' \cdot X_i + B' \\ \frac{dX_j}{dt} = A'' \cdot X_j + B'' \end{cases} \text{ avec } X = (X_i, X_j) \quad (49)$$

$n$  étant la dimension du système d'état à intégrer.

On peut alors résoudre séparément l'intégration des deux sous-systèmes d'état ( $i$ ) et ( $j$ ), ce qui montre comment des systèmes différentiels d'ordre supérieur à quatre peuvent être intégrés. Cependant ces découplages naturels sont rares, d'autant plus que, dans les circuits susceptibles d'être étudiés, interviennent des phénomènes de résonances synonymes de fortes interactions entre toutes les variables d'état.

En fait, par le développement des convertisseurs multi-résonants qui, jusqu'alors réservés aux faibles puissances, tendent à s'imposer dans toutes les applications, les choix technologiques modernes vont plutôt dans le sens d'un "durcissement" des systèmes d'état, (i.e. de fortes interactions entre les équations d'état).

### 5.2.1.2 Solutions possibles

Nous proposons trois types de solutions pour dépasser ce problème [LANIER97] :

#### 5.2.1.2.1 Choisir une autre méthode d'intégration

La méthode d'intégration procédant par développement de Taylor n'est pas limitée par la taille des systèmes d'équations différentielles, et dans ce sens, elle semble plus générique.

Cependant, l'emploi de cette méthode poserait d'autres problèmes :

- La solution fournie est un développement limité à l'origine et donc sa validité est restreinte au rayon de convergence de la série. Ce rayon de convergence n'est pas maîtrisable génériquement. Pour formuler le modèle de dimensionnement tel que nous le proposons, il est nécessaire de disposer des expressions temporelles exactes sur un intervalle de temps de l'ordre d'une période.

- De plus, des retraitements symboliques sont opérés sur ces expressions : intégrations et dérivations. Comment les réaliser sur les développements limités ? L'utilisation des développements de Taylor pour intégrer symboliquement les systèmes d'état du modèle de fonctionnement n'est pas compatible avec le modèle de dimensionnement que nous proposons. Elle supposerait en fait une complète reformulation du problème et beaucoup de questions restent encore en suspens dans cette optique.

### 5.2.1.2.2 Employer une approche mixant numérique et formel

Une idée que nous avons mise en œuvre avec le concours du Laboratoire de Modélisation et de Calcul (LMC) dans le cadre du D.E.A. de David Lanier [LANIER97] coencadré avec G. Villard, M. Barkatou et L. Gerbaud, propose de ne pas chercher à réaliser la factorisation du polynôme de façon symbolique, mais de la remettre "à plus tard", où une méthode numérique classique sera utilisable, et en utilisant "entre-temps" dans la formulation symbolique du modèle de dimensionnement les expressions faisant intervenir les symboles "vides", appelés *rootofs*, des racines de ces polynômes : leur expression symbolique n'est pas explicitement formulée. Bien sûr, en utilisant cette technique, aucun système d'équations différentielles linéaires ne peut théoriquement résister à l'intégration, et surtout, nous nous sommes aperçus que pratiquement tout le modèle de dimensionnement était formulable, aussi bien les critères que les gradients. Les retraitements nécessaires pour fonder le modèle contraint sur ce principe, sont exposés dans le chapitre IV.

Cependant une information est perdue lorsque l'on formule les expressions temporelles des signaux avec ces symboles vides : Elles ne sont plus écrites sous une forme exploitable aisément, en particulier les pulsations ne peuvent plus en être extraites. Les critères utilisant cette information ne peuvent donc plus être formulés.

D'autre part, la taille et la complexité des expressions "semi-formelles" qui découlent de cette méthode ont demandé beaucoup de remaniements dans la méthode de traduction du modèle de dimensionnement en problème d'optimisation (cf. chap. IV), et montrent encore des problèmes de gestion de la taille des formules pour le calcul des valeurs efficaces.

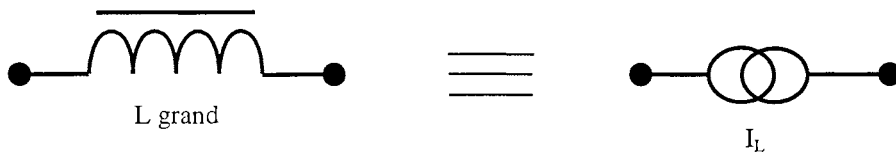
Les performances de cette méthode, en terme de précision notamment, restent à tester.

### 5.2.1.2.3 Découpler les variables d'état

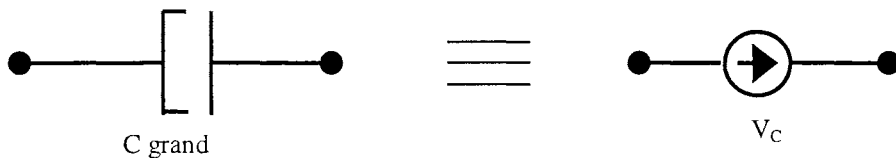
Afin d'obtenir des sous-systèmes d'état intégrables avec les méthodes d'intégration classiques, plusieurs méthodes permettent de réaliser de tels découplages.



- Des modifications de topologie, qui consistent à faire des hypothèses simplificatrices sur les dimensions de certains composants passifs du circuit du convertisseur. Ainsi, par exemple si on suppose que la valeur d'une inductance est très grande (inductance de filtrage), cela implique que  $\frac{di_L}{dt}$  est très faible par rapport aux périodes considérées dans le convertisseur. On peut donc approximer le courant qui traverse cette inductance par un courant constant, d'où la simplification de la topologie du circuit qui consiste à remplacer l'inductance par une source de courant constant.



De même pour une capacité de forte valeur



- Au niveau des systèmes d'état, on remplace la ligne correspondante dans le système d'équations différentielles

$$\frac{di_L}{dt} = f(i_L(t)) \quad / \quad \frac{dv_C}{dt} = g(v_C(t)) \quad (50)$$

par

$$i_L = I_L \text{ (constante)} \quad / \quad v_C = V_C \text{ (constante)} \quad (51)$$

qui induit un découplage trivial dans le système

$$\left\{ \begin{array}{l} i_L = I_L \\ \frac{dX_j}{dt} = A'' \cdot X_j + B'' \text{ pour } 2 < j < n \end{array} \right. / \left\{ \begin{array}{l} v_C = V_C \\ \frac{dX_j}{dt} = A'' \cdot X_j + B'' \text{ pour } 2 < j < n \end{array} \right. \quad (52)$$

Cette manipulation revient en fait à enlever un degré au système d'état.

Ces simplifications peuvent être opérées sur l'ensemble des configurations du mode de fonctionnement, ou seulement pour quelques-unes (celles qui posent effectivement le problème d'intégration).

*Inconvénients de ces simplifications* : Les composants sur lesquels on impose les hypothèses restrictives disparaissent du problème de dimensionnement, et une reformulation des contraintes est à reprendre séparément par rapport aux valeurs moyennes à considérer.

### 5.2.2 Problème de la traduction en un problème d'optimisation

La traduction du modèle de dimensionnement en un problème d'optimisation passe par la formulation des gradients, et puis par une phase de codage des expressions symboliques des critères et gradients dans un langage de programmation (cf. chap. IV).

Or la formulation du modèle telle que nous l'avons présentée, fait apparaître une structure fortement imbriquée des expressions symboliques des critères. Aussi, nous proposons de tirer profit de cette architecture du modèle de dimensionnement, pour améliorer l'efficacité de la programmation : en ordonnant les critères dans l'ordre chronologique de leur évaluation, une programmation par composition est rendue possible. Ceci, en supprimant les redites de code, permet d'envisager un accroissement notable de la rapidité des calculs, et une diminution non moins importante de l'espace mémoire occupée par le programme d'analyse. De plus, la méthode de composition est très avantageuse pour calculer les gradients : à partir du calcul de quelques dérivées partielles, tous les gradients peuvent être reconstruits par multiplications et additions de ces expressions, en suivant un arbre de composition à définir (Chap. III). L'efficacité concerne ici la programmation du programme de calcul des sensibilités, en supprimant encore une fois, mais de façon encore plus spectaculaire, les redites de code.

### 5.3 Limites intrinsèques du modèle

Du fait des choix qui ont été faits pour poser les fondations et formuler le modèle de dimensionnement que nous avons mis en œuvre, les limites intrinsèques de ce modèle sont connues :

- Le niveau de finesse du modèle de dimensionnement est celui du modèle de fonctionnement à topologie variable : macroscopique. De ce fait, le domaine de validité du dimensionnement est restreint aux phénomènes électriques (il ne peut pas prendre en compte les phénomènes de C.E.M. par exemple).
- La dimension des systèmes d'état est limitée par la contrainte de l'intégration formelle des systèmes d'état. Mais on peut espérer soulever cette limitation par l'emploi de la méthode

semi-numérique que nous proposons au §5.2.1.2.2. Cependant, du fait de la finesse déjà limitée du modèle, il est inutile de travailler sur des circuits de convertisseurs trop complexes, puisque cela n'améliore plus significativement la précision du dimensionnement. Une simplification des circuits est toujours souhaitable lorsqu'elle est possible, de façon à manipuler des systèmes d'état réduits de taille raisonnable ( $< 10$ ).

- Le dimensionnement est réalisé pour un mode de fonctionnement figé. Dans le cas d'un fonctionnement "multi-modal" du convertisseur (cas des convertisseurs multi-résonants), c'est dans le mode de fonctionnement initial que le dimensionnement est réalisé. Ceci peut être un handicap si rien n'impose le choix d'un mode plutôt qu'un autre. Par contre, cela est un avantage quand un ou plusieurs modes doivent absolument être écartés, car alors, le modèle "multi-modal" de dimensionnement peut être obtenu par une duplication des modèles particuliers correspondants à chaque mode acceptable. Ce travail ne consiste pas en une simple juxtaposition puisque tous les modèles partagent des données communes, et ont des contraintes communes sur des données différentes.

- Les tolérances sur les dimensions des composants ne sont pas gérées à ce niveau de modèle. Le dimensionnement est fourni à une valeur donnée de tous les paramètres. Cependant on désire parfois obtenir un dimensionnement valable, sur un intervalle de valeurs pour certains paramètres (paramètres à tolérances).

Pour dépasser cette limite - similaire en terme de problématique à la précédente -, nous proposons deux solutions :

- Une solution au niveau du modèle de dimensionnement qui met en œuvre une nouvelle duplication. Nous proposons de diviser les intervalles de valeurs acceptables pour les paramètres à tolérance en  $n$  paramètres discrets, et ensuite formuler  $n$  fois les critères de dimensionnement qui font intervenir le paramètre en question.

Par exemple, voici une discrétisation en cinq valeurs pour une tolérance de 10% sur  $P$  :  $P - P/10, P - P/20, P, P + P/20, P + P/10,$ )

- Une solution hors-modèle, fondée sur des rebouclages opérés sur l'outil final de dimensionnement, que nous présentons au Chapitre V.

## **6 Conclusion**

Dans ce chapitre, nous présentons quels éléments composent notre modèle de dimensionnement générique (critères, entrées, fonction objectif, cahier des charges), et comment nous proposons de les formuler symboliquement pour qu'il soit possible de générer le modèle automatiquement, quel que soit le problème de dimensionnement, pour toute structure, dans n'importe quel mode de fonctionnement à l'état permanent figé. Pour cela, nous scindons l'ensemble des critères en deux groupes : les critères physiques qui assurent la cohérence du modèle avec le fonctionnement, et les critères de dimensionnement à partir desquels le concepteur peut formuler ses propres critères spécifiques ; et nous formulons une fonction objectif multicritère qui prend en compte la cohérence entre le dimensionnement et le fonctionnement du convertisseur étudié. Enfin nous faisons une critique du modèle de dimensionnement ainsi construit.

L'information sur laquelle repose l'ensemble du modèle de dimensionnement étant maintenant définie, nous pouvons mettre en œuvre la méthodologie qui nous permet de construire ce dernier. C'est cette méthodologie que nous présentons dans le chapitre III suivant.



*Chapitre III :*  
*Notre méthodologie de*  
*dimensionnement des convertisseurs*  
*statiques*



# CHAPITRE III : NOTRE METHODOLOGIE DE DIMENSIONNEMENT DES CONVERTISSEURS STATIQUES

Afin d'intégrer outils, méthodes et données nécessaires pour formuler génériquement notre problème de dimensionnement et le résoudre comme un problème d'optimisation, la démarche que nous proposons procède en six étapes. Leur agencement est présenté dans la Figure III.1 ci-dessous.

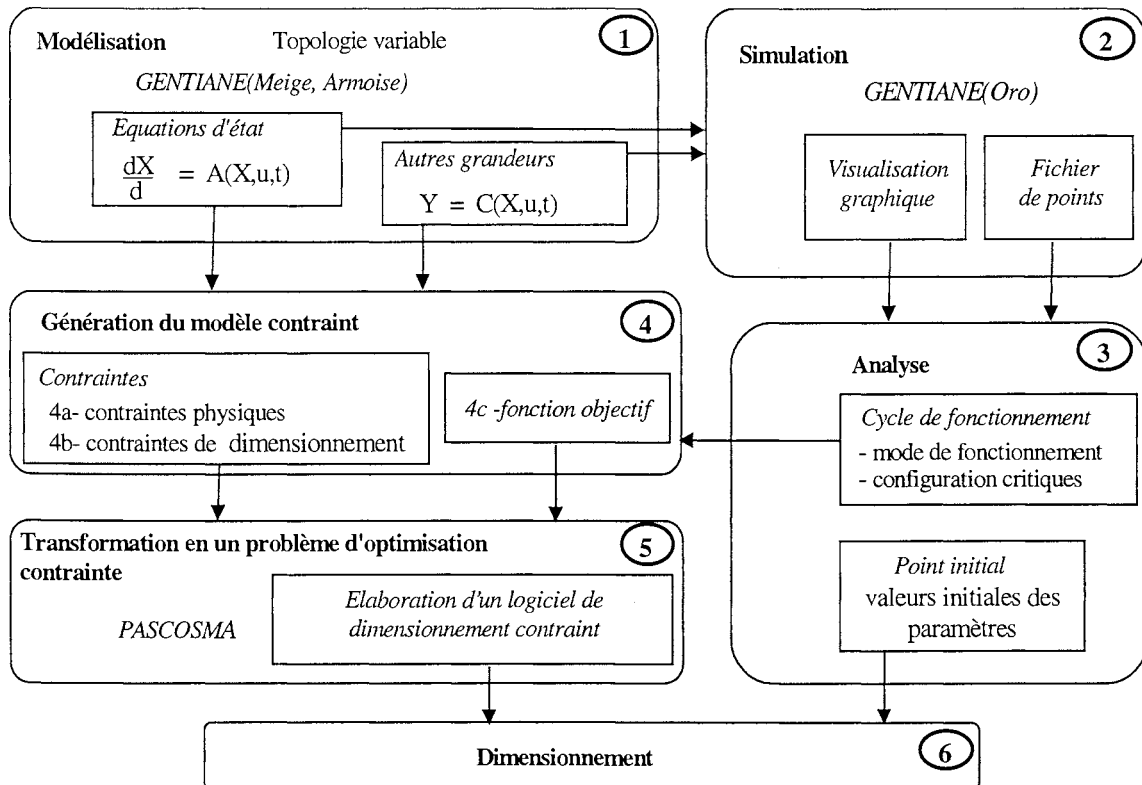


Figure III.1 : Les six étapes de notre approche



## 1 *Enchaînement des étapes* (Figure III.1)

Tout commence par la description électrique du circuit du convertisseur de l'étude : il suffit de fournir les équations des nœuds de ce circuit.

A partir de ces équations, la première étape consiste à **(1) construire automatiquement le modèle** à topologie variable **du fonctionnement** du convertisseur. Ce modèle est à la base de la formulation du modèle de dimensionnement. Mais pour réaliser cette formation, des informations sur le fonctionnement effectif du convertisseur sont nécessaires (mode, dates critiques, phénomènes critiques – crêtes -, ...). Donc ensuite il faut faire une **simulation (2)** du fonctionnement du convertisseur à partir de ce modèle. Puis une **analyse (3)** de cette simulation sont préalables à la **formulation du modèle de dimensionnement (4)**. Ensuite, on peut réaliser la **traduction du modèle de dimensionnement en un problème d'optimisation contrainte (5)**, qui consiste à élaborer le logiciel dédié de dimensionnement contraint. La dernière étape consiste à utiliser ce **logiciel de dimensionnement (6)**, auquel on doit fournir un jeu de valeurs de départ pour les paramètres d'entrées.

Ce jeu de valeurs est en grande partie donné par le point de fonctionnement initial, mais certains paramètres d'entrées (les entrées secondaires, cf. chap. II, § 4), artificiellement apparus pendant la formulation du modèle de dimensionnement (durées des configurations, date des valeurs crêtes ...), doivent être initialisés autrement : l'analyse de la simulation effectuée précédemment (3) peut jouer ce rôle.

## 2 *Présentation des 6 étapes*

D'après leur enchaînement, on peut maintenant spécifier les caractéristiques de toutes les étapes de la méthodologie : leur rôle, l'information dont elles ont besoin, l'information qu'elles fournissent et les méthodes et outils qu'elles utilisent.

### 2.1 **ETAPE 1 : Modélisation**

#### 2.1.1 **Rôle**

C'est l'étape initiale de la méthodologie de dimensionnement qui consiste à construire le modèle de fonctionnement du convertisseur étudié, modèle de base pour la simulation et l'élaboration du modèle de dimensionnement.

### 2.1.2 Entrées

A ce stade, le mode de fonctionnement du convertisseur n'est pas nécessairement connu. Aussi, pour pouvoir parer à tous les cas de figure, c'est une modélisation sans a priori sur le fonctionnement du convertisseur statique que nous réalisons. La seule information qui soit donc nécessaire est la description du circuit : cette information est entièrement donnée par les équations des nœuds du circuit du convertisseur, et par la déclaration de la nature des composants.

### 2.1.3 Sorties

Le modèle de fonctionnement construit par cette étape est un modèle à topologie variable. Celui-ci fournit un système d'état pour chaque configuration viable du circuit, les règles d'enchaînements entre ces configurations et les expressions de toutes les grandeurs (autres que variables d'état) en fonction des variables d'état.

#### **Propriétés des systèmes d'état :**

Les systèmes d'état sont les équations différentielles vérifiées par les variables d'état (courants dans les inductances et tensions dans les capacités) pour chaque configuration. Ces systèmes différentiels contiennent donc toute l'information sur le fonctionnement du convertisseur pendant chaque configuration : c'est pourquoi ils sont à la base de la méthodologie de dimensionnement que nous proposons.

Les paramètres qui émaillent ces équations sont les paramètres primaires (cf. chap. II, § 4) du dimensionnement (on trouve les inductances  $L$ , les capacités  $C$ , les résistances  $R$ , les valeurs maximales des sources de tension  $E$ , les rapports de transformation  $N$  ...).

Les propriétés fondamentales de ces systèmes pour la suite de l'approche sont :

- leur linéarité due au choix de la topologie variable,
- leur simplicité : cette étape de modélisation passe par une phase de réduction des systèmes d'état, ce qui se caractérise par des dimensions variables de ces systèmes d'état d'une configuration à l'autre.

## 2.1.4 Méthode

Cette étape est entièrement automatisée. Elle est fondée sur une approche formelle qui réalise le travail suivant :

- la construction des équations des mailles du circuit à partir des équations des nœuds,
- la recherche des configurations électriquement viables, fondée sur des règles d'électrotechnique (complémentarité des commutations, viabilité des équations ...),
- la formulation des équations différentielles du circuit pour chaque configuration,
- la réduction des systèmes d'état ainsi formulés,
- la programmation des tests de commutation qui régissent le passage d'une configuration à l'autre,
- et la formulation des expressions des courants et tensions dans tous les composants, pour chaque configuration, en fonction des variables d'état (qui sont les courants dans les inductances et les tensions dans les capacités).

## 2.2 ETAPE 2 : Simulation

### 2.2.1 Rôle

Nous nous plaçons toujours dans l'hypothèse la plus générale où le fonctionnement du convertisseur n'est pas connu a priori. Dans ce cadre, cette étape de simulation est nécessaire pour réaliser une analyse qualitative du point de fonctionnement initial qui consiste, pour le concepteur :

- A déterminer le mode pour ce point de fonctionnement : c'est dans ce mode qui sera figé, que tout le dimensionnement sera effectué par la suite,
- A opérer une expertise du fonctionnement afin d'en dégager les points critiques : par exemple la valeur crête d'un courant ou d'une tension qu'il faudra chercher à restreindre. Certains critères du dimensionnement peuvent être dégagés pendant cette étape,
- Et à visualiser l'allure des courbes pour détecter d'éventuelles simplifications possibles.

### 2.2.2 Entrées

Par hypothèse, les seules entrées disponibles à ce stade de la méthodologie sont les données du point de fonctionnement initial, le mode de fonctionnement n'est généralement pas encore connu. Issu de l'étape 1, le modèle à topologie variable complet de la structure étudiée est aussi à notre disposition.

### 2.2.3 Sorties

Cette étape fournit en sortie :

- Une information d'ordre qualitatif pour élaborer les critères. Par exemple la séquence des configurations qui forme le mode de fonctionnement, ou les configurations et les dates critiques pour lesquelles les signaux atteignent des extremums.
- Une information d'ordre quantitatif qui servira à initialiser indirectement, après l'étape 3 d'analyse, le processus de dimensionnement. Il s'agit du fichier de points calculés pour chaque signal (courant et tension) de chaque composant sur au moins une période de fonctionnement du convertisseur, à l'état permanent.

### 2.2.4 Méthode

- C'est une simulation numérique, car à ce stade de la méthodologie, on n'a pas encore réalisé l'intégration symbolique des systèmes d'état.
- C'est une simulation sans a priori, car le mode de fonctionnement n'est pas connu.

Plusieurs simulateurs sans a priori sont à notre disposition (SPICE, SABER, CIRCUIT, SUCCESS...). Ils permettent tous une étude qualitative similaire du fonctionnement de la structure étudiée. Le seul inconvénient de l'utilisation de ces outils est la nécessité de décrire à nouveau la structure du convertisseur, car ils n'utilisent pas le même modèle que celui utilisé pour le dimensionnement.

Cet inconvénient n'est pas majeur en ce qui concerne l'étude qualitative.

Cependant, il est indispensable de réaliser au moins une simulation du modèle à topologie variable issu de GENTIANE-MEIGE (dans notre cas), sur lequel l'ensemble du modèle de dimensionnement est fondé, avec l'approche GENTIANE (cf. Figure III.2) [GERBAUD97].

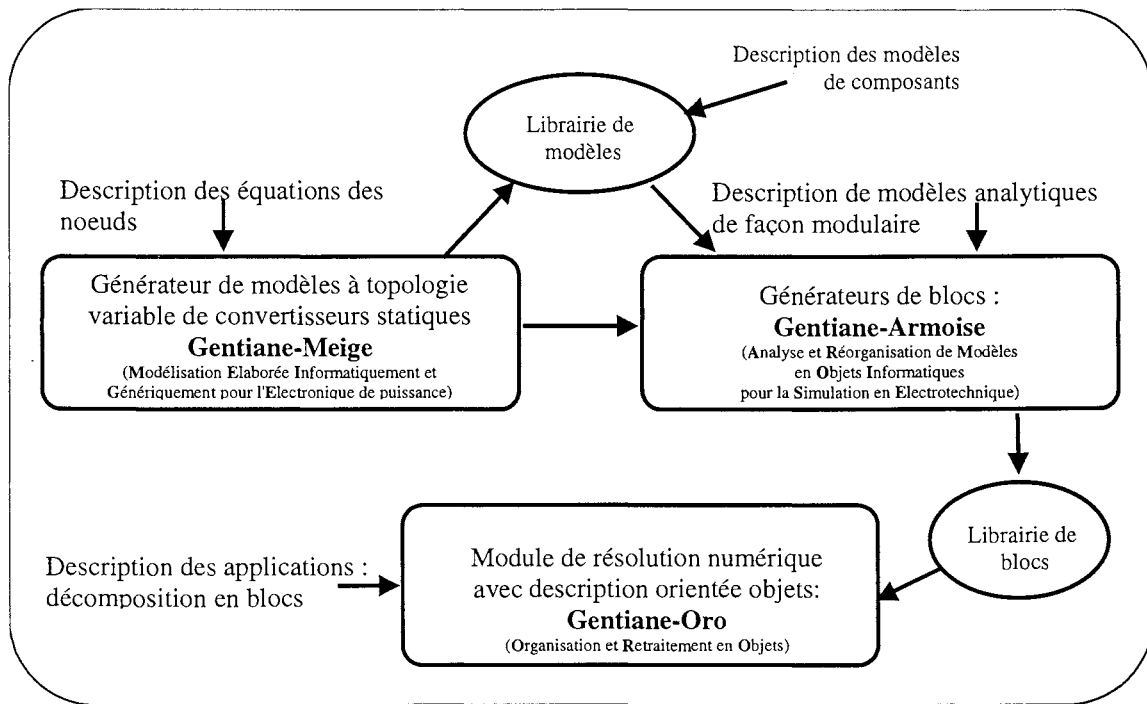


Figure III.2 : L'approche GENTIANE pour la modélisation des entraînements électromécaniques

L'utilisateur peut réaliser cette simulation numérique avec le module GENTIANE-ORO qui utilise directement le modèle de fonctionnement fourni par GENTIANE-MEIGE sans intervention extérieure. L'intérêt de cette simulation est double :

- Dans un premier temps, cette simulation permet de vérifier la validité du modèle généré par GENTIANE-MEIGE, par comparaison avec des simulations réalisées avec d'autres outils et des modèles plus fins, et ainsi, en particulier, d'analyser l'influence des hypothèses simplificatrices.
- Dans un deuxième temps, cette simulation GENTIANE-ORO, réalisée sur au moins une période de fonctionnement à l'état permanent, est nécessaire pour mettre en œuvre l'analyse quantitative qui sert à initialiser l'algorithme d'optimisation utilisé dans l'étape 6. Pour cette initialisation, certaines données du point de fonctionnement sont nécessaires : en particulier les valeurs des variables d'état au moment des transitions de phases, et les durées des configurations, qui interviennent dans la formulation des critères de continuité des variables d'état (cf. chap. II, § 2.3.1). Il faut que les valeurs de tous les paramètres d'entrée soient suffisamment précises pour que le mode de fonctionnement simulé pour le modèle à topologie variable GENTIANE-MEIGE fondateur du dimensionnement soit dans un état permanent. Si les valeurs saisies à ce niveau ne sont pas suffisamment précises, cela se traduit en terme de contrainte, par des violations importantes de certains critères (les critères de continuité en particulier, qui sont très sensibles aux paramètres d'entrée cf. chap. V, § 3.2.3.4.2). Partir d'un tel point de départ où des

contraintes sont fortement violées, - on dira que le point initial n'est pas viable physiquement -, altère fortement, et peut même faire échouer la convergence de l'algorithme d'optimisation.

Nous avons appelé *besoin de cohérence entre le modèle de dimensionnement et le modèle de fonctionnement* cette nécessité de simuler le fonctionnement de la structure à l'état permanent avec le modèle à topologie variable de GENTIANE-MEIGE, afin de pouvoir initialiser correctement l'algorithme d'optimisation.

Sachant que le modèle simulé servira de fondation au modèle de dimensionnement, nous savons d'ores et déjà que nous aurons à notre disposition les expressions symboliques de tous les signaux du circuit, aussi nous envisageons de mettre en œuvre une « *simulation symbolique* » de la structure, s'appuyant sur les travaux entrepris sur les méthodes d'intégration des systèmes d'équations différentielles mixant techniques numériques et formelles que nous présentons au chapitre IV. Ceci éliminerait les imprécisions dues aux intégrations numériques, et assurerait une cohérence sans faille entre le modèle de dimensionnement et le modèle de fonctionnement.

Enfin, il faut garder à l'esprit que dans l'approche que nous proposons, cette étape de simulation numérique est sortie du processus de dimensionnement à proprement dite, ce n'est pas l'outil d'analyse de résolution du problème direct, qui lui est fondée sur des évaluations de relations symboliques.

### 2.3 ETAPE 3 : Analyse de la simulation

#### 2.3.1 Rôle

Cette étape a un rôle double :

- D'une part, cette étape sert à extraire l'information nécessaire pour formuler le modèle de dimensionnement du fichier de points issu de la simulation.

Cette information contient :

- l'identification des configurations présentes dans le mode de fonctionnement, et l'ordre d'enchaînement de ces configurations,
- les configurations critiques pendant lesquelles les signaux passent par des extremums.

- D'autre part, cette étape est nécessaire pour initialiser le logiciel d'optimisation contrainte avec les valeurs des entrées secondaires qui apparaissent lors de la formulation du modèle de dimensionnement.

Ces paramètres secondaires sont :

- les durées des configurations,
- les dates des extremums,
- la fréquence de fonctionnement du convertisseur,
- les conditions initiales pour chaque variable d'état au début d'un cycle du mode retenu.

### 2.3.2 Entrées

Toutes les données nécessaires à l'exécution de cette étape sont contenues dans le fichier de points obtenu après la simulation, à condition que celle-ci ait été effectuée sur au moins une période de fonctionnement du convertisseur, et à l'état permanent. Bien sûr, tous les signaux auront été simulés et stockés dans le fichier.

### 2.3.3 Sorties

Cette analyse de la simulation donne une expertise quantitative cette fois-ci pour l'étape suivante de formulation du modèle de dimensionnement :

- les configurations sont identifiées par un numéro générique (GENTIANE-MEIGE),
- pour chaque signal les configurations critiques sont caractérisées par un marqueur entier qui prend ses valeurs dans l'ensemble  $\{-1, 0, 1\}$ 
  - 1 si pendant cette configuration, le signal passe par un extremum négatif,
  - +1 si pendant cette configuration, le signal passe par un extremum positif,
  - 0 si aucun extremum remarquable n'est atteint pendant cette configuration.
- cette étape fournit aussi un fichier de valeurs contenant les valeurs initiales des paramètres secondaires pour l'étape finale de dimensionnement.

### 2.3.4 Méthode

Toutes ces valeurs sont extraites par des analyses du tableau de données issues de la simulation. Il s'agit de processus itératifs procédant par comparaisons, scrutations et extractions.

## 2.4 ETAPE 4 : Génération du modèle de dimensionnement

### 2.4.1 Rôle

L'étape de génération du modèle de dimensionnement consiste à formuler l'outil d'analyse nécessaire pour la résolution du problème inverse, et à mettre en place l'information nécessaire pour construire l'outil d'exploration de l'espace de recherche (ici un algorithme d'optimisation).

### 2.4.2 Entrées

L'information nécessaire pour la mise en œuvre de cette étape est de trois ordres :

- Le fonctionnement du convertisseur.

Cette information est contenue d'une part dans les systèmes d'état, et les expressions de tous les signaux fournis par le modèle à topologie variable issu de GENTIANE-MEIGE (étape 1), et l'enchaînement des configurations qui constituent le mode fourni par l'analyse qualitative de l'étape 3.

- Les données génériques de construction du modèle de dimensionnement, qui sont les paramètres d'entrée secondaires présentés dans le Chapitre II, § 4.3 (durées des configurations, dates des extremums, conditions initiales des variables d'état...).

- Les besoins du concepteur, qui se manifestent par des critères de dimensionnement à formuler. Ces besoins proviennent d'une analyse d'expert qui peut être issue de l'analyse de la simulation (§ 2.3) ou de l'expérience extérieure du concepteur.

Ces critères peuvent être implantés au coup par coup par le concepteur, ou issus de bibliothèques contenant les équations qui relient les dimensions avec les expressions de base pour des ensembles de composants souvent utilisés (par exemple des transformateurs).

### 2.4.3 Sorties

A la fin de cette étape, le modèle de dimensionnement est en place pour sa future traduction en problème d'optimisation. Il contient :

- Les données génériques pour générer chaque instruction du programme d'analyse du problème inverse que nous détaillons dans le paragraphe 2.4.4 ci-dessous,
- l'information générique nécessaire pour générer chaque instruction du programme de calcul des sensibilités, indispensable pour programmer l'outil d'optimisation (cf. § 2.4.4).



#### 2.4.4 Méthode

La méthode que nous proposons pour formuler de façon générique ce modèle de dimensionnement, est exposée dans le chapitre II. Elle est fondée sur un nombre fini de retraitements symboliques des variables d'état (des intégrations et des dérivations) et sur une décomposition générique de l'information nécessaire pour générer chaque instruction des programmes d'analyse et de calcul des sensibilités.

La méthode ainsi définie se veut automatisable, les retraitements symboliques automatisés présentés ici sont détaillés dans le chapitre IV qui expose l'implantation informatique de l'approche.

- Les retraitements symboliques réalisés pendant cette étape sont :
  - l'intégration des systèmes d'état pour obtenir les expressions symboliques des  $X(t)$ ,
  - les calculs des  $Z_x = \left\{ \frac{dX}{dt}, \int X dt, \int X_i X_j dt \right\}$ ,
  - les calculs des  $Z_y = \left\{ Y, \frac{dY}{dt}, \int Y dt, \int Y^2 dt \right\}$  à partir des résultats précédents,
  - les calculs des dérivées partielles des  $\frac{\partial Z_x}{\partial E}$  et  $\frac{\partial Z_y}{\partial E}$  par rapport aux paramètres d'entrée,
  - la formulation des différents critères, et de leurs gradients en fonction des paramètres d'entrée.

Le programme d'analyse consiste à affecter les critères en fonction des paramètres d'entrée, des paramètres internes et des autres critères, dans l'ordre ascendant de l'arbre de composition.

- La structuration de l'information nécessaire pour formuler les critères est :
  - l'identificateur du critère (son nom), qui est construit automatiquement sur la base d'une convention préétablie
  - l'expression symbolique à utiliser dans l'ensemble  $Z_x$  ou  $Z_y$ ,
  - la formule du critère en fonction des expressions symboliques précédentes,
  - la date où s'exprime le critère.

Les trois derniers points de cette information étant explicitement contenus dans la formulation des critères que nous avons présentée dans le chapitre II.

Ainsi, à partir de cette information, nous pouvons générer le programme d'analyse, qui consiste à réaliser les affectations des critères en fonction des paramètres d'entrée (primaires et secondaires, cf. chap. II, § 4.3), des paramètres internes (cf. chap. II, § 4.1.1.1.3), et des autres critères.

- La structuration de l'information nécessaire pour formuler les gradients est définie ci-dessous.
  - Elle contient l'information pour formuler les dérivées partielles :
    - le nom du critère, issu d'une convention,
    - les paramètres dont dépend ce critère, information contenue dans son expression symbolique,
    - les expressions symboliques à utiliser pour calculer les dérivées partielles, que nous stockons de façon adéquate dans des variables nommées avec une convention préétablie.

Ainsi, tout le processus de formulation des dérivées partielles peut être automatisé.

- La structuration de l'information contient aussi les éléments pour réaliser la composition :
  - les dépendances de chaque critère du modèle de dimensionnement par rapport à l'ensemble des paramètres (primaires, secondaires et critères), ce qui permet de construire un arbre de dépendance automatiquement lorsqu'on lui a fourni toutes les expressions symboliques ;
  - les expressions symboliques des critères non substitués, soit que nous avons formulées automatiquement pour les critères physiques, et les critères de base (valeurs moyennes, valeurs efficaces, valeurs maximales), soit qui ont été fournies par le concepteur, et qui doivent être écrites en fonction des critères de base.

A partir de ces deux informations, nous calculons automatiquement les gradients (c'est à dire les  $\frac{dC}{dE}$ ) par composition, ou par substitution comme nous le montrons dans les paragraphes § 2.5.3.1 et § 2.5.3.2 ci-dessous.

## 2.5 ETAPE 5 : Traduction en un problème d'optimisation

### 2.5.1 Rôle

Cette cinquième étape réalise la génération de l'outil d'optimisation dédié à la résolution du problème de dimensionnement spécifique étudié.

### 2.5.2 Entrées

L'information d'entrée de cette étape est le modèle de dimensionnement, c'est à dire :

- l'information structurée pour générer le programme d'analyse présentée au § 2.4,
- l'information structurée pour générer le programme d'optimisation il s'agit de l'information pour générer les gradients (présentée au § 2.4).

### 2.5.3 Méthode

Deux méthodes sont possibles dans cette étape pour réaliser le travail :

#### 2.5.3.1 Substitutions

Faire une génération des programmes d'analyse et de calcul des sensibilités par substitution consiste à réécrire les expressions des critères et gradients en fonction des seuls paramètres d'entrée du problème de dimensionnement.

Cette tâche est entièrement automatisée et prise en charge par l'outil de génération du logiciel d'optimisation, actuel (PASCOSMA v. 1 [WURTZ96a]). Cependant cette méthode n'est plus optimale pour notre gamme de problèmes qui font intervenir des expressions complexes et imbriquées.

- D'une part, cette méthode de calcul des critères et des gradients fait apparaître beaucoup trop de "redites" pendant l'évaluation, comme l'illustre l'exemple ci-dessous.

Exemple :

$$\begin{aligned} C_1 &= C_1(A, B) \\ \text{Soient deux critères :} \\ C_2 &= C_2(C_1, B, D)' \end{aligned}$$

A, B, D sont des paramètres d'entrée indépendants.

La méthode de calcul par substitutions passe d'abord par la phase de substitution

$$\begin{aligned} C_1 &= C_1(A, B) \\ C_2 &= C_2(A, B, D)' \end{aligned}$$

puis, la génération du code de calcul correspondant est ensuite opérée (idem et ceci est même plus critique pour les gradients).

On voit bien que cela revient à programmer deux fois le calcul de  $C_1(A, B)$ .

- D'autre part, cette méthode pose rapidement un problème de taille du code généré.

En conséquence, c'est l'efficacité du programme de résolution du problème inverse qui est affectée : les temps de calcul sont ralentis par les redites, et la taille du programme est inutilement grande.

#### 2.5.3.2 Composition

Pour cette raison, une génération des programmes d'analyse et de calcul des sensibilités par composition est préférée pour nos problèmes de dimensionnement. Cela consiste à ne plus réécrire

les critères en fonction des paramètres d'entrée, mais de les générer tels quels, en fonction des paramètres dont ils dépendent directement.

Exemple : Pour l'exemple des critères C1 et C2 présenté plus haut :  
 $C_1 = C_1(A, B)$   
 $C_2 = C_2(C_1, B, D)$   
l'arbre de composition peut être schématisé par la Figure III.3 ci-contre.

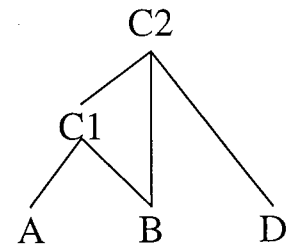


Figure III.3 : Un exemple d'arbre de composition

Le seul problème consiste à s'assurer de l'ordre dans lequel sont programmés les critères pour que leur évaluation successive soit possible.

Cet ordre est donné par le sens ascendant dans l'arbre de dépendance des critères.

Exemple : ici C<sub>1</sub> doit être programmé avant C<sub>2</sub>

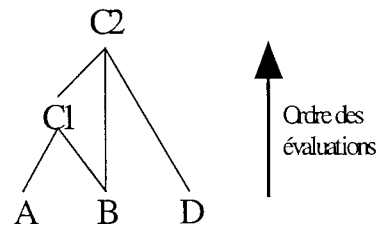


Figure III.4 : Un exemple de progression dans un arbre de composition

Cet arbre de dépendance est donc une information indispensable pour la réalisation de l'étape de traduction en un problème d'optimisation.

- Toutes les dépendances des paramètres du dimensionnement entre eux sont contenues dans les expressions symboliques des critères, et peuvent donc être automatiquement retrouvées par un retraitement adéquat par Macsyma.
- En ce qui concerne l'ordre de programmation des évaluations dans le programme d'analyse par composition que nous générons automatiquement, c'est l'ordre dans lequel nous formulons les critères dans le modèle de dimensionnement qui assurent une bonne progression dans l'arbre de composition. Cet ordre est connu pour les critères automatiques (critères de fonctionnement, et critères de dimensionnement, cf. chap. IV, § 3.2.2.3).

Que ce soit par substitution ou par composition, la construction du programme de résolution du problème inverse consiste

- premièrement, à traduire les expressions symboliques des critères et gradients en un code exécutable : traduction en un langage programmable et compilation ;
- deuxièmement, à faire l'édition de lien avec la routine d'optimisation choisie.

Ces considérations montrent que cette étape de traduction en un problème d'optimisation est entièrement automatisable : la traduction par substitution est automatisée, et prise en charge par PASCOSMA ; nous sommes dans la phase de tests pour l'implantation de la traduction par composition dans notre méthodologie.

## 2.6 ETAPE 6 : Dimensionnement

Cette dernière étape de la méthodologie consiste à utiliser l'outil de dimensionnement dédié à la structure et au fonctionnement du convertisseur étudié. Deux exemples de session de dimensionnement illustrent ceci dans le chapitre V.

Une session de dimensionnement passe par quatre phases :

- **Une phase d'initialisation des paramètres d'entrée** avec des valeurs physiquement viables, c'est-à-dire qui sont issues d'un point de fonctionnement dans le mode choisi par le dimensionnement.

Un jeu de valeurs convenables est donné par l'analyse de la simulation du point de fonctionnement initial (§ 2.3) (étapes 1 et 3)

- **Une phase de saisie du cahier des charges**, ce qui consiste à appliquer des contraintes sur les critères et paramètres du dimensionnement. Ces contraintes peuvent être une valeur fixe, un intervalle continu de valeurs. Il est aussi possible de ne pas contraindre un critère.
- **Une phase de lancement du dimensionnement**. Le programme détermine, dans l'espace de recherche restreint par les contraintes sur les paramètres, le jeu de valeurs pour les paramètres qui satisfait les contraintes sur les critères et qui minimise la fonction objectif. Ce calcul - problème inverse - est réalisé par un processus itératif accéléré par un algorithme d'optimisation du gradient.

- **Une phase d'analyse du dimensionnement.** A la fin du calcul, un compte rendu est affiché qui renseigne l'utilisateur sur la validité de la solution fournie, en particulier il donne la liste des contraintes violées.

L'analyse de la solution est facilitée par les visualisations graphiques proposées par l'outil :

- courbes d'évolution des paramètres et critères pendant le processus itératif de recherche,
- courbes de sensibilité d'un critère en fonction d'un paramètre d'entrée.

Un rebouclage sur ces quatre actions est laissé à la liberté de l'utilisateur. En effet, à partir de l'analyse, un réajustement des contraintes et des valeurs initiales des paramètres est réalisable, suivi du lancement d'un nouveau dimensionnement et d'une nouvelle analyse, et ainsi de suite ... Les deux applications présentées dans le chapitre V montrent l'utilité de ces rebouclages pour affiner le dimensionnement obtenu, et en particulier pour prendre en compte l'aspect discret des composants.

Le réajustement possible des contraintes consiste en une relaxation ou une restriction des contraintes (cf. Figure III.5).

- La relaxation est un élargissement de l'intervalle de valeurs acceptable pour le paramètre, ce qui peut aller jusqu'à ignorer le paramètre (si c'est un critère de sortie).
- La restriction d'une contrainte est un rétrécissement de l'intervalle de valeurs acceptable pour un paramètre ou un critère, ce qui peut aller jusqu'à une valeur fixe.

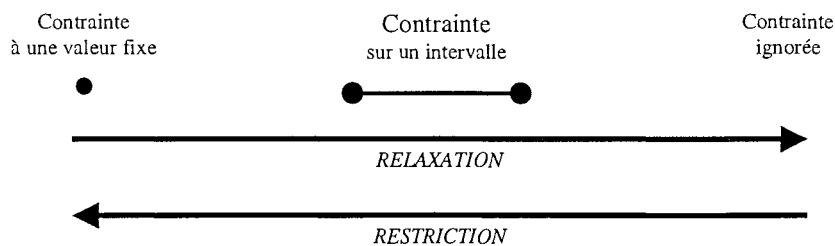


Figure III.5 : Les actions possibles sur les contraintes

En début de dimensionnement, il est préférable d'avoir des paramètres d'entrée fortement restreints, (à des valeurs fixes) et des critères de sortie très lâches (voire ignorés). L'ajustement du dimensionnement procède alors par relaxations et restrictions successives des paramètres d'entrée et des critères de sortie (cf. Figure III.6).

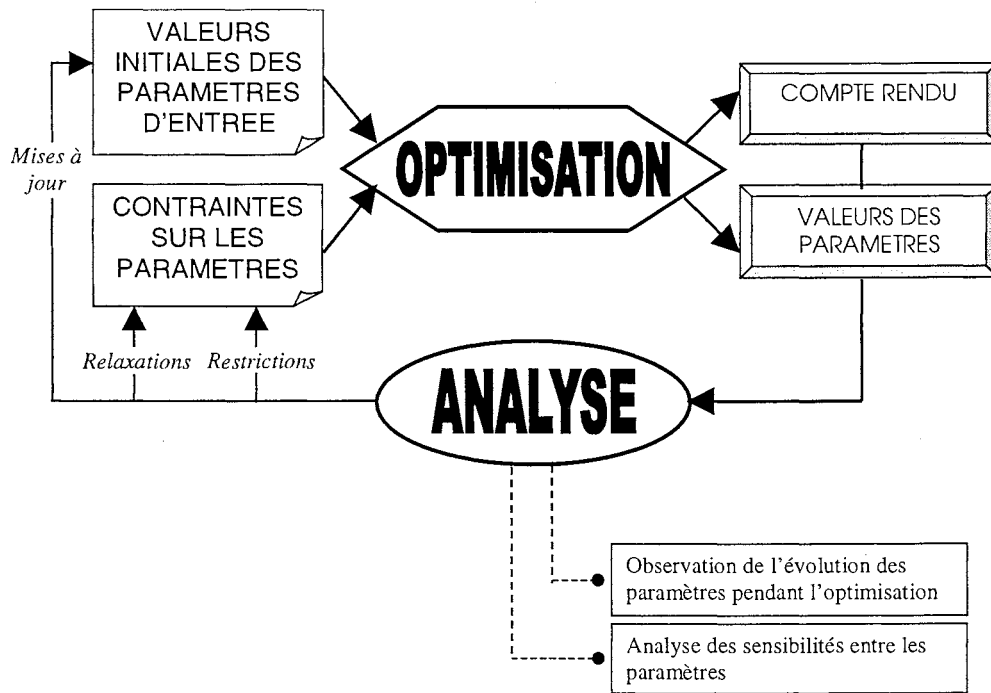


Figure III.6 : L'ajustement du dimensionnement

### 3 Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes, et leur agencement, pour la mise en œuvre de la méthodologie que nous proposons pour dimensionner un convertisseur de façon générale. D'une part, nous avons vu comment nous intégrons les différents outils de modélisation, de simulation et d'analyse dans cette approche de dimensionnement, ainsi que les méthodes utilisées. D'autre part, nous avons pu expliciter succinctement quelle information doit transiter entre les outils, et comment formuler cette information de façon à pouvoir la manipuler par des techniques informatiques, et donc ainsi automatiser l'ensemble de la démarche. Aussi, dans le prolongement de ces considérations plus ou moins théoriques, le chapitre suivant détaille l'implantation informatique de l'approche fondée sur ces outils et méthodes.

*Chapitre IV :*  
*Implantations*





## CHAPITRE IV : IMPLANTATIONS

L'implantation de la méthodologie de construction de l'outil de dimensionnement des convertisseurs que nous proposons repose sur quelques choix techniques généraux. Après avoir présenté les techniques fondamentales utilisées, nous ferons un bilan du niveau d'avancement de l'implantation : les étapes de l'approche qui avaient déjà été réalisées par ailleurs et ce qui a été réalisé dans ce travail. Enfin, nous exposerons le niveau d'intégration de la méthodologie actuelle et à venir.

### 1 *Choix techniques généraux*

Les trois techniques fondamentales de construction de notre méthodologie sont :

- **l'utilisation d'outils de conception et de modélisation existants** pour réaliser certaines étapes de la méthodologie,
- **le calcul symbolique** pour formuler le modèle de dimensionnement,
- **des méthodes numériques** éprouvées pour programmer l'outil final de dimensionnement dédié.

#### 1.1 Outils existants utilisés

##### 1.1.1 GENTIANE

GENTIANE (pour GENération et Traitement Informatiques, Analytiques et Numériques pour l'Electrotechnique) est un logiciel de simulation des ensembles machine – convertisseur statique – commande (Figure IV.1) [GERBAUD96].

Il possède une module de modélisation des convertisseurs statiques par topologie variable : GENTIANE-MEIGE (pour Modélisation Elaborée Informatiquement et Génériquement pour l'Electronique de Puissance) et un module de génération de modèle globaux et de programmes associés GENTIANE-ARMOISE (pour Analyse et Réorganisation de Modèles et Objets Informatiques pour la Simulation en Electrotechnique). Ces deux modules couplés fournissent le modèle de fonctionnement à topologie variable indispensable dans la méthodologie. De plus, le troisième module GENTIANE-ORO (pour Organisation et Retraitements en Objets) opère la simulation des applications du système modélisé, en s'appuyant sur le même modèle de base (à

topologie variable) dont les systèmes d'état sont intégrés par une méthode Runge Kutta d'ordre 4 avec adaptation automatique du pas de calcul.

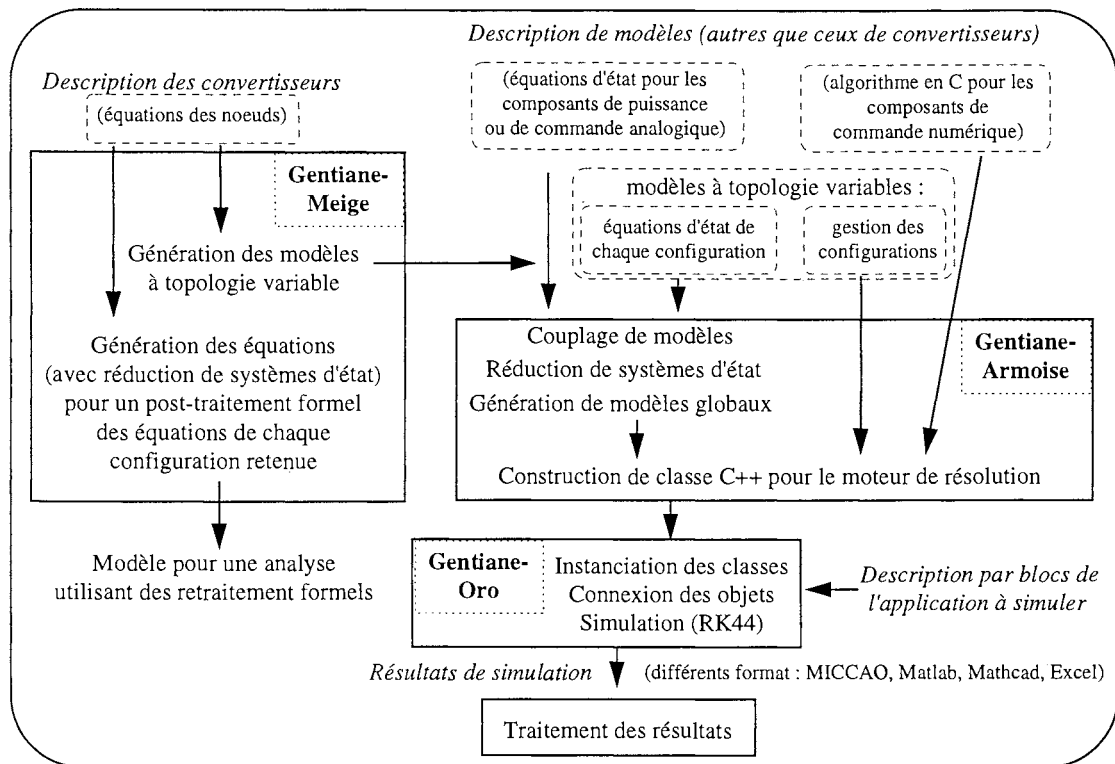


Figure IV.1 : GENTIANE [GERBAUD96]

### 1.1.2 PASCOMA

PASCOSMA (Programme d'Analyse, de Synthèse de Conception d'Optimisation de Systèmes Modélisables Analytiquement) est une méthodologie de génération automatique de logiciels de dimensionnement qui procède en sept étapes. Il est fondé sur une formulation analytique des équations de dimensionnement, et fut initialement conçu pour le dimensionnement de machines électriques.(cf. Figure IV.2) [WURTZ96].

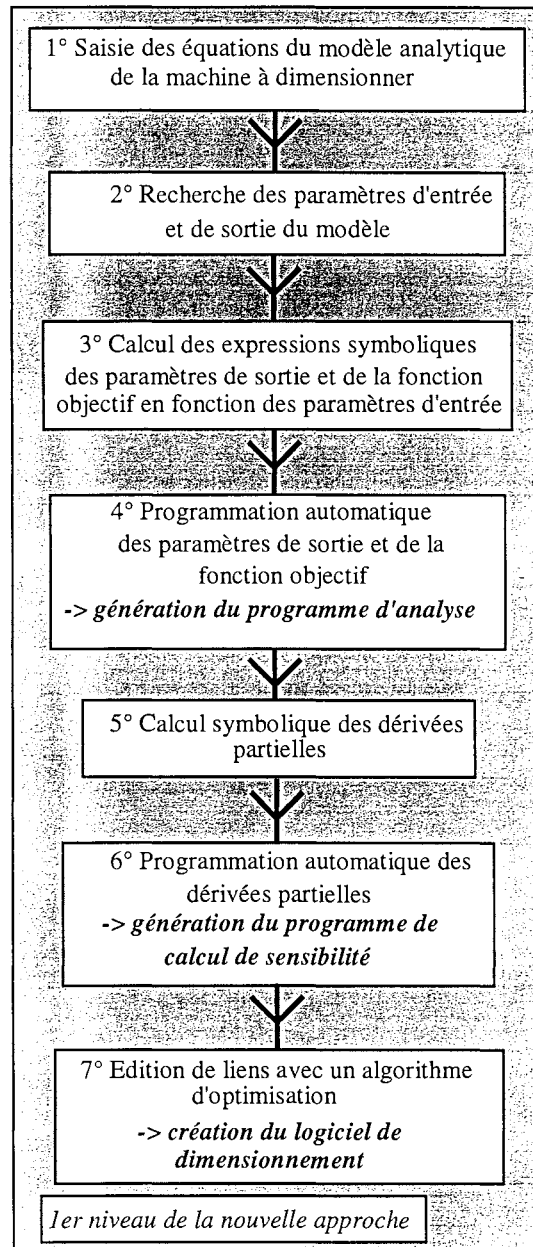


Figure IV.2 : Les sept étapes de l'approche PASCOSMA [WURTZ96]

De plus, il génère une interface conviviale pour utiliser le logiciel de dimensionnement automatique sous contraintes généré.

Cependant, la méthode employée par PASCOSMA V.1 pour la traduction du problème de dimensionnement en problème d'optimisation n'est plus efficace pour nos problèmes (SUBSTITUTIONS [WURTZ96]). Quelques aménagements ont donc été apportés à la méthodologie PASCOSMA initiale afin d'y introduire la composition. Ces aménagements consistent en la suppression des étapes suivantes :

- l'étape 3 de calcul des expressions symboliques des critères et de la fonction objectif en fonction des paramètres d'entrées, c'est-à-dire l'étape de substitution, qui n'a plus lieu d'être car nous l'avons remplacée par une étape procédant par composition,
- l'étape 5 de calcul symbolique des dérivées partielles qui est déjà prise en charge dans l'étape 4 de formulation du modèle de dimensionnement de notre méthodologie de dimensionnement.

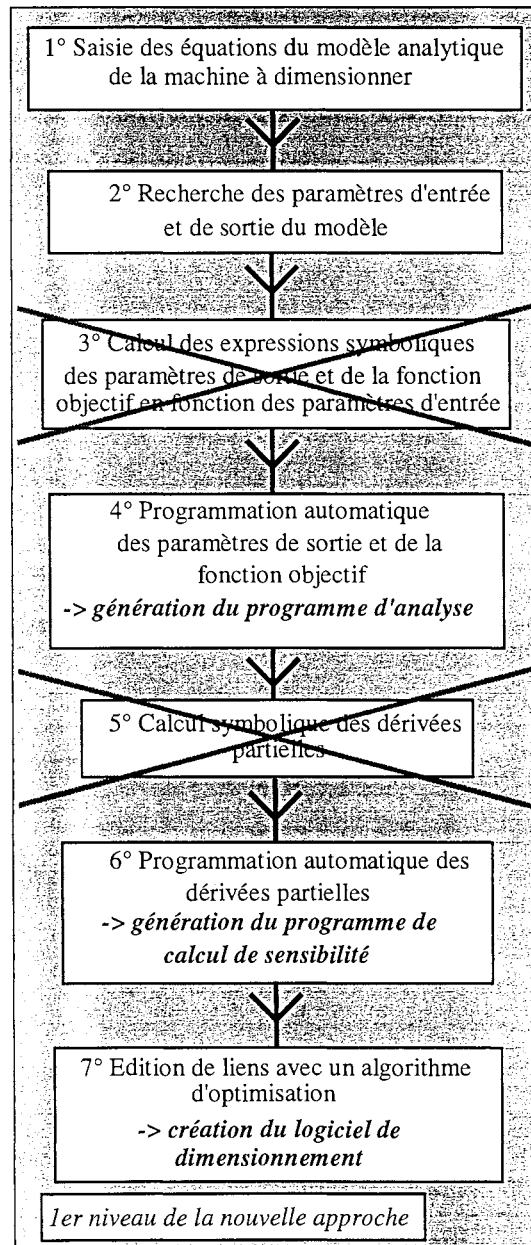


Figure IV.3 : L'approche PASCOSMA modifiée

### 1.1.3 Mathcad

Mathcad est un outil d'analyse très intéressant pour deux raisons :

- Du fait de ces capacités en matière de visualisation graphique, il est possible d'étudier l'évolution des courbes représentatives des courants et tensions dans tous les composants après la simulation, et cela par un simple chargement du fichier de points issus de l'étape de simulation.
- La version 6 de ce logiciel offre des capacités de programmation itérative très évoluées qui permettent l'analyse quantitative fine des fichiers de simulation demandée par la méthodologie.

L'analyse qualitative et quantitative de la simulation peut être réalisée efficacement avec cet outil. Nous avons programmé une feuille de calcul Mathcad pour une application donnée, ce qui fournit un moule réutilisable pour toutes les autres applications, car

- les fonctions de scrutation dans les tableaux sont pré-programmées,
- les modifications à faire sur les paramètres sont aisées,
- les ajouts et suppressions de courbes à visualiser sont rapides et aisés, et
- les calculs sont rapides.

## 1.2 Calcul symbolique

La formulation du modèle de dimensionnement est entièrement fondée sur des retraitements symboliques. Pour cela, Macsyma et Maple, deux logiciels de calculs symboliques, sont à notre disposition et ont des rôles complémentaires.

### 1.2.1 Macsyma

Macsyma est le premier logiciel de calcul symbolique utilisé dans la méthodologie. Le choix de ce logiciel est justifié par l'étendue et la puissance des fonctionnalités qu'il propose, et historiquement Macsyma a servi au développement de GENTIANE. De plus, pour ses capacités de manipulation et de traduction des expressions symboliques en langage de programmation (Fortran ou C), il a été préféré à d'autres logiciels du même type (Mathematica, Maple, Mathcad).

La plus grande partie de l'étape de formulation du modèle de dimensionnement est développée dans le langage de programmation évolué de Macsyma.

Les procédures et fonctions ainsi programmées réalisent les aspects suivants :

### 1.2.1.1 *L'intégration symbolique des systèmes d'état*

Macysma propose une méthode classique (Laplace) pour réaliser cette tâche. La programmation de l'intégration est générique dans ce cas. Cependant, pour faire face à un éventuel échec de cette intégration, nous avons développé une procédure de transformation spécifique des matrices d'état inspirée de [HAYES96] pour réaliser un découplage des variables d'état, afin de rendre possible une intégration classique. Cependant ce type de procédure n'est valable que pour un type de systèmes donné, et donc une structure donnée. Ainsi, la programmation de ces procédures spécifiques ne peut être réalisée qu'au cas par cas, et ne peut pas être étendue de manière à pouvoir traiter de façon exhaustive tous les systèmes d'état.

### 1.2.1.2 *Les retraitements symboliques sur les variables d'état*

Ces retraitements nécessaires pour formuler les critères de dimensionnement sont présentés en détail par la suite. Il s'agit de dérivations par rapport aux paramètres primaires et par rapport au temps

- d'intégrations temporelles  $\int dt$  des variables d'état  $X$ , et de produits de couples de variables d'état  $(X_i \cdot X_j)$ , afin d'exprimer les  $\int X dt$  et  $\int X_i \cdot X_j dt$
- de dérivations par rapport aux entrées primaires  $\frac{\partial}{\partial E'}$  de ces dernières expressions ( $\int X dt$  et

$$\int X_i \cdot X_j dt) \text{ et des } \frac{dX}{dt}, \text{ afin d'exprimer les } \left( \frac{\partial \frac{dX}{dt}}{\partial E'}, \frac{\partial \int X dt}{\partial E'}, \frac{\partial \int X_i \cdot X_j dt}{\partial E'} \right)$$

### 1.2.1.3 *Les sommes algébriques opérées sur toutes ces précédentes expressions symboliques*

Ceci est fait afin de calculer les  $\left( Y, \frac{dY}{dP}, \int Y dt, \int Y^2 dt \right)$  sachant que  $Y_k = \sum_i \gamma_{ki} X_i + \delta_k$  (cf. (59)

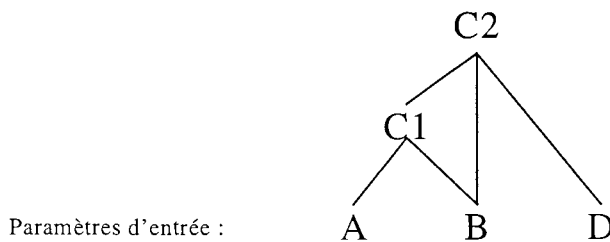
p. 93)

### 1.2.1.4 Le calcul des gradients par composition

Ceci consiste en la construction de l'arbre de dépendance des variables du problème de dimensionnement, puis la formulation symbolique des dérivées des critères en fonction des paramètres d'entrée, en utilisant les variables intermédiaires qui contiennent les expressions symboliques des dérivées des variables par rapport aux paramètres dont elles dépendent directement. Ces dernières expressions étant calculées indépendamment (en 1.2.1.2 et 1.2.1.3).

L'exemple suivant illustre cet aspect de composition :

- arbre de dépendance des critères  $C_1(A,B)$  et  $C_2(C_1,D,B)$



- dérivées calculées :  $G1 = \frac{\partial C1}{\partial A}$ ,  $G2 = \frac{\partial C1}{\partial B}$ ,  $G3 = \frac{\partial C2}{\partial C1}$ ,  $G4 = \frac{\partial C2}{\partial D}$ ,  $G5 = \frac{\partial C2}{\partial B}$

- gradients :  $\frac{dC1}{dA} = G1$ ,  $\frac{dC1}{dB} = G2$ ,  $\frac{dC1}{dD} = 0$   
 $\frac{dC2}{dA} = G1 \cdot G3$ ,  $\frac{dC2}{dB} = G2 \cdot G3 + G5$ ,  $\frac{dC2}{dD} = G4$

### 1.2.1.5 Les traductions des expressions symboliques en Fortran.

Toutes les expressions précédents permettent de formuler les expressions symboliques des critères et gradients, qui sont traduites en Fortran afin d'obtenir la procédure de calcul numérique appellable par PASCOSMA.

## 1.2.2 Maple

L'utilisation de Maple comme outil de calcul formel dans la méthodologie, est apparue intéressante et nécessaire pour réaliser l'intégration symbolique partielle avec rootof présentée en . En effet, ce logiciel possède une fonctionnalité intégrée d'intégration symbolique des systèmes d'équations différentielles, qui propose la formulation des solutions en fonction des racines d'un polynôme (polynôme caractéristique réduit et factorisé) lorsqu'une solution purement radicale n'est pas



accessible (degré supérieur à quatre ou système trop complexe). Les travaux dans ce cadre sont encore en cours. Même s'ils ne sont pas finalisés, ils présentent des caractéristiques génériques que nous avons d'ores et déjà souhaités prendre en compte.

### 1.2.2.1 L'intégration symbolique des systèmes d'état

Le programme développé par David LANIER pendant son DEA permet d'intégrer les systèmes d'état de degré supérieur à quatre en formulant automatiquement les variables d'état en fonction du temps et des "rootofs", racines d'un polynôme issu de la réduction du polynôme caractéristique du système différentiel comme suit (53) :

$$X_i(E', t, \alpha_1, \alpha_2, \dots, \alpha_k) \quad \text{avec} \quad Q(E', \alpha_1, \alpha_2, \dots, \alpha_k) = 0 \quad (53)$$

que l'on généralise par la notation suivante (54) :

$$X_i(E', t, \alpha) \quad \text{avec} \quad Q(E', \alpha) = 0 \quad (54)$$

Ces expressions peuvent toujours être formulées sous la forme d'une somme de deux éléments (55) :

$$\begin{aligned} X_i(E', t, \alpha) &= p(E', t) + \sum_{i=1}^n q(E', t, \alpha_i) \\ &= \Gamma + \Phi \end{aligned} \quad (55)$$

- un premier élément contenant la partie de l'expression indépendante de la racine numérique :

$$\Gamma = p(t)$$

- un second élément contenant la partie dépendant du "rootof" :

$$\Phi = \sum_{i=1}^n q(\alpha_i, t)$$

### 1.2.2.2 Les retraitements symboliques

Les retraitements symboliques sur les variables d'état ainsi exprimés consistent à effectuer les mêmes calculs que ceux présentés en 1.2.1.2 :

- dérivations par rapport au temps  $\frac{dX}{dt}$ ,
- dérivations par rapport aux paramètres primaires  $\frac{\partial X}{\partial E'}$ ,
- intégrations temporelles  $\int X dt$  et  $\int X_i \cdot X_j dt$ ,

- et dérivations partielles par rapport aux  $E'$  afin d'obtenir les  $\left( \frac{\partial \frac{dX}{dt}}{\partial E'}, \frac{\partial \int X dt}{\partial E'}, \frac{\partial \int X_i \cdot X_j dt}{\partial E'} \right)$  qui

servent pour la formulation des gradients.

Le logiciel Maple permet de réaliser ces retraitements sur les variables d'état exprimées en fonction des rootofs comme défini en (55).

### 1.3 Méthodes numériques

Après la formulation symbolique du modèle de dimensionnement, des calculs numériques doivent être réalisés : la résolution du problème d'optimisation dual d'une part passe par des évaluations numériques des variables du modèle et d'autre part nécessite l'utilisation d'un algorithme d'optimisation. De plus, la formulation semi-numérique des variables d'état, proposée dans le cas de systèmes d'état de dimension supérieure à quatre, sous-entend le calcul numérique des racines d'un polynôme. Les deux procédures utilisées pour l'optimisation et la résolution du polynôme sont issues de la bibliothèque Harwell :

subroutine VF13AD pour l'algorithme d'optimisation

subroutine PA17 pour le calcul des racines d'un polynôme

Cette bibliothèque a été choisie pour des raisons historiques (l'algorithme d'optimisation est celui de PASCOSMA), aussi parce que les algorithmes et les programmes ont fait preuve de leur robustesse depuis leur développement (1983, 1973) et sont régulièrement révisés et optimisés (release 1995).

Le langage que nous avons choisi pour programmer les évaluations numériques des critères et gradients du modèle de dimensionnement est le FORTRAN. Ce langage permet une bonne traduction des expressions symboliques en code, et est bien adapté au calcul numérique. Cette traduction est automatiquement prise en charge par le logiciel de calcul formel que nous employons : MACSYMA et de plus elle est optimisée de façon à accélérer les temps de calcul (déclarations automatiques des variables internes).

## 2 Etapes déjà réalisées par ailleurs

La Figure IV.4 qui se réfère à la figure III.1 du chapitre III synthétisant la démarche en six étapes, présente en grisé les différentes étapes qui utilisent des outils réalisés dans l'équipe C.D.I du L.E.G, et récupérées partiellement voir totalement.

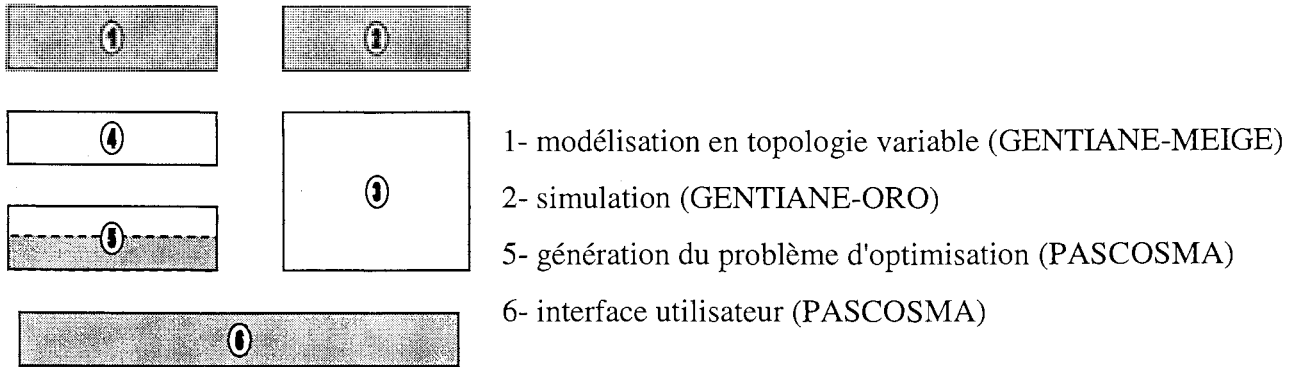


Figure IV.4 : Les étapes déjà réalisées

## 3 Etapes en cours de réalisation

La Figure IV.5 présente les étapes que nous avons réalisées et qui sont encore en cours de tests.

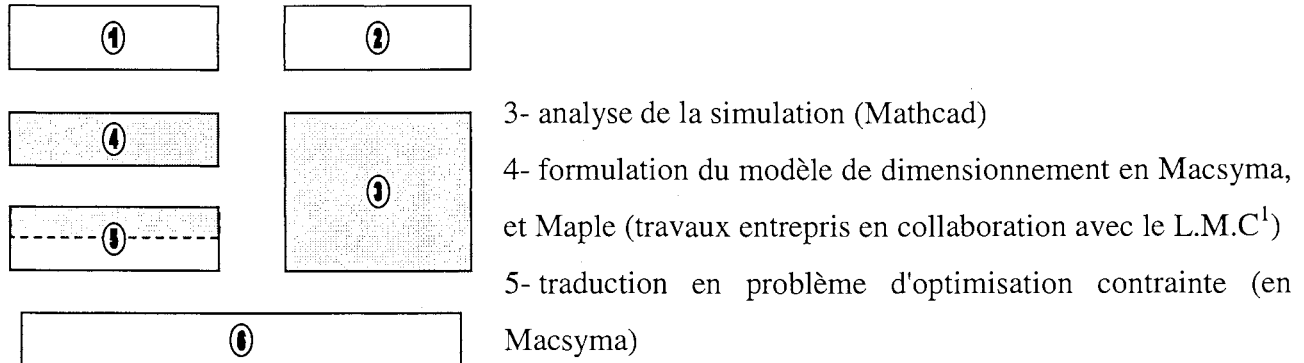


Figure IV.5 : Les étapes en cours de réalisation

Nous allons maintenant développer ces aspects.

### 3.1.1 Analyse de la simulation

Le développement informatique de cette étape a consisté à mettre au point un programme générique d'analyse du fichier de points issu de la simulation, avec le logiciel Mathcad.

<sup>1</sup> Laboratoire de Modélisation et de Calcul

Ce travail se traduit actuellement par la programmation de trois fonctions :

- une fonction d'extraction du mode de fonctionnement qui donne l'enchaînement des configurations sur une période,
- une fonction de calcul des durées des configurations,
- une fonction d'extraction des valeurs initiales de chaque variable d'état, au début de chaque configuration sur une période.

Quant à la tâche d'extraction de l'information sur les extremums des courants et tensions (configurations pendant lesquelles ils apparaissent, date et valeur), elle reste encore entièrement à la charge de l'utilisateur.

Ce travail consiste à scruter chaque colonne du fichier de données, de repérer les valeurs extrémales (en valeur absolue), de noter le numéro (*config*) de la configuration correspondante, et de calculer la date relative réduite  $\tilde{t}_{ext}$  de cet événement par la relation suivante (56), où  $t_{ext}$  est la date naturelle de l'extremum,  $t0_{config}$  est la date du début de la configuration pendant laquelle a lieu l'extremum en question, et  $t_{config}$  est la durée de la configuration.

$$\tilde{t}_{ext} = \frac{t_{ext} - t0_{config}}{\tau_{config}} \quad (56)$$

Une fonctionnalité de simulation numérique, utilisant les expressions issues de l'intégration formelle des systèmes d'état, pourrait aussi avantageusement enrichir l'analyse du fonctionnement du convertisseur à ce niveau.

Les capacités graphiques du logiciel Mathcad permettent d'envisager ce travail, cependant sa mise en œuvre est freinée par l'incompatibilité des expressions symboliques issues de Macsyma avec Mathcad. Cela veut dire que le développement d'un tel outil d'analyse est pour l'instant subordonné à l'écriture manuelle des expressions symboliques obtenues après l'intégration des systèmes d'état, travail très fastidieux vue la taille des expressions générées, et source de nombreuses erreurs.

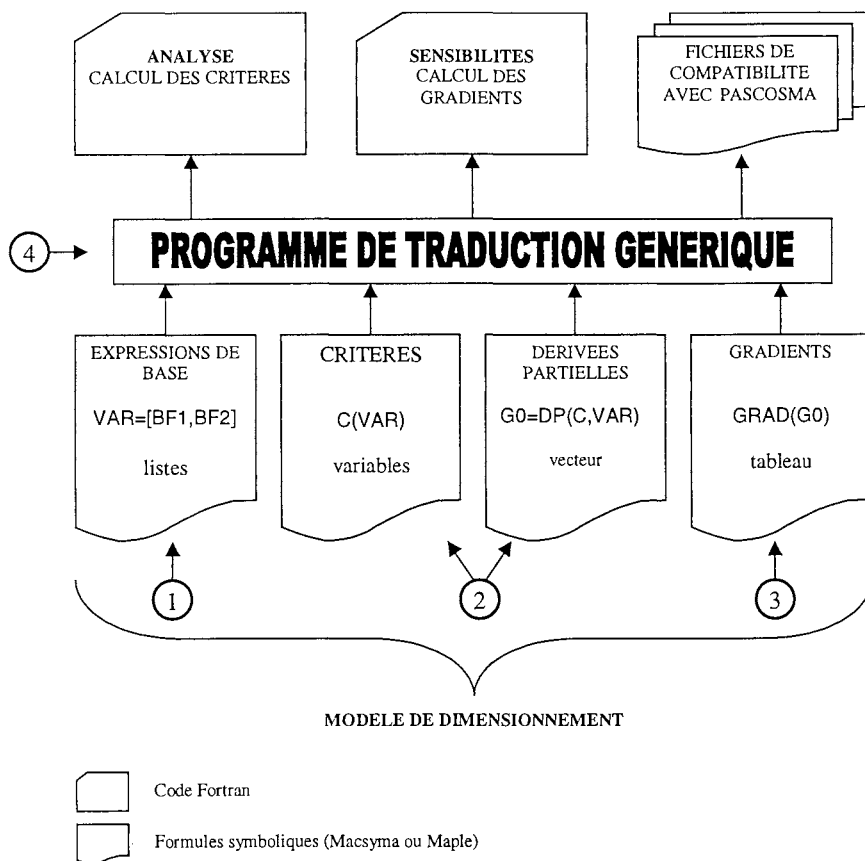
De plus, cette simulation n'est envisageable que pour une intégration symbolique pure (sans rootof) des systèmes d'état.

On pourrait envisager de remplacer le module de simulation numérique de GENTIANE par une simulation "semi-symbolique" dont les seules résolutions numériques seraient le calcul des *rootofs* pour obtenir l'évolution des variables en fonction du temps, sans passer par une intégration de type Runge-Kutta.

### 3.1.2 Formulation du modèle de dimensionnement et traduction du problème d'optimisation

Nous proposons de réaliser le travail en quatre étapes indépendantes (cf. Figure IV.6) :

- 1- le calcul des expressions de base qui servent à construire l'ensemble du modèle de dimensionnement,
- 2- la formulation symbolique des critères et des dérivées partielles de ces critères en fonction des expressions de base,
- 3- la formulation des gradients en fonction des dérivées partielles (par composition),
- 4- la génération du code d'évaluation des critères et des gradients en FORTRAN.



**Figure IV.6 : La traduction du modèle contraint en un problème d'optimisation pour PASCOSMA**

#### 3.1.2.1 Calcul des expressions de base

##### 3.1.2.1.1 Définition

Les expressions de base sont (cf. chap. II) :

- les  $X(t)$  et  $Y(t)$ , pour formuler les critères de continuité, les valeurs maximales, les tests ;
- les  $\frac{dX}{dt}$  et les  $\frac{dY}{dt}$  pour formuler les pentes ;
- les  $\int X dt$  et les  $\int Y dt$  pour formuler les valeurs moyennes ;
- les  $\int X^2 dt$  et les  $\int Y^2 dt$  pour formuler les valeurs efficaces ;

et de plus, on ajoute les dérivées partielles de ces grandeurs par rapport aux paramètres d'entrées, expressions qui servent à calculer les gradients de tous les critères, soient :

- les  $\frac{\partial X}{\partial E}, \frac{\partial Y}{\partial E}, \frac{\partial}{\partial E} \left( \frac{dX}{dt} \right), \frac{\partial}{\partial E} \left( \frac{dY}{dt} \right), \frac{\partial}{\partial E} \left( \int X dt \right), \frac{\partial}{\partial E} \left( \int Y dt \right), \frac{\partial}{\partial E} \left( \int X_i X_j dt \right), \frac{\partial}{\partial E} \left( \int Y^2 dt \right)$ .

Toutes ces expressions peuvent être calculées à partir des expressions symboliques des variables d'état, qu'elles fassent ou non intervenir des racines indéterminées symboliquement. Dans le cas où l'intégration du système d'état n'est pas complètement formelle (apparition de "rootofs"), les variables d'état s'écrivent :

$$X(\alpha, t) = f(t) + \sum_{i=1}^n g(\alpha_i, t) \quad (57)$$

où

- $g(t)$  est la partie de l'expression indépendante des racines (les rootofs),
- $\sum_{i=1}^n f(\alpha_i, t)$  est la partie qui dépend des racines,  $f(\alpha, t)$  est la même fonction quelle que soit  $\alpha$ ,
- $n$  est le degré du polynôme  $Q(\alpha)$  non factorisé symboliquement, et  $n$  est donc aussi le nombre de racines de ce polynôme  $Q(\alpha) = 0$ , qui seront calculées numériquement.

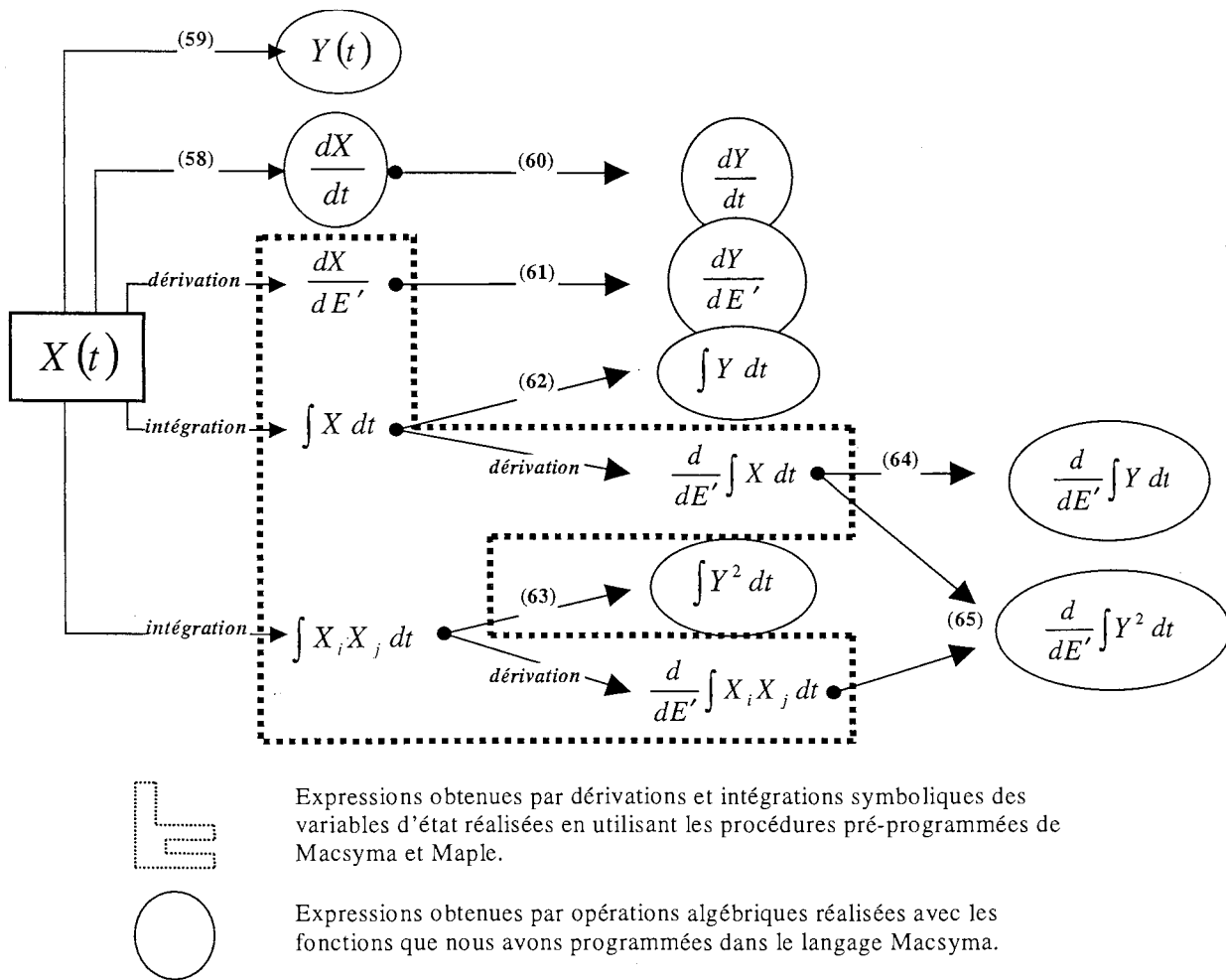


Figure IV.7 : Le processus automatique de calcul des expressions de base

L'ordre que nous proposons pour réaliser le calcul des expressions de base permet de diminuer les temps de génération, et les tailles des expressions à manipuler par Macsyma, en réutilisant au maximum les expressions déjà formulées.

Nous calculons d'abord les expressions symboliques des dérivées temporelles des variables d'état en utilisant les systèmes d'état données par GENTIANE-MEIGE (58)

$$\frac{dX}{dt} = A \cdot X + B \quad (58)$$

Puis nous calculons les variables  $Y$  non variables d'état avec le système matriciel offert par GENTIANE-MEIGE (59)

$$Y(t) = C \cdot X(t) + D(t) \quad (59)$$

avec  $C = (\gamma_{ij})$  et  $D = (\delta_i)$

Ensuite nous exprimons les dérivées temporelles des  $Y$  toujours en utilisant le système matriciel (60) où  $C$  est indépendant du temps, et où nous avons réalisé la dérivation temporelle du vecteur  $D$  :

$$\frac{dY}{dt} = C \cdot \frac{dX}{dt} + \frac{dD}{dt} \quad (60)$$

Sur le même principe de composition algébrique, nous calculons automatiquement à l'aide d'un module que nous avons programmé en Macsyma, toutes les équations symboliques des autres expressions de base faisant intervenir les  $Y$  :

$$\frac{\partial Y}{\partial E'} = \frac{\partial C}{\partial E'} \cdot X(t) + C \cdot \frac{\partial X}{\partial E'} + \frac{\partial D}{\partial E'} \quad (61)$$

$$\int Y dt = C \cdot \int X dt + \int D dt \quad (62)$$

$$\int Y_i^2 dt = \sum_j \left[ \left( \gamma_{ij}^2 \int X_j^2 dt \right) + \left( 2\gamma_{ij} \int \delta_i \cdot X_j dt \right) + \left( 2\gamma_{ij} \sum_{k < j} \gamma_{ik} \int X_j \cdot X_k dt \right) \right] + \int \delta_i^2 dt \quad (63)$$

$$\frac{\partial}{\partial E} \int Y dt = \frac{\partial C}{\partial E} \cdot \int X dt + C \cdot \frac{\partial}{\partial E} \int X dt + \frac{\partial}{\partial E} \int D dt \quad (64)$$

$$\frac{\partial}{\partial E} \int Y_i^2 dt = \sum_j \left[ \left( 2\gamma_{ij} \frac{\partial \gamma_{ij}}{\partial E} \int X_j^2 dt \right) + \left( \gamma_{ij}^2 \frac{\partial}{\partial E} \int X_j^2 dt \right) + \left( 2 \frac{\partial \gamma_{ij}}{\partial E} \int \delta_i \cdot X_j dt \right) \right. \\ \left. + \left( 2\gamma_{ij} \frac{\partial}{\partial E} \int \delta_i \cdot X_j dt \right) + \left( 2 \sum_{k < j} \left( \gamma_{ij} \frac{\partial \gamma_{ik}}{\partial E} \int X_j \cdot X_k dt + \gamma_{ij} \gamma_{ik} \frac{\partial}{\partial E} \int X_j \cdot X_k dt \right) \right) \right] + \frac{\partial}{\partial E} \int \delta_i^2 dt \quad (65)$$

### 3.1.2.1.2 Méthode de calcul

Pour calculer les expressions de base, nous procédons en deux temps :

#### 3.1.2.1.2.1 Calculs des expressions de base liées directement aux variables d'état $X$

Il ne s'agit que de procédures d'intégrations et de dérivations afin d'obtenir, à partir des expressions symboliques des  $X$ , les expressions symboliques des  $\frac{dX}{dt}$ ,  $\frac{\partial X}{\partial E}$ ,  $\frac{\partial}{\partial E} \left( \frac{dX}{dt} \right)$ ,  $\int X dt$ ,  $\frac{\partial}{\partial E} \left( \int X dt \right)$ ,  $\int X_i X_j dt$ , et  $\frac{\partial}{\partial E} \left( \int X_i X_j dt \right)$ . Ces dérivations et intégrations sont toujours calculables symboliquement pour nos problèmes.



Deux cas de figure peuvent apparaître :

- Soit l'intégration formelle des systèmes d'état donne des variables d'état entièrement symbolique  $X(t)$ , auquel cas les dérivations et intégrations peuvent être réalisées par les méthodes symboliques classiques.

Dans ce cas, des procédures automatiques programmées avec le langage Macsyma effectuent ces opérations.

- Soit l'intégration des systèmes d'état fait intervenir des racines non radicales, et alors on a une formulation "quasi-symbolique" des variables d'état en fonction des "rootofs" qui seront évalués numériquement :

$$X(\alpha, t)$$

( $\alpha$  étant le vecteur appelé rootof des racines du polynôme non factorisable symboliquement)

Dans ce cas, dérivations et intégrations sont encore possibles, mais les méthodes classiques proposées par Macsyma ne suffisent plus. Maple permet de réaliser ces opérations, et les procédures de calcul des expressions symboliques ont été programmées dans l'environnement de ce logiciel [LANIER97].

Un résultat important [LANIER97] à noter dans le cas d'apparitions de *rootofs* après l'intégration des systèmes différentiels, est le fait que toutes les expressions de base obtenues par ces retraitements s'expriment en fonction des seuls  $E'$ ,  $t$ , et  $\alpha$  ; aucun autre paramètre n'intervient après ces retraitements, et en particulier aucune variable autre que les *rootofs*  $\alpha$  ne sera à calculer numériquement.

Les expressions  $\psi$  calculées ont toutes la forme suivante :

$$\psi(E', \alpha, t) = p(E', t) + \sum_{i=1}^n q(E', \alpha_i, t) \quad (66)$$

Bien que ce travail ne soit pas terminé, nous avons souhaité prendre en compte les spécificités qu'il apportait dans notre formulation du modèle de dimensionnement.

### 3.1.2.1.2.2 Calculs des expressions de base liées aux autres grandeurs $Y$ non variables d'état

Il s'agit de sommes algébriques des expressions calculées précédemment.

Se pose à ce niveau un problème de formalisme des expressions issues des premiers calculs symboliques : il faut mettre au point une forme pour ces expressions de manière à pouvoir les

manipuler aisément, c'est à dire les additionner, les multiplier, voir les formules (59), (60), (61), (62), (63), (64), et (65).

Le formalisme que nous proposons repose sur la notion de listes, notion fondamentale pour la programmation avec la langage de Macsyma - Macsyma étant développé en Lisp -. Nous présentons le principe des retraitements formels reposant sur la manipulation de ces liste en annexe A.

### **3.1.2.2 Formulation symbolique des critères et dérivées partielles**

Deux méthodes peuvent être employées pour réaliser la formulation des critères :

#### **3.1.2.2.1 Une formulation par substitution**

Cette formulation par substitution consiste à développer les expressions de chaque critère en fonction des seuls paramètres d'entrée.

Prenons l'exemple trivial de deux critères  $C_1$  et  $C_2$ , et trois paramètres d'entrée A,B et D,

$$C_1 = A + B$$

$$C_2 = C_1 \times D$$

La formulation par substitutions consiste à reformuler  $C_2$  par  $C_2 = (A + B) \times D$

Cette méthode est efficace pour des expressions simples des critères et l'obtention des gradients à partir des formules substituées est aisée : il suffit de réaliser les dérivations partielles par rapport à chaque paramètre d'entrée.

Exemple :

$$\frac{dC_2}{dA} = \frac{\partial}{\partial A} ((A + B) \cdot D) = D$$

$$\frac{dC_2}{dB} = \frac{\partial}{\partial B} ((A + B) \cdot D) = D$$

$$\frac{dC_2}{dD} = \frac{\partial}{\partial D} ((A + B) \cdot D) = A + B$$

C'est la méthode de formulation employée dans la première version du logiciel PASCOSMA.

Cependant cette méthode n'est plus judicieuse lorsque la structure du modèle de dimensionnement est trop imbriquée, i.e. quand des critères sont formulés à partir d'autres critères. En effet, la méthode de substitution provoque alors des redites de calcul (et des tailles d'expression non gérables). Notre modèle de dimensionnement est caractérisée par une forte imbrication.

Dans la Figure IV.8 ci-dessous, nous rappelons la structure du modèle de dimensionnement proposée.

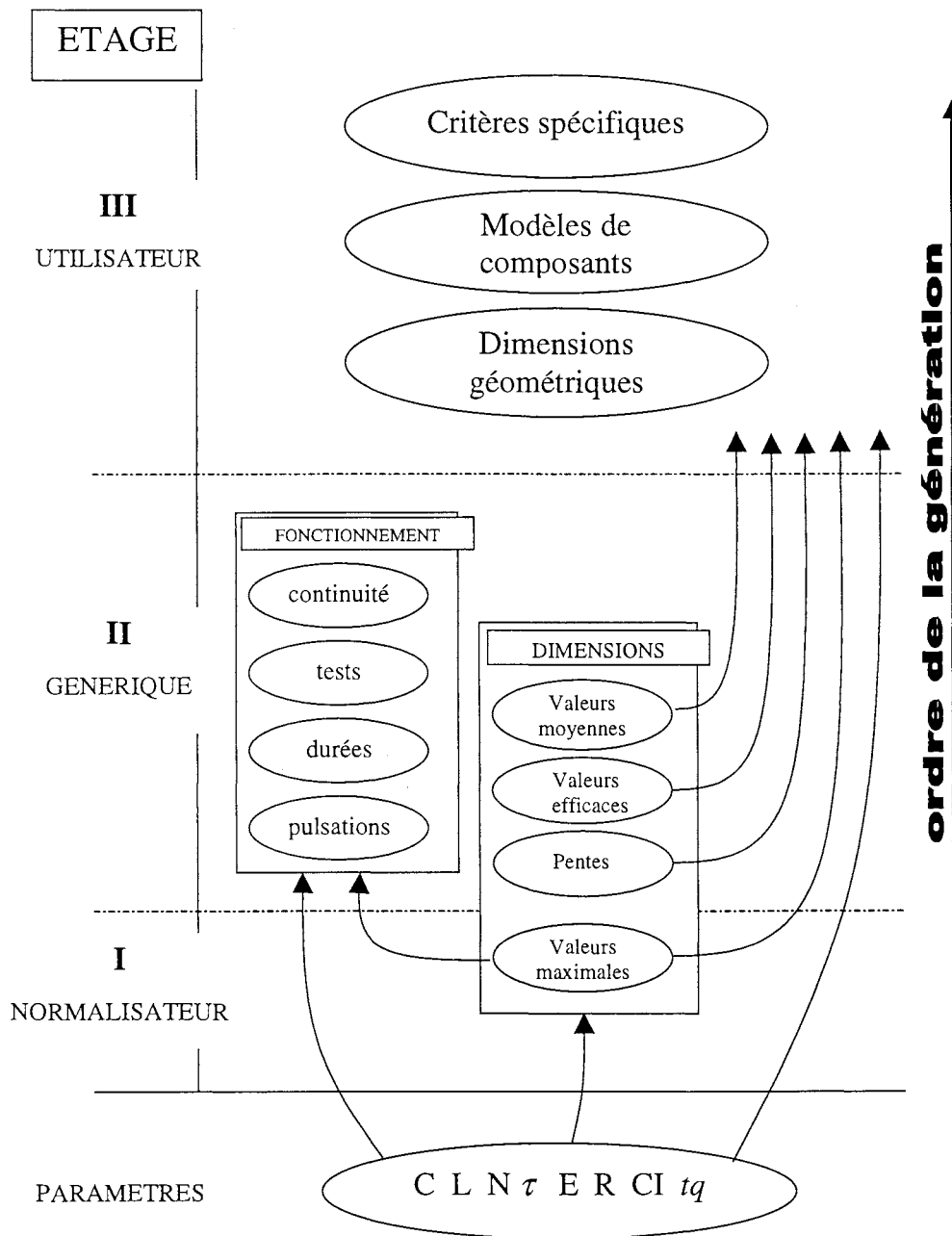


Figure IV.8 : L'architecture du modèle de dimensionnement

### 3.1.2.2.2 Formulation par composition

Dans le cas d'imbrications dans le modèle de dimensionnement, nous préconisons une formulation symbolique des critères par une méthode dite "de composition". Il s'agit de formuler chaque critère en fonction des paramètres dont il dépend directement (paramètre d'entrée ou autre critère), ainsi que les dérivées partielles de chaque critère en fonction des paramètres dont il dépend directement.

Exemple :

Pour notre exemple,  
Du calcul des dérivées partielles,

$$\frac{\partial C_1}{\partial A} = \frac{\partial}{\partial A} (A + B) = 1$$

$$\frac{\partial C_1}{\partial B} = \frac{\partial}{\partial B} (A + B) = 1$$

$$\frac{\partial C_2}{\partial C_1} = \frac{\partial}{\partial C_1} (C_1 \cdot D) = D$$

$$\frac{\partial C_2}{\partial D} = \frac{\partial}{\partial D} (C_1 \cdot D) = C_1$$



On déduit les gradients :

$$\frac{dC_1}{dA} = 1$$

$$\frac{dC_1}{dB} = 1$$

$$\frac{dC_2}{dA} = \frac{\partial C_2}{\partial C_1} \cdot \frac{\partial C_1}{\partial A} = D$$

$$\frac{dC_2}{dD} = C_1 = (A + B)$$

### 3.1.2.3 Formulation des gradients

On peut, d'après l'architecture du modèle (cf. Figure IV.8), construire l'arbre de dépendance des critères du dimensionnement :

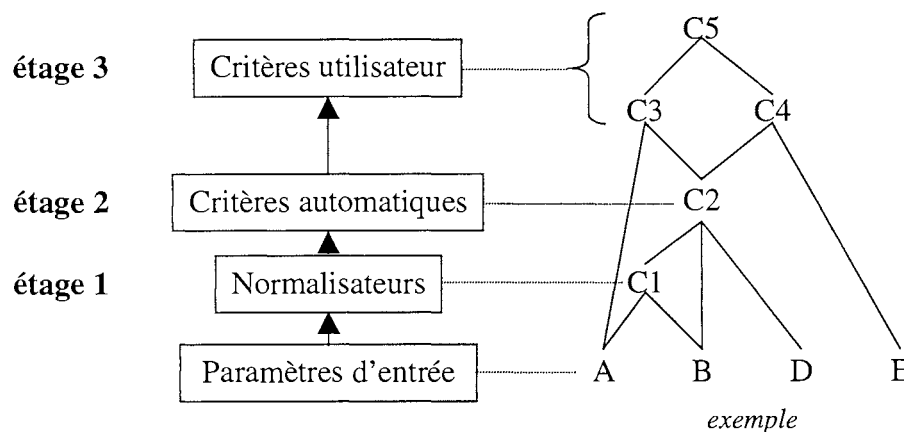


Figure IV.9 : L'arbre de dépendance des critères du dimensionnement

- A sa base, on trouve les paramètres d'entrée (primaire et secondaire).
- Le premier étage dans l'arbre contient les critères sur les valeurs maximales, qui dépendent uniquement des paramètres d'entrée. Ces critères sont des critères de dimensions, mais aussi ils servent à normaliser les critères de fonctionnement.  
Pour cela, ils doivent donc être formulés en premier.
- Le second étage contient les critères générés automatiquement :
  - critères de fonctionnement (continuité, tests de commutation, durées, pulsations) qui dépendent des paramètres d'entrée et des critères normalisateurs (qui sont les valeurs maximales, cf. chap. II § 3.3),

- critères de dimensions (valeurs efficaces, valeurs moyennes) formulés en fonction des paramètres d'entrée.
- Le dernier étage contient tous les critères définis par l'utilisateur. Ces derniers sont formulés à partir des critères de dimensions électriques définis précédemment, mais aussi ils peuvent être eux-mêmes imbriqués. Aussi, une procédure d'ordonnement des critères utilisateur de l'arbre de dépendance. Ainsi l'utilisateur peut implanter ses propres critères sans se soucier de l'ordre de leur évaluation (dans l'algorithme d'optimisation).

Les deux premiers étages de l'arbre de dépendance sont construits de façon statique – l'imbrication de critères est connue -, le troisième étage est construit dynamiquement - après l'implantation des critères utilisateur -.

A partir des expressions symboliques composées, et de l'arbre de composition ainsi construit, il est possible de formuler les gradients (i.e. les dérivées des critères en fonction des paramètres d'entrée). Pour cela, il suffit de s'appuyer sur les dérivées partielles de chaque critère en fonction des paramètres dont il dépend directement :

Exemple : Pour notre exemple, cela donne :

$$\frac{\partial C_5}{\partial C_3}, \frac{\partial C_5}{\partial C_4}, \frac{\partial C_3}{\partial A}, \frac{\partial C_3}{\partial C_2}, \frac{\partial C_4}{\partial C_2}, \frac{\partial C_4}{\partial D}, \frac{\partial C_2}{\partial C_1}, \frac{\partial C_2}{\partial C}, \frac{\partial C_1}{\partial A}, \text{ et } \frac{\partial C_1}{\partial B}$$

Les gradients s'obtiennent par composition des dérivées partielles :

Exemple :

$$\frac{dC_5}{dD} = \frac{\partial C_5}{\partial C_4} \cdot \frac{\partial C_4}{\partial D}$$

$$\frac{dC_5}{dA} = \frac{\partial C_5}{\partial C_3} \cdot \left( \frac{\partial C_3}{\partial A} + \frac{\partial C_3}{\partial C_2} \cdot \frac{\partial C_2}{\partial C_1} \cdot \frac{\partial C_1}{\partial A} \right) + \frac{\partial C_4}{\partial C_2} \cdot \frac{\partial C_2}{\partial C_1} \cdot \frac{\partial C_1}{\partial A}$$

### 3.1.2.4 Génération du code d'évaluation des critères et des gradients

La génération du code d'évaluation des critères et des gradients consiste à traduire les expressions symboliques correspondantes en un langage exécutable.

Nous avons choisi le FORTRAN car il est bien adapté pour ce travail<sup>2</sup>, et Macsyma offre les fonctionnalités pour traduire ses expressions symboliques en FORTRAN, en prenant de plus en charge l'optimisation du code généré.

La structure imbriquée, provoquée par la composition des expressions, implique des évaluations successives des formules, pour les critères et pour les dérivées partielles.

Un logiciel générique a été développé dans le langage Macsyma pour réaliser cette génération automatiquement.

A partir des quatre briques fondamentales du modèle de dimensionnement, qui sont

- les expressions de base  $Z_x, Z_y, \frac{\partial Z_x}{\partial E}$  et  $\frac{\partial Z_y}{\partial E}$ ,
- les expressions symboliques des critères en fonction des expressions de base,
- les expressions symboliques des dérivées partielles en fonction aussi des expressions de base,
- les expressions symboliques de la composition des gradients,

ce logiciel génère le code FORTRAN

- d'évaluations séquentielles des critères (le programme d'analyse),
- d'évaluations séquentielles des gradients (le programme de calcul des sensibilités),

et des fichiers permettant d'assurer la compatibilité du code que nous avons programmé avec le reste de la méthodologie PASCOMA qui s'occupe de

- la création de l'interface graphique, et de
- la traduction du problème de dimensionnement en un problème d'optimisation (édition de lien avec l'algorithme d'optimisation).

### **3.1.2.5 Remarque sur la structure du modèle de dimensionnement ainsi formulé**

On peut remarquer une forte indépendance des quatre briques fondamentales du modèle contraint (expressions de base, expressions des critères, expressions des dérivées partielles, expressions de gradients) cf. Figure IV.6 p. 91. Chacune de ces briques peut être formulée indépendamment des trois autres. Pour l'instant, chacune d'elle est construite successivement, mais on pourrait réaliser ces calculs en parallèle, ce qui accélérerait la formulation du modèle de dimensionnement que nous proposons.

---

<sup>2</sup> FORTRAN = FORmula TRANslator

### 3.1.3 Intégration / Couplage informatique

L'intégration de la méthodologie est sa capacité à mettre en jeu des méthodes et outils différents et complémentaires tout en restant cohérente, afin de pouvoir être pilotée aisément (par un utilisateur ou un système expert).

La méthodologie que nous proposons est un mixage de simulation numérique, calcul symbolique, programmation automatique, synthèse automatique de modèles, analyse de résultats automatiques, optimisation, génération automatique d'interfaces, qui s'appuie sur différents outils informatiques - Macsyma, Maple, Mathcad, bibliothèques Harwell, GENTIANE, PASCOSMA -.

Elle s'articule en six étapes, et donc c'est la qualité des liens entre les étapes et les outils qui assure la cohérence de cette méthodologie.

Ces liens sont l'information partagée par les étapes successives. Plus cette information sera spécifiée et structurée, plus elle sera formalisable efficacement en objet informatique (sous forme de fichiers) et donc manipulable par des processus automatiques.

En fait, l'intégration de la méthodologie dépend directement du niveau de généralité atteint pour la formulation de l'information partagée interne (entre les étapes), et externe (entre l'utilisateur/pilote et la méthodologie).

Cet effort de structuration de l'information a déjà abouti à l'automatisation des liaisons entre certaines étapes :

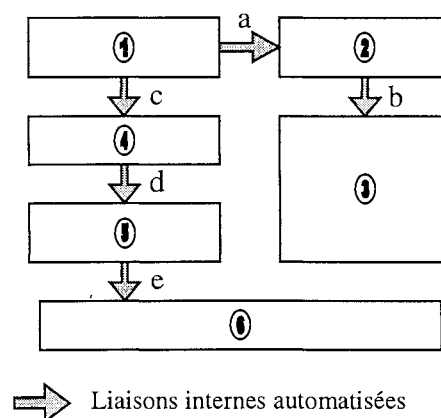


Figure IV.10 : Les liaisons internes automatisées

- a) Entre l'étape 1 de modélisation et l'étape 2 de simulation : un seul fichier générique contenant les données du point de fonctionnement initial.

- b) Entre l'étape 2 de simulation et l'étape 3 de l'analyse de la simulation : un fichier compatible Mathcad généré par le logiciel de simulation ORO.
- c) Entre l'étape 1 de modélisation et l'étape 4 de formulation du modèle de dimensionnement : un fichier générique contenant les systèmes d'état et les expressions réduites de toutes les variables du circuit.
- d) L'information nécessaire pour passer de l'étape 4 de formulation du modèle de dimensionnement à l'étape 5 de traduction en problème d'optimisation est formulée génériquement et générée automatiquement dans plusieurs fichiers (cf. §3.1.2 p. 91)
- e) Le lien entre l'étape 5 et l'étape 6 est le programme généré automatiquement d'optimisation dédié au problème de dimensionnement donné.

Ce travail nécessite de renommer toutes les grandeurs en respectant un formalisme imposé par PASCOSMA.

Exemple :  $eqnt : nt = nta+ntb+ntc+ntd$

Cela nous a obligé à générer une table de correspondance entre les variables du modèle de dimensionnement et celui utilisé dans PASCOSMA.

L'interactivité (i.e. les liaisons entre l'utilisateur et la méthodologie) participe à l'intégration de la méthodologie, dans le sens où la formulation générique de l'information externe (i.e. apportée par l'utilisateur) conditionne l'intégration de cette approche de dimensionnement dans une méthodologie globale de conception (jouant sur les structures par exemple) automatisable (emploi de systèmes experts).

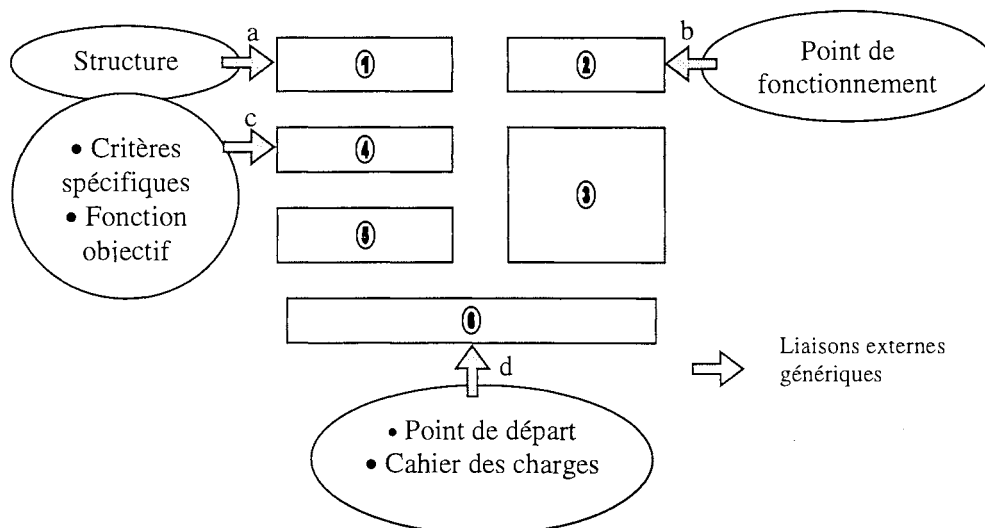


Figure IV.11 : Relations externes génériques



L'intervention du pilote se fait à quatre niveaux, et l'information fournie à chaque fois est structurée génériquement :

- a) au début de l'étape 1, la structure du convertisseur est donnée par les équations des nœuds du circuit ;
- b) au début de l'étape 2, le point de fonctionnement est saisi dans un fichier de description générique ;
- c) au début de l'étape 4, les critères spécifiques du dimensionnement sont formulés génériquement dans un fichier de formulation :

Il s'agit d'une liste de critères ayant la forme (cf. Figure V.27 du chapitre V) :

```
liste_criteres_utilisateur : [critère 1 = expression-symbolique 1,
                             critère 2 = expression-symbolique 2,
                             ...
                             ]
```

sans souci d'ordre, la construction de l'arbre de dépendance étant réalisée automatiquement ;

- d) pendant l'étape 6, les valeurs initiales et le cahier des charges peuvent être modifiés aisément, soit de façon conviviale et interactive grâce à l'interface graphique générique offerte par PASCOSMA, soit de façon automatique en agissant sur les fichiers génériques.

## 4 Conclusion

Dans ce chapitre, nous nous sommes davantage penchés sur les problèmes d'implantation de la méthodologie en termes d'outils et de techniques de calcul (retraitements formels, et calcul numérique). Il en ressort qu'une grande partie de la méthodologie est automatisable : à la fois les étapes et les liaisons internes entre ces étapes. Nous précisons quelles étapes étaient déjà automatisées au début de la mise en œuvre de notre approche, et quelle voie nous avons choisie pour automatiser les autres. Mais un grand intérêt de ce chapitre concerne les considérations que nous présentons (en termes de formulation générique du modèle de dimensionnement, d'implantation informatique, et de modifications des outils existant) pour mettre en œuvre dans notre méthodologie de dimensionnement des techniques d'intégration utilisant un minimum de calcul numérique (par exemple la méthode des *rootofs*).

Dans le chapitre suivant, nous présentons deux applications dimensionnées avec cette approche qui n'était pas encore entièrement automatisée, dans le but de montrer la pertinence de cette méthode.

*Chapitre V :*  
*Applications*



# CHAPITRE V : APPLICATIONS

## 1 Introduction

Dans ce chapitre, nous allons présenter deux problèmes de dimensionnement de convertisseurs statiques que nous avons résolus avec la méthodologie que nous venons de présenter :

- La première application traite un problème de dimensionnement pour un convertisseur multirésonant à thyristor dual forward (ZVS-MRC<sup>1</sup>), dont la structure est relativement simple, mais dont le fonctionnement est assez versatile [TANG90,92]. Nous profiterons de cette application pour illustrer la mise en œuvre de notre méthodologie de dimensionnement.
- La seconde application traite du cas d'école du hacheur série intéressant pour le CALC<sup>2</sup> qu'il contient. Nous présenterons quelques résultats de dimensionnement obtenus, et cette application sera l'occasion pour présenter les prémices de la méthode que nous envisageons pour prendre en compte les aspects discrets des composants.

## 2 Mise en garde

Les deux applications que nous présentons dans ce chapitre utilisent des modèles de dimensionnement qui ne prennent pas en compte toutes les considérations développées dans le chapitre II. En particulier, ici, les modèles de dimensionnement sont normalisés sur des ordres de grandeur qui sont considérées comme des entrées du problème (et non pas sur les valeurs maximales des signaux), tous les critères sur les tests de commutation n'ont pas été implantés (ici sont seuls pris en compte tous les tests d'amorçage et de blocage, et quelques tests de non-blocage et de non-amorçage indispensables à la viabilité du modèle ont été rajoutés), et la partie physique de la fonction objectif n'est pas normalisée. D'autre part, pour les deux applications, le modèle de dimensionnement a été construit manuellement, et les gradients ont été calculés par substitution (calcul pris en charge par PASCOSMA).

L'automatisation, et le modèle complet tels que nous les avons présentés dans ce mémoire, sont encore en phase d'implantation et de test.

---

<sup>1</sup> Zero Voltage Switching MultiResonant Converter

<sup>2</sup> Cellule d'Aide à La Commutation

Les résultats présentés dans ce chapitre doivent donc être interprétés comme une validation de la démarche, et de l'*esprit* du modèle de dimensionnement.

### 3 Une session utilisateur, application au dimensionnement du convertisseur multirésonant en thyristor dual forward

#### 3.1 Présentation du ZVS-MRC

Nous allons illustrer la mise en œuvre de la méthodologie sur un exemple de dimensionnement du convertisseur multirésonant en thyristor dual forward, dont la difficulté n'est pas tant l'obtention d'un résultat, mais plutôt la cohérence du dimensionnement obtenu face au fonctionnement observé, car ce convertisseur a la faculté (fâcheuse pour nous) de pouvoir fonctionner, à l'état permanent, dans trois modes différents [TANG92].

Le circuit complet de ce convertisseur peut être schématisé comme suit (Figure V.1):

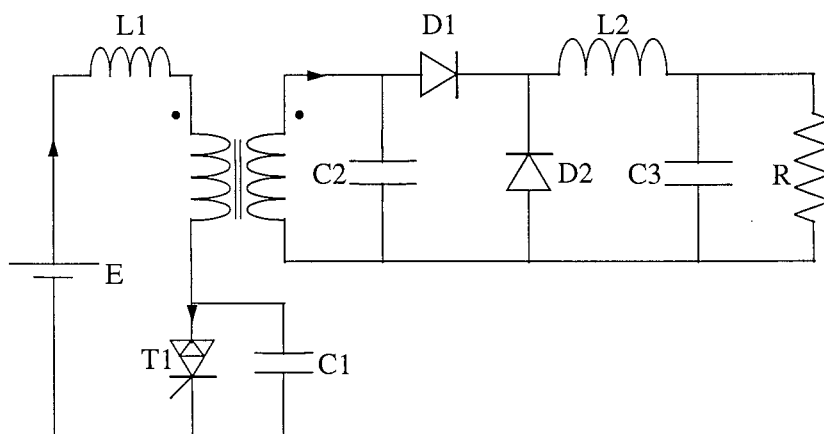


Figure V.1 : La structure du ZVS-MRC

Pour son dimensionnement, le circuit de ce convertisseur est classiquement simplifié en appliquant les hypothèses suivantes :

- 1- L'inductance de filtrage étant très grande, le filtre de sortie et la charge peuvent être représentés par une source de courant constant  $I_0$ . Par conséquent, les composants  $L_2$ ,  $R$  et  $C_3$  ne seront plus dimensionnés par le modèle que nous proposons, par contre ils peuvent être dimensionnés sur d'autres critères, notamment de filtrage, qui peuvent être formulés indépendamment.
- 2- L'inductance magnétisante du transformateur est très grande. Dans notre simplification, nous supposons cette inductance infinie, ce qui correspond à un courant magnétisant nul. Cette hypothèse acceptable pour un convertisseur forward où l'inductance magnétisante n'intervient

pas dans le fonctionnement du transformateur, ce qui n'est plus vrai dans le cas d'un convertisseur flyback.

3- Pour simplifier les expressions que nous avons dues formuler manuellement (l'automatisation n'était pas encore terminée), nous avons pris un rapport de transformation 1 pour le transformateur.

Le circuit simplifié correspondant est donc le suivant (Figure V.2) :

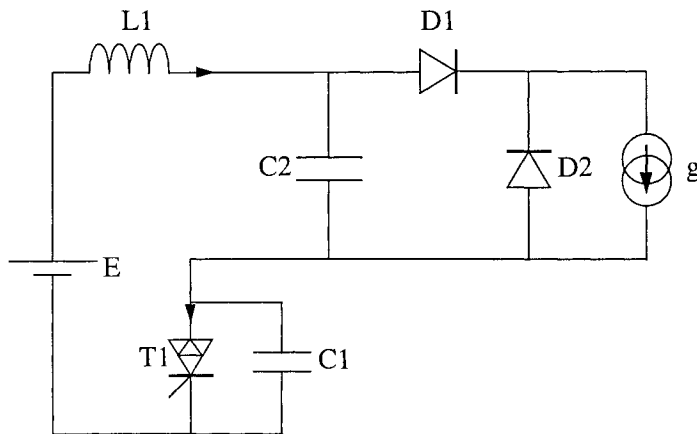


Figure V.2 : Le circuit simplifié du ZVS-MRC

Les configurations électriquement viables pour le fonctionnement de ce convertisseur sont au nombre de quatre, et caractérisées par l'état des semiconducteurs (tableau V.1) :

	Le thyristor T1 conduit	T1 ne conduit pas
D1 conduit et D2 ne conduit pas	<p>Figure V.3 : configuration C1</p>	<p>Figure V.4 : configuration C2</p>
D2 conduit et D1 ne conduit pas	<p>Figure V.5 : configuration C3</p>	<p>Figure V.6 : configuration C4</p>

tableau V.1 : Les quatre configurations observées par Tang pour le fonctionnement du ZVS-MRC

- configuration C1 (Figure V.3) : le thyristor T1 et la diode D1 conduisent,
- configuration C2 (Figure V.4) : la diode D1 conduit seule,
- configuration C3 (Figure V.5) : le thyristor T1 et la diode D2 conduisent,
- configuration C4 (Figure V.6) : la diode D2 conduit seule.

Fondé sur ces quatre configurations, le fonctionnement du convertisseur multirésonant forward accepte trois modes (Figure V.7, Figure V.8, et Figure V.9) :

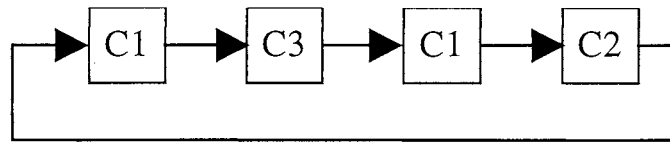


Figure V.7 : Mode 1

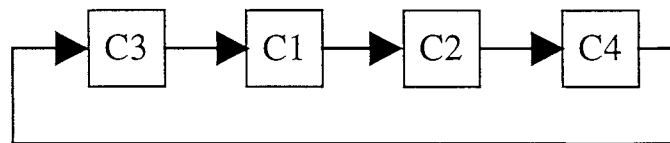


Figure V.8 : Mode 2

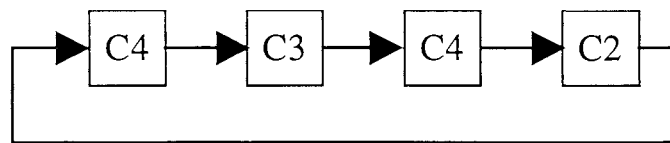


Figure V.9 : Mode 3

### 3.2 Une session utilisateur pour dimensionner le ZVS-MRC avec l'approche que nous avons mise en œuvre.

Trois tâches distinctes doivent être accomplies sur la méthodologie, et donc l'acteur sera tour à tour :

- EXPERT pour mener à bien une première tâche de modélisation du problème de dimensionnement,
- SIMPLE UTILISATEUR d'outils informatiques mis à sa disposition pour une seconde tâche de génération de l'outil de dimensionnement dédié,
- CONCEPTEUR dans une troisième et dernière tâche qui est la session de dimensionnement à proprement dite, fondée sur l'utilisation de l'outil dédié.

Nous allons détailler ces trois tâches dans la suite du chapitre.

#### 3.2.1 MODELISATION

Le travail de modélisation du point de vue utilisateur touche les quatre premières étapes de la méthodologie.

Le but de ce travail est de mettre en place la connaissance nécessaire en vue de la construction du modèle de dimensionnement. Cette tâche qui fait intervenir de l'expertise, ne peut pas être automatisée par les méthodes classiques de programmation procédurale, aussi, en attendant son

éventuelle prise en charge par un système expert, nous présentons les 11 actions fondamentales nécessaires pour mener à bien ce travail.

### 3.2.1.1 Description du circuit (action n°1)

D'après le circuit de la structure, l'information que doit fournir l'utilisateur est :

- la nature des composants, leur nombre, et la manière de les identifier (Figure V.10),

```

/*
  Declaration des composants actifs
*/
interrupteurs: [[thyristor_dual,t,1], [diode,d,2]]$

/*
  Declaration des composants passifs
*/
identificateur_des_inductances:l$
nombre_d_inductances:1$

identificateur_des_condensateurs:c$
nombre_de_condensateurs:2$

nombre_de_resistances:1$

```

**Figure V.10 : La déclaration des composants pour MEIGE**

**Remarque :** Dans cette description du circuit, nous insérons une résistance R1 en parallèle avec la source de courant qui modélise la charge. Elle permet de résoudre des problèmes de démarrage pour la simulation. Sa valeur est très élevée ( $\sim 10^9 \Omega$ ), et elle n'influe pas sur les résultats de simulation obtenus à l'état permanent, aussi cette résistance R1 n'apparaît pas dans le modèle de dimensionnement.

- et les équations des nœuds du circuit (Figure V.11).

```

/*
  Equations des noeuds du circuit
*/
equations_du_systeme:
[
  i_e=i_t1+i_c1,
  i_t1+i_c1+i_d2=i_c2+i_g+i_r1,
  i_d1+i_d2=i_g+i_r1,
  i_l1=i_c2+i_d1,
  i_e=i_l1
]$

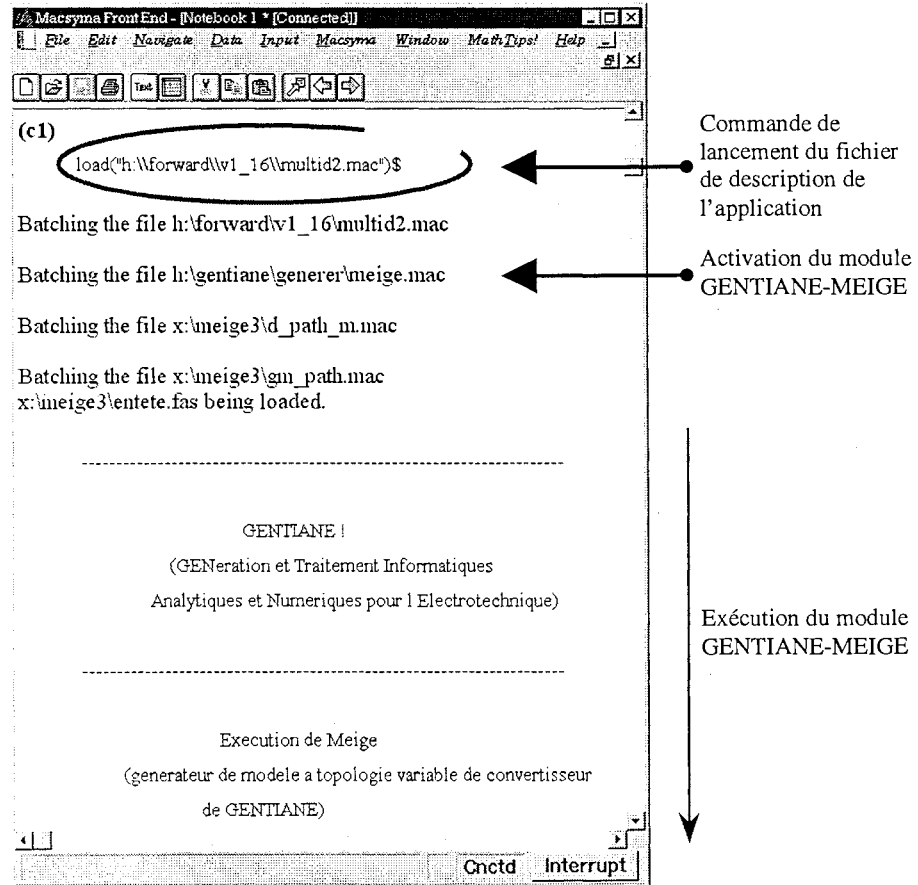
```

**Figure V.11 : La déclaration des équations des nœuds pour MEIGE**



### 3.2.1.2 Lancement de GENTIANE (action n°2)

Ce fichier est la seule information nécessaire pour réaliser une modélisation par l'outil GENTIANE. Dans l'environnement de Macsyma, on charge ce fichier (Figure V.12) qui effectue toutes les déclarations, et lance le module GENTIANE-MEIGE de construction du modèle à topologie :



**Figure V.12 : Lancement de GENTIANE-MEIGE dans l'environnement Macsyma (pour le fichier multid2.mac)**

Dans un premier temps, le module GENTIANE-MEIGE nous fournit le modèle de fonctionnement à topologie variable, sous la forme de trois fichiers.

- Le premier (Figure V.13) contient la correspondance entre les numéros, et les configurations qu'ils identifient :

configuration	oro	semiconducteur
1	1	
2	2	t_1
3	<b>3</b>	<b>d_1</b>
4	<b>4</b>	<b>t_1 d_1</b>
5	<b>5</b>	<b>d_2</b>
6	<b>6</b>	<b>t_1 d_2</b>
7	7	d_1 d_2
8	8	t_1 d_1 d_2

Figure V.13 : Le fichier GENTIANE-MEIGE d'identification des configurations

Nous avons reporté en gras les configurations observées par Tang. Ci-dessous, le Tableau V.2 présente les correspondances entre les configurations identifiées par un numéro-MEIGE (en simulation) et la notation précédemment utilisée :

Numéro-MEIGE	TANG	Configuration
3	C2	D1 conduit seule
4	C1	T1 et D1 conduisent
5	C4	D2 conduit seule
6	C3	T1 et D2 conduisent

Tableau V.2 : Correspondance entre les identifications des configurations

- Le second fichier contient les lois comportementales de la partie puissance du convertisseur, c'est à dire les systèmes d'équations différentielles vérifiées par les variables d'état, et les formules pour calculer les autres variables du circuit (non variables d'état, par exemple le courant dans un condensateur).
- Les systèmes d'état sont écrits dans un format générique, et cela pour chaque configuration électriquement viable du circuit. Dans la Figure V.14, nous avons extrait les quatre systèmes qui correspondent aux quatre configurations qui sont observés par Tang [TANG90,92].

```
void pmultid2 :: calpb3 (void)
{
    di_l1_dt = -((v_c2+v_c1-e)/l_1);
    dv_c1_dt = i_l1/c_1;
    dv_c2_dt = -((v_c2+(-i_l1+g)*r_1)/c_2/r_1);
}
void pmultid2 :: calpb4 (void)
{
    di_l1_dt = -((v_c2-e)/l_1);
```

```

    dv_c2_dt = -((v_c2+(-i_l1+g)*r_1)/c_2/r_1);
}
void pmultid2 :: calpb5 (void)
{
    di_l1_dt = -((v_c2+v_c1-e)/l_1);
    dv_c1_dt = i_l1/c_1;
    dv_c2_dt = i_l1/c_2;
}
void pmultid2 :: calpb6 (void)
{
    di_l1_dt = -((v_c2-e)/l_1);
    dv_c2_dt = i_l1/c_2;
}

```

**Figure V.14 : Les systèmes d'état générés par GENTIANE-MEIGE pour les configurations qui nous intéressent**

- Les formules pour calculer les autres variables du circuit sont elles aussi écrites sous un format bien défini. La Figure V.15 ci-dessous présente leur forme pour les quatre configurations qui nous intéressent :

```

case 3 :

    v_l1 = -v_c2-v_c1+e;
    i_c2 = -((v_c2+(-i_l1+g)*r_1)/r_1);
    i_r1 = v_c2/r_1;
    u_e = v_l1+v_c2+v_c1;
    i_c1 = i_l1;
    v_r1 = v_c2;
    i_g = -i_r1+i_l1-i_c2;
    i_e = i_l1;
    u_g = -v_r1;
    break;

case 4 :
    v_l1 = -v_c2+e;
    i_c1 = 0;
    i_c2 = -((v_c2+(-i_l1+g)*r_1)/r_1);
    i_r1 = v_c2/r_1;
    dv_c1_dt = 0;
    v_r1 = v_c2;
    i_g = -i_r1+i_l1-i_c2;
    i_e = i_l1;
    u_e = v_r1+v_l1;
    u_g = -v_r1;
    break;

case 5 :
    i_g = g;
    v_l1 = -v_c2-v_c1+e;
    i_r1 = 0;
    i_c1 = i_l1;
    u_e = v_l1+v_c2+v_c1;
    i_c2 = i_l1;
    i_e = i_l1;

```

```

v_r1 = 0;
u_g = 0;
break;

case 6 :
i_g = g;
v_l1 = -v_c2+e;
i_c1 = 0;
i_r1 = 0;
dv_c1_dt = 0;
u_e = v_l1+v_c2;
i_c2 = i_l1;
i_e = i_l1;
v_r1 = 0;
u_g = 0;
break;

```

**Figure V.15 : Les équations des variables du circuit non variables d'état pour les configurations qui nous intéressent**

- Le troisième fichier contient :
  - Les fonctions de calcul des variables concernant les interrupteurs en fonction des variables d'état (Figure V.16), et la partie commande, c'est à dire les tests de commutation qui régissent le passage d'une configuration à l'autre pendant le fonctionnement - c'est une modélisation sans a priori -.

```

case 3 :
v_t1 = -v_r1-v_l1+u_e;
i_d1 = i_r1+i_g;
v_d2 = -v_r1;
break;

case 4 :
i_t1 = i_l1-i_c1;
i_d1 = i_r1+i_g;
v_d2 = v_l1-u_e;
break;

case 5 :
v_t1 = v_c1;
v_d1 = -v_l1-v_c1+u_e;
i_d2 = i_r1+i_g;
break;

case 6 :
i_t1 = i_l1-i_c1;
v_d1 = -v_l1+u_e;
i_d2 = i_r1+i_g;
break;

```

**Figure V.16 : Les équations des signaux de command générés par MEIGE pour les configurations qui nous intéressent**

Les équations sont aussi reformulables pour un mode donné dans un format qui permet une exploitation directe des équations (en Macsyma) pour construire le modèle de dimensionnement.

Dans un deuxième temps, le module GENTIANE-ORO nous permet de réaliser une simulation dédiée à l'application étudiée (Figure V.18), en utilisant le modèle de fonctionnement issu de GENTIANE-MEIGE, et un fichier dans lequel on précise les données du point de fonctionnement (cf. Figure V.17 ci-dessous).

```
rem:Definition des parametres

AP:Gachette,frequence,357200
AP:Gachette,alpha,0.69
AP:Convertisseur,l_1,0.000037
AP:Convertisseur,c_1,1.88e-9
AP:Convertisseur,c_2,1.77e-8
AP:Convertisseur,e,48.0
AP:Convertisseur,r_1,1.0e9
AP:Convertisseur,ri,0.0
AP:Convertisseur,g,0.5
```

**Figure V.17 : Un extrait du fichier de description du point de fonctionnement pour GENTIANE-ORO**

```
M$ Invite de commandes - forwd2_w11_v16_nt
8 : Annuler la commande
Reponse (par defaut 1):
La simulation est maintenant prete.
Voici l'ordre dans lequel vont etre effectues les calculs...
Traitement de la boucle Puissance qui comprend dans l'ordre les outils :
Gachette Configuration Convertisseur Chron
Le pas de la boucle par defaut est de 5E-008 seconde(s)
La date de debut de simulation est : 0 seconde(s).
Entrez à présent la date de fin de simulation, en secondes :
Reponse (par defaut 0.01): 0.01
La simulation est lancée. Veuillez patienter ...
Presser sur la touche s pour arreter
Instant courant : 0.00368424
```

**Figure V.18 : Un lancement de la simulation GENTIANE-ORO**

La simulation de GENTIANE-ORO est rapide car elle est fondée sur un modèle à topologie variable, et elle réalise une adaptation dynamique du pas de calcul (en utilisant le principe des PDI<sup>3</sup> [GERBAUD97]). Ainsi, 10 millisecondes de fonctionnement de ce convertisseur (dont la fréquence est 350 kHz) sont simulées en 172 secondes avec le module GENTIANE-ORO sur un Pentium cadencé à 75 MHz avec 32 Mo de RAM. (il faut diviser ces temps de simulation par 4 pour obtenir les performances atteintes sur un Pentium 200 256 Mo).

De plus, ce module nous fournit un fichier de points au format compatible avec beaucoup de logiciel d'analyse (Excel, Mathcad, Matlab, MICCAO).

### 3.2.1.3 Analyse (action n°3)

On lance la feuille de calcul Mathcad (cf. Figure V.19) qui utilise le fichier de points généré par le simulateur GENTIANE-ORO dans l'étape précédente.

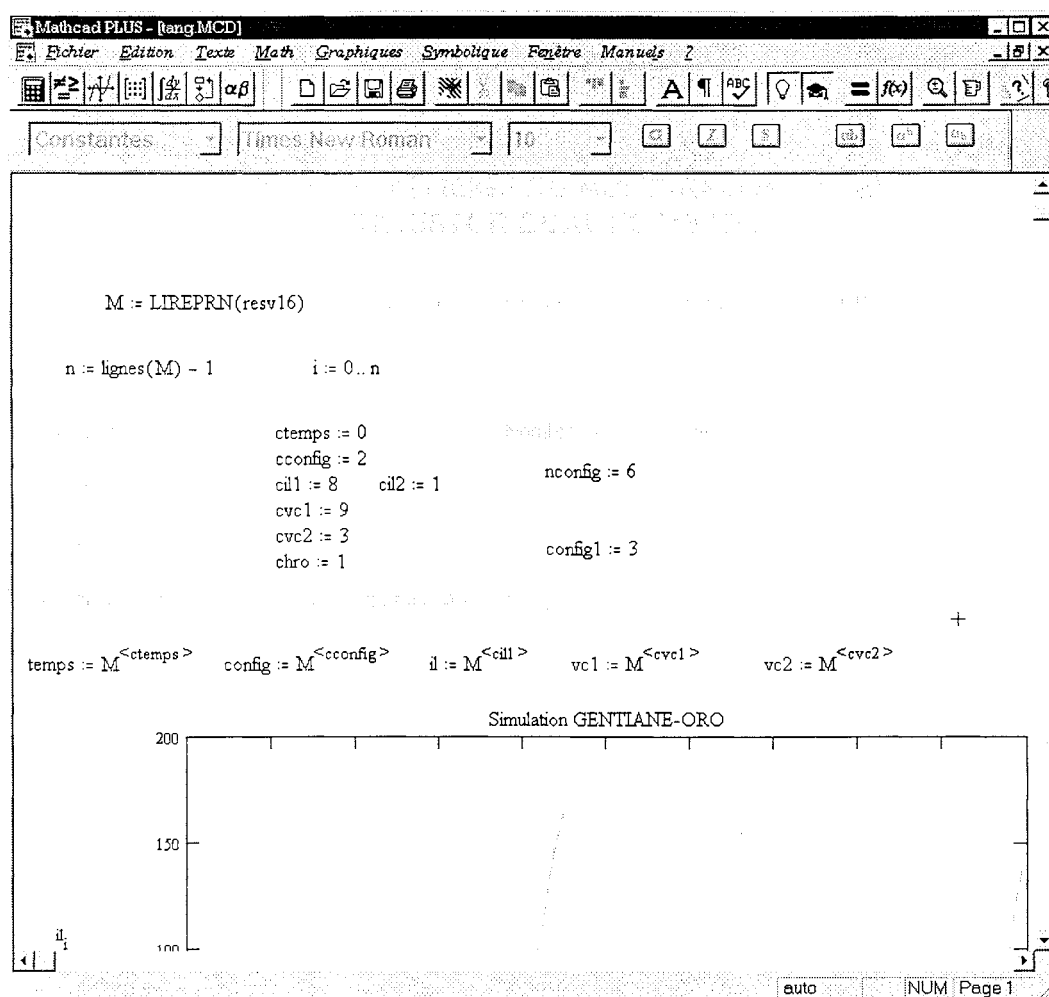
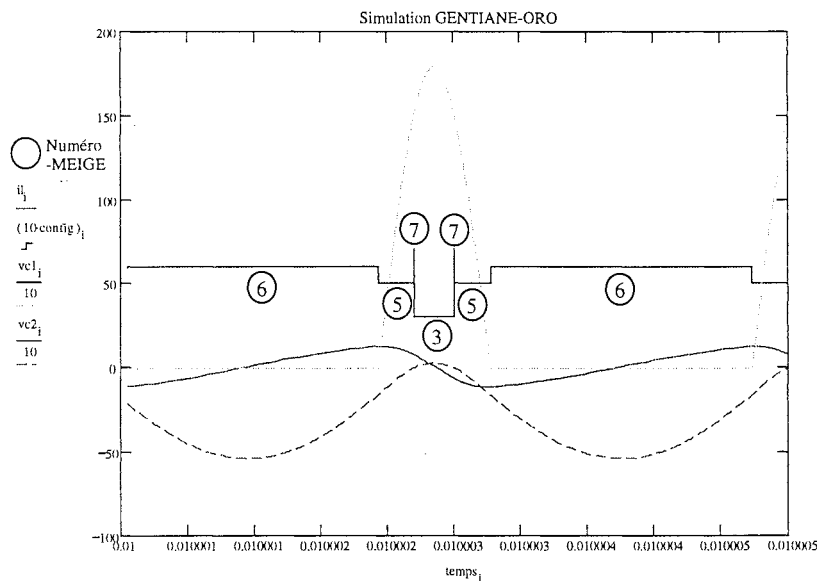


Figure V.19 : Un extrait de la feuille de calcul Mathcad 6+

<sup>3</sup> Prochaine Date Importante

A ce stade, nous utilisons le logiciel Mathcad comme un simple tableur. La feuille de calcul que nous avons développée dans le langage de programmation de Mathcad 6 nous permet de réaliser l'analyse de la simulation à deux niveaux :

- Cette feuille générique de calcul nous permet d'obtenir aisément une visualisation graphique des courbes représentant l'évolution des courants et tensions pour tous les composants du circuit (cf. Figure V.20). Grâce aux capacités graphiques de Mathcad, il est possible de choisir les courbes à afficher, de donner des échelles, de construire plusieurs graphes, de choisir leurs dimensions..., et donc, ainsi, de personnaliser son analyse.



**Figure V.20 : Les courbes représentatives de quelques signaux du fonctionnement du ZVS-MRC tracées avec notre feuille de calcul Mathcad**

**Remarque :** La visualisation de ces courbes permet aussi, dans une certaine mesure, de vérifier la validité des simplifications effectuées sur le circuit.

- Les quelques procédures génériques de calcul que nous avons programmées dans le langage Mathcad nous permettent d'extraire les données utiles du tableau de points, et ainsi la feuille de calcul nous fournit un tableau de données brutes sur le fonctionnement du convertisseur simulé (resultat) :

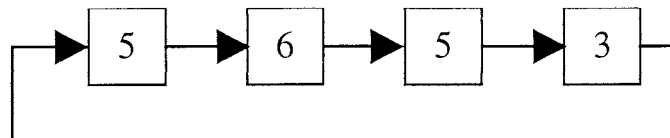
$$\text{resultat} = \begin{bmatrix} 5 & 2.69 \cdot 10^{-7} & 0.096 & 11.982 & 0 & -159.153 \\ 7 & 1 \cdot 10^{-11} & 3.572 \cdot 10^{-6} & 6.958 & 1.502 \cdot 10^3 & 0 \\ 3 & 3.01 \cdot 10^{-7} & 0.108 & 6.958 & 1.502 \cdot 10^3 & 0 \\ 7 & 1 \cdot 10^{-11} & 3.572 \cdot 10^{-6} & -6.635 & 1.53 \cdot 10^3 & 0 \\ 5 & 2.774 \cdot 10^{-7} & 0.099 & -6.635 & 1.53 \cdot 10^3 & 0 \\ 6 & 1.952 \cdot 10^{-6} & 0.697 & -11.953 & 0 & -162.571 \end{bmatrix}$$

qu'il faut lire avec les entrées :

Numéro-MEIGE	Durée [s]	Durée normalisée	courant dans L1 [A]	tension aux bornes de C1 [V]	tension aux bornes de C2 [V]

**Tableau V.3 : Tableau de données brutes fourni par notre feuille de calcul Mathcad 6+**

Donc, de le Tableau V.3 ci-dessus, on peut extraire l'enchaînement des configurations identifiées par leur numéro-MEIGE (Figure V.21), et ainsi obtenir une formulation générique du mode de fonctionnement du convertisseur pour le point de fonctionnement initial (cf. Figure V.24). On reconnaît le mode 3 (Figure V.9) décrit par Tang dans [TANG92] :



**Figure V.21 : Mode 3 identifié par les numéros-MEIGE**

**Remarque :** Deux configurations numérotées 7, imprévues par la théorie, viennent s'intercaler autour de la configuration 3. Elles correspondent à une phase de transition où les deux diodes conduisent simultanément, entre la configuration C2 (3) pendant laquelle seule la diode D1 conduit, et celle C4 (5) où seule D2 conduit. Du fait de leur durée très brève, nous avons décidé de ne pas prendre en compte ces configurations dans le modèle de dimensionnement.

A partir des courbes de la Figure V.20 ci-dessus, une première analyse qualitative permet de noter les points critiques du fonctionnement (courants et tensions crêtes, montées et descentes abruptes des signaux), et cette analyse sert aussi à déterminer certaines contraintes et objectifs qui devront être pris en compte pour le dimensionnement. Pour ces raisons, l'information qualitative ainsi extraite de cette analyse doit être fournie à l'étape de génération du modèle de dimensionnement.



Par exemple, pour le cas du convertisseur multirésonant qui nous intéresse, l'observation de l'évolution de la tension aux bornes du condensateur C1 montre un pic très important (~2000 V). Pour notre problème de dimensionnement, nous décidons de déclarer la valeur du maximum de cette tension comme un critère à minimiser.

### 3.2.1.4 Test de l'intégration (action n°3)

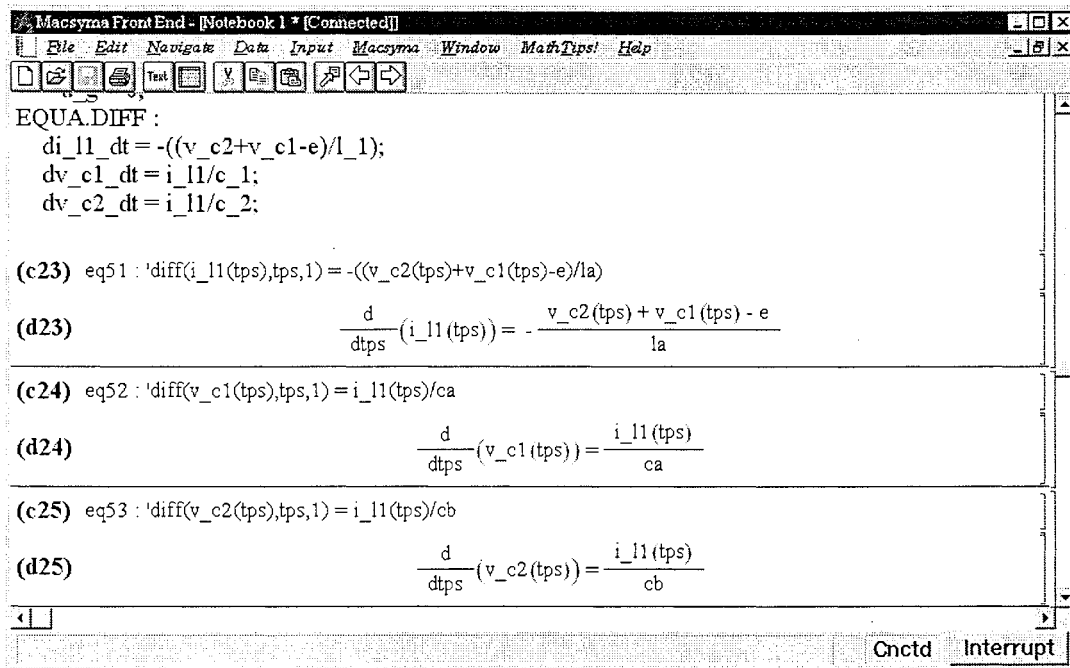


Figure V.22 : Les équation différentielles du système d'état de la configuration 5 dans l'environnement Macsyma

Nous avons développé un module programmé avec le langage symbolique de Macsyma qui s'intègre dans l'étape 4 de la génération du modèle de dimensionnement, mais qui peut aussi être lancé en marge de la méthodologie (cf. Figure V.22), en particulier pour tester l'intégration formelle des systèmes d'équations différentielles, car le succès de ce calcul est une condition nécessaire pour la bonne marche de la suite de la méthodologie. En effet, deux cas de figures peuvent se présenter à la suite d'un test d'intégration symbolique sur les systèmes d'état du mode de fonctionnement observé pour le convertisseur étudié :

- On observe l'échec de l'intégration symbolique pour un ou plusieurs des systèmes d'état, auquel cas trois actions sont possibles :
  - On peut décider de simplifier le circuit afin de supprimer une variable d'état, ce qui impose de recommencer la session utilisateur depuis la première action : il faut décrire le circuit simplifié,

relancer GENTIANE, et faire une nouvelle analyse en particulier pour vérifier la validité des simplifications effectuées.

- On peut souhaiter ne réaliser la simplification que pour le ou les systèmes d'état provoquant l'échec de l'intégration formelle, auquel cas il faut aussi reprendre les actions 3.2.1.1, 3.2.1.2 et 3.2.1.3.
- On peut préférer garder le circuit complet et réaliser une intégration faisant intervenir des *rootofs* avec le module d'intégration Maple (cf. chap. IV). Ce choix influe sur la suite de la session dans la mesure où certains critères ne seront plus formulables dans le modèle de dimensionnement (les critères faisant intervenir les pulsations). A l'heure actuelle, la possibilité de construire le modèle de fonctionnement sur les variables d'état dépendant des *rootofs* numériques n'est pas encore totalement finalisée, et nous ne connaissons pas encore les performances de cette méthode.
- L'intégration symbolique est un succès (Figure V.23), et dans ce cas on peut passer à l'action suivante, car on est alors sûr de pouvoir construire l'ensemble du modèle de dimensionnement.

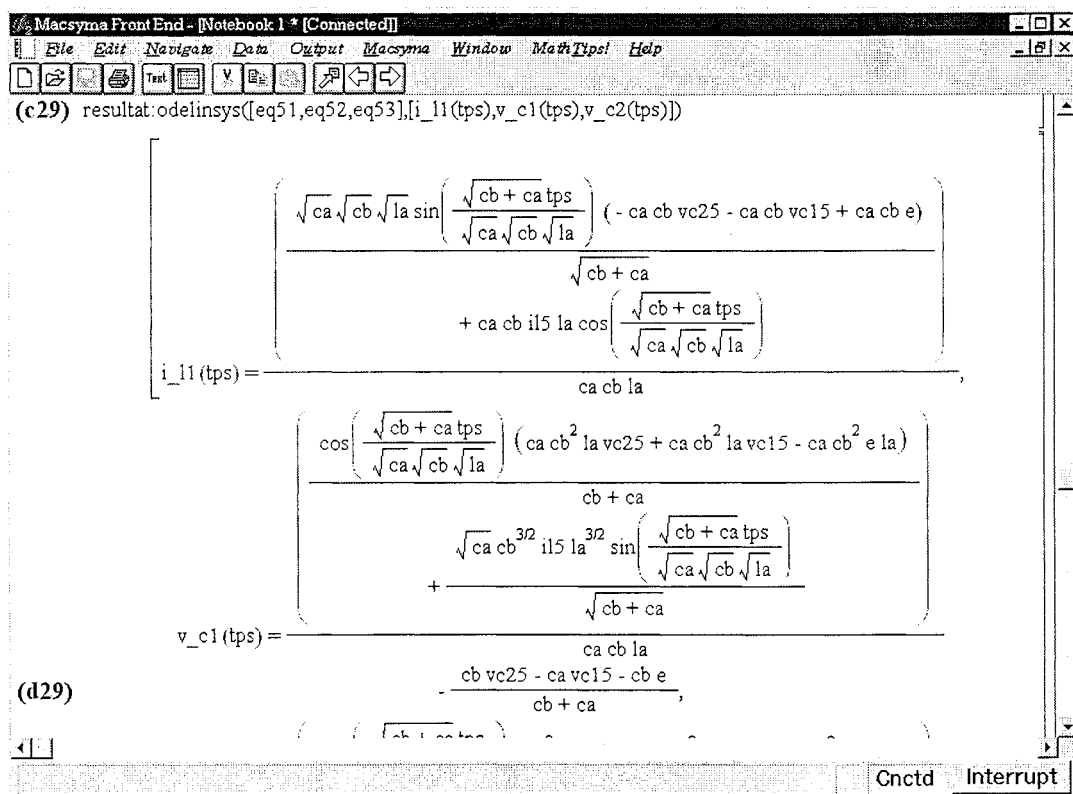


Figure V.23 : L'intégration symbolique pour le système 5 réalisée par Macsyma est un succès

### 3.2.1.5 Déclaration du modèle de dimensionnement (action n°5)

C'est à ce niveau qu'il faut fournir à l'outil de construction du modèle de dimensionnement l'information nécessaire pour mener à bien la génération.

Le modèle que nous présentons pour cette application a été construit manuellement, par conséquent toutes ces informations n'ont pas été formalisées pour cet exemple.

Aussi, à titre d'illustration, nous présentons ici des extraits du fichier de déclaration d'un modèle de dimensionnement tel qu'il sera écrit pour le générateur automatique en cours de développement. Toute la description est réalisée à base de listes dont voici la formulation pour chaque élément de l'ensemble des données génériques :

#### 3.2.1.5.1 Mode de fonctionnement

Nous écrivons ceci sous la forme d'une liste

```
/* mode de fonctionnement */
mode:[5,6,5,3]$
```

Figure V.24 : Le mode de fonctionnement est déclaré sous la forme d'une liste

#### 3.2.1.5.2 Emplacements des extremums des courants et tensions pour chaque composant

Dans une liste représentant chaque configuration successive dans l'ordre [5,6,5,3], on note :

+1 si un extremum positif est atteint pendant cette configuration,

-1 si un extremum négatif est atteint pendant cette configuration,

0 si aucun extremum n'est atteint pendant cette configuration

T1 et C1 sont en parallèle, la tension maximale à leur bornes ainsi que l'intensité du courant qui les traverse sont égales. Ce type d'expertise peut être déclarée dans le fichier en affectant les listes `liste_i_maxi_egaux` et `liste_v_maxi_egaux`.

```
/* liste des emplacements des extremums de courants */
liste_i_maxi      : [
                    [[L,1],[-1,0,+1,0]],
                    [[C,1],[-1,0,+1,0]],
                    [[C,2],[-1,0,+1,0]],
                    [[D,1],[0,0,0,+1]],
                    [[D,2],[+1,0,0,0]]
];

/* liste des maximums identiques */
liste_i_maxi_egaux: [[T,1],[C,1]];
```

```

/* liste des emplacements des extremums de tensions */
liste_v_maxi      : [
                    [[C,1],[0,0,0,+1]],
                    [[C,2],[0,-1,0,+1]],
                    [[L,1],[0,+1,0,-1]],
                    [[D,1],[0,-1,0,0]],
                    [[D,2],[0,0,0,-1]]
                    ];

/* liste des maximums identiques */
liste_v_maxi_egaux : [[[T,1],[C,1]]];

```

Figure V.25 : Les listes de déclaration des extremums

### 3.2.1.5.3 Composants dont on veut calculer les courants et tensions moyennes ou efficaces

L'utilisateur sélectionne les grandeurs qu'il souhaite considérer (Figure V.26) :

```

/* liste des composants dont on veut calculer le courant moyen
qu'ils traverse */
liste_comp_moy_i   : [[L,1],[C,1],[C,2],[T,1],[D,1]];

/* liste des composants dont on veut calculer la tension moyenne
a leurs bornes */
liste_comp_moy_v   : [[L,1],[C,1],[C,2],[T,1],[D,1]];

/* liste des composants dont on veut calculer le courant efficace
qui les traverse */
liste_comp_eff_i   : [[L,1]];

/* liste des composants dont on veut calculer la tension efficace
a leurs bornes */
liste_comp_eff_v   : [[C,2]];

```

Figure V.26 : Les listes de déclaration des valeurs moyennes et efficaces à formuler

### 3.2.1.5.4 Quelques exemples de critères utilisateurs

L'utilisateur écrit les formules permettant de caractériser certains composants ou propriétés (Figure V.27). Il n'a pas à se soucier de l'ordre dans lequel il déclare ses critères, une procédure automatique se charge de ce travail pour suivre l'ordre ascendant de l'arbre de composition.

```

/* liste des equations des criteres que desire contraindre le
designer */
liste_eq_uti:[

/* pertes (expressions approchees) */
Pfer = KA*Fd^(1/3)*Bm^2,
Pcu  = KB*Fd*Idaeff^2,

/* puissances */

```

```
Palim = Ea*I1bmoy,  
Ps     = Vs*idbmoy,  
  
rend   = Ps/Palim,  
  
/* tension de sortie moyenne */  
Vs     = vcbmoy  
  
];
```

Figure V.27 : La liste des critères utilisateur

### 3.2.1.5.5 Un exemple de fonction objectif

Finalement, l'utilisateur décrit sa fonction objectif comme une somme de critères-objectifs pondérés (cf. Figure V.28)

```
/* Fonction objectif utilisateur (de dimensionnement) */  
fobj      : Pa*Pfer+Pb*Pcu+Pc*Idaeff+Pd*Idamax$
```

Figure V.28 : La déclaration de la fonction objectif

## 3.2.2 Génération de l'outil de dimensionnement dédié

Cette tâche consiste à construire le modèle symbolique de dimensionnement (§ 3.2.2.1) tel que nous l'avons présenté dans le chapitre II, puis à générer l'outil final de dimensionnement dédié à notre problème fondé sur ce modèle (§ 3.2.2.2). Lorsque l'étape de la génération du modèle de dimensionnement sera entièrement automatisée, pour cette tâche de construction de l'outil final dédié au problème de dimensionnement spécifique, l'acteur sera alors un simple utilisateur de deux outils informatiques à exécuter successivement.

### 3.2.2.1 Lancement de la génération du modèle de dimensionnement (action n°6)

Cette tâche est aujourd'hui en grande partie automatisée, mais pour l'exemple étudié, le modèle de dimensionnement a été formulé manuellement sur la base d'une partie des critères génériques présentés dans le chapitre II. Ce travail a demandé plusieurs jours en comptant les erreurs d'écriture qui émaillent nécessairement ce genre de travail fastidieux et répétitif (d'où l'intérêt, et la nécessité à un certain point, de l'automatisation en cours actuellement). Nous présentons le fichier de description de ce modèle en annexe C.

A noter que pour des contraintes de compatibilité avec PASCOSMA, nous devons renommer toutes les variables utilisées selon un format imposé (en particulier les chiffres sont exclus, et les noms ne doivent pas dépasser 7 caractères). Ainsi C1 sera appelé *ca*. Cf. annexe D pour la table de correspondance entre les paramètres définis au chapitre II et les paramètres PASCOSMA.

Le critère principal pour lequel nous réalisons le dimensionnement est la valeur maximale de la tension aux bornes du condensateur C1. La Figure V.29 ci-dessous présente la formulation de ce critère, ici réalisée manuellement.

```

/* TENSION MAXI */
/* positive aux bornes de la capacite C1 : pendant la config 4 [MEIGE 3] */

eqtvcamax: tgvca=diff(ratsubst(tvcamax, tps, eqdvca), tvcamax) * tvcamax;

eqntvcamax : tvcamax=ntvcama*td;
eqvcamax : vcamax=ratsubst(tvcamax, tps, eqdvca);

```

**Figure V.29 : La formalisation du critère de la valeur crête de la tension aux bornes du condensateur C1**

### 3.2.2.2 Lancement de la génération PASCOSMA (action n°7)

Une fois le modèle de dimensionnement construit, PASCOSMA se charge de la génération de l'outil de dimensionnement dédié. Pour cela, il faut lui fournir le fichier de description du modèle analytique dans un format spécifique. A partir de ce fichier, PASCOSMA fait le lien avec la procédure d'optimisation, et il génère une interface utilisateur spécifique à notre problème de dimensionnement. De plus, dans cette version, PASCOSMA se charge aussi du calcul des gradients par substitution. La génération totale de l'outil de dimensionnement sur un Pentium 100 MHz 32 Mo a duré 6 heures et demi.

### 3.2.3 Session de dimensionnement

Cette dernière tâche s'appuie sur l'utilisation de l'outil de dimensionnement dédié au problème de dimensionnement étudié (cf. Figure V.30 : L'interface de l'outil de dimensionnement dédiée générée par PASCOSMA).

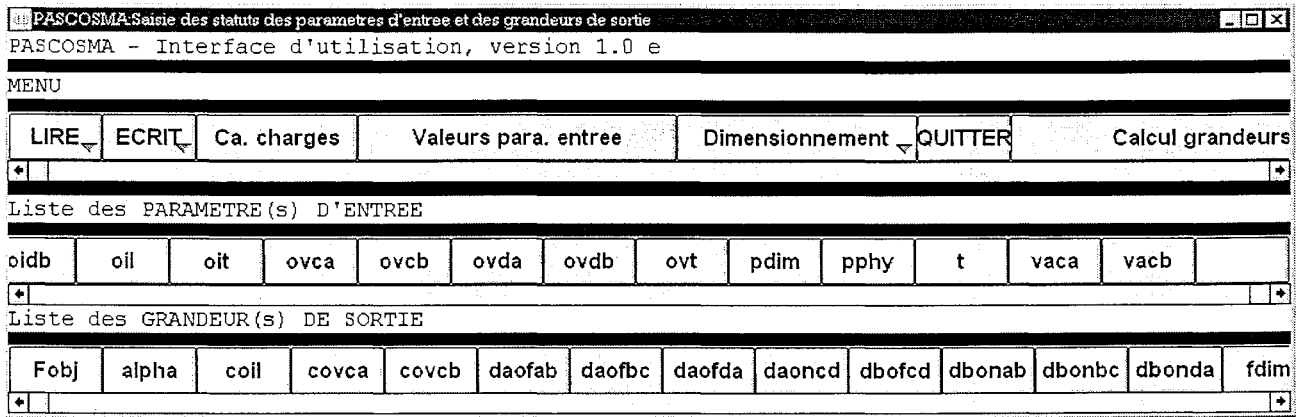


Figure V.30 : L'interface de l'outil de dimensionnement dédiée générée par PASCOSMA

### 3.2.3.1 Saisie du point de départ (action n°8)

Il est nécessaire d'initialiser l'algorithme d'optimisation par un jeu de valeurs pour les paramètres d'entrée, et celles-ci doivent être données avec suffisamment de précision pour être physiquement cohérente, c'est-à-dire correspondre au mode de fonctionnement figé sur lequel est fondé l'ensemble du modèle de dimensionnement.

Pour être certain d'initialiser correctement l'outil, nous proposons d'extraire les données nécessaires de la simulation effectuée précédemment dans un but d'analyse du fonctionnement. Pour cela, dans le programme Mathcad que nous avons développé (action n°3), nous avons mis en place un certain nombre de procédures dont le rôle est d'extraire les valeurs des entrées non accessibles directement par le concepteur (les entrées secondaires cf. chap. II) à partir de la simulation. Toutes les procédures d'extraction nécessaires ne sont pas encore programmées, toutefois le Tableau V.3 nous fournit les durées normalisées des configurations, et la condition initiale pour chaque variable d'état au début du mode. La Figure V.31 ci-dessous est une fenêtre affichée dans l'interface de PASCOSMA qui nous permet de visualiser le point de départ que nous avons saisi.

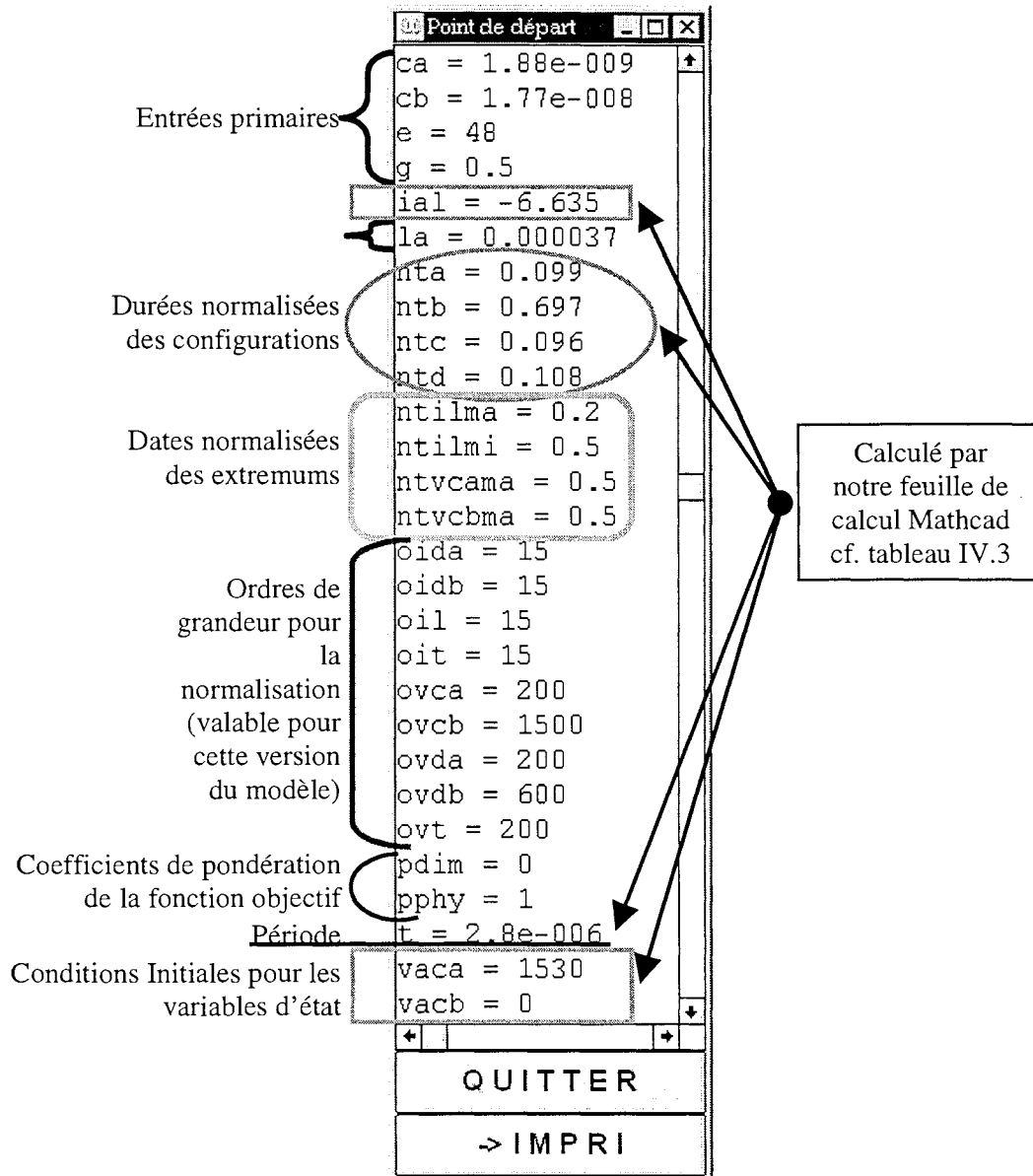


Figure V.31 : Le point de départ de l'optimisation (cf. annexe D)

D'autre part, l'initialisation des paramètres d'entrée peut être réalisée à partir des valeurs obtenues après un précédent dimensionnement, lorsque le concepteur est amené à procéder à des rebouclages afin d'affiner ses résultats, ou pour prendre en compte l'aspect discret du choix des composants.

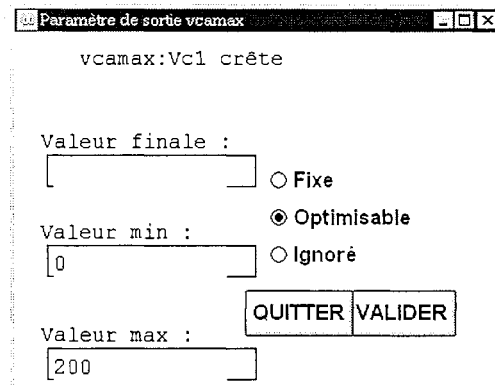
**Remarque :** Nous rappelons que les grandeurs ont dû être renommées par rapport aux notations utilisées dans le modèle de dimensionnement (cf. annexe D pour la table de correspondance).

### 3.2.3.2 Etablissement du cahier des charges (action n°9)

L'affectation des contraintes sur les paramètres du dimensionnement est facilitée par l'interface qui permet de modifier très aisément les données que l'on fournit à l'outil. Ainsi, à tout instant pendant



la session de dimensionnement, il est possible de relaxer ou de restreindre chaque paramètre du dimensionnement, en agissant de façon interactive sur les paramètres de l'interface (cf. une fenêtre de saisie ci-dessous, Figure V.32).



**Figure V.32 : Fenêtre de saisie de la contrainte sur le critère de la valeur crête de la tension aux bornes de C1**

De cette façon, on donne notre cahier des charges pour le dimensionnement simple que l'on veut réaliser sur notre structure simplifiée du ZVS-MRC, dans le mode de fonctionnement figé (mode 3).

Les points importants de ce cahier des charges sont :

- On désire diminuer la tension crête aux bornes du condensateur C1 à une valeur comprise entre 0 et 200 Volts, pour cela on contraint le critère  $vcamax$  sur l'intervalle [0 ; 200] (cf. Tableau V.4).
- Mais on ne veut pas trop descendre en fréquence, pour cela on contraint la période à rester dans une plage de valeurs comprises entre 1  $\mu$ s et 0,1 ms.

Pour mener à bien ce dimensionnement, voici le cahier des charges que nous avons saisi (Tableau V.4) :

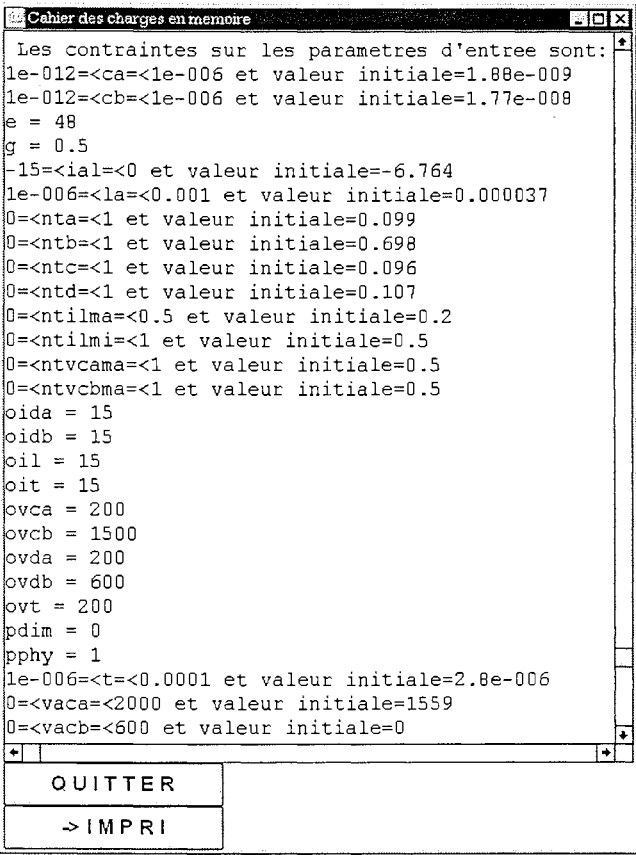
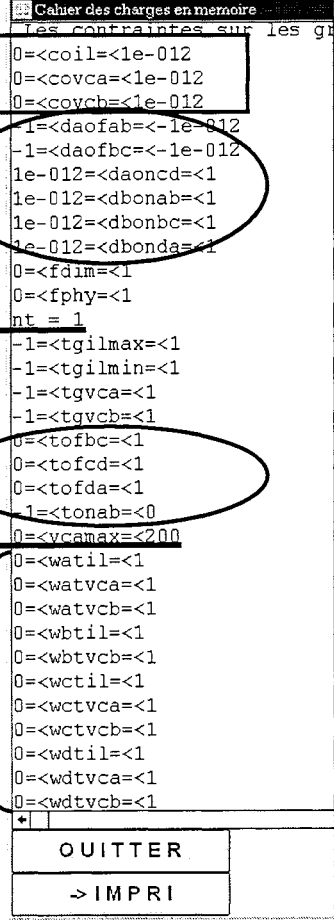
CAHIER DES CHARGES	
Contraintes sur les paramètres d'entrée	Contraintes sur les paramètres de sortie
 <pre> Les contraintes sur les parametres d'entree sont: 1e-012=&lt;ca=&lt;1e-006 et valeur initiale=1.88e-009 1e-012=&lt;cb=&lt;1e-006 et valeur initiale=1.77e-008 e = 48 g = 0.5 -15=&lt;ial=&lt;0 et valeur initiale=-6.764 1e-006=&lt;la=&lt;0.001 et valeur initiale=0.000037 0=&lt;nta=&lt;1 et valeur initiale=0.099 0=&lt;ntb=&lt;1 et valeur initiale=0.698 0=&lt;ntc=&lt;1 et valeur initiale=0.096 0=&lt;ntd=&lt;1 et valeur initiale=0.107 0=&lt;ntilma=&lt;0.5 et valeur initiale=0.2 0=&lt;ntilmi=&lt;1 et valeur initiale=0.5 0=&lt;ntvcama=&lt;1 et valeur initiale=0.5 0=&lt;ntvcbma=&lt;1 et valeur initiale=0.5 oida = 15 oidb = 15 oil = 15 oit = 15 ovca = 200 ovcb = 1500 ovda = 200 ovdb = 600 ovt = 200 pdim = 0 pphy = 1 1e-006=&lt;t=&lt;0.0001 et valeur initiale=2.8e-006 0=&lt;vaca=&lt;2000 et valeur initiale=1559 0=&lt;vacb=&lt;600 et valeur initiale=0                     </pre>	 <pre> 0=&lt;coil=&lt;1e-012 0=&lt;covca=&lt;1e-012 0=&lt;covcb=&lt;1e-012 -1=&lt;daofab=&lt;-1e-012 -1=&lt;daofbc=&lt;-1e-012 1e-012=&lt;daoncd=&lt;1 1e-012=&lt;dbonab=&lt;1 1e-012=&lt;dbonbc=&lt;1 1e-012=&lt;dbonda=&lt;1 0=&lt;fdlm=&lt;1 0=&lt;fphy=&lt;1 nt = 1 -1=&lt;tgilmax=&lt;1 -1=&lt;tgilmin=&lt;1 -1=&lt;tgvc=&lt;1 -1=&lt;tgvcb=&lt;1 0=&lt;tofbc=&lt;1 0=&lt;tofcd=&lt;1 0=&lt;tofda=&lt;1 -1=&lt;tonab=&lt;0 0=&lt;vcamax=&lt;200 0=&lt;watil=&lt;1 0=&lt;watvca=&lt;1 0=&lt;watvcb=&lt;1 0=&lt;wbtil=&lt;1 0=&lt;wbtvcb=&lt;1 0=&lt;wctil=&lt;1 0=&lt;wctvca=&lt;1 0=&lt;wctvcb=&lt;1 0=&lt;wdtil=&lt;1 0=&lt;wdtvca=&lt;1 0=&lt;wdtvcb=&lt;1                     </pre>
	<p>Critères de continuité</p> <p>Critères sur les tests de commutation</p> <p>Période normalisée</p> <p>Valeur crête de <math>V_{C1}</math></p> <p>Critères sur les pulsations</p>

Tableau V.4 : Le cahier des charges (cf. annexe D)

### 3.2.3.3 Calculs : lancement de l'optimisation et calcul des résultats (action n°10)

Une fois ce cahier des charges entré, nous pouvons lancer un dimensionnement. Le logiciel nous a fourni un compte rendu de son exécution qui a duré **quelques secondes** (cf. Figure V.33). Ici, le compte-rendu est 1, ce qui nous annonce que le dimensionnement est un succès, et l'algorithme a convergé en 64 itérations. Ainsi, nous avons obtenu un nouveau jeu de paramètres d'entrées et de sorties qui vérifient toutes les contraintes, et qui minimisent la fonction objectif. On peut alors visualiser les résultats obtenus (Tableau V.5).

RESULTATS BRUTS DE L'OPTIMISATION	
Paramètres d'entrée	Paramètres de sortie
<pre> Valeur des parametres d'entree... ca = 1.08711e-007 cb = 1.95832e-007 e = 48 g = 0.5 ial = -3.59654 la = 0.000243127 nta = 0.142992 ntb = 0.365891 ntc = 0.0706521 ntd = 0.420465 ntilma = 0.5 ntilmi = -6.71826e-026 ntvcama = 0.556066 ntvcbma = 0.400621 oida = 15 oidb = 15 oil = 15 oit = 15 ovca = 200 ovcb = 1500 ovda = 200 ovdb = 600 ovt = 200 pdim = 0 pphy = 1 t = 0.000029538 vaca = 91.0741 vacb = 0.0000143033                 </pre> <p>QUITTER</p> <p>-&gt; IMPRI</p>	<pre> PASCOSMA Fenêtre d'affichage f = 0.0000750413 alpha = 0.634109 coil = 1.45046e-006 covca = 0.0000728359 covcb = 7.54893e-007 daofab = -0.360413 daofbc = -0.201816 daofda = 0.0333333 daoncd = 4.3357e-006 dbofcd = 0.0333333 dbonab = 0.0333333 dbonbc = 0.0333333 dbonda = -1.91107e-006 fdim = 0.281785 fphy = 0.0000750413 ibl = -2.49252 icl = 3.3968 idl = 4.01609 ilmax = 3.82842 ilmin = 3.59654 nt = 1. tgilmax = 0.312978 tgilmin = 5.02728e-026 tgvca = 0.997924 tgvcb = -0.371042 tofbc = 0.226453 tofcd = 0.169682 tofda = 0.455298 tonab = -0.193877 vbca = -38.7754 vbcb = -72.0825 vcamax = 200. vcbmax = 100.77                 </pre> <p>QUITTER</p> <p>-&gt; IMPRI</p>

Tableau V.5 : Les résultats bruts du dimensionnement

```

PASCOSMA Compte-rendu
PASCOSMA
Compte-rendu:
  |
  | aide
  |
  | Nombre d'itérations
  | 64
  |
  | Parametres d'entree violes
  | ou arrivees en butee
  |
  | ntilma
  | ntilmi
  |
  | Grandeurs de sortie violees
  | ou arrivees en butee
  |
  | coil
  | covca
  | covcb
  | dbonda
    
```

Figure V.33 : Le compte-rendu de l'optimisation

### 3.2.3.4 Analyse et rebouclages éventuels (action n°11)

L'outil ainsi généré offre aussi la possibilité d'une analyse fine du dimensionnement obtenu. Nous allons illustrer ces fonctionnalités sur quelques exemples.

#### 3.2.3.4.1 Evolution des grandeurs

Il est possible de visualiser l'évolution de chaque paramètre pendant l'optimisation (cf. Figure V.34, et Figure V.37), ce qui permet de comprendre la dynamique du dimensionnement, et ainsi de pouvoir corriger efficacement les contraintes lorsqu'elles sont violées ou arrivent en butée pendant le processus d'optimisation.

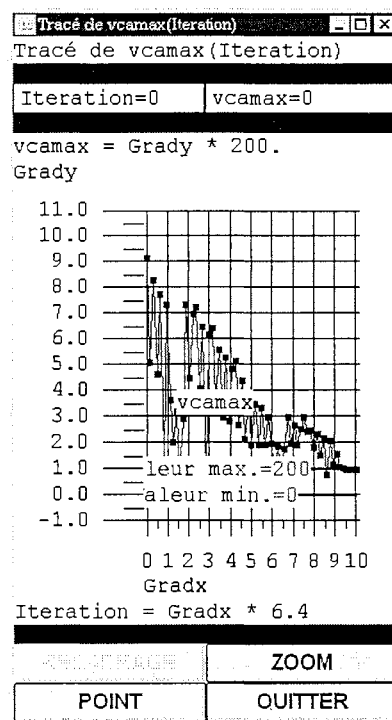


Figure V.34 : L'évolution de  $V_{C1 \max}$  (vcamax) en fonction du nombre d'itérations pendant le processus d'optimisation

#### 3.2.3.4.2 Etude de sensibilité

On peut visualiser la sensibilité d'un paramètre par rapport à un autre, ce qui permet à l'expert d'entreprendre une analyse de sa structure qui était jusqu'alors très difficile à réaliser : une analyse des tolérances de ses paramètres.

Nous avons visualiser dans la Figure V.35 la sensibilité du critère  $V_{C1 \max}$  (vcamax), représentant la tension maximale aux bornes de C1, par rapport à la valeur de la capacité C1 dans une plage de tolérance de  $\pm 10\%$  autour de la valeur calculée ( $1,09 \cdot 10^{-7}$  F, cf. Tableau V.5). Cette courbe permet

de constater une linéarité sur cet intervalle, avec une pente d'environ 4 % qui dénote d'une sensibilité relativement faible de ce critère par rapport à C1.

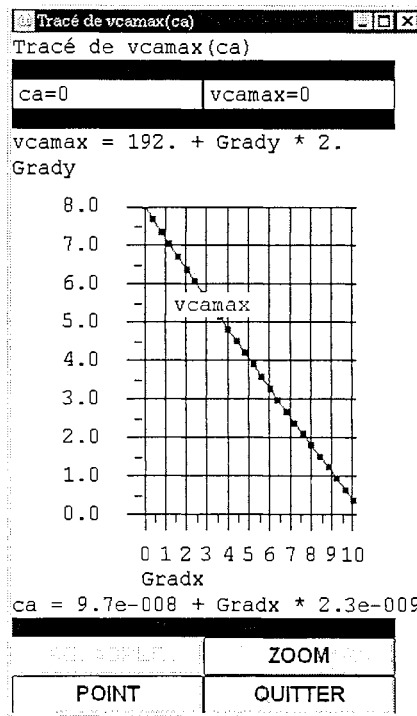


Figure V.35 : La sensibilité de  $V_{C1 \max}$  (vcamax) par rapport à la valeur de la capacité C1

En revanche, la Figure V.36 qui présente la courbe représentant l'évolution du critère contVC1 ( $covca$ ) de continuité de la variable d'état  $V_{C1}$  en fonction de la valeur de C1, sur le même intervalle de tolérance, montre une hypersensibilité : la pente est de l'ordre de  $10^6$  ! Cela signifie qu'une variation de  $1 \mu F$  sur C1 provoque une discontinuité de  $V_{C1 \max}$  (200 à 350 V) pour la commutation entre les phases extrêmes du mode. Attention, ceci n'est valable que pour un jeu de valeurs pour les paramètres, et peut être très différent d'une itération du dimensionnement à l'autre. En particulier, cela veut donc dire qu'en cours de dimensionnement, on a de fortes chances de passer par des points en dehors du mode de fonctionnement. Toutefois, lorsque le compte-rendu du dimensionnement renvoie un succès, le résultat final respecte le mode (car toutes les contraintes physiques que nous avons définies sont respectées).

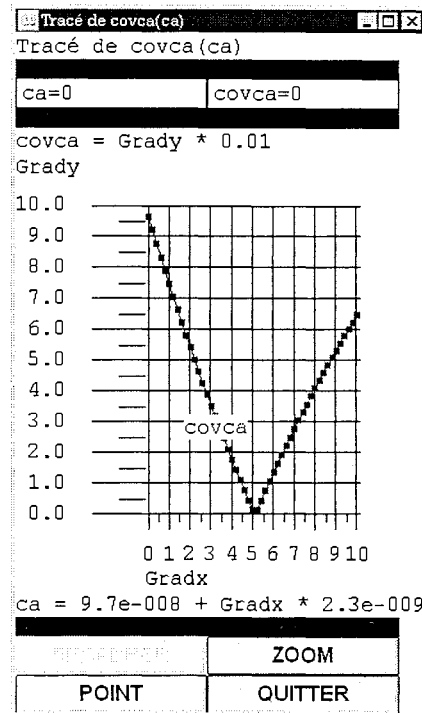


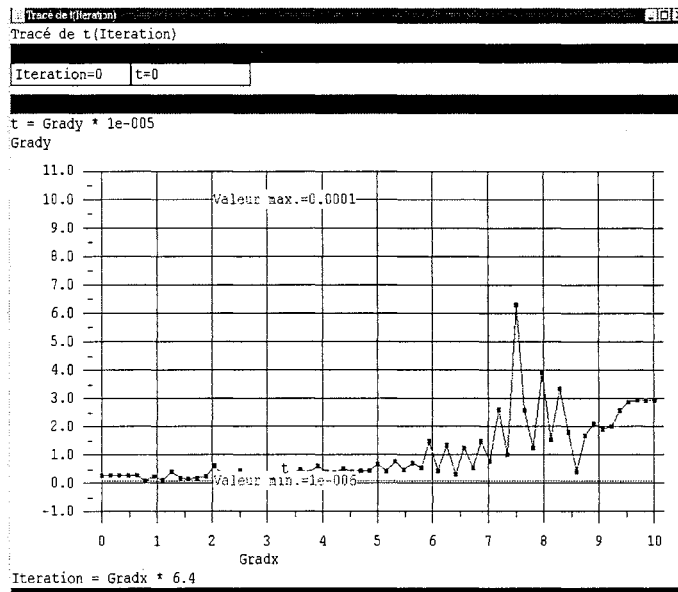
Figure V.36 : La sensibilité de contVC1 ( $covca$ ) par rapport à la valeur de la capacité C1 ( $ca$ )

A partir de cette simple courbe, on se rend donc mieux compte de la sensibilité du mode de fonctionnement par rapport aux paramètres pour une telle structure de convertisseur multirésonant : un petit écart sur ces paramètres, et le fonctionnement sort du mode choisi pour le dimensionnement, et alors toutes les expressions du modèle sont erronées, et le dimensionnement n'a plus de validité physique... (c'est alors qu'on peut parler de *dimensionnement virtuel*).

- De ce constat découlent :
  - ♦ la nécessité de contraindre sévèrement les critères de fonctionnement : ici, on peut voir (Tableau V.4) que la contrainte sur les critères  $covca$ ,  $covcb$ , et  $coil$  (qui s'attachent à assurer la continuité respectivement de la tension aux bornes de C1, de la tension aux bornes de C2, et du courant dans L1) est très resserrés (sur l'intervalle  $[0 ; 10^{-12}]$ ) ;
  - ♦ et l'intérêt d'implanter ces critères dans la fonction objectif de manière à *forcer la convergence*.
- D'autre part, une deuxième conséquence est le besoin d'un point de départ sûr, c'est-à-dire qui vérifie le plus scrupuleusement possible les contraintes de fonctionnement, car les risques de divergence de l'algorithme d'optimisation sur ces contraintes sont très grands. Pour obtenir ce point initial sûr, il est nécessaire de réaliser une simulation fine du fonctionnement du convertisseur étudié, et cela AVEC LE MEME MODELE que celui sur

lequel est fondé le modèle de dimensionnement. Ceci est pris en compte dans notre méthodologie, et cette précaution permet, en quelque sorte, de mettre l'algorithme d'optimisation *sur de bon rails* dès le début de la session de dimensionnement.

Nous avons aussi visualisé l'évolution de la période pendant le processus d'optimisation (Figure V.37).



**Figure V.37 : L'évolution de la période  $t$  en fonction du nombre d'itérations pendant le processus d'optimisation**

Cette figure nous montre que la période croît sensiblement pendant le dimensionnement, et la valeur finale obtenue ( $29,5 \cdot 10^{-5}$  s, soit une fréquence de 33,9 kHz) ne nous satisfait pas. Aussi nous décidons de réaliser un nouveau dimensionnement où la période sera contrainte plus sévèrement (sur l'intervalle  $[1,5 \cdot 10^{-6}; 10 \cdot 10^{-6}$  s] qui correspond en fréquence à [100 ; 667 kHz]), afin que l'on ne descende pas trop en fréquence, et en contrepartie, nous relaxons la contrainte sur le critère  $v_{cmax}$  dans l'intervalle [0 ; 350 V].

Le dimensionnement obtenu avec ce nouveau cahier des charges est un succès, les résultats obtenus (cf. Tableau V.6) montrent que la descente en fréquence est maîtrisée.

Point de fonctionnement de départ		Session 1		Session 2	
Paramètres	Point initial	Cahier des charges 1	Résultats 1	Cahier des charges 2	Résultats 2
C1 [nF]	1,88	[0,001 ; 1000]	109	[0,001 ; 1000]	14,8
C2 [nF]	17,7	[0,001 ; 1000]	196	[0,001 ; 1000]	35,1
L1 [ $\mu$ H]	37,0	[1 ; 1000]	243	[1 ; 1000]	19,3
E [V]	48,0	Fixe	48,0	Fixe	48,0
g [A]	0,50	Fixe	0,50	Fixe	0,5
F [kHz]	357	[10 ; 1000]	33,9	[100 ; 667]	303
Vcamax [V]	1800	[0 ; 200]	200,0	[0 ; 350]	350,0

Tableau V.6 : Le bilan des deux sessions de dimensionnement

Les simulations effectuées avec le module GENTIANE-ORO des points de fonctionnement obtenus après les sessions 1 et 2 viennent valider le dimensionnement ainsi réalisé (cf. Figure V.38 et Figure V.39).

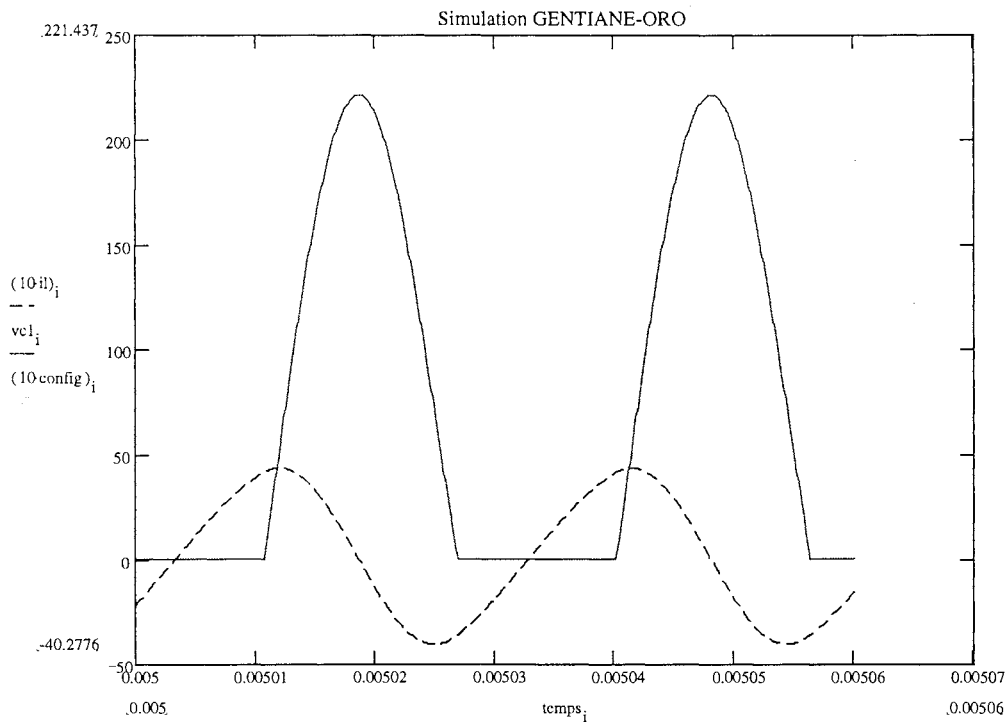


Figure V.38 : Les courbes représentatives de  $i_{L1}(t)$  et  $v_{C1}(t)$  pour le point de fonctionnement dimensionné après la session 1



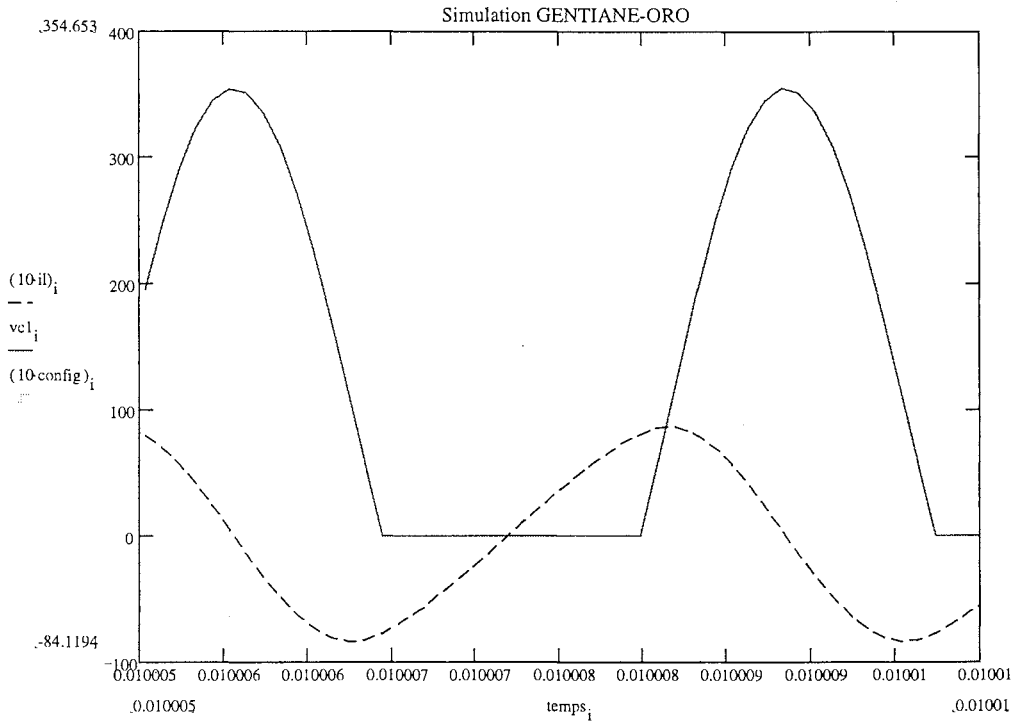


Figure V.39 : Les courbes représentatives de  $i_{L1}(t)$  et  $v_{C1}(t)$  pour le point de fonctionnement dimensionné après la session 2

#### 4 Seconde application : le cas d'école du hacheur série

Cette seconde application du cas d'école du hacheur série va nous permettre d'appréhender la flexibilité de l'outil de dimensionnement, et l'intérêt que cela apporte au niveau de la prise en compte de l'aspect discret des dimensions des composants.

##### 4.1 Structure et mode de fonctionnement

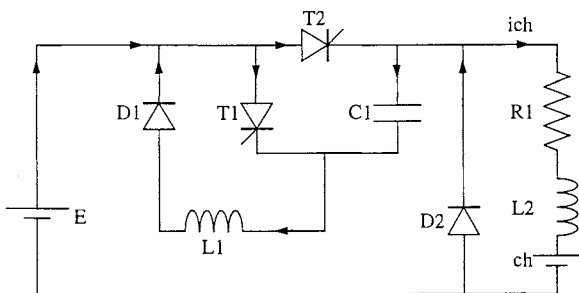


Figure V.40 : La structure du hacheur série

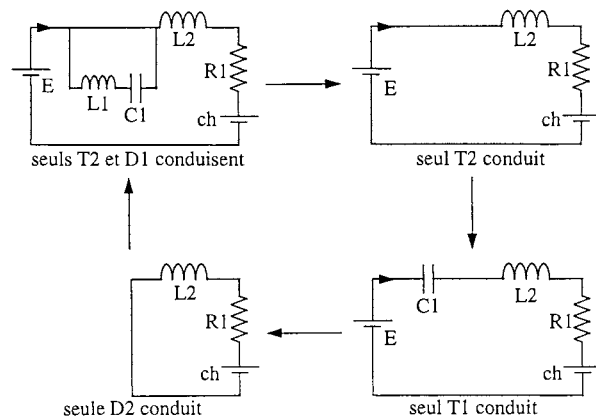


Figure V.41 : Le mode de fonctionnement du hacheur série

## 4.2 Problème spécifique de dimensionnement

Le problème simple de dimensionnement que nous désirons résoudre consiste à lisser le courant de charge à une valeur moyenne d'environ 30 A, et cela en jouant sur trois paramètres du circuit :

- la valeur de la capacité  $C1$ ,
- la valeur de l'inductance  $L1$ ,
- et la fréquence de hachage.

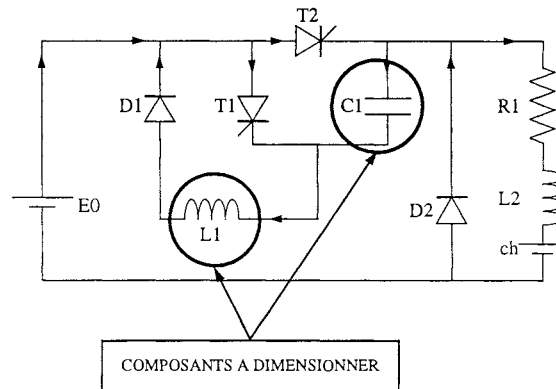


Figure V.42 : Notre problème simple de dimensionnement du hacheur série

### 4.2.1 Cahier des charges

A partir de ces spécifications, nous avons défini le cahier des charges (Figure V.44), qui fait intervenir deux critères que nous avons appelés :

- $i_{chmoy}$  pour la valeur du courant moyen dans la charge que l'on désire entre 29 et 30 A,
- et  $\Delta i_{ch}$  pour la valeur de l'ondulation du courant dans la charge que l'on veut inférieur à 5 A.

La Figure V.43 ci-dessous présente la formulation du critère qui est un critère propre à l'utilisateur.

```
/* Ondulation du courant de charge */
eqondu : deltich=2*(ichmax-Ichmoy);
```

Figure V.43 : La formulation du critère  $\Delta i_{ch}$

**CRITERES DE DIMENSIONNEMENT :**

*courant dans la charge "à peu près constant à 30 A"*

→ valeur moyenne du courant de charge :  $i_{chmoy}$

→ ondulation du courant de charge :  $\Delta i_{ch}$

**CAHIER DES CHARGES**

- Composants à dimensionner :  $\begin{cases} 1e-006 < C1 < 0.0001 \text{ F} \\ 1e-005 < L1 < 0.001 \text{ H} \end{cases}$
- Courant moyen de charge :  $29 \text{ A} < i_{chmoy} < 31 \text{ A}$
- Ondulation du courant dans la charge :  $0 \text{ A} < \Delta i_{ch} < 5 \text{ A}$
- Fréquence du hachage :  $1 \text{ kHz} < F < 10 \text{ kHz}$

Figure V.44 : Le cahier des charges

#### 4.2.2 Session de dimensionnement

Le modèle de dimensionnement a été construit, et l'outil de dimensionnement généré (Figure V.45), en suivant le processus décrit dans les paragraphes § 3.2.1, et § 3.2.2.

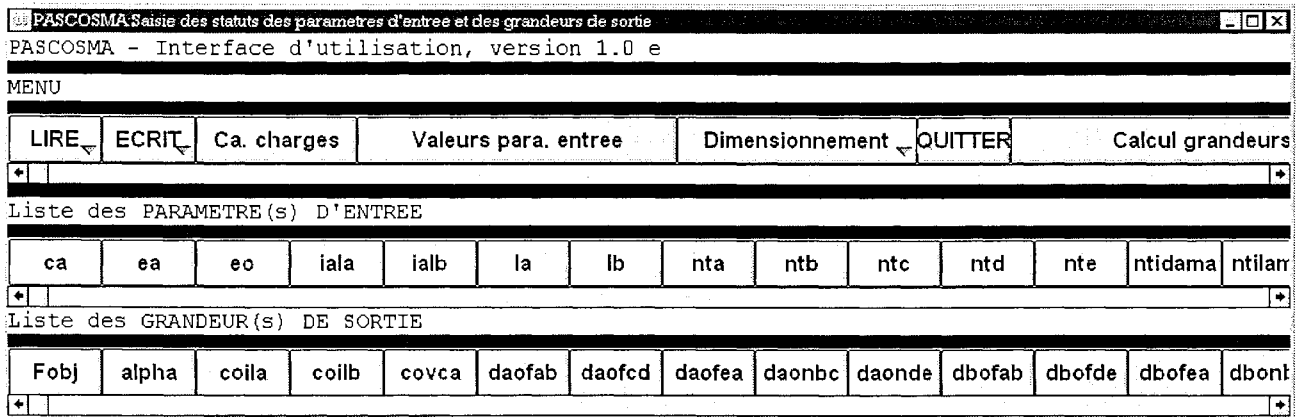
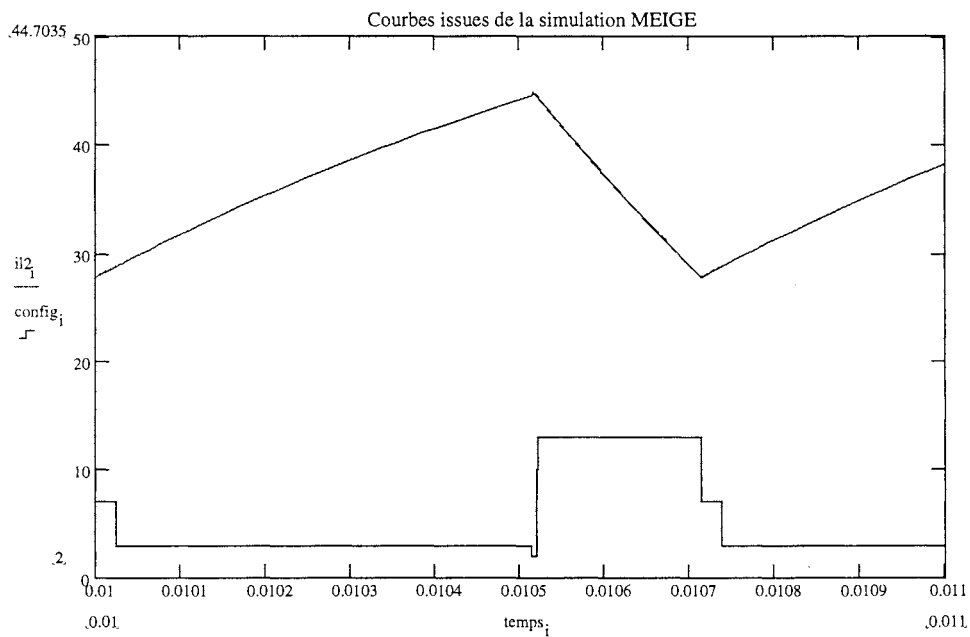


Figure V.45 : L'interface de l'outil généré pour le dimensionnement du hacheur série

D'autre part, nous disposons d'un point de fonctionnement que nous avons simulé avec le module GENTIANE-ORO (cf. Tableau V.7). Une rapide analyse de cette simulation nous montre que les performances du circuit avec ces dimensions ne sont pas compatibles avec nos requêtes de conception (cf. Figure V.46, et Tableau V.7) : le courant moyen dans la charge a une intensité de 37,1 A, et l'ondulation de ce courant s'approche des 15 A.

POINT DE FONCTIONNEMENT INITIAL (premier jeu de valeurs pour les paramètres)								
Paramètres d'entrée						Critères		
E [V]	ch [V]	R1 [ $\Omega$ ]	L1 [mH]	L2 [H]	C1 [ $\mu$ F]	F [kHz]	Ichmoy [A]	$\Delta$ ich [A]
120	48	1	0,06	0,001	0,36	1,4	37,1	14,8
CAHIER DES CHARGES (contraintes sur les paramètres d'entrée et les critères)								
fixe	fixe	fixe	[0,01 ; 1]	fixe	[1 ; 100]	[1 ; 10]	[29 ; 31]	[0 ; 5]

**Tableau V.7 : Etat des principaux paramètres au début du processus d'optimisation**



**Figure V.46 : La simulation GENTIANE-ORO du point de fonctionnement initial**

Pour prendre en compte l'aspect discret des composants C1 et L1, nous avons réalisé deux sessions de dimensionnement successives :

1. Une première session en relaxant les contraintes sur C1 et L1 : nous autorisons leurs évolutions sur une large plage continue de valeurs. Les résultats obtenus pour cette optimisation sont donnés dans le Tableau V.9. On remarque que les critères sont vérifiés après ce premier dimensionnement : le cahier des charges est respecté.

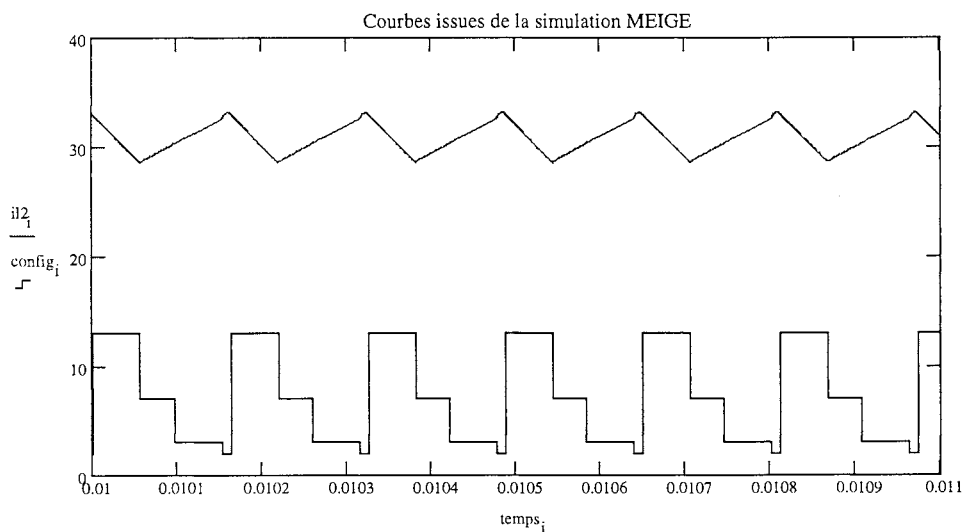
Paramètres d'entrée						Critères		
E [V]	ch [V]	R1 [ $\Omega$ ]	L1 [mH]	L2 [H]	C1 [ $\mu$ F]	F [kHz]	Ichmoy [A]	$\Delta$ ich [A]
120	48	1	<b>0,115</b>	0,001	<b>1,44</b>	<b>6,17</b>	<b>31</b>	<b>3,5</b>

**Tableau V.8 : Résultat pour les principaux paramètres après le premier dimensionnement**

2. Une seconde session en prenant le point dimensionné précédemment comme point initial, et en réalisant une restriction sévère sur la contrainte appliquée à C1 : cela consiste à affecter cette contrainte à une valeur discrète fixe, la plus proche de celle obtenue après le premier dimensionnement (ici, on considère que nous avons à notre disposition une capacité de 1,5  $\mu$ F). La simulation effectuée avec le module GENTIANE-ORO de ce nouveau point de fonctionnement (Figure V.47) corréle les résultats obtenus (cf. Tableau V.9).

Paramètres d'entrée						Critères		
E [V]	ch [V]	R1 [ $\Omega$ ]	L1 [mH]	L2 [H]	C1 [ $\mu$ F]	F [kHz]	Ichmoy [A]	$\Delta$ ich [A]
120	48	1	<b>0,112</b>	0,001	<b>1,5</b>	<b>6,14</b>	<b>31</b>	<b>3,5</b>

**Tableau V.9 : Résultat pour les principaux paramètres après le second dimensionnement**



**Figure V.47 : La simulation GENTIANE-ORO du point de fonctionnement obtenu après le second dimensionnement**

**Remarque :** Dans ce cas d'école, le second dimensionnement n'altère que très peu les premiers résultats obtenus (cf. Tableau V.8 et Tableau V.9), et le nombre d'étapes dans la session de dimensionnement est faible. Dans un problème classique de dimensionnement de convertisseur où le nombre de composants à dimensionner est plus grand, les résultats de dimensionnement peuvent être fortement perturbés à chaque étape. Ainsi, on s'aperçoit que l'ordre dans lequel on opère la restriction des contraintes n'est pas sans conséquence sur le dimensionnement final.

## **5 Conclusion**

Ces applications qui présentent deux cas simples de dimensionnement de convertisseurs statiques montrent le processus de construction, et quelques utilisations possibles de l'outil de dimensionnement dédié à un problème de dimensionnement.

Le fait que nous soyons capable de spécifier les actions à réaliser pour résoudre un problème de dimensionnement en utilisant notre méthodologie, montre la capacité d'intégration de notre approche, dans le sens où le rôle des outils et acteurs (expert/utilisateur/concepteur) est bien défini : six étapes (modélisation, simulation, analyse, formalisation du modèle de dimensionnement, génération de l'outil de dimensionnement, dimensionnement), et onze actions détaillées dans les paragraphes § 3.2.1.1 à § 3.2.3.4. Et ceci permet d'envisager une future automatisation de l'ensemble de la méthodologie avec des techniques d'intelligence artificielle, techniques qui emploieront des systèmes experts par exemple.



## *Avantages et inconvénients*





# AVANTAGES ET INCONVENIENTS DE LA DEMARCHE PROPOSEE

## 1 *Avantages*

La démarche que nous proposons présente plusieurs avantages :

- Son principe de base qui vise à traiter le problème de dimensionnement comme un problème d'optimisation contrainte formulé par des techniques symboliques permet d'obtenir des résultats rapides et précis.
- Sa capacité à être automatisable permet de décharger le concepteur d'un grand nombre de tâches répétitives, délicates et par conséquent sources d'erreurs, telles que les retraitements symboliques qui, lorsqu'ils sont réalisés manuellement, sont rapidement limités afin de pouvoir être exploités par l'utilisateur. En revanche, ceci empêche l'utilisateur de modifier le modèle généré, et tout doit être régénéré pour une petite modification.
- Son architecture modulaire, mais aussi la vitesse d'exécution des différents outils qui la composent (simulateur, outil de dimensionnement, outil d'analyse de sensibilités), rendent la démarche susceptible de pouvoir s'intégrer facilement dans une méthodologie de conception globale des convertisseurs statiques. Elle s'inscrit ainsi dans une voie nouvelle empruntée pour le développement des outils de C.A.O<sup>1</sup> qui vise à privilégier les interactions entre outils. Dans ce cadre, l'approche que nous proposons offre des outils d'analyse d'une nouvelle génération : simulation temporelle rapide, outil d'analyse de sensibilité, interface de dimensionnement, et un faisceau de liens qui les relie. Cette trame est la base d'un futur outil complet de dimensionnement des convertisseurs statique à partir de leur structure

---

<sup>1</sup> Conception Assistée par Ordinateur

## 2 Inconvénients

Les inconvénients de l'approche de dimensionnement que nous avons présentés dans ce mémoire concernent les hypothèses de mise en œuvre :

- Il est impossible avec la méthodologie actuelle de dimensionner le convertisseur dans des états transitoires. Cependant, ce n'est pas dans ces états que le concepteur cherchera généralement à réaliser son dimensionnement. D'autre part, avec cette méthode, le dimensionnement est obligatoirement attaché à un mode figé, ce qui n'est pas forcément demandé par le concepteur.
- Le dimensionnement obtenu n'est pas un optimum absolu, car l'algorithme du gradient utilisé dans le programme PASCOSMA peut être piégé par un extremum local. Néanmoins, lorsqu'un résultat est obtenu, on peut le considérer meilleur que le point initial. De plus, des méthodes d'amélioration de ces algorithmes sont entrepris afin de dépasser cet inconvénient.
- Les imperfections des composants ne sont pas prises en compte au niveau du modèle (qui est continu par construction). Le dimensionnement offert par l'outil final offre un point de fonctionnement unique, c'est à dire qu'il donne une valeur précise pour chaque paramètre. Par conséquent, d'une part, il n'est pas possible de contraindre les dimensions sur des critères de tolérance ; et d'autre part, le concepteur ne peut pas obtenir directement un dimensionnement valable pour une plage de variation pour certains paramètres. Cependant, la méthodologie proposée offre un outil de visualisation des sensibilités en marge de l'outil de dimensionnement, qui permet d'analyser a posteriori, dans ces optiques, les résultats obtenus.

## *Conclusion et perspectives*



## CONCLUSION ET PERSPECTIVES

Dans ce rapport, nous proposons une approche de dimensionnement des convertisseurs statiques. L'environnement intégré de dimensionnement qui en résulte permet de fournir presque automatiquement (il reste encore à achever les tests relatifs au module de génération automatique des modèles de dimensionnement) des outils dédiés de dimensionnement qui sont caractérisés par :

- une rapidité d'obtention d'une solution (quelques secondes),
- une procédure d'optimisation,
- un modèle générique de dimensionnement,
- une analyse temporelle via simulation,
- une analyse de sensibilité via l'optimisation,
- une intervention restreinte de l'utilisateur, qui se limite à la caractérisation des composants, et à une analyse d'expert des résultats fournis en simulation temporelle et en dimensionnement (sensibilité),
- l'intégration possible dans une démarche globale de conception, qui peut solliciter de nombreuses fois les outils de CAO conçus sans pénaliser les temps de traitement.

Cependant, les travaux entrepris dans cette thèse, qui inaugurent la voie dans de nouveaux champs d'investigation sur le sujet de la conception des convertisseurs statiques, ont dégagé de nouveaux problèmes qui ouvrent de nombreuses perspectives :

### **Au niveau des modèles :**

- Les problèmes de limitation dus à l'intégration formelle des systèmes d'états nous ont amenés à mettre en œuvre une solution mixant les techniques de calcul symbolique et des méthodes de calcul numérique. Ces travaux n'en sont encore qu'au point de départ, même si tout est en place dans le modèle de dimensionnement actuel pour les prendre en compte, les tests de validation n'ont pas encore été réalisés.
- Nous proposons aussi d'implanter dans la méthodologie, au niveau de l'étape de formulation du modèle de dimensionnement, une fonctionnalité permettant d'aider le concepteur pour simplifier le modèle de fonctionnement, et ainsi de fonder le dimensionnement sur des modèles simplifiés (tels que les modèles moyens par exemple).

- Afin de prendre en compte le dimensionnement dans plusieurs modes de fonctionnement lorsque cela est envisageable (cas du ZVS-MRC), nous envisageons la mise en œuvre de la construction d'un modèle multi-modal et multi-point-de-fonctionnement, dont le principe a été évoqué à la fin du chapitre II, § 5.4.
- Nous souhaitons piloter la construction du modèle de dimensionnement par des techniques d'intelligence artificielle afin de diminuer les redites (critères identiques) non détectées actuellement.
- Il faudra implanter des critères dans le domaine fréquentiel, qui permettraient de prendre en compte les problèmes de stabilité par exemple.
- On pourrait transposer la démarche de modélisation générique des convertisseurs statiques en vue du dimensionnement à certains composants dont les structures présentent de la généricité, (par exemple les transformateurs).

### **Au niveau des outils :**

- Il faudra développer et intégrer dans la méthodologie une simulation semi-symbolique fondée sur le modèle issu de l'intégration formelle des systèmes d'état utilisant des "rootofs". C'est un travail qui permettra d'estimer les performances de la formulation en "rootofs" de la résolution des équations différentielles en la dissociant du problème d'optimisation.
- Nous proposons aussi de mettre à la disposition de l'utilisateur une bibliothèque de modèles de composants tels que transformateurs, filtres, etc...à vocation dimensionnement, qui pourront être implantés aisément dans le modèle global de dimensionnement.
- Certaines interfaces entre outils restent encore à être informatisées. La généricité de l'information partagée par les étapes permet d'envisager ce travail avec sérénité, ce qui améliorera l'ensemble de l'utilisation de la méthodologie.
- Pour permettre une aide au tolérancement et au choix discret des composants, nous proposons de mettre en place une technique de Branch and Bound sur l'outil de dimensionnement.
- Et enfin, pour profiter au mieux de la structure du générateur de modèles de dimensionnement que nous avons développé, il sera possible d'envisager de mettre en œuvre une parallélisation des procédures de ce générateur, permettant ainsi un gain énorme en temps de génération.

## *Références bibliographiques*





## REFERENCES BIBLIOGRAPHIQUES

- AHRENS91** Ahrens, S. ; Kortaberria, I. ; Bigeon, J. ; Poupart, D. **Expert system for designing converters for large synchronous machines.** EPE'91. 4th European Conference on Power Electronics and Applications, 3-6 Sept. 1991, Florence, Italie, vol. 2, p. 266-71
- AMBROSIO95** D'Ambrosio, J.G. ; Birmingham, W.P. **Preference-directed design.** [AI EDAM] *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Juin 1995, vol 9, Iss 3, p. 219-230
- BAKKER93** Bakker, R.R. ; Dikker, F. ; Tempelman, F. ; Wognum, P.M. **Diagnosing and solving over-determined constraint satisfaction problems.** IJCAI'93. 30<sup>th</sup> International Joint Conference on Artificial Intelligence, 28 Aout-3 Sept. 1993, Chambéry, France, p. 276-281
- BAYARDO94** Bayardo, R.J., Jr.; Miranker, D.P. **An optimal backtrack algorithm for tree-structured constraint satisfaction problems.** *Artificial Intelligence*, Nov. 1994, vol 71, Iss 1, p. 159-181
- BEN AHMED95** Ben Ahmed, A. ; Lucidarme, J. ; Desesquelles, P.F. **Méthode semi-numérique de pré-dimensionnement des machines à aimants permanents et à bobinage global.** *Journal de Physique III France 5* (1995), Juin 1995, vol 5, Iss 6, p. 703-725
- BERGEON97a** Bergeon, S. ; Gerbaud, L. ; Bigeon, J. **Principe de la formulation symbolique des contraintes génériques de dimensionnement d'un convertisseur statique.** Numelec'97, 2<sup>nd</sup>e Conférence Européenne sur les méthodes numériques en Electromagnétisme, 19-21 Mars 1997, Lyon, France, p. 32-33
- BERGEON97b** Bergeon, S. ; Gerbaud, L. ; Wurtz, F. ; Roudet, J. **A methodology using constrained optimisation techniques for the design of static converters.** EPE'97 7<sup>th</sup> European Conference on Power Electronics and Applications conference, 8-10 Sept. 1997, Trondheim, Norvège, vol 2, p. 2.767-2.772
- BHAT90** Bhat, A.K.S. **Analysis, optimization and design of a series-parallel resonant converter.** APEC'90 5<sup>th</sup> Annual Applied Power Electronics Conference and Exposition, 11-16 Mars 1990, Los Angeles, CA, USA, p. 155-164
- BHAT95** Bhat, A.K.S. ; Biswas, A. ; Iyengar, B.S.R. **Analysis and design of (LC)(LC)-type series-parallel resonant converter.** *IEEE Transactions on Aerospace and Electronic Systems*, Juillet 1995, vol 31, Iss 3, p. 1186-1193

- 
- BOUCHE95** Bouche, J.M. ; Wurtz, F. ; Gerbaud, L. **Automatic generation of softwares for the design of static converters.** EPE'95 6<sup>th</sup> European Conference on Power Electronics and Applications, 19-21 Sept. 1995, Seville, Espagne, vol 3, p. 304-309
- CAMERON92** Cameron, D.E. ; Lang, J.H. ; Belanger, D. **The Computer-Aided Design Of Variable-Reluctance Generators.** APEC'92 7<sup>th</sup> Annual Applied Power Electronics Conference and Exposition, 23-27 Févr. 1992, Boston, MA, USA, p. 114-120
- CHHAYA92** Chhaya, S.M. ; Bose, B.K. **Expert system based automated design technique of a voltage-fed inverter for induction motor drive.** IAS'92 1992 IEEE Industry Applications Society Annual Meeting, 4-9 Oct. 1992, Houston, Texas, vol 1, p. 770-778
- CHHAYA93** Chhaya, S.M. ; Bose, B.K. **Expert system based automated simulation and design optimization of a voltage-fed inverter for induction motor drive.** IECON'93. International Conference on Industrial Electronics, Control, and Instrumentation, 15-19 Nov. 1993, Maui, HI, USA, vol 2, p. 1065-1070
- DAVENPORT93** Davenport, J. ; Siret, Y. ; Tournier, E. **Calcul formel. Systemes et algorithmes de manipulations algebriques.** Editions Masson 1993 275 pp., ISBN : 2 225 84200
- DEBE93** Debebe, K. ; Rajagopalan, V. ; Sankar, T.S. **Expert system based instruction aid for power electronics.** IPEC'93. International Power Engineering Conference 1993, 18-19 Mars 1993, Singapour, vol 2, p. 549-554
- DIAZ90** Diaz, A.R. ; Belding, B. **Modeling and solution strategies for hierarchical design optimization.** *Engineering optimization 1990*, vol 16, Iss 4, p. 247-273
- ECONOMOU94** Economou, G.-P.K. ; Anagnostopoulos, G.C. ; Theodosiou, D.T. ; Stouraitis, T. ; Goutis, C.E. **Non-linear optimization: artificial neural network solution techniques applied to the optimum linear feedback control of linear discrete-time dynamic systems.** EUROMICRO'94. 20<sup>th</sup> EUROMICRO Conference. System Architecture and Integration, 5-8 Sept. 1994, Liverpool, UK, p. 637-643
- ERTAN93** Ertan, H.B. ; Aftahi, M. **SPCAD, a tool for computer aided design of single phase induction motors.** IMACS-TC1'93 4<sup>th</sup> international conference. Association for Mathematics and Computers in Simulation. Computational Aspects of Electromechanical Energy Converters and Drives, 7-9 Juillet 1993, Montréal, Canada, p. 311-316

- FAIZ95 Faiz, J. ; Sharifian, B.B. **Comparison of Two Optimization Techniques for the Design of a Three-Phase Induction Motor Using Three Different Objective Functions.** *ETEP. European Transactions on Electrical Power Engineering*, Mai/Juin 1995, vol 5, Iss 3, p. 199-205
- FERRIEUX-FOREST94 Ferrieux, J.-P. ; Forest, F. **Alimentations à découpage convertisseurs à résonance principes-composants-modélisation.** Editions Masson, collection technologies, ISBN : 2-225-84555-7
- FEZZANI93 Fezzani, D. ; Piquet, H. ; Cheron, Y. ; Metz, M. **Design of static converters: an expert system approach.** EPE'93 The European Power Electronics Association, 13-16 Sept. 1993, Brighton, UK, p. 47-52
- FEZZANI96 Fezzani, D. ; Piquet, H. ; Foch, H. **Système expert pour la conception en électronique de puissance.** *Journal de Physique III France 6 (1996)*, Mars 1996, vol 6, Iss 3, p. 349-361
- FEZZANI97 Fezzani, D. ; Piquet, H. ; Foch, H. **Expert System for the CAD in Power Electronics - Application to UPS.** *IEEE Transactions on Power Electronics*, Mai 1997, vol 12, Iss 3, p. 578-586
- FRON94 Fron, A. **Programmation par contraintes.** Editions Addison-Wesley France, avril 1994, ISBN : 2-87908-062-2
- GERBAUD97 Gerbaud, L. ; Retière, N. ; Bolopion, A. **Automatic building of changeable topological model of static converters for the electrical drive simulation.** EPE'97 7<sup>th</sup> European Conference on Power Electronics and Applications conference , 8-10 Sept. 1997, Trondheim, Norvège, vol 2, p. 2.572-2.577
- GERBAUD96 Gerbaud, L. ; Bolopion, A. ; Bignon, J. **GENTIANE : a simulation package for electromechanical systems.** ELECTRIMACS'96 5<sup>th</sup> international conference on modelling and simulation of electric machines converters and systems, 17-19 Sept. 1996, St Nazaire, France, vol 3, p. 689-694
- HAMMA95 Hamma, F. ; Meynard, T.A. ; Tourkhani, F. ; Viarouge, P. **Characteristics and Design of Multilevel Choppers.** PESC'95 26<sup>th</sup> Annual IEEE Power Electronics Specialists Conference, p. 1208-1214
- HAMILL93 Hamill, D.C. ; Bateson, K.N. **Design-oriented analysis of a resonant zvs/zcs dc-dc converter.** EPE'93 The European Power Electronics Association, 13-16 Sept. 1993, Brighton, UK, p. 23-29
- HARWELL87 Harwell Subroutine Library, 10 Oct. 1987
- HAWORTH93 Haworth, M.S. ; Birmingham, W.P. **Towards optimal system-level design.** 30<sup>th</sup> Design Automation Conference, 14-18 Juin 1993, Dallas, TX, USA, p. 434-438

- 
- HAYES96** Hayes, J.G. ; Hall, J.T. ; Egan, M.G. ; Murphy, J.M.D. **Full-Bridge, Series-Resonant Converter Supplying the SAE J-1773 Electric Vehicle Inductive Charging Interface.** PESC'96 27<sup>th</sup> Annual IEEE Power Electronics Specialists Conference, 23-27 Juin 1996, Baveno, Italie, vol 2, p. 1913-1918
- HSIEH92** Hsieh, J. ; Liu, C.-C. **An integrated knowledge- and algorithm-based method for power converter design.** PESC'92 23<sup>rd</sup> Annual IEEE Power Electronics Specialists Conference, 29 Juin-3 Juillet 1992, Tolède, Espagne, vol 2, p. 1316-1323
- KNORR95** Knorr, U. ; Zacharias, L. **Innovative simulation methods in power electronics and drive technology on personal computers.** EPE'95 6<sup>th</sup> European Conference on Power Electronics and Applications, 19-21 Sept. 1995, Séville, Espagne, vol 3, p. 3.039-3.043
- KONE93** Kone, A.D. ; Nogarède, B. ; Lajoie Mazenc, M. **Le dimensionnement des actionneurs électriques : un problème de programmation non linéaire.** *Journal de Physique III France 3 (1993)*, Févr. 1993, vol 3, Iss 2, p. 285-301
- KUO-PENG91** Kuo-Peng, P ;Chéron, Y. ; Cussac, P. **Program for the computer aided design of the nonreversible series-resonant converter operating above natural frequency.** EPE'91. 4<sup>th</sup> European Conference on Power Electronics and Applications, 3-6 Sept. 1991, Florence, Italie, vol 4, p. 231-236
- KUO-PENG93** Kuo-Peng, P. **Conception assistée par ordinateur des convertisseurs à résonance série non réversibles.** Thèse de l'Institut National Polytechnique de Toulouse, France, 24 Sept. 1993
- KUROE95** Kuroe, Y. **Computer-Aided Method of Optimal Design for Transient Behaviors of Power Electronic Control Systems.** IPEC-Yokohama'95, International Power Electronics Conference, 3-7 Avril 1995, Yokohama, Japon, p. 228-233
- LANIER97** Lanier, D. **Rapport de stage de DEA de Mathématiques Appliquées : Le dimensionnement des convertisseurs statiques,** Juin 1997
- LAU90** Lau, B. **Computer-aided design of a DC-to-DC switching converter.** APEC'90. 5<sup>th</sup> Annual Applied Power Electronics Conference and Exposition, 11-16 Mars 1990, Los Angeles, CA, USA, p. 619-28
- LAVIEVILLE88** Laviéville, J-P. **Contribution à la conception d'alimentations continues de puissance à résonance. Etude et réalisation d'un convertisseur 50 Kw. 20 KHz.** Thèse de l'Institut National Polytechnique de Grenoble, France, 17 Juin 1988

- LECHEVALIER97** Lechevalier, C. ; Gerbaud, L. ; Bigeon, J. **A functional description for static converter structures design.** EPE'97 7<sup>th</sup> European Conference on Power Electronics and Applications conference , 8-10 Sept. 1997, Trondheim, Norvège, vol 2, p. 2.794-2.799
- LEI95** Lei, X. ; Povh, D. ; Kulicke, B. **A Non-linear Programming Approach Based on Quasi-Newton Techniques for Optimization of Power Systems.** ETEP. *European Transactions on Electrical Power Engineering*, Nov./Déc. 1995, vol 5, Iss 6, p. 367-373
- LEVITIN95** Levitin, G. ; Mazal-Tov, S. ; Elmakis, D. **Optimal Allocation of Selectionalizers in Radial Distribution Networks.** PES'95 Stockholm Power Tech. International Symposium on Electric Power Engineering, 18-22 Juin 1995, Stockholm, Suède, p. 761-765
- MACSYMA** **Macsyma Mathematics and System Reference Manual** 16<sup>th</sup> edition, Copyright © 1996 Macsyma Inc.
- MANDHANA96** Mandhana, O.P. **Computer aided analysis and design of load commutated resonant converters.** PESC'96 27<sup>th</sup> Annual IEEE Power Electronics Specialists Conference, 23-27 Juin 1996, Baveno, Italie, vol 1, p. 279-285
- MANTOOTH** Mantooth, H.A. ; Fiegenbaum, M. **Modeling with an analog hardware description language.** Ed. Kluwer Academic, ISBN : 0-7933-9516-6
- MAPLE5** Monagan, M.B. ; Geddes, K.O. ; Labahn, G. ; Vorkoetter, S. **Maple V Programming Guide** © Waterloo Maple, Inc., Ed. Springer
- MARANESI92** Maranesi, P. ; Naummi, G. **Design optimization of switching regulators.** PESC'92 23<sup>rd</sup> Annual Power Electronics Specialists Conference, 29 Juin-3 Juillet 1992, Tolède, Espagne, vol 2, p. 1310-1315
- MARANESI94** Maranesi, P. ; Naummi, G. ; Vanore, A. **Computer-aided design of HF converters: new needs, new tools.** IECON'94 20<sup>th</sup> International Conference on Industrial Electronics, Control and Instrumentation, 5-9 Sept. 1994, Bologne, Italie, vol 3, p. 1682-1687
- MATHCAD6** **Manuel de l'utilisateur Mathcad PLUS 6.0**, Mathsoft Europe, P.O. Box 12358, Edimbourg EH11 4GN, Royaume Uni, Copyright © 1991-1995 MathSoft, Inc.
- MATT94** Matt, D. ; Prieur, F. ; Glaize, C. **Simulation numérique en électronique de puissance. Méthode de la topologie variable.** *Journal de Physique III France 4 (1994)*, Janv. 1994, vol 4, Iss 1, p. 55-73

- 
- MAULIK92 Maulik, P.C. ; Carley, L.R. ; Rutenbar, R.A. **A Mixed-Integer Nonlinear Programming Approach to Analog Circuit Synthesis.** 29<sup>th</sup> ACM/IEEE Design Automation Conference, 8-12 juin 1992, Anaheim, CA, USA, p. 698-702
- MENDEZ95 Mendez, A. ; Bordonau, J. ; Peracaula, J. **An object-oriented approach to computer aided analysis of power electronics systems.** EPE'95 6<sup>th</sup> European Conference on Power Electronics and Applications, 19-21 Sept. 1995, Séville, Espagne, vol 1, p. 1.783-1.788
- MOSES91 Moses, J.A. ; Kirtley, J.L.Jr. ; Lang, J.H ; Tabors, R.D. ; de Cuadra, F. **A Computer-Based Design Assistant for Induction Motors.** IAS'91 IEEE Industry Applications Society Annual Meeting, 28 Sept.-4 Oct. 1991, Dearborn, MI, USA, vol 1, p. 1-7
- NAITOH95 Naitoh, K. **Four-Group Equation of Genetic Algorithm.** *JSME International Journal, Series C [Dynamics, Control, Robotics, Design and Manufacturing]*, Juin 1995, vol 38, Iss 2, p. 240-248
- NIQUIL97 Niquil, Y. ; Guillot, J. **Optimiser la gestion du stock de plutonium.** *EPURE*, Janvier 1997, vol 53, p. 21-29
- PIQUET94 Piquet, H. ; Cambronne, J.P. ; Saadate, S. ; Glaize, C. **Méthodologie et outils pour la conception en électronique de puissance.** EPF'94, 3-6 Sept. 1991, Cachan, France, p. 25-30
- RAJAGOPALAN92 Rajagopalan, V. **Modeling and Simulation of Power Electronic Converters for Power Supplies.** APEC'92 7<sup>th</sup> Annual Applied Power Electronics Conference and Exposition, 23-27 Févr. 1992, Boston, MA, USA, p. 27-32
- REATTI94 Reatti, A. **Steady-state analysis including parasitic components and switching losses of back and boost DC-DC PWM converters under any operating condition.** *International Journal of Electronics*, Novembre 1994, vol 77, Iss 5, p. 679-701
- SAAD93 Saad, M. ; Rafesthain, T. ; Rajagopalan, V. ; Al-Haddad, K. **Computer-aided design of direct static converters.** *Canadian Journal of Electrical and Computer Engineering*, Juillet 1993, vol 18, Iss 3, p. 97-103
- SAHRAOUI93 Sahraoui, O. ; Gerbaud, L. ; Normand, O. ; Roudet, J. **Contribution à la modélisation des convertisseurs statiques pour la simulation à topologie variable.** *Journal de Physique III France 3(1993)*, Juin 1993, vol 3, Iss 6, p. 1221-1241
- SALUDJIAN97 Saludjian, L. **Optimisations en électrotechnique par algorithmes génétiques.** Thèse de l'Institut National Polytechnique de Grenoble, France, 27 Juin 1997

- SCHELLMANN96 Schellmanns, A. **Conception d'une alimentation à découpage haute fréquence.** Rapport de DEA Génie Electrique de l'institut National Polytechnique de Grenoble, 17 Sept. 1996
- SHCHEGLOV93 Shcheglov, A.Yu. **The constraint method for optimizing complex systems.** *Telecommunications and Radio Engineering, Part 2 [Radio Engineering]*, Janvier 1993, vol 48, Iss 1, p. 34-38
- SHIH95 Shih, C.J. ; Wangsawidjaja, R.A.S. **Multiobjective fuzzy optimization with random variables in a mix of fuzzy and probabilistic environment.** 4<sup>th</sup> IEEE International Conference on Fuzzy Systems and 2<sup>nd</sup> International Fuzzy Engineering Symposium, 20-24 Mars 1995, Yokohama, Japon, vol 3, p. 1163-1170
- SINGH94 Singh, B. ; Singh, B.N. ; Saxena R.B. **Design optimization of current source inverter fed squirrel cage induction motor.** *Electric Machines and Power Systems*, Juillet/Aout 1994, vol 22, Iss 4, p. 467-480
- SPIAZZI96 Spiazzi, G. ; Rossetto, L. ; Mattavelli, P. **Design optimization of Soft-Switched Insulated DC/DC Converters With Active Voltage Clamp.** IAS'96 IEEE Industry Applications Conference. 31<sup>st</sup> IAS Society Annual Meeting, p. 1169-1176
- SRIDHAR95 Sridhar, K. ; Lang, J.H. ; Umans, S.D. **Integrated design of a boost converter.** APEC'95 10<sup>th</sup> Annual Applied Power Electronics Conference and Exposition, 5-9 Mars 1995, Dallas, TX, USA, vol 1, p. 160-168
- SUN93 Sun, J. ; Groststollen, H. **Averaged modeling and analysis of resonant converters.** PESC'93 24<sup>th</sup> Annual IEEE Power Electronics Specialists Conference, 20-24 Juin 1993, Seattle WA, USA, p. 707-713
- SUN96 Sun, J. ; Groststollen, H. **A symbolic Computation Package for Averaged Analysis of Power Electronic Systems.** APEC'96 11<sup>th</sup> annual applied Power Electronics Conference and Exposition, 3-7 Mars 1996, San Jose, CA, USA, p. 96-102
- SUN97 Sun, J. ; Groststollen, H. **Symbolic Analysis Methods for Averaged Modeling of Switching Power Converters.** *IEEE Transactions on Power Electronics*, Mai 1997, vol 12, Iss 3, p. 537-546
- TANG90 Tang, W. ; Tabisz, W.A. ; Lotfi, A. ; Lee, F.C. ; Vorperian, V. **DC analysis and design of forward multi-resonant converter.** PESC'90 21<sup>st</sup> Annual IEEE Power Electronics Specialists Conference, 11-14 Juin 1990, San Antonio, Texas, p. 862-869
- TANG92 Tang, W. ; Leu, C.S. ; Lee, F.C. **Charge Control for Zero-Voltage-Switching Multi-Resonant Converter.** p. 195-199



- 
- TOURKHANI91** Tourkhani, F. **Etude et conception d'un convertisseur à résonance série réversible en puissance.** Mémoire de maîtrise de l'école des gradués de l'université Laval, Québec, Septembre 1991
- TOURKHANI95** Tourkhani, F ; Viarouge, P. ; Meynard, T.A. **A CAD environment for power converters.** *Systems Analysis Modelling Simulation*, 1995, vol 18-19, p. 523-526
- TOURKHANI96** Tourkhani, F. **Environnement de CAO pour l'optimisation du dimensionnement global des convertisseurs statiques. Application à la CAO d'un onduleur multiniveaux.** Thèse de l'université Laval, Québec, Dept. de génie électrique, Janvier 1996
- TOURKHANI97** Tourkhani, F. ; Viarouge, P. ; Meynard, T.A. ; Gagnon, R. **Power Converter Steady-State Computation Using The Projected Lagrangian Method.** PESC'97 28<sup>th</sup> Annual IEEE Power Electronics Specialists Conference, 22-27 Juin 1997, St Louis, Missouri, USA, p. 1359-1363
- VIAROUGE93** Viarouge, P. ; Tourkhani, F. ; Kamwa, I. ; Le-Huy, H. **Nonlinear optimization techniques for the design of static converters.** IMACS-TC1'93 4<sup>th</sup> international conference. Association for Mathematics and Computers in Simulation. Computational Aspects of Electromechanical Energy Converters and Drives, 7-9 Juillet 1993, Montréal, CA, p. 543-547
- WEBSTER95** Webster, R. ; Ngo, K.D.T. **Computer-based symbolic circuit analysis and simulation.** IECON'95 21<sup>st</sup> International Conference on Industrial Electronics, Control and Instrumentation, 6-10 Nov. 1995, Orlando, FL, USA, vol. 1, p. 772-779
- WURTZ96a** Wurtz, F. **Une nouvelle approche pour la conception sous contraintes de machines électriques.** Thèse de l'Institut National Polytechnique de Grenoble, France, 28 Mai 1996
- WURTZ96b** Wurtz, F. ; Bigeon, J. ; Poirson, C. **A methodology and a tool for the computer aided design with constraints of electrical devices.** *IEEE Transactions on Magnetics*, Mai 1996, vol 32, Iss 3, p. 1429-1432  
10<sup>th</sup> Conference on the Computation of Electromagnetics Fields, 10-13 Juillet 1995, Berlin, Allemagne

# *ANNEXES*



# ANNEXE A

## LES RETRAITEMENTS SYMBOLIQUES SUR LES BLOCS-FONCTIONS

Nous présentons dans cette annexe les principes sur lesquels l'ensemble des retraitements symboliques sont réalisés par notre logiciel de formulation automatique du modèle de dimensionnement, en prenant en compte la présence éventuelle de *rootofs* dans les expressions. Tous ces calculs sont fondés sur des manipulations de listes. Dans le langage Macsyma, une liste est un ensemble d'éléments qui peuvent être hétérogènes. De nombreuses fonctions héritées du Lisp<sup>1</sup> permettent d'agir efficacement sur ces objets informatiques.

- Notations :

$L : [A, B]$ , L est la liste contenant les éléments A et B.

On peut imbriquer plusieurs listes :  $L : [A, [B, C]]$

- Application aux expressions dépendant des rootofs

$$\begin{aligned}\psi(\alpha, t) &= p(t) + \sum_{i=1}^n q(\alpha_i, t) \\ &= \Gamma + \Phi\end{aligned}\tag{66}$$

Ces expressions peuvent toujours être formulées sous la forme d'une somme de deux éléments : un premier élément contenant la partie de l'expression indépendante de la racine numérique :

$$\Gamma = p(t)$$

un second élément contenant la partie dépendant du ROOTOF :  $\Phi = \sum_{i=1}^n q(\alpha_i, t)$ .

Nous proposons alors de traiter ces expressions comme des listes regroupant chacun de ces éléments. Et pour formaliser ces deux éléments, nous proposons aussi d'utiliser une liste générique que nous avons appelée « bloc-fonction ».

Un "bloc-fonction" *BF* est une liste qui contient deux éléments :

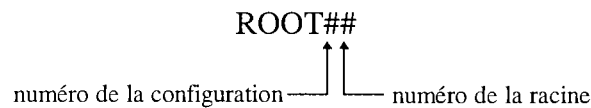
---

<sup>1</sup> Macsyma est programmé en Common Lisp

$$BF : [[\text{la liste des racines}], \text{l'expression de la fonction}] \quad (67)$$

Une liste de blocs-fonctions permet de formaliser les expressions symboliques contenant des racines non radicales, de façon générique, quelle que soit le logiciel de calcul formel utilisé (Maple ou Macsyma).

- *La liste des racines.* Elle est vide si la fonction est indépendante des rootofs, sinon elle contient les identificateurs des variables contenant les valeurs des racines déterminées numériquement. Ces identificateurs sont générés automatiquement. Ils s'expriment :



- *L'expression de la fonction f* en fonction de la racine identifiée par le symbole RØØTØF.

Exemple : Soit  $\psi = \Gamma + \Phi$ ,

$$\text{avec } \Gamma = p(t) = A \cdot \sin(\omega t), \text{ et } \Phi = \sum_{i=1}^5 q(\alpha_i, t) = \sum_{i=1}^5 B \cdot e^{C \alpha_i t}.$$

On suppose qu'il y a cinq racines  $\alpha_i$  et qu'il s'agit de la première configuration du mode.

Les blocs-fonctions correspondants sont :

- pour  $\Gamma$ , indépendant des racines :

```
TAU : [
      [,
      LAMBDA=A*SIN(OMEGA*T)
      ]
```

- pour  $\Phi$  :

```
PHI : [
      [ROOT11,ROOT12,ROOT13,ROOT14,ROOT15],
      GAMMA=B*EXP(C*RØØTØF*T)
      ]
```

et donc la liste de blocs-fonctions qui formalise  $\psi$  est la suivante :

```
PSI : [TAU,PHI]
PSI : [
      [, LAMBDA= A*SIN(OMEGA*T) ],
      [[ROOT11,ROOT12,ROOT13,ROOT14,ROOT15], GAMMA=B*EXP(C*RØØTØF*T) ]
      ]
```

Toutes les fonctionnalités de traitement des listes fournies par Macsyma permettent de manipuler ces symboles de façon élégante et efficace, et de réaliser une traduction optimale des expressions correspondantes.

Le principe de traduction de chaque bloc-fonction en code exécutable (FORTRAN) consiste :

A programmer la fonction à un argument du bloc-fonction.

Pour notre exemple :

```
FUNCTION LAMBDA(RØØTØF)
LAMBDA = A*SIN(OMEGA*T)
END
```

```
FUNCTION GAMMA(RØØTØF)
GAMMA = B*EXP(C*RØØTØF*T)
END
```

A programmer l'instruction d'affectation de la variable correspondante qui fait appel, autant de fois qu'il y a de racines dans la liste des racines, à la fonction précédemment définie, en passant en argument chaque élément de la liste des racines.

Retour à l'exemple :

```
TAU = LAMBDA(0)
PHI = GAMMA(ROOT11) + GAMMA(ROOT12) + GAMMA(ROOT13) + GAMMA(ROOT14) + GAMMA(ROOT15)
```

Ce mécanisme de traduction générique a été automatisé dans la démarche. La programmation unique de la fonction permet de diminuer considérablement la taille du code généré, en évitant les redites d'un codage purement procédural.

Par conséquent, la traduction d'une liste de bloc-fonction consiste simplement à traduire chaque bloc-fonction indépendamment, puis de faire la somme de chacune des variables ainsi affectées.

Exemple :       $\text{PSI} = \text{TAU} + \text{PHI}$

- Propriétés :

Ce sont les listes de blocs-fonctions ainsi définies qui sont les symboles manipulés pendant les phases de retraitement formels nécessaires pour formuler les expressions (59), (60), (61), (62), (63), (64) et (65) du chapitre IV, pp. 92-93, à partir des expressions de base. Ces retraitements symboliques se résument à des sommes algébriques, alors les calculs à réaliser sur les listes de blocs-fonctions sont des additions, et des multiplications.

- Additions de listes blocs-fonctions :

Vu le principe de traduction des blocs-fonctions en langage exécutable, additionner deux listes de blocs-fonctions revient simplement à regrouper ces deux listes dans une liste commune.

Exemple :      Soient deux listes de blocs-fonctions,  $X : [\text{BF1}, \text{BF2}, \text{BF3}]$ , et  $Y : [\text{BF4}, \text{BF5}]$ ,  
leur somme est aisément formulable :  
 $X + Y : [\text{BF1}, \text{BF2}, \text{BF3}, \text{BF4}, \text{BF5}]$

Ceci est programmable aisément avec Macsyma (à l'aide une fonction de base issue du Lisp).

- Multiplications de blocs-fonctions

Il s'agit de multiplier les listes de blocs-fonctions par une expression symbolique classique.

Exemple :  $\frac{\partial C}{\partial E} \cdot X(t)$ , cf. (61) p. 93 ;  $C \cdot \int X dt$ , cf. (62) p. 93.

Cela revient à multiplier chaque bloc-fonction de la liste par le facteur commun, ce qui consiste à multiplier la fonction de chaque bloc-fonction par l'expression en facteur.

Une procédure automatique réalisant cette multiplication a été programmée.

Le formalisme employé pour les expressions de base permet aussi d'opérer les imbrications de sommes nécessaires pour le calcul des  $\frac{\partial}{\partial E} \left( \int Y^2 dt \right)$  (cf. (65) p. 93), il s'agit alors grossièrement d'imbrications judicieuses de blocs-fonctions.

# ANNEXE B

## FICHER MATHCAD D'ANALYSE DE LA SIMULATION GENTIANE-ORO

Voici la feuille de calcul Mathcad que nous avons programmée pour réaliser l'analyse du fonctionnement du ZVS-MRC présenté au chapitre V. Cette feuille peut être réutilisée pour une autre application avec peu de modifications.

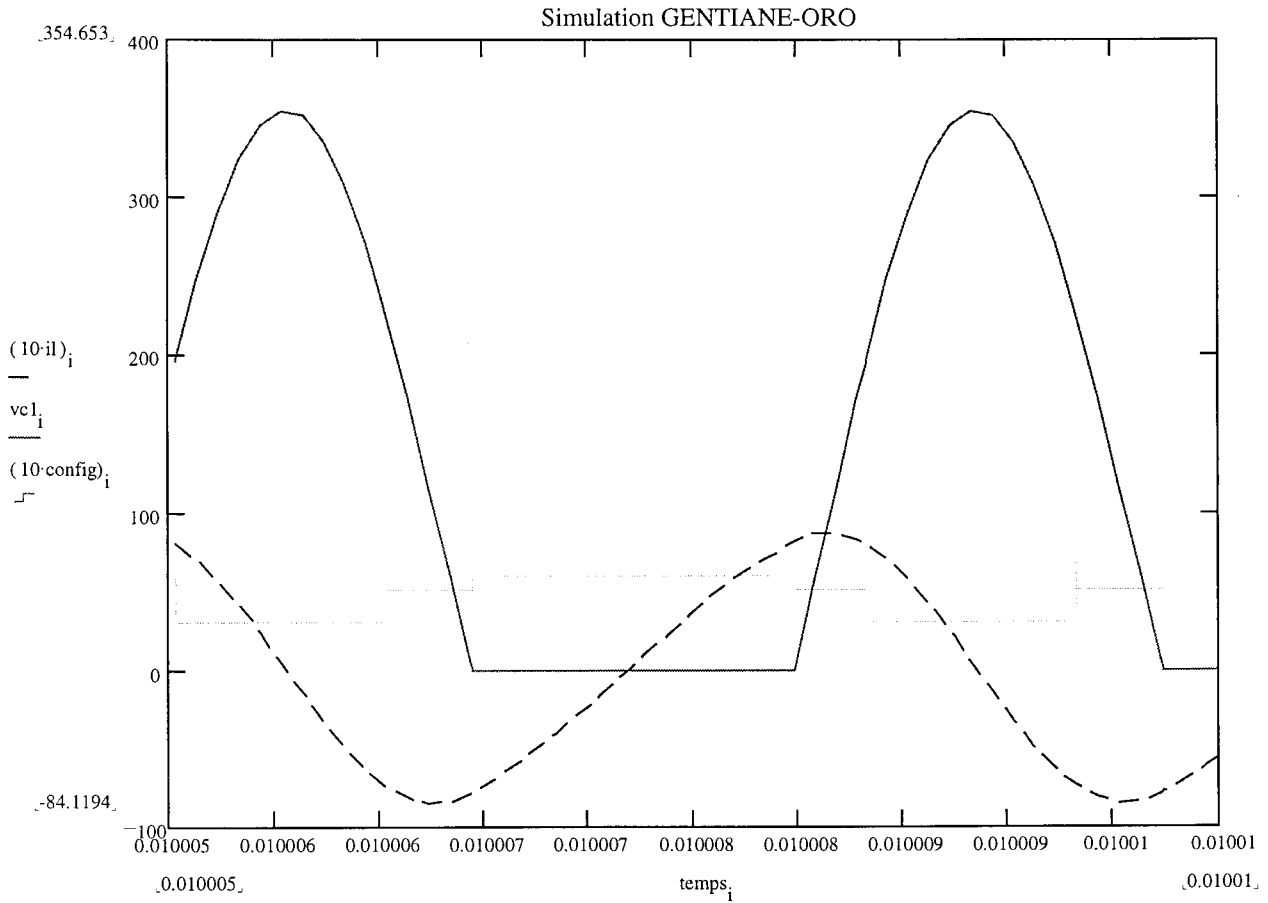
### SIMULATION MEIGE DU MULTI-RESONANT EN THYRISTOR DUAL FORWARD

```

M := LIREPRN(tang350)
Données brutes dans le fichier resultat.csv issu de GENTIANE-ORO
n := lignes(M) - 1
i := 0..n
Colonne des temps :
Colonne des config :
    courant I      :
    tension c1     :
    tension c2     :
    chrono         :
ctemps := 0
Nombre de configurations
cconfig := 2
nconfig := 6
cil1 := 5
cil2 := 1
cvc1 := 6
Première configuration
cvc2 := 3
config1 := 5
chro := 1
Création des vecteurs temps, config, courant, et compteur
temps := M<ctemps>
config := M<cconfig>
il := M<cil1>
vc1 := M<cvc1>
vc2 := M<cvc2>

```





### Sélection des données particulières en fin de configuration

```

commutation(M, cconf) := L-0
                        j-0
                        for i ∈ 0..lignes(M) - 2
                            if Mi, cconfig = Mi+1, cconfig
                                Lj ← i + 1
                                j ← j + 1
                        L
    
```

lcommut := commutation(M, cconfig)

### Construction des vecteurs entrees

```

donnees (M, lcom, col) := V-0
                        j-0
                        for i ∈ 0..lignes(lcom) - 1
                            Vj ← Mlcom, col
                            j ← j + 1
                        V
    
```

### Construction du vecteur durée des configurations

```

calculateduree(M, lcom, chro) :=
  D ← 0
  j ← 0
  for i ∈ 0..lignes(lcom) - 2
    Dj ← Mlcomi+1, chro - Mlcomi, chro
    j ← j + 1
  D

```

### Création du vecteur des configurations

config := donnees(M, lcommut, cconfig)

#### On se limite à une période

```

unepereode(V, Vconf, config1, nconfig) :=
  R ← 0
  j ← 0
  while Vconfj ≠ config1
    j ← j + 1
  for i ∈ 0..nconfig - 1
    Ri ← Vj+i
  R

```

lcommut1 := unepereode(lcommut, config, config1, nconfig + 1)

### Création des vecteurs données

duree :=  $\frac{\text{unepereode}(\text{calculateduree}(M, \text{lcommut}, \text{chro}), \text{config}, \text{config1}, \text{nconfig})}{10^9}$

il := unepereode(donnees(M, lcommut, cil1), config, config1, nconfig)

vc1 := unepereode(donnees(M, lcommut, cvc1), config, config1, nconfig)

vc2 := unepereode(donnees(M, lcommut, cvc2), config, config1, nconfig)

config := unepereode(config, config, config1, nconfig)

periode :=  $\sum$  duree

frequence :=  $\frac{1}{\text{periode}}$

alpha :=  $\frac{\text{duree}_2}{\text{periode}}$

periode =  $3.295 \cdot 10^{-6}$

frequence =  $3.035 \cdot 10^5$

alpha = 0.105

nduree :=  $\frac{\text{duree}}{\text{periode}}$

resultat<sup><0></sup> := config

resultat<sup><1></sup> := duree

resultat<sup><2></sup> := nduree

resultat<sup><3></sup> := il

resultat<sup><4></sup> := vc1

resultat<sup><5></sup> := vc2

$$\text{resultat} = \begin{bmatrix} 5 & 4.103 \cdot 10^{-7} & 0.125 & -7.361 & 224.081 & 0 \\ 6 & 1.538 \cdot 10^{-6} & 0.467 & -7.735 & 0 & -95.024 \\ 5 & 3.455 \cdot 10^{-7} & 0.105 & 8.112 & 0 & -83.037 \\ 7 & 1 \cdot 10^{-11} & 3.035 \cdot 10^{-6} & 7.944 & 196.631 & 0 \\ 3 & 1.001 \cdot 10^{-6} & 0.304 & 7.944 & 196.637 & 0 \\ 7 & 1 \cdot 10^{-11} & 3.035 \cdot 10^{-6} & -7.366 & 223.999 & 0 \end{bmatrix}$$

# ANNEXE C

## FICHER DE DESCRIPTION DU MODELE DE DIMENSIONNEMENT DU ZVS-MRC (VERSION NON AUTOMATISEE)

Voici le fichier de description du modèle de dimensionnement du ZVS-MRC présenté au chapitre V. Nous l'avons construit entièrement manuellement, et nous avons reporté en marge :

| : Ce qui peut être automatisé, mais qui nécessite l'intervention ponctuelle de l'utilisateur (des choix)

| : Ce qui ne peut pas être automatisé, et qui est formulé par le concepteur

Le reste est complètement automatisé dans l'outil que nous sommes en train de tester actuellement.

```

/*
Stephane Bergeon
mardi 8 octobre 1996

equations de dimensionnement pour le multiresonant
utilisant le modele multid1 sans resistance en parallele avec la source de
courant

MODE OBSERVE PAR LA SIMULATION [FROISSE]:  5 - 6 - 5 - 3
      DANS L ARTICLE [TANG]   : T2 - T1 - T2 - T3
*/

/* premiere phase D2 conduit seule          [FROISSE 5] - [TANG T4]
*/
eqail  : ratsubst(ial,il5,ratsubst(vaca,vc15,ratsubst(vacb,vc25,rhs(res[11]))));
eqavca : ratsubst(ial,il5,ratsubst(vaca,vc15,ratsubst(vacb,vc25,rhs(res[12]))));
eqavcb : ratsubst(ial,il5,ratsubst(vaca,vc15,ratsubst(vacb,vc25,rhs(res[13]))));

eqavt  : eqavca;
eqavda : eqavcb;
eqaidb : g;

/* deuxieme phase T1 et D2 conduisent      [FROISSE 6] - [TANG T1]
*/
eqbil  : ratsubst(vbcb,vc26,ratsubst(ibl,il6,rhs(res[14])));
eqbvca : vbca;
eqbvcb : ratsubst(vbcb,vc26,ratsubst(ibl,il6,rhs(res[15])));

```

```

eqbit : eqbil;
eqbvda : eqbvcb;
eqbidb : g;

/* troisieme phase D2 conduit seule [FROISSE 5] - [TANG T4]
*/
eqcil : ratsubst(icl,il5,ratsubst(vcca,vc15,ratsubst(vccb,vc25,rhs(res[11]))));
eqcvca : ratsubst(icl,il5,ratsubst(vcca,vc15,ratsubst(vccb,vc25,rhs(res[12]))));
eqcvcb : ratsubst(icl,il5,ratsubst(vcca,vc15,ratsubst(vccb,vc25,rhs(res[13]))));

eqcvda : eqcvcb;
eqcidb : g;
eqcvt : eqcvca;

/* quatrieme phase D1 conduit seule [FROISSE 3] - [TANG T3]
*/
eqdil : ratsubst(idl,il3,ratsubst(vdca,vc13,ratsubst(vdcb,vc23,rhs(res[6]))));
eqdvca : ratsubst(idl,il3,ratsubst(vdca,vc13,ratsubst(vdcb,vc23,rhs(res[7]))));
eqdvcb : ratsubst(idl,il3,ratsubst(vdca,vc13,ratsubst(vdcb,vc23,rhs(res[8]))));

eqdvdb : -eqdvcb;
eqdida : g;
eqdvt : eqdvca;

/* CONTRAINTES DE CONTINUITÉ NE FAISANT INTERVENIR QUE n+1 ENTREES */
/* DE COURANT */
conail : coil=abs(ial-ratsubst(td,tps,eqdil))/oil; /* 0 */
conbil : ibl=ratsubst(ta,tps,eqail);
concil : icl=ratsubst(tb,tps,eqbil);
condil : idl=ratsubst(tc,tps,eqcil);

/* DE TENSION */
conavca: covca=abs(vaca-ratsubst(td,tps,eqdvca))/ovca; /* 0 */
conbvca: vbca=ratsubst(ta,tps,eqavca);
concvca: vcca=ratsubst(tb,tps,eqbvca);
condvca: vdca=ratsubst(tc,tps,eqcvca);

conavcb: covcb=abs(vacb-ratsubst(td,tps,eqdvcb))/ovcb; /* 0 */
conbvcb: vbcb=ratsubst(ta,tps,eqavcb);
concvcb: vccb=ratsubst(tb,tps,eqbvcb);
condvcb: vdcb=ratsubst(tc,tps,eqcvcb);

/* CONTRAINTES DE DUREE */
eqnta : ta=t*nta; /* 0<nta<1*/
eqntb : tb=t*ntb; /* 0<ntb<1*/
eqntc : tc=t*ntc; /* 0<ntc<1*/
eqntd : td=t*ntd; /* 0<ntd<1*/

eqnt : nt=nta+ntb+ntc+ntd; /* =1 */
eqalpha: alpha=(ta+tc+td)/t; /* 0.3 */

/* CONTRAINTES DE COHERENCE DES ENCHAINEMENTS DE CONFIGURATION */

/* commutation 1-2 :
allumage T1 commande a l ouverture*/
testabton : tonab=ratsubst(ta,tps,eqavt)/ovt; /* <0 */
/*non allumage D1*/
testabdaof : daofab=ratsubst(ta,tps,eqavda)/ovda; /* <0 */
/*non extinction D2*/
testabdbon : dbonab=ratsubst(ta,tps,eqaidb)/oidb; /* =>0 */

/* commutation 2-3 :

```

```

    extinction T1 explicitee par la contrainte de duree eqalpha car T1 est
    commande a l ouverture*/
testbctof   : tofbc=ratsubst(tb,tps,eqbit)/oit;          /* >0 */
/*non allumage D1*/
testbcdaof  : daofbc=ratsubst(tb,tps,eqbvda)/ovda;      /* <0 */
/*non extinction D2*/
testbcdbon  : dbonbc=ratsubst(tb,tps,eqbidb)/oidb;      /* =>0 */

/* commutation 3-4 :
   extinction D2*/
testcddbof  : dbofcd=ratsubst(tc,tps,eqcidb)/oidb;      /* <=0 */
/*allumage D1*/
testcddaon  : daoncd=ratsubst(tc,tps,eqcvda)/ovda;      /* =>0 */
/*non allumage T*/
testcdtof   : tofcd=ratsubst(tc,tps,eqcvt)/ovt;        /* =>0 */

/* commutation 4-1 :
   allumage D2*/
testdadbon  : dbonda=ratsubst(td,tps,eqdvdb)/ovdb;      /* =>0 */
/*extinction D1*/
testdadaof  : daofda=ratsubst(td,tps,eqdida)/oida;      /* <=0 */
/*non allumage T*/
testdatof   : tofda=ratsubst(td,tps,eqdvt)/ovt;        /* =>0 */

/* PULSATIONS */
eqawil:wail=first(une_fois(cherche_pulsation(eqail,tps)));
eqbwil:wbil=first(une_fois(cherche_pulsation(eqbil,tps)));
eqcwil:wail=first(une_fois(cherche_pulsation(eqcil,tps)));
eqdwil:wdil=first(une_fois(cherche_pulsation(eqdil,tps)));

eqawvca:wavca=first(une_fois(cherche_pulsation(eqavca,tps)));
/*eqbwvca:wbvca=first(une_fois(cherche_pulsation(eqbvca,tps)));*/
eqcwvca:wcvca=first(une_fois(cherche_pulsation(egcvca,tps)));
eqdwvca:wdvca=first(une_fois(cherche_pulsation(eqdvca,tps)));

eqawvcb:wavcb=first(une_fois(cherche_pulsation(eqavcb,tps)));
eqbwvcb:wbvcb=first(une_fois(cherche_pulsation(eqbvcb,tps)));
eqcwvcb:wcvcb=first(une_fois(cherche_pulsation(egcvcb,tps)));
eqdwvcb:wdvcb=first(une_fois(cherche_pulsation(eqdvcb,tps)));

eqawtil:watil=wail*ta/%pi;                               /* 0< <1 */
eqbwtil:wbttil=wbil*tb/%pi;                               /* 0< <1 */
eqcwtil:wcttil=wcil*tc/%pi;                               /* 0< <1 */
eqdwttil:wdttil=wdil*td/%pi;                             /* 0< <1 */

eqawtvca:watvca=wavca*ta/%pi;                             /* 0< <1 */
/*eqbwtvca:wbtvca=wbvca*tb/%pi;*/
eqcwtvca:wctvca=wcvca*tc/%pi;                             /* 0< <1 */
eqdwtvca:wdtvca=wdvca*td/%pi;                             /* 0< <1 */

eqawtvcb:watvcb=wavcb*ta/%pi;                             /* 0< <1 */
eqbwtvcb:wbtvcb=wbvcb*tb/%pi;                             /* 0< <1 */
eqcwtvcb:wctvcb=wcvcb*tc/%pi;                             /* 0< <1 */
eqdwtvcb:wdtvcb=wdvcb*td/%pi;                             /* 0< <1 */

/* COURANT MAXIMUM POSITIF dans l inductance : pendant la config 3 [froisse 5]
*/
|
|
| eqtilmax : tgilmax=diff(ratsubst(tilmax,tps,eqcil),tilmax)*tilmax; /* =0 */
|
|
| eqntilmax: tilmax=ntilma*tc;
| eqilmax : ilmax=ratsubst(tilmax,tps,eqcil);              /* 13 */

```

```

|
| /* COURANT MAXIMUM NEGATIF dans l inductance : pendant la config 1 [froisse 5]
| */
|
| eqtilmin : tgilmin=diff(ratsubst(tilmin, tps, eqail), tilmin)*tilmin; /* =0 */
|
| eqntilmin : tilmin=ntilmi*ta;
| eqilmin : ilmin=-ratsubst(tilmin, tps, eqail); /* 13 */
|
|
| /* TENSION MAXI */
| /* positive aux bornes de la capacite C1 : pendant la config 4 [froisse 3] */
|
| eqtvcamax: tgvca=diff(ratsubst(tvcamax, tps, eqdvca), tvcamax)*tvcamax; /* =0 */
|
| eqntvcamax : tvcamax=ntvcama*td;
| eqvcamax : vcamax=ratsubst(tvcamax, tps, eqdvca); /* 2000 */
|
| /* negative aux bornes de la capacite C2 : pendant la config 2 [froisse 6] */
|
| eqtvcbmax: tgvcb=diff(ratsubst(tvcbmax, tps, eqbvcb), tvcbmax)*tvcbmax; /* =0 */
|
| eqntvcbmax : tvcbmax=ntvcbma*tb;
| eqvcbmax : vcbmax=-ratsubst(tvcbmax, tps, eqbvcb); /* 600 */
|
|
| /* TENSION MOYENNE */
|
| /* COURANT MOYEN */
|
| /* FONCTION OBJECTIF */
|
| /* Fonction objectif physique */
| fobj1 : Fphy=coil+covca+covcb;
| /* Fonction objectif de dimensionnement */
| fobj2 :
| Fdim=ilmax^2/(4*oil^2)+ilmin^2/(4*oil^2)+vcamax^2/(4*ovca^2)+vcbmax^2/(4*ovcb^2)
| ;
|
| /* FONCTION OBJECTIF FINALE = combinaison des 2 fonctions objectif precedemment
| definies
| fait intervenir 2 parametres ponderaux pphy et pdim qui sont alors des
| entrees du modele
| */
| fobj0 : F=pphy*Fphy+pdim*Fdim;
|
| /* ECRITURE DU FICHIER EQTEST.MAC */
| load(gentran);
|
| delfile("h:\prethese\multires\multid1\donnees\eqtest.mac");
|
| GENTRANOUT("h:\prethese\multires\multid1\donnees\eqtest.mac");
| gentranlang:c;
| clinelen:30000;
| maxexpprintlen:4000;
| gentran(literal(cr, "pow(a,b) :=a^b;", cr));
|
| gentran(literal(cr, "conail:", eval(lhs(conail)), "=eval(", eval(rhs(conail)), ")", cr,
| r));
| gentran(literal(cr, "conbil:", eval(lhs(conbil)), "=eval(", eval(rhs(conbil)), ")", cr,
| r));

```

```

gentran(literal(cr,"concil:",eval(lhs(concil)),"=eval(",eval(rhs(concil)),"");",cr));
gentran(literal(cr,"condil:",eval(lhs(condil)),"=eval(",eval(rhs(condil)),"");",cr));

gentran(literal(cr,"conavca:",eval(lhs(conavca)),"=eval(",eval(rhs(conavca)),"");",cr));
gentran(literal(cr,"conbvca:",eval(lhs(conbvca)),"=eval(",eval(rhs(conbvca)),"");",cr));
gentran(literal(cr,"concvca:",eval(lhs(concvca)),"=eval(",eval(rhs(concvca)),"");",cr));
gentran(literal(cr,"condvca:",eval(lhs(condvca)),"=eval(",eval(rhs(condvca)),"");",cr));

gentran(literal(cr,"conavcb:",eval(lhs(conavcb)),"=eval(",eval(rhs(conavcb)),"");",cr));
gentran(literal(cr,"conbvcb:",eval(lhs(conbvcb)),"=eval(",eval(rhs(conbvcb)),"");",cr));
gentran(literal(cr,"concvcb:",eval(lhs(concvcb)),"=eval(",eval(rhs(concvcb)),"");",cr));
gentran(literal(cr,"condvcb:",eval(lhs(condvcb)),"=eval(",eval(rhs(condvcb)),"");",cr));

gentran(literal(cr,"eqnta:",eval(lhs(eqnta)),"=eval(",eval(rhs(eqnta)),"");",cr));
;
gentran(literal(cr,"eqntb:",eval(lhs(eqntb)),"=eval(",eval(rhs(eqntb)),"");",cr));
;
gentran(literal(cr,"eqntc:",eval(lhs(eqntc)),"=eval(",eval(rhs(eqntc)),"");",cr));
;
gentran(literal(cr,"eqntd:",eval(lhs(eqntd)),"=eval(",eval(rhs(eqntd)),"");",cr));
;
gentran(literal(cr,"eqnt:",eval(lhs(eqnt)),"=eval(",eval(rhs(eqnt)),"");",cr));
gentran(literal(cr,"eqalpha:",eval(lhs(eqalpha)),"=eval(",eval(rhs(eqalpha)),"");",cr));

gentran(literal(cr,"testabton:",eval(lhs(testabton)),"=eval(",eval(rhs(testabton)),",");",cr));
gentran(literal(cr,"testabdaof:",eval(lhs(testabdaof)),"=eval(",eval(rhs(testabdaof)),",");",cr));
gentran(literal(cr,"testabdbon:",eval(lhs(testabdbon)),"=eval(",eval(rhs(testabdbon)),",");",cr));

gentran(literal(cr,"testbctof:",eval(lhs(testbctof)),"=eval(",eval(rhs(testbctof)),",");",cr));
gentran(literal(cr,"testbcdaof:",eval(lhs(testbcdaof)),"=eval(",eval(rhs(testbcdaof)),",");",cr));
gentran(literal(cr,"testbcdbon:",eval(lhs(testbcdbon)),"=eval(",eval(rhs(testbcdbon)),",");",cr));

gentran(literal(cr,"testcdtof:",eval(lhs(testcdtof)),"=eval(",eval(rhs(testcdtof)),",");",cr));
gentran(literal(cr,"testcddbfof:",eval(lhs(testcddbfof)),"=eval(",eval(rhs(testcddbfof)),",");",cr));
gentran(literal(cr,"testcddaon:",eval(lhs(testcddaon)),"=eval(",eval(rhs(testcddaon)),",");",cr));

gentran(literal(cr,"testdatof:",eval(lhs(testdatof)),"=eval(",eval(rhs(testdatof)),",");",cr));
gentran(literal(cr,"testdadaof:",eval(lhs(testdadaof)),"=eval(",eval(rhs(testdadaof)),",");",cr));
gentran(literal(cr,"testdadbon:",eval(lhs(testdadbon)),"=eval(",eval(rhs(testdadbon)),",");",cr));

```



```

gentran(literal(cr,"eqawil:",eval(lhs(eqawil)),"=eval(",eval(rhs(eqawil)),"");",c
r));
gentran(literal(cr,"eqbwil:",eval(lhs(eqbwil)),"=eval(",eval(rhs(eqbwil)),"");",c
r));
gentran(literal(cr,"eqcwil:",eval(lhs(eqcwil)),"=eval(",eval(rhs(eqcwil)),"");",c
r));
gentran(literal(cr,"eqdwil:",eval(lhs(eqdwil)),"=eval(",eval(rhs(eqdwil)),"");",c
r));

gentran(literal(cr,"eqawvca:",eval(lhs(eqawvca)),"=eval(",eval(rhs(eqawvca)),"");
",cr));
/*gentran(literal(cr,"eqbwvca:",eval(lhs(eqbwvca)),"=eval(",eval(rhs(eqbwvca)),"
");",cr));*/
gentran(literal(cr,"eqcwvca:",eval(lhs(eqcwvca)),"=eval(",eval(rhs(eqcwvca)),"");
",cr));
gentran(literal(cr,"eqdwvca:",eval(lhs(eqdwvca)),"=eval(",eval(rhs(eqdwvca)),"");
",cr));

gentran(literal(cr,"eqawvcb:",eval(lhs(eqawvcb)),"=eval(",eval(rhs(eqawvcb)),"");
",cr));
gentran(literal(cr,"eqbwvcb:",eval(lhs(eqbwvcb)),"=eval(",eval(rhs(eqbwvcb)),"");
",cr));
gentran(literal(cr,"eqcwvcb:",eval(lhs(eqcwvcb)),"=eval(",eval(rhs(eqcwvcb)),"");
",cr));
gentran(literal(cr,"eqdwvcb:",eval(lhs(eqdwvcb)),"=eval(",eval(rhs(eqdwvcb)),"");
",cr));

gentran(literal(cr,"eqawtil:",eval(lhs(eqawtil)),"=eval(",eval(rhs(eqawtil)),"");
",cr));
gentran(literal(cr,"eqbwttil:",eval(lhs(eqbwttil)),"=eval(",eval(rhs(eqbwttil)),"");
",cr));
gentran(literal(cr,"eqcwttil:",eval(lhs(eqcwttil)),"=eval(",eval(rhs(eqcwttil)),"");
",cr));
gentran(literal(cr,"eqdwttil:",eval(lhs(eqdwttil)),"=eval(",eval(rhs(eqdwttil)),"");
",cr));

gentran(literal(cr,"eqawtvca:",eval(lhs(eqawtvca)),"=eval(",eval(rhs(eqawtvca))
,");",cr));
/*gentran(literal(cr,"eqbwtvca:",eval(lhs(eqbwtvca)),"=eval(",eval(rhs(eqbwtvca)
),");",cr));*/
gentran(literal(cr,"eqcwtvca:",eval(lhs(eqcwtvca)),"=eval(",eval(rhs(eqcwtvca))
,");",cr));
gentran(literal(cr,"eqdwtvca:",eval(lhs(eqdwtvca)),"=eval(",eval(rhs(eqdwtvca))
,");",cr));

gentran(literal(cr,"eqawtvcb:",eval(lhs(eqawtvcb)),"=eval(",eval(rhs(eqawtvcb))
,");",cr));
gentran(literal(cr,"eqbwtvcb:",eval(lhs(eqbwtvcb)),"=eval(",eval(rhs(eqbwtvcb))
,");",cr));
gentran(literal(cr,"eqcwtvcb:",eval(lhs(eqcwtvcb)),"=eval(",eval(rhs(eqcwtvcb))
,");",cr));
gentran(literal(cr,"eqdwtvcb:",eval(lhs(eqdwtvcb)),"=eval(",eval(rhs(eqdwtvcb))
,");",cr));

gentran(literal(cr,"eqtilmax:",eval(lhs(eqtilmax)),"=eval(",eval(rhs(eqtilmax))
,");",cr));
gentran(literal(cr,"eqntilmax:",eval(lhs(eqntilmax)),"=eval(",eval(rhs(eqntilmax
)),");",cr));
gentran(literal(cr,"eqilmax:",eval(lhs(eqilmax)),"=eval(",eval(rhs(eqilmax)),"");
",cr));

```

```
gentran(literal(cr,"eqtilmin:",eval(lhs(eqtilmin)),"=eval(",eval(rhs(eqtilmin)),
");",cr));
gentran(literal(cr,"eqntilmin:",eval(lhs(eqntilmin)),"=eval(",eval(rhs(eqntilmin
)),");",cr));
gentran(literal(cr,"eqilmin:",eval(lhs(eqilmin)),"=eval(",eval(rhs(eqilmin)),");
",cr));

gentran(literal(cr,"eqtvcamax:",eval(lhs(eqtvcamax)),"=eval(",eval(rhs(eqtvcamax
)),");",cr));
gentran(literal(cr,"eqntvcamax:",eval(lhs(eqntvcamax)),"=eval(",eval(rhs(eqntvca
max)),");",cr));
gentran(literal(cr,"eqvcamax:",eval(lhs(eqvcamax)),"=eval(",eval(rhs(eqvcamax)),
");",cr));

gentran(literal(cr,"eqtvcbmax:",eval(lhs(eqtvcbmax)),"=eval(",eval(rhs(eqtvcbmax
)),");",cr));
gentran(literal(cr,"eqntvcbmax:",eval(lhs(eqntvcbmax)),"=eval(",eval(rhs(eqntvcb
max)),");",cr));
gentran(literal(cr,"eqvcbmax:",eval(lhs(eqvcbmax)),"=eval(",eval(rhs(eqvcbmax)),
");",cr));

gentran(literal(cr,"fobj1:",eval(lhs(fobj1)),"=eval(",eval(rhs(fobj1)),");",cr))
;
gentran(literal(cr,"fobj2:",eval(lhs(fobj2)),"=eval(",eval(rhs(fobj2)),");",cr))
;
gentran(literal(cr,"fobj0:",eval(lhs(fobj0)),"=eval(",eval(rhs(fobj0)),");",cr))
;

GENTRANSHUT("h:\\prethese\\multires\\multid1\\donnees\\eqtest.mac");
remarray(res);
```

## ANNEXE D

### TABLEAU DE CORRESPONDANCE ENTRE LES PARAMETRES PASCOSMA ET CEUX PRESENTES AU CHAPITRE II POUR L'APPLICATION DU ZVS-MRC

Paramètres	PASCOSMA	Chapitre II	Description
Entrées	ca	C1	Valeur de la capacité $C_1$
	cb	C2	Valeur de la capacité $C_2$
	e	E	Valeur de la tension d'entrée E
	g	G	Valeur du courant g
	ial	iL10	Condition initiale du courant dans l'inductance $L_1$
	la	L1	Valeur de l'inductance $L_1$
	nta	$\tilde{t}_1$	Les durées normalisées des configurations
	ntb	$\tilde{t}_2$	
	ntc	$\tilde{t}_3$	
	ntd	$\tilde{t}_4$	
	ntilma	$\tilde{t}_{ext}$	Les dates normalisées des extremums
	ntilmi		
	ntvcama		
	ntvcbma		
	oida	Non présentés	Les ordres de grandeur des crêtes pour les signaux (la normalisation est réalisée à partir de ces paramètres d'entrée)
	oidb		
	oil		
	oit		
	ovca		
	ovcb		
ovda			
ovt			

	pdim	$w_{dim}$	Les facteurs de pondération de la fonction objectif
	pphy	$w_{phys}$	
	t	$T$	La période de fonctionnement
	vaca	vC10	Conditions initiales des tensions aux bornes de condensateurs
	vacb	vC20	
Critères	coil	ContiL1	Les continuités des variables d'état
	covca	ContvC1	
	covcb	ContvC2	
	daofab	Non-amorçage	Les tests de commutations pour la diode D1
	daofbc	Non-amorçage	
	daoncd	Amorçage	
	dbonab	Amorçage	Les tests de commutations pour la diode D2
	dbonbc	Non-blocage	
	dbonda	Non-blocage	
	fdim	$f_{dim}$	La fonction objectif de dimensionnement
	fphy	$f_{phys}$	La fonction objectif physique
	nt	$\tilde{T}$	La période normalisée
	tgilmax	$\tilde{S}$	Les pentes normalisées des signaux aux dates des extremums
	tgilmin		
	tgvca		
	tgvcb		
	tofbc	Blocage	Les tests de commutation dans le thyristor dual
	tofcd	Non-amorçage	
	tofda	Non-amorçage	
	tonab	Amorçage	
	ilamax	$i_{L1 \max}$	Courant maxi dans $L_1$
	vcamax	$v_{C1 \max}$	Tension maxi dans $C_1$
	vcbmax	$v_{C2 \max}$	Tension maxi dans $C_2$
	watil	Non présentés	Des pulsations normalisées (similaire à la notion de période normalisée a été introduite dans le chapitre II)
	watvca		
	watvcb		
	wbttil		
wbtvcb			
wctil			

## NOTES



## **Résumé**

Ce travail de thèse propose une méthodologie de dimensionnement s'adaptant à un grand nombre de structures de convertisseurs statiques. Elle intègre différents outils préexistants, et s'appuie sur la formulation automatique, avec des techniques de calcul symbolique, d'un modèle de dimensionnement. Celui-ci prend en charge, à un même niveau, l'aspect physique du fonctionnement et l'aspect conception qui s'exprime en termes de respect d'un cahier des charges. La connaissance fondamentale pour mettre en œuvre cette démarche se veut minimale, et comprend les équations des nœuds du circuit et les spécifications du cahier des charges. L'intervention de l'utilisateur dans la méthodologie est réduite au maximum et simplifiée. Sur ces principes, cette approche de CAO cherche à fournir au concepteur, via l'ordinateur, en plus de logiciels d'analyse, un véritable outil d'aide au dimensionnement intégrable dans une méthodologie globale de conception des convertisseurs statiques.

## **Mots clefs :**

Dimensionnement, optimisation, CAO, calcul symbolique, convertisseur statique, génération automatique de modèle

## **Abstract**

This work defines and carries out a methodology for the sizing of static converter structures. It integrates several pre-existing tools, and is based on the automatic formulation with symbolical calculation techniques of a sizing model. This one takes into account - at the same level - the physical aspect of the operating and the design aspect in terms of the specification board requirements. The basic knowledge to complete the approach is minimal, it is made of the node equations of the circuit and the specification board. In the methodology, the user actions are reduced and simplified as much as possible. From those principles, added to the analysis packages, a sizing aided tool is given to the designer. This whole set of tools can be integrated in a global approach for the CAD of static converters.

## **Keywords :**

Sizing, optimisation, CAD, symbolic calculation, static converter, automatic building of models