# ANALYSE FONCTIONNELLE DES CONVERTISSEURS STATIQUES EN VUE DE LA CONCEPTION

Christophe Lechevalier

▶ **To cite this version:**

Christophe Lechevalier. ANALYSE FONCTIONNELLE DES CONVERTISSEURS STATIQUES EN VUE DE LA CONCEPTION. Energie électrique. Institut National Polytechnique de Grenoble - INPG, 1998. Français. NNT : . tel-00820608

**HAL Id: tel-00820608**

**https://theses.hal.science/tel-00820608**

Submitted on 6 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

Christophe LECHEVALIER
Ingénieur E.N.S.I.E.G

pour obtenir le grade de DOCTEUR

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(*Arrêté ministériel du 30 mars 1992*)

Spécialité Génie Electrique

---

# ANALYSE FONCTIONNELLE DES CONVERTISSEURS STATIQUES EN VUE DE LA CONCEPTION

---

Le 30 novembre 1998

Composition du jury :

| | | |
|---|---|---|
| Messieurs : | H. MOREL | Président et Rapporteur |
| | H. FOCH | Rapporteur |
| | J. ROUDET | |
| | R. DARNAULT | |
| | L. GERBAUD | |
| | J. BIGEON | |

Thèse préparée au sein du Laboratoire d'Electrotechnique de Grenoble

Pour Jaime

Mon collègue

du bureau.

Christophe

Je tiens à remercier,

Monsieur Hervé MOREL, Charge de recherche au CNRS, rattaché au CEGELY, qui m'a fait l'honneur d'être rapporteur et de présider le jury de cette thèse.

Monsieur Henry FOCH, Professeur à L'INP de Toulouse et directeur du LEEI, qui a accepté d'être rapporteur de cette thèse.

Monsieur James ROUDET, Professeur UJF, pour sa participation à ce jury et de thèse et pour les discussions que nous avons eues.

Monsieur Régis DARNAULT, Docteur Ingénieur de la société CECLA, pour avoir accepté de participer à ce jury de thèse et pour m'avoir initié au fonctionnement de la micro informatique du L.E.G.

Monsieur Jean BIGEON, Directeur de recherche au CNRS, et directeur de l'équipe Conception et Diagnostic Intégrés du L.E.G. Je tiens particulièrement à le remercier pour la confiance qu'il m'a accordée, tout d'abord en m'accueillant dans son équipe, puis au long de ces années. Je ne citerai pas ici ses nombreuses qualités par peur d'en oublier, mais je dirai seulement qu'il sait instaurer dans son équipe un climat convivial alliant franchise, écoute, confiance, exigence qui permet à chacun d'exprimer ses qualités. Pendant ces années, il a toujours été disponible pour répondre à mes questions de nature très diverses.

Monsieur Laurent GERBAUD, Chargé de recherche CNRS. Bien plus qu'un responsable comme il aime à se définir, il a été pour moi un soutien sans faille pour faire avancer mon travail. Mon aventure avec lui a commencé par un projet de D.E.A. pour développer son « bébé : Gentiane » puis a continué par ce travail de thèse. Les nombreuses discussions que nous avons eues, ont été des occasions uniques de partager une quantité considérable d'idées. S'il fallait décrire Laurent en trois mots je citerai : enthousiasme, rigueur, disponibilité. Plutôt que de m'exprimer encore plus sur Laurent je voudrai simplement lui dire Merci.

Mes remerciements ne seraient pas complets sans les personnes qui ont eu un rôle important pendant ce travail.

Monsieur Albert Foggia, pour la confiance qui m'a accordé en me chargeant d'une étude, et les discussions scientifiques que nous avons pu avoir.

Je tiens particulièrement à remercier Claude Morand et Hervé Roux-Buisson pour avoir été mes tuteurs pendant mon activité d'enseignement à l'U.J.F. en tant que moniteur. Ils ont toujours été disponibles pour répondre à mes questions et m'ont accordé toute leur confiance.

Merci aux administrateurs du réseau PC : Marie-Thérese Lobinoux, Florence Francois et Bruno Ferrari. Ils m'auront permis d'apprendre beaucoup de chose sur le fonctionnement et l'organisation d'un réseau.

Une prime spéciale pour Alain Bolopion qui m'a fait l'honneur de relever le défi de relire ce manuscrit avec des délais records. Il a toujours été disponible pour répondre à mes nombreuses questions.

Frédéric Wurtz pour les discussions scientifiques et les nombreux conseils qu'il m'a donnés sur Pascosma (ne pas prononcer les 's') et l'utilisation de l'environnement Ilog avec ses nombreux pièges.

Comment l'oublier ! Je remercie mon collègue de bureau Jaime Fandino. Pendant ces trois ans, notre cohabitation a toujours été dans la joie et la bonne humeur. Nos discussions ont toujours été intéressantes et une source inépuisable de détente.

Je tiens à remercier mon compagnon d'arme, Stéphane Bergeon aussi appelé « Stéphane le magnifique ». Sous son air calme et réfléchi se cache un très grand séducteur et un bon vivant. Pendant 3 ans nous avons eu des discussions de travail, des moments moins studieux, mais c'était un moyen efficace de nous motiver mutuellement.

Patrick Podczaki pour les moments que nous avons partagés. S'il n'avait pas choisi une carrière dans le domaine technique, je crois qu'il aurait pu être un auteur à succès de sitcom.

Merci à Corinne Perret. Je ne sais pas comment la décrire sans dégrader la réalité. Je dirai simplement : Corinne merci pour tout ...

Je tiens à remercier Eric Atienza qui je pense pourra très bien me remplacer (et même faire mieux que moi) dans l'animation de la salle micro. Entamer une polémique avec lui est une épreuve de force très éprouvante, mais une fois la discussion finie, payez lui un café et tout rentre dans l'ordre. Si un jour vous avez quelques minutes, essayez de débattre avec à propos de son super moteur asynchrone ...

Jean-Michel Guichon aussi appelé le maître de l'overclocking. Derrière son air de Saint, il se cache un maître de la technologie multimédia sur PC. Si un jour vous avez envie de monter une machine très puissante pour jouer à Quake XX avec un maximum d'images par secondes, alors allez le voir.

Les membres de l'équipe CDI que je n'ai pas encore cités : Basma Belhabib, Coralie Coutel, Christophe Sauvey, Jean Marc Bouché.

Merci aux autres thésards du LEG avec qui j'ai partagé discussions, déjeuners, soirées, et autre : je peux citer par exemple Ambroise Schellmanns, Patrick Froissier, Raphaël Peuget, Gwénaelle Le Coat, Charlotte Gillot, Nadège Piette, Séverine Guffon, Fabrice LeDoze, Jean Christophe Crebier, Alain Ground, Joël Conrad, Jean Schutz, Nicolas Retière, Samuel Nugues, Amando Fonseca et Bertrand Raison pour leur innombrables blagues...

Merci à toutes les personnes qui m'ont soutenu tout au long de mon parcours : Brice, Claude, José, Yazid, Emeric, Manu, Nadine, Stephi, ...

Enfin merci à ceux que j'ai oublié par inadvertance, mais ils se reconnaîtront.

A ma Famille

# Sommaire


LISTE DES ILLUSTRATIONS 7

GLOSSAIRE 11

INTRODUCTION GÉNÉRALE 15

CHAPITRE 1 : POSITIONNEMENT DU PROBLÈME 17

I. La conception des convertisseurs statiques 18
I.1. Présentation des convertisseurs statiques 18
I.1.1. Principe 18
I.1.2. Applications 19
I.1.3. Les points délicats du fonctionnement 21
I.2. Les problèmes liés à la conception de convertisseurs statiques 22
I.3. Quelques méthodes de conception en électronique de puissance 23
I.3.1. Utilisation des cellules de commutation 23
I.3.1.1. Présentation 23
I.3.1.2. Utilisation et avantages 24
I.3.1.3. Exemple de système utilisant cette approche 24
I.3.1.4. Limites par rapport à notre vision du problème de conception des structures 25
I.3.2. L'approche matricielle [FRA96] 26
I.3.2.1. Présentation 26
I.3.2.2. Utilisation et avantages 27
I.3.2.3. Limites par rapport à notre vision du problème de conception. 27
I.3.3. Les approches systèmes experts pour les entraînements électromécaniques 28

II. L'analyse fonctionnelle dans le processus de conception en génie électrique 28

II.1. Description générale du processus de conception 28
II.1.1. La définition du cahier des charges 30
II.1.2. La définition ou la modification de la structure 30
II.2. Quelques méthodes de conception ou d'analyse avec une approche fonctionnelle 31
II.2.1. Une modélisation fonctionnelle d'un dispositif de l'électromagnétisme 31
II.2.1.1. Présentation 31
II.2.1.2. Evaluation par rapport à notre problème 33
II.2.2. Méthode SADT 33
II.2.2.1. Présentation 33
II.2.2.2. Evaluation par rapport notre problème 35
II.2.3. Une approche déjà formalisée en mécanique 36
II.2.3.1. présentation 36
II.2.3.2. Evaluation par rapport à notre problème 37
II.2.4. L'approche fonctionnelle pour la conception en général 38
III. Une méthode de modélisation pour l'analyse : les graphes de liens 38
III.1. Présentation 38
III.1.1. Effort/flux 39
III.1.2. La représentation graphique 40
III.1.3. La causalité 40
III.1.4. Les éléments 40
III.2. Utilisation 42
III.3. Limites par rapport à nos besoins 43
IV. Notre problématique pour une approche fonctionnelle 43
IV.1. Les enjeux de notre travail 44
IV.2. Les hypothèses de travail 44
IV.3. Les avantages de notre approche 45
IV.4. Les limites et les points délicats 46
IV.5. La méthode de travail 48

**CHAPITRE 2 : LES FONCTIONS 51**

I. Définition des notions fondamentales utilisées pour notre application 51
I.1. Les grandeurs de références de l'approche proposée : Effort / Flux 51
I.2. Notions complémentaires pour la modélisation fonctionnelle 52
I.2.1. Notion de topologie 52
I.2.2. Notion de fonctionnalité 53
I.2.3. Notion de technologie 53
I.2.4. Notion de contrainte 54
I.3. Point de vue fonctionnel 55
I.4. Point de vue structurel 56
II. Modélisation des fonctions 57
II.1. Présentation 57
II.2. Les constituants 59
II.2.1. La classe Forme_d_onde : modélisation de l'évolution des grandeurs 59

II.2.2. La Classe Couple_EF : identification des points de sortie des interfaces 63
II.2.2.1. Définition 63
II.2.2.2. Représentation graphique 63
II.2.3. La classe Interface : regroupement des objets Couple_EF dépendants d'une fonction 64
II.2.4. La modélisation des contraintes énergétique 66
II.2.4.1. La classe Lien_de_contrainte : Description des actions d'une interface entre deux Couple_EF 66
II.2.4.2. La classe Etat_de_contrainte : modélisation de l'évolution dans le temps des objets Lien_de_contrainte 68
II.2.4.3. La classe Transition : modélisation des conditions de changement d'objets Etat_de_contrainte 70
II.2.4.4. Représentation graphique de contraintes modélisées par les objets Lien_de_contrainte 71
II.3. La classe Fonction_generique : Classe principale de caractérisation des fonctions 71
II.3.1. La classe Fonctions_energie 73
II.3.2. La classe Fonction_stocker_energie 73
II.3.3. La classe Fonction_contrôle_de_flux. 74
II.3.4. La classe Fonctions_de_retour 74
II.3.5. La classe Fonction_gyrateur 76
II.3.5.1. Présentation 76
II.3.5.2. Application des gyrateurs 78
II.3.5.3. Perspective des gyrateurs 79
II.4. Création des objets Fonction : instanciation d'une classe Fonction 80
II.4.1. Principe 80
II.4.2. Création des interfaces 81
II.4.2.1. L'initialisation d'une interface (T.1) 82
II.4.2.2. La création des contraintes (T.2) 83

**CHAPITRE 3 : LE GRAPHE FONCTIONNEL 85**

I. Les éléments d'un graphe fonctionnel 85
I.1. Les liens de transfert d'énergie (LTE) 85
I.2. Les connexions 86
I.2.1. Définition 86
I.3. Les boucles 87
I.3.1. Définition 87
I.3.2. Le point de vue topologique 87
I.3.3. Le point de vue énergétique 88
I.3.3.1. La causalité 88
I.3.3.2. Le niveau des flux 90
I.4. Les Liens temporels 93
I.4.1. Présentation 93
I.4.2. Les différents liens temporels 93
I.4.2.1. Le Lien de Simultanéité (LS). 93
I.4.2.2. Le Lien de Non-Simultanéité (LNS) 94

I.4.2.3. Le Lien Séquentiel (LSeq). 94

II. Le graphe fonctionnel 95
II.1. La modélisation d'un graphe fonctionnel 96
II.1.1. Composition 96
II.1.2. Intérêt de la décomposition du graphe en boucles 96
II.2. La méthodologie de construction d'un graphe fonctionnel 97
II.3. La construction des boucles 99
II.3.1. Présentation 99
II.3.2. Principe 100
II.4. La Vérification de la cohérence énergétique 103
II.4.1. Construction des boucles implicites 103
II.4.1.1. Présentation des boucles implicites 103
II.4.1.2. Principe de la recherche des boucles implicites 105
II.4.2. Les vérifications sans connaître le mode de fonctionnement 108
II.4.3. Les vérifications à partir du mode de fonctionnement 110

III. Construction de la solution finale 111
III.1. Les éléments de la solution finale 111
III.2. Le lien fonctionnel / structurel 113
III.2.1. De la fonction vers les structures 113
III.2.1.1. Présentation 113
III.2.1.2. Le cas des fonctions de contrôle du flux 114
III.2.2. Le lien inverse : structurel-fonctionnel 116

**CHAPITRE 4 : LE PROTOTYPE INFORMATIQUE 119**

I. L'environnement de programmation 119
I.1. Nos critères de choix 119
I.2. L'environnement de programmation ILOG-LELISP 121
I.2.1. Présentation 121
I.2.1.1. LE-LISP [LEL94] 121
I.2.1.2. AIDA [AID94] 121
I.2.1.3. MASAI [MAS94]I 122
I.2.1.4. SMECI [SME94] 122
I.2.2. Concepts utilisés dans l'environnement 123

II. Réalisation du prototype 123
II.1. L'implantation du modèle fonctionnel 123
II.1.1. La définition dynamique des classes de fonctions 124
II.1.2. La définition dynamique des domaines énergétiques 126
II.1.3. La gestion dynamique des règles 126
II.2. Les interactions graphiques de notre prototype 129
II.2.1. La couche objet du noyau 129
II.2.2. La couche graphique pour l'édition de valeurs des champs 130
II.2.3. La gestion interactive du graphe fonctionnel 131

III. Evolution de l'implantation 134

**CHAPITRE 5 : DES EXEMPLES DE CONCEPTION FONCTIONNELLE 137**

I. Exemple de conception d'une structure avec isolation galvanique 138
I.1. Cahier des charges 138
I.2. La construction du graphe fonctionnel 140
I.2.1. Etape 1 : La liaison entre la fonction « source » et la fonction « isolation ». 140
I.2.2. Etape 2 : La liaison entre la fonction « isolation » - et la fonction « charge ». 141
I.2.3. Etape 3.1 : Analyse de la cohérence avec la fonction « charge en courant » 143
I.2.3.1. La contrainte sur le flux de la charge en courant 143
I.2.3.2. Analyse de la Boucle T1 avec la topologie du cas 1 143
I.2.3.3. Avec la topologie du cas 2 146
I.2.4. Etape 3.2 : Analyse de la cohérence avec la charge en tension 147
I.2.4.1. Analyse de la Boucle T1 avec la topologie du cas 1 147
I.2.4.2. Analyse de la Boucle T1 avec la topologie du cas 2 148
I.3. La construction des solutions structurelles 149
I.3.1. Une solution structurelle avec la charge en courant 149
I.3.2. Une solution structurelle avec la charge en tension 152
II. Exemple d'un convertisseur continu – continu 152
II.1. Cahier des charges 152
II.2. La construction du graphe fonctionnel 154
III. Conclusion 158

**CONCLUSION GÉNÉRALE ET PERSPECTIVES 159**

**BIBLIOGRAPHIE 163**

**ANNEXES 169**

Annexe I : Les graphes de liens 170
Annexe II : La décomposition structurelle 171
II.1. Un extrait des classes de service pour la caractérisation des classes de structures 172
II.1.1. La classe Segment 172
II.1.2. La Classe Quadrant 173
II.1.2.1. La Classe Quadrant Alternatif 173
II.1.2.2. La classe Quadrant Continu 174
II.2. Structure Cellule Hacheur 175
II.2.1. Structure Cellule Hacheur Tension 175
II.2.1.1. Structure Cellule Hacheur Tension avec Isolation 177

II.2.2. Structure Cellule Hacheur Courant 179
II.2.2.1. Structure Cellule Hacheur Courant avec Isolation 181
Annexe III : Les classes participant à la modélisation des fonction 183
III.1. Les classes élémentaires de notre modèle fonctionnel 183
III.2. La classe Formes_d_onde 184
III.3. La classe Niveau 185
III.4. La classe Couple_EF 188
III.5. La classe Interface 189
III.6. La modélisation des contraintes 190
III.6.1. La classe Contrainte_cyclique 190
III.6.2. La classe Lien_de_contrainte 190
III.6.3. La Classe Etat_de_contrainte 191
III.7. La Classe Transition 193
III.8. La classe Fonction 193
Annexe IV : Les classes utiles à la modélisation du graphe fonctionnel 194
IV.1. La classe Connexion 194
IV.2. Les Boucles 194
IV.2.1. La classe Boucle 194
IV.2.2. Le point de vue topologique 195
IV.2.3. Le point de vue énergétique 198
IV.3. La Classe Lien_temporels 200
IV.4. La Classe Graphe_fonctionnel 201
Annexe V : La définition des différentes classes du type niveau 203

# Liste des illustrations

## Liste des figures

Figure 1.1 : Le processus général de conception. 17
Figure 1.2 : Les notions de structure et de topologie 19
Figure 1.3 : Une commutation réelle 21
Figure 1.4 : Schéma d'une cellule de commutation élémentaire 23
Figure 1.5 : Schéma d'une cellule de commutation à plusieurs sources 24
Figure 1.6 : La topologie de l'alimentation à concevoir [FEZ 96A] 25
Figure 1.7 : Un convertisseur matriciel de principe 27
Figure 1.8 : Un processus de conception d'un convertisseur statique 29
Figure 1.9 : Une décomposition structurelle d'un contacteur [BRU94] 31
Figure 1.10 : La décomposition fonctionnelle proposée pour le contacteur 32
Figure 1.11 : Exemple de lien fonction-structure 33
Figure 1.12 :La symbolisation des actigrammes 34
Figure 1.13 :La symbolisation des datagrammes 35
Figure 1.14 : Le diagramme SADT de la commande d'un redresseur 35
Figure 1.15 : Quelques variables de flux et d'effort [LAU98] 39
Figure 1.16 : Représentation graphique d'un lien Bond Graph 40
Figure 1.17 : Les situations de causalité de la jonction 1 41
Figure 1.18 : Les situations de causalité de la jonction 0 41
Figure 1.19 : Un exemple simple de circuit électrique 42
Figure 1.20 : Le Graphe de lien correspondant au circuit de la figure 1.19 42
Figure 1.21 : Notre démarche pour la modification d'une structure 45
Figure 2.1 :Exemples des deux types de technologie 54
Figure 2.2 : Un graphe fonctionnel de principe d'un convertisseur statique 55
Figure 2.3 :Modélisation objet d'une fonction contrôle de flux avec présentation de quelques liens entre les objets 58
Figure 2.4 : Définition d'une grandeur sinusoïdale 59
Figure 2.5 : Exemple d'une forme d'onde évoluée 60
Figure 2.6 : Les différents niveaux 61
Figure 2.7 :Décomposition d'une forme d'onde sinus approché 61
Figure 2.8 : Décomposition d'une forme MLI 62
Figure 2.9 : Représentation des objets Couple_EF 64
Figure 2.10 : Organisation de la classe Interface 66
Figure 2.11 : La hiérarchie des classes lien_de_contraintre 68
Figure 2.12 : Définition des objets Etats_de_contrainte d'une fonction de contrôle de flux 68
Figure 2.13 : La décomposition en objets Etat_de_contrainte d'une fonction sinusoïdale 69

Figure 2.14 :La même fonction que la précédente avec un déphasage différent des grandeurs ..... 70
Figure 2.15 : Exemple de définition des objets Transition ..... 70
Figure 2.16 : Représentation d'une interface avec ses contraintes ..... 71
Figure 2.17 : hiérarchie des fonctions ..... 72
Figure 2.18 : Exemple d'un graphe fonctionnel mal construit ..... 75
Figure 2.19 : Une illustration de la nécessité de la fonction de retour ..... 76
Figure 2.20 : La représentation structurelle de la fonction isolation magnétique ..... 78
Figure 2.21 : Modélisation fonctionnelle d'un transformateur ..... 79
Figure 2.22 : La décomposition en tâche du processus de création d'une interface ..... 82
Figure 3.1 : Représentation des LTE et des connexions ..... 86
Figure 3.2 : Une boucle simple de démonstration ..... 88
Figure 3.3 : Un exemple de conflit flux ..... 89
Figure 3.4 : Organigramme de la causalité dominante ..... 89
Figure 3.5 :La vérification des flux dans une boucle ..... 90
Figure 3.6 : Les descriptions graphiques des objets Formes_d_onde ..... 91
Figure 3.7 : La modélisation objet des boucles ..... 92
Figure 3.8 : La symbolisation d'un LS ..... 94
Figure 3.9 : La symbolisation d'un LNS ..... 94
Figure 3.10 : La symbolisation des LSeq ..... 95
Figure 3.11 : Un exemple de circuit conflictuel ..... 97
Figure 3.12 : Une démarche possible de conception fonctionnelle ..... 98
Figure 3.13 : Equivalences dans la connexion des gyrateurs ..... 100
Figure 3.14 :Organigramme de la procédure de création des boucles ..... 101
Figure 3.15 : Fonctionnement de la procédure de construction des boucles ..... 102
Figure 3.16 : Un exemple de la fermeture automatique d'une boucle ..... 103
Figure 3.17 : Un exemple de boucles implicites ..... 105
Figure 3.18 : Le réseau de travail correspondant à la figure 3.17 ..... 106
Figure 3.19 : Le principe de la recherche des boucles indépendantes du réseau ..... 107
Figure 3.20 :L'exploration des boucles du réseau de la figure 3.18 ..... 108
Figure 3.21 : La vérification de la contrainte de continuité ..... 109
Figure 3.22 : Le principe de la première étape de la vérification des boucles avec gyrateur ..... 110
Figure 3.23 : Le processus de vérification complète de la cohérence à partir du mode fonctionnement. ..... 111
Figure 3.24 : La séparation des points de vue fonctionnel et structurel ..... 112
Figure 3.25 : Un graphe fonctionnel correspondant au cas standard 1 pour la détermination des interrupteurs ..... 115
Figure 3.26 : Un graphe fonctionnel correspondant au cas standard 2 ..... 116
Figure 4.1 : Description d'une fenêtre avec AIDA ..... 122
Figure 4.2 : Le fichier de définition permettant la création dynamique des sous-classes de la classe Fonction_generique ..... 124
Figure 4.3 : Exemple de règle de définition des interfaces liées à une fonctionnalité ..... 125
Figure 4.4 : La description de la fonction définie par la règle de la figure 4.3 ..... 126
Figure 4.5 : Un fichier de description des domaines énergétiques ..... 126
Figure 4.6 : L'arbre des tâches de l'algorithme de vérification de la causalité d'une boucle ..... 127
Figure 4.7 : La définition de la classe boucle ..... 128
Figure 4.8 : Un exemple de règle de vérification de causalité ..... 128
Figure 4.9 : La fenêtre graphique utilisée pour le paramétrage de la création des interfaces ..... 129
Figure 4.10 : La structure générale de notre application ..... 129
Figure 4.11 : Les algorithmes de sauvegarde et de restauration ..... 130
Figure 4.12 : Schématisation du fonctionnement de l'interaction I.H.M. - Noyau ..... 131

Figure 4.13 : L'éditeur graphique des interfaces et les méthodes de l'objet interface invoquées en cas d'action de l'utilisateur ..... 131
Figure 4.14 : L'IHM attachée au graphe fonctionnel ..... 132
Figure 4.15 : La structure informatique pour la gestion du graphe fonctionnel ..... 133
Figure 4.16 : L'I.H.M. attachée au graphe fonctionnel pour l'édition des boucles ..... 133
Figure 5.1 : La formulation du cahier des charges, la fonction source ..... 138
Figure 5.2 : La formulation du cahier des charges, la fonction isolation ..... 139
Figure 5.3 : La formulation du cahier des charges, les caractéristiques de sortie ..... 140
Figure 5.4 : Le graphe fonctionnel après l'étape 1 ..... 141
Figure 5.5 : Etape 2, introduction de la boucle 2, le graphe fonctionnel dans le cas 1 ..... 142
Figure 5.6 : Etape 2, introduction de la Boucle 2, le graphe fonctionnel dans le cas 2 ..... 142
Figure 5.7 : La Boucle 3 pour assurer la contrainte de flux sur la charge en courant ..... 143
Figure 5.8 : L'analyse de la Boucle T1 avec la charge en courant dans le cas 1 ..... 144
Figure 5.9 : L'utilisation d'une fonction de démagnétisation ..... 145
Figure 5.10 :La démagnétisation en utilisant une boucle de flux auxiliaire et une fonction de « dissipation » ..... 146
Figure 5.11 : L'analyse de la Boucle T1 avec la charge en courant dans le cas 2 ..... 147
Figure 5.12 : La cohérence de la Boucle T1 avec la charge en tension dans le cas 1 ..... 148
Figure 5.13 : La cohérence de la Boucle T1 avec la charge en tension dans le cas 2 ..... 149
Figure 5.14 : La représentation des fonctions de contrôle ..... 150
Figure 5.15 : La décomposition fonctionnelle de la fonction FC 1 ..... 150
Figure 5.16 : Quelques solutions structurelles avec la charge en courant ..... 151
Figure 5.17 : une solution structurelle avec la charge en tension ..... 152
Figure 5.18 : Le cahier des charges ..... 152
Figure 5.19 :La représentation par l'I.H.M. des fonctions du cahier des charges ..... 153
Figure 5.20 : La définition des objets Forme_d_onde ..... 153
Figure 5.21 : La construction d'une boucle avec notre prototype ..... 155
Figure 5.22 : Une fenêtre signalant une erreur de causalité ..... 155
Figure 5.23 : La boucle modifiée ..... 156
Figure 5.24: Le graphe fonctionnel modifié ..... 156
Figure 5.25 : Le résultat de la vérification de cohérence ..... 157

# Liste des tableaux

Tableau 1.1 : Une classification des composants de puissance ..... 20
Tableau 1.2 : Quelques dispositifs pouvants être considérés avec une fonctionnalité de conversion d'énergie. ..... 47
Tableau 1.3 : Le lien fonction structure ..... 47
Tableau 2.1 : Transitions à réaliser pour la forme MLI ..... 63
Tableau 2.2 : Définition des différents Couple_EFs ..... 63
Tableau 2.3 : Principaux types d'interface ..... 65
Tableau 2.4 : La particularisation des interfaces ..... 66
Tableau 2.5 : Quelques particularisations des fonctions énergies ..... 73
Tableau 2.6 : Particularisations des fonctions de stockage ..... 73
Tableau 2.7 : La Propagation des contraintes par les gyrateurs ..... 77
Tableau 2.8 : Quelques définitions d'interfaces à partir des fonctionnalités ..... 81
Tableau 2.9 : Les informations liées aux types des interfaces ..... 81
Tableau 2.10 : Définition des champs des états de contrainte ..... 83
Tableau 3.1 : Exemple de natures physiques de liens de transfert d'énergie ..... 85

Tableau 3.2 : Les états de boucle possibles. ..... 91
Tableau 3.3 : La table de vérité ..... 92
Tableau 3.4 :Les différents types de lien de simultanéité ..... 93
Tableau 3.5 : Les différents types de lien de non simultanéité ..... 94
Tableau 3.6 : Les différents types de lien séquentiels ..... 95
Tableau 5.1 : Les notations utilisées dans les graphes fonctionnels ..... 138

# Glossaire

**Objet :** structure regroupant des données et des actions. Il comporte une partie définissant la structure et une partie comportement de l'objet.

**Instance :** objet

**Classe :** la classe est la description d'une famille d'objets ayant la même structure et le même comportement.

**Instanciation :** l'instanciation est l'opération qui crée un objet (nommé aussi instance) à partir du modèle d'une classe.

**Champ :** variable nommée permettant de mémoriser une donnée

**Méthode :** une méthode est une fonction attachée à une classe pour décrire le comportement des instances.

**Machine virtuelle :** couche logicielle permettant de simuler le comportement d'un ordinateur réel. Cette technique est utilisée notamment pour être indépendant d'une plate-forme matérielle.

**Notion d'héritage :** notion qui permet des créer des classes spécifiques (avec des champs supplémentaires) à partir d'une classe appelée classe mère.

**Classe Couple_EF :** les objets de cette classe sont utilisés pour identifier les points d'entrée-sortie ou de passage des interfaces.

**Classe Etat_de_contrainte :** cette classe permet de modéliser l'évolution des contraintes en terme de causalité dans le temps. La causalité est définie par une grandeur imposée, et une grandeur moule.

**Classe Fonction_generique :** cette classe permet la caractérisation des fonctions, à l'aide d'une fonctionnalité et d'un ensemble d'interfaces.

**Classe Forme_d_onde :** ses objets permettent de modéliser l'évolution des grandeurs dans le temps. Ces dernières sont représentées par une suite d'évolutions élémentaires : les niveaux.

**Classe Interface :** l'interface est une entité qui regroupe les points d'entrée-sortie (Couple_EF) d'une fonction qui possèdent des liens fonctionnels entre eux.

**Classe Lien_de_contrainte :** ses objets permettent de modéliser les contraintes entre les différents objets couple_EF en terme de causalité, niveau de flux et niveau d'effort.

**Classe Lien_de_Contrainte :** description des actions d'une interface entre deux objets Couple_EF.

**Classe Niveau :** les objets de cette classe permettent de modéliser l'évolution qualitative des grandeurs. Elément de base pour modéliser les formes d'onde.

**Classe Transition :** modélisation des conditions de changement d'objets Etat_de_contrainte

**Fonctionnalité :** rôle essentiel d'une fonction.

**Classe Boucle :** cette classe permet de modéliser les parcours de flux dans un graphe fonctionnel permettant un transfert d'énergie.

**Classe Connexion :** un objet connexion permet de matérialiser les nœuds d'un graphe.

**Lien de transfert d'énergie (LTE) :** chemin d'un flux d'énergie d'une fonction à une autre. Celui-ci est matérialisé structurellement par un support physique, par exemple un fil électrique.

**Technologie fonctionnelle :** nous définirons de cette façon les choix technologiques nécessaires pour faire avancer le processus de conception fonctionnel.

**Point de vue fonctionnel :** le point de vue fonctionnel d'un graphe est un ensemble de fonctions reliées entre elles par des liens énergétiques via les interfaces.

**Point de vue structurel :** celui-ci définit une solution technique complète (prête au dimensionnement) directement issue du graphe fonctionnel final ; tous ses composants sont spécifiés et un mode de fonctionnement est choisi.

**Technologie structurelle :** cette technologie est un choix technique pour réaliser une fonction.

**Topologie :** la topologie est l'ensemble des liens énergétique d'un graphe.

**Contraintes électriques :** ces contraintes permettent de caractériser les actions des fonctions sur leur monde extérieur. Ces contraintes sont définies à partir des formes d'onde et une causalité inspirée de celle des graphes de liens.

**Contraintes temporelles :** ces contraintes permettent de signaler des contraintes énergétiques incompatibles.

**Classe Lien_temporel :** ses objets permettent de matérialiser les contraintes fonctionnelles temporelles entre des fonctions.

**Classe Lien_de_Simultaneite (LS) :** le lien de simultanéité indique que les deux fonctions liées doivent autoriser le passage d'un flux en même temps.

**Classe Lien_de_non_Simultaneite (LNS) :** le lien de non-simultanéité est le lien opposé aux LS. Il signifie donc que les fonctions liées ne doivent pas autoriser un flux en même temps.

**Classe Lien_sequentiel (Lseq) :** le lien séquentiel permet de relier les fonctions qui doivent autoriser un flux de façon séquentielle.

# Introduction générale

La conception des structures de convertisseur statique est une phase délicate de l'activité de conception. Le concepteur est amené à y faire des choix parmi les nombreuses solutions qui lui sont offertes. Couramment il procède de deux façons :

- soit, il modifie une structure déjà existante pour qu'elle réponde à de nouvelles spécifications,
- soit, il procède par assemblage de structures de base.

Plusieurs démarches de ce type ont été formalisées sous forme de systèmes experts. Les outils ainsi obtenus se contentent souvent de réaliser l'étape de pré-dimensionnement. Cela se fait dans un cadre, bien défini en terme de classe de structures et d'applications, et n'offrent pas d'ouverture vers de l'innovation.

Nous proposons ici d'utiliser l'analyse fonctionnelle pour structurer la démarche de conception des structures de convertisseurs statiques, en nous restreignant à l'aspect puissance. En effet, pour les aspects commande, il existe des formalismes très adaptés qui ont fait leurs preuves (SADT). L'approche proposée devra être intégrée dans un environnement plus complet de conception puisqu'elle ne concerne que la phase d'élaboration des structures. A l'avenir, il restera donc à prendre en compte les étapes d'analyse, de dimensionnement et d'expérience.

Le niveau d'abstraction ainsi obtenu nous permet d'être relativement détaché des technologies de fabrication. Ceci permet d'envisager d'obtenir de nouvelles structures et de mieux prendre en compte les évolutions technologiques. A long terme, un système expert construit autour de notre démarche pourra explorer de façon efficace un grand nombre de solutions répondant à un cahier des charges donné.

Le chapitre 1 présente les problèmes liés à la conception de structure de convertisseurs statiques. Dans un premier temps, il analyse quelques méthodes et outils de conception utilisés couramment pour permettre la définition de structure, avec en particulier les méthodes basées sur les cellules de commutation ou les structures matricielles. Dans un second temps, il présente quelques démarches d'analyse fonctionnelle appliquées dans plusieurs domaines :

électromécanique et mécanique. Dans un troisième temps, ce chapitre présente une méthode d'analyse énergétique utilisable pour les convertisseurs statiques : les graphes de liens. Enfin, il introduit les fondements de notre approche fonctionnelle du problème de conception. Pour cela, il pose les hypothèses de travail et analyse les intérêts et limites de l'approche.

Le chapitre 2 commence par spécifier certains termes généraux et par définir les points de vue fonctionnel et structurel d'un convertisseur statique. Puis, il décrit les concepts liés à notre représentation orientée objet des fonctions. Notamment, il présente les différentes classes d'objets que nous avons introduites pour la modélisation des fonctions. Puis, il se termine en présentant les principales classes de fonctions pouvant être utilisées pour élaborer la partie puissance d'un convertisseur statique.

Le chapitre 3 définit les principales classes liées à la définition d'un graphe fonctionnel. A partir de là, il présente la méthode – en partie automatisée – développée pour analyser les graphes, et en particulier vérifier leur cohérence énergétique. Enfin il présente les problèmes liés au passage d'un graphe fonctionnel, qui est un niveau de représentation abstrait, vers une représentation structurelle.

Le chapitre 4 expose la construction d'un prototype d'outil informatique permettant d'effectuer la manipulation des fonctions ainsi que la vérification de la cohérence des graphes fonctionnels. Ce chapitre commence par présenter nos critères pour réaliser l'implantation du modèle. Ensuite, il développe le squelette informatique mis en place afin d'obtenir un système d'aide à la conception facilement extensible et intégrable dans un environnement de conception complet.

Le chapitre 5 présente deux exemples de conception de structure de convertisseurs statiques en utilisant l'analyse fonctionnelle. Ceux-ci illustrent comment le processus de conception se trouve modifié par l'apport de l'analyse fonctionnelle. Afin de présenter l'intérêt de notre approche fonctionnelle, le premier exemple présente une démarche complète de la définition du cahier des charges à la spécification de la structure. Le second exemple présente les traitements pouvant déjà être effectués par notre prototype informatique, qui est encore en cours de développement.

# Chapitre 1 : Positionnement du problème

D'une façon générale, le processus de conception d'un convertisseur statique peut se décomposer en quatre étapes principales comme l'illustre la figure 1.1. A partir d'un cahier des charges défini par le client (l'utilisateur du dispositif), le concepteur doit définir en premier lieu une structure qui peut répondre au problème posé puis dimensionner les différents composants de celle-ci pour enfin pouvoir construire un prototype. Le concepteur est souvent conduit à réaliser plusieurs itérations entre ces étapes.

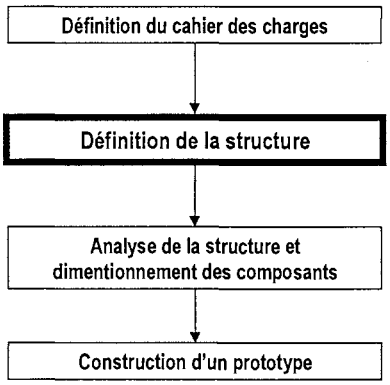



**Figure 1.1 : Le processus général de conception.**

Parmi toutes ces phases, la définition de la structure reste un problème complexe. Le concepteur est amené à effectuer plusieurs choix importants qui conditionnent le reste de l'activité de conception. Il existe différents systèmes experts permettant de l'aider pendant cette phase, mais souvent ces outils se contentent de réaliser du pré-dimensionnement et d'effectuer des choix de composants ; donc ils n'apportent pas des structures vraiment nouvelles.

Dans notre étude, nous proposons d'utiliser l'analyse fonctionnelle pour aider le concepteur dans sa tâche de conception des structures de convertisseurs statiques. Cette méthode d'analyse nous apporte un niveau d'abstraction qui permet de réduire la dépendance vis à vis d'une technologie ou d'une technique de fabrication. Le convertisseur est alors construit à partir des fonctionnalités qu'il doit remplir avec en particulier celle de « transférer de l'énergie ». Comme cette seule fonctionnalité n'est pas suffisante pour définir une structure à elle seule, nous serons amenés à définir les différentes fonctions d'un convertisseur ainsi que le formalisme pour leur manipulation pour arriver finalement à une structure réelle.

Notre but final est de réaliser les bases d'un outil de conception (éventuellement partiellement automatique) de structures de convertisseurs statiques qui devra aider le concepteur. Nous pensons que par l'analyse fonctionnelle, nous obtiendrons une grande flexibilité vis à vis de la technologie, donc des solutions plus aptes à évoluer avec celle-ci. Une approche de ce type nous permet de contrôler l'évolution d'une solution en cours de conception et offre une ouverture vers l'obtention de solutions autres que celles que l'on aurait en raisonnant uniquement sur des critères structurels. La phase suivante sera d'étendre notre modèle fonctionnel pour qu'il puisse prendre en compte les entraînements électromécaniques complets.
Dans ce chapitre, nous commencerons par présenter la problématique liée à la conception de structures de convertisseurs, puis nous examinerons quelques travaux déjà réalisés sur la conception fonctionnelle de dispositifs techniques dans des domaines divers (notamment électromécanique, systèmes d'informations et mécanique). Enfin, la dernière partie de ce chapitre présentera les fondements de notre modèle fonctionnel ainsi que ses limites.

# I. La conception des convertisseurs statiques

Le but de ce paragraphe est de mettre en évidence les problèmes liés à la conception de structures de convertisseurs statiques. Dans un premier temps, nous présenterons leur principe, puis dans un second temps les points délicats et enfin quelques méthodes de synthèse de structures.

## I.1. Présentation des convertisseurs statiques

### I.1.1. Principe

Les convertisseurs statiques ont été conçus pour réaliser des transferts d'énergie de façon contrôlable d'une source électrique vers une charge. Ils sont principalement construit à partir d'un ensemble de semi-conducteurs. L'énergie transférée est contrôlée par commande à l'ouverture et à la fermeture de ces derniers. Dans certains cas, des composants supplémentaires, tels que des condensateurs ou des inductances, peuvent être utilisés pour réaliser des stockages temporaires d'énergie. Le concepteur doit alors agencer tous ces composants pour pouvoir réaliser sa structure. Il existe quatre grands types de conversion [MOH89].

- **AC/DC**. Cette classe permet le transfert de l'énergie d'une source alternative (signal périodique à valeur moyenne nulle) vers une charge continue (signal à valeur moyenne non nulle). Les structures permettant de réaliser ce type de conversion sont les redresseurs.

- **DC/AC**. Ces dispositifs sont les convertisseurs duaux des précédents. Par exemple, les structures réalisant cette opération sont les onduleurs de tension ou les commutateurs de courant.

- **DC/DC**. Ces convertisseurs permettent un transfert d'énergie d'une source continue vers une charge continue. Les structures classiques avec cette fonctionnalité sont les hacheurs

- **AC/AC**. Ces structures permettent de transférer de l'énergie d'une source alternative vers une charge alternative. Les structures réalisant ce transfert sont les cycloconvertisseurs ou pour avoir une flexibilité et des performances plus grandes, un assemblage de deux structures : une AC/DC et une DC/AC.

Dans ce document, nous appelons topologie l'ensemble des liens électriques d'un convertisseur statique. La structure de celui-ci est une topologie, avec en plus une spécification pour tous les composants. Par exemple, ces deux notions sont illustrées pour un onduleur sur la figure 1.2.

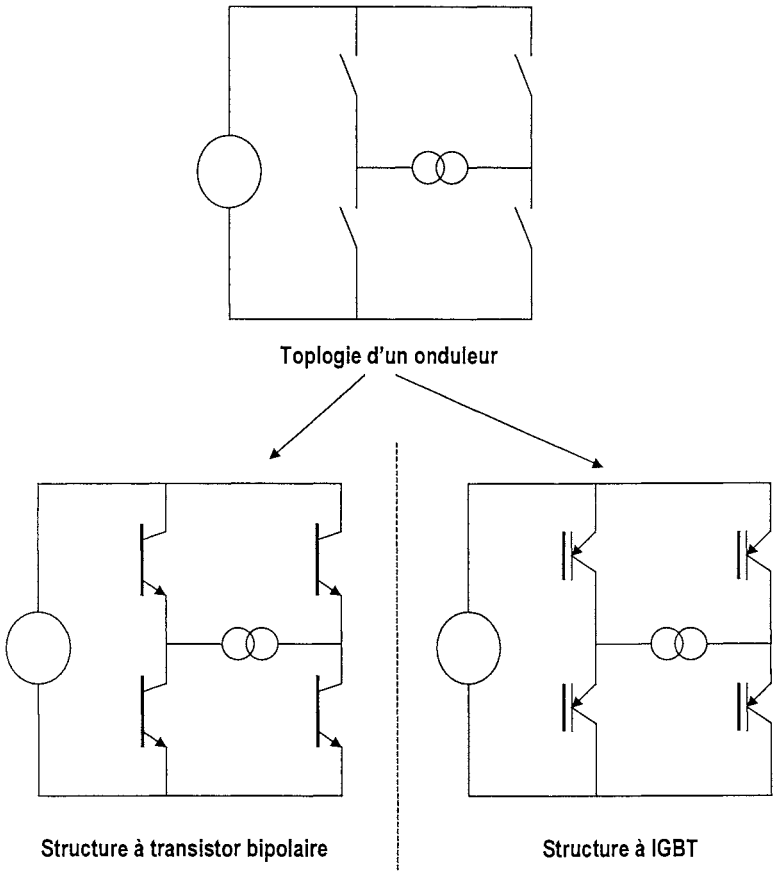



**Figure 1.2 : Les notions de structure et de topologie**

## I.1.2. Applications

Les convertisseurs statiques permettent d'obtenir des alimentations délivrant des courants ou des tensions à des fréquences variables avec des rendements qui sont souvent élevés. La grande souplesse d'utilisation des convertisseurs statiques fait que leurs applications sont nombreuses. Les limitations en termes de puissance proviennent essentiellement des composants utilisés pour les interrupteurs. Ceux-ci ne sont pas parfaitement contrôlables et leurs capacités en termes de courants commutés et de tensions supportées sont limitées.

Actuellement, la technologie industrielle des semi-conducteurs propose les sept types principaux d'interrupteurs électroniques présentés dans le tableau 1.1 [MOH89].

| Technologie du Composant | Gamme de puissance | Vitesse de commutation | Caractérise idéale de fonctionnement. |
|---|---|---|---|
| **Diode** | Elevée | Elevée | Segment 2 : I> 0 pour V =0<br>Segment 3 : I = 0 pour V<0<br>***Non contrôlable*** |
| **Thyristor** | Elevée | Basse | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>Segment 3 : I = 0 pour V<0<br>***Contrôle du passage du segment 1 au segment 2*** |
| **Thyristor GTO**<br>Energie de commande élevée | Elevée | Basse | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>Segment 3 : I = 0 pour V<0<br>***Contrôle du passage du segment 1 au segment 2***<br>***Contrôle du passage du segment 2 au segment 1*** |
| **Transistor Bipolaire**<br>Energie de commande élevée | Moyenne | Moyenne | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>***Contrôle du passage du segment 1 au segment 2***<br>***Contrôle du passage du segment 2 au segment 1*** |
| **Transistor MOS**<br>Energie de commande faible | Basse | Elevée | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>Segment 4 : I < 0 pour V=0<br>***Contrôle du passage du segment 1 au segment 2***<br>***Contrôle du passage du segment 2 au segment 1*** |
| **IGBT**<br>Energie de commande faible | Moyenne | Moyenne | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>Segment 4 : I < 0 pour V=0<br>***Contrôle du passage du segment 1 au segment 2***<br>***Contrôle du passage du segment 2 au segment 1*** |
| **MCT**<br>Energie de commande faible | Moyenne | Moyenne | Segment 1 : I = 0 pour V >0<br>Segment 2 : I > 0 pour V=0<br>Segment 3 : I = 0 pour V<0<br>***Contrôle du passage du segment 1 au segment 2***<br>***Contrôle du passage du segment 2 au segment 1*** |

**Tableau 1.1 : Une classification des composants de puissance.**

En observant ce tableau, il est possible de remarquer qu'il n'existe pas de composant parfaitement contrôlable ou présentant des performances universelles (haute puissance, haute fréquence de commutation). C'est pourquoi le concepteur devra composer ces différents éléments pour obtenir la fonction qu'il désire avec les bonnes performances.
La tendance actuelle est d'augmenter la fréquence de commutation pour réduire la taille (notamment en réduisant la taille des composants magnétiques), mais le concepteur se trouve confronté au problème suivant :

***plus la fréquence de commutation est élevée, plus le coût des composants additionnels à ajouter au convertisseur (les éléments de filtrage en particulier) est réduit, mais aux dépens d'une conception est plus délicate.***

Pour résoudre ces problèmes, il peut être amené à ajouter des nouvelles caractéristiques à son convertisseur. Une modélisation fonctionnelle est bien adaptée dans ce cas parce qu'elle

permet d'avoir une représentation générique du fonctionnement du convertisseur (détachée en grande partie d'une technologie de réalisation). Dans cette tâche, le concepteur doit choisir une technologie de réalisation et ajouter des fonctions pour supprimer des problèmes de fonctionnement.

### I.1.3. Les points délicats du fonctionnement

Comme nous l'avons vu dans le paragraphe précédent, les convertisseurs statiques sont constitués de composants en commutations (entre les différents segments). Ceux-ci offrent des rendements élevés mais des contraintes nouvelles apparaissent en raison de la continuité de l'énergie.

- **Les commutations provoquent des échauffements.** En effet, une commutation ne se produit pas de façon instantanée (un cas plus réaliste est illustré sur la figure 1.3). Il en résulte que les commutations sont génératrices de pertes par effet Joule. Pour réduire les échauffements, il est possible d'adjoindre un circuit d'aide à la commutation ("snubber") ou de s'arranger pour provoquer une commutation à courant nul (ZCS) ou sous une tension nulle (ZVS) [FER94] [MOH89].

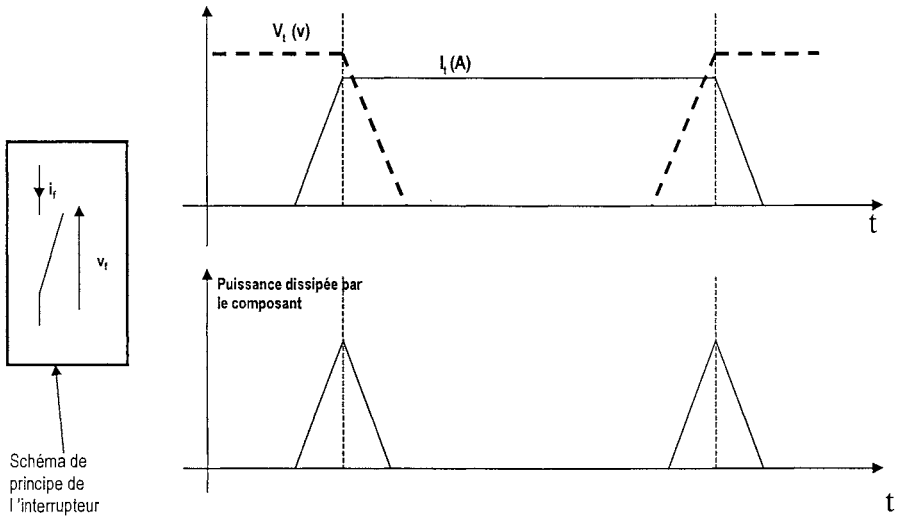



**Figure 1.3 : Une commutation réelle**

- **Les circuits électriques ne sont pas parfaits.** Les connexions électriques d'un convertisseur statique comportent toujours des inductances parasites. Celles-ci vont intervenir pour ralentir les montées de courant et provoquer des empiètements indésirables dans le circuit. Il faudra prendre garde que ces mauvaises commutations ne provoquent pas de contraintes destructives sur les composants. Ainsi, nous pouvons dire que la qualité de fabrication devient un critère très d'important si la fréquence de commutation augmente.

- **Les dispositifs construits ne produisent pas naturellement des courants sinusoïdaux**. Les nombreuses harmoniques engendrées peuvent provoquer des

échauffements des lignes électriques ou la création de parasites à hautes fréquences gênants pour la radio, les télécommunications, le fonctionnement d'autres dispositifs ou nocifs pour le convertisseur lui-même. Le concepteur sera donc obligé d'ajouter des composants pour réduire ces parasites dûs aux convertisseurs [RAB95].

## I.2. Les problèmes liés à la conception de convertisseurs statiques

La phase de définition de structure est une opération délicate. L'expert est obligé de se poser un certain nombre de questions pour résoudre les problèmes liés à la définition de la structure en évitant les problèmes de fonctionnement précédemment cités.

- **Quelle combinaison d'interrupteurs utiliser ?** La définition de la topologie, donc des interrupteurs utilisés, est une phase importante. Les méthodes les plus simples sont de procéder par assemblage de macro-fonctions (redresseur, hacheur, onduleur) ou de cellules de commutation. L'expert peut par exemple décider de synthétiser un convertisseur AC/AC avec une tension et une fréquence variables par un assemblage d'un redresseur puis d'un hacheur et d'un onduleur à fréquence variable. C'est une solution parmi d'autres. Cependant, il est difficile, sans comparaison, de dire qu'elle est la meilleure.

- **Quelles sont les combinaisons d'état d'interrupteurs utilisables et interdites ?** Sur une topologie donnée avec N interrupteurs, il existe en théorie $2^N$ combinaisons possibles. Une analyse du circuit doit être faite pour déterminer les combinaisons interdites et détecter où sont les incohérences.

- **Quels sont les enchaînements possibles pour ces configurations ?** Il faut définir une séquence de fonctionnement des interrupteurs pour obtenir effectivement les bonnes formes d'onde en sortie et donc le bon type de transfert d'énergie. Dans cette étape, la simulation peut être une première approche pour déterminer le mode de fonctionnement de principe, mais elle ne permet pas de prendre facilement en compte les imperfections. En effet, cette opération demande de connaître avec précision les caractéristiques des composants, avec des modèles suffisamment fins et des temps de simulation très longs. Souvent, pour gagner du temps, il est nécessaire faire appel à des règles d'expertise et de réserver la simulation fine pour l'étape de vérification des hypothèses utilisées pour définir la structure.

- **Quelles sont les contraintes technologiques qui ont un impact sur le fonctionnement de la structure choisie ?** La nature technologique des composants influe sur le fonctionnement de la structure globale. Par exemple, les circuits résonnants (pour les modes ZVS) ou les circuits d'aide à la commutation peuvent entraîner des pics de courant ou des surtensions qui sont des contraintes supplémentaires sur les composants du montage.

## I.3. Quelques méthodes de conception en électronique de puissance

Il existe plusieurs méthodes pour formaliser l'obtention de structures de convertisseurs. Parmi celles-ci nous pouvons citer les méthodes basées sur l'utilisation des cellules de commutation ou des convertisseurs matriciels.

### I.3.1. Utilisation des cellules de commutation

#### *I.3.1.1. Présentation*

La méthode la plus répandue pour la conception des convertisseurs est de les modéliser comme un ensemble de structures de bases appelées cellules de commutation. Ces dernières constituent un ensemble d'interrupteurs fonctionnant de façon complémentaire. Une cellule permet de relier une ou plusieurs sources de tension à une source de courant en respectant les règles de fonctionnement des convertisseurs statiques [CHE89] :

- **une source de tension n'est jamais court-circuitée,**
- **deux sources de tensions ne sont jamais directement connectées,**
- **une source de courant n'est jamais ouverte.**

Cette approche définit alors deux types principaux de cellules de commutation qui garantissent ces règles [FOC].

- **La cellule de commutation élémentaire**. Celle-ci relie une source de tension continue à une source de courant et comporte deux interrupteurs ayant des fonctionnements complémentaires. Dans ce type de cellule (représentée sur la figure 1.4), il y a un interrupteur qui contrôle le transfert d'énergie et l'autre qui garantit la continuité du courant dans la source de courant.

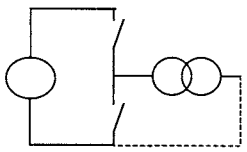


**Figure 1.4 : Schéma d'une cellule de commutation élémentaire**

- **La cellule Polyphasée**. Cette cellule comporte n interrupteurs groupés suivant le schéma de la figure 1.5 . Ces n interrupteurs fonctionnent donc de façon cyclique et sont de nature identique. On a donc deux possibilités pour les interrupteurs :

soit, ils sont commandés à l'amorçage et ont un blocage spontané,
soit, ils sont commandés au blocage et ont un amorçage spontané.

Dans ce cas, tous les interrupteurs ont la même fonctionnalité, la cellule garantit alors la circulation du courant (la fermeture des mailles) et définit un décalage temporel dans le fonctionnement des interrupteurs.

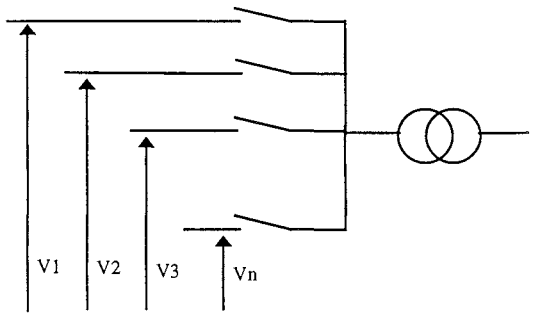



**Figure 1.5 : Schéma d'une cellule de commutation à plusieurs sources**

A partir de ces cellules de base il est possible de définir des cellules muti-niveaux [DAV97, DAV97A]. Celles-ci sont constituées de plusieurs cellules élémentaires capables de délivrer plusieurs niveaux de tension afin d'obtenir une modulation plus fine du signal de sortie.

### *I.3.1.2. Utilisation et avantages*

A partir de ces ensembles, il est possible de construire les structures classiques des convertisseurs statiques (hacheur, redresseur, onduleur) avec une vision idéale de leur fonctionnement. L'approche cellule de commutation est une première méthode qui permet de gérer la détermination de la structure, notamment grâce aux avantages suivants.

- **Les problèmes de cohérence sont gérés de façon implicite par les cellules.** La notion de cellule gère de façon implicite les problèmes de cohérence énergétique qui peuvent s'exprimer en terme de causalité avec une approche de type graphe de lien.

- **Le nombre de configurations possibles est directement réduit.** En effet, si un convertisseur est décomposé en cellules, alors le mode de fonctionnement est une combinaison des fonctionnements des différentes cellules.

### *I.3.1.3. Exemple de système utilisant cette approche*

A partir de la notion de cellule, il est possible de mettre en place une approche orientée système expert. Par exemple, dans sa thèse, D. Fezzani [FEZ96] propose un système de conception des ASI (Alimentation Sans Interruption). Dans cette approche, la topologie de départ est figée (voir figure 1.6) et il propose un système expert modulaire en 3 étapes.

- **Etape 1, Synthèse de convertisseurs.** A partir de la définition de la topologie et des formes d'ondes des sources et des charges, un module d'expertise est capable de déterminer la nature réelle (que nous qualifierons de structurelle) des interrupteurs. A partir des formes d'onde en courant et tension, il caractérise d'abord la technologie

des interrupteurs, puis les contraintes de fonctionnement, et enfin il en déduit les commandes.

- **Etape 2, choix et dimensionnement des circuits d'aide à la commutation**. En fonction des interrupteurs choisis et de leur commande, il est souvent nécessaire de leur adjoindre un circuit dit d'aide à la commutation, qui permet de faciliter les commutations, de canaliser l'énergie et de réduire les pertes pendant les changements d'états des semi-conducteurs.

- **Etape 3, les choix et dimensionnement des dispositifs de filtrage**. La dernière étape est le dimensionnement des dispositifs de filtrage pour réduire les perturbations produites par le convertisseur.

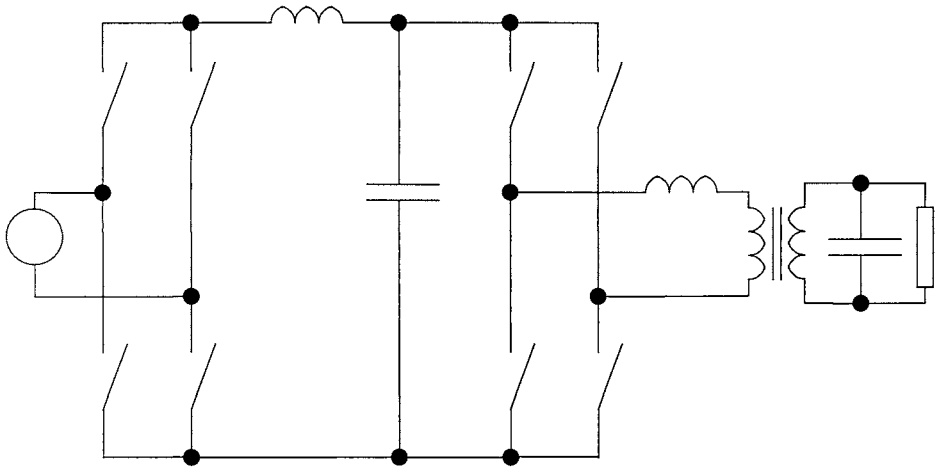


**Figure 1.6 : La topologie de l'alimentation à concevoir [FEZ 96A]**

Le système proposé cherche à reproduire le raisonnement des experts. Pour cela, il propose une décomposition en modules d'expertise caractérisés par un but, un cahier des charges, des données propres et des explications. Le processus de conception est géré de façon automatique par un algorithme qui enchaîne l'exécution des différents modules.
Ce système présente l'avantage d'effectuer une conception complète d'un convertisseur. Cependant, il ne s'applique que dans un cas particulier parce que le raisonnement de conception est figé, ainsi que la topologie. Cette approche ne permet donc pas d'obtenir un système capable d'innover en terme de topologie.
En résumé, c'est une approche structurelle de la définition du convertisseur et suite à cet exemple une question se pose : **Comment définir un outil de conception générique pour les structures de convertisseurs statiques ?**

### *I.3.1.4. Limites par rapport à notre vision du problème de conception des structures*

L'approche cellule de commutation est séduisante parce qu'elle permet de réduire de façon efficace les possibilités des combinaisons possibles des structures. Elle est cependant limitée par rapport à notre vision du problème de conception pour les raisons suivantes.

- **Les cellules ont été définies pour un fonctionnement idéal.** La modélisation des éléments parasites est problématique. Pour les prendre en compte, il est nécessaire de définir autant de cellules que de cas possibles. Par exemple, les empiètements peuvent être pris en compte en ajoutant des inductances, mais le mode de fonctionnement s'en retrouve modifié.

- **Les cellules peuvent se décomposer en sous-fonctions.** La notion cellules est très utile pour concevoir des structures de convertisseur « à la main » mais elle s'avère peu performante pour un processus automatique. En effet, il faudrait définir une cellule pour chaque cas possible, le nombre de combinaisons peut être grand (notamment en prenant en compte les isolations magnétiques, les imperfections de fonctionnement ou les éléments parasites) et personne ne peut garantir que nous aurons toutes les solutions possibles (leur nombre peut évoluer avec la technologie). Il nous paraît alors plus intéressant d'utiliser un niveau de granularité plus fin pour les fonctions de base des convertisseurs, les cellules étant retrouvées automatiquement par un processus de vérification de cohérence. Avec cette méthode, nous pouvons très facilement intégrer des nouvelles technologies sur les semi-conducteurs ou même des composants magnétiques tels que des transformateurs.

## I.3.2. L'approche matricielle [FRA96]

### *I.3.2.1. Présentation*

L'approche matricielle de convertisseurs, développée notamment au L2EP(Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille), modélise les interrupteurs d'un convertisseur statique comme des fonctions de commutation. En effet, ils peuvent être considérés comme des dipôles pouvant établir une connexion binaire (ouvert - fermé) dans un circuit électrique où ils sont insérés. En les rendant idéaux, ils peuvent se modéliser comme des fonctions de connexion (notées F) prenant deux valeurs telles que :

l'interrupteur est considéré parfaitement contrôlable (à l'ouverture et à la fermeture) et bidirectionnel en tension et courant avec,

F = 0 quand l'interrupteur est fermé
F = 1 quand l'interrupteur est ouvert

Cette fonction permet alors de formuler mathématiquement l'effet de l'interrupteur sur le convertisseur avec les relations suivantes :

$$i(t) = F(t) \cdot is(t) \qquad (1.1)$$

$$u(t) = (1 - F(t)) \cdot us(t) \qquad (1.2)$$

avec i(t), u(t) les courants et tensions modulés et is(t) us(t) les courants et tensions commutés des sources

En utilisant cette méthode, le convertisseur (voir l'exemple de la figure 1.7 ) peut se modéliser comme une matrice d'interrupteurs (ou de fonctions de commutation) reliant C

sources de courant vers L sources de tension. A partir de cette matrice, la méthodologie définit un certain nombre de traitements mathématiques qui permettent de mettre en évidence les cellules de commutation, et de déterminer par un raisonnement formel la partie commande et régulation des convertisseurs.

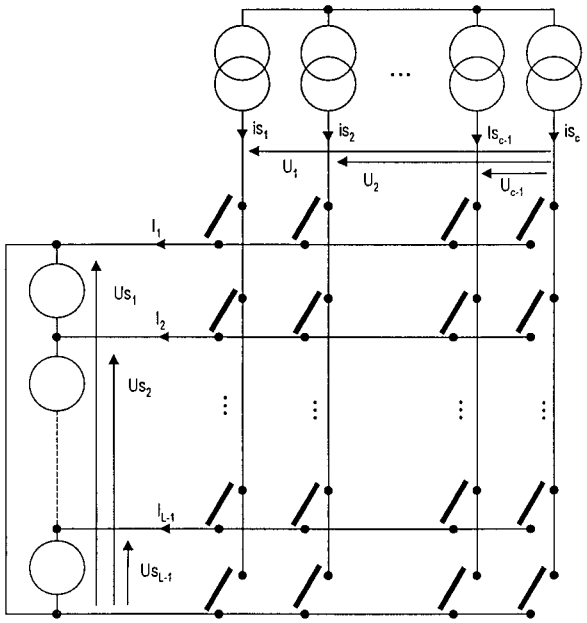



**Figure 1.7 : Un convertisseur matriciel de principe**

### *I.3.2.2. Utilisation et avantages*

Cette approche est utilisée principalement pour la définition de la commande, elle permet de prendre en compte mathématiquement le fonctionnement discret d'un convertisseur. Elle a été notamment utilisée pour la conception d'un onduleur de tension [CAM96] avec une commande en modulation de largeur d'impulsion (MLI) ou la définition des interrupteurs d'une structure [FRA96].
Cette méthode est plus mathématique et systématique que la précédente mais présente des limites vis à vis de notre vision de la conception.

### *I.3.2.3. Limites par rapport à notre vision du problème de conception.*

Cette méthode fige la topologie du convertisseur de façon matricielle donc se focalise sur la définition de la commande et présente les limites suivantes pour la définition de la structure.

- **Les convertisseurs modélisés sont de type direct**. La topologie matricielle ne permet pas de prendre en compte les éléments de stockage d'énergie qui présentent une causalité variable et les éléments avec un comportement plus complexe comme les transformateurs.

- **Les interrupteurs sont idéaux.** L'approche ne prend pas en compte les imperfections des interrupteurs réels, en réalité les semiconducteurs ne sont pas

complètement commandables et ils n'acceptent pas tous les sens et les niveaux de courant.

- **Le fonctionnement est idéal.** Comme pour les cellules de commutation, les modes de fonctionnement transitoires -par exemple les empiètements ou la gestion des temps de garde- ne sont pas pris en compte.

### I.3.3. Les approches systèmes experts pour les entraînements électromécaniques

Le choix des structures de convertisseurs statiques est souvent la phase problématique de la conception d'entraînements électromécaniques. L'approche la plus courante est l'utilisation d'une base de donnée pilotée par un système expert permettant le choix de structures en appliquant une série de règles heuristiques. Un environnement de ce type est alors une aide au calcul des paramètres de la commande. Parmi les applications, nous pouvons citer des travaux sur les machines alternatives alimentées par onduleur [SAA93], [CHE92] ou l'alimentation des gros entraînements [CHH95], [GER93].
L'inconvénient majeur de ces approches est qu'elles figent l'ensemble des solutions possibles dans une base donnée. Celle-ci est alors construite avec des critères structurels (donc liés à une technologie et une solution technique) et ne permet pas de réaliser une certaine «innovation». La mise en œuvre de ces outils peut être lourde et la maintenance problématique. Une approche fonctionnelle permettra d'acquérir une certaine indépendance par rapport aux solutions techniques, et facilitera la maintenance des outils ainsi construits.

## II. L'analyse fonctionnelle dans le processus de conception en génie électrique

### II.1. Description générale du processus de conception

Le processus de conception d'un convertisseur statique peut être décrit par l'organigramme de la figure 1.8. Pendant celui-ci, l'analyse fonctionnelle est utile sur plusieurs étapes :

- **L'étape de définition du cahier des charges.** L'analyse fonctionnelle permet de définir les fonctionnalités à remplir par le projet ou le produit à concevoir. Cette technique est très utilisée actuellement dans des domaines très divers comme l'informatique pour la spécification de logiciels [OUS97] ou d'une façon plus générale pour la spécification de projets [CER96].

- **L'étape de définition de la structure.** Ici, L'analyse fonctionnelle permet d'obtenir un niveau d'abstraction suffisant pour s'affranchir de la technologie et donc réaliser une exploration plus efficace et structurée des différentes solutions. Une même représentation fonctionnelle peut conduire à plusieurs solutions en utilisant des technologies différentes. De plus, cette indépendance vis à vis de la technologie permet de prendre en compte facilement des nouvelles évolutions techniques.

- **L'étape de modification de la structure.** Si au cours de la conception il est nécessaire d'effectuer une modification de la structure, l'expert a plusieurs possibilités :
  - soit, il choisit de remplir une fonction par un autre choix technologique, par exemple une fonction interrupteur peut être remplie par un transistor bipolaire à fréquence moyenne ou un transistor MOS pour les fréquences plus élevées,
  - soit, il choisit d'ajouter des fonctionnalités à son système, comme par exemple des fonctions de filtrage.

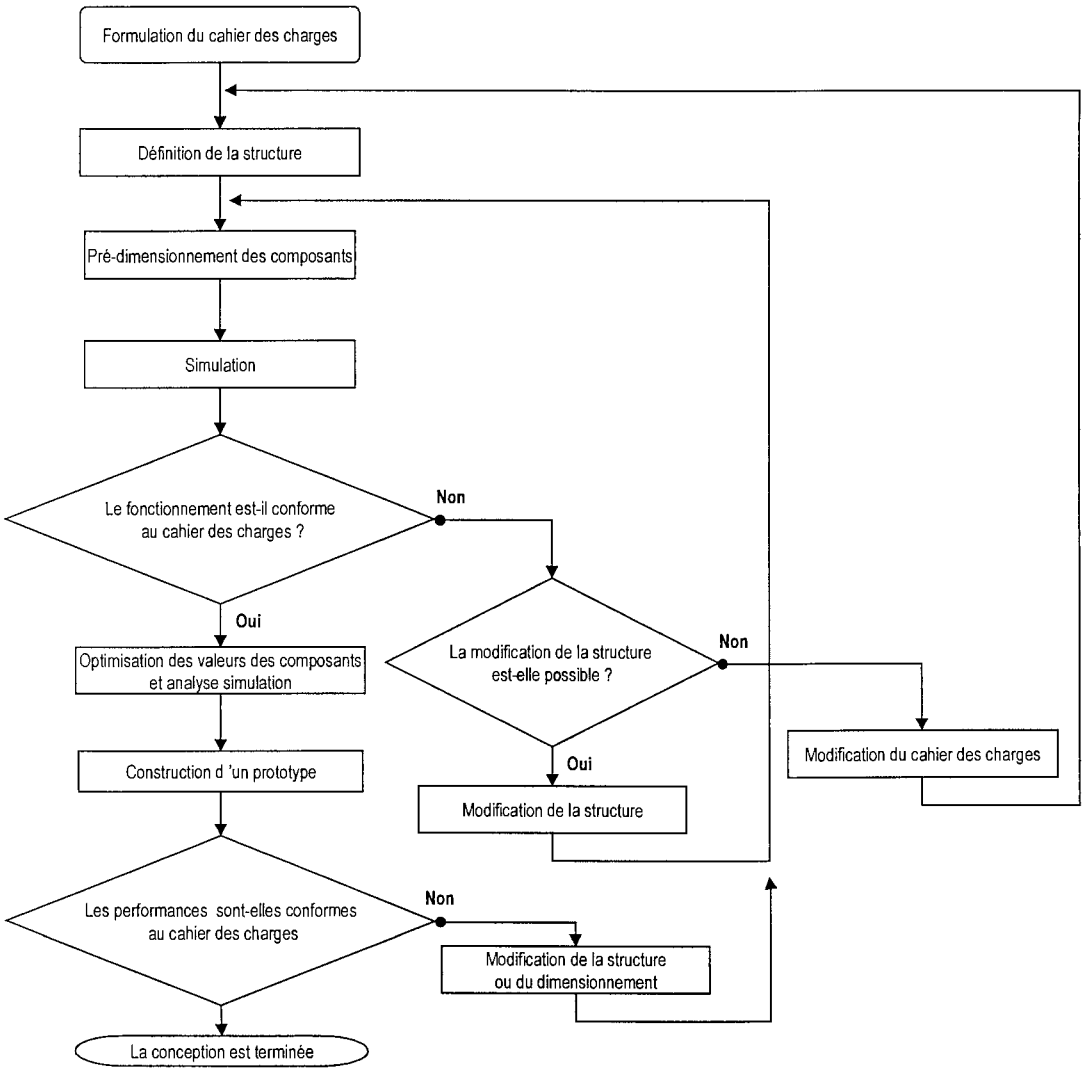



**Figure 1.8 : Un processus de conception d'un convertisseur statique**

D'une façon générale, il est possible de définir une trame de la méthodologie fonctionnelle dans le cas des convertisseurs statiques.

### II.1.1. La définition du cahier des charges

La définition du cahier des charges est l'étape fondamentale du processus de conception. Elle permet au concepteur du produit de définir les besoins de l'utilisateur final qui seront traduits en besoins et contraintes sur le produit à concevoir.
Un exemple de cahier des charges fonctionnel dans notre cas de l'électronique de puissance peut être :

le produit doit fournir une tension continue de 48 V à partir du réseau de secteur ;
les contraintes peuvent être :

- le convertisseur doit être le plus léger possible,
- le taux d'harmoniques rejetées sur le secteur doit être réduit,
- le bruit émis par le convertisseur doit être réduit,
- le rendement doit être élevé,
- l'encombrement doit être réduit.

Ce cahier des charges peut être défini a priori, mais peut aussi être remis en cause dans le processus de conception en cas d'échec dans la recherche de structure ou de non-respect de tous les critères. Dans ces cas il est nécessaire de modifier ou de relâcher des contraintes ou des besoins.

### II.1.2. La définition ou la modification de la structure

La phase de définition de la structure est une phase d'assemblage des éléments requis pour réaliser le produit permettant de satisfaire le cahier des charges. La définition fonctionnelle d'un convertisseur peut se réaliser en plusieurs étapes.

**1. Définition des transferts (transformations) d'énergie à réaliser**. Cette partie consiste à spécifier les fonctionnalités à remplir par le convertisseur. Cette étape est liée à la définition du cahier des charges, donc des fonctions sources et charges du système.

**2. Définition des chemins d'énergie à réaliser**. Il est nécessaire d'établir des liaisons énergétiques entre les fonctions définies à l'étape précédente pour obtenir un processus de conversion d'énergie.

**3. Spécification de tous les éléments requis en fonction d'une technologie**. Les fonctions ne sont qu'une représentation abstraite d'un élément réel, c'est le choix d'une technologie qui déterminera quel est le composant réel ou l'ensemble de composants réels pouvant les remplir.

**4. Définition de la commande**. La dernière étape est la spécification de la commande du système. Cette étape permet de définir l'enchaînement des états des fonctions de contrôle pour piloter la quantité d'énergie transférée.

## II.2. Quelques méthodes de conception ou d'analyse avec une approche fonctionnelle

Dans ce paragraphe nous effectuons une étude de quelques méthodes de modélisation, d'analyse et de conception fonctionnelle. Celles-ci portent sur des problèmes voisins du nôtre.

### II.2.1. Une modélisation fonctionnelle d'un dispositif de l'électromagnétisme

#### *II.2.1.1. Présentation*

Cette étude a été réalisée au sein du LEG dans le cadre d'un projet de DEA [BRU94]. Elle avait pour objectif d'essayer de définir un premier modèle fonctionnel d'un contacteur électromagnétique.
Ce travail présente plusieurs phases.

- **La définition d'une décomposition structurelle d'un contacteur [GEN91].** Ce dernier est décomposé en petits ensembles élémentaires que l'on peut définir comme une structure. Puis par assemblage, il est possible de construire un arbre de composition structurelle. On en conclut que cette représentation est très adaptée pour un système expert de pré-dimensionnement, mais ne permet pas l'évolution de la structure de l'ensemble, notamment du nombre de composants et de la forme géométrique. La figure 1.9 présente la décomposition structurelle proposée par ce projet.

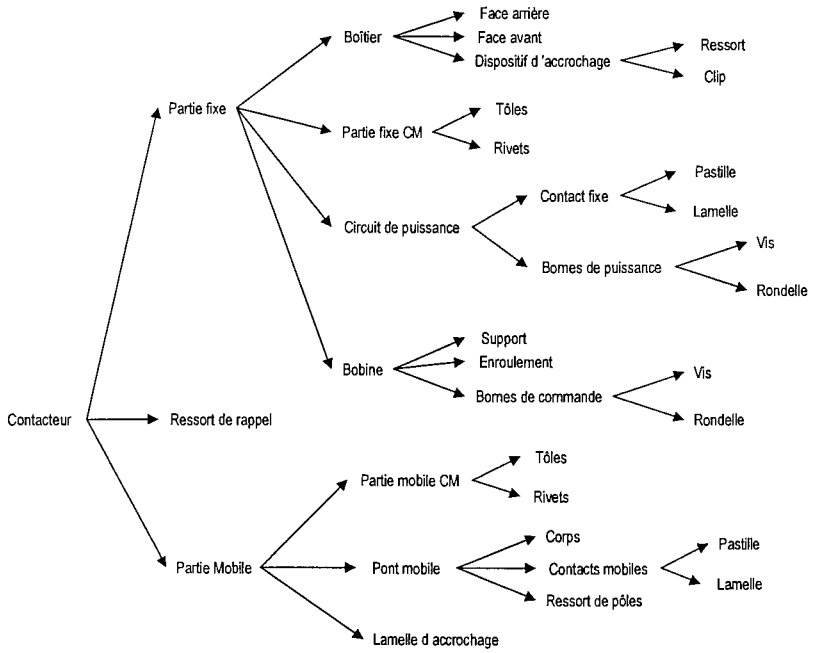



**Figure 1.9 : Une décomposition structurelle d'un contacteur [BRU94]**

- **La définition d'une décomposition fonctionnelle du même contacteur** (cf. figure 1.10). A partir de la description précédente, il est possible de définir un certain nombre de fonctionnalités à remplir (et par conséquent des fonctions) pour le même

contacteur. Ces fonctions sont liées entre elles par des liens de composition à partir de la fonction de base d'un contacteur : « un interrupteur commandé ». Les fonctions feuilles (aussi appelées pseudo-fonctions) de l'arbre de décomposition doivent alors être remplies par les différentes structures. Pour que cette approche puisse être innovante, la décomposition ne doit pas être trop précise. La question délicate à résoudre est : ***« où finissent les fonctions et où commencent les structures ? »***.

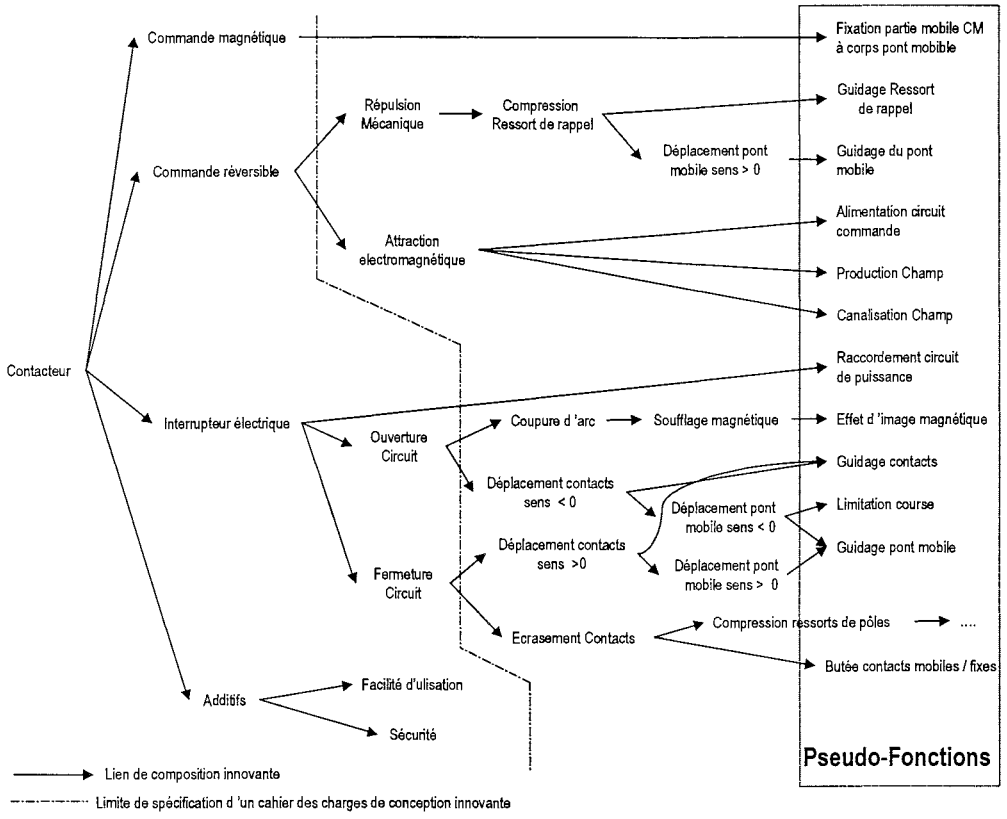



**Figure 1.10 : La décomposition fonctionnelle proposée pour le contacteur**

- **La définition des liens entre pseudo-fonctions et structures.** La dernière étape est la définition du lien entre une fonction et une ou plusieurs structures. Celui-ci permet de relier les deux représentations précédemment citées. C'est la formalisation de ce passage qui conduira effectivement à une certaine innovation.

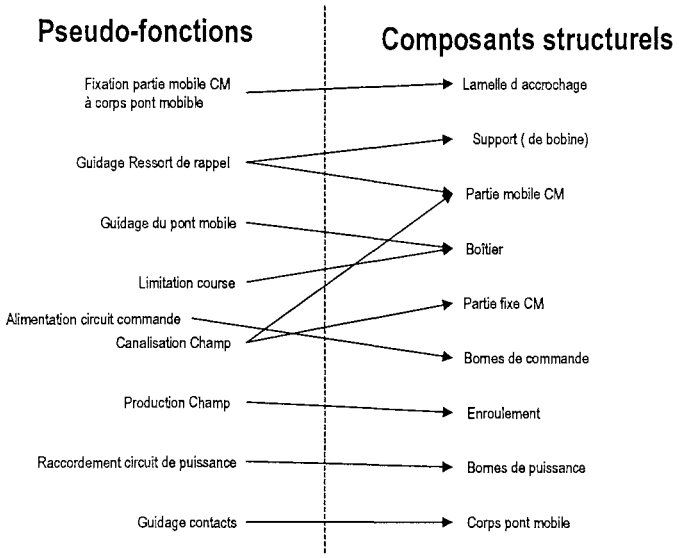



**Figure 1.11 : Exemple de lien fonction-structure**

- **La représentation des deux graphes dans un système informatique orienté objet** de modélisation de la connaissance : SHOOD [NGU92B]. Ceci devra permettre, à long terme, la manipulation des données en vue de la conception.

### *II.2.1.2. Evaluation par rapport à notre problème*

Dans cette approche, le dispositif est modélisé fonctionnellement à partir d'objets avec une sémantique forte. Les fonctionnalités sont complètement définies par l'utilisateur à partir d'une représentation qu'il doit se faire du fonctionnement du dispositif. Il en résulte que la modélisation fonctionnelle ne peut pas être unique. Cette représentation multiple rend la formulation générique du problème de conception difficile voire impossible.
La seconde limitation par rapport à nos besoins est que cette approche ne prend pas en compte de façon explicite l'aspect énergétique du fonctionnement du dispositif. Il est donc quasiment impossible de vérifier la cohérence du système en cours de conception sur des critères énergétiques.
Enfin, le système étudié est statique par rapport au processus de transfert d'énergie. Il ne représente qu'un seul état énergétique contrairement aux convertisseurs statiques qui peuvent en présenter plusieurs (à cause des changements d'état des interrupteurs).
Mais, de cette étude, il ressort qu'il est nécessaire de bien formaliser la notion de fonction afin de réduire au maximum l'utilisation de connaissances de type heuristique dans la définition de celle-ci.

051 095898 1

## II.2.2. Méthode SADT

### *II.2.2.1. Présentation*

La méthode SADT [LIS91] est une méthode structurée pour analyser et spécifier le fonctionnement d'un système. Le modèle SADT permet de formaliser l'état du système étudié et d'en faire ressortir les points critiques en décrivant les fonctions et leurs interactions. Il est

surtout utilisé pour spécifier les systèmes d'information mais il permet aussi la modélisation d'un système physique.
La modélisation SADT propose deux aspects de modélisation pour un système.

- **La décomposition des activités (ou fonctions).** Le diagramme obtenu est appelé actigramme. Il met l'accent sur les activités du système représentées par des boîtes et reliées par des flèches pour modéliser les données échangées (des objets, des informations). Le formalisme graphique correspondant est décrit sur la figure 1.12. Notons qu'il existe quatre types de données :

  Les entrées : les données transformées par l'activité
  Les sorties : les données créées par l'activité
  Les contrôles : les données qui modifient le processus de conversion des entrées vers les sorties.
  Les mécanismes : les moyens nécessaires pour la mise en œuvre de l'activité.

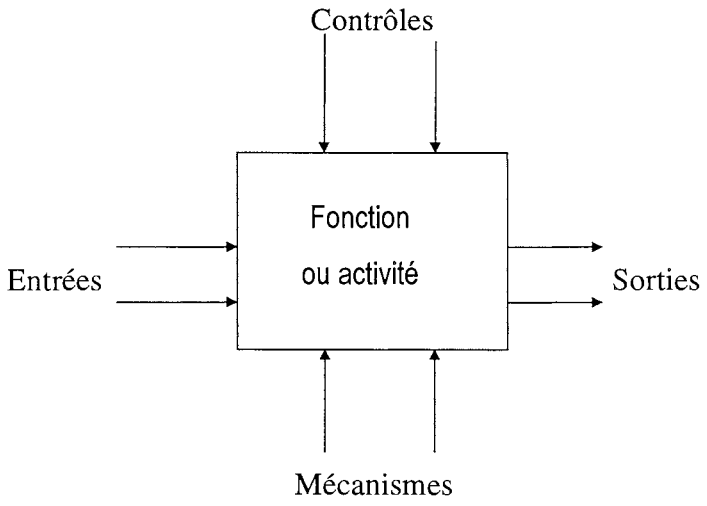



**Figure 1.12 :La symbolisation des actigrammes**

- **La décomposition des données.** Un datagramme est le diagramme dual du précédent. Il représente les données par des boîtes noires où les activités sont représentées par des flèches pour matérialiser les liens entres fonctions. La figure 1.13 présente la sémantique attachée à la représentation graphique des datagrammes.

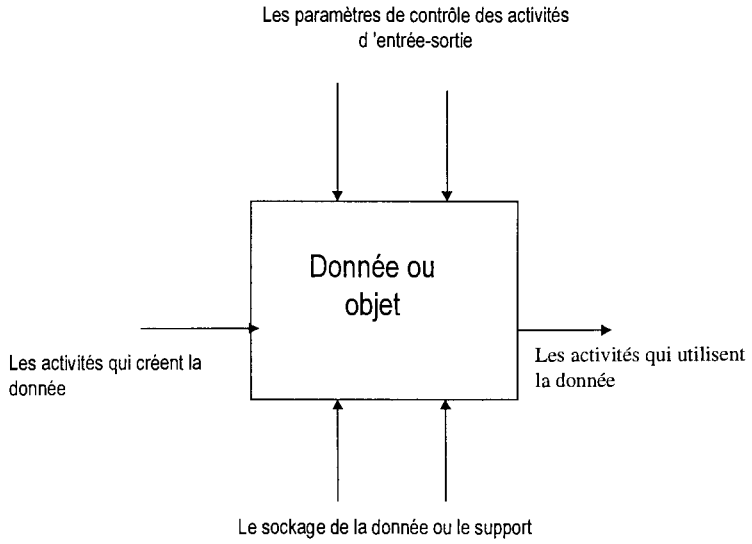



**Figure 1.13 :La symbolisation des datagrammes**

A partir des boîtes de base, SADT propose une méthode pour construire un diagramme complet par décompositions successives des activités ou des données ainsi qu'une grille de lecture des diagrammes.

### *II.2.2.2. Evaluation par rapport notre problème*

La notion SADT est bien une représentation des fonctions d'un système, mais reste essentiellement un outil de description de systèmes. C'est l'utilisateur qui décrit et vérifie sa description. La modélisation d'un système physique est possible mais la méthodologie SADT ne propose aucun outil formel pour le vérifier en terme de cohérence sur les connexions énergétique, le respect de la loi de conservation de l'énergie, etc.

Cependant, ces diagrammes peuvent s'avérer efficaces pour modéliser la partie commande d'un entraînement électromécanique. La fonctionnalité de cette partie est de fabriquer des signaux de commande (qui peuvent être considérés comme des informations) donc de capter ces informations en certains points du système pour le piloter en envoyant des données sur les fonctions de contrôle de celui-ci. La figure 1.14 présente un diagramme SADT pour le pilotage d'un redresseur.

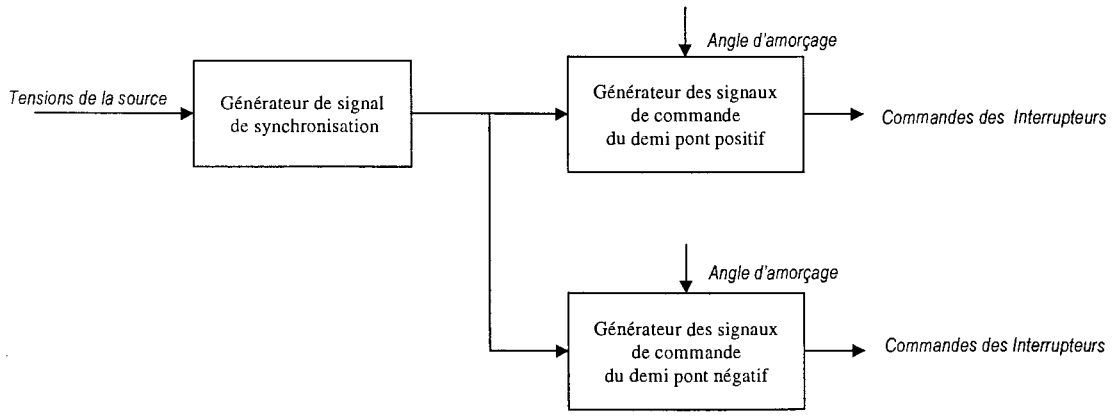



**Figure 1.14 : Le diagramme SADT de la commande d'un redresseur**

## II.2.3. Une approche déjà formalisée en mécanique

### *II.2.3.1. présentation*

Une série de travaux sur la conception fonctionnelle a déjà été réalisée en mécanique par le Laboratoire 3S de Grenoble (Laboratoire des Sols Solides Structures). Ceux-ci présentent une représentation fonctionnelle de dispositifs mécaniques [CON 96] qui est destinée à s'insérer dans une démarche de conception intégrée [BEL94].
La représentation fonctionnelle des mécaniciens est construite à partir des constatations suivantes :

- Une représentation classique (dessin de définition en mécanique) est trop pauvre parce qu'elle ne donne pas assez d'informations sur les fonctionnalités de l'objet à concevoir, sur l'environnement d'utilisation et sur le système de production.

- Dans le cadre d'une conception routinière, quelle que soit l'industrie, un nouveau produit est toujours construit à partir d'une modélisation plus ou moins substantielle de l'ancienne version. Pour mener à bien la conception, et notamment pour permettre les modifications, il est nécessaire d'avoir un bon paramétrage du modèle produit. Suivant leurs ampleurs, il peut être nécessaire de fabriquer un nouveau jeu de paramètres.

Le modèle produit doit définir un objet de façon minimale, c'est à dire qu'il doit se limiter aux surfaces fonctionnelles intervenant dans la réalisation des liaisons mécaniques. Le produit est alors toujours considéré comme un système qui est caractérisé par :

- sa finalité ; les besoins qu'il satisfait,

- sa frontière ; pour définir clairement le monde extérieur,

- ses relations avec l'environnement,

- son organisation ; ses principaux constituants.

La représentation fonctionnelle est un modèle orienté objet où l'expression des fonctions est caractérisée par les transferts d'énergie. Ces deniers sont définis à partir des déplacements de matière (le flux d'énergie est un flux de déplacement de matière), et la définition des frontières des fonctions est caractérisée par les surfaces fonctionnelles. Les relations entre les ces dernières sont modélisées par une méthode formelle appelée S.A.T.T (Surfaces Associées Topologiquement et Technologiquement) permettant d'effectuer des opérations comme des décompositions en série ou en parallèle afin de faire apparaître de nouvelles liaisons ou de nouvelles fonctions.

La méthodologie définit les variantes suivantes pour les classes de fonctions.

- **Les fonctions principales**. Ces fonctions sont utilisées pour modéliser la raison d'être d'un produit et les relations qu'il établit entre deux parties de l'environnement

- **Les fonctions principales participant à l'action**. Ces fonctionnalités permettent de modéliser le fait que la fonction permet l'existence d'un mouvement (donc un flux de matière) entre deux parties d'un environnement.

- **Les fonctions principales d'opposition à l'action**. Dans ce cas, le produit doit empêcher la réalisation d'une action et en particulier il en empêche l'existence d'un mouvement.

- **Les fonctions auxiliaires.** Ces fonctions sont couplées à la fonction principale, pour modéliser le contrôle de celle-ci ou la modification de ses caractéristiques.

- **Les fonctions contraintes**. Elles permettent d'établir des relations entre le produit et un ou plusieurs éléments de son environnement. Elles contraindront les choix technologiques.

Enfin, des objets appelés liaisons sont utilisés pour modéliser les échanges entre les frontières de l'émetteur et du récepteur du flux. Ils réalisent l'interface entre les composants.
L'approche des mécaniciens présente l'intérêt d'une représentation objet reposant sur des considérations énergétiques d'un produit technique à concevoir ; mais elle n'est pas applicable directement dans le cas des convertisseurs statiques pour les raisons que nous évoquons dans le paragraphe suivant.

### *II.2.3.2. Evaluation par rapport à notre problème*

Notre problème de conception des convertisseurs statiques diverge du problème des mécaniciens pour plusieurs raisons. Dans notre cas, il n'y a pas de mouvement mais bien des flux d'énergie ( le courant, ... cf. les graphes de liens §III. ) entre les différentes fonctions.

- **Les parcours des flux ne sont pas fixes.** Contrairement à la mécanique, les convertisseurs statiques comportent des fonctions qui changent d'état. Cela correspondrait à des liaisons mécaniques pouvant être rompues. Ces éléments particuliers ne sont pas pris en compte par l'approche mécanicienne.

- **Les actions des fonctions sur leur monde extérieur évoluent avec le temps.** Il est donc important de vérifier que deux fonctions connectées sont compatibles à tout instant. Si on fait une analogie avec la mécanique, les surfaces fonctionnelles d'entrée/sortie des convertisseurs statiques se déforment au cours du temps.

- **Il important de vérifier que les fonctions acceptent le passage du flux.** Le flux électrique ou magnétique (au sens graphe de lien) peut être considéré comme un fluide qui faut canaliser. Il est important que les fonctions acceptent son passage.

- **Le parcours du flux peut se modéliser par un schéma de type circuit.** Mais contrairement au schéma cinématique, les éléments de celui-ci ne sont pas nécessairement des fonctions du système.

### II.2.4. L'approche fonctionnelle pour la conception en général

L'analyse fonctionnelle apparaît comme une méthodologie intéressante à utiliser pour la définition de produits techniques. Mais, en analysant un certain nombres de travaux [UME97], [CHA93], il apparaît que le concept de fonction est très difficile à définir dans l'absolu. En fait, il demeure mal défini ou très subjectif. Chaque discipline, pour ne pas dire chaque méthode de conception, a une définition particulière de la fonction.
D'une façon générale, la fonction peut être considérée comme une boîte noire remplissant un service (une fonctionnalité) échangeant des flux d'énergie, de matière ou d'information. Il est possible de définir deux types de fonctions : les fonctions transformant le flux et les fonctions sans transformation. La formalisation générique des fonctionnalités étant problématique, chaque outil doit définir les éléments suivants :

- son concept de fonction,
- les méthodes attachées aux fonctions : pour leur composition et décomposition de fonctions, leur description et leur manipulation,
- des méthodes liées à la gestion du processus de conception,
- une mise en œuvre informatique.

En ce qui concerne l'aspect énergétique, il ressort de notre analyse bibliographique que les graphes de liens (couramment appelé Bond-Graph) sont une bonne base de départ pour la modélisation de l'aspect énergétique des systèmes (et leur dynamique).

# III. Une méthode de modélisation pour l'analyse : les graphes de liens

## III.1. Présentation

La théorie des graphes de lien a été établie à l'origine pour une utilisation en automatique et en mécanique [BOR95]. Elle a été généralisée pour être applicable à une multitude de domaines physique. Elle permet d'avoir un fondement solide pour la modélisation d'ensembles avec un formalisme fonctionnel pour les raisons suivantes.

- Elle propose un formalisme générique de représentation des mouvements d'énergie dans les systèmes dynamiques. Il est donc judicieux de définir les fonctions de base et le cahier des charges à partir de considérations énergétiques afin de caractériser les transferts d'énergie.
- Elle propose un formalisme de graphe avec une analyse de cohérence énergétique (en terme de causalité). Une adaptation de cette analyse aux graphes fonctionnels nous permettra de mener une série de vérifications.

Dans les paragraphes suivants, nous présenterons de façon générale les graphes de liens puis nous soulignerons leurs limites par rapport à une modélisation fonctionnelle.

### III.1.1. Effort/flux

Les graphes de liens sont une représentation graphique de l'expression énergétique de la dynamique du système à étudier. Ce dernier est alors décomposé en sous-systèmes et les liens indiquent comment l'énergie est échangée entre ces sous-systèmes. Le modèle obtenu indique si l'énergie est stockée, modifiée ou dégradée à l'intérieur. Pour établir des règles de cohérence et de causalité, les graphes de liens s'appuient sur les lois de la thermodynamique, comme nous le précisons maintenant.

- **Le Premier Principe de la Thermodynamique impose la continuité temporelle de l'énergie.** Ce principe peut se traduire par la conservation de l'énergie. Donc toute l'énergie stockée par le système doit être restituée. Les systèmes d'électronique sont toujours régis par ce principe et il faudra prendre soin de ne pas stocker de l'énergie de façon infinie.

***Remarque : Les sources de flux et d'effort ne respectent pas le premier principe.***

- **Il y a continuité spatiale de l'énergie.** Ainsi, les variations d'énergie sont uniquement dues à des transferts d'énergie.

- **Le Second Principe impose qu'un transfert d'énergie soit proportionnel à un flux de particules (réelles ou fictives).** Le coefficient de proportionnalité est une variable intensive appelée effort. Le flux est mesuré par une variable de flux noté f pour indiquer le nombre de particule par unité de temps. La variable effort est noté e.

La théorie des graphes de liens nous donne la signification physique de chaque variable (effort ou flux) suivant le domaine énergétique. La figure 1.15 propose cette définition dans les cas courants des entraînements électriques.

| Domaine physique | Variable de flux | Variable d'effort |
|---|---|---|
| Electrique | Le courant, i | La tension ou la différence de potentiel, u |
| Thermique | Le flux d'entropie, Fs | La température, T |
| Hydraulique | Le débit volumique, ρv | La pression, p |
| Mécanique en translation | La vitesse, v | La force, f |
| Mécanique en rotation | La vitesse angulaire, ω | Le couple, Γ |
| Magnétique | La dérivée du flux magnétique, λ | La force magnétomotrice, M |

**Figure 1.15 : Quelques variables de flux et d'effort [LAU98]**

### III.1.2. La représentation graphique

Deux composants A et B échangent de l'énergie, alors ils sont représentés par le schéma de la figure 1.16. Le lien est représenté par une demi-flèche dirigée dans le sens d'écoulement d'énergie. Il relie un port de sortie de A vers un port d'entrée de B.

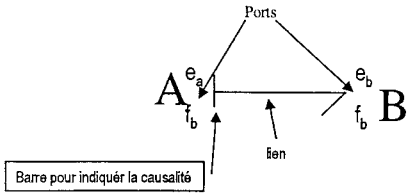



**Figure 1.16 : Représentation graphique d'un lien Bond Graph**

### III.1.3. La causalité

La théorie des graphes de liens associe deux variables à chaque port (effort et flux). Le lien impose que la variable effort du port du composant A ($e_a$) soit toujours identique à la variable effort du composant B($e_b$) et que le flux du composant A ($f_a$) soit toujours identique au flux du composant B ($f_b$).
Les relations définies par le lien imposent que les deux composants ne peuvent pas imposer la même variable. Cette contrainte est définie comme étant la causalité :

***Soit la fonction A impose l'effort (qui est subi par la fonction B) et subit le flux (imposé par la fonction B) et réciproquement.***

Par convention une barre verticale est placée du coté du composant qui impose le flux pour indiquer la causalité d'un lien. Dans l'exemple de la figure 1.16, la fonction A impose le flux dans la liaison. Avec une telle représentation, il est alors possible de mettre en place des processus d'analyse de la causalité.

### III.1.4. Les éléments

Les graphes de liens possèdent donc plusieurs catégories d'éléments normalisés qui permettent de définir la modélisation du système (voir annexe I).

- **Les éléments sources d'énergie**. Ces éléments produisent des mouvements d'énergie dans le système en fournissant le flux (c'est donc une source de flux notée Sf) ou l'effort (pour une source effort notée Se).

- **L'élément dissipatif noté R**. Celui-ci dissipe l'énergie pour la transformer en chaleur. C'est une résistance isotherme.

- **Les éléments jonctions**. Afin de réaliser des réseaux complexes, la théorie des graphes de liens définit des jonctions. Celles-ci sont des éléments à plusieurs ports qui établissent des relations entres les différents liens. Il est possible de distinguer deux types de jonctions.

***La jonction 1 appelée aussi jonction de flux.*** *Celle-ci impose un flux commun sur tous ses ports alors que la relation sur l'effort est telle que le principe de conservation d'énergie est respecté.*

Si une jonction 1 possède k liens alors :

$$f_i = f \quad \forall \quad i \in [1,k] \tag{1.3}$$

et la conservation de l'énergie donne $\sum_{i=1}^{k} e_i = 0$ (1.4)

*Il en résulte qu'un seul élément connecté peut imposer le flux. La figure 1.17 présente un exemple de cette situation. Ici, le cas 2 est écarté parce que deux fonctions imposent le flux et le cas 3 ne peut pas convenir parce que la condition sur l'effort risque de ne pas être respectée.*

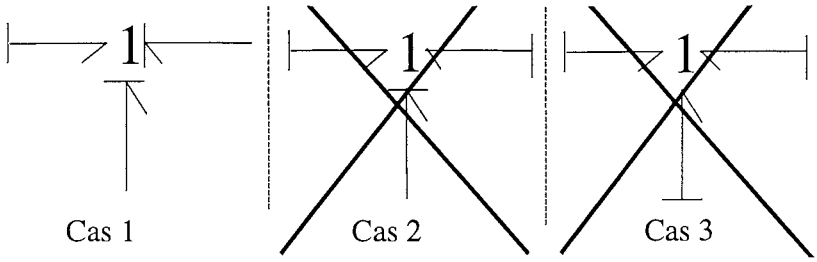



**Figure 1.17 : Les situations de causalité de la jonction 1**

***La jonction 0 aussi appelée jonction d'effort.*** *Celle-ci impose un effort commun sur tous ses ports, c'est la jonction duale de la précédente, un seul élément peut imposer l'effort sur une connexion de ce type. Les équations constitutives sont les suivantes :*

Si une jonction 1 possède k liens alors :

$$e_i = e \quad \forall \quad i \in [1,k] \tag{1.5}$$

et la conservation de l'énergie donne $\sum_{i=1}^{k} f_i = 0$ (1.6)

*La figure 1.18 figure présente un exemple des différents cas de causalité pour une connexion à 3 ports. Le cas 3 est une situation indéterminée.*

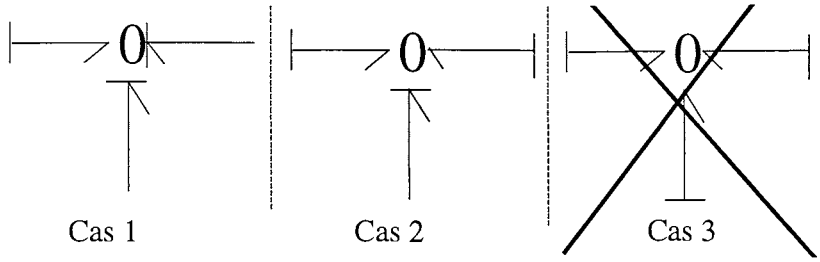



**Figure 1.18 : Les situations de causalité de la jonction 0**

- **Les éléments de stockage**. Il y a deux façons de stoker l'énergie : soit par accumulation du flux (les capacités notées C), soit en accumulant l'effort (les inerties notées I). La causalité de ces éléments est en générale difficile à établir parce qu'elle

n'est pas fixe. Par exemple, une capacité subit l'effort lorsqu'elle accumule de l'énergie et impose un effort sur le reste du système lorsqu'elle restitue de l'énergie.

- **Les éléments de transformation énergie**. Ceux-ci permettent la modélisation des composants en établissant une relation linéaire entre leurs domaines énergétiques. La transformation peut se faire en conservant la causalité (les transformateurs notés TF) ou en modifiant la causalité (les gyrateurs notés GY).

## III.2. Utilisation

Afin d'illustrer l'utilisation des graphes de liens, prenons le circuit résonnant de la figure 1.19.

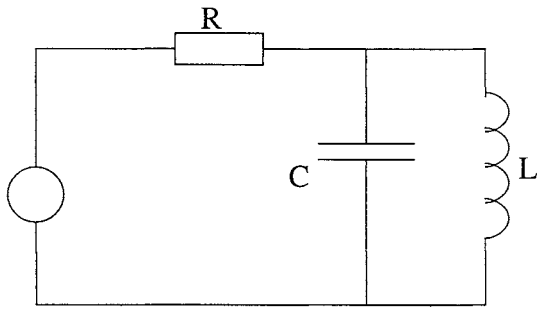



**Figure 1.19 : Un exemple simple de circuit électrique**

Ce circuit présente un élément qui dissipe l'énergie (la résistance R), un élément d'accumulation de l'effort (le condensateur C) et un élément d'accumulation du flux (l'inductance L). La méthode de passage d'un circuit vers un Graphe de lien, qui est largement décrite dans la littérature [DEL91, MAS 92], permet d'obtenir le graphe de la figure 1.20.

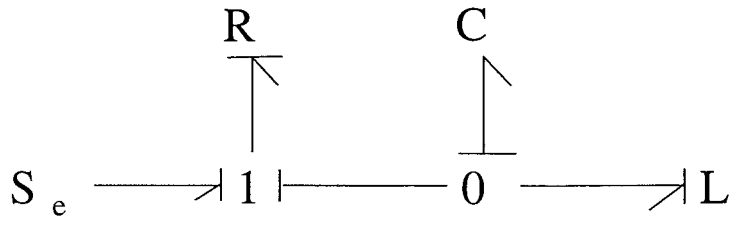



**Figure 1.20 : Le Graphe de lien correspondant au circuit de la figure 1.19**

A partir de ce graphe correct d'un point de vue causal, il est facile d'obtenir un jeu d'équations soluble de façon séquentielle.
Il se pose cependant un problème pour les composants avec plusieurs états comme les interrupteurs. Il n'y a aucun élément standard dans la théorie des graphes de liens qui permettent de prendre en compte ces éléments. Des travaux proposent néanmoins une modélisation des interrupteurs suivant deux approches différentes.

- **L'interrupteur est une résistance variable [DUC92]**. Ce raisonnement est alors similaire à celui utilisé pour la simulation avec méthodes dites à topologie fixe.

- **L'interrupteur est un élément particulier avec une causalité variable [ALL93].** Le circuit est alors modélisé par plusieurs graphes de liens et la modélisation est dite à topologie variable. On peut formaliser l'enchaînement des graphes en cours de simulation à l'aide des réseaux de Pétri.

### III.3. Limites par rapport à nos besoins

La modélisation par les graphes de liens permet d'obtenir une bonne représentation de l'état énergétique du système mais n'est pas suffisante dans une activité de conception, dans laquelle les flux d'énergie dans le système sont mal connus.

- **L'analyse de causalité est imprécise.** Cette notion a été définie pour la mise en équation des systèmes en vue de la simulation. Bien que la causalité puisse avoir une signification physique, une faute de causalité ne signifie pas toujours que le système n'est pas viable [LAU98].

- **La modélisation des interrupteurs n'est pas suffisante.** Les méthodes de modélisation des interrupteurs à l'aide des Bond-Graphs proposent un couplage avec un système logique pour déterminer les états des interrupteurs, le raisonnement se fait alors sur plusieurs graphes de liens différents. La vérification de la causalité a priori (sans connaître le mode fonctionnement) demande à examiner toutes les combinaisons possibles.

- **La vérification a priori des sens des courants est impossible.** Les graphes de liens ont été créés en vue de la simulation en s'appuyant sur les équations (donc quantitatives). Ils ne proposent pas actuellement des règles d'expertise ou d'analyse pour déterminer ou contrôler les sens des courants sans simulation.

- **Le formalisme graphique des graphes de liens suppose que les flux sont bouclés.** La représentation graphique de style graphe de liens ne peut pas être utilisée pour un système en cours de construction puisque cette condition n'est pas nécessairement respectée.

Les graphes de liens présentent des concepts intéressants comme la notion de causalité, la prise en compte de plusieurs domaines énergétiques. Cependant leur utilisation directe en conception est difficile parce qu'ils sont liés aux équations. Il est donc nécessaire de les adapter pour une utilisation efficace en conception.

## IV. Notre problématique pour une approche fonctionnelle

A partir de l'analyse précédente de différentes approches, nous pouvons conclure que l'approche fonctionnelle est un état d'esprit à avoir pour guider le processus de conception.
Les sous paragraphes suivants présentent les enjeux de notre travail, les hypothèses que nous avons choisi de prendre pour réduire le cadre de notre étude ainsi les limites et les points délicats de notre travail.

## IV.1. Les enjeux de notre travail

Avec notre travail, nous désirons proposer une autre approche du problème de conception des structures de convertisseurs statiques. Celle-ci ne se limite pas au choix de composants, comme c'est le cas pour les systèmes experts classiques, et permet la conception de la topologie.
Ainsi, notre système devra permettre de trouver un maximum de solutions sans a priori afin que l'expert puisse avoir un grand choix de possibilités. Il pourra ensuite supprimer les solutions qu'il juge non performantes. Cette ouverture de notre démarche devrait nous permettre de tendre vers une conception innovante, parce que le raisonnement fonctionnel, permet de limiter au maximum l'utilisation de critères technologiques. Ces derniers sont alors introduits le plus tard possible pendant le processus de conception, pour prendre en compte les évolutions technologiques et obtenir de nouvelles solutions structurelles à partir du même cahier des charges.
Le processus de conception devra être le plus souple possible, c'est à dire que l'expert (l'utilisateur) peut aussi intervenir comme un acteur à part entière. Le système obtenu doit alors travailler sur les données pour gérer leur cohérence et c'est l'expert qui oriente le processus de conception.
A long terme, l'analyse fonctionnelle présentée ici, pourra donc être incluse dans un système à bases de connaissances, pour essayer de modéliser la connaissance des experts sur le choix des fonctions. Cela donnera un système expert d'aide à la conception de structures de convertisseurs statiques.

## IV.2. Les hypothèses de travail

Afin de mener à bien notre étude, nous avons posé les hypothèses suivantes :

- **Nous n'étudions que la partie puissance d'un convertisseur.** L'analyse fonctionnelle définie dans cette étude permet la définition de contraintes vis à vis de la partie commande (au sens de la commande des semi-conducteurs). Après notre analyse, nous connaîtrons l'ensemble des combinaisons inutilisables ainsi qu'un mode de fonctionnement pour le convertisseur statique.

- **Notre outil ne comporte aucune expertise sur le processus de conception.** Seules les règles permettant de gérer la cohérence sont effectivement implantées de façon informatique. Cette restriction nous permet d'avoir un outil plus flexible, l'expertise est apportée par l'utilisateur. Des études futures permettront de définir ces modules d'expertise afin d'automatiser l'exploration des solutions fonctionnelles

- **L'analyse du fonctionnement du convertisseur est faite de façon qualitative.** Tant que la structure n'est pas construite, toute simulation est impossible. Nous sommes donc obligés de manipuler des informations de nature qualitative. Par exemple, il possible de savoir si un courant est sinusoïdal mais il est impossible de connaître avec précision son amplitude.

- **Un graphe fonctionnel est une fonction mais dans un premier temps nous ne gérons pas la composition de fonctions.** En effet, nous ne développons pas de concepts particuliers pour la gestion de la composition de fonctions, mais les graphes fonctionnels que nous développons sont en réalité des fonctions à part entière. Ainsi, un graphe fonctionnel peut toujours être remplacé par une fonction.

- **Nous privilégierons la formulation de structures et fonctions de base.** Afin de ne pas restreindre l'espace des solutions et d'être ouvert, nous proposons de raisonner avec un maximum de fonctions élémentaires.

## IV.3. Les avantages de notre approche

Cette approche présente plusieurs avantages et intérêts.

- **L'approche fonctionnelle permet de gérer facilement l'évolution des solutions.** Cette représentation propose un formalisme détaché au maximum de notions technologiques. Ce niveau d'abstraction permet de prendre en compte facilement les évolutions techniques, par exemple pour faire évoluer une solution structurelle (cf. figure 1.21). Il faudra d'abord construire la représentation fonctionnelle de notre dispositif. Ceci nous permet de travailler sur les services fournis par notre produit, il est d'ailleurs tout à fait possible d'ajouter de nouvelles fonctionnalités. Ensuite il faudra reconstruire une ou plusieurs solutions structurelles en effectuant des choix techniques.

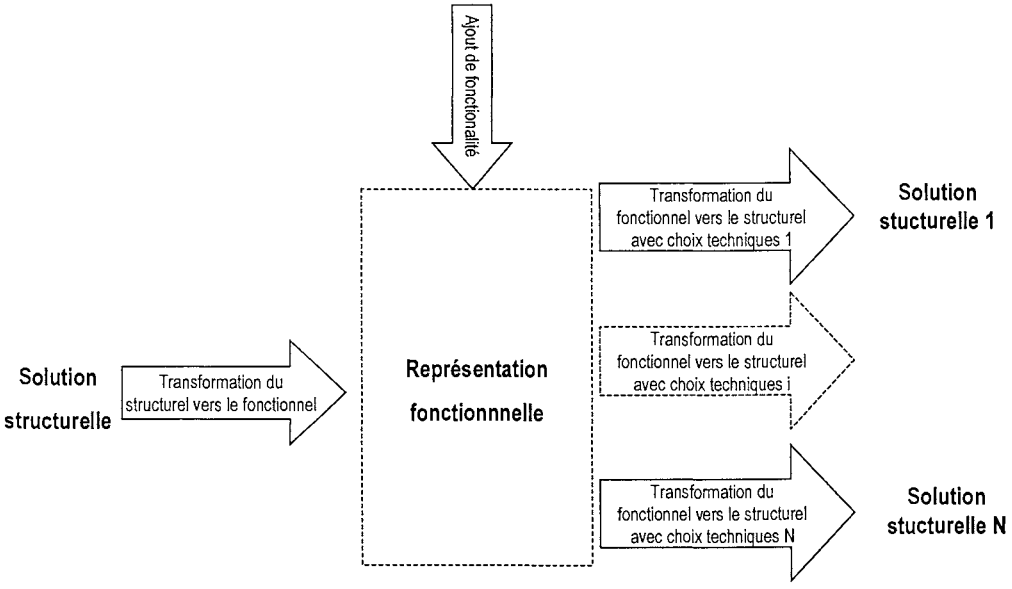



**Figure 1.21 : Notre démarche pour la modification d'une structure**

- **Avec notre méthode, nous pouvons vérifier a priori la cohérence énergétique du produit en cours de conception.** Ceci est un avantage en conception parce que même les solutions partielles peuvent être vérifiées. Le concepteur peut alors se servir

de ces nouvelles connaissances (construites automatiquement par l'analyse énergétique) sur le fonctionnement de son dispositif pour s'orienter dans son activité de conception. Cette analyse de cohérence est a priori possible parce qu'elle est effectuée sans équations donc sans simulation. Nous obtenons alors des temps de réaction quasiment instantanés lorsqu'un problème potentiel est détecté.

- **Notre méthode de conception permet de capitaliser le savoir et le savoir faire.** En effet, il est possible de mémoriser les différentes étapes de la conception. Les solutions en cours de construction sont caractérisées par les fonctionnalités qu'elles remplissent et les contraintes fonctionnelles sur les différentes fonctions. Les choix technologiques sont quant à eux formalisés de façon explicite par les liens entre fonctions et structures. De plus, l'analyse proposée pour les solutions fonctionnelles apporte une connaissance pour l'élaboration de la commande des convertisseurs statiques.

- **Cette approche est bien adaptée à une utilisation dans un système expert** en raison du formalisme de représentation des données.

- **Cette approche permet de raisonner sur des critères qualitatifs,** contrairement à l'approche Bond Graph.

- **Enfin le concepteur pourra choisir, s'il le désire, une méthode conception progressive.** Il peut par exemple considérer dans un premier temps une fonction seulement par son comportement extérieur puis la décomposer sous la forme d'un graphe fonctionnel (composé de ses sous-fonctions) pour continuer sa conception en temps voulu.

## IV.4. Les limites et les points délicats

Le raisonnement fonctionnel présente beaucoup d'avantages, mais possède des difficultés que nous allons présenter maintenant.

- L'analyse fonctionnelle est une méthode très générale qui est utilisée dans beaucoup de domaines différents comme la spécification de projet, la conception de programmes informatiques ou même la définition de produits techniques (c'est notre but). Cette approche générique ne peut évidemment pas être performante pour tous les domaines avec le même formalisme. Il est donc nécessaire de définir approche particulière pour chaque domaine tout en gardant la philosophie fonctionnelle.

- De plus, comme nous l'avons déjà dit, l'analyse fonctionnelle limite au mieux la prise en compte des aspects technologiques. En pratique, dans son raisonnement, l'expert doit très vite utiliser des critères technologiques pour faire progresser la spécification du produit à concevoir. Par exemple, une fonctionnalité de conversion d'énergie peut se réaliser par les dispositifs du tableau 1.2. Pour continuer sa conception, l'expert

doit choisir d'entrée de jeu une technologie de base pour sa réalisation. Ainsi, on peut dire que pour notre application, une solution purement fonctionnelle n'existe pas.

| Type conversion | Choix technologique | Dispositifs |
|---|---|---|
| Electrique/Electrique | Utilisation d'Interrupteurs électroniques | Convertisseurs statiques |
| Electrique/Mécanique/Electrique | Utilisation de moteurs électriques | Convertisseurs a groupes tournants |
| Electrique/Magnétique/Electrique | Utilisation de la technologie magnétique | Transformateurs |
| Electrique/Mécanique | Utilisation de la technologie électromagnétique | Moteur électrique |

**Tableau 1.2 : Quelques dispositifs pouvants être considérés avec une fonctionnalité de conversion d'énergie.**

- L'approche fonctionnelle présente la difficulté de la définition de la limite fonction/structure. Celle-ci est très délicate à exprimer parce qu'elle dépend des fonctions de base de la formulation fonctionnelle utilisée. Suivant les cas, le lien fonction/structure peut être exprimé de plusieurs façons comme cela est présenté dans le tableau 1.3.

| Caractéristique du lien | Formalisation du passage fonctionnel vers structurel | Formalisation du passage structurel vers fonctionnel |
|---|---|---|
| 1 fonction correspond à n structures | Facile, mais la structure finale ne peut pas être optimale en terme de composants | L'identification des fonctions peut être problématique |
| 1 fonction correspond à 1 structure | Facile | Facile |
| n fonctions correspondent à 1 structure | Moyen, il faut bien gérer la composition des fonctions | Facile |
| n fonctions correspondent à m structures | Difficile | Difficile |

**Tableau 1.3 : Le lien fonction structure**

- Actuellement nous n'avons pas formalisé ce lien, mais il semblerait qu'il se présente sous plusieurs formes dans le cas des convertisseurs statiques. L'idéal est de parvenir à un lien du type « une fonction correspond à n structures » pour la phase de conception. Le passage vers une solution structurelle est simple, mais il est alors nécessaire de prévoir une optimisation de la solution en terme de composants. Il

faudra identifier les redondances de composants comme par exemple deux condensateurs connectés en parallèle.

- La réduction d'une solution structurelle peut être délicate. En effet, dans notre approche, si l'on souhaite modifier cette solution, nous demandons au concepteur de retourner à la solution fonctionnelle qui a permis de l'obtenir. Or, par la réduction du graphe structurel, si on n'y prend garde, on peut aboutir au problème délicat à n fonctions pour m structures. Ainsi, une mémorisation des réductions des graphes structurels et de passage des fonctons vers les structures est indispensable.

## IV.5. La méthode de travail

Nous proposons de formaliser une approche fonctionnelle pour la conception des convertisseurs. Pour arriver à un résultat, nous avons choisi une démache de travail en six étapes.

- **1. Analyse d'un choix assez vaste de structures connues.** Cette étape nous a permis d'identifier les différents composants présents dans les convertisseurs statiques.

- 2. **Formalisation d'une base de donnée orientée objet des structures.** Celle-ci regroupe toutes les structures classiques avec pour chacune des informations de caractérisation. Cette étape nous a aussi permis d'extraire des règles d'assemblage de structures. Un extrait de cette base de données est fourni à titre d'exemple en annexe II.

- 3. **Définition des fonctionnalités.** Cette étape est réalisée à partir de la précédente pour établir le concept de fonction.

- **4. Formalisation des critères pour caractériser les graphes fonctionnels.** Il est nécessaire de définir les méthodes de description, de construction et d'analyse des graphes fonctionnels. Celles-ci doivent permettre de guider le concepteur pendant la phase d'élaboration d'une solution fonctionnelle répondant à un cahier des charges donné.

- **5. Mise en œuvre du modèle objet adapté.** Cette étape consiste en la construction du modèle objet adapté à notre formalisme fonctionnel afin de construire une base de connaissance. Cela concerne à la fois la représentation des fonctions et des graphes fonctionnels, ainsi que les méthodes d'analyse et d'aide à l'utilisateur.

- **6 Implantation du modèle sous forme d'un outil.** Celui-ci devra être évolutif c'est-à-dire qu'il devra permettre l'ajout facile des nouvelles fonctions et les procédures implantées devront être les plus génériques possible. Notre outil sera construit de façon indépendante d'un environnement de programmation (dans la mesure du

possible) pour permettre les évolutions faciles et il devra pouvoir s'intégrer dans une chaîne complète de conception.

Les deux premières étapes constituent un travail considérable de pré-analyse incontournable pour établir le modèle fonctionnel. Cependant, les résultats obtenus disparaissent complètement dans ce rapport au profit de la présentation des fonctions et des graphes fonctionnels. Cependant, ce travail sera repris plus tard pour construire la base de données relative aux structures et définir les liens entres les fonctions et les structures qui ne sont pas encore formalisés.

Le chapitre suivant présente notre modélisation des fonctions adaptées au convertisseur statique, ainsi que les fondements de notre représentation objet. Le chapitre 2 s'attachera à décrire les fonctions tandis que le chapitre 3 présentera les graphes fonctionnels.

# Chapitre 2 : Les Fonctions

Ce chapitre présente le principal élément de la modélisation fonctionnelle : la fonction.
Mais avant de parler de fonction ou de structure, il est important de rappeler que la représentation proposée ici ne concerne que la partie puissance des convertisseurs statiques et de leur charge.
Nous développerons donc ici les notions utilisées dans notre modèle, ainsi que les notions de fonctions (ou point de vue fonctionnel) et de structures (ou point de vue structurel).
Puis, nous présenterons la modélisation d'une fonction. Pour cela, nous présenterons les fondements et les modélisations orientées objets pour les différents éléments, ce qui permet de bien faire le lien entre les deux aspects fonctionnel et structurel.

## 0. Préliminaire

Dans ce chapitre, nous ne présentons que les concepts utilisés pour la modélisation des différentes classes permettant de représenter les fonctions. En complément, l'annexe III développe complètement la structure réelle de ces classes, avec notament les différents champs et leur signification.

Afin de simplifier la lecture, nous utiliserons la notation « objet Toto » au lieu de « objet instance de la classe Toto » (voir glossaire).

## I. Définition des notions fondamentales utilisées pour notre application

### I.1. Les grandeurs de références de l'approche proposée : Effort / Flux

Le fonctionnement des systèmes électrotechniques (comme tous les systèmes dynamiques) est régi par des échanges énergétiques. En particulier, un convertisseur statique est un dispositif qui permet de gérer des échanges d'énergie électrique. Pour mener à bien cette opération, un convertisseur peut comporter des fonctions de contrôle de l'énergie, de stockage d'énergie et même de changement de domaine énergétique. Donc, si l'on veut garantir le fonctionnement

du système final, il est important de mettre en place un formalisme pour caractériser les actions des fonctions sur leur monde extérieur (c'est-à-dire les transferts d'énergie) et les interactions entre celles-ci.
Comme nous l'avons vu au premier chapitre, le formalisme Bond-Graph permet de modéliser les systèmes dynamiques et d'effectuer une première vérification en terme de causalité ; mais il n'est pas suffisant dans un contexte de conception. En effet, avec cette approche, il nous est impossible, par exemple, de vérifier les sens du courant au cours du temps ou de modéliser une évolution complexe d'une grandeur. Lorsqu'il conçoit sa structure le concepteur a besoin d'avoir une idée - même grossière - des évolutions qu'il manipule en plus de la notion de causalité pour conclure sur la viabilité de sa structure.
Par exemple, une source de tension alternative (donc avec une causalité effort) ne peut pas être connectée directement à une charge en courant (dans une causalité courant) demandant une tension unidirectionnelle (un moteur à courant continu par exemple), bien que les règles de causalité soient respectées.

La simulation étant impossible sur une structure en cours de construction, notamment parce que les valeurs des composants ne sont pas définies, nous proposons ici une modélisation qualitative des grandeurs. Cela nous permet d'aller plus loin que le formalisme établi par les graphes de liens et de lever l'ambiguïté introduite par la causalité. Ici, nos variables effort et flux sont caractérisées par des formes d'onde qui permettent d'étendre la notion de causalité en caractérisant grossièrement l'évolution des grandeurs. Cette description qualitative pourra être définie par l'expert pour décrire les actions de sa fonction sur son environnement extérieur.

## I.2. Notions complémentaires pour la modélisation fonctionnelle

### I.2.1. Notion de topologie

Les différentes fonctions d'un graphe fonctionnel forment un réseau avec plusieurs trajets possibles pour la circulation des flux d'énergie. Ces trajets ont un support physique et sont appelés liens de transfert d'énergie (voir § Ch 3 : I.1). Ces liens s'établissent en réalité entre les interfaces des fonctions (voir le § II.2.3. ).
La topologie d'un système est l'ensemble des liens de transfert d'énergie qui forment un réseau. Celle-ci peut être hétérogène, c'est à dire qu'elle comporte des liens de natures différentes.
En considérant les interfaces comme des points de vue des fonctions formant des multi-poles, la topologie forme un réseau de Kirchoff [BOI86] avec toutes ses propriétés. Parmi celles-ci, la plus importante est :

***Le flux doit obligatoirement être bouclé pour obtenir un transfert d'énergie entre deux interfaces de fonctions.***

Cette règle très importante justifie l'utilisation de la notion de boucle que nous introduirons au chapitre 3.

### I.2.2. Notion de fonctionnalité

La fonction peut être considérée comme une boîte noire ayant un rôle essentiel dans le système. On appellera fonctionnalité ce rôle. Dans un convertisseur statique, il est possible de définir les fonctionnalités principales suivantes :

- « **Fournir de l'énergie** ». Les fonctions avec cette fonctionnalité modélisent les sources d'énergie. Elles sont utiles pour décrire le cahier des charges.

- « **Recevoir de l'énergie** ». Les fonctions de ce type modélisent les charges (comme les moteurs ...).

- « **Transférer de l'énergie** ». Ces fonctions permettent par exemple d'encapsuler des convertisseurs complets, pour réaliser une conception modulaire.

- « **Changer de domaine énergétique** ». Ces fonctions permettent de modéliser les dispositifs permettant de passer d'un domaine d'énergie à un autre. Nous pouvons citer par exemple les gyrateurs électrique-magnétique qui permettent de passer du domaine électrique au magnétisme et qui correspondent structurellement aux solénoïdes.

- « **Contrôler le passage d'un flux** ». Ces fonctions sont une représentation générale d'une fonction interrupteur.

- « **Assurer le retour d'un flux**». Ces fonctions particulières sont utilisées pour fermer les boucles de flux comme nous le verrons au § ch 3 : I.3.

Une fonction a une seule fonctionnalité et peut être décomposée éventuellement en sous-fonctions pour faciliter l'étape de conception.

### I.2.3. Notion de technologie

Une technologie est une solution technique pour résoudre un problème, dans le but de réaliser une fonction ou pour construire une structure. La technologie peut alors être séparée en deux types :

- **La technologie structurelle (pour figer des composants)**. Celle-ci correspond à un choix technique pour réaliser une fonction. La figure 2.1 présente l'exemple d'une fonction de contrôle de flux électrique (c'est à dire un courant) qui est réalisée entre autre par un transistor Mos ou bipolaire.

- **La technologie fonctionnelle (ou de savoir-faire)**. Cette technologie peut se caractériser par les choix que l'on doit faire en cours de conception. Ceux-ci orientent la solution sans la figer complètement et permettent de conserver un point de vue fonctionnel. Par exemple, un moteur peut avoir la fonctionnalité de faire tourner une roue ; maintenant il nécessaire de choisir une technologie possible pour continuer plus

en avant dans la conception de celui-ci. L'expert peut alors choisir d'utiliser l'électricité comme source d'énergie ce qui oriente le produit vers un dispositif électromagnétique(plutôt que thermique). Souvent ces choix induisent des choix spécifiques et des ajouts de nouvelles fonctions. En électronique de puissance, nous choisirons par exemple d'utiliser une technologie de commutation douce, dure ou résonnante ; puis, suivant le choix fait, nous pourrons déduire des contraintes fonctionnelles sur le reste du convertisseur statique.

En résumé La figure 2.1 présente un exemple des deux choix technologiques dans le cas d'une fonction de contrôle de flux.

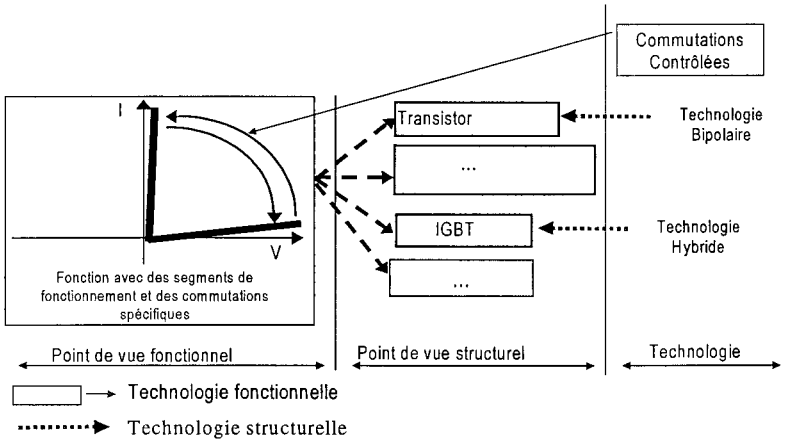



**Figure 2.1 :Exemples des deux types de technologie**

## I.2.4. Notion de contrainte

Dans un graphe fonctionnel, nous pouvons définir deux types de contraintes :

- **Les contraintes énergétiques.** Celles-ci permettent de caractériser les actions des fonctions sur leur monde extérieur. Ces contraintes sont définies à partir des formes d'onde et une causalité inspirée de celle des graphes de liens. Elles présentent notamment plusieurs états qui s'enchaînent au cours du temps. Par exemple une fonction de contrôle du flux présente deux états de contrainte, un état passant où le flux est positif ou négatif et un état bloqué où le flux est forcé à une valeur nulle.

- **Les contraintes temporelles.** La combinaison des états de contraintes permet de définir des états énergétiques d'un graphe fonctionnel. Il est alors nécessaire d'effectuer une analyse des contraintes énergétiques afin de placer des contraintes temporelles dans le but de supprimer un certain nombre de combinaisons (voir ch 3 I.4.). De cette analyse, nous pouvons déduire des critères pour l'élaboration de la commande du convertisseur statique à concevoir.

Les deux paragraphes suivants présentent les notions fondamentales de l'approche fonctionnelle.

## I.3. Point de vue fonctionnel

Le concept de fonction peut se définir dans le cadre des convertisseurs statiques de la façon suivante :

***Une fonction est une entité définie par une fonctionnalité caractérisant son action et les échanges d'énergie ou d'informations avec son monde extérieur.***

Dans le cas des convertisseurs statiques, les échanges avec l'extérieur sont de deux types :

- **De nature énergétique.** La fonction échange de l'énergie avec son monde extérieur par des points d'entrée-sortie identifiés qui définissent des contraintes d'association des fonctions entre elles. Ces points sont aussi appelés interfaces comme nous les définirons au § III.2.3

- **De nature informationnelle.** La fonction peut aussi échanger des informations relatives à son fonctionnement (pour son contrôle ou sa commande par exemple). Dans le cas de l'électronique de puissance, une fonction de contrôle de flux reçoit des ordres de commande de son monde extérieur.

Ainsi, un convertisseur est une fonction qui se compose de plusieurs sous-fonctions qui définissent un graphe fonctionnel par leur agencement.

***Un graphe fonctionnel est un ensemble de fonctions reliées entre elles par des liens énergétiques via les interfaces.***

La figure 2.2 présente un graphe fonctionnel de principe d'un convertisseur. Ici, les fonctions F.1, F.2, F.3 sont des fonctions quelconques et échangent de l'énergie sous le contrôle des fonctions F.C.1 et F.C.2. La fonction F.4 envoie des informations à ces deux dernières afin de contrôler leurs actions dans le temps, cette fonction a une fonctionnalité de commande.

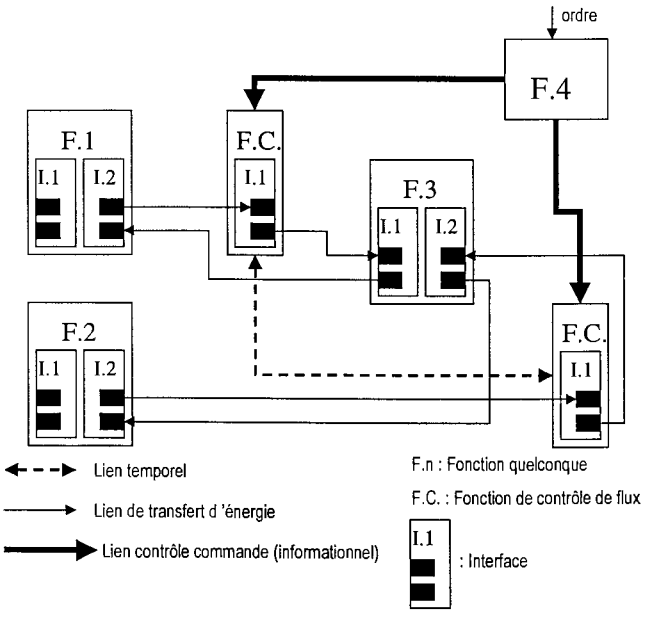



**Figure 2.2 : Un graphe fonctionnel de principe d'un convertisseur statique**

Dans cette étude, nous ne prenons pas en compte l'aspect commande et les liens de commande associés, mais il nous semble que les SADT peuvent compléter efficacement ce manque. Notons que l'analyse nous permet néanmoins d'établir des liens temporels qui sont des contraintes importantes pour la conception de la commande.

Le concepteur pourra utiliser une démarche de conception progressive, pour faire évoluer sa connaissance de son dispositif.

***Une partie de l'acte de conception fonctionnel consiste à passer d'un graphe fonctionnel à un autre et peut nécessiter des choix technologiques fonctionnels afin d'affiner une solution.***

Le graphe fonctionnel final est obtenu lorsque l'on est obligé de ne faire que des choix technologiques structurels et qu'il n'est plus possible d'ajouter de fonctions. Nous donnerons une description plus complète de la construction d'une solution fonctionnelle dans le chapitre 5.

## I.4. Point de vue structurel

Le point de vue structurel est une transformation des fonctions du graphe fonctionnel en utilisant des choix technologiques structurels. Cette étape consiste à choisir une solution technique avec des contraintes sur les valeurs caractéristiques des composants. Les fonctions de contrôle sont par exemple transformées en diode, transistor, ... et ainsi de suite.

***Le point de vue structurel est une solution technique complète (prête au dimensionnement) directement issue du graphe fonctionnel final ; tous ses composants sont spécifiés et un mode de fonctionnement est choisi.***

Ainsi, le point de vue structurel de la partie énergétique correspond au schéma électrique du convertisseur qui est une représentation plus pauvre que la représentation fonctionnelle, parce qu'elle ne présente pas explicitement des informations sur le fonctionnement de celui-ci.
Le passage du graphe fonctionnel au graphe structurel peut se résumer à une «simple» traduction du graphe fonctionnel final où l'on a supprimé les redondances de composants afin de minimiser leur nombre. Cependant le passage inverse n'est pas une opération triviale en raison de la perte d'informations due aux simplifications de la solution structurelle. Nous reviendrons plus précisément sur ce problème de lien entre fonctions et structures dans le chapitre 3.

# II. Modélisation des fonctions

Le but de ce paragraphe est de présenter les concepts utilisés pour définir notre modèle orienté objet afin de modéliser les fonctions. Nous commencerons par décrire les constituants des fonctions, puis nous développerons les différents types de fonctions que nous avons déjà définis afin de concrétiser les concepts présentés. Enfin, nous décrirons les principales méthodes utiles à leur manipulation.

## II.1. Présentation

En utilisant le paradigme objet [ESC96], nous pouvons donner une définition plus précise de la fonction.

- **Une fonction est un objet caractérisé par une fonctionnalité** et qui échange de l'énergie avec son monde extérieur via des interfaces.

- **L'interface est aussi un objet qui permet de regrouper des points d'entrée-sortie du flux d'énergie.** Ces derniers sont liés entre eux par des contraintes fonctionnelles.

- La fonction peut posséder éventuellement des sous-fonctions qui forment alors un sous-graphe fonctionnel (modélisé aussi par un objet).

Ce découpage donne naissance à un réseau d'objets liés entre entre-eux. La figure 2.3 présente la modélisation objet réelle d'une fonction de contrôle de flux. Nous ne présentons dans ce chapitre que les concepts attachés aux différents objets. Pour plus de détails sur la structure des classes et la définition de leurs champs, le lecteur se reportera à l'annexe III.

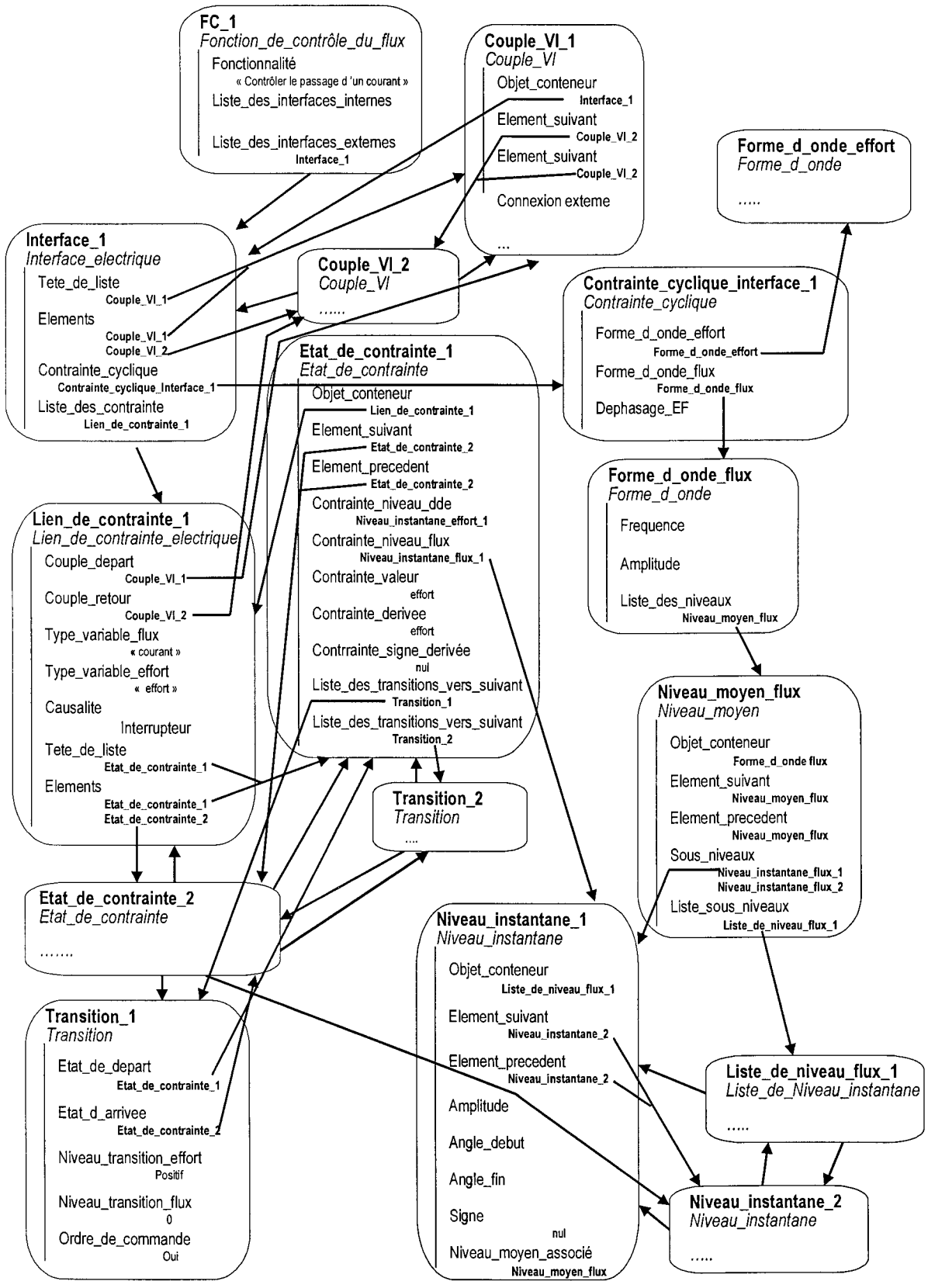



**Figure 2.3 :Modélisation objet d'une fonction contrôle de flux avec présentation de quelques liens entre les objets**

Cette modélisation englobe seulement l'aspect énergétique de la fonction, une représentation similaire doit être mise en place pour la partie commande de la fonction. Nous allons maintenant présenter comment nous avons structuré les fonctions en classes.

## II.2. Les constituants

Une fonction impose ou subit des contraintes de son monde extérieur par le biais de ses interfaces. Cependant, avant de les présenter, nous devons définir la notion de forme d'onde qui permet de compléter la notion de causalité afin de caractériser les transferts d'énergie.

### II.2.1. La classe Forme_d_onde : modélisation de l'évolution des grandeurs

La classe Forme_d_onde est définie pour caractériser qualitativement l'évolution de chaque grandeur (courant, tension, flux magnétique, etc.) au cours du temps. Cette modélisation permet de compléter la notion de causalité afin de fournir des informations qualitatives sur les évolutions temporelles des efforts et des flux sans réaliser de simulation.
La description proposée ici repose sur le fait que les variables en électronique de puissance sont périodiques (dans une optique de fonctionnement en régime permanent) et évoluent en général suivant une combinaison d'évolutions simples (souvent elles comportent des passages quasi constants ou proches d'une sinusoïde ).
Ainsi, nous proposons de modéliser les formes d'onde comme une suite d'objets Niveau. La figure 2.4 présente un exemple de définition d'une grandeur sinusoïdale : elle est définie comme une suite périodique d'un "niveau sinus positif" puis d'un "niveau sinus négatif".

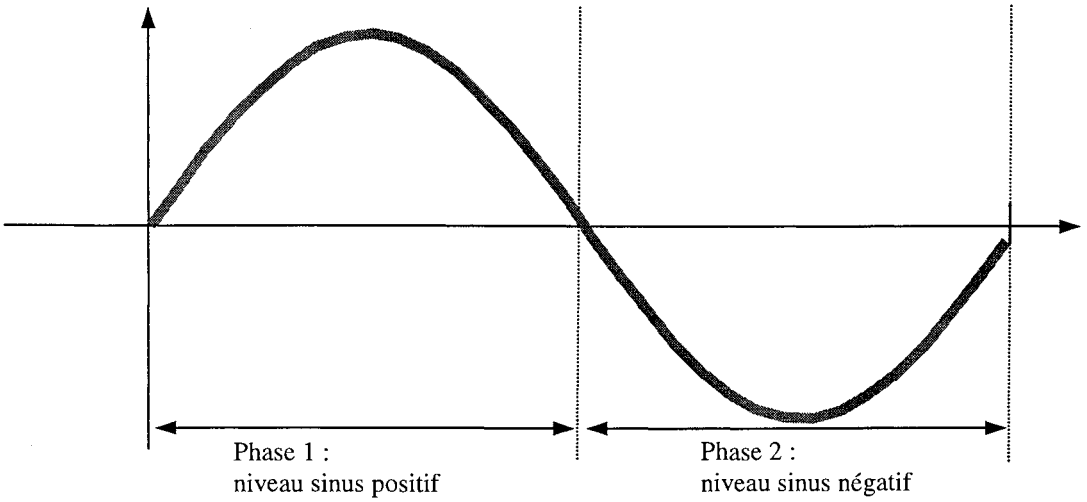



**Figure 2.4 : Définition d'une grandeur sinusoïdale**

Par cette méthode, notre description prend en compte les deux dimensions de l'évolution d'une grandeur :

- **La dimension amplitude :** par la nature des différents niveaux et leurs caractérisations
- **La dimension temporelle :** par la durée des niveaux et leurs enchaînements

Cependant, pour satisfaire les besoins du concepteur utilisant une modélisation fonctionnelle, la classe Forme_d_onde doit répondre aux critères suivants :

- **Elle doit être qualitative.** En effet, c'est le concepteur qui définit les objets de type Forme_d_onde qu'il souhaite aux bornes des éléments de son convertisseur. La forme finale ne sera jamais exactement la même que celle désirée.

- **Elle doit permettre d'appréhender le fonctionnement réel du convertisseur.** Un convertisseur ne peut fournir réellement que des formes simples (constantes, exponentielles, morceaux de sinus, rampes).

- **Elle doit permettre de prendre en compte les formes d'onde évoluées comme des formes MLI (Modulation de Largeur d'Impulsion).** La classe Forme_d_onde est caractérisée par une valeur « moyenne » (qui sera approchée en moyenne) et une valeur instantanée réellement subie par le système pour qu'il réagisse comme si on appliquait la valeur moyenne. Ainsi, le convertisseur doit être conçu en fonction des caractéristiques (amplitude, fréquence) de la forme instantanée. La figure 2.5 illustre l'exemple d'un objet Forme_d_onde dans le cas d'une forme sinusoïdale évoluée.

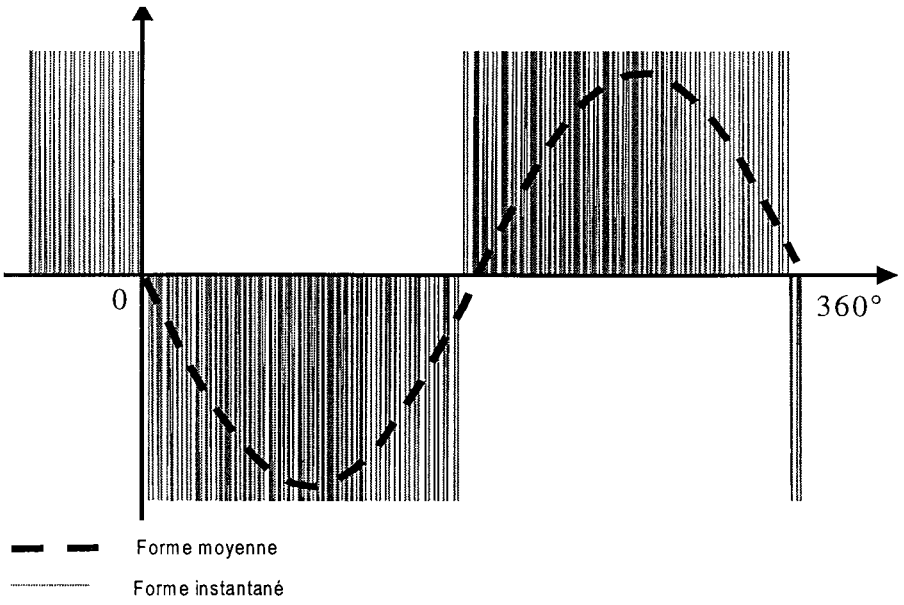



**Figure 2.5 : Exemple d'une forme d'onde évoluée**

Pour satisfaire ces contraintes, nous utilisons en réalité une décomposition double des formes d'onde.

- Pour décrire la forme d'onde idéale, la classe Forme_d_onde se décompose en objets Niveaux_moyens .

- Pour modéliser la façon d'obtenir les niveaux moyens, ces derniers sont décomposés en objets Niveaux_instantanés .

Les différents objets niveaux permettent de caractériser des évolutions standard et très simples. Pour des raisons pratiques, nous les avons organisés de façon hiérarchique comme le montre la figure 2.6 (cf. annexe V).

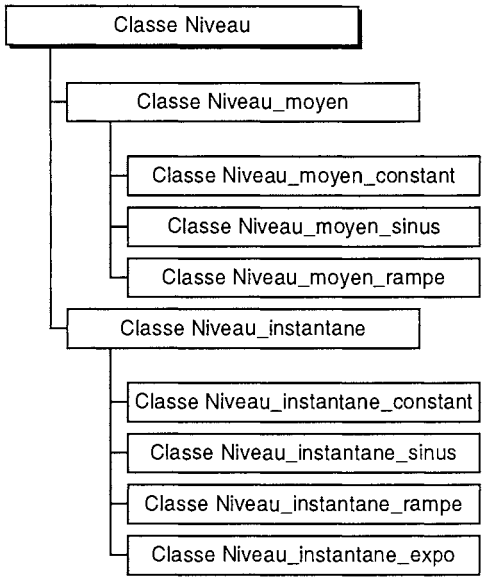



**Figure 2.6 : Les différents niveaux**

Une classe objets niveau est alors caractérisée par

- sa nature (moyen ou instantané),
- une forme caractéristique (constant, sinus, rampe, ...),
- des champs de caractérisation (amplitude, signe ...).

La figure 2.7 présente comment il est possible de modéliser une forme d'onde sinusoïdale qui est approchée par une suite de niveaux constants.

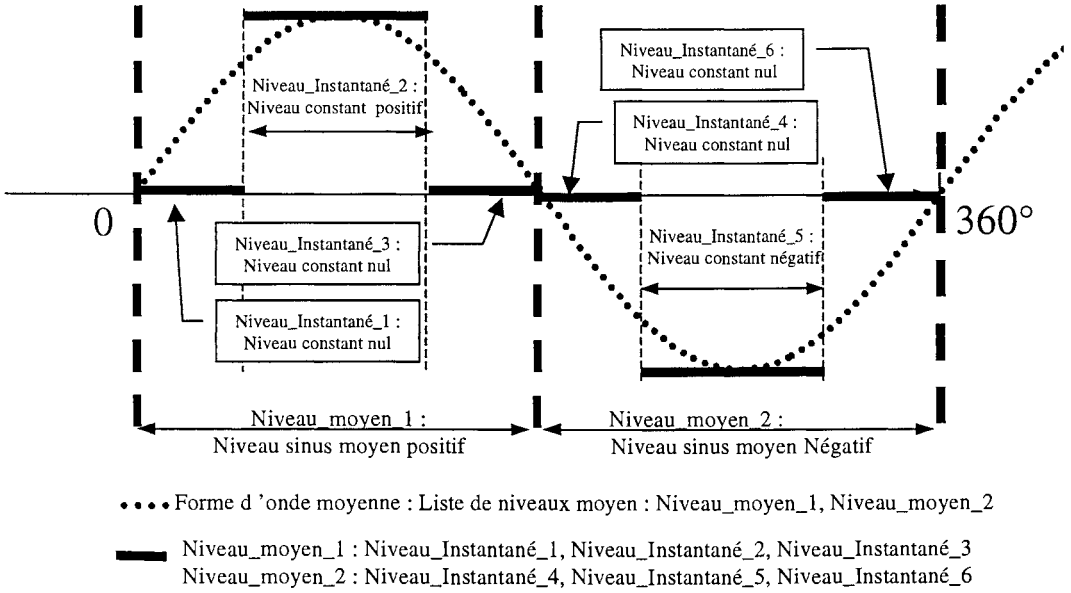



**Figure 2.7 :Décomposition d'une forme d'onde sinus approché**

Dans ce cas, le convertisseur doit fournir les six niveaux instantanés ainsi que leur enchaînement :

Niveau_instantane_1 = > Niveau_instantane_2 => Niveau_instantane_3 = > Niveau_instantane_4 => Niveau_instantane_5 = > Niveau_instantane_6

Dans le cas d'une forme d'onde plus complexe, par exemple une sinusoïde obtenue par une modulation de largeur d'impulsion, nous ne décrivons que les niveaux d'une période de modulation. Ainsi, cela ne concerne que les niveaux instantanés à réaliser ainsi que leur enchaînement sur une période de modulation. La figure 2.8 donne un exemple de cette forme de décomposition. Les objets Niveau_moyen caractérisent la forme que le concepteur désire approcher, alors que sa décomposition en objets Niveau_instantané caractérise le motif de base utilisé pour réaliser la modulation.

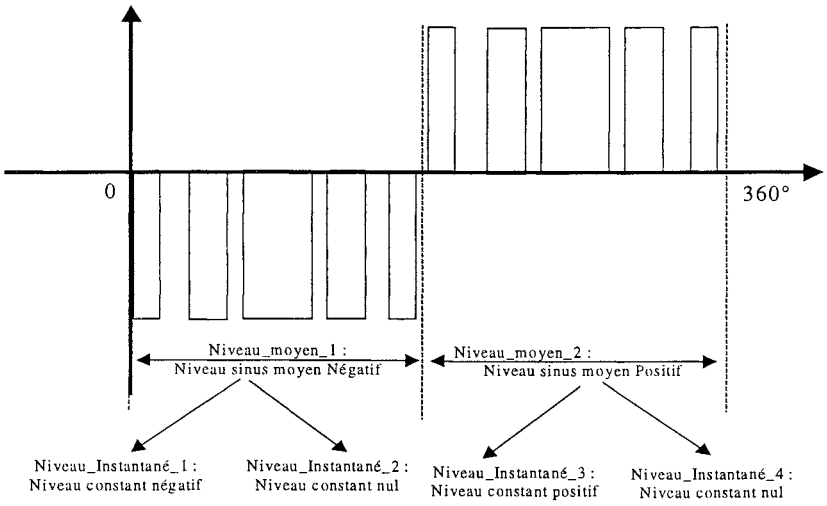



**Figure 2.8 : Décomposition d'une forme MLI**

Cette description nous permet de déduire selon un point de vue fonctionnel les choses suivantes.

- Il faut réaliser les niveaux modélisés par les objets : Niveau_instantane_1, Niveau_instantane_2 et Niveau_instantane_3, Niveau_instantane_4. Les objets Niveau_instantane_2 et Niveau_instantane_4 sont semblables mais nous les dissocions du fait qu'ils appartiennent à des objets Niveau_moyen différents.

- Il faut gérer les enchaînements présentés dans le tableau 2.1. Les transitions entre niveaux moyens correspondent à un changement de motif de la modulation. Il est alors nécessaire de considérer plusieurs cas. En effet, la modulation peut être synchrone (c'est-à-dire que nous n'avons que les transitions du cas 3) ou asynchrone (tous les cas sont possibles).

| Phase de fonctionnement | | Transitions à réalisées |
|---|---|---|
| Niveau_moyen_1 | | *Niveau_instantane_1* => *Niveau_instantane_2*<br>*Niveau_instantane_2* => *Niveau_instantane_1* |
| Niveau_moyen_2 | | *Niveau_instantane_3* => *Niveau_instantane_4*<br>*Niveau_instantane_4* => *Niveau_instantane_3* |
| Transition Niveau_moyen_1 Niveau_moyen_2 | Cas 1 : | *Niveau_instantane_1* => *Niveau_instantane_3* |
| | Cas 2 : | *Niveau_instantane_1* => *Niveau_instantane_4* |
| | Cas 3 : | *Niveau_instantane_2* => *Niveau_instantane_3* |
| | Cas 4 | *Niveau_instantane_2* => *Niveau_instantane_4* |
| Transition Niveau_moyen_2 Niveau_moyen_1 | Cas 1 : | *Niveau_instantane_3* => *Niveau_instantane_1* |
| | Cas 2 : | *Niveau_instantane_3* => *Niveau_instantane_2* |
| | Cas 3 : | *Niveau_instantane_4* => *Niveau_instantane_1* |
| | Cas 4 : | *Niveau_instantane_4* => *Niveau_instantane_1* |

**Tableau 2.1 : Transitions à réaliser pour la forme MLI**

Avant de présenter les interfaces qui constituent les éléments principaux des fonctions, nous allons définir une autre notion importante : le Couple_EF.

## II.2.2. La Classe Couple_EF : identification des points de sortie des interfaces

### *II.2.2.1. Définition*

La théorie de systèmes dynamiques démontre que les transferts d'énergie dans les systèmes dynamiques sont régis par la combinaison d'une grandeur extensive (l'effort) et d'une grandeur intensive (le flux) [DEL92]. Une source peut alors imposer un effort (ou un flux) et le récepteur par réaction impose le flux (ou l'effort). Ces deux grandeurs sont indissociables lors de la caractérisation de la circulation des flux, c'est pour cela que nous avons défini les objets Couple_EF qui permettent d'identifier les points de passage ou les points de connexion des interfaces des fonctions. Afin de prendre en compte plusieurs domaines énergétiques, il existe plusieurs types de Couple_EF. Le tableau 2.2 présente quelques possibilités.

| Système physique | Nature de l'effort | Nature du flux | Couple_EF |
|---|---|---|---|
| Electrique | Tension | Courant | Couple_VI |
| Magnétique | Force magnétomotrice | Dérivé du flux magnétique | Couple_FemDfm |

**Tableau 2.2 : Définition des différents Couple_EFs**

### *II.2.2.2. Représentation graphique*

Dans tous les diagrammes fonctionnels, nous utiliserons la notation de la figure 2.9 pour chaque Couple_EF.

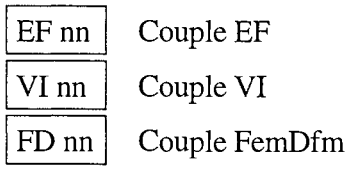
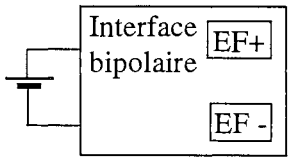
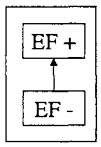
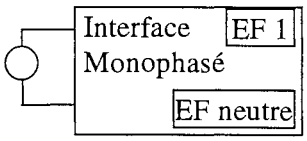
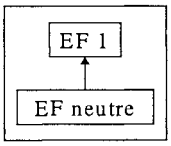
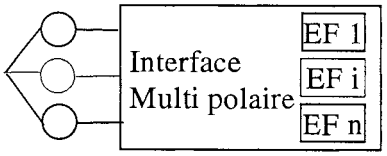
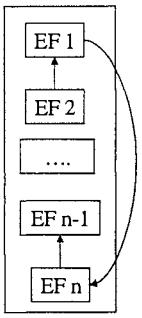



**Figure 2.9 : Représentation des objets Couple_EF**

### II.2.3. La classe Interface : regroupement des objets Couple_EF dépendants d'une fonction

L'interface est une entité qui regroupe les points d'entrée-sortie d'une fonction qui possèdent des liens fonctionnels entre eux. C'est donc un point de vue énergétique de la fonction qui permet d'identifier facilement ses points d'échanges énergétiques avec l'extérieur.

Par définition, la somme des flux circulant par les objets Couple_EF d'une interface est toujours nulle. Les principaux types d'interfaces sont alors définis suivant les modèles du tableau 2.3.

| Type d'interface | Exemple | Définition des objets Liens_de_contraintes |
|---|---|---|
| Fonction avec une interface continue |  |  Objet Lien_de_contrainte |
| Fonction avec une interface monophasée |  |  Objet Lien_de_contrainte |
| Fonction avec une interface polyphasée |  |  Objet Lien_de_contrainte |

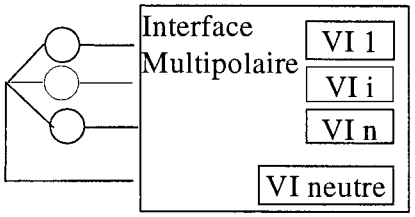
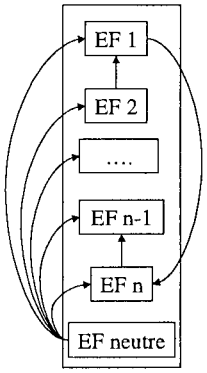
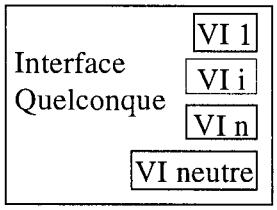
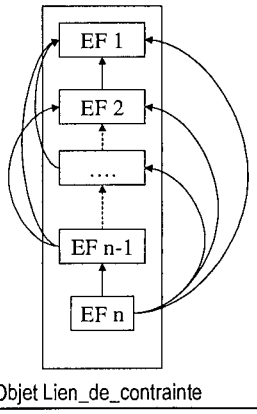
| | | |
|---|---|---|
| Fonction avec une interface polyphasée avec neutre |  |  |
| Fonction avec une interface quelconque |  |  |

**Tableau 2.3 : Principaux types d'interface**

Suivant le type de l'interface, il est possible de définir des contraintes fonctionnelles énergétiques qui représentent un chemin possible pour le flux. Les interfaces sont donc construites en respectant les critères suivants.

- **Lorsque l'interface comporte deux objets Couple_EF**, ils sont liés par un objet Lien_de_contrainte.

- **Lorsque l'interface est polyphasée avec plus de deux objets Couple_EF, sans Couple_EF neutre ;** ces objets constituent alors les éléments d'une liste cyclique et sont liés deux à deux par un objet Lien_de_contrainte. Ce dernier est défini à l'aide des mêmes objets Forme_d_onde, mais avec un décalage temporel (de valeur 360 / nombre_de_phases). Cela signifie que les objets Lien_de_contriante d'une même interface sont définis de la même façon (aux symétries près représentées par les déphasages).

- **Lorsque l'interface est polyphasée avec plus de deux objets Couple_EF et avec un Couple_EF neutre ;** les objets Lien_de_contrainte sont crées de la même façon que précédemment mais ici chaque Couple_EF est lié au Couple_EF neutre par un Lien_de_contrainte.

Nous reviendrons en détail dans le paragraphe III.2.4.3 sur la définition de la classe lien_de_contrainte.

Il est nécessaire de particulariser l'objet Interface suivant son domaine énergétique. Par exemple, une interface électrique comportera des objets Couple_VI (plutôt que des objets Couple_EF utilisés dans le cas général) et des objets Lien_de_contrainte_electrique. Le tableau 2.4 présente différentes possibilités de spécialisation de cette classe.

| | **Domaine général** | **Domaine électrique** | **Domaine magnétique** |
|---|---|---|---|
| **Classe de l'objet interface** | Interface | Interface_electrique | Interface_magnetique |
| **Classe des objets du champ Elements** | Couple_EF (effort/flux) | Couple_VI (tension,courant) | Couphe_FemDfm (fem, dérivée temporelle du flux magnétique) |
| **Classe des objets Lien_de_contrainte** | Lien_de_Contrainte | Lien_de_ Contrainte_electrique | Lien_de_ Contrainte_magnetique |

**Tableau 2.4 : La particularisation des interfaces**

Afin de gérer simplement l'extension du modèle, la classe interface est organisée de façon hiérarchique comme le montre la figure 2.10. Ses sous-classes n'ont pas de champs propres, mais cette spécialisation permet de déterminer facilement le type de chaque objet Interface. L'ajout d'un nouveau type d'interface se fait seulement en ajoutant une nouvelle sous-classe. Les méthodes de gestion de la classe Interface étant écrites de façon générique, elles seront utilisables pour toutes les sous-classes de la classe Interface.

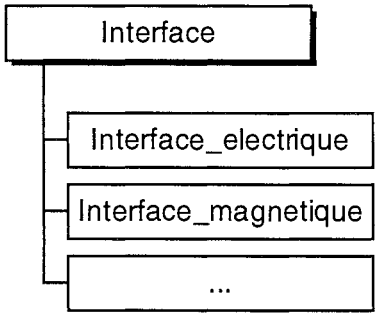



**Figure 2.10 : Organisation de la classe Interface**

Les paragraphes suivants présentent la description des contraintes énergétiques liant les objets Couple_EF.

## II.2.4. La modélisation des contraintes énergétique

### *II.2.4.1. La classe Lien_de_contrainte : Description des actions d'une interface entre deux Couple_EF*

Les objets Lien_de_contrainte sont utilisés afin de modéliser les actions énergétiques d'une fonction entre deux objets Couple_EF d'une même Interface. Ils permettent de caractériser la

façon dont une fonction impose ou subit un transfert d'énergie. Un tel transfert se fait obligatoirement par une circulation de flux entre deux objets Couple_EF d'une même Interface. La façon dont la fonction impose un transfert s'appelle la causalité (de façon similaire aux graphes de lien) et il est possible de dénombrer plusieurs cas.

- **L'effort est imposé (causalité Effort)**. Dans ce cas, l'interface impose une différence d'effort entre les deux objets Couple_EF et subit le flux, les fonctions avec des causalités de ce type sont des sources d'effort.

- **La variation de l'effort est imposée (causalité Derivee_Effort).** Ce cas a été ajouté par rapport à la notion classique des graphes de liens pour modéliser des comportements capacitifs. Cette causalité signifie que la différence d'effort entre les deux objets Couple_EF ne doit pas subir de discontinuités en valeur au cours du temps.

Il existe des définitions équivalentes au niveau du flux circulant entre deux Couple_EF.

- **Le flux est imposé (causalité Flux).**

- **La variation de flux est imposée (causalité Derivee_flux).**

De plus, nous ajoutons les causalités suivantes pour compléter la caractérisation.

- **Aucune grandeur n'est imposée (causalité Passif)**. Dans ce cas, la fonction n'impose a priori aucune grandeur. Le comportement est similaire à celui d'une résistance : si la fonction subit l'effort (et respectivement le flux) alors elle réagit avec le flux (et respectivement l'effort).

- **Les cas particuliers (causalité Quelconque).** Dans ce cas, la causalité est variable et la description de contraintes doit être décomposée en états (cf. § III.2.4.2).

- **Les interrupteurs (causalité Interrupteur)**. C'est une déclinaison particulière de la causalité quelconque, la contrainte présente quatre états comme l'illustre la figure 2.12. Cette valeur particulière de causalité est utilisée pour identifier facilement les fonctions de contrôle de flux.

***Avertissement sur la notion de causalité : Par abus de langage, nous avons introduit ici les termes « causalité dérivée » et « causalité interrupteur » par extension de la notion de causalité définie par les Bond Graphs . De ce fait, ces notions n'ont pas de signification physique, mais elles permettent de caractériser des comportements spéciaux (des éléments de stockage, et des interrupteurs) avec une caractérisation homogène. Il aurait été plus judicieux de parler de « contrainte sur la variation » au lieu de causalité dérivée mais l'utilisation trop fréquente du terme contrainte dans notre modèle est une source de confusion.***

***Remarque : les fonctions de contrôle de flux ont un comportement particulier parce qu'elles ne peuvent avoir que quatre états : le flux peut être positif ou négatif avec un effort nul ou l'effort peut être positif ou négatif seulement avec un flux nul. Elles sont définies avec la causalité Interrupteur et ne comportent que quatre objets Etat_de_contrainte.***

D'une façon similaire aux classes héritées d'Interface et aux classes héritées de Couple_EF, la classe Lien_de_contrainte est organisée de façon hiérarchique pour prendre en compte les domaines énergétiques. En complément, elle possède deux champs propres (nommés Type_variable_flux et Type_variable_effort) qui sont utilisés pour particulariser les classes en terme de nom de variable comme le montre la figure 2.11.

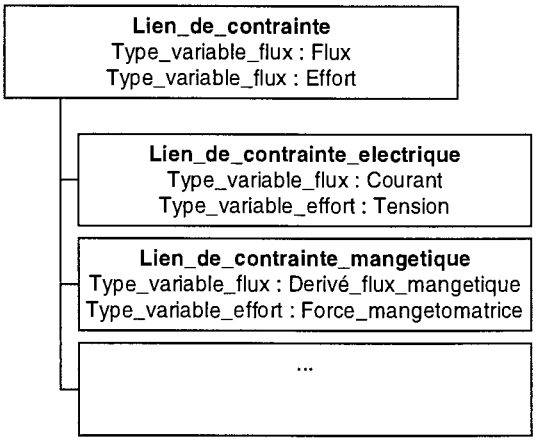



**Figure 2.11 : La hiérarchie des classes lien_de_contraintre**

### *II.2.4.2. La classe Etat_de_contrainte : modélisation de l'évolution dans le temps des objets Lien_de_contrainte*

Pour certains éléments, la causalité n'est par constante au cours du fonctionnement (c'est par exemple le cas pour les interrupteurs). Dans ce cas, l'objet Lien_de_contrainte est composé d'une liste d'objets Etat_de_contrainte contenus dans son champ Elements. La figure 2.12 présente par exemple les objets possibles Etat_de_contrainte d'une fonction de contrôle de flux.

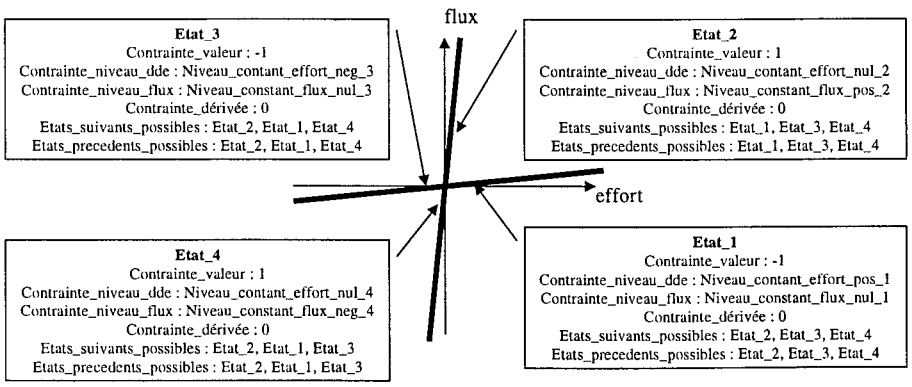



**Figure 2.12 : Définition des objets Etats_de_contrainte d'une fonction de contrôle de flux**

Chaque objet Etat_de_contrainte est défini à partir d'un objet Niveau_instantané de l'objet Forme_d_onde_effort attaché au Lien_de_contrainte, et d'un objet Niveau_instantané de l'objet Forme_d_onde_flux attaché au Lien_de_contrainte. La notion de causalité doit être étendue avec les notions supplémentaires suivantes.

- En plus de la notion d'effort ou de flux imposé, nous définissons un niveau imposé et un niveau moule (à respecter si possible). La définition de ce dernier fait appel à l'expérience de l'expert alors que le niveau imposé fait partie de la définition même de la fonction.

- La notion de dérivée imposée est étendue, pour prendre en compte le signe de celle-ci, et peut être définie de façon indépendante de la notion précédente.

En réalité, les objets Etat_de_contrainte sont toujours construits (même lorsque la causalité est constante) pour obtenir une modélisation homogène de toutes les fonctions.
Par exemple, la figure 2.13 présente la décomposition en objets Etat_de_contrainte dans le cas d'une source de tension sinusoïdale. Ici, la forme du courant a été fixée par le concepteur et ne constitue pas une contrainte ; elle est seulement une estimation moyenne de la forme de courant recherchée.

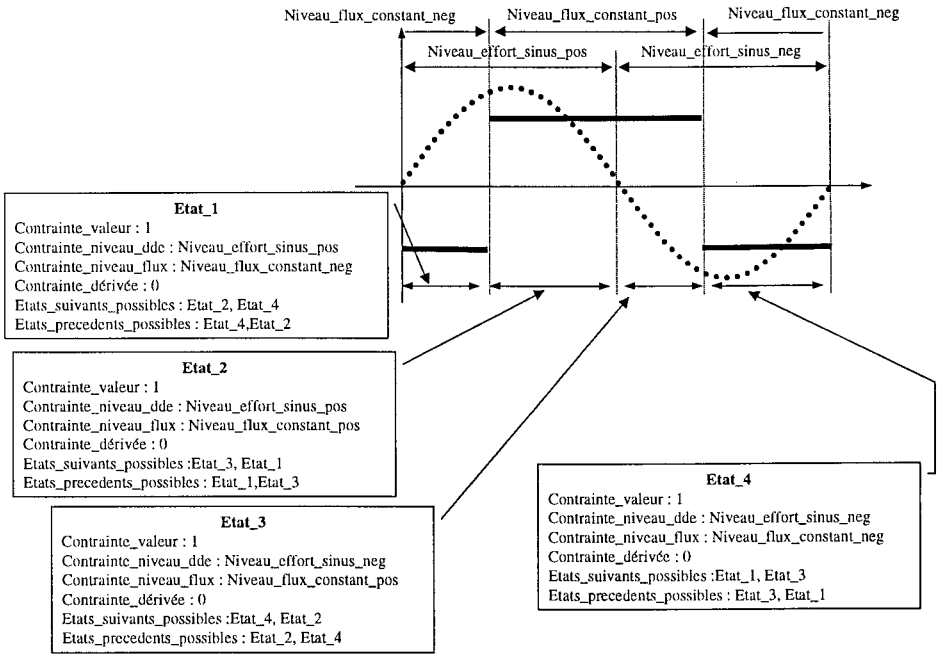



**Figure 2.13 : La décomposition en objets Etat_de_contrainte d'une fonction sinusoïdale**

Dans ce cas, il y a quatre états qui s'enchaînent de façon cyclique. Il est important de noter que cette représentation dépend complètement du déphasage entre les grandeurs flux et effort. Ceci est illustré sur la figure 2.14 qui présente la même fonction sinusoïdale avec un déphasage différent. Ceci implique qu'il est très important de fixer certaines valeurs de déphasage afin de spécifier les enchaînements.

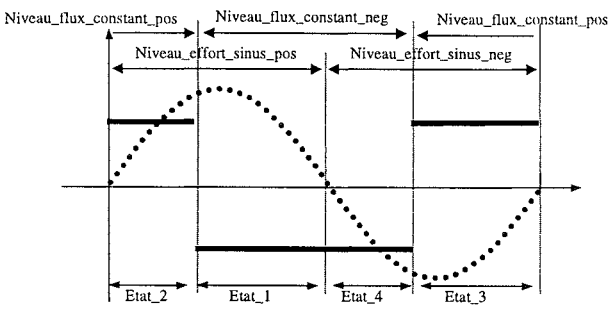



**Figure 2.14 :La même fonction que la précédente avec un déphasage différent des grandeurs**

L'analyse des enchaînements entre les objets Etat_de_contrainte permettra de spécifier les fonctions de contrôle de flux et de vérifier par exemple si les contraintes de continuité sur les grandeurs sont toujours respectées au cours du temps (voir chapitre 3).
Nous avons vu comment sont modélisés les différents états des objets Lien_de_contrainte, nous allons maintenant présenter la modélisation des changements d'états.

### *II.2.4.3. La classe Transition : modélisation des conditions de changement d'objets Etat_de_contrainte*

Les objets de la classe Transition permettent de modéliser les conditions de changement d'état d'une fonction. Ils sont notamment utilisés pour la modélisation des fonctions de contrôle de flux. Les transitions permettent de mémoriser les conditions de changement d'objets Etat_de_contrainte. Ceux-ci peuvent se produire naturellement ou sur un ordre de commande. La figure 2.15 présente un exemple de la définition des objets Transition utilisés dans le cas d'une fonction de contrôle de flux.

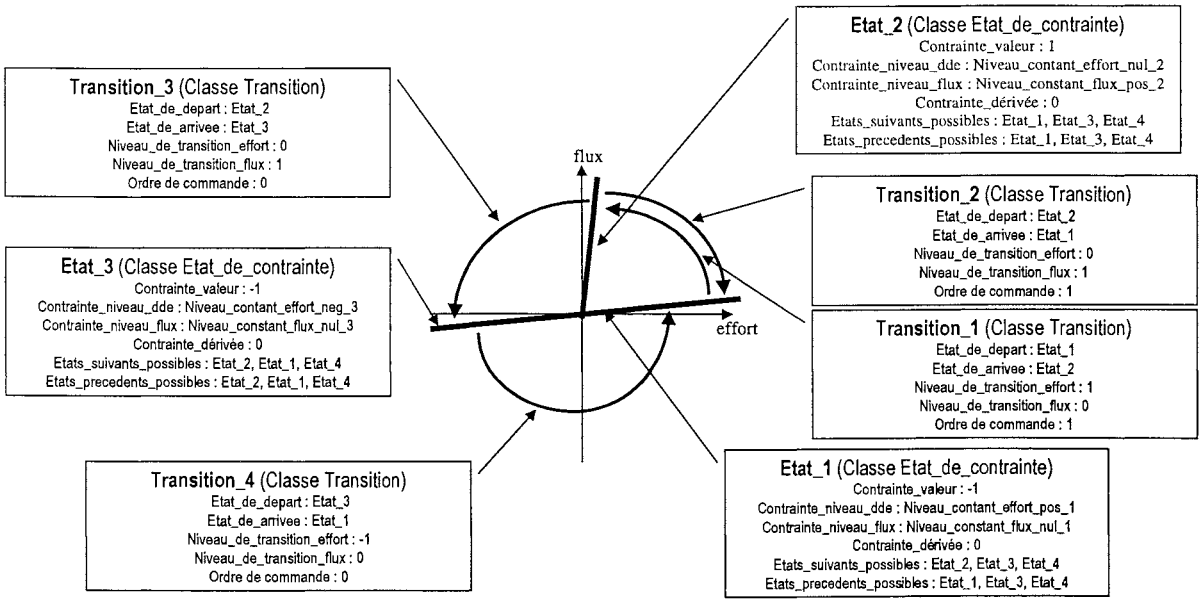



**Figure 2.15 : Exemple de définition des objets Transition**

### *II.2.4.4. Représentation graphique de contraintes modélisées par les objets Lien_de_contrainte*

Dans les différents graphes fonctionnels, les contraintes sont représentées dans les interfaces. La figure 2.16 présente un exemple dans le cas d'une interface avec deux objets Couple_EF.

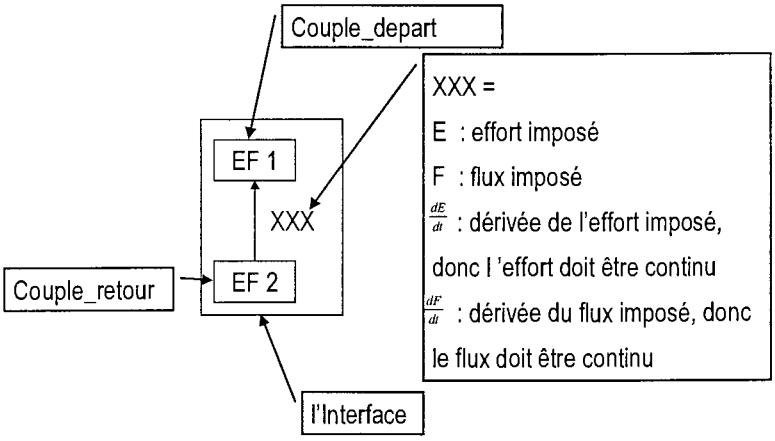



**Figure 2.16 : Représentation d'une interface avec ses contraintes**

## II.3. La classe Fonction_generique : Classe principale de caractérisation des fonctions

Nous pouvons maintenant définir la classe Fonction_generique. Celle-ci sert de base pour la définition de toutes les fonctions. Elle définit la caractérisation d'une fonction ainsi que la liste de toutes ses interfaces avec le monde extérieur.

Les fonctionnalités sont définies pour représenter le rôle principal de la fonction qui peut être par exemple :

- **Fournir.** Ces fonctions ont un comportement source d'énergie.
- **Consommer.** Ce sont les fonctions qui consomment de l'énergie, elles correspondent structurellement à une charge.
- **Contrôler un flux.** Ce sont les fonctions de contrôle de flux correspondent structurellement aux interrupteurs.
- **Assurer un retour de flux.** Ces fonctions garantissent l'indépendance des boucles de flux (cf. Chapitre III).
- **Stocker.** Ces fonctions permettent de stocker de l'énergie.
- **Transférer de l'énergie.** Ces fonctions permettent d'encapsuler des graphes fonctionnels complets pour définir des fonctions plus complexes.

Les autres fonctionnalités sont définies en particularisant une des fonctionnalités précédentes par l'ajout d'une causalité et d'une indication sur une forme d'onde. Avec cette méthode, il est possible de définir par exemple les fonctionnalités suivantes :

- **Fournir une tension continue.** Cette fonctionnalité définit une fonction fournissant de l'énergie avec une interface électrique présentant une forme d'onde continue pour la grandeur effort ; du point de vue structurel, cela peut être une batterie par exemple.

- **Consommer un courant continu.** Nous utiliserons cette fonctionnalité pour définir une fonction consommant de l'énergie via une interface électrique avec une forme d'onde continue pour le courant ; du point de vue structurel, c'est une source de courant qui absorbe de l'énergie

- **Contrôler un courant.** C'est une fonction de contrôle de flux dans le domaine électrique. C'est par exemple un interrupteur statique.

- Etc.

Pour des raisons d'efficacité dans la reconnaissance des fonctions, nous avons effectué la classification présentée sur la figure 2.17. Cette méthode permet de retrouver très facilement les fonctions par leur classe ; ainsi la fonctionnalité permet donc une connaissance plus fine de la fonction.

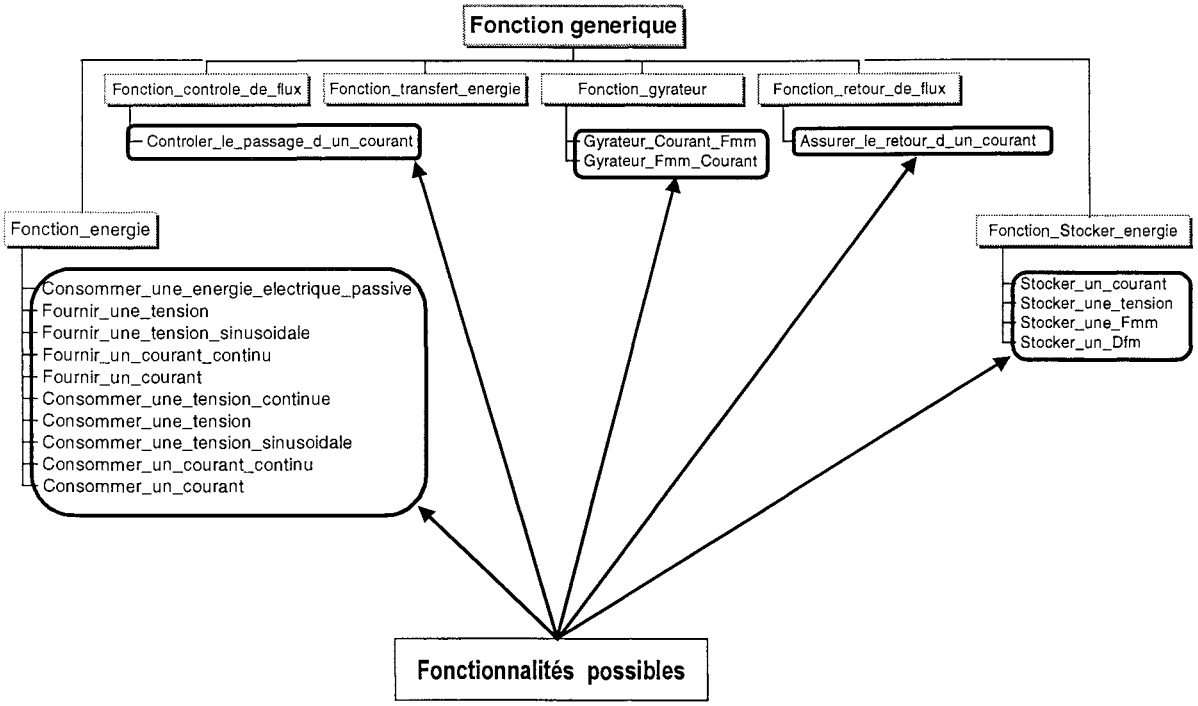



**Figure 2.17 : hiérarchie des fonctions**

Nous allons maintenant présenter les différentes sous-classes de la classe Fonction_generique.

### II.3.1. La classe Fonctions_energie

Les fonctions de cette classe produisent ou consomment de l'énergie. En effet, la différence entre ces derniers types de fonctions est purement conventionnelle. Il est cependant nécessaire de définir un sens positif pour le transfert d'énergie afin de pouvoir définir le cahier des charges et les raisonnements de l'expert pendant l'acte de conception.
Les fonctionnalités de base (consommer ou fournir) sont complétées pour obtenir une définition automatique des interfaces. Le tableau 2.5 présente quelques exemples de spécialisations. Celles-ci peuvent être combinées pour fabriquer de nouvelles fonctions et avoir des fonctionnalités explicites.

| Type particularisation | Nom Particularisation | Domaine énergétique | Utilisation |
|---|---|---|---|
| Causalité | Flux | Général | Interface avec une causalité flux |
| | Courant | Electrique | Interface_electrique avec une causalité flux |
| | Effort | Général | Interface avec une causalité effort |
| | Tension | Electrique | Interface_electrique avec une causalité effort |
| | Passif ou Passive | Général | Interface avec une causalité passive |
| Energétique | Energie Electrique | Electrique | Utilisé pour définir une interface électrique |
| Sur les Formes | Alternatif | | Les interfaces sont monophasées ou polyphasées avec ou sans neutre |
| | Continu | | Les interfaces sont bipolaires (2 objets Couple_EF) |

**Tableau 2.5 : Quelques particularisations des fonctions énergies**

### II.3.2. La classe Fonction_stocker_energie

Pour réaliser des transformations d'énergie dans les convertisseurs, il est souvent nécessaire de réaliser des stockages d'énergie. C'est pourquoi nous avons défini ce type de fonction avec une fonctionnalité principale : Stocker. Comme pour les fonctions d'énergie, il existe des particularisations possibles pour cette fonctionnalité (voir le tableau 2.6).

| Type particularisation | Nom Particularisation | Domaine énergétique | Utilisation |
|---|---|---|---|
| Grandeur stockée | Flux | Général | Interface avec un stockage du flux |
| | Courant | Electrique | Interface_electrique avec un stockage du flux |
| | Effort | Général | Interface avec un stockage de l'effort |
| | Tension | Electrique | Interface_electrique avec un stockage de l'effort |

**Tableau 2.6 : Particularisations des fonctions de stockage**

Il n'y a que deux façons de réaliser cette fonction dans un système physique.

- **Par accumulation du flux**. Par le biais de ses interfaces, la fonction impose l'effort et accumule le flux. Ce dernier constitue un niveau d'énergie accumulé dans la

fonction. Il en résulte que le flux ne doit pas être discontinu (le niveau d'énergie ne peut varier de façon instantanée). Les fonctions de ce type comportent donc une interface avec une causalité Dérivée_flux. D'un point de vue structurel, elles peuvent correspondre par exemple à une inductance.

- **Par accumulation de l'effort**. C'est la fonction duale de la précédente. Ici, les fonctions de ce type ont une causalité Derivee_effort. Sur le plan structurel, elles peuvent correspondre par exemple à un condensateur.

### II.3.3. La classe Fonction_contrôle_de_flux.

Les fonctions de contrôle de flux régissent le fonctionnement des convertisseurs. Elles autorisent ou non le passage d'un flux dans une branche d'une topologie. Ces fonctions comportent donc plusieurs états :

- **un état passant** où la fonction autorise le passage d'un flux (sans transformation de celui-ci) ; dans ce cas, la contrainte effort est imposée à un niveau nul,

- **un état bloqué** où la fonction interdit le passage d'un flux ; dans ce cas, la contrainte flux devient alors imposée à un niveau nul.

### II.3.4. La classe Fonctions_de_retour

Une boucle de flux est constituée de fonctions reliées entre elles via des connexions.
Un graphe fonctionnel est un ensemble de boucles interconnectées où transite un flux. Afin de permettre d'envisager le fonctionnement de chaque boucle de façon indépendante, nous utilisons des fonctions spéciales (appelée fonctions de retour) en complément des fonctions de contrôle du flux.

**Remarque : Les fonctions de retour sont considérées équivalentes aux fonctions de contrôle, pendant le raisonnement fonctionnel. Ce n'est qu'au moment du passage à la représentation structurelle que ces fonctions sont réellement spécifiées.**

Afin d'illustrer ce problème, prenons l'exemple de la figure 2.18. Dans ce cas, le concepteur définit sans précautions trois boucles :

- la boucle 1 : source – FC1 (fonction de contrôle de flux) – charge,

- la boucle 2 : source – FC2 (fonction de contrôle de flux) – charge1 – FC1,

- la boucle 3 : source – FC3 (fonction de contrôle de flux) – charge1.

Ici, il en résulte que le graphe n'est pas viable. En effet le passage à l'état passant de la fonction de contrôle de flux FC1 (pour autoriser un flux dans la boucle_1) provoque la mise en court-circuit de la fonction Charge_1 (boucle I_1). Ce fonctionnement n'est peut-être pas

prévu à l'origine et peut donc nécessiter de modifier le graphe fonctionnel. Ces modifications peuvent s'avérer lourdes dans le cas d'un graphe complexe.

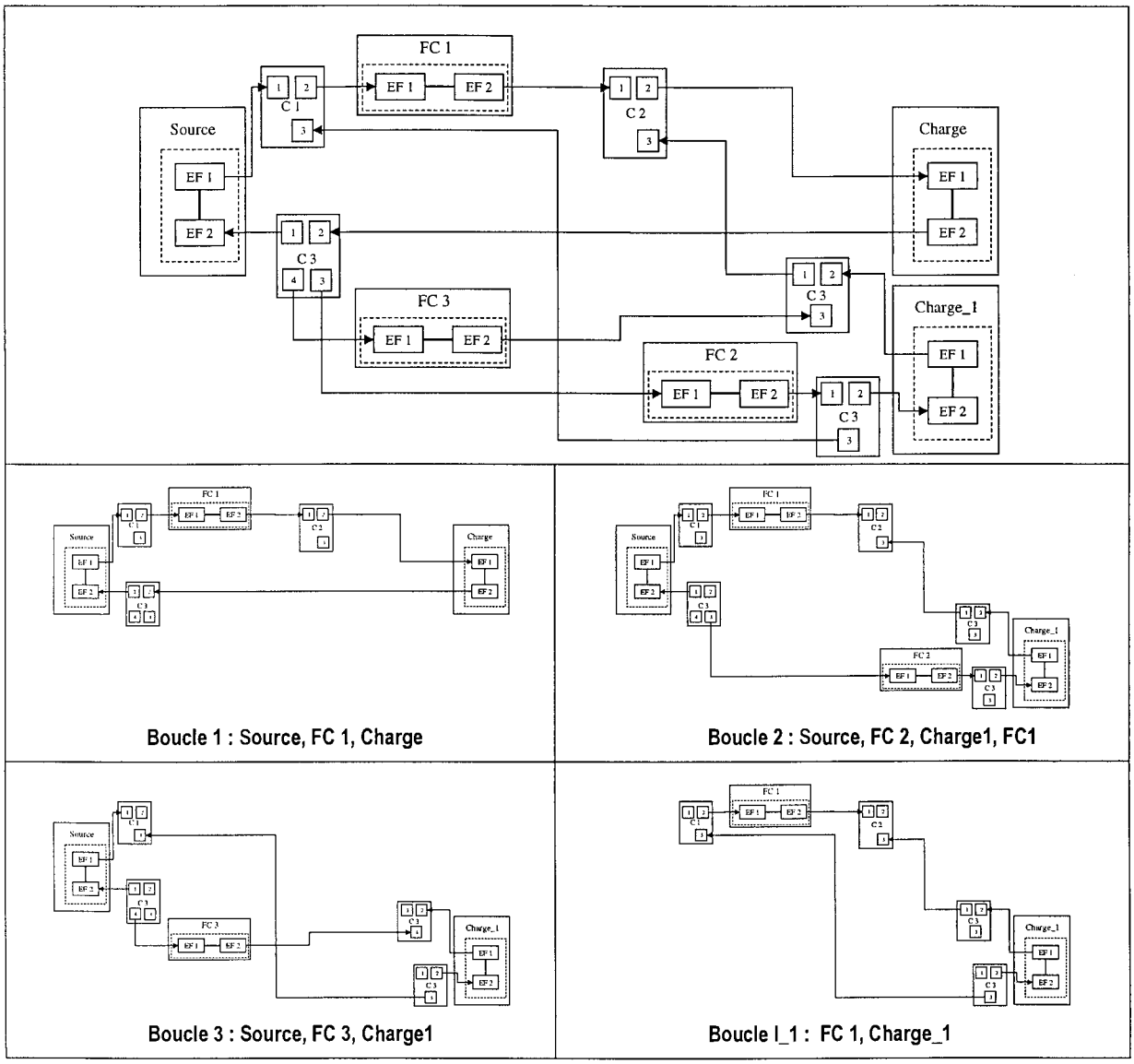



**Figure 2.18 : Exemple d'un graphe fonctionnel mal construit**

Pour simplifier la tâche du concepteur, il est préférable d'utiliser systématiquement une fonction miroir de la fonction de contrôle de flux (notre fonction de retour), pour fermer chaque boucle de flux indépendamment des autres boucles lors de son élaboration. Il en résulte que le problème ne se produit plus parce que les fonctions n'ont plus de liaison directe entre elles (voir figure 2.19) et les boucles peuvent fonctionner de façon indépendante.

Une fois le graphe fonctionnel complet construit, il est possible de supprimer les fonctions de retour (en les remplacent par un lien de transfert simple) si elles s'avèrent effectivement inutiles. Dans le cas contraire, elles seront remplacées par des fonctions de contrôle de flux (structurellement des interrupteurs) dont la commande sera définie à l'aide des résultats du processus d'analyse de cohérence du graphe fonctionnel.

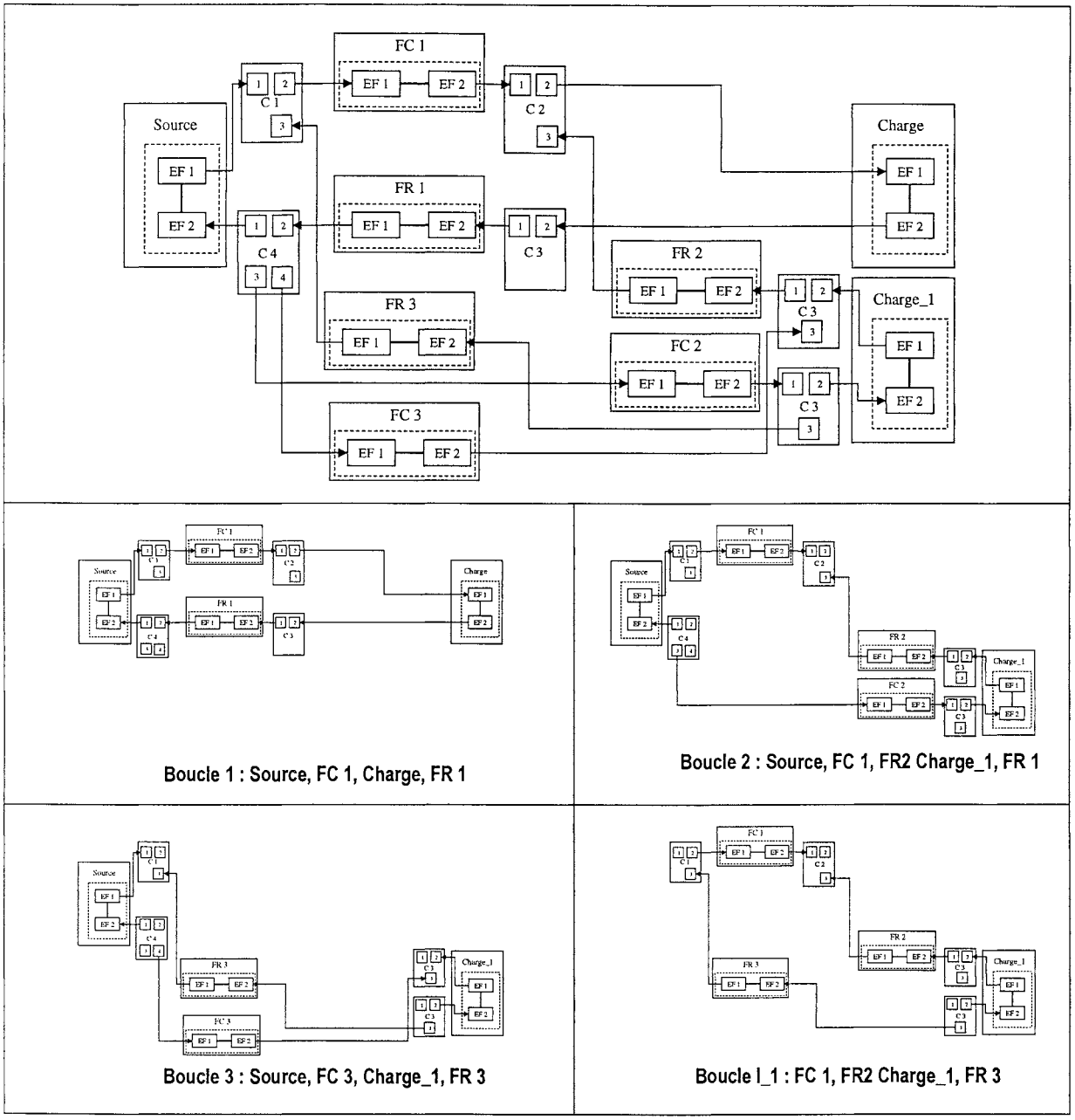



**Figure 2.19 : Une illustration de la nécessité de la fonction de retour**

## II.3.5. La classe Fonction_gyrateur

### *II.3.5.1. Présentation*

Les fonctions gyrateurs permettent de modéliser de façon fonctionnelle les dispositifs présentant des changements de domaine énergétique.

Compte tenu des domaines énergétiques qui apparaissent dans le cadre des convertisseurs statiques, nous n'allons définir ici qu'un seul type de gyrateur : le gyrateur électrique-magnétique (noté Gyrateur_Courant_Fmm) qui peut être aussi le gyrateur magnétique-électrique (noté Gyrateur_Fmm_Courant) suivant la convention utilisée sur le sens de transfert d'énergie.

Ces dispositifs peuvent être représentés par des équations linéaires. Ainsi, il est possible de définir les relations suivantes pour les modéliser :

si $\alpha$ est le rapport de transfert du gyrateur alors

$$F_{d2} = \frac{E_{d1}}{\alpha} \qquad (2.1)$$

$$E_{d2} = \alpha \cdot F_{d1} \qquad (2.2)$$

avec $E_{d1}$ l'effort du domaine 1, $E_{d2}$ l'effort du domaine 2

et $F_{d1}$ le flux du domaine 1, $F_{d2}$ le flux du domaine 2

De ces relations constitutives, il en résulte que les causalités sont aussi modifiées de part et d'autre du gyrateur. Le tableau 2.7 présente les quatre principaux cas possibles sur la propagation de la causalité dans les gyrateurs.

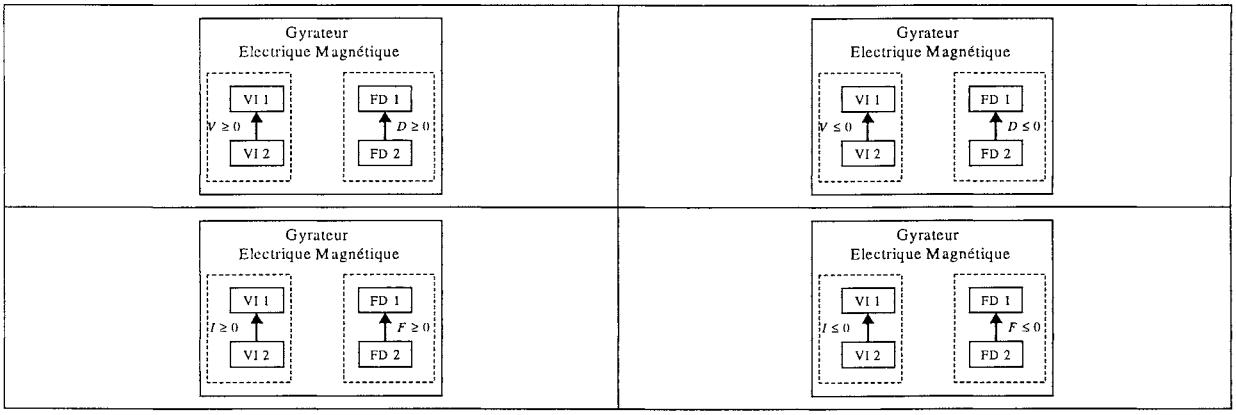



**Tableau 2.7 : La Propagation des contraintes par les gyrateurs**

Maintenant, dans le cas particulier du gyrateur magnétique-électrique, on obtient les relations suivantes.

Le domaine 1 est le domaine magnétique, donc le flux est la dérivée du flux magnétique (notée $Dfm = \frac{d\phi m}{dt}$ où $\phi m$ le flux magnétique) et l'effort est la force magnétomotrice c'est-à-dire une tension (notée $Fmm$).

Le domaine 2 est le domaine électrique, donc le flux est un courant (noté $I$) et l'effort une tension (noté $E$).

Le rapport $\alpha$ est une constante notée $N$.

L'équation ( 2.1 ) devient $N \cdot I = Fmm$. **( 2.3 )**

Puis l'équation ( 2.2 ) devient $E = N \cdot \frac{d\phi m}{dt}$ (loi de Lentz pour un solénoïde de N spires). **( 2.4 )**

Les équations ( 2.3 ) et ( 2.4 ) sont les relations constitutives d'un solénoïde et donnent le point de vue structurel de ce gyrateur : une bobine.

### *II.3.5.2. Application des gyrateurs*

La principale application des gyrateurs dans la conception des convertisseurs statiques est la modélisation des fonctions d'isolation « magnétique ». En effet pour réaliser une isolation énergétique, il est parfois nécessaire de réaliser un changement de domaine énergétique. Pour cela, plusieurs solutions sont envisageables.

- **Une isolation optique.** L'énergie électrique est convertie de l'électricité vers la lumière (domaine optique) puis transformée à nouveau en électricité. La technologie actuelle pour cette forme d'isolation ne permet pas de transférer une forte puissance ce qui la rend inutilisable pour la partie puissance du convertisseur. Néanmoins, cette solution peut être très performante pour la commande.

- **Une isolation magnétique.** L'énergie électrique est cette fois-ci convertie en énergie magnétique. Cette technologie permet de faire circuler une grande quantité d'énergie, mais présente quelques restrictions et des inconvénients. En effet, le circuit magnétique n'est pas neutre par rapport au parcours du flux, il sera nécessaire de prendre en compte ses effets pour obtenir une bonne modélisation du fonctionnement.

Pour réaliser les fonctions d'isolation dans la partie puissance (le cadre de notre étude), nous utiliserons donc un ensemble de gyrateurs électrique-magnétique que l'on peut représenter sous sa forme structurelle par le schéma de la figure 2.20. Cependant, pour canaliser le flux magnétique, il est nécessaire d'utiliser un circuit magnétique en matériau ferromagnétique qui n'est pas transparent au transfert énergétique (contrairement au domaine électrique, où les fils comportent une résistance que l'on peut presque toujours négliger).

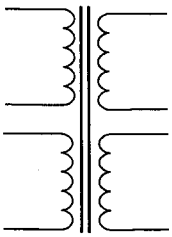


**Figure 2.20 : La représentation structurelle de la fonction isolation magnétique**

Afin d'avoir une représentation uniforme, nous considérons que les liens de transfert d'énergie ne modifient pas l'énergie quel que soit le domaine énergétique (ni dissipateur, ni inertie, ni capacité). Si besoin est, nous modélisons leur influence par l'ajout d'une fonction fictive dans le graphe fonctionnel. Afin de trouver un graphe fonctionnel pour la fonction isolation, partons de la physique du problème :

Pour le $k^{eme}$ gyrateur électrique magnétique nous avons les relations :

$$Fmm_k = N_k \cdot I_k \qquad (2.5)$$

$$E_k = N_k \cdot \frac{d\phi m}{dt} \qquad (2.6)$$

Quel que soit le circuit réel, il est toujours possible (par une série de transformations) de trouver un circuit magnétique équivalent tel que tous les enroulements soient parcourus par le même flux. Ils sont alors liés par la relation suivante :

$$\sum_{k=1}^{ng} Fmm_k = \Re \cdot \phi m \qquad (2.7)$$

avec $\Re$ la réluctance équivalente du circuit de parcours du flux (c'est-à-dire le circuit magnétique si on néglige les fuites) et ng le nombre de gyrateurs

Si l'on pose $Cm = \frac{1}{\Re}$ et en faisant apparaître la grandeur flux du domaine magnétique $Dfm = \frac{d\phi m}{dt}$ l'équation ( 2.7 ) devient

$$Cm \cdot \sum_{k=1}^{ng} Fmm_k = \int Dfm \Leftrightarrow Dfm = Cm \frac{d\sum_{k=1}^{ng} Fmm_k}{dt} \qquad (2.8)$$

Le circuit magnétique se comporte donc comme une capacité magnétique puisque qu'il accumule la somme des efforts qui lui sont appliqués. A titre d'exemple, un transformateur à quatre enroulements peut se modéliser selon le schéma de la figure 2.21. La circulation d'un flux positif, va entraîner une accumulation d'effort dans la capacité magnétique, il sera donc nécessaire de faire circuler un flux négatif pour provoquer une décharge magnétique (ou une accumulation de l'effort dans le sens opposé). Il sera aussi nécessaire de garantir la continuité de l'effort dans la capacité magnétique.

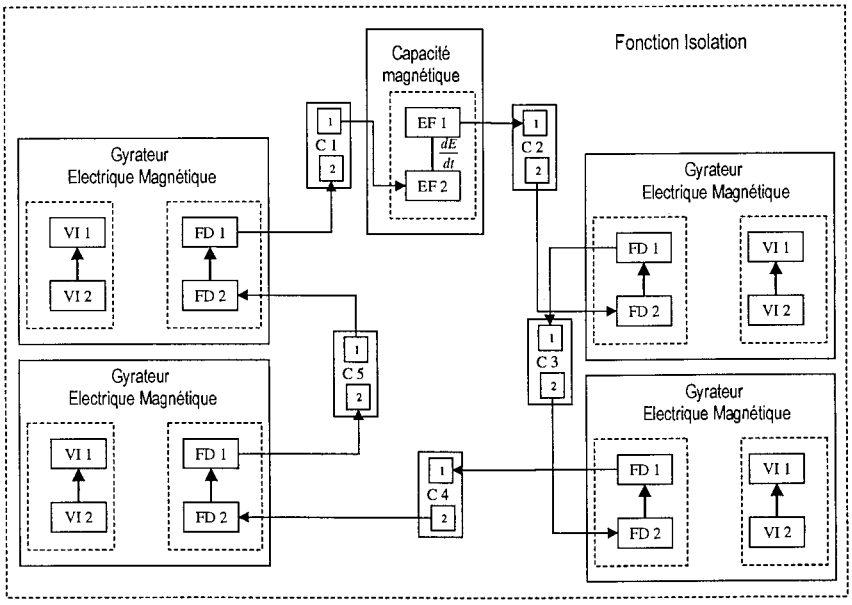



**Figure 2.21 : Modélisation fonctionnelle d'un transformateur**

### *II.3.5.3. Perspective des gyrateurs*

Comme nous venons de le voir, les gyrateurs permettent la modélisation fonctionnelle des dispositifs ayant des changements de domaine énergétique. Ils permettent notamment de

modéliser les fonctions d'isolation, mais il est tout à fait envisageable de définir des ensembles plus complexes comme des moteurs électriques en utilisant une méthode similaire. En effet, un système électromagnétique peut se décrire comme un ensemble de gyrateurs et le circuit magnétique effectue un couplage spatial des flux. Il reste à trouver une modélisation fonctionnelle qui modélise bien les contraintes énergétiques du moteur en utilisant une approche similaire à celle des transformateurs.

## II.4. Création des objets Fonction : instanciation d'une classe Fonction

### II.4.1. Principe

Les fonctions peuvent se résumer à une fonctionnalité qui conditionne la création d'une ou plusieurs interfaces. La création des fonctions comporte trois étapes :

**1. L'étape de définition de la fonctionnalité.** Le concepteur doit choisir la fonctionnalité de la fonction qu'il veut créer parmi les choix possibles. Le processus de création pourra créer un objet Fonction à partir de la classe appropriée. Le tableau 2.8 rappelle les différents choix possibles. Il est alors possible d'effectuer les opérations suivantes.

**2. La création des interfaces.** A partir de la causalité, le processus de création peut déterminer le nombre d'interfaces à créer, leur causalité et leurs formes d'onde caractéristiques. Le tableau 2.8 présente des exemples de définition des interfaces en fonction des différentes fonctionnalités. Le processus de création des interfaces sera expliqué au paragraphe suivant.

**3. La création de la représentation graphique.** Dans l'optique de la réalisation d'un outil de C.A.O., il est nécessaire de créer une représentation graphique de chaque fonction pour permettre à l'utilisateur de manipuler les différentes fonctions existantes et effectuer des liaisons énergétiques entre elles.

| Fonctionnalité | Type d'interface | Causalité | Forme d'onde |
|---|---|---|---|
| Fournir_une_tension | Interface_electrique | Effort | Non défini à priori |
| Fournir_une_tension_sinusoidale | Interface_electrique Monophasé ou polyphasée | Effort | Effort : sinusoïdal<br>Courant : alternatif |
| Fournir_un_courant_continu | Interface_electrique bipolaire | Flux | Effort : non défini<br>Flux : constant |
| Fournir_un_courant | Interface_electrique | Flux | Non défini à priori |
| Consommer_une_tension | Interface_electrique | Effort | Non défini à priori |
| Consommer_un_courant | Interface_electrique | Flux | Non défini à priori |
| Controler_le_passage _d_un_courant | Interface_electrique | Interrupteur | Effort : alternatif<br>Flux : alternatif |
| Assurer_le_retour _d_un_courant | Interface_electrique | Interrupteur | Effort : alternatif<br>Flux : alternatif |
| Gyrateur_Courant_Fmm | Interface_electrique | Gyrateur | Effort : alternatif<br>Flux : alternatif |
| | Interface_magnetique | Gyrateur | Effort : alternatif<br>Flux : alternatif |

**Tableau 2.8 : Quelques définitions d'interfaces à partir des fonctionnalités**

## II.4.2. Création des interfaces

Les interfaces caractérisent les transferts énergétiques des fonctions. Il est pratique de définir les procédures générales de manipulation des interfaces au niveau énergétique pour permettre une extension facile du modèle. Seules les données regroupées dans le tableau 2.9 sont différentes suivant le type des interfaces.

| | Domaine énergétique | | |
|---|---|---|---|
| | ***Général*** | ***Electrique*** | ***Magnétique*** |
| **Type des interfaces** | Interface | Interface_electrique | Interface_magnetique |
| **Type des Couple_EF** | Couple_EF | Couple_VI | Couple_FemDfm |
| **Type des Etats de contrainte** | Etats_de_contrainte | Etats_de_contrainte | Etats_de_contrainte |
| **Type des contraintes cycliques** | Contrainte_cyclique | Contrainte_cyclique | Contrainte_cyclique |
| **Type des Liens de contrainte** | Contrainte | Contrainte_electrique | Contrainte_magnetique |
| **Nom de la grandeur effort** | Effort | Tension | Force magnétomotrice |
| **Nom de la grandeur flux** | Flux | Courant | Dérivée flux magnétique |

**Tableau 2.9 : Les informations liées aux types des interfaces**

La classification des interfaces est quant à elle intéressante pour particulariser ou modifier les méthodes de manipulation de celles-ci. Quel que soit le domaine énergétique, la procédure de

création d'une interface est la même. Sa décomposition en tâches (en procédures) est décrite sur la figure 2.22 que nous allons expliquer maintenant.

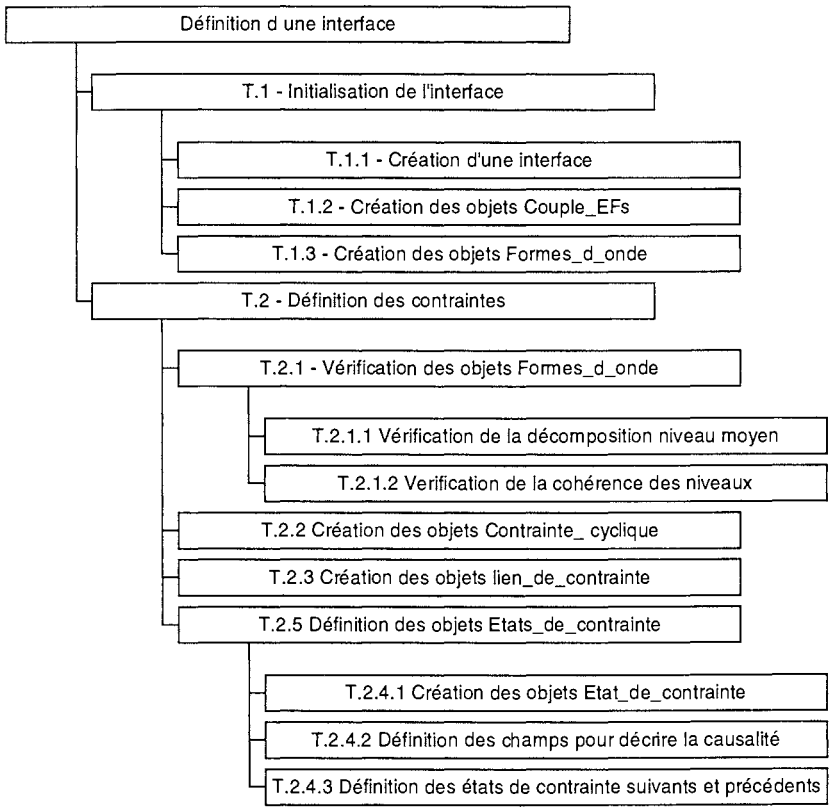



**Figure 2.22 : La décomposition en tâche du processus de création d'une interface**

### *II.4.2.1. L'initialisation d'une interface (T.1)*

C'est l'étape préliminaire de la création d'une interface. Elle consiste en la création d'un objet interface du bon type (T.1.1) qui va déterminer le type des objets Lien_de_contraintes et des objets Couple_EF en accord avec le tableau 2.9.
Dans un second temps, la tâche T.1.2 crée les objets Couple_EF en fonction du type de l'interface. Leur nombre doit être déterminé par l'expert en fonction des règles suivantes :

1. *Si l'interface est de type continu alors le nombre d'objets Couple_EF est égal à 2.*
2. *Si l'interface est de type alternatif sans neutre alors le nombre d'objets Couple_EF doit être supérieur à ou égal 2.*
3. *Si l'interface est du type alternatif avec neutre alors le nombre d'objets Couple_EF doit être supérieur ou égal à 1.*

Enfin la tâche T.1.3 crée les objets Forme_d_onde (pour caractériser le flux et l'effort) s'ils n'ont pas été définis par le type de la causalité en appliquant les règles suivantes :

4. *Si l'interface est de type continu alors la forme d'onde est décomposée en un niveau moyen. C'est à l'expert de déterminer le signe des niveaux.*

5. *Si l'interface est de type alternatif alors la forme d'onde est décomposée en deux niveaux moyens de longueur 180° avec des signes opposés (sans tenir compte la phase).*

### *II.4.2.2. La création des contraintes (T.2)*

La procédure de création des contraintes est fondamentale. C'est elle qui conditionne la bonne utilisation des interfaces. Elle est composée de cinq étapes principales :
La première phase (T.2.1) est la vérification des objets Forme_d_onde. Ils sont vérifiés avec les critères suivants.

6. *Les formes d'onde doivent être décomposées en objets Niveau_moyen pour décrire le comportement de l'interface*
7. Les niveaux moyens doivent être décomposés en niveaux instantanés. En effet, ces derniers doivent être obtenus par le convertisseur, et ils sont la base de la définition des états de contrainte.
8. Les niveaux doivent être continus d'un point de vue temporel. Cela signifie que la fin du niveau N doit correspondre au début du Niveau N+1. Cette vérification doit être menée pour les niveaux instantanés et les niveaux moyens.
9. *La forme d'onde doit être définie sur une période de fonctionnement complète. En d'autres termes, les angles des niveaux doivent aller de 0° (pour le premier niveau moyen ) à 360 (pour le dernier niveau moyen).*

Les deux tâches qui suivent (T.2.2, T.2.3) permettent la création des différents objets remplissant les champs de l'interface : la contrainte cyclique (partie commune de l'interface) et les liens de contraintes.
Ensuite la tâche T.2.4 permet de définir les états de contraintes pour chaque contrainte à partir de la description en niveaux instantanés des formes d'onde. Chaque état est crée à partir d'un niveau instantané de la forme d'onde effort combiné avec un niveau instantané de la forme d'onde flux (T2.4.1).

Enfin la tâche T.2.4.2 permet de définir les champs décrivant les grandeurs imposées en fonction de la causalité principale (qui est définie au niveau du lien de contrainte). Cette opération est résumée sur le tableau 2.10.

| Causalité principale | Contrainte valeur | Contrainte dérivée | Contrainte signe dérivée |
|---|---|---|---|
| Effort | Niveau_dde (1) | Derivée_effort (1) | Définie par expert |
| Flux | Niveau_flux (-1) | Derivée_flux (-1) | Définie par expert |
| Passif | Aucune (0) | Aucune (0) | Aucune |
| Derivee_effort | Aucune (0) | Derivée_effort (1) | Définie par expert |
| Derivee_flux | Aucune (0) | Derivée_flux (-1) | Définie par expert |
| Quelconque | Définie par expert | Définie par expert | Définie par expert |
| Interrupteur | 4 états Voir Figure 2.12 | | |

**Tableau 2.10 : Définition des champs des états de contrainte**

Ce chapitre a présenté les concepts de notre approche ainsi que la modélisation des fonctions avec leurs décompositions en objet. Le chapitre suivant va maintenant présenter les graphes fonctionnels, ainsi que leur vérification afin de déduire automatiquement des informations sur la définition de la commande.

# Chapitre 3 : Le graphe fonctionnel

Les structures de base de la description des fonctions étant définies, ce chapitre présente d'abord les éléments – autres que les fonctions - d'un graphe fonctionnel. En particulier, cela concerne la notion de lien de transfert d'énergie, les connexions, la notion de boucle et les liens fonctionnels. Puis, il décrit la construction des graphes fonctionnels avec ses différentes phases : la création des fonctions puis le processus d'analyse en vue de la gestion de la cohérence.

## I. Les éléments d'un graphe fonctionnel

### I.1. Les liens de transfert d'énergie (LTE)

Un LTE est un canal de transport de l'énergie d'une fonction à une autre. Ils sont caractérisés par un support physique ainsi qu'un procédé physique qui régit le transfert. Le tableau 3.1 présente quelques types de lien de transfert d'énergie. Par convention, les LTE n'ont pas d'action sur l'énergie, c'est-à-dire qu'ils ne sont ni dissipateurs ni réservoirs d'énergie. Si besoin, nous prenons en compte les effets du transfert en ajoutant une fonction fictive qui modélise les effets souhaités. L'analyse des fonctions d'isolation en est une illustration.

| Système physique | Supports physiques du transfert d'énergie<br>**Procédé physique** |
|---|---|
| Electrique | Fils électriques<br>(en métal, conducteur électrique)<br>**Par conduction** |
| Magnétique | Matériaux ferromagnétiques<br>air<br>**Par conduction et par rayonnement** |
| Thermique | Tous<br>air<br>**Par conduction, rayonnement et convection** |
| Mécanique | Solide<br>**Par conduction** |

**Tableau 3.1 : Exemple de natures physiques de liens de transfert d'énergie**

D'un point de vue informatique, les LTE ne sont pas matérialisés par des objets ; ce sont des références mutuelles entre un objet Connexion (via son champ listes_de_Couple_EF) et un Couple_EF (via son champ Connexion_externe ). La figure 3.1 illustre les LTE ainsi que les connexions que nous allons développer maintenant.

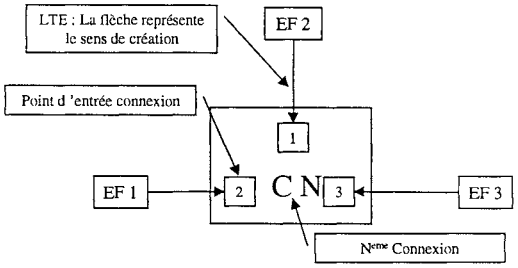



**Figure 3.1 : Représentation des LTE et des connexions**

## I.2. Les connexions

### I.2.1. Définition

Une connexion est un objet destiné à associer plusieurs LTE. Ainsi, via leurs objets Couple_EF, les fonctions entrent en interaction et peuvent échanger de l'énergie entre elles. Contrairement aux graphes de liens, les connexions sont toutes de type effort et vérifient les équations (3.1) et (3.2) d'un point de vue structurel.

Une connexion reliant N Couple_EF établit les relations :

$$\sum_k F_k = 0 \qquad (3.1)$$

$$\forall E_k \quad E_k = E \qquad (3.2)$$

où E est l'effort imposé de la connexion, $E_k$ est l'effort du $k^{ème}$ Couple_EF et $F_k$ est le flux du $k^{ème}$ Couple_EF

Dans une représentation fonctionnelle (donc qualitative), il n'est pas possible de vérifier si ces équations sont cohérentes. Cependant, la viabilité de la connexion peut être vérifiée en transformant ces équations en règles de cohérence.

L'équation ( 3.1 ) correspond aux règles :

**- au moins un flux de ne doit pas être imposé à une connexion**
**-à tout instant, une connexion doit posséder au moins deux flux avec des sens opposés**

Dans le cas des convertisseurs statiques, un LTE arrivant sur une connexion peut avoir N états. Donc, pour une connexion comportant L liens, on obtient un nombre important de combinaisons ($N^L$ états au maximum). Ce processus est très coûteux en temps et peut se remplacer de façon avantageuse par une analyse sur les boucles implicites de flux du graphe comme nous le développerons au § II.5, après une présentation des boucles au § I.3.

## I.3. Les boucles

### I.3.1. Définition

L'une des propriétés importantes des graphes fonctionnels est que le flux doit être bouclé pour autoriser un transfert d'énergie entre deux éléments. La construction d'un convertisseur passe obligatoirement par une phase d'analyse et de définition des différents chemins de transfert d'énergie. Pour faciliter cette tâche, nous proposons de construire des boucles de flux entre les différents objets Couple_EF afin d'établir les transferts d'énergie. A partir de ces boucles, il nous sera possible de mener une analyse structurée pour définir des contraintes fonctionnelles sur des éléments du graphe.
Les boucles présentent deux points de vue complémentaires que nous allons décrire maintenant.

### I.3.2. Le point de vue topologique

Ce point de vue permet de décrire la topologie des boucles afin de la positionner par rapport au graphe fonctionnel en mémorisant le parcours d'un flux au travers des interfaces des fonctions et des connexions qui la composent.
Afin de simplifier la construction du graphe fonctionnel et donc des boucles, nous effectuons les restrictions suivantes.

10. *Un LTE doit toujours appartenir à une boucle.*
11. *Un objet Couple_EF est relié à une et une seule connexion. Cette restriction ramène les problèmes de cohérence uniquement sur les connexions.*
12. *Deux connexions ne peuvent pas être liées ensemble par un LTE. Si le concepteur désire mener une opération de ce type alors les deux connexions seront fusionnées.*
13. *Une boucle ne peut pas relier deux interfaces avec des domaines énergétiques différents.*

Avec ces règles, nous obtenons une structure de graphe fonctionnel très simple, mais qui permet néanmoins de prendre en compte tous les cas. Ainsi, une boucle est toujours construite avec la même structure de base : deux objets Couple_EF de la même interface suivis d'un objet Connexion. La figure 3.2 présente l'exemple d'une boucle simple.

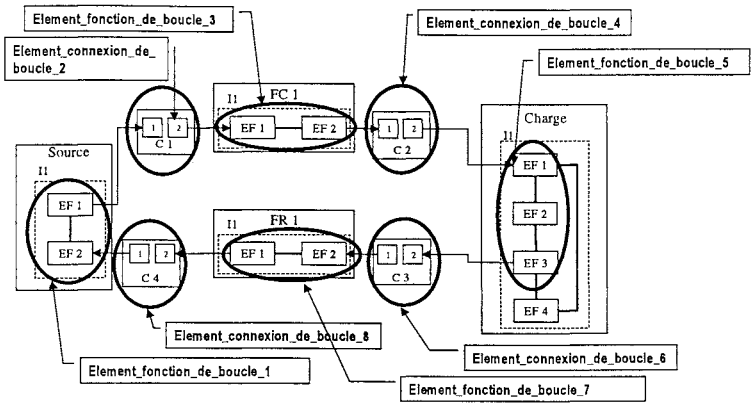



**Figure 3.2 : Une boucle simple de démonstration**

## I.3.3. Le point de vue énergétique

Une fois une boucle fermée, il est nécessaire de vérifier sa viabilité énergétique. Cette vérification de cohérence doit porter sur plusieurs aspects que nous allons présenter maintenant.

### *I.3.3.1. La causalité*

La causalité est vérifiée dans une boucle en appliquant les règles principales suivantes.

14. *Dans une boucle indépendante, une seule fonction peut imposer le flux sinon il est possible d'avoir un conflit de type flux. Cette règle n'est cependant pas applicable dans le cas où cette boucle est insérée dans un graphe fonctionnel plus complet. La figure 3.3 présente un exemple de ces cas de figure.*
15. *La règle précédente s'applique d'une façon similaire pour les risques de conflit potentiel de type dérivée_flux.*
16. *Au moins une fonction ne doit pas imposer l'effort sinon il y a un conflit de type effort.*
17. *Au moins une fonction ne doit pas avoir de contrainte sur la dérivée de l'effort sinon il y un conflit de type dérivée_effort*

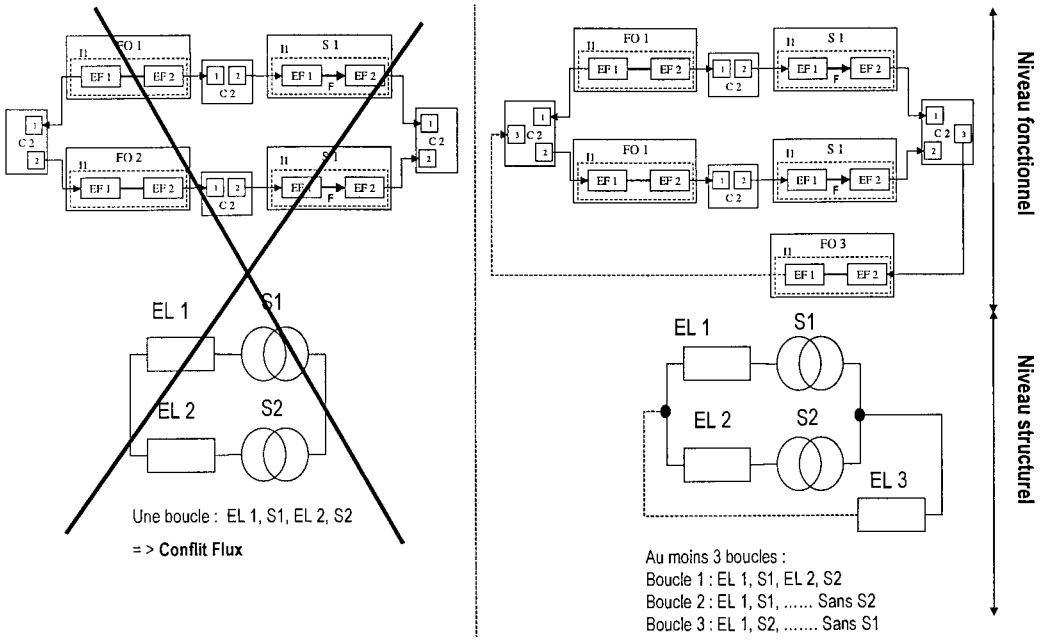



**Figure 3.3 : Un exemple de conflit flux**

Une fois que la boucle a été vérifiée d'un point de vue causalité, il est possible de définir une causalité dominante qui permet de caractériser l'état de cohérence énergétique de la boucle. Celui-ci devra être utilisé au moment de la vérification de cohérence, notamment avec des graphes fonctionnels ayant des fonctions gyrateurs (voir § II.3). Dans les boucles, comme le traitement des flux est plus riche que celui des efforts, nous avons hiérarchisé les causalités : flux, puis dérivée_flux, puis effort, puis dérivée_effort, puis passif. L'organigramme de la figure 3.4 présente la procédure permettant de déterminer la causalité dominante.

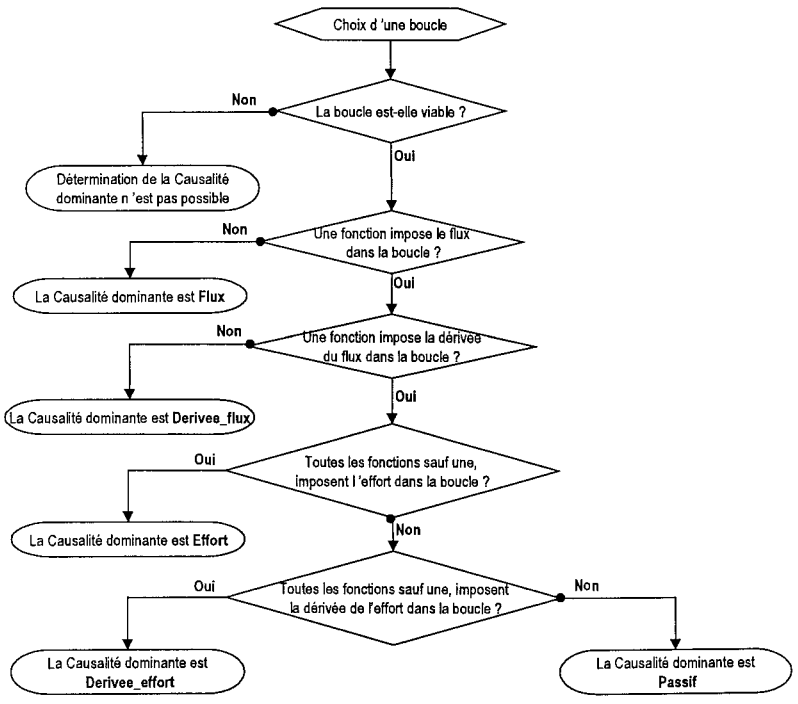



**Figure 3.4 : Organigramme de la causalité dominante**

### *I.3.3.2. Le niveau des flux*

La vérification de la causalité est une première étape. Cependant elle n'est pas suffisante pour s'assurer la cohérence des boucles de flux. Pour garantir le fonctionnement du convertisseur, il est nécessaire de vérifier les règles suivantes.

18. *Toutes les fonctions doivent être compatibles en terme de sens de flux.*
19. *Lorsqu'un niveau de flux est imposé dans une boucle, toutes les fonctions de la boucle doivent pouvoir accepter ce niveau de flux.*
20. *Les contraintes de continuité de flux doivent être vérifiées*

Pour gérer tous les aspects portant sur les niveaux de flux, nous proposons de construire des états pour les boucles. Un tel état est une photographie pour un niveau de flux donné et peut être valable pendant une certaine durée. Grâce à la description qualitative des formes d'onde et des états de contraintes, il est possible de mener ce type de raisonnement en utilisant les règles suivantes.

21. *Si une fonction FA présente une causalité flux dans une boucle, alors cette dernière a une forme_d_onde_flux construite avec les Niveaux_flux des objets Etat_de_contrainte de FA.*
22. *Si aucune fonction n'impose le flux, alors les Niveaux_flux de la boucle sont indéterminés. L'expert doit alors choisir la fonction qui détermine le Niveau_flux de la boucle.*
23. *Lorsqu'une fonction de contrôle de flux est dans une boucle alors elle impose un niveau de flux nul lorsqu'elle interdit le passage d'un flux.*

Pour illustrer l'intérêt des états, prenons l'exemple de la boucle dont la topologie est décrite par la figure 3.5.

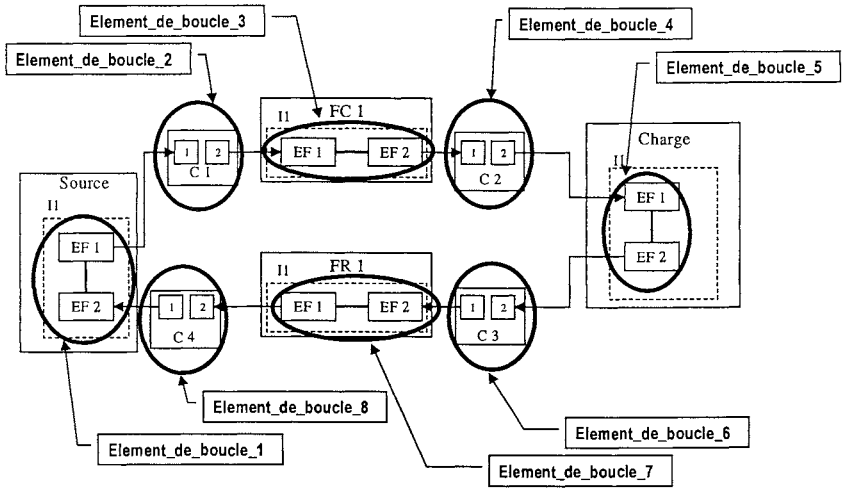



**Figure 3.5 :La vérification des flux dans une boucle**

Les objets Forme_d_onde attachés aux interfaces de la fonction Source (avec une causalité de type effort) et la fonction Charge (avec une causalité de type flux) sont décrites par la figure 3.6. Nous ne décrivons pas ici la fonction de contrôle FR1 parce qu'elle a un comportement identique à la fonction FC1.

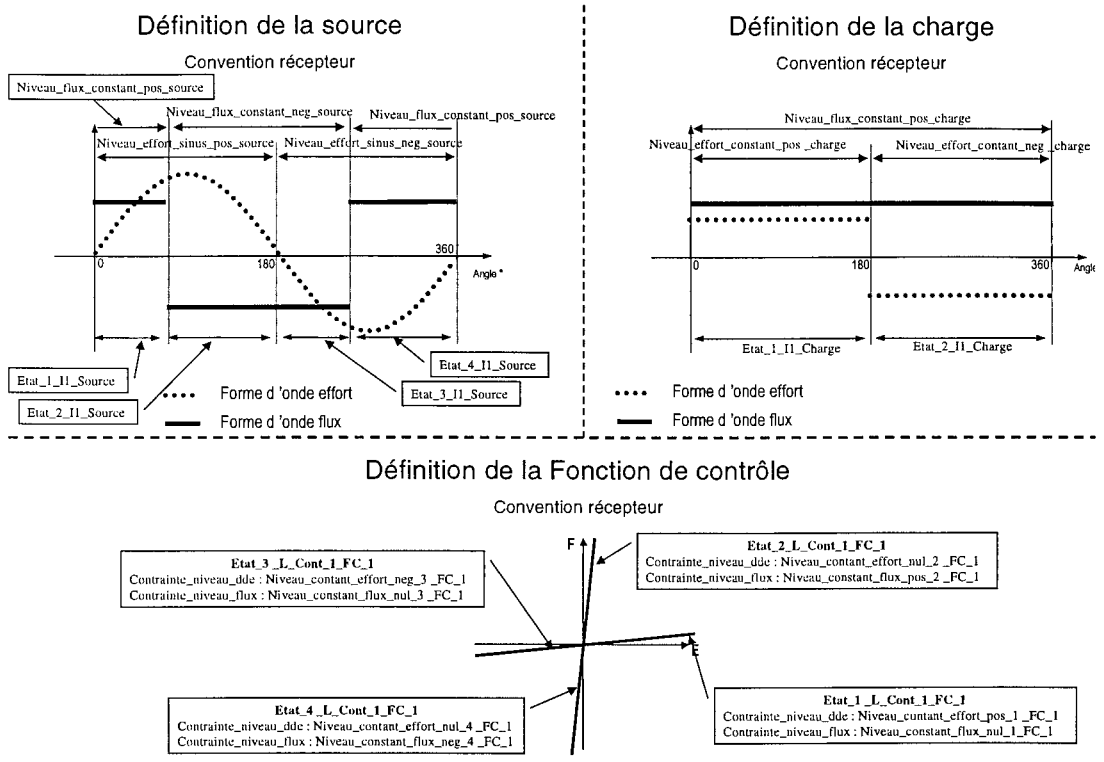



**Figure 3.6 : Les descriptions graphiques des objets Formes_d_onde**

Pour vérifier que les fonctions sont compatibles entre elles, il est nécessaire de vérifier que le niveau de flux imposé par la fonction charge est supporté par toutes les fonctions de la boucle. Par exemple, le tableau 3.2 présente les états de boucles possibles pour notre exemple.

| **Etat de boucle** | 1 | 2 | 3 |
|---|---|---|---|
| **Nom de l'objet informatique** | Etat_de_boucle_1 | Etat_de_boucle_2 | Etat_de_boucle_3 |
| **Niveau de flux imposé dans la boucle** | Niveau_flux_constant_pos | Niveau_constant_flux _nul_1_FC_1 | Niveau_constant_flux _nul_2_FC_1 |
| **Etat_de_contrainte compatible de la fonction Source** | Etat_1_I1_Source<br>Etat_4_I1_Source | Etat_1_I1_Source<br>Etat_1_I2_Source<br>Etat_1_I3_Source<br>Etat_1_I4_Source | Etat_1_I1_Source<br>Etat_1_I2_Source<br>Etat_1_I3_Source<br>Etat_1_I4_Source |
| **Etat_de_contrainte compatible de la fonction Charge** | Etat_1_I1_Charge<br>Etat_2_I2_Charge | | |

**Tableau 3.2 : Les états de boucle possibles.**

Ce tableau est construit en utilisant la table de vérité partielle présentée dans le tableau 3.3.

| | | | Niveau de flux de l'Etat_de_boucle | |
|---|---|---|---|---|
| | | | Nul | Positif (pos) |
| Niveau de flux de l'Etat_de_contrainte | Nul | Imposé | Compatible | **Non Compatible** |
| | | Libre | Compatible | Compatible sous certaines conditions |
| | Positif (pos) | Imposé | **Non Compatible** | Compatible si les objets Niveau sont les mêmes |
| | | Libre | Compatible | Compatible |

**Tableau 3.3 : La table de vérité**

Afin de déterminer si l'objet Etat_de_boucle est réellement viable, il est nécessaire de faire une analyse complémentaire sur les efforts. Cependant, pour que cette analyse soit possible, il faut mettre en place une arithmétique entre les différents objets Niveaux afin de pouvoir déterminer les combinaisons de flux et les combinaisons d'effort possibles. Par exemple, nous devrons être capables de déterminer quand la fonction Source impose un niveau effort sinusoïdal positif sur la fonction charge pendant l'état de boucle 1. Cette possibilité n'est pas offerte pour le moment dans notre prototype d'outil, mais notre modèle objet a été conçu pour permettre son implémentation.

Ainsi, les boucles sont modélisées suivant un réseau dont les nœuds sont des objets (cf. annexe IV). A titre d'exemple, la figure 3.7 illustre cette représentation informatique dans le cas de la boucle de la figure 3.5.

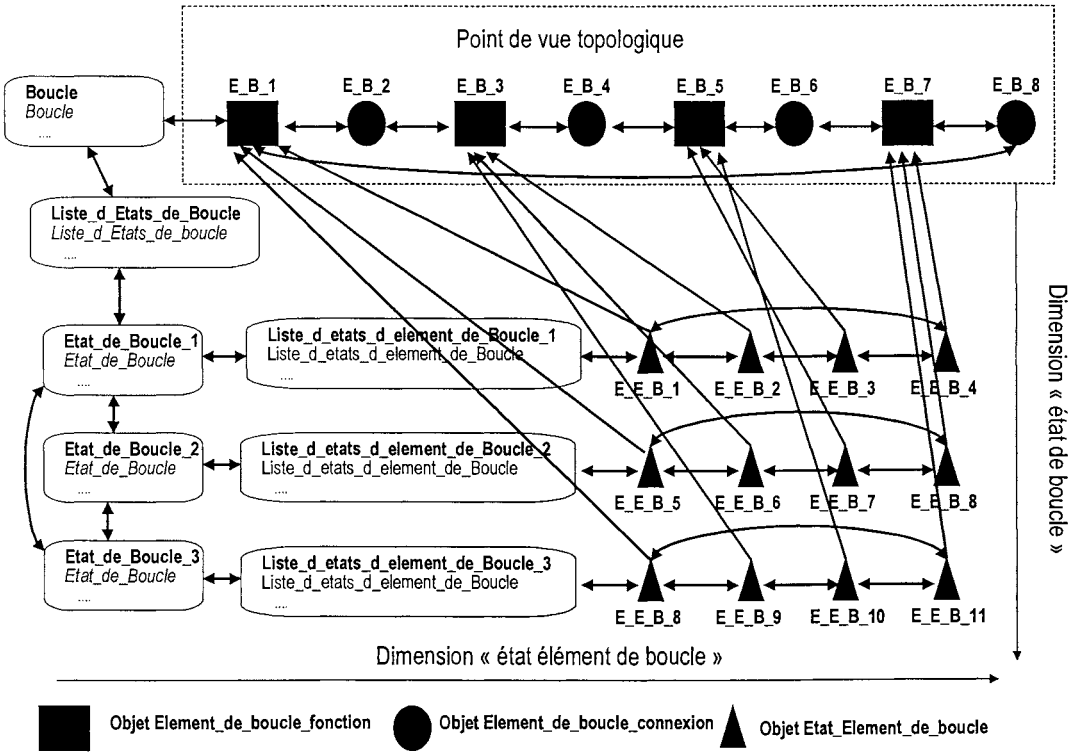



**Figure 3.7 : La modélisation objet des boucles**

## I.4. Les Liens temporels

### I.4.1. Présentation

Nous avons déjà défini les contraintes fonctionnelles énergétiques, qui sont très utiles pour la vérification de la cohérence. Il y a plusieurs façons de résoudre les problèmes de cohérence.

- **En modifiant la topologie du graphe fonctionnel.** C'est une opération qui doit être faite au cas par cas par le concepteur.

- **En plaçant des contraintes sur les états des fonctions.** Ces dernières doivent être contrôlables (donc du type contrôle de flux) pour que ces actions soient réalisables. Les contraintes se répercutent sur la définition de leur commande. L'intérêt de cette solution est qu'elle ne provoque pas de modification de la topologie et l'analyse peut être automatisée facilement.

C'est dans cette optique que nous avons défini les liens fonctionnels temporels. Ils établissent des contraintes sur les différentes parties du graphe fonctionnel. Il en existe trois types qui sont présentés dans le paragraphe suivant.

### I.4.2. Les différents liens temporels

#### *I.4.2.1. Le Lien de Simultanéité (LS).*

Le lien de simultanéité indique que les deux éléments liés doivent autoriser le passage d'un flux en même temps. Le tableau 3.4 présente les différentes possibilités ainsi que leur sémantique particulière.

| Type de lien | Eléments liés | Sémantique |
|---|---|---|
| **LS simple** | Un nombre quelconque de fonctions | Les fonctions liées doivent être actives en même temps |
| **LS sur une boucle** | **Toutes les fonctions** de contrôle et de retour de la boucle | Pour que la boucle établisse un flux, il faut que les fonctions liées soient toutes à l'état passant (établissant ou autorisant un flux) |
| **LS entre boucle** | **Toutes les fonctions** de contrôle et de retour des boucles | Les boucles doivent établir un flux en même temps. Celles-ci doivent avoir chacune un LS sur une boucle. |

**Tableau 3.4 :Les différents types de lien de simultanéité**

La figure 3.8 présente la symbolisation des LS dans les schémas fonctionnels

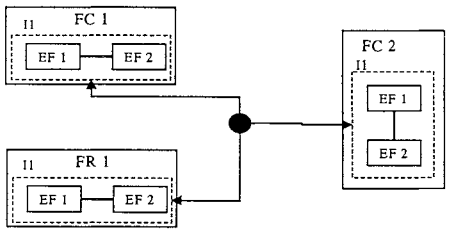



**Figure 3.8 : La symbolisation d'un LS**

### *I.4.2.2. Le Lien de Non-Simultanéité (LNS)*

Le lien de non-simultanéité est le lien opposé aux LS. Il signifie donc que les éléments liés ne doivent pas autoriser un flux en même temps. Ainsi, le nombre de fonctions ne devant pas être dans l'état où le flux est nul en même temps est variable. Le tableau 3.5 présent les types de LNS

| Type de lien | Eléments liés | Sémantique |
|---|---|---|
| **LNS simple** | Un nombre quelconque de fonctions | Les fonctions liées ne doivent pas établir un flux en même temps |
| **LNS sur une boucle** | **Au moins deux fonctions** de contrôle ou de retour de la boucle (en raison de leur rôle interchangeable) | Pour que la boucle ne puisse pas établir un flux il suffit qu'une fonction liée soit à l'état bloqué (interdisant le passage d'un flux). |
| **LNS entre boucle** | **Au moins une fonction** de contrôle ou de retour de chaque boucle | Les boucles ne doivent pas établir un flux en même temps. |

**Tableau 3.5 : Les différents types de lien de non simultanéité**

Un problème de compatibilité entre les deux types de liens se pose. Ainsi, nous utilisons la convention suivante : **les LNS sont prédominants sur les LS.** Ce choix est justifié par le fait que les LNS soient utilisés pour résoudre des conflits potentiels dans le graphe fonctionnel.
La figure 3.9 donne un exemple de représentation des LNS.

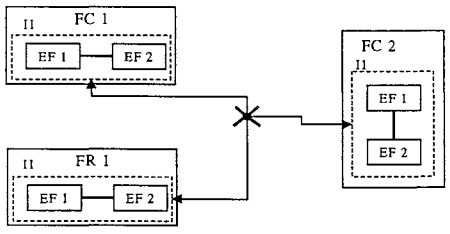



**Figure 3.9 : La symbolisation d'un LNS**

### *I.4.2.3. Le Lien Séquentiel (LSeq).*

Le troisième lien est le lien Séquentiel. Il permet de relier les éléments qui doivent fonctionner de façon séquentielle. Par exemple, ces liens permettent de résoudre les

problèmes de continuité de flux (cf. exemple du Chapitre 5). Le tableau 3.6 présente les deux types de LSeq possibles.

| Type de lien | Eléments liés | Sémantique |
|---|---|---|
| **LSeq simple** | Un nombre quelconque de fonctions | Les fonctions liées doivent établir un flux de façon séquentielle |
| **LSeq entre boucles** | Toutes les fonctions de contrôle et de retour de chaque boucle (en raison de leur rôle interchangeable) | Les boucles doivent établir un flux de façon séquentielle. |

**Tableau 3.6 : Les différents types de lien séquentiels**

Les LSeq sont illustrés par la figure 3.10.

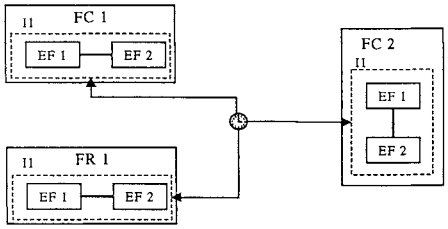



**Figure 3.10 : La symbolisation des LSeq**

Nous venons de présenter tous les éléments du graphe fonctionnel. Dans le paragraphe suivant, nous présentons comment nous gérons ces éléments dans leur cycle de vie (création, vérification, spécification).

# II. Le graphe fonctionnel

Le graphe fonctionnel permet de décrire complètement le convertisseur du point de vue fonctionnel, topologique et énergétique. Outre sa construction en étapes successives (en ajoutant des fonctions, des boucles du flux ou des liens), il constitue le cœur du processus de conception utilisant l'approche fonctionnelle.

Ce paragraphe présente donc la modélisation du graphe fonctionnel avec ses constituants, puis une méthode possible de construction, et enfin les procédures automatiques que nous avons mises en place pour la vérification de la cohérence.

## II.1. La modélisation d'un graphe fonctionnel

### II.1.1. Composition

Un graphe fonctionnel est composé des éléments suivants.

- **Un ensemble de fonctions.** Celles-ci sont définies par l'expert. C'est une liste qui représente la décomposition en fonctions du graphe.
- **Un ensemble de boucles de flux.** Celles-ci permettent aux fonctions d'échanger de l'énergie via leurs interfaces et un réseau de liens énergétiques (c'est-à-dire LTE). Elles sont définies par l'expert en fonction des mouvements d'énergie qu'il veut définir entre les éléments. Nous les appellerons les boucles principales.
- **Un ensemble de contraintes énergétiques.** Celles-ci sont attachées aux interfaces et qui permettent la vérification de la cohérence énergétique du graphe.
- **Un ensemble de liens temporels.** Ceux-ci permettent d'éviter les incohérences et de gérer l'évolution temporelle du graphe.
- **Un ensemble d'états.** Ceux-ci permettent de définir les phases de transfert d'énergie dans les graphes. Un cycle de fonctionnement (aussi appelé mode) d'un convertisseur sera modélisé avec une approche fonctionnelle comme une liste cyclique d'états du graphe. Ces derniers sont définis à partir d'un état de chaque boucle principale (faite par le concepteur).

### II.1.2. Intérêt de la décomposition du graphe en boucles

Il est très avantageux de décrire un graphe par un ensemble de boucles pour diverses raisons.

- **Cette modélisation est très représentative du fonctionnement des convertisseurs statiques.** En effet, tous les transferts d'énergie sont régis par la circulation de flux dans des boucles.
- **Elle permet de définir simplement un état du graphe fonctionnel.** Celui-ci est un ensemble d'objets Etat_de_boucle.
- **Elle permet une approche plus globale de la gestion de cohérence dans le graphe.** En effet, il suffit de vérifier toutes les boucles où il circule un flux pour être assuré que le graphe est cohérent.

L'exemple de la figure 3.11 illustre l'utilisation de la description en boucles d'un circuit électrique. Les conclusions obtenues sont facilement transposables à un graphe fonctionnel. Ici les connexions C 1 et C2 peuvent être en conflit potentiel suivant l'état des quatre

interrupteurs. Avec une analyse des connexions, il faut considérer seize états pour chaque connexion et pour chaque état, il est nécessaire d'analyser le circuit pour déterminer les contraintes sur les flux. Ce processus peut s'avérer long si le graphe est compliqué.

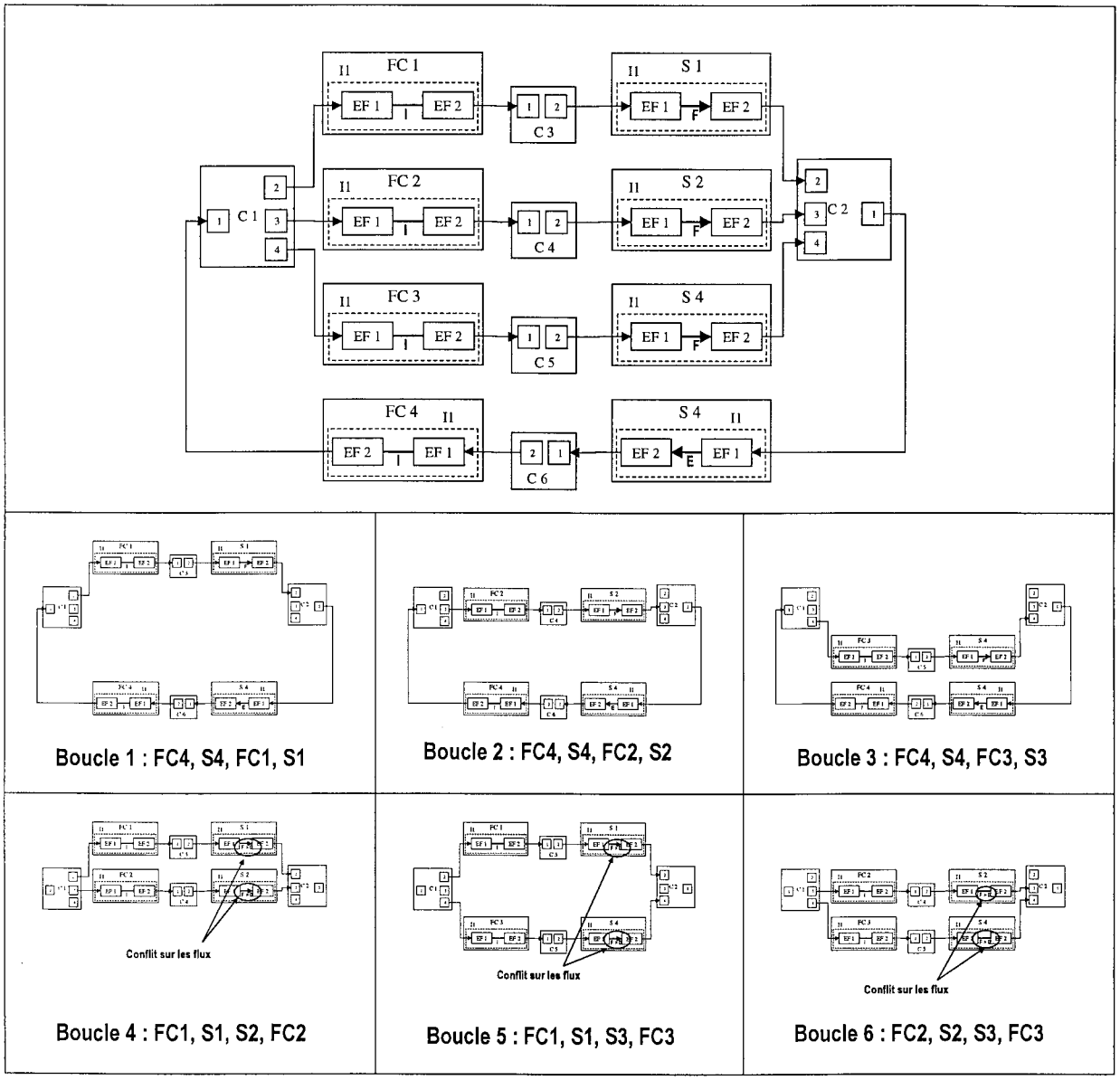



**Figure 3.11 : Un exemple de circuit conflictuel**

Avec une description en boucles, il suffit d'analyser la causalité (donc d'appliquer un ensemble de règles) sur les six boucles de flux possibles dans le circuit pour en déduire les incohérences dans le circuit (en termes de boucles de flux et de connexion du circuit).

## II.2. La méthodologie de construction d'un graphe fonctionnel

Avec une approche fonctionnelle, il est nécessaire d'utiliser une démarche différente de la conception structurelle. Le processus de conception d'un convertisseur peut être par exemple décrit par l'organigramme de la figure 3.12.

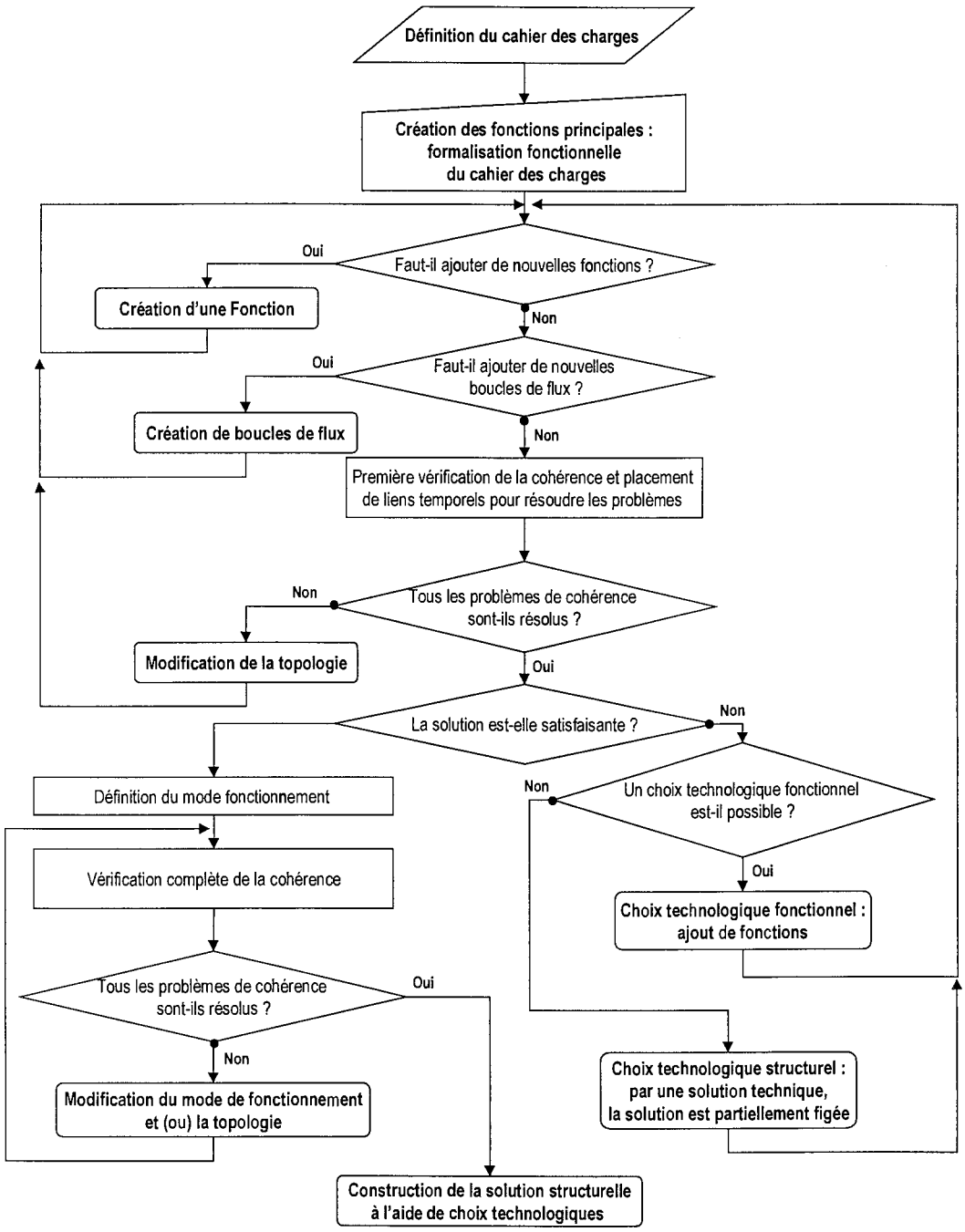



**Figure 3.12 : Une démarche possible de conception fonctionnelle**

L'expert est le seul maître pour définir les fonctions et construire les boucles de flux afin de permettre un transfert d'énergie entre les fonctions. Cependant, il est nécessaire d'effectuer une vérification de cohérence en terme de flux et de causalité pour vérifier la compatibilité en terme énergétique. De cette analyse, il est alors possible d'extraire des liens fonctionnels pour supprimer des états interdits du graphe.

En cas de blocage du processus de conception, l'expert doit effectuer des choix technologiques pour débloquer la situation afin d'arriver à un graphe fonctionnel satisfaisant, c'est-à-dire où il ne reste plus que des choix de solutions techniques à faire.

L'étape finale est le passage de la solution fonctionnelle à la solution structurelle, c'est-à-dire la spécification des fonctions afin de préparer le dimensionnement des composants.

## II.3. La construction des boucles

### II.3.1. Présentation

Après avoir défini puis instancié les fonctions pour construire son graphe fonctionnel, l'expert doit définir les boucles de flux afin de caractériser les transferts d'énergie. Celles-ci seront mémorisées comme des boucles principales et vont servir de base pour l'analyse de cohérence. Afin de simplifier la gestion du graphe, notre système assume les restrictions suivantes.

- **Les connexions doivent être créées automatiquement.** Cela assure une cohérence topologique du graphe : un Couple_EF est relié à une seule connexion ce qui ramène tous les problèmes de connexion de flux au niveau des objets connexions mêmes.

- **Les fonctions Gyrateur sont traitées de façon particulière.** En effet, lorsqu'une boucle comporte des éléments de ce type, l'analyse de causalité se trouve complexifiée. Plusieurs cas de figure peuvent se produire.

  - La boucle ne comporte qu'un élément gyrateur et présente une causalité dominante flux, alors le flux est imposé dans la boucle.

  - La boucle ne comporte qu'un seul élément gyrateur et toutes les autres fonctions ont une causalité effort alors le gyrateur subit l'effort de la boucle.

  - La boucle ne comporte qu'un seul élément gyrateur et une fonction avec une causalité passive, alors la causalité est indéterminée sur le gyrateur

  - La boucle comporte plusieurs éléments gyrateurs, alors il faut parcourir le graphe pour trouver les boucles dont on connaît les causalités et les propager. Pour simplifier cette opération, les interfaces des fonctions Gyrateur ne peuvent appartenir qu'à une seule boucle. Cette simplification évite d'avoir à choisir entre plusieurs boucles pour effectuer la propagation des contraintes. Cette restriction peut se contourner facilement en dupliquant la fonction gyrateur et en connectant les fonctions comme le montre la figure 3.13.

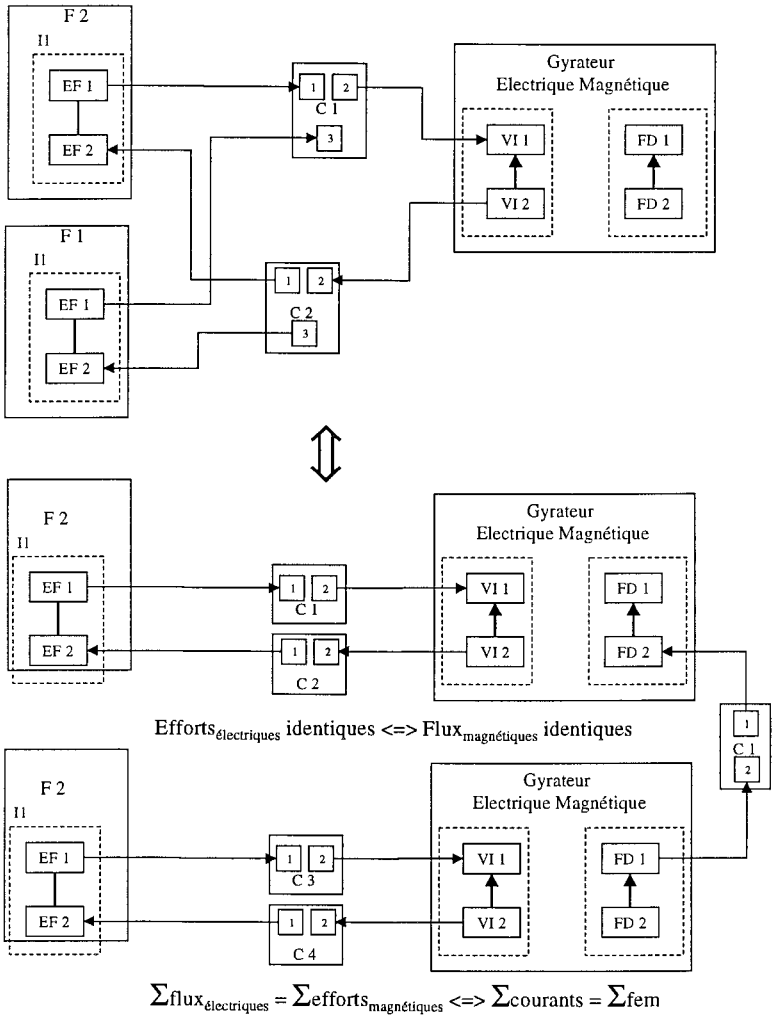



**Figure 3.13 : Equivalences dans la connexion des gyrateurs**

## II.3.2. Principe

La procédure de création des boucles est résumée par l'organigramme de la figure 3.14. Cette procédure possède plusieurs parties.

- **Une partie interaction avec l'expert (zone 1).** Là, l'expert doit choisir les Couple_EFs à ajouter à la boucle. Cette interaction peut être menée via une interface graphique.

- **Une partie chargée de construire la topologique de la boucle (zone 2).**

- **Une partie chargée de contrôler les choix de l'expert (zone 3).** Cela sert pour respecter les restrictions sur la construction des boucles présentée au paragraphe précédent.

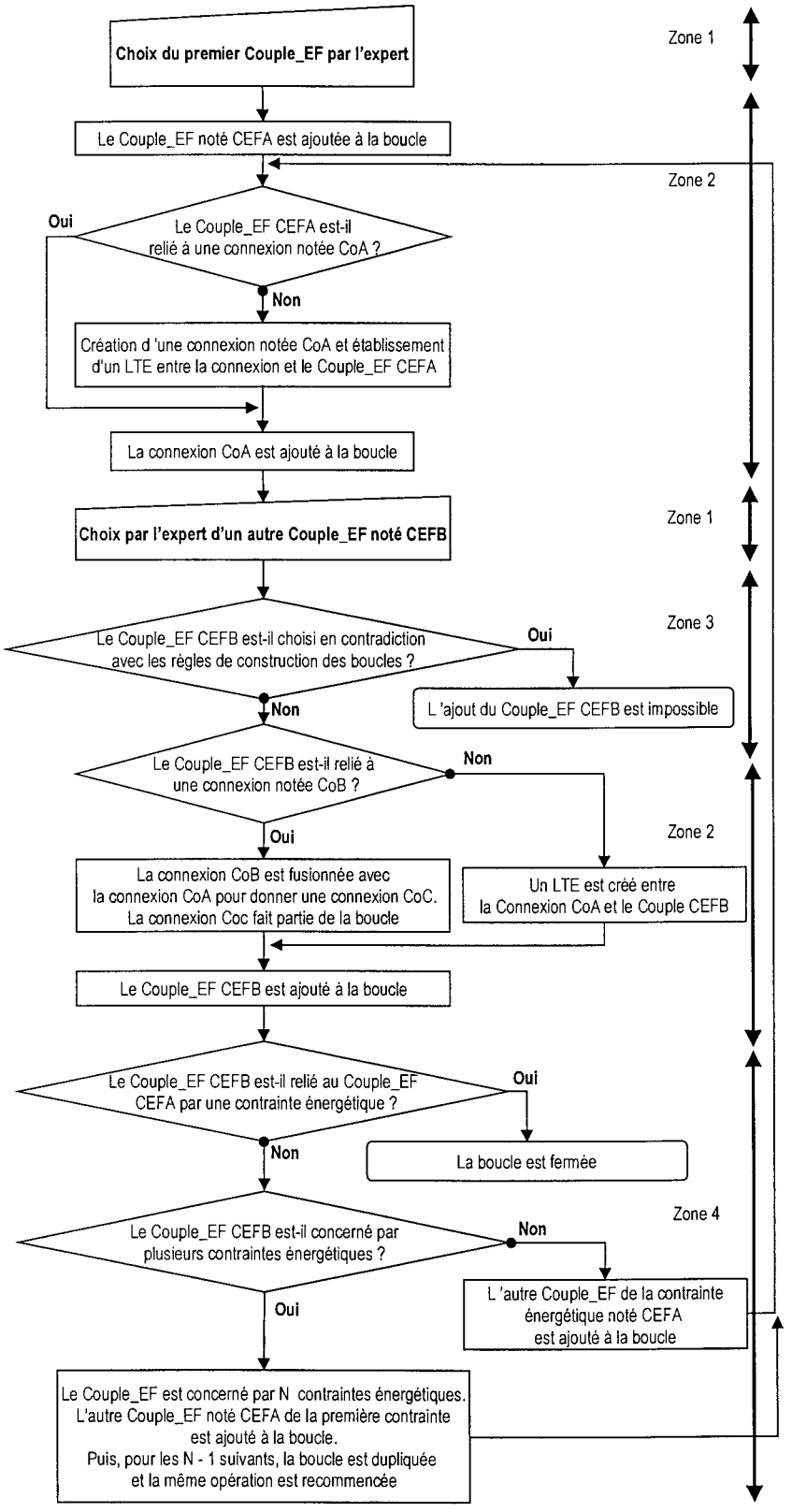



**Figure 3.14 :Organigramme de la procédure de création des boucles**

- **Une partie chargée de trouver les suites possibles à la boucle (zone 4).** En effet lorsqu'une interface comporte plus que deux objets Couple_EF, il y a plusieurs chemins possibles continuer la construction de la boucle dans cette même interface.

Dans ce cas, nous proposons par convention, de créer une boucle pour chaque chemin possible du flux. La figure 3.15 présente un exemple de l'application de cette règle.

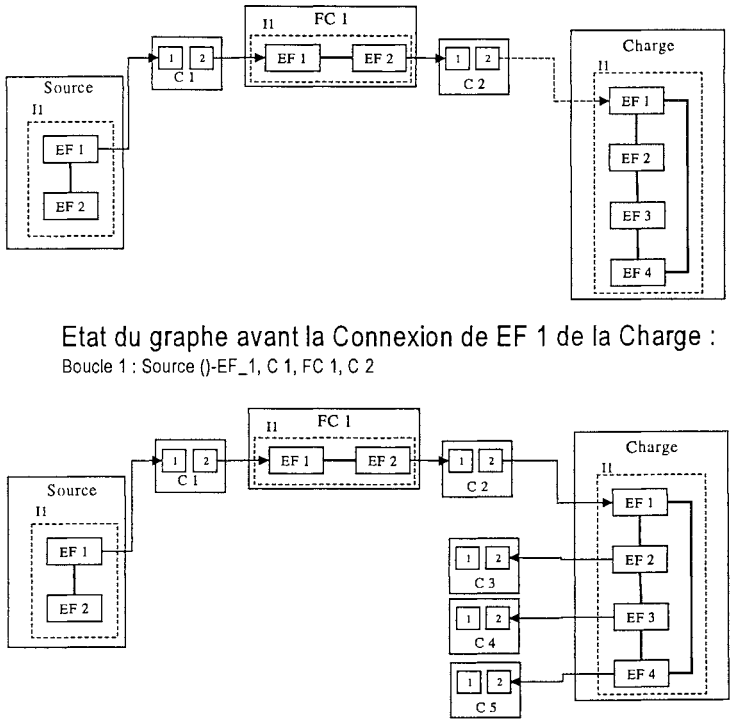
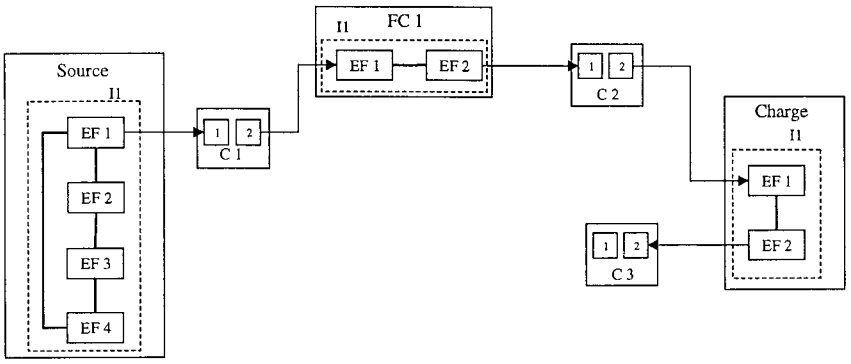



**Figure 3.15 : Fonctionnement de la procédure de construction des boucles**

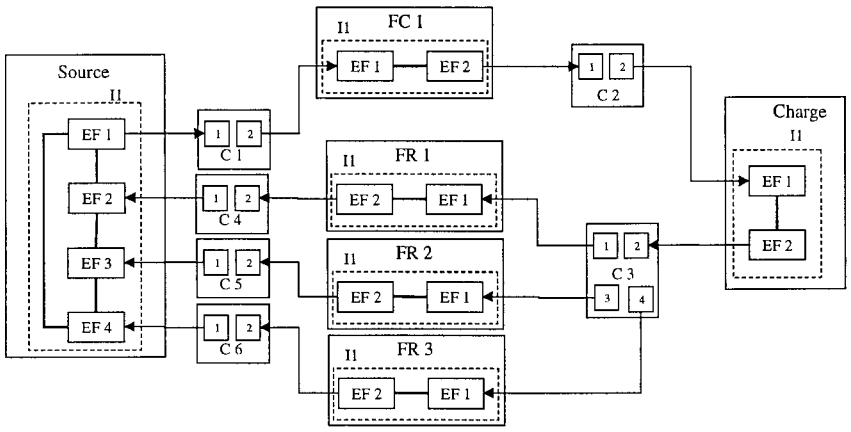
En complément de la procédure de base, l'expert doit choisir d'appliquer les deux règles suivantes pour accélérer la construction des boucles.

24. *L'expert peut ajouter un lien énergétique avec contrôle, c'est-à-dire un LTE et une fonction de contrôle du flux..*
25. *L'expert peut décider de fermer la boucle par tous les chemins possibles. Les boucles sont alors fermées en ajoutant des fonctions de retour. La figure 3.16 présente un exemple pour illustrer cet aspect.*




Etat du graphe avant la Fermeture automatique de la boucle :

Boucle 1 : Source ()-EF_1, C 1, FC 1, C 2, Charge EF_1-EF_2, C3




Etat du graphe après :

Boucle 1 : Source EF_2-EF_1, C 1, FC 1, C 2, Charge EF_1-EF_2, C3, FR 1 EF_1-EF_2, C4

Boucle 2 : Source EF_3-EF_2-EF_1, C 1, FC 1, C 2, Charge EF_1-EF_2, C3, FR 2 EF_1-EF_2, C5

Boucle 3 : Source EF_4-EF_1, C 1, FC 1, C 2, Charge EF_1-EF_2, C3, FR 3 EF_1-EF_2, C6

**Figure 3.16 : Un exemple de la fermeture automatique d'une boucle**

## II.4. La Vérification de la cohérence énergétique

Cette étape est fondamentale dans l'approche fonctionnelle. En effet, il est très important de vérifier que les flux d'énergie n'entrent pas en conflit dans tout le graphe. En cas de conflit, les problèmes énergétiques provoqueront une destruction certaine de tout ou partie du dispositif.

Nous proposons ici une démarche (menée automatiquement par l'ordinateur) qui permet de détecter et de résoudre en partie ces incohérences énergétiques. Les paragraphes suivants présentent dans l'ordre les différentes étapes de ce processus.

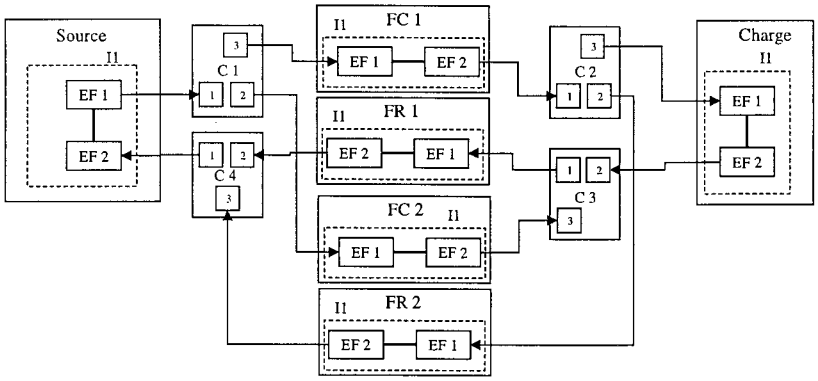
### II.4.1. Construction des boucles implicites

#### *II.4.1.1. Présentation des boucles implicites*

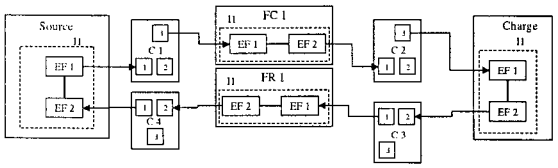
Nous avons vu dans les paragraphes précédents, qu'un graphe fonctionnel est constitué d'un ensemble de boucles construites par le concepteur. Cependant, les interconnexions de ces

boucles créent de nouvelles boucles qui sont souvent indésirables. Afin d'illustrer ce problème, prenons l'exemple de la figure 3.17.

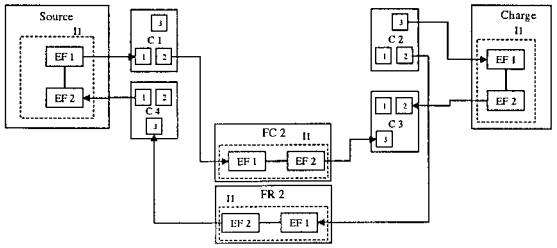
**Graphe fonctionnel étudié**




**Boucles construites par l'utilisateur**

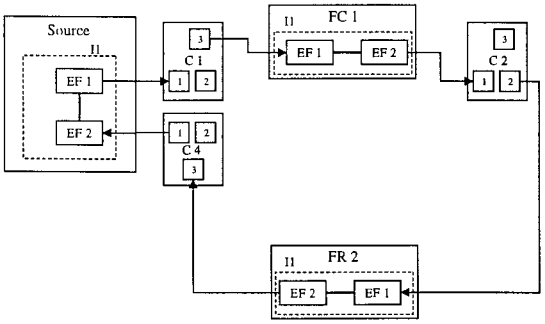


Boucle 1 : Source EF_2-EF_1, C 1, FC 1 EF_1-EF_2, C 2, Charge EF_1-EF_2, C 3, FR 1 EF_1-EF_2, C 4

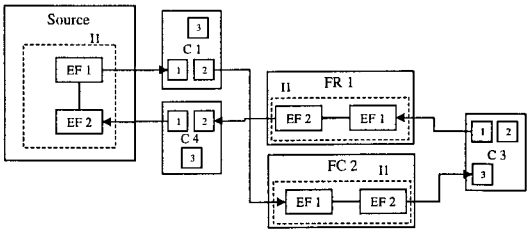


Boucle 2 : Source EF_2-EF_1, C 1, FC 2 EF_1-EF_2, C 3, Charge EF_2-EF_1, C 2, FR 2 EF_1-EF_2, C 4

**Boucles implicites du graphe**



Boucle_I1 : Source EF_2-EF_1, C 1, FC 1 EF_1-EF_2, C 2, FR 2 EF_1-EF_2, C 4



Boucle_I2 : Source EF_2-EF_1, C 1, FC 2 EF_1-EF_2, C3, FR 1 EF_1-EF_2, C 4

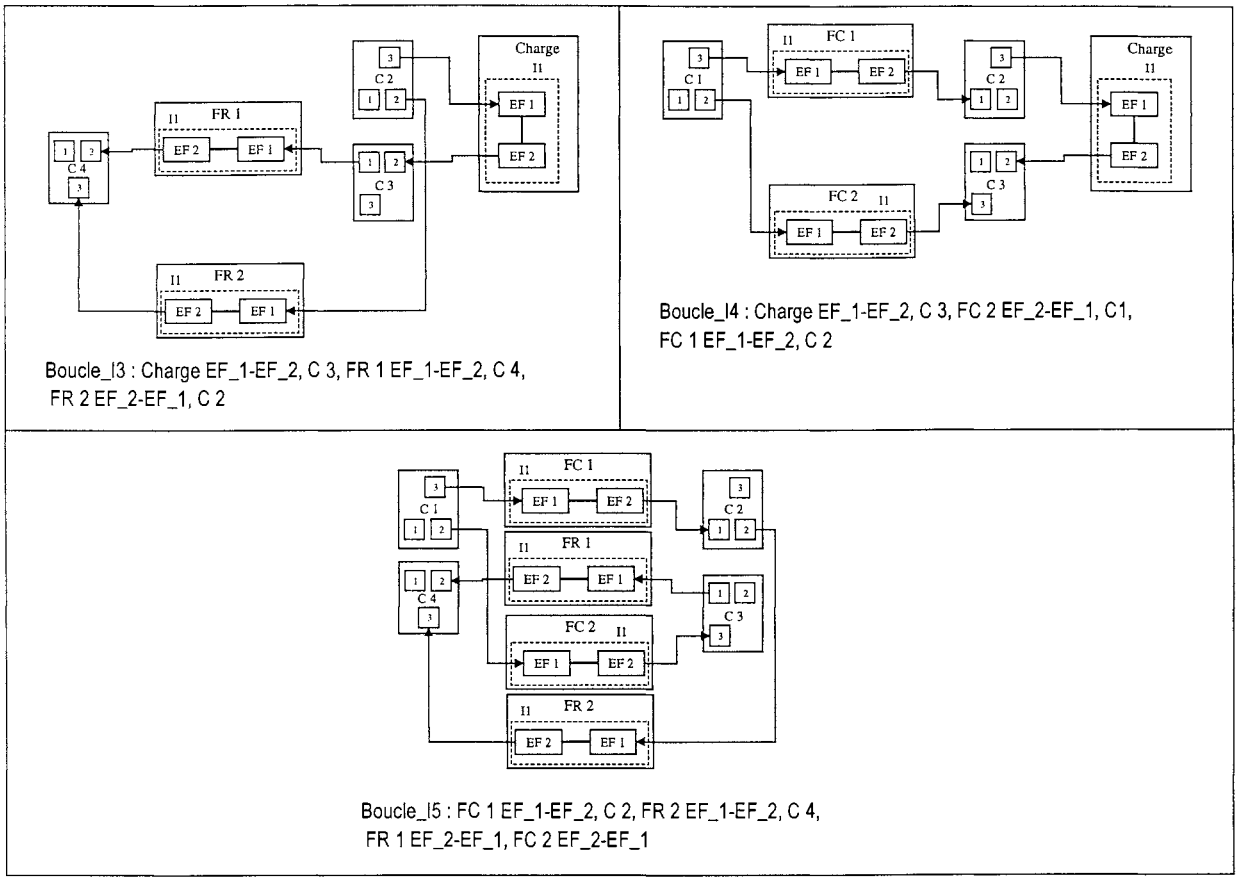



**Figure 3.17 : Un exemple de boucles implicites**

En analysant les boucles créées, il est possible de faire différentes constatations, dont les deux suivantes.

- Les boucles Boucle_I1 et Boucle_I2 peuvent connecter en direct les deux objets Couple_EF de la fonction source, ce qui revient à annuler la différence d'effort entre eux. Si cette fonction présente une causalité Effort (c'est-à-dire impose l'effort à une valeur non nulle), le flux ne doit pas circuler par ces chemins.

- Les boucles Boucle_I3 et Boucle_I4 peuvent créer une annulation de la différence d'effort sur la fonction charge. Si celle-ci possède une contrainte de flux continue, alors ces voies de passage de flux peuvent servir à garantir que cette condition n'est pas violée.

Les boucles implicites ne sont donc pas toujours nocives. Elles peuvent aussi être utilisées pour fournir des alternatives à l'expert dans une optique de résolution de problèmes ou de réduction du nombre de fonctions (et donc du nombre de composants).

### *II.4.1.2. Principe de la recherche des boucles implicites*

La recherche des boucles implicites est basée sur une méthode d'analyse de graphe. En effet, notre représentation fonctionnelle peut être traduite en un réseau plus simple constitué de nœuds et d'arcs non orientés en effectuant les simplifications suivantes.

- **Un LTE est un arc du nouveau graphe.** C'est un lien non orienté puisque le flux peut circuler dans n'importe quel sens.

- **Une connexion est un nœud du graphe.**

- **Une interface est un sous-réseau où les nœuds sont les objets Couple_EF et les contraintes énergétiques les arcs.**

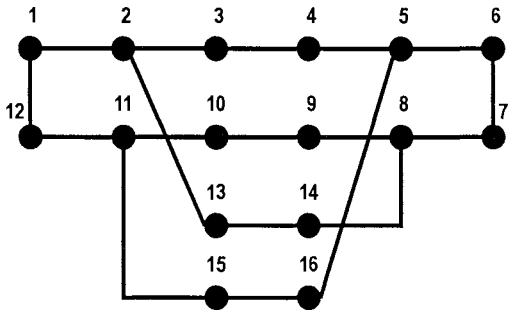



**Figure 3.18 : Le réseau de travail correspondant à la figure 3.17**

L'algorithme de recherche des boucles implicites est décrit par la figure 3.19. Il est construit à partir d'une procédure d'exploration en profondeur du réseau [PRI94].

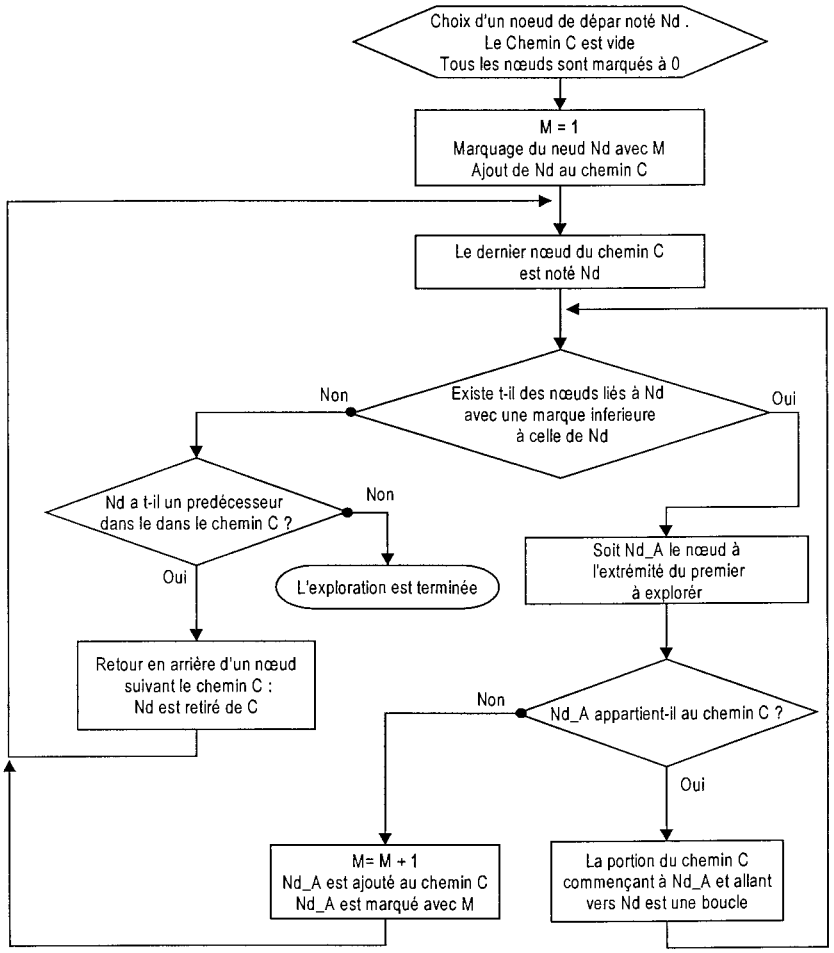



**Figure 3.19 : Le principe de la recherche des boucles indépendantes du réseau**

Dans notre cas, l'organigramme se déplace de nœud en nœud en marquant de façon judicieuse son chemin pour trouver une boucle. Ainsi, il remonte le chemin jusqu'à trouver une nouvelle branche. La figure 3.20 présente son fonctionnement dans le cas du réseau de la figure 3.18.

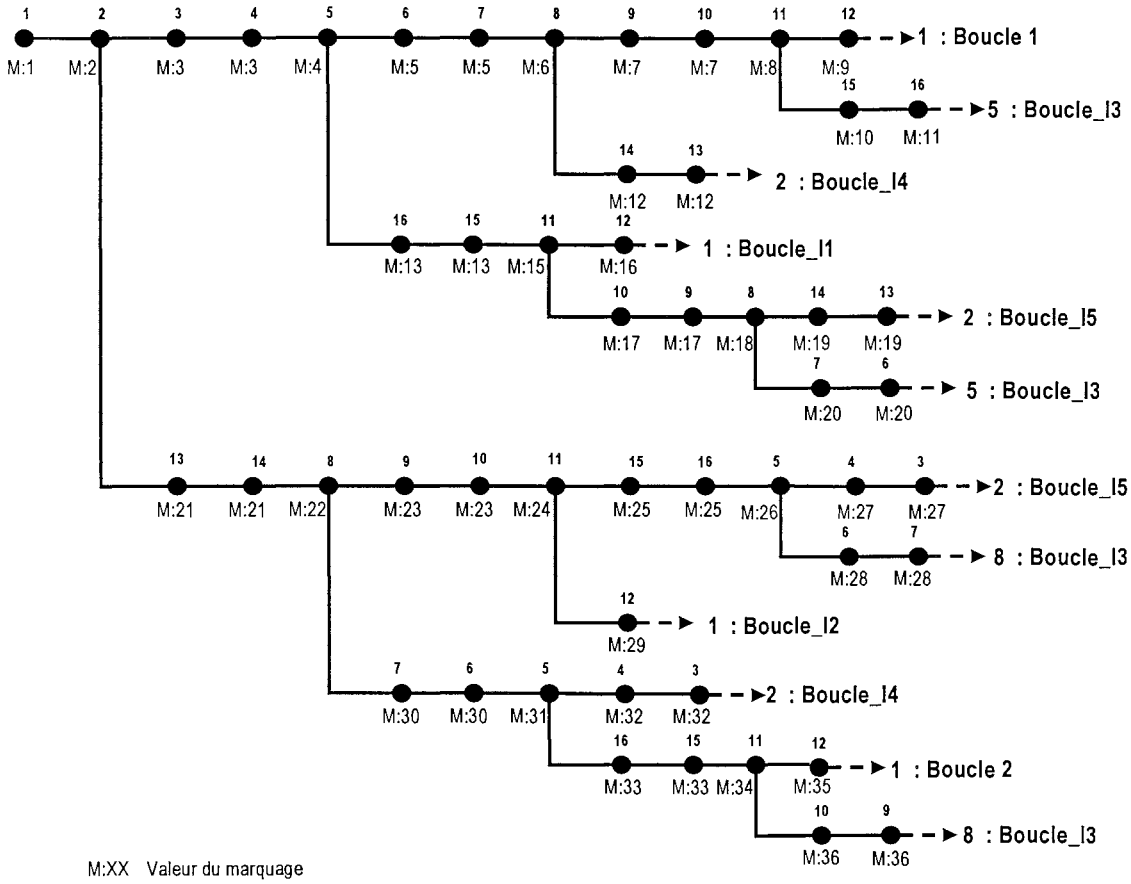



**Figure 3.20 :L'exploration des boucles du réseau de la figure 3.18**

## II.4.2. Les vérifications sans connaître le mode de fonctionnement

Lors de la première phase d'analyse, toutes les boucles sont vérifiées sur des critères de causalité et de flux en appliquant les règles de cohérence présentées au paragraphe I.3.3. L'algorithme de vérification de la cohérence présente plusieurs phases que nous allons détailler maintenant.

- **Phase 1. Vérification des boucles principales avec au plus un gyrateur en terme de causalité et de sens du flux.** Cette phase vérifie que l'expert n'a pas introduit d'incohérences avec les boucles qu'il a créées. Le comportement à tenir suivant le résultat de l'analyse est régi par les règles suivantes.

26. *Si une boucle principale est en conflit, alors il appartient à l'expert de modifier la topologie pour résoudre le problème.*
27. *Si une boucle principale ne comporte pas d'incohérence, alors il faut placer un LS sur celle-ci pour indiquer les fonctions de contrôle de flux à activer pour autoriser la circulation d'un flux.*

- **Phase 2. Vérification des boucles implicites avec au plus un gyrateur en terme de causalité et de sens du flux.** Dans un second temps, il est nécessaire de détecter les objets Lien_de_contrainte en conflit éventuel, en appliquant les règles suivantes.

28. *Si une boucle implicite est en conflit, alors il faut placer un LNS sur celle-ci pour éviter que les fonctions de contrôle du flux autorisent un flux en même temps.*
29. *Si une boucle implicite ne comporte pas d'incohérence énergétique, alors aucun lien temporel n'est placé sur elle mais elle pourra être utilisée pour résoudre un problème de continuité éventuelle du flux.*

- **Phase 3 : Vérification des contraintes de continuité.** Les règles de continuité sur les flux doivent être vérifiées pour éviter les décharges violentes d'énergie dans le système. Par exemple, dans le cas d'un convertisseur statique, elles peuvent provoquer des surtensions. Le processus de vérification de la continuité est résumé par l'organigramme de la figure 3.21. Il permet de déterminer les problèmes potentiels de continuité et de fournir les liens temporels (LSeq) qui permettent de les résoudre.

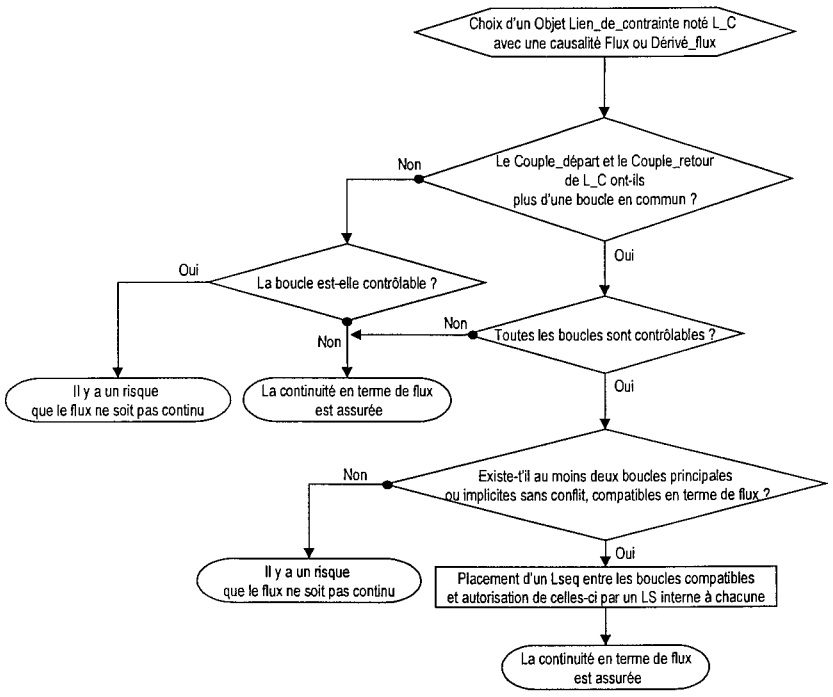



**Figure 3.21 : La vérification de la contrainte de continuité**

- **Phase 4 : Vérification de la cohérence des boucles contenant plusieurs gyrateurs.** Ces derniers posent un problème particulier parce qu'ils réalisent des couplages entre des boucles avec un changement de domaine énergétique. Pour vérifier la cohérence du graphe dans ces cas là, il est nécessaire d'utiliser le concept des causalités dominantes. En toute rigueur, il est nécessaire d'effectuer un certain nombre de vérifications en considérant toutes les combinaisons d'états de boucle, mais ce processus serait extrêmement coûteux en temps. Nous proposons plutôt de décomposer la vérification des boucles avec des fonctions Gyrateur en deux étapes. La première est décrite par la figure 3.22. Elle est effectuée avant de fixer le mode de

fonctionnement, afin d'aider le concepteur à progresser dans la construction de sa solution.

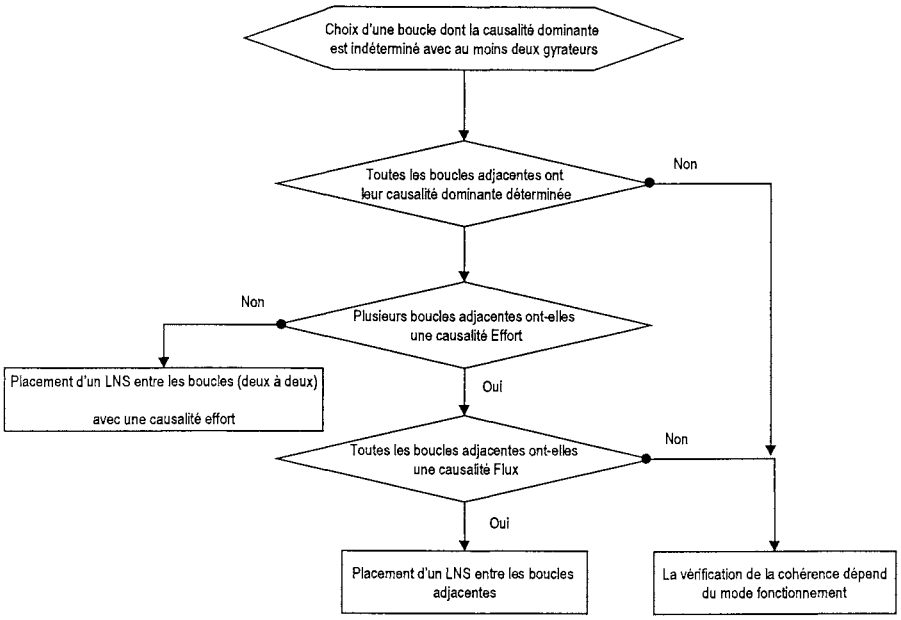



**Figure 3.22 : Le principe de la première étape de la vérification des boucles avec gyrateur**

**Définition** : deux boucles sont **adjacentes** si elles sont liées par une fonction gyrateur.

La seconde étape est développée dans le paragraphe suivant. Elle est effectuée après avoir fixé le mode de fonctionnement du convertisseur.

### II.4.3. Les vérifications à partir du mode de fonctionnement

Nous avons vu qu'un objet Etat_de_graphe est composé par combinaison d'un objet Etat_de_boucle de chaque boucle. Ces objets permettent donc de construire une cartographie des niveaux de flux dans tout le graphe. Il reste maintenant à vérifier que les interactions (dues au gyrateur) entre les boucles ne posent pas de problèmes énergétiques.
L'algorithme de la figure 3.23 permet de vérifier la viabilité et la causalité dominante des autres boucles. Pour cela, il cherche à déterminer les causalités dominantes et les niveaux de flux de toutes les boucles par un processus itératif.
Nous effectuons la propagation des contraintes de causalité en utilisant deux principes que nous présentons maintenant.

- **A partir de la notion de causalité dominante**, il est possible de vérifier la cohérence des boucles avec plus d'un gyrateur en appliquant les règles suivantes.

30. *S'il existe plus d'une boucle adjacente (lié par un gyrateur) avec une causalité dominante de type Effort alors il y a un conflit de type Flux sur la boucle.*
31. *S'il existe plus d'une boucle adjacente (liée par un gyrateur) avec une causalité Derivée_l'effort alors elles ne peuvent être qu'identiques sinon il y a conflit de type Dérivée_flux.*

32. *Au moins une boucle adjacente ne doit pas avoir une causalité type flux sinon il y a un conflit de type effort.*
33. *Au moins une boucle adjacente ne doit pas avoir une causalité type Derivée_flux sinon il y a un conflit de type Dérivée_effort.*

- **A partir des niveaux de flux des différents états du graphe**, il est possible de vérifier que les contraintes de continuité du flux et de l'effort sont effectivement respectées par le mode fonctionnement choisi.

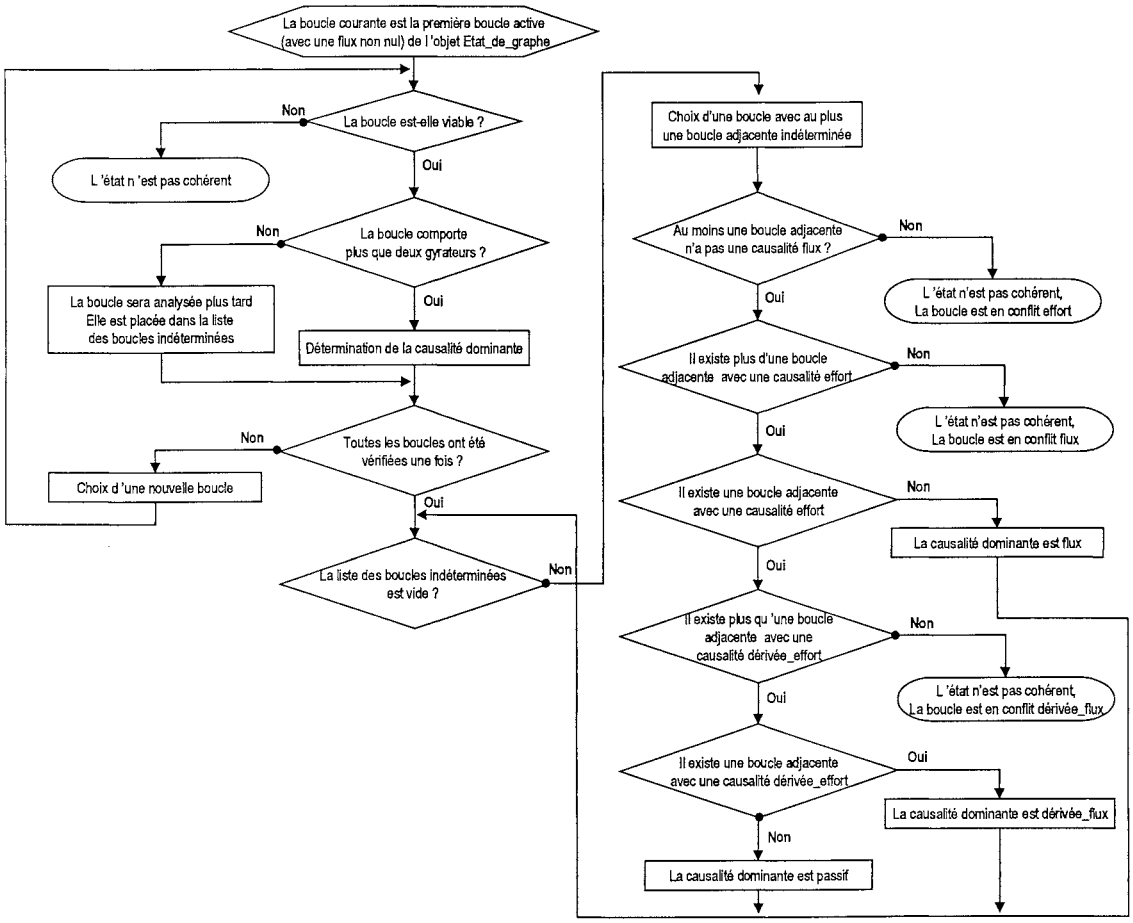



**Figure 3.23 : Le processus de vérification complète de la cohérence à partir du mode fonctionnement.**

# III. Construction de la solution finale

## III.1. Les éléments de la solution finale

La définition de la solution, dite structurelle, est la dernière étape de la conception des convertisseurs statiques. Pendant cette phase, l'expert doit spécifier les différentes fonctions en faisant des choix techniques pour leur réalisation et en donnant des ordres de grandeur pour les caractéristiques des composants.

En réalité, il n'existe pas de dichotomie parfaite entre les points de vue structurel et fonctionnel. Pendant la phase de conception, il peut être nécessaire d'effectuer des choix technologiques fonctionnels et des choix techniques qui figent une partie de la solution. Ainsi, l'évolution produite peut être modélisée par le schéma de la figure 3.24.

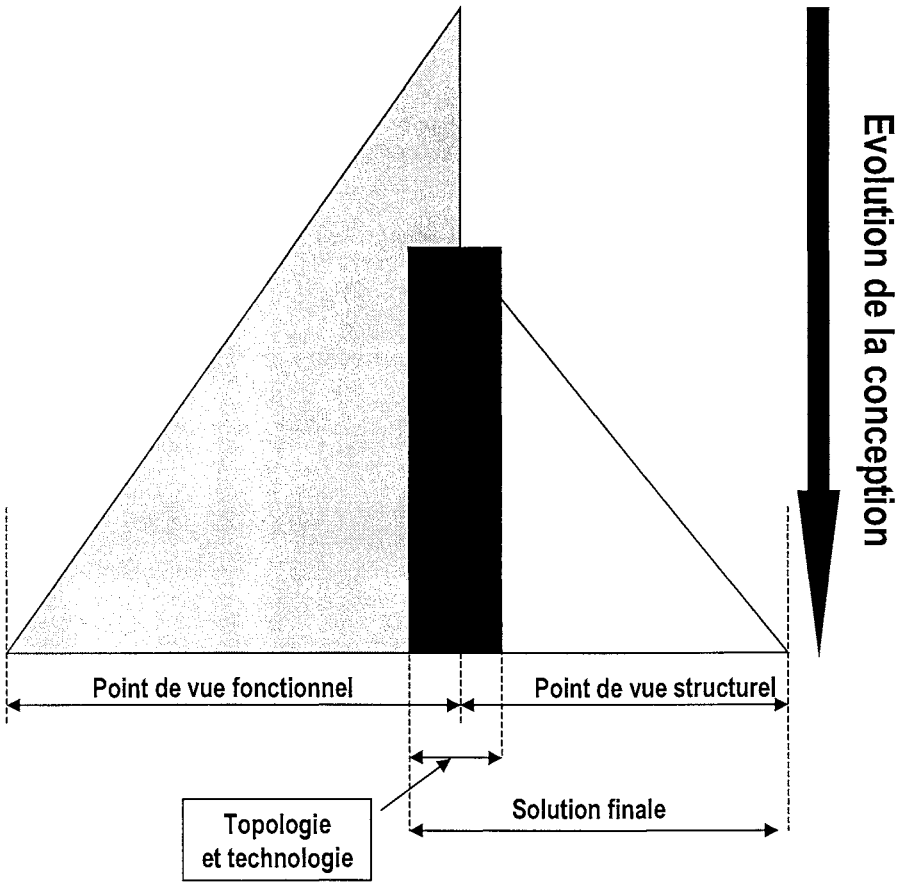



**Figure 3.24 : La séparation des points de vue fonctionnel et structurel**

Dans le cas d'un convertisseur, **le point vue fonctionnel** peut être défini par les aspects suivants.

- **Les fonctionnalités à remplir** qui sont matérialisées par les fonctions qui échangent de l'énergie.
- **Le mode de fonctionnement** du convertisseur.
- **Des fonctions dont il est nécessaire de figer une technologie globale.** Par exemple, le passage du domaine électrique au domaine magnétique pour les fonctions d'isolation.

De l'autre côté, **le point de vue structurel** est défini par les éléments suivants :

- **La décomposition en composants réels de toutes les fonctions.** L'expert peut par exemple décider qu'un interrupteur est réalisé à l'aide d'un thyristor et d'un circuit d'aide à la commutation. Pour cela, il a plusieurs solutions à sa disposition sans que la fonction remplie par l'ensemble ne soit modifiée et remise en cause.

- **La nature des composants.** Par exemple, un interrupteur peut être réalisé par une diode ou un transistor ou un autre type de semi-conducteur.

- **Les valeurs précises pour les caractéristiques des composants.** Par exemple, la valeur ou un ordre de grandeur d'un condensateur ou d'une inductance.

Enfin, la topologie finale du convertisseur statique est issue des deux moyens d'étude (fonctionnel ou structurel) parce qu'elle régit les transferts d'énergie entre les fonctions mais aussi entre les différentes structures. Ainsi, elle est obtenue à partir de la définition du graphe fonctionnel et de l'assemblage des différentes solutions techniques.

La solution finale - le schéma de construction du convertisseur - est l'ensemble du point de vue structurel et de la topologie. Pour obtenir un dispositif qui remplit bien la fonction pour lequel il a été conçu, il faut lui adjoindre le mode de fonctionnement issu du point de vue fonctionnel.

***Un graphe fonctionnel peut être transformé en plusieurs structures différentes suivant les solutions techniques mises en place mais le mode de fonctionnement global (l'enchaînement des modes de transfert d'énergie) reste toujours le même.***

## III.2. Le lien fonctionnel / structurel

Maintenant que la conception d'un graphe fonctionnel est bien formalisée (pour sa modélisation, sa définition et sa vérification), il se pose un problème à résoudre : le passage des fonctions aux structures. Les fonctions peuvent être considérées comme des moules génériques qui peuvent être remplies par une ou plusieurs structures.

### III.2.1. De la fonction vers les structures

#### *III.2.1.1. Présentation*

La définition d'une ou plusieurs structures à partir d'une fonction avec ses contraintes, peut se résoudre simplement en utilisant une base de données qui fournit une ou plusieurs solutions techniques pour remplir une fonction. Cette approche est envisageable dans le cas des convertisseurs statiques parce que le nombre de fonctions de base est faible (fonctions d'énergie, fonctions de contrôle ou de retour de flux, gyrateur, etc.).
L'expertise nécessaire pour mener à bien cette étape est alors codée implicitement dans la définition de la base de données. Celle-ci doit alors comporter les informations suivantes.

- **Un ou plusieurs ensembles (dits structures) de composants réels sont liés à une fonction.** Le critère de recherche est alors une solution technique.

  Par exemple si l'expert choisit le critère technique « Utiliser un transistor » alors une fonction de contrôle avec un fonctionnement correspondant à celui du thyristor dual

peut être réalisé à l'aide d'un transistor commandé de façon particulière et associé à une diode.

- **La décomposition d'une structure en plusieurs composants.**

  Par exemple, une structure résonante est composée d'un condensateur et d'une inductance.

- **Un ordre de grandeur de la valeur des composants ou des relations entre les valeurs des composants.**

  Par exemple, pour qu'un circuit soit effectivement résonant la fréquence de la grandeur d'excitation doit être de l'ordre de sa fréquence caractéristique (ou de résonance de l'ensemble).

- **Le réseau de LTE entre les différents composants**. Celui-ci permettra de compléter la topologie par rapport à sa définition fonctionnelle.

  Par exemple, le condensateur et l'inductance d'un circuit résonant peuvent être connectés en parallèle ou en série.

Toute cette connaissance a été en partie analysée (l'annexe II en présente un extrait), mais n'est pas encore formalisée pour être exploitable directement à partir de notre définition des fonctions. Nous avons néanmoins formalisé le principe de la procédure de passage dans le cas des fonctions de contrôle du flux.

### *III.2.1.2. Le cas des fonctions de contrôle du flux*

La spécification des fonctions de contrôle en interrupteurs réels est une phase essentielle de la définition de la structure. A partir de l'analyse de la cohérence, nous obtenons un certain nombre de liens temporels qui sont des contraintes pour la définition des enchaînements, donc des commutations des semi-conducteurs réels. Il est alors possible d'identifier plusieurs cas standards qui correspondent en partie aux cellules de commutations

- **Cas standard 1 : deux fonctions de contrôle du flux liées par un LNS dans deux boucles liées par un LSEQ.** Ce cas correspond souvent à la cellule de commutation complémentaire [FOC]. Dans ce cas, les interrupteurs peuvent être contrôlables de façon complémentaire mais les contraintes se répercutent sur la commande où un interrupteur est un composant commandable et l'autre une diode (si les contraintes en terme d'effort le permettent).

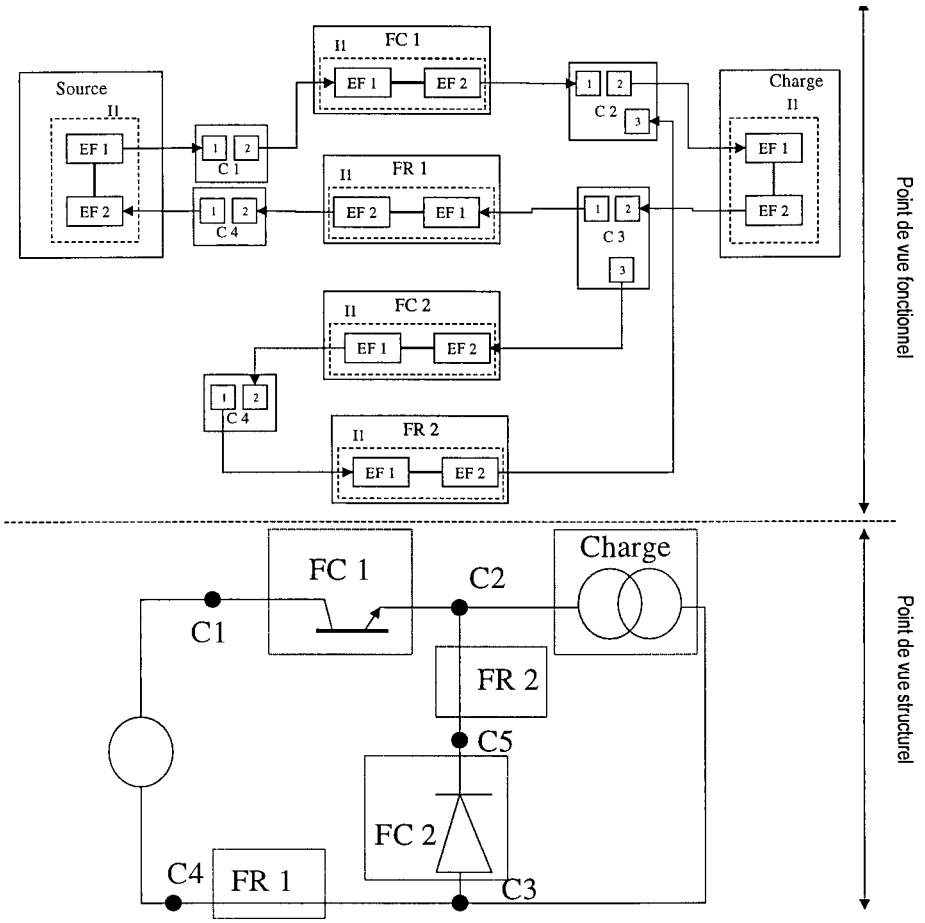



**Figure 3.25 : Un graphe fonctionnel correspondant au cas standard 1 pour la détermination des interrupteurs**

- **Cas standard 2 : plus de deux fonctions de contrôle du flux, dans des boucles différentes liées par un LNS et un LSeq, (donc commandées de façon séquentielle)**. Ce cas correspond à une cellule polyphasée et il peut être résolu en spécifiant les interrupteurs comme des thyristors (si les contraintes le permettent).

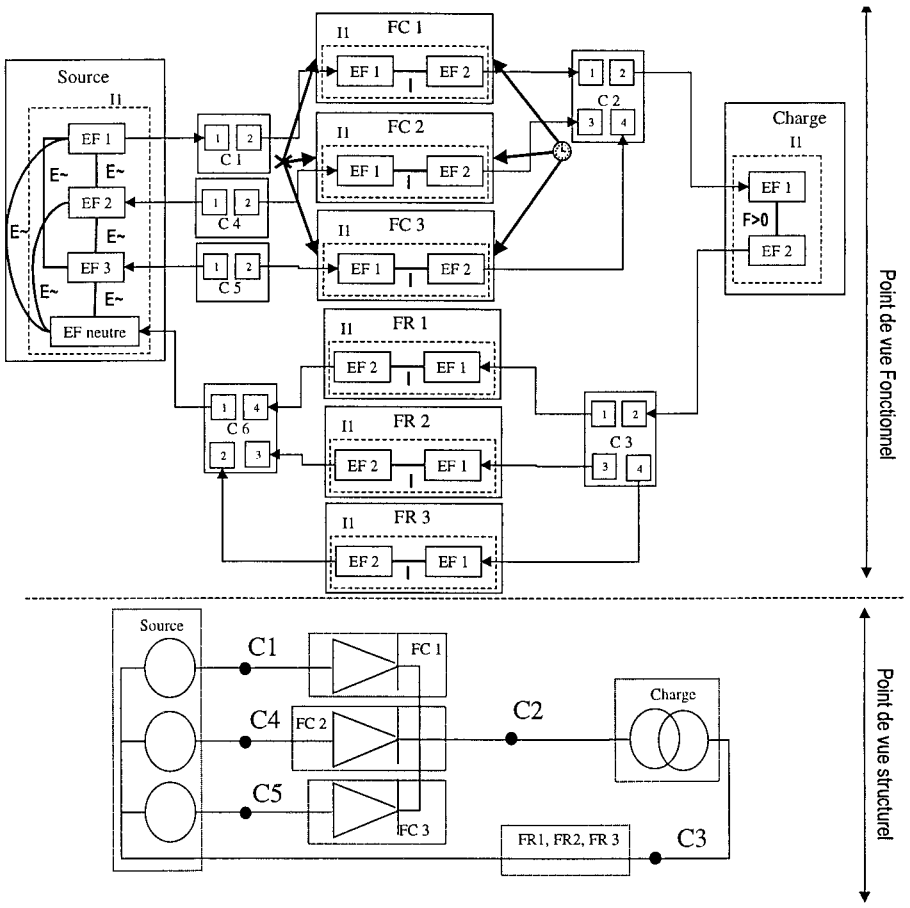



**Figure 3.26 : Un graphe fonctionnel correspondant au cas standard 2**

Ces solutions sont très courantes dans les convertisseurs actuels. L'approche fonctionnelle permet de les identifier facilement et surtout de façon automatique. Cependant, elle ne prend toute sa dimension que si un concepteur désire définir une commande spéciale dans le but d'améliorer les performances ou traiter de grosses structures.

## III.2.2. Le lien inverse : structurel-fonctionnel

Le problème du lien inverse est beaucoup plus délicat à mettre en œuvre, notamment lorsque que l'on ne possède que la solution finale. Les difficultés viennent de plusieurs aspects.

- **La solution finale est moins riche en informations**. En effet, à partir de cette dernière, nous connaissons directement le mode de fonctionnement (c'est-à-dire l'enchaînement de transferts de flux) et la topologie, mais il manque des informations de nature fonctionnelle comme les objets Lien_temporel ou les objets Lien_de_contrainte.

- **La structure à analyser peut avoir été optimisée en terme de composants**. Dans ce cas, un composant peut contribuer à deux fonctions ce qui complique sérieusement l'identification des fonctions.

La définition d'un graphe fonctionnel à partir d'une structure existante demande en réalité une analyse équivalente à la conception complète d'un graphe à partir du cahier des charges. A

partir de la structure, il est possible d'identifier rapidement les principales fonctions du système en utilisant les considérations suivantes :

- **Les ensembles sources du système peuvent être identifiés comme des fonctions sources.** Les contraintes doivent être définies pour représenter le comportement externe des sources (en terme d'effort et de flux).

- **Les ensembles charges sont transformés en une fonction modélisant leur comportement externe (de la même façon que les sources).**

- **Les éléments filtrant sont transformés en leur équivalent fonctionnel.** Par exemple, un filtre de courant peut être considéré comme une inertie de flux.

- **Les interrupteurs et leur circuit d'aide à la commutation sont remplacés par une fonction de contrôle de flux** avec des états de contraintes correspondantes et les transitions associées aux conditions de commutation.

Les Chapitres 2 et 3 ont présenté les concepts utilisés pour construire notre modèle orienté objet en vue la conception fonctionnelle. Celui-ci définit les fonctions, les boucles de flux et les graphes fonctionnels. De plus, il permet la vérification du graphe de cohérence d'un point de vue énergétique pour en déduire des critères pour la réalisation de la commande du convertisseur. Le chapitre suivant présente les concepts utilisés pour construire un premier prototype utilisant notre modèle. Celui-ci nous permet de mettre au point les algorithmes et les règles de cohérence de ne notre approche.

# Chapitre 4 : Le prototype informatique

Le modèle objet présenté dans les chapitres précédents a été spécifié de façon indépendante d'un langage de programmation ou d'un environnement de développement informatique. Le but ce chapitre est de justifier nos choix pour l'implantation réelle de ce modèle. Dans la première partie de ce chapitre, nous présenterons nos besoins par rapport à un environnement de développement pour réaliser un premier prototype. Puis dans la seconde partie, nous développerons la structure informatique réelle de cet outil ainsi que les méthodes mises en œuvre pour obtenir effectivement un système évolutif.

## I. L'environnement de programmation

### I.1. Nos critères de choix

Les modèles objets sont maintenant très répandus. Ils garantissent une extension facile du logiciel, à condition qu'il ait été bien construit. Il existe d'ailleurs dans la littérature [OUS97] beaucoup de méthodes qui permettent de guider l'obtention d'une représentation objet, mais aucune ne garantit que le logiciel va correspondre aux besoins des utilisateurs. C'est le rôle du prototype informatique de tester les fonctionnalités auprès des utilisateurs. Ces critères sont particulièrement vrais pour notre outil qui sert à vérifier les actions de l'utilisateur.
Afin de tester la viabilité de notre modèle fonctionnel, nous proposons donc un outil informatique construit rapidement, dans un environnement particulier, développé en utilisant les possibilités offertes par ce dernier. Ensuite, l'expérience acquise par l'utilisation de ce premier prototype nous permettra d'avoir une réflexion concrète pour spécifier une structure idéale de l'outil d'analyse fonctionnelle.
Les besoins par rapport à notre outil peuvent se décomposer de la façon suivante.

- **L'environnement de développement doit permettre d'intégrer un modèle objet.** Celui doit fournir une interface permettant d'accéder aux attributs des objets (les champs et les méthodes) pour faciliter la phase de description du modèle objet et surtout permettre le suivi de l'état des différents objets pendant l'exécution de l'outil, ainsi que la phase de recherche des erreurs de fonctionnement en cours de programmation.

- **L'environnement doit fournir un formalisme de règle.** Cette fonctionnalité n'est pas obligatoire mais facilite grandement l'implantation (et la maintenance) des connaissances de nature heuristique. De plus, une utilisation astucieuse de telles règles permet d'étendre facilement le modèle. En effet, les objets fournissent les informations et procédures de base pour représenter et manipuler les données, alors que les règles permettent de coder de façon simple des actions liées à une problématique spécifique sur ces données. L'environnement devra donc fournir un moteur d'inférence capable d'enchaîner dynamiquement les différentes règles.

- **L'environnement doit fournir un protocole de gestion de plusieurs solutions.** Il nous permettra ainsi d'implanter un processus de conception automatique si le besoin s'en fait sentir. Notre environnement devra donc être capable de mémoriser des solutions partielles pour les réutiliser plus tard.

- **L'environnement doit permettre la sauvegarde et la restauration de l'état des données à tout instant du raisonnement (contexte).** En d'autres termes, l'environnement doit permettre facilement de sauvegarder (et de restaurer plus tard) les objets en mémoire à tout instant du raisonnement. Cette fonctionnalité permet d'implanter facilement la sauvegarde de solutions pour une utilisation ultérieure.

- **L'environnement doit fournir un langage de programmation dynamique.** Ce type de langage permet de tester rapidement des procédures ou des règles, sans passer par une phase de compilation qui peut s'avérer lourde. Il contribue alors à la flexibilité du système parce qu'il permet la création dynamique de classes d'objets (voir II.1.1)

- **L'environnement doit offrir des possibilités en terme de développement Interface Homme Machine (I.H.M.).** L'interface graphique peut sembler un élément annexe d'un outil d'aide à la conception, mais en réalité c'est elle qui permet de le rendre utilisable. Les environnements de développement modernes proposent une batterie d'outils permettant de concevoir des interfaces utilisateur graphique très rapidement. La difficulté de cette étape est d'arriver à définir une I.H.M. conviviale et répondant aux besoins de l'utilisateur du logiciel. Souvent, le prototype permet de tester l'ergonomie de l'I.H.M. afin de spécifier correctement l'outil final.

Notre outil sera alors constitué d'un noyau minimal contenant uniquement les classes et les méthodes génériques de notre modèle (c'est-à-dire relative à la création des fonctions, des boucles, etc.). Les autres classes sont alors définies dynamiquement au moment de l'exécution grâce à un fichier de configuration simple. Cette structuration permet une extension facile du modèle objet, par exemple pour spécifier la classe Interface afin d'ajouter un domaine énergétique (voir § II.1.2). L'I.H.M est quant à elle un module indépendant dialoguant avec ce noyau (voir § II.2).
Nous avons choisi l'environnement ILOG-LELISP pour implanter notre prototype parce qu'il répondait à tous nos critères. L'analyse de la phase de développement réel de notre prototype

et l'étude de l'ergonomie de l'I.H.M. nous permettra de définir un cahier des charge pour l'outil final.

## I.2. L'environnement de programmation ILOG-LELISP

### I.2.1. Présentation

L'environnement de programmation que nous utilisons est construit par la société ILOG à partir du langage de programmation LELISP qui est interprété et entièrement dynamique. En réalité, il comporte très peu d'éléments de base (fonctions lisp, structures de données). Cependant, il permet de définir très facilement de nouvelles structures de données ou de nouvelles fonctions pendant l'exécution d'un programme, sans la moindre étape de compilation.
ILOG-LE-LISP nous propose un environnement composé de quatre parties que nous décrivons maintenant.

#### *I.2.1.1. LE-LISP [LEL94]*

LE-LISP est un langage de programmation compatible avec le langage Lisp, mais il comporte plusieurs éléments supplémentaires :

- **des structures de données** supplémentaires ainsi que les fonctions associées, comme les ensembles et les tables de hachage ;
- **la possibilité d'avoir une représentation objet** des données ;
- **une machine virtuelle multitâche** qui permet d'obtenir un environnement indépendant de toute plate-forme matérielle ; en d'autres termes, LELISP présente une couche logicielle qui encapsule toutes les ressources matérielles de l'ordinateur sur lequel le programme s'exécute.

Ce langage est utilisé pour la programmation de tout le reste de l'environnement de développement ainsi que pour les parties procédurales de notre application.

#### *I.2.1.2. AIDA [AID94]*

AIDA est un moteur graphique orienté objet qui permet la construction d'interfaces graphiques muti-fenêtrées (à partir du langage LE LISP). Celles-ci sont constituées d'un ensemble d'objets de base fournis par ce moteur logiciel. La figure 4.1 présente un exemple de fenêtre graphique définie à l'aide du moteur AIDA ainsi que sa décomposition en objets.

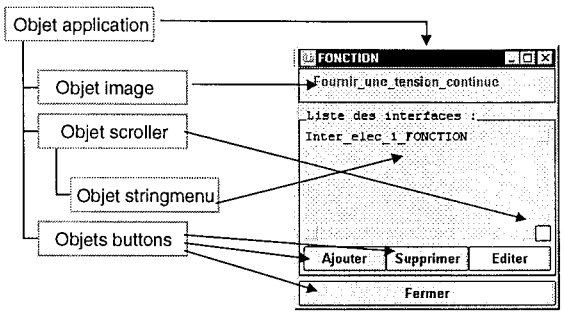



**Figure 4.1 : Description d'une fenêtre avec AIDA.**

### *I.2.1.3. MASAI [MAS94]I*

Cet outil est un éditeur graphique qui permet de construire rapidement la description en objet de fenêtres graphiques avec une I.H.M qui ressemble à celle d'un éditeur de dessins. MASAI permet donc :

- de définir l'aspect visuel d'une fenêtre en cours d'édition,
- de définir la décomposition en objet AIDA d'une fenêtre,
- de définir les procédures à exécuter lorsque l'utilisateur effectue une action sur un des objets de l'interface graphique,
- de programmer automatiquement le code source LELISP qui permet d'afficher une fenêtre.

### *I.2.1.4. SMECI [SME94]*

SMECI est une boîte d'outils informatiques qui permet la construction de systèmes experts. Il propose un modèle objet basé sur le formalise des « frames » [OUS96], [ESC96].

- **Les classes d'objets sont composées de caractéristiques stockées dans des structures de données spéciales appelées champs.** Ces derniers sont eux-même caractérisés par des informations supplémentaires telles que le nom du champ, le type de la donnée qu'il contient, la nature du champ, des contraintes en terme de valeur et des informations nécessaires à la manipulation du champ. En complément, SMECI propose plusieurs éditeurs qui permettent de définir la structure des classes par une interface graphique.
- **Un objet spécial (appelé prototype) est associé à chaque classe.** Il permet de contraindre la valeur des champs.
- **Les classes comportent des méthodes définies par l'utilisateur** (de façon statique ou dynamique).

- **Les classes comportent des méthodes spéciales (appelée démons) déclenchées automatiquement** lorsque le programme effectué crée un objet ou modifie la valeur des champs d'un objet.

En complément, SMECI propose un certain nombre de fonctionnalités qui permettent le codage et la manipulation de la base connaissance du système expert.

- **La modélisation de la connaissance déductive par des règles regroupées en tâches.** Celles-ci sont organisées de façon hiérarchique pour organiser la connaissance. SMECI propose des éditeurs graphiques qui permettent de pister l'application de ces règles.

- **Un moteur d'inférence qui permet d'enchaîner l'exécution des règles** (par les tâches) sur les objets en mémoire.

- **La mémorisation de l'évolution du contexte mémoire,** c'est-à-dire de l'ensemble des objets présents en mémoire à un instant donné du raisonnement, par la notion d'état.

- **La sauvegarde et la restauration d'un contexte mémoire**.

### I.2.2. Concepts utilisés dans l'environnement

Notre prototype utilise le formalise de modélisation objet des données proposé par SMECI. Le formalise de règle offert par ce dernier nous permet de coder les règles de cohérence de façon simple.
La partie graphique de notre application (IHM) est gérée par un assemblage d'objets AIDA, construit à l'aide de SMECI.
Le graphe fonctionnel est stocké à l'aide des objets Grapher proposés par AIDA. Ce dernier nous propose également des méthodes de gestion et de parcours de ces graphes.
Les paragraphes suivants expliquent la structure générale de notre prototype pour gérer les différents aspects de notre modèle.

## II. Réalisation du prototype

Nous présentons dans ce paragraphe la réalisation informatique de notre prototype. Nous commencerons par décrire l'environnement de développement, et notre utilisation de celui-ci, puis l'implantation du modèle présenté dans les chapitres précédents, et enfin les possibilités d'évolution offertes par notre structure informatique.

### II.1. L'implantation du modèle fonctionnel

Notre prototype est construit de façon à pouvoir être étendu facilement. C'est-à-dire qu'il doit respecter les critères suivants :

- **L'ajout d'une fonction doit être facile et rapide.** Ceci doit être possible, de préférence, sans modifier le code de notre programme.

- **L'ajout de nouveaux domaines énergétiques doit être possible.** Cela nous permet de prendre en compte de nouvelles contraintes fonctionnelles puisque notre formalisme n'est pas attaché au domaine électrique.

- **L'ajout de nouvelles règles de cohérence doit être possible.**

La dynamicité du langage LE-LISP nous permet d'écrire la procédure réalisant la création des classes dynamiquement au moment de l'exécution du programme. Nous allons maintenant présenter comment nous avons défini la syntaxe des fichiers de configuration pour que notre prototype puisse créer les classes de fonctions et les classes d'objets pour définir les domaines énergétiques.

### II.1.1. La définition dynamique des classes de fonctions

Afin de permettre une extension facile du modèle, les fonctions sont construites dynamiquement au moment de l'exécution du programme. L'utilisateur peut ainsi définir les différentes classes de fonctions pour que notre prototype puisse les instancier. La figure 4.2 présente un exemple de fichier de configuration (nommé fonct.ini) permettant de créer les classes de fonctions utilisées pour nos exemples de test. Cette description crée la classification des fonctions que nous avons définie au chapitre 2.

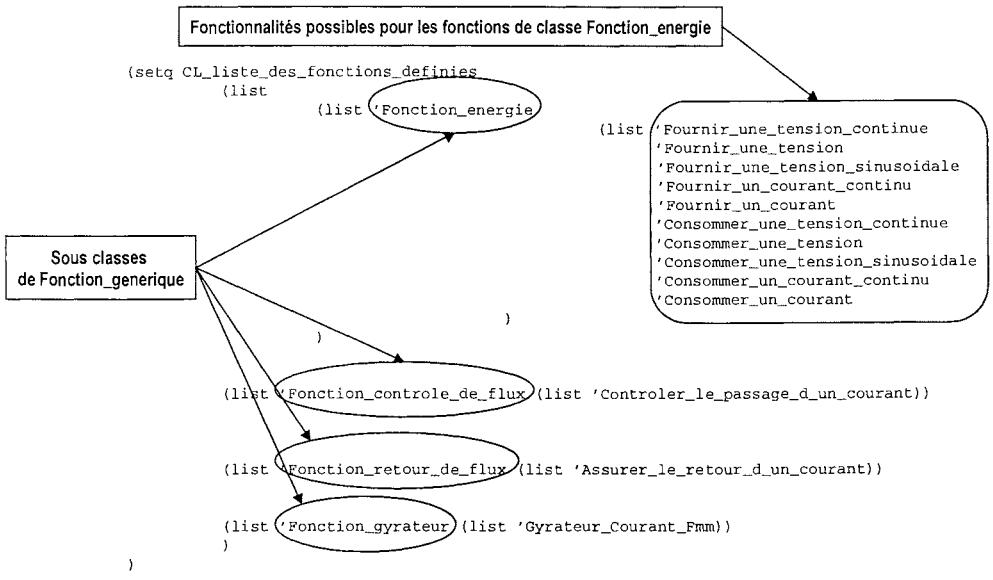



**Figure 4.2 : Le fichier de définition permettant la création dynamique des sous-classes de la classe Fonction_generique**

En utilisant le formalisme de règles de SMECI, la définition des interfaces et des contraintes associées aux fonctionnalités se fait grâce à un fichier de description qui définit les interfaces en indiquant les informations suivantes :

- Les objets interface à créer,
- le type d'interface à créer,
- les formes d'onde caractéristiques,
- la causalité des contraintes.

L'utilisateur ne doit spécifier dans ce fichier que les informations liées à la fonctionnalité décrite. Les données manquantes seront demandées de façon interactive (via une interface graphique) au moment de l'instanciation de la fonction (c'est-à-dire la création d'un objet de la classe). La figure 4.3 présente la règle permettant la définition de la fonctionnalité « fournir une tension continue ».

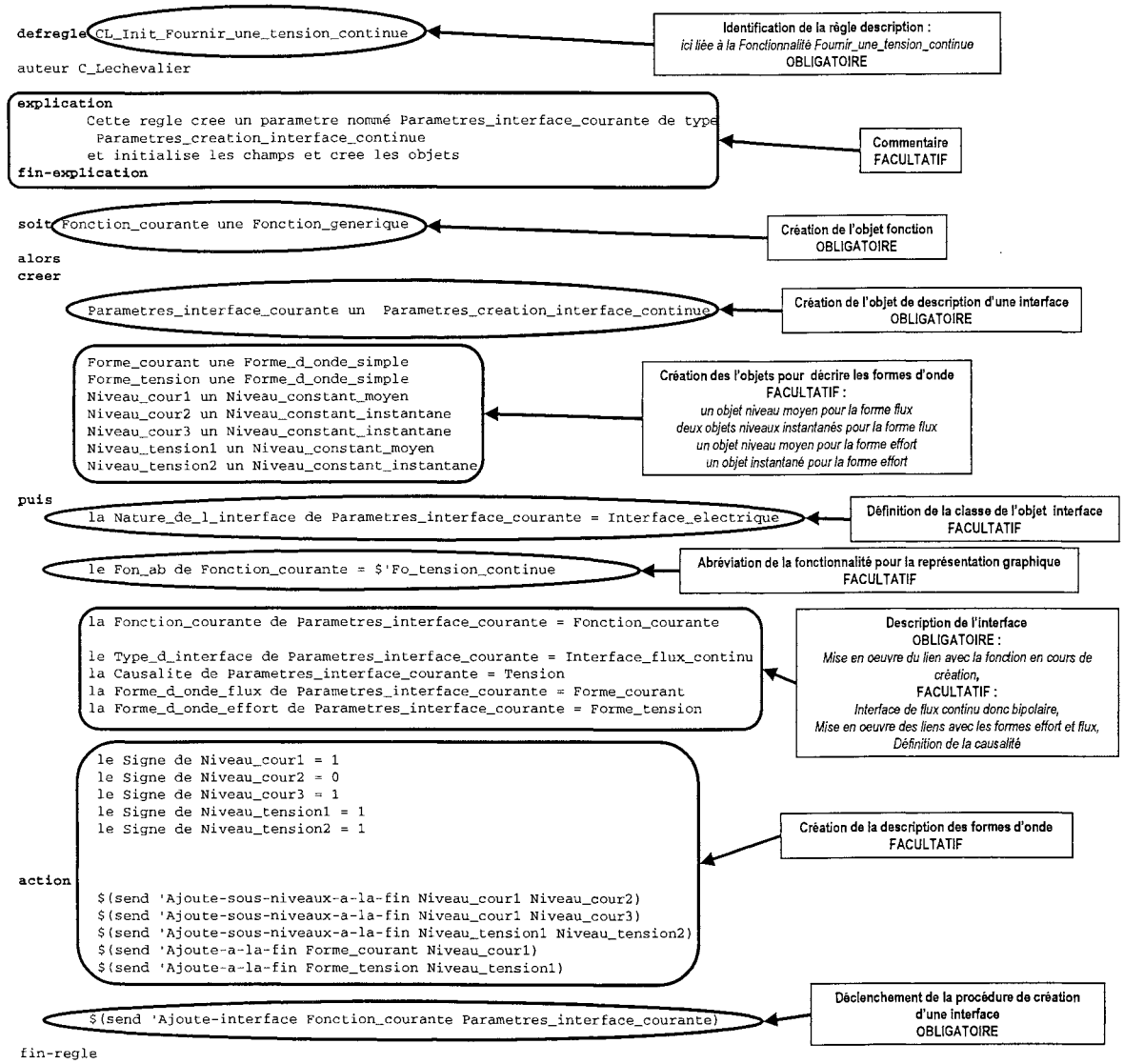



**Figure 4.3 : Exemple de règle de définition des interfaces liées à une fonctionnalité**

Cette règle permet de décrire le type énergétique des interfaces, les formes d'onde caractéristiques et la causalité associée. Ici, la règle crée une fonction avec une causalité Tension et les objets Forme_d_onde correspondants.

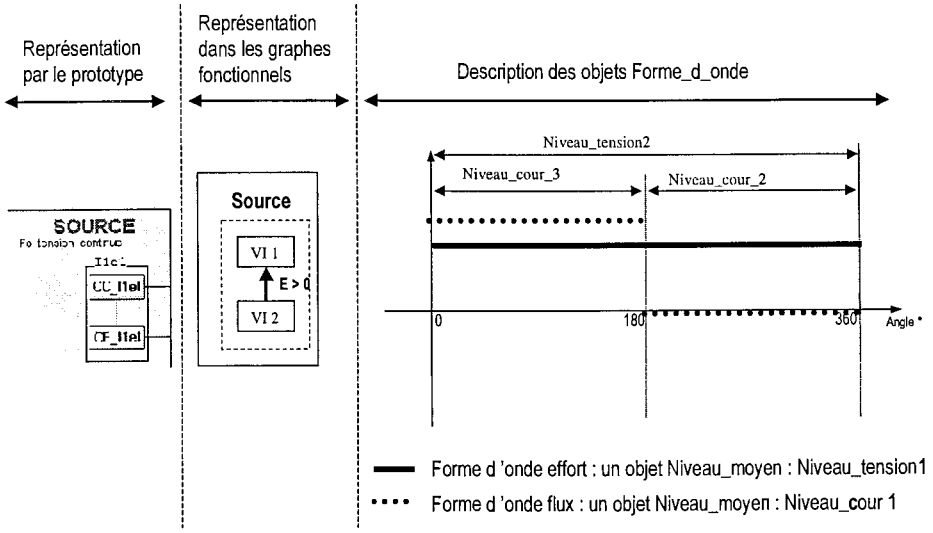



**Figure 4.4 : La description de la fonction définie par la règle de la figure 4.3**

### II.1.2. La définition dynamique des domaines énergétiques

De la même façon que pour les fonctions, notre prototype ne propose que la définition générique du domaine énergétique. En d'autres termes, le modèle de base définit seulement les classes Interface, Couple_EF et Lien de contrainte (voir § ch2 : III.2). Les sous-classes qui permettent de définir des domaines énergétiques sont définies grâce au fichier de description des domaines énergétiques construit suivant le modèle de la figure 4.5.

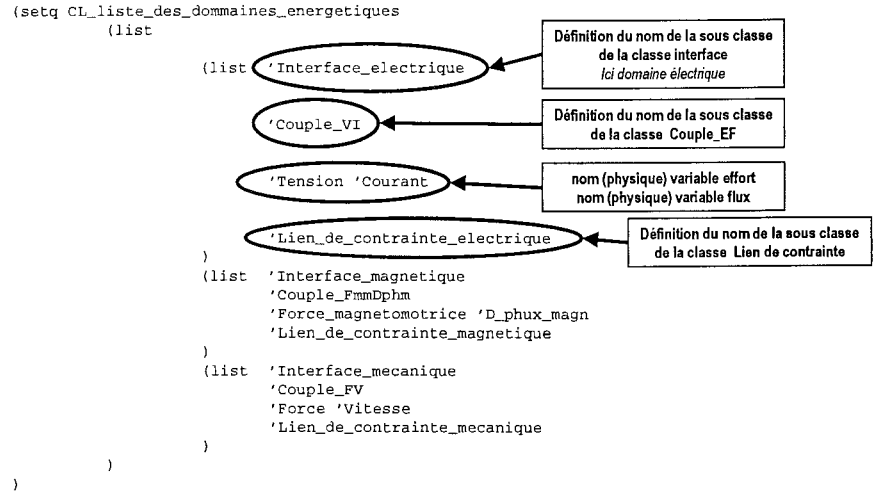



**Figure 4.5 : Un fichier de description des domaines énergétiques**

### II.1.3. La gestion dynamique des règles

Nous avons choisi d'implanter les algorithmes de vérification de la cohérence de façon particulière. Ils sont décomposés en deux parties.

- **Une partie composée de règles.** Cette structure de règles permet d'écrire très simplement les tests et provoquer des réactions adaptées.

- **Une partie méthodes de service.** Nous avons attaché aux objets de classe Boucle de notre noyau un certain nombre de méthodes de service qui permettent de connaître des informations sur les boucles et de simplifier l'écriture des règles.

Prenons l'exemple de l'algorithme de vérification de boucle. Il est structuré sous forme de taches et sa décomposition correspond à la figure 4.6.

- **CL_Verification-causalite-d-une-boucle.** La tâche applique les règles qui cherchent à trouver les boucles en conflit. En cas de conflit, un LNS (objet Lien_de_non_simultaneite) est placé sur la boucle.

- **CL_Autorisation-d-une-boucle.** Cette tâche place un LS (objet Lien_de_simultaneite) sur les fonctions qui ne sont pas en conflit.

- **CL_Verification-causalite-d-une-boucle-gyrateur :** Cette tâche tente de déterminer la causalité des boucles avec gyrateur pour déterminer les conflits éventuels.

- **CL_Création-d-etat-de-boucle :** Cette tâche construit les états de boucle pour les boucles de flux autorisées (donc sans conflit) en vue de la vérification de la cohérence.

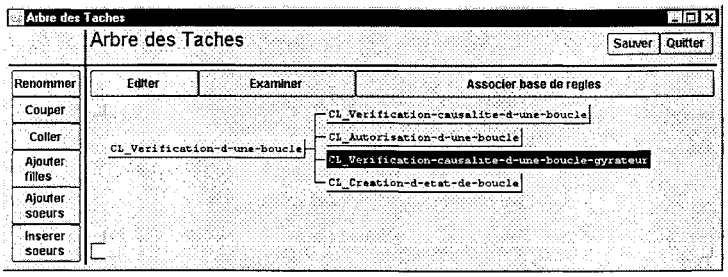



**Figure 4.6 : L'arbre des tâches de l'algorithme de vérification de la causalité d'une boucle**

En parallèle, la classe boucle présente un certain nombre de méthodes de service (présentées par la figure 4.7) pour l'écriture des règles de cohérence. Celles-ci permettent :

- d'obtenir des informations sur le type (la classe) des fonctions de la boucle,

- de rechercher des liens de contraintes de la boucle,

- de placer des liens temporels sur la boucle.

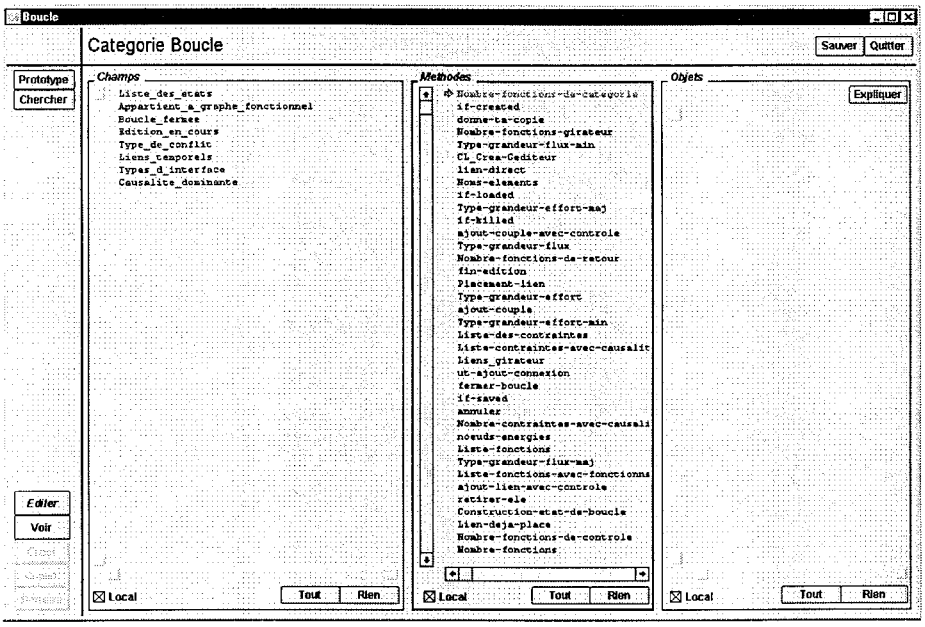



**Figure 4.7 : La définition de la classe boucle**

Une règle de vérification de la causalité peut être définie par le fichier source de la figure 4.8. Dans notre prototype actuel, il existe 15 règles similaires à celle-ci dans le seul but d'effectuer la vérification de la cohérence des boucles.

```
;
;*****************************************************************
;****
;**** Regle CL_Verification-causalite-effort-avec-cont
;****
;*****************************************************************
;
defregle CL_Verification-causalite-effort-avec-cont

explication
        Cette regle déclenche si la boucle Bo ne comporte que des causalités Tension
fin-explication

soit Bo une Boucle
        et graph = Appartient_a_graphe_fonctionnel^Bo

si Boucle_fermee^Bo = 1
   et   Nombre-fonctions-girateur^Bo < 2
   et   $(send 'Nombre-contraintes-avec-causalite Bo 'Effort)  > 0
   et   Nombre-fonctions-de-controle^Bo + Nombre-fonctions-de-retour^Bo > 1
   et   $(send 'Nombre-contraintes-avec-causalite Bo 'Effort)
        = Nombre-fonctions^Bo - Nombre-fonctions-de-controle^Bo - Nombre-fonctions-de-retour^Bo
   et  $(send 'Lien-deja-place Bo () 'Lien_de_non_simultaneite) = ()
   et  le Type_de_conflit de Bo = ()

alors
        le Type_de_conflit de Bo = Effort-I
action
        $(S-add-value Liste_des_boucles_en_conflit graph Bo)
        $(send 'voir-boucle Appartient_a_graphe_fonctionnel^Bo Bo 3)
        $(CL_Alerte (catenate "Verification de la boucle " name^Bo)
                    "ERREUR de causalite"
                    (catenate "La boucle est en conflit " (send 'Type-grandeur-effort-maj Bo))
                    "")
        $(send 'voir-boucle Appartient_a_graphe_fonctionnel^Bo () ())
        $(send 'Placement-lien Bo () 'Lien_de_non_simultaneite)
fin-regle
```



**Figure 4.8 : Un exemple de règle de vérification de causalité**

## II.2. Les interactions graphiques de notre prototype

La majorité des logiciels d'aide à la conception modernes proposent une interface graphique même simple pour leur interaction avec l'utilisateur. Cette I.H.M. est la vitrine du logiciel et sa convivialité favorise la simplification de l'utilisation. Cet élément permet aussi de garantir simplement que l'utilisateur entre des données cohérentes pour l'utilisation du programme. Par exemple, la figure 4.9 présente la fenêtre de notre prototype qui permet le paramétrage de la procédure de création d'une interface.

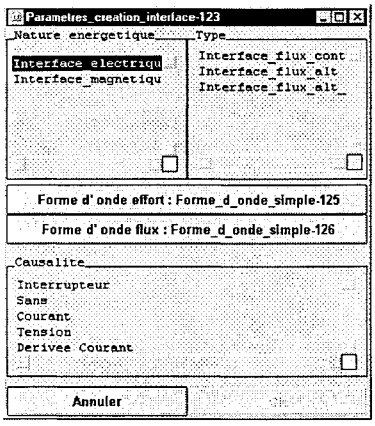



**Figure 4.9 : La fenêtre graphique utilisée pour le paramétrage de la création des interfaces**

Pour que la maintenance du logiciel obtenu soit simple, il est nécessaire de prévoir les interactions graphiques (I.H.M.) lorsque l'on implante le modèle objet de l'application. Nous avons choisi de construire un logiciel structuré en trois couches que nous présentons dans les paragraphes suivants.

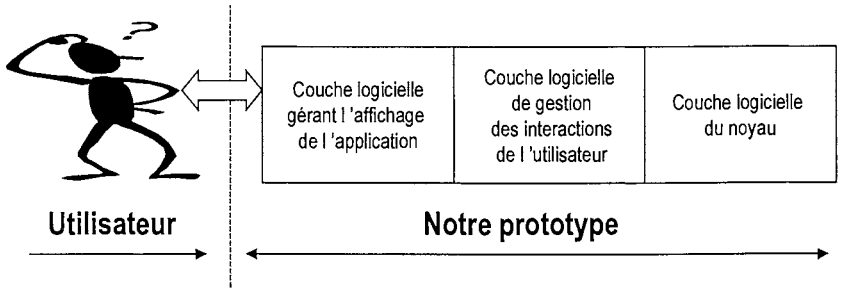



**Figure 4.10 : La structure générale de notre application**

### II.2.1. La couche objet du noyau

Celle-ci est définie avec le formalisme SMECI. Cette couche contient le modèle fonctionnel de base défini dans les chapitres 2 et 3, et les liens vers les autres couches afin de définir de façon dynamique les interactions graphiques.
Le noyau remplit les services suivants.

- **Il permet de créer une nouvelle fonction,** en appliquant l'algorithme de création de fonction présenté au chapitre 2.

- **Il permet la vérification de la cohérence,** en appliquant les règles présentées au chapitre 3.

- **Il permet la sauvegarde et la restauration du contexte mémoire du logiciel.** Ces opérations s'effectuent suivant l'algorithme de la figure 4.11 qui sauvegarde les différents objets du noyau mais aussi un certain nombre d'informations sur les autres couches graphiques afin de les restaurer dynamiquement plus tard.

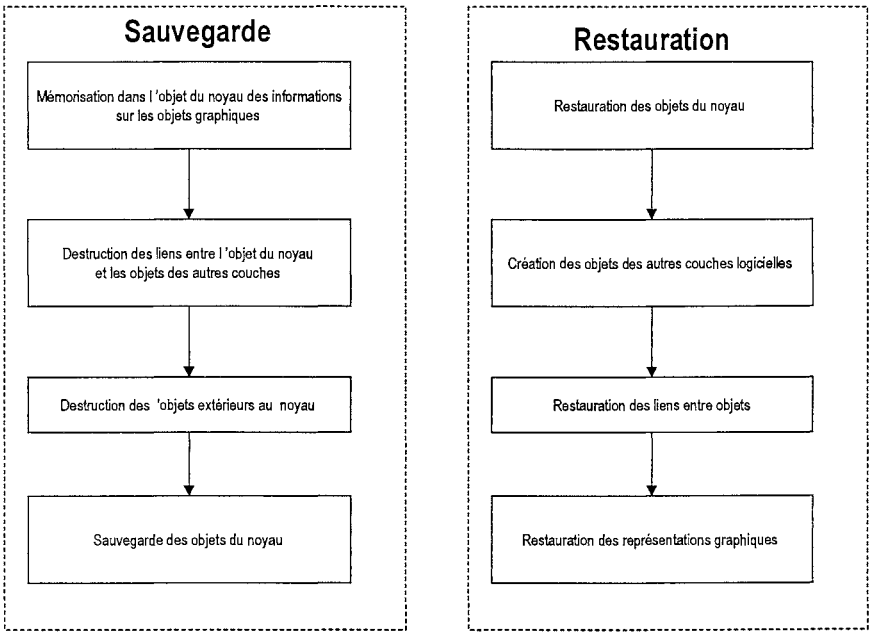



**Figure 4.11 : Les algorithmes de sauvegarde et de restauration**

Cette couche logicielle constitue le moteur de fonctionnement du logiciel. Elle est parfaitement autonome et contient toute l'intelligence de notre prototype pour manipuler le modèle fonctionnel et contrôler le comportement des couches graphiques.

### II.2.2. La couche graphique pour l'édition de valeurs des champs

Cette couche logiciel est constituée d'objets graphiques (appelés éditeurs graphiques) qui servent à la gestion de fenêtres graphiques de l'I.H.M. Ces éditeurs graphiques permettent à l'utilisateur de modifier la définition des fonctions de façon interactive. Nous concevons des éditeurs spéciaux parce que c'est un moyen efficace de contrôler les données entrées par l'utilisateur en le guidant. L'I.H.M. nous garantit que les données et les ordres en entrée de notre application sont corrects. Le fonctionnement d'une structure de ce type peut se modéliser comme sur la figure 4.12.

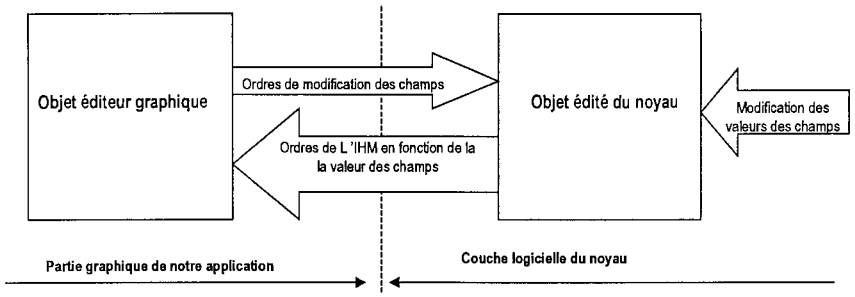



**Figure 4.12 : Schématisation du fonctionnement de l'interaction I.H.M. - Noyau**

Par exemple, les interfaces peuvent être modifiées par l'éditeur graphique présenté sur la figure 4.13.

Les objets interfaces du noyau possèdent un ensemble de méthodes qui sont appliquées lorsque l'utilisateur effectue une action sur l'un des boutons de l'interface.

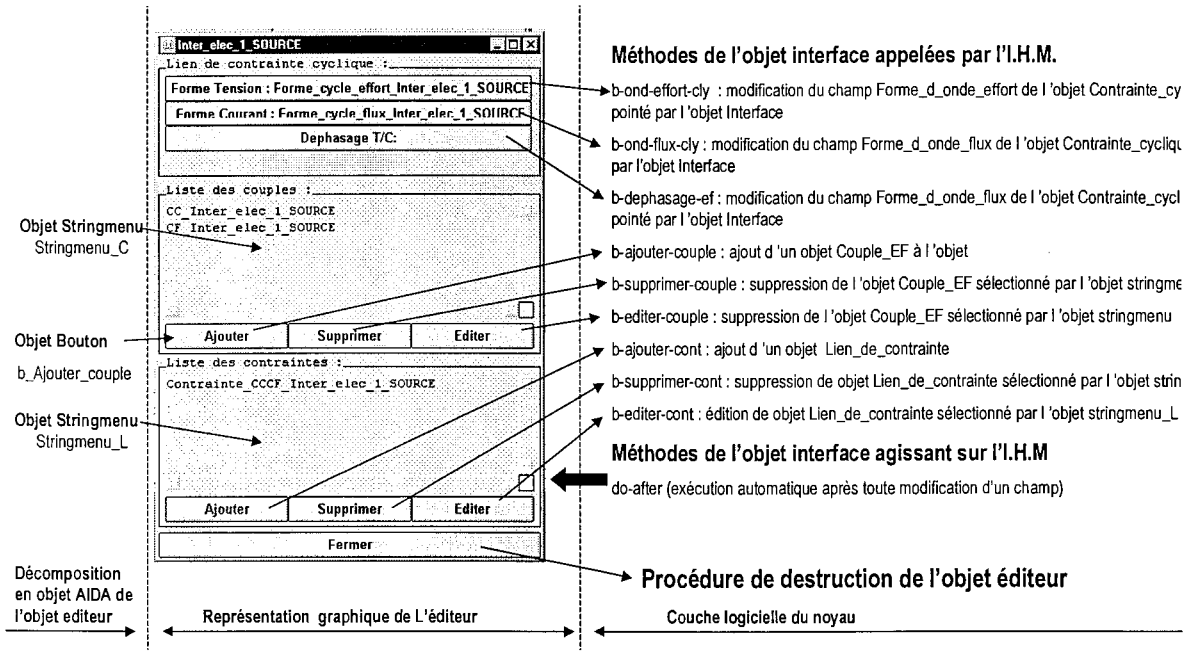



**Figure 4.13 : L'éditeur graphique des interfaces et les méthodes de l'objet interface invoquées en cas d'action de l'utilisateur**

Les couches d'édition graphique et le noyau peuvent alors être développés de façon indépendante. Il est donc possible de changer l'aspect graphique de notre application sans aucune modification du noyau.

### II.2.3. La gestion interactive du graphe fonctionnel

L'I.H.M. attachée (voir figure 4.14) à un objet Graphe_fonctionnel requière une structure informatique plus complexe. En effet, celle-ci doit effectuer plusieurs actions.

- **Gérer la valeur des champs de l'objet Graphe_fonctionnel.** Cette fenêtre fait office d'éditeur graphique pour un objet Graphe_fonctionnel.

- **Gérer la représentation schématique du graphe fonctionnel.** C'est à dire gérer l'aspect graphique du graphe fonctionnel avec la représentation schématique des

fonctions et leurs points de sortie énergétique (les objets Couple_EF et Interfaces). L'I.H.M. doit pouvoir représenter une vue graphique du graphe fonctionnel.

- **Gérer le fonctionnement du noyau.** L'I.H.M. doit offrir des possibilités pour ajouter ou éditer une fonction, vérifier la cohérence du graphe, restaurer ou sauvegarder un graphe fonctionnel (via l'objet tableau de bord).

- **Gérer la construction des boucles de façon interactive.**

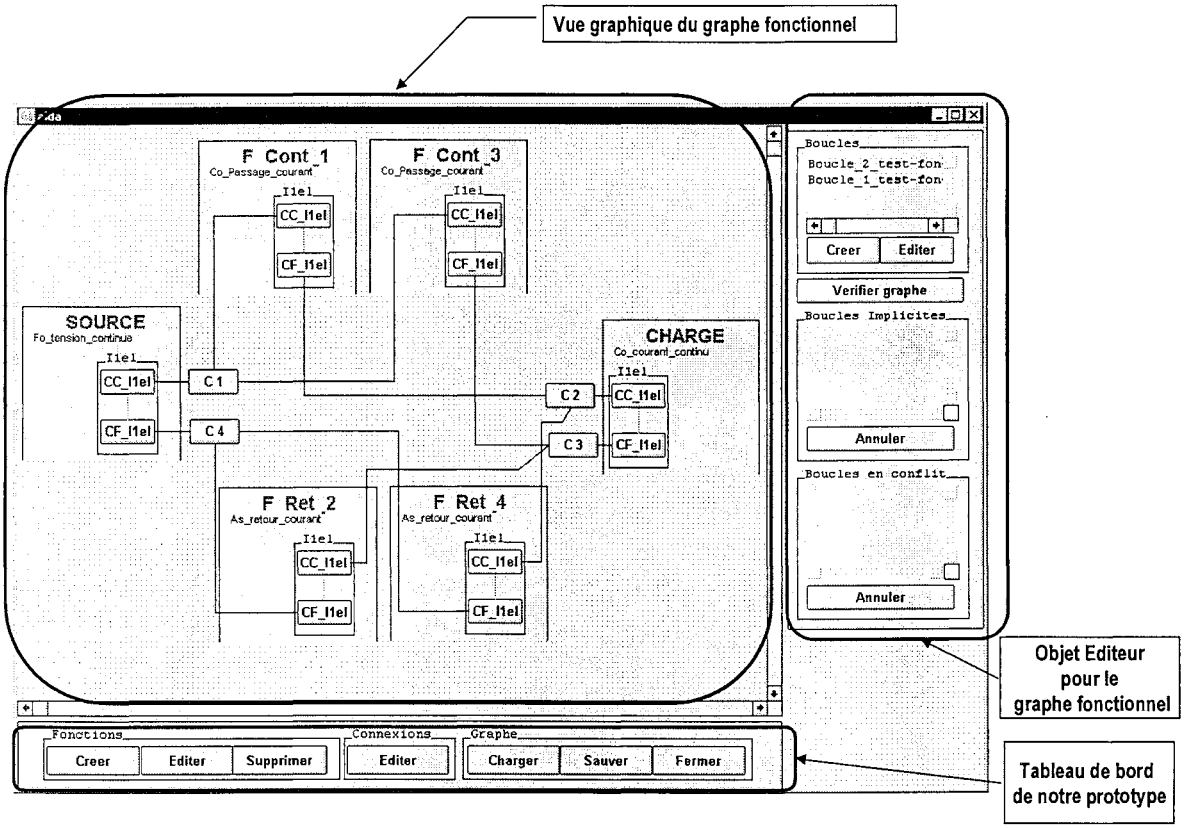



**Figure 4.14 : L'IHM attachée au graphe fonctionnel**

Afin de respecter ces critères, nous avons choisi la structure informatique à trois couches logicielles présentée sur la figure 4.15. Pour gérer les interactions de l'utilisateur sur la vue graphique du graphe fonctionnel, nous utilisons une couche intermédiaire fournie par le moteur graphique AIDA (le grapheur d'AIDA [AID94]). Avec cette struture le noyau reste toujours indépendant de la gestion graphique et des interactions de l'utilisateur, il se contente de recevoir les ordres (sous la forme de déclenchement de méthodes) et de réagir sur la couche intermédiaire et les éditeurs pour répercuter les actions.

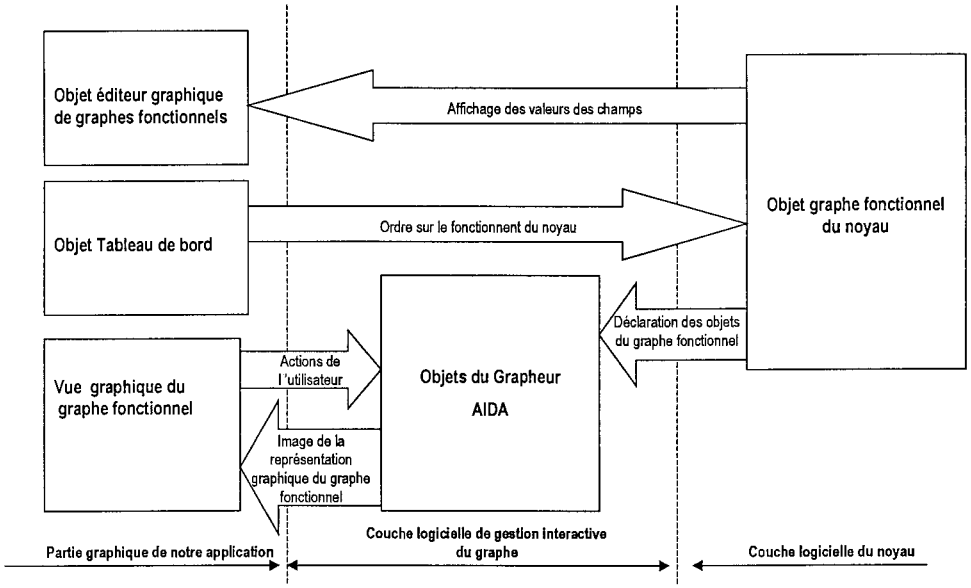



**Figure 4.15 : La structure informatique pour la gestion du graphe fonctionnel**

Nous utilisons une structure équivalente pour réaliser la construction des boucles. L'interface graphique obtenue est présentée sur la figure 4.16 dans ce mode de fonctionnement. L'objet éditeur de graphe est remplacé par un éditeur de boucle.

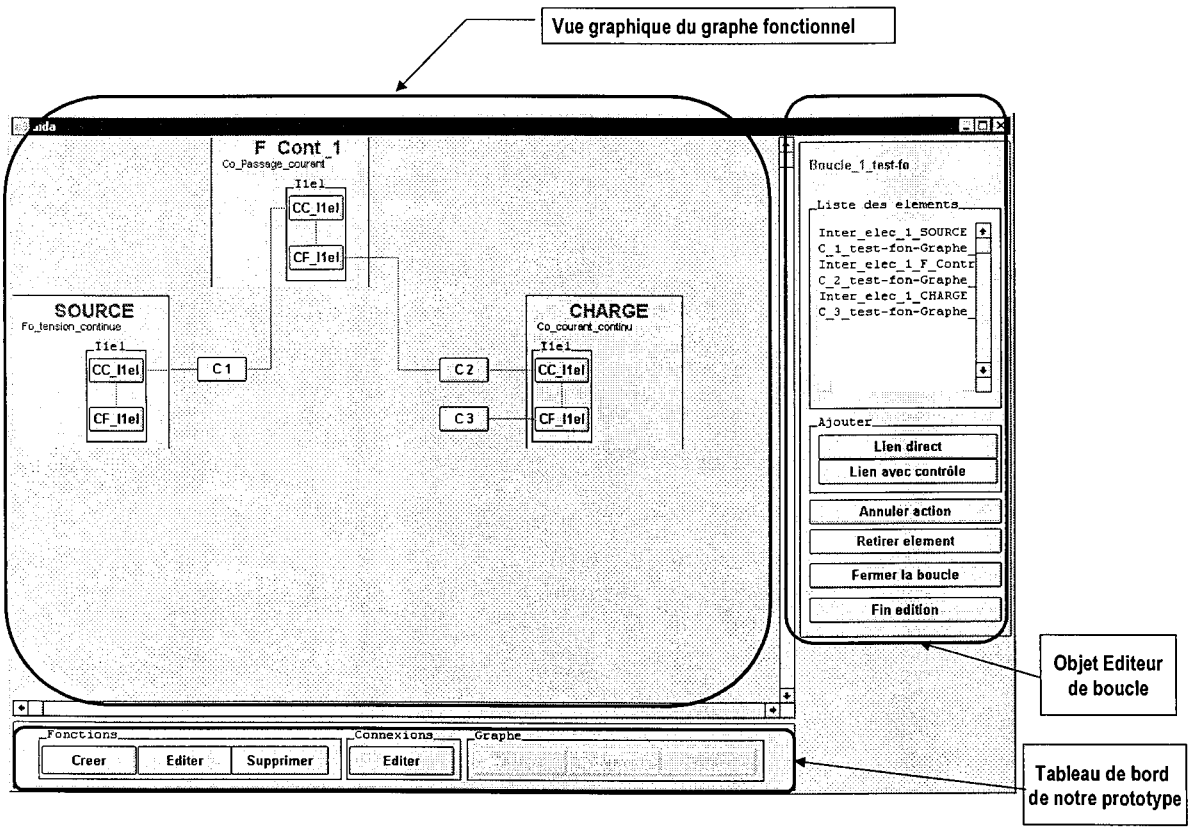



**Figure 4.16 : L'I.H.M. attachée au graphe fonctionnel pour l'édition des boucles**

## III. Evolution de l'implantation

Nous avons maintenant un premier prototype qui nous permet de valider l'utilisation de l'analyse fonctionnelle sur les convertisseurs statiques. Celui-ci nous offre un environnement qui permet les opérations suivantes.

- **Création et édition des fonctions.** Celles-ci peuvent être définies suivant la procédure dynamique présentée dans ce chapitre et être modifiées de façon interactive par l'utilisateur.

- **Création et édition des boucles.** Il est possible de décrire toutes les boucles de flux, de façon interactive. Le prototype s'assure que les boucles respectent nos conditions de description.

- **Construction des boucles implicites.** Nous appliquons pour cela l'algorithme présenté au chapitre 3.

- **Vérification des boucles en terme de causalité.** Notre prototype permet de vérifier les boucles sans fonction gyrateur.

Il reste maintenant à définir les procédures pour l'aide à la détermination du mode de fonctionnement et de la commande.
L'environnement ILOG LE-LISP répond bien à nos besoins mais nous avons maintenant l'obligation de changer d'environnement de développement pour réaliser les évolutions futures de notre prototype. En effet, cet atelier de logiciels est maintenant un produit dont la pérennité est menacée. Il nous paraît plus judicieux d'utiliser l'expérience acquise sur la construction de ce premier prototype pour réaliser une autre version dans un environnement différent. Deux choix se proposent à nous :

- **L'environnement ILOG C++** : Il offre une structure similaire à l'environnement ILOG-LE-LISP, mais il est plus récent et il propose plusieurs modules écrits autour du langage C++. Parmi ceux-ci, nous pouvons noter les éléments suivants.

  - **Un module graphique ILOG Views.** Celui propose un moteur graphique (plus performant qu'AIDA) avec des outils de construction visuels (similaire à MASAI),

  - **Un module de gestion de règles ILOG Rules.** Celui-ci propose un formalise de règles qui permet d'implanter des règles d'expertise dans un programme.

La transposition de la gestion dynamique du modèle peut nous sembler problématique avec le langage C++. En effet, ce langage n'est que difficilement dynamique, donc la mise en œuvre d'une structure similaire à la notre peut demander des développements très lourds.

- **Le langage Java.** Le langage Java nous paraît une alternative intéressante parce qu'il est complètement objet et permet une certaine dynamicité [RAI98]. Il permet par exemple la gestion dynamique des classes d'objets. Cependant JAVA est un langage jeune, donc en constante évolution, et il paraît encore difficile de trouver des environnements robustes pour ce langage qui permettent la construction visuelle d'interfaces graphiques ou la gestion de règles d'expertise.

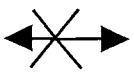
Ce chapitre a présenté les éléments importants de la construction de notre premier prototype. Nous allons maintenant exposer des exemples de définition de structure en utilisant la démarche fonctionnelle.

# Chapitre 5 : Des exemples de conception fonctionnelle

Ce chapitre a pour objectif de présenter autour d'exemples, comment l'analyse fonctionnelle peut servir pour la conception de structures. Dans un premier temps, nous examinerons la conception d'une structure avec isolation galvanique, puis nous présenterons l'utilisation de notre prototype sur un exemple plus simple.

## 0. Préliminaire : notations

Dans ce chapitre, nous utiliserons les notations présentées dans le tableau 5.1 pour décrire les graphes fonctionnels.

| | |
|---|---|
| | Lien de contrainte énergétique : XX peut prendre les valeurs suivantes :<br>E > 0 : L'effort est imposé positif — F > 0 : Le flux est imposé positif<br>E < 0 : L'effort est imposé négatif — F < 0 : Le flux est imposé négatif<br>E ~ : L'effort est imposé alternatif — F ~ : Le flux est imposé alternatif<br>$\frac{dE}{dt}$ : l'effort doit donc être continu — $\frac{dF}{dt}$ : Le flux doit être continue<br>I : La causalité est du type Interrupteur. — Q : La causalité est du type quelconque |
| | Couple EF, ou une de ses classes dérivé : YY peut prendre les valeurs suivantes<br>EF : Couple_EF<br>VI : Couple_VI<br>FD : Couple_FmmDfm |
| | Interface |
| | Lien de transfert d'énergie (LTE) avec indication du sens positif de parcours du flux |
|  | Lien de non-simultanéité (LNS) |
| | Lien de simultanéité (LS) |
| | Lien séquentiel (Lseq) |

| | |
|---|---|
| FCi | Fonction de contrôle du flux numéro i |
| FRi | Fonction de retour (du flux) numéro i, pour fermer une boucle |
| GY_i | Fonction gyrateur numéro i |
| Ci | Connexion numéro i du graphe externe aux fonctions |
| CIi | Connexion numéro i du graphe interne aux fonctions |

**Tableau 5.1 : Les notations utilisées dans les graphes fonctionnels**

# I. Exemple de conception d'une structure avec isolation galvanique

Le but de cet exemple est de présenter une démarche de conception d'une structure avec isolation. La première phase est la définition du cahier des charges en vue du processus de conception fonctionnel.

## I.1. Cahier des charges

Dans cet exemple nous cherchons à définir un convertisseur ayant les caractéristiques suivantes.

- **L'énergie est fournie par une source de tension.** Celle-ci peut être modélisée par une fonction avec une fonctionnalité Source et une contrainte sur l'effort (imposé à une valeur constante).

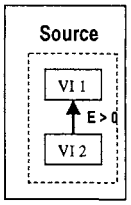



**Figure 5.1 : La formulation du cahier des charges, la fonction source**

- **Le convertisseur doit assurer une isolation galvanique.** Cette fonctionnalité peut être remplie en ajoutant une fonction isolation définie en utilisant la technologie magnétique. Ce choix de technologie s'avère nécessaire pour pouvoir définir les contraintes externes de cette fonction. Cette dernière est modélisée à l'aide de deux fonctions « gyrateur » et une fonction « capacité magnétique » pour prendre en compte les effets du circuit magnétique (voir § ch1 III.5.2). Ce choix est un exemple typique d'utilisation de la technologie pour guider la définition des fonctions.

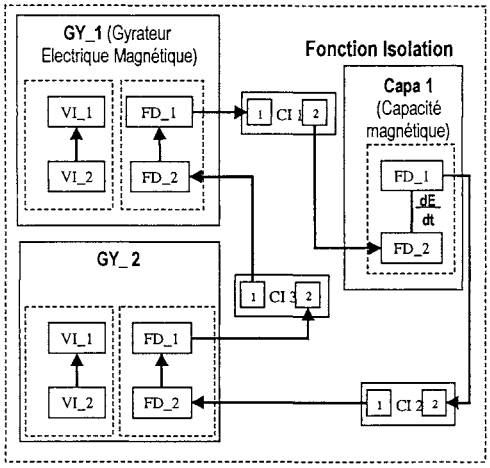



**Figure 5.2 : La formulation du cahier des charges, la fonction isolation**

- **Le convertisseur doit fournir une tension continue quelle que soit la charge.** Cette condition nous permet de déterminer l'interface de sortie de notre convertisseur. D'un point de vue fonctionnel, nous pouvons assurer que cette contrainte est garantie en ajoutant une fonction « transformation de causalité » (notée Transformation). Il est alors possible de maîtriser les caractéristiques d'entrée de cette dernière afin d'obtenir une expression simple des échanges énergétiques de notre convertisseur. La figure 5.3 représente par exemple deux cas possibles. Dans toute la suite, les assemblages « transformation1-charge » et « transformation2-charge » seront remplacés par les fonctions « charge en courant » et « charge en tension ».

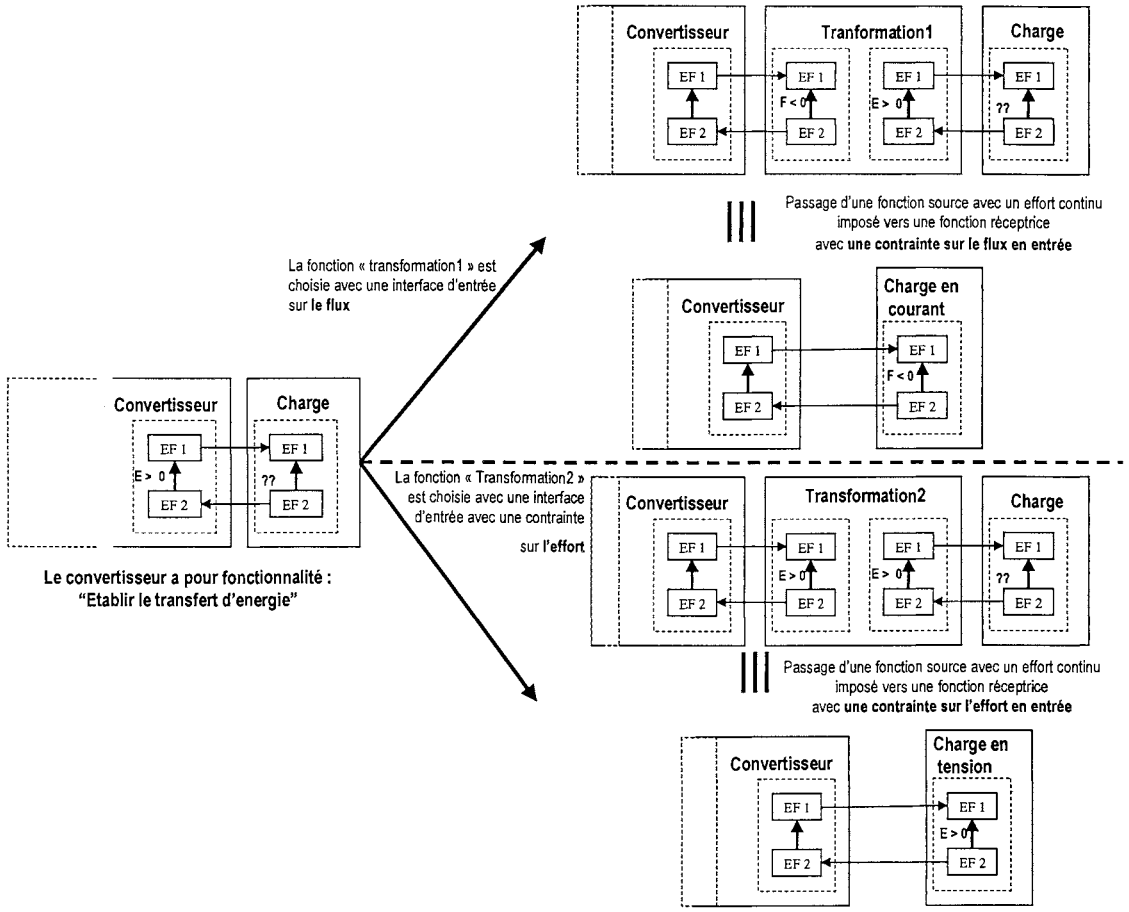


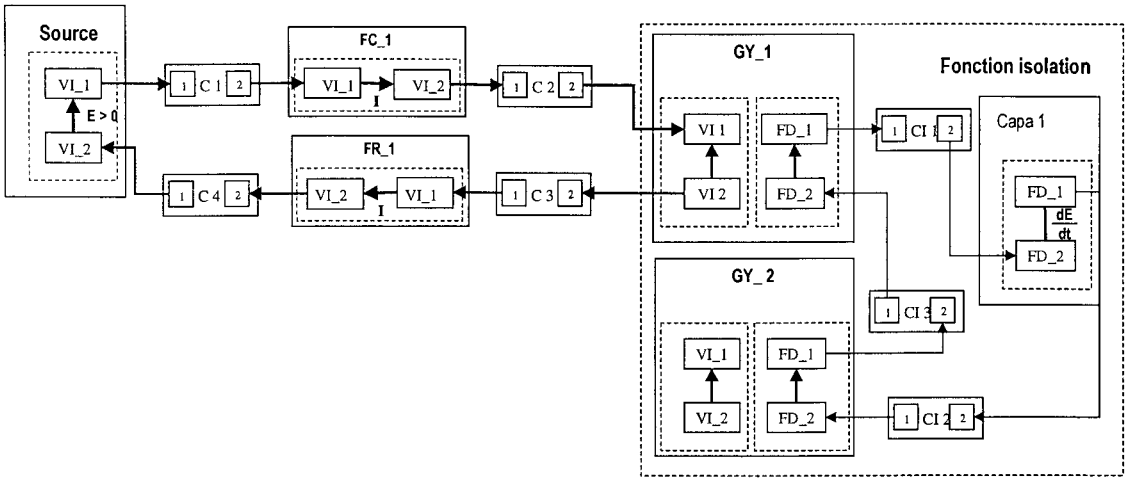

**Figure 5.3 : La formulation du cahier des charges, les caractéristiques de sortie**

## I.2. La construction du graphe fonctionnel

Nous devons maintenant définir le graphe fonctionnel relatif à notre convertisseur. Ce processus se décompose en plusieurs étapes que nous allons présenter ci-dessous.

### I.2.1. Etape 1 : La liaison entre la fonction « source » et la fonction « isolation ».

Cette étape permet de construire une liaison énergétique entre la fonction « source » et la fonction « isolation ». Ceci se fait en établissant une boucle de flux entre l'interface de la fonction « source » et une interface électrique d'une des fonctions « Gyrateur Electrique Magnétique » de la fonction « isolation ». Parce que la liaison doit contrôler la quantité d'énergie transférée, nous ajoutons une fonction de « contrôle de flux » (FC1). En complément, une fonction « de retour » (FR1) est insérée afin de fermer la boucle (cf. ch2 : II.3.4). La figure 5.4 montre le graphe fonctionnel obtenu après cette étape.




Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

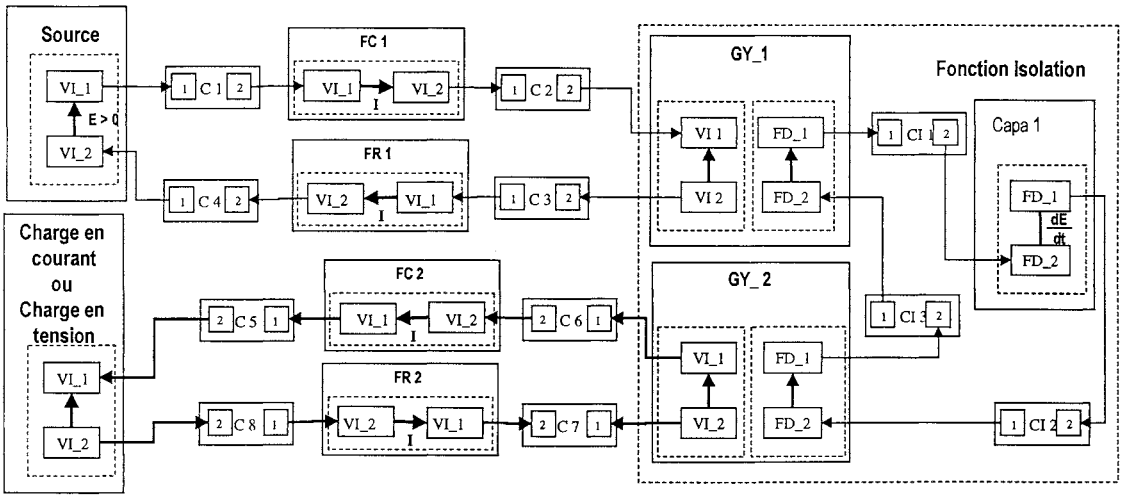
Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

**Figure 5.4 : Le graphe fonctionnel après l'étape 1**

### I.2.2. Etape 2 : La liaison entre la fonction « isolation » - et la fonction « charge ».

Au cours de la deuxième étape, nous réalisons la deuxième liaison énergétique de notre convertisseur. Nous devons construire une boucle énergétique (noté Boucle 2) entre la fonction « gyrateur magnétique 2 » (GY_2) et la fonction « charge » (qui est en réalité la fonction « charge en courant » ou la fonction « charge en tension »). Cette fois-ci nous avons deux choix possibles pour la réalisation de cette boucle qui est obtenue par l'introduction d'une fonction de contrôle du flux (FC2) et d'une fonction de retour (FR2).

- Soit la Boucle 2 est construite pour que le flux entre dans la fonction « charge » par le couple_VI VI_1. Le graphe fonctionnel obtenu est décris sur la figure 5.5.



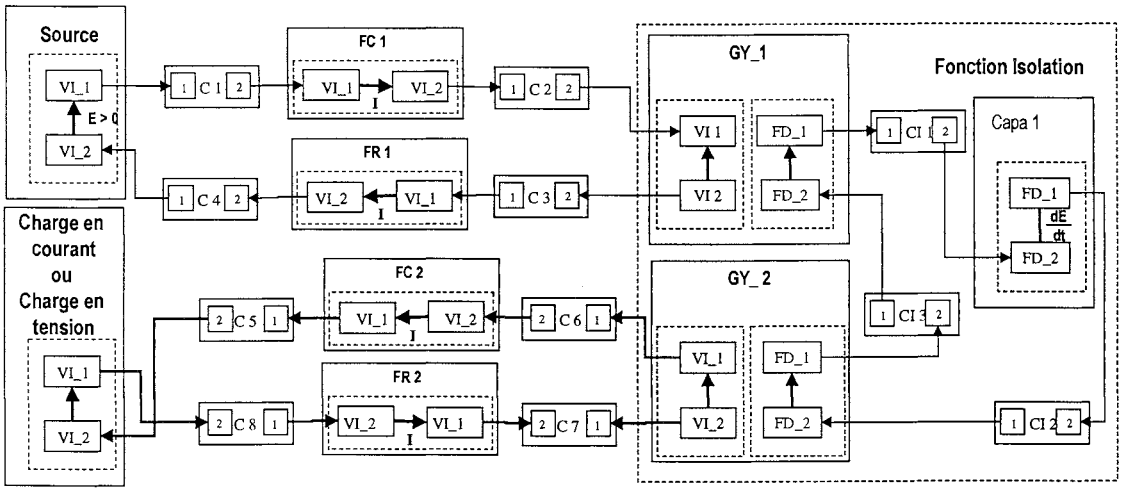

Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

Boucle 2 : GY_2 VI_2-VI_1, C6, FC 2 VI_2-VI_1, C5, **Charge ???? VI_1-VI_2**, C8, FR 2 VI_2-VI_1, C7

**Figure 5.5 : Etape 2, introduction de la boucle 2, le graphe fonctionnel dans le cas 1**

- Soit la Boucle 2 est construite avec le flux entrant dans la fonction charge part le couple_VI VI2. Le graphe fonctionnel obtenu correspond à la figure 5.6.




Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

Boucle 2 : GY_2 VI_2-VI_1, C6, FC 2 VI_2-VI_1, C5, **Charge ??? VI_2-VI_1**, C8, FR 2 VI_2-VI_1, C7

**Figure 5.6 : Etape 2, introduction de la Boucle 2, le graphe fonctionnel dans le cas 2**

A ce stade de la conception nous disposons de quatre graphes fonctionnels possibles (deux topologies avec deux charges possibles). Seul une analyse de cohérence en terme de causalité nous permettra de discerner quelles sont les bonnes solutions.

### I.2.3. Etape 3.1 : Analyse de la cohérence avec la fonction « charge en courant »

Nous allons maintenant simuler le fonctionnement du processus de vérification de la cohérence en terme de causalité pour les topologies possibles de notre problème. La fonction « charge » de notre graphe fonctionnel est ici la fonction « charge en courant » (avec une contrainte sur le flux).

#### *I.2.3.1. La contrainte sur le flux de la charge en courant*

Le courant sur la fonction « charge en courant » ne doit pas être interrompu (ceci est matérialisé par la contrainte sur le flux). Cette fonction ne peut appartenir qu'à la Boucle 2 puisque que le flux peut être interdit par la fonction FC 2. Il est donc nécessaire de créer une nouvelle boucle (notée Boucle 3) qui permettra d'assurer la continuité du courant. La complémentarité du fonctionnement est modélisée par le lien séquentiel entre les boucles. La figure 5.7 montre l'introduction de la Boucle 3 dans le cas de la topologie du cas 1. Cette boucle est obtenue par l'ajout d'une fonction de contrôle du flux FC3 et d'une fonction de retour FR3. Une analyse similaire peut être menée avec la topologie du cas 2.

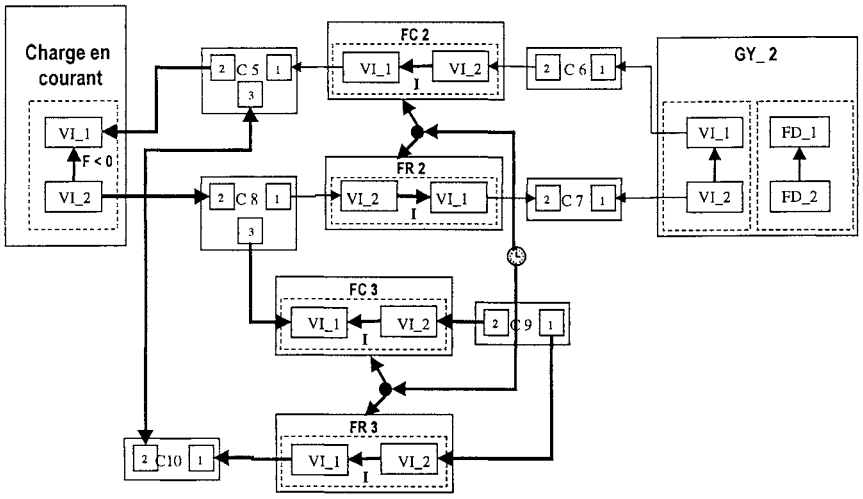
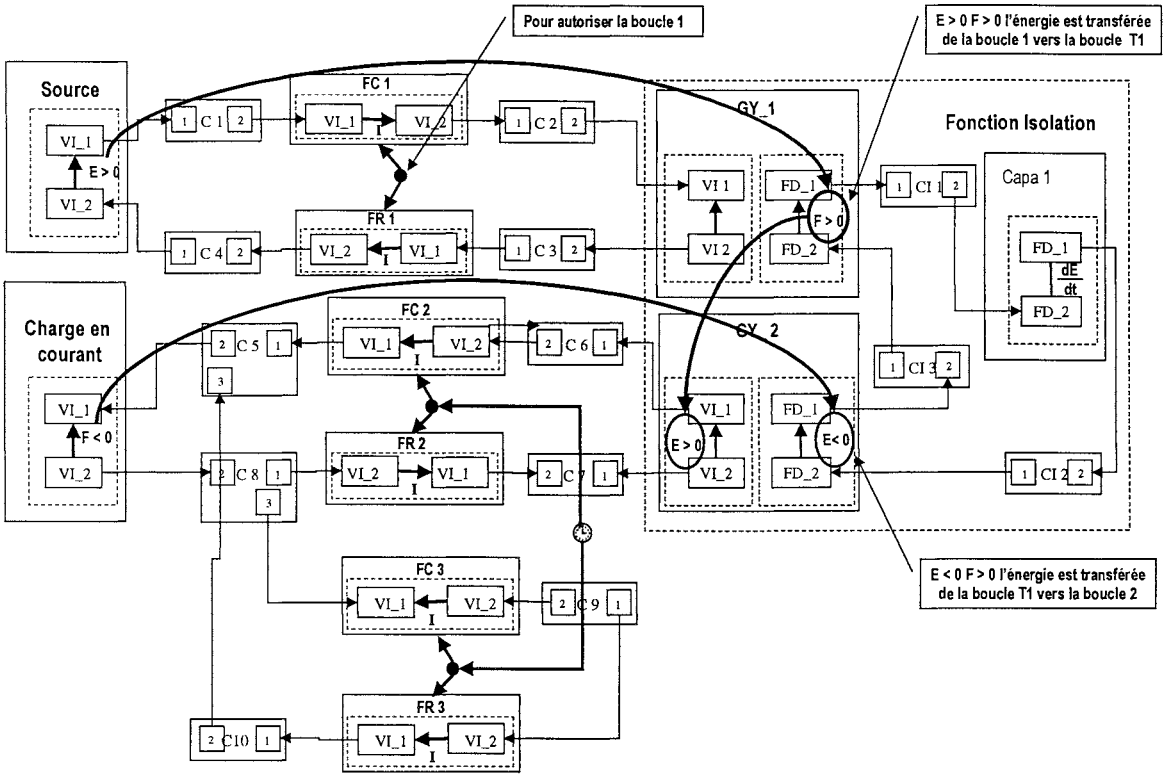



**Figure 5.7 : La Boucle 3 pour assurer la contrainte de flux sur la charge en courant**

#### *I.2.3.2. Analyse de la Boucle T1 avec la topologie du cas 1*

Ici, l'analyse de causalité de la Boucle T1 s'effectue par propagation des contraintes entre les boucles (voir figure 5.8).




Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

Boucle 2 : GY_2 VI_2-VI_1, C6, FC 2 VI_2-VI_1, C5, **Charge en courant VI_1-VI_2**, C8, FR 2 VI_2-VI_1, C7

Boucle 3 : **Charge en courant VI_1-VI_2**, C8, FC 3 VI_1-VI_2, C9, FR 3 VI_2-VI_1, C10

**Figure 5.8 : L'analyse de la Boucle T1 avec la charge en courant dans le cas 1**

- La Boucle 1 comporte une fonction « gyrateur » et une fonction avec une contrainte tension imposée : les fonctions de « contrôle du flux » et de « retour du flux » sont passives. La causalité dominante de cette boucle est du type Tension ; donc par propagation, la fonction GY_1 impose une contrainte sur le flux dans la boucle T1.

- La Boucle 2 comporte une seule fonction avec une contrainte de flux. Par propagation, la fonction GY_2 impose une contrainte effort sur la boucle T1.

- La Boucle T1 est donc viable puisqu'elle ne comporte qu'une fonction imposant le flux. Le fonctionnement simultané de la Boucle 1 et de la Boucle 2 est possible. Dans ce cas de figure, un transfert d'énergie est possible puisque la fonction « gyrateur » GY_2 fonctionne en extracteur d'énergie (transfert de la Boucle T1 vers la Boucle 2) alors que GY 1 transfère l'énergie de la Boucle 1 vers la Boucle T1.

La circulation d'un flux dans la Boucle T1 va provoquer l'accumulation d'effort dans la capacité magnétique (donc une élévation de son niveau effort). Il faut donc prévoir une boucle permettant la restitution de cet effort accumulé en permettant la circulation d'un flux négatif dans la Boucle T1. Plusieurs solutions s'offrent à nous pour résoudre ce problème.

- Nous pouvons ajouter une fonction de « déstockage de l'effort » (structurellement de démagnétisation) qui permet l'application d'un effort particulier sur l'un des gyrateurs pour provoquer la restitution de l'effort accumulé par la capacité magnétique. La figure 5.9 présente un exemple.

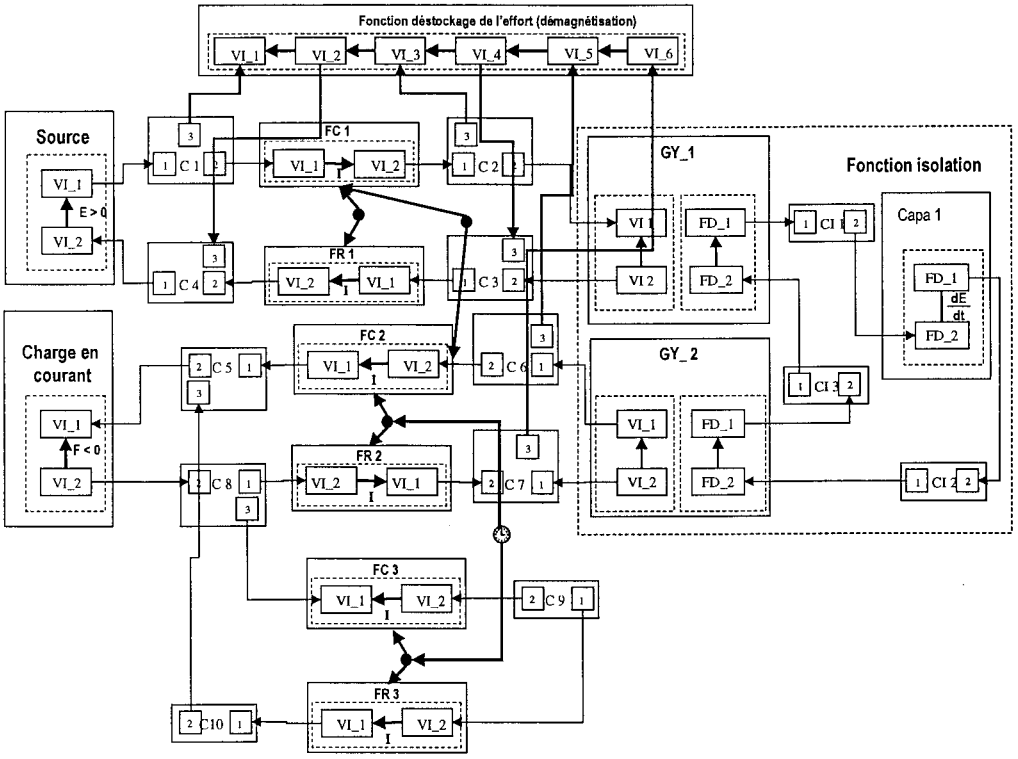
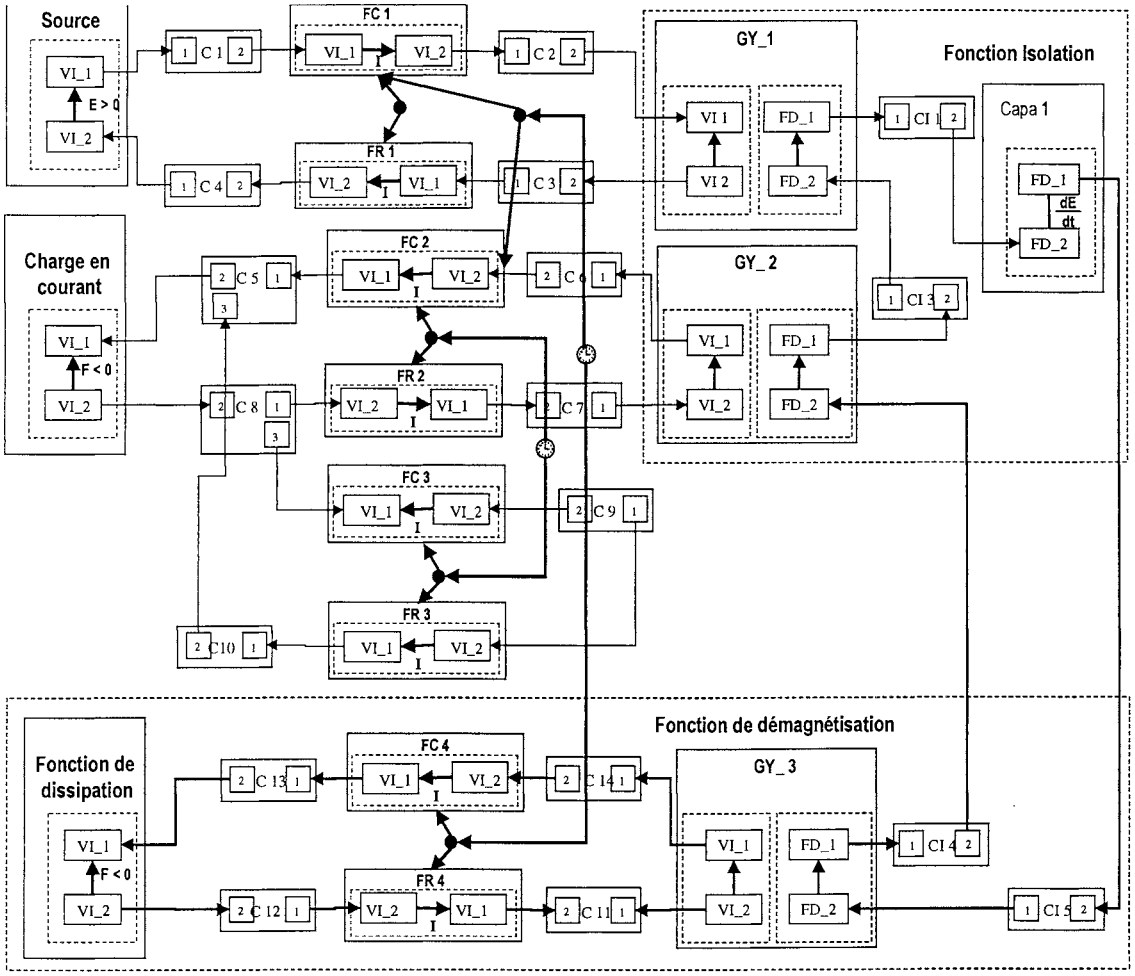



**Figure 5.9 : L'utilisation d'une fonction de démagnétisation**

**Remarque : La fonction de « destockage de l'effort » a une interface de type quelconque, parce que la technique de réalisation de cette fonction peut demander des connexions multiples dans notre graphe fonctionnel.**

- Il est possible d'ajouter une autre fonction gyrateur associée à une boucle commandée (notée Boucle 4) de façon judicieuse pour provoquer le destockage de l'effort dans la fonction capa1. Cette boucle est construite à partir d'une fonction de contrôle du flux (FC4), d'un gyrateur magnétique-électrique (GY_3) et d'une fonction de retour (FR4). La figure 5.10 décrit le nouveau graphe fonctionnel dans ce cas. La Boucle 4 permet de provoquer un flux en sens inverse par rapport au flux créé par la Boucle 1. Afin de signaler le fonctionnement complémentaire de ces deux ensembles, nous plaçons un lien séquentiel entre la Boucle 1 et la Boucle 4.




Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

Boucle T2 : GY_3 FD2-FD1, CI 4, GY_2 FD_2-FD_1, CI 3, GY 1 FD_2-FD_1, CI 1, Capa 1 FD_2-FD_1, CI 5

Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

Boucle 2 : GY_2 VI_2-VI_1, C6, FC 2 VI_2-VI_1, C5, **Charge en courant VI_1-VI_2**, C8, FR 2 VI_2-VI_1, C7

Boucle 3 : **Charge en courant VI_1-VI_2**, C8, FC 3 VI_1-VI_2, C9, FR 3 VI_2-VI_1, C10

Boucle 4 : Fonction de dissipation VI_1-VI_2, C12, FR_4 VI_2-VI_1, C11, GY_3 VI_2-VI_1, C14, FC_4 VI_2-VI_1, C13

**Figure 5.10 :La démagnétisation en utilisant une boucle de flux auxiliaire et une fonction de « dissipation »**

### *I.2.3.3. Avec la topologie du cas 2*

Une analyse similaire peut être conduite sur cette topologie. Ici, les deux fonctions « gyrateur » apportent de l'énergie vers la boucle T1 en même temps, donc on ne crée pas de transfert d'énergie, ce qui nous fait rejeter cette solution. La figure 5.11 nous permet donc de conclure que **ce graphe fonctionnel n'est pas viable.**

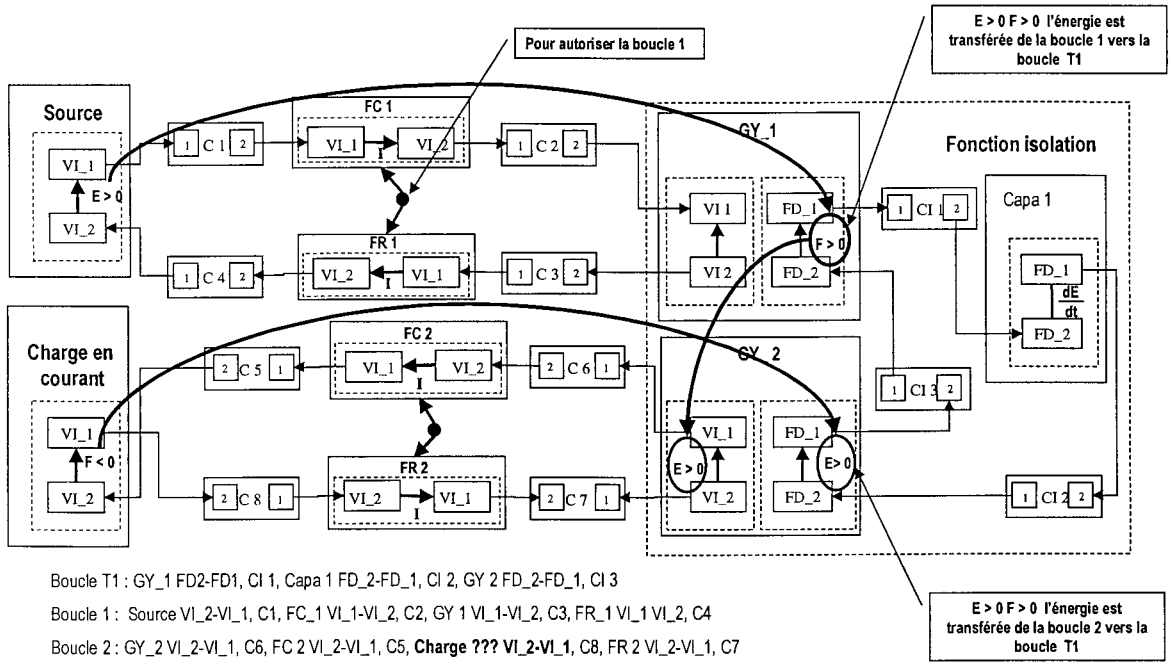



**Figure 5.11 : L'analyse de la Boucle T1 avec la charge en courant dans le cas 2**

## I.2.4. Etape 3.2 : Analyse de la cohérence avec la charge en tension

Nous nous préoccupons maintenant de la deuxième fonction « charge » possible de notre cahier des charges : la fonction « charge en tension » avec une contrainte sur l'effort.

### *I.2.4.1. Analyse de la Boucle T1 avec la topologie du cas 1*

Le processus de vérification de la cohérence commence par vérification de la Boucle 1 et la Boucle 2 afin de déterminer leur causalité dominante.

- La causalité dominante de la Boucle 1 est Effort puisqu'elle ne contient qu'une fonction avec une contrainte effort.

- La causalité dominante de la Boucle 2 est Dérivée_effort puisqu'elle ne contient qu'une fonction avec une contrainte Dérivée_effort.

Donc si ces deux boucles sont actives en même temps, la Boucle I1 se retrouve en conflit de type Flux puisqu'elle comporte deux contraintes sur le flux. Il est donc préférable d'éviter ce conflit potentiel en interdisant le fonctionnement simultané de ces boucles par un lien de non simultanéité entre ces deux boucles. Le graphe fonctionnel peut alors être représenté par la figure 5.12.

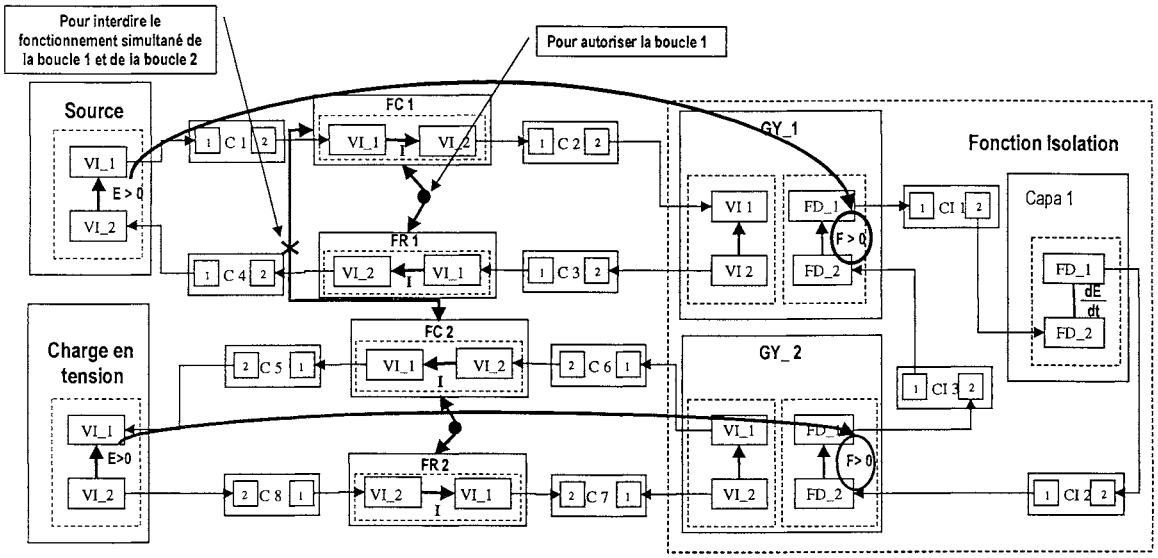



**Figure 5.12 : La cohérence de la Boucle T1 avec la charge en tension dans le cas 1**

Nous devons maintenant vérifier la contrainte de continuité sur la capacité magnétique capa1. Celle-ci nous impose une contrainte de continuité sur sa variable effort. Ici nous avons deux cas possibles :

- Lorsque la Boucle 1 est active (FC 1 et FR 1 autorisent le passage d'un flux), le flux est positif dans la Boucle T1.

- Lorsque la Boucle 2 est active, le flux est encore positif dans la Boucle T1.

Donc la « capacité magnétique » est toujours parcourue par un flux positif, ce qui provoquera une accumulation d'énergie et en final une saturation nuisible pour le système. **Cette structure n'est donc pas viable.**

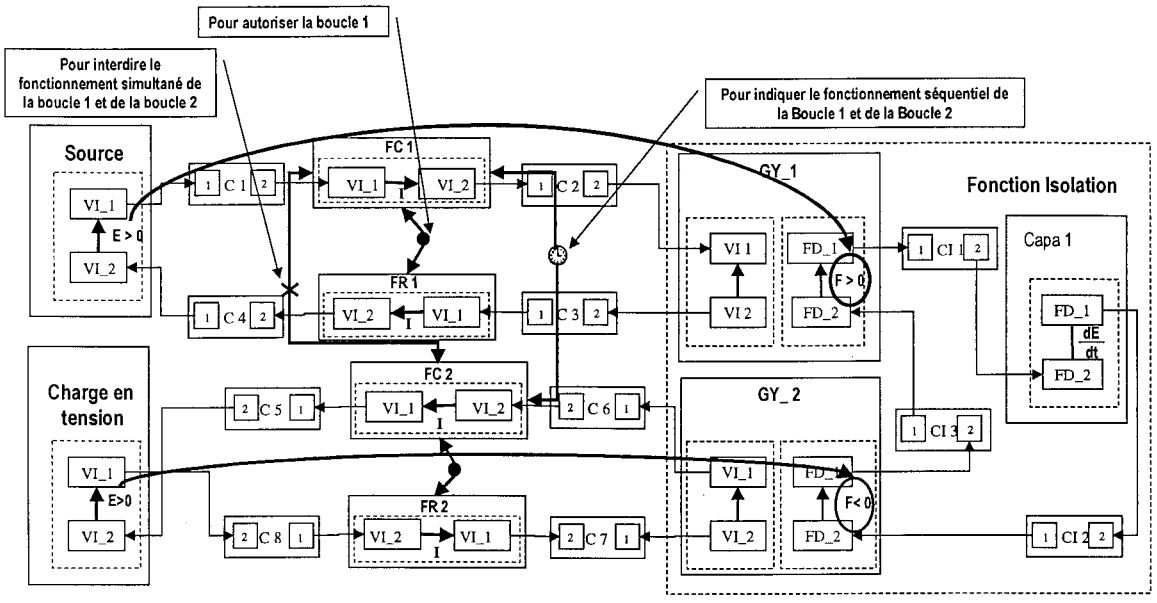
### *I.2.4.2. Analyse de la Boucle T1 avec la topologie du cas 2*

Nous pouvons mener une analyse similaire avec une analyse de cohérence à partir de la topologie du cas 2. Avec cette topologie, la boucle T1 peut être parcourue par des flux alternatifs :

- Lorsque la Boucle 1 est active (FC 1 et FR 1 autorisent le passage d'un flux), le flux est positif dans la Boucle T1.

- Lorsque la Boucle 2 est active, le flux est par contre négatif dans la Boucle T1.

Pour que cette structure soit viable, la Boucle 1 et la Boucle 2 ne peuvent pas être activées en même temps (dans le cas contraire, il y a un conflit flux sur la boucle T1). Pour matérialiser cette contrainte, il suffit placer un lien séquentiel pour garantir la contrainte de continuité comme le montre la figure 5.13.




Boucle T1 : GY_1 FD2-FD1, CI 1, Capa 1 FD_2-FD_1, CI 2, GY 2 FD_2-FD_1, CI 3

Boucle 1 : Source VI_2-VI_1, C1, FC_1 VI_1-VI_2, C2, GY 1 VI_1-VI_2, C3, FR_1 VI_1 VI_2, C4

Boucle 2 : GY_2 VI_2-VI_1, C6, FC 2 VI_2-VI_1, C5, **Charge ??? VI_2-VI_1**, C8, FR 2 VI_2-VI_1, C7

**Figure 5.13 : La cohérence de la Boucle T1 avec la charge en tension dans le cas 2**

## I.3. La construction des solutions structurelles

La dernière étape est la construction des solutions structurelles. Cette opération consiste principalement à spécifier les différentes fonctions en structures.

- Les fonctions « transformation » peuvent être spécifiées en structures réelles (des filtres) à base d'inductances et de condensateurs.

- Lorsque deux fonctions de « contrôle du flux » sont liées par un Lien_séquentiel l'une des deux est une diode.

- Dans notre cas, il n'y a pas d'interaction entre les boucles (et pas de boucle implicite), l'utilisation des fonctions de retour est inutile, celles-ci seront remplacées par un simple fil.

### I.3.1. Une solution structurelle avec la charge en courant

Nous utilisons le graphe fonctionnel de la figure 5.9 comme point de départ de la construction de la solution structurelle.

La première étape est la spécification des fonctions de contrôle en accord avec les contraintes. Ici la Boucle 1 d'un point de vue fonctionnel est parcourue par un flux (courant) positif ou nul. La fonction de contrôle FC_1 peut être spécifiée par une fonction à deux segments : segment 1 U > 0 I=0 ; segment 2 U = 0 I > 0 (voir figure 5.14). En conséquence, les fonctions FC_2 et FC_3 doivent changer d'état automatiquement pour respecter les liens séquentiels.

L'analyse des efforts et des flux permet de vérifier qu'un interrupteur qui correspond structurellement à cette fonction est une diode (segment 2 U= 0 I > 0, segment 3 U < 0 I= 0).

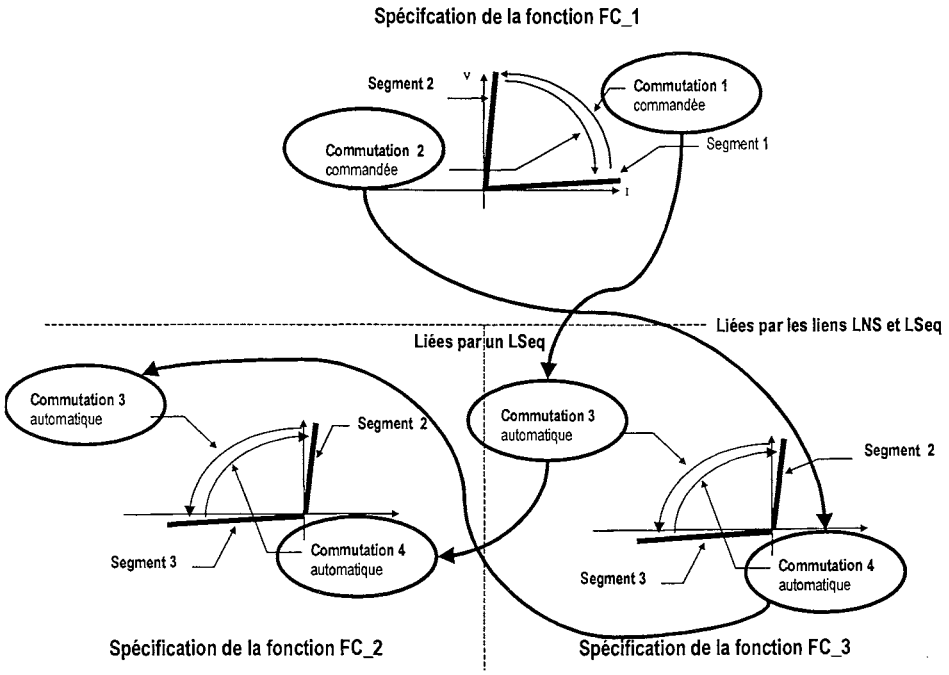



**Figure 5.14 : La représentation des fonctions de contrôle**

Remarque : avec notre approche nous retrouvons les cellules de commutation (FC_1, FC_2) et (FC_2, FC3).

A partir de ces interrupteurs de base, nous pouvons maintenant construire la solution structurelle en effectuant les choix technologiques suivants.

- **Les commutations sont douces.** La fonction de contrôle est spécifiée comme étant un interrupteur associé à un circuit résonnant. La figure 5.15 présente la décomposition fonctionnelle de la fonction FC 1. Cette technologie permet un fonctionnement à haute fréquence, avec des pertes réduites. Suivant la nature du contrôle de l'interrupteur résonnant, nous obtenons les deux solutions de la figure 5.16.

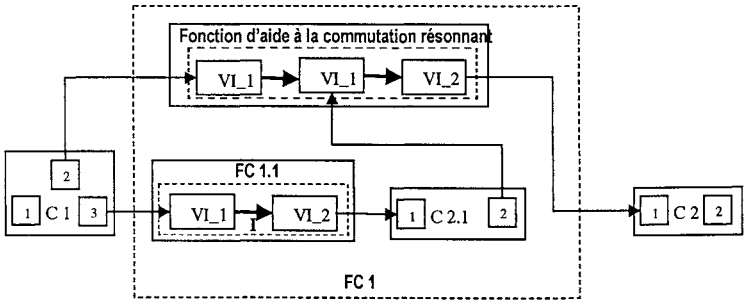



**Figure 5.15 : La décomposition fonctionnelle de la fonction FC 1**

- **La démagnétisation se fait par résonance.** Nous avons vu qu'il est nécessaire d'appliquer un effort particulier pour forcer un flux négatif dans la boucle T1. Nous proposons d'utiliser un circuit résonant composé d'un condensateur avec une valeur permettant la résonance avec l'inductance de fuite du transformateur pour provoquer ce phénomène.

- **La fonction FC_1 est rendue bidirectionnelle en courant (par une diode).** Ce choix nous permet d'envisager l'utilisation d'un interrupteur résonnant commandable au blocage et d'étendre les possibilités de fonctionnement de la structure. Dans ce cas la source peut accepter une inversion du courant.

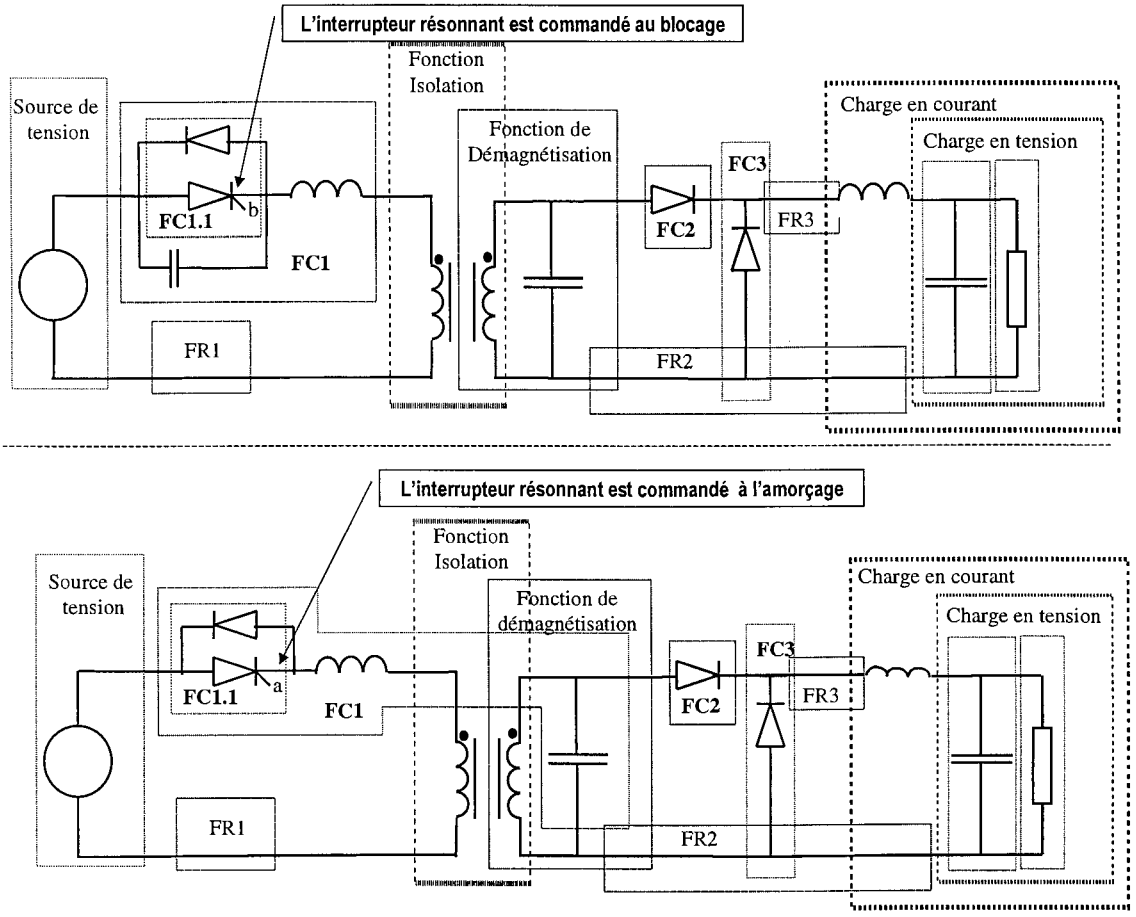



**Figure 5.16 : Quelques solutions structurelles avec la charge en courant**

A ce stade de la conception, il est nécessaire d'effectuer une étude fonctionnelle de la commande pour pouvoir spécifier correctement les interrupteurs. Par exemple l'interrupteur uniquement commandable au blocage n'existe pas en production industrielle. Souvent, il est construit à partir d'un interrupteur classique (un IGBT par exemple) associé à une commande particulière.

### I.3.2. Une solution structurelle avec la charge en tension

Il est possible de mener le même raisonnement pour le graphe fonctionnel de la figure 5.13. Nous obtenons avec la charge en tension la solution structurelle de la figure 5.17.

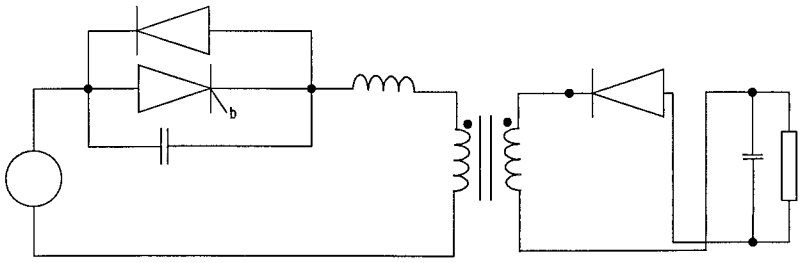


**Figure 5.17 : une solution structurelle avec la charge en tension**

Cet exemple montre qu'il est possible d'obtenir, l'aide de notre méthode d'analyse fonctionnelle, les principales structures avec isolation galvanique. De plus, le graphe fonctionnel permet de déduire facilement le mode de fonctionnement du convertisseur. Nous pouvons déterminer automatiquement la nécessité d'une fonction de démagnétisation ou gérer les interactions entre les enroulements.

Dans sa version actuelle, pour des problèmes d'état d'avancement de l'implantation du modèle d'analyse fonctionnelle, notre prototype ne permet pas d'analyser des exemples aussi complexes, mais nous allons maintenant montrer son fonctionnement sur un exemple plus simple.

## II. Exemple d'un convertisseur continu – continu

### II.1. Cahier des charges

Le cahier des charges de cet exemple consiste en la conception d'un convertisseur en une source de tension continue vers une charge avec une tension imposée (une batterie par exemple). Il peut se formuler sous la forme de la figure 5.18

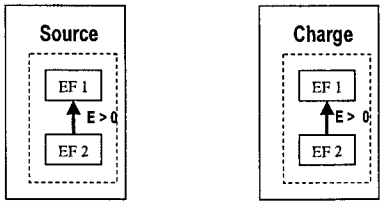



**Figure 5.18 : Le cahier des charges**

Ce cahier des charges peut se définir à l'aide de notre prototype en utilisant son I.H.M. (voir figure 5.19).

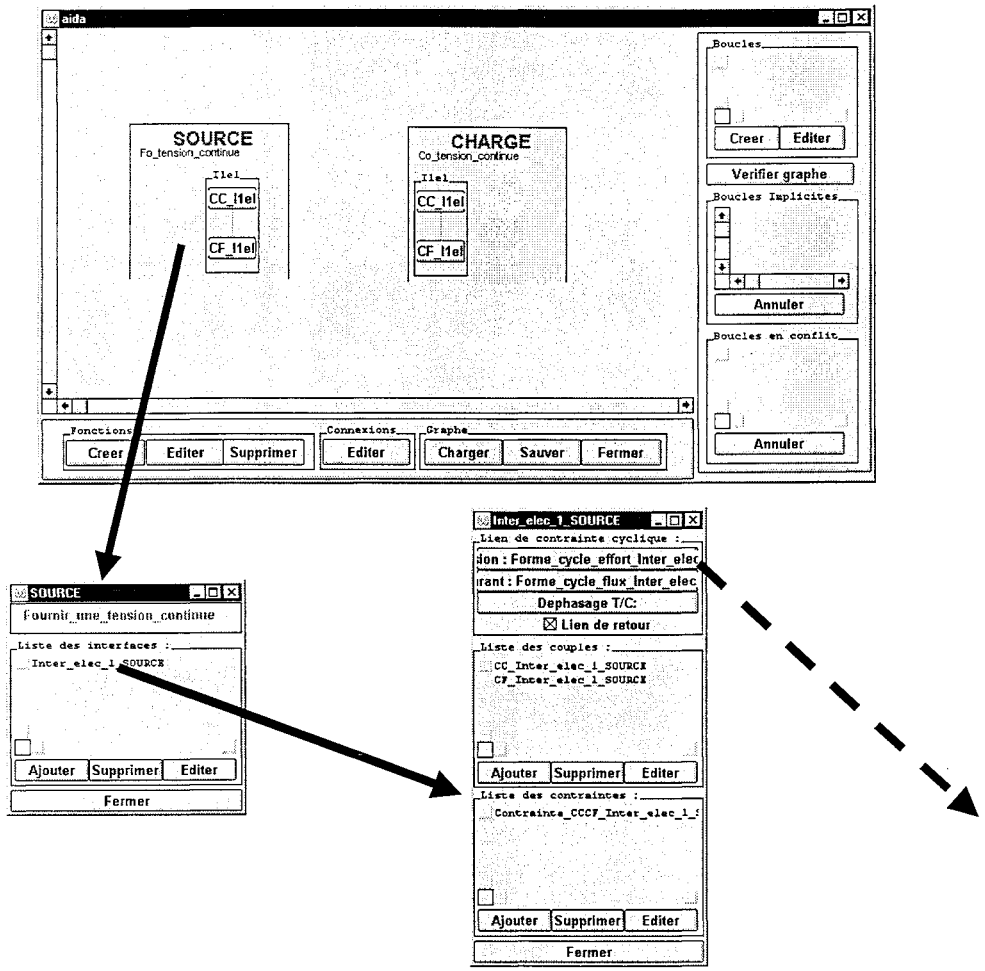



**Figure 5.19 :La représentation par l'I.H.M. des fonctions du cahier des charges**

- **La fonction source est définie avec la fonctionnalité « Fournir une tension continue ».** Celle-ci est associée à une interface bipolaire avec une causalité effort, les objets forme_d_onde sont définis comme le montre la figure 5.20.

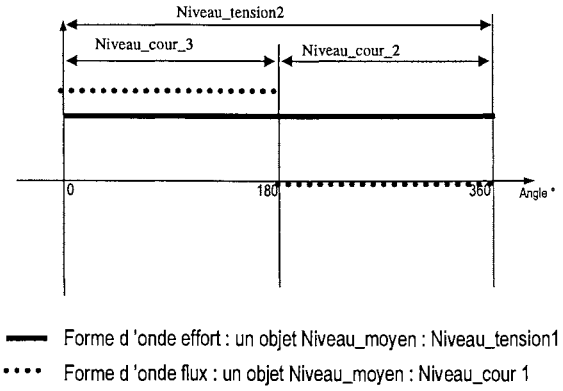



**Figure 5.20 : La définition des objets Forme_d_onde**

- **La fonction charge est définie avec la fonctionnalité « Consommer une tension continue ».** De la même façon, l'interface associé est bipolaire avec une causalité effort et les objets Forme_d_onde sont définis de manière semblable (voir la figure 5.20.)

## II.2. La construction du graphe fonctionnel

A partir de la définition du cahier des charges présenté précédemment, nous pouvons maintenant construire étape par étape le graphe de notre exemple en utilisant notre prototype.

La construction de cette boucle se fait de façon interactive (en utilisant l'interface graphique) en quatres étapes (voir figure 5.21).

**1. Un objet Boucle est créé.** L'utilisateur doit alors sélectionner l'objet Couple_EF de départ pour construire la boucle. Dans notre cas, il sélectionne l'objet Couple_EF nommé CC_Iel1 de la fonction SOURCE

**2. Ajout d'une fonction de contrôle du flux à la boucle.** Cela revient à créer une fonction de contrôle de flux (F Cont 1)et à la connecter à la boucle en cours de construction.

**3. Connexion de la fonction charge en ajoutant le couple EF nommé CC_Iel1 de la fonction CHARGE**

**4. L'utilisateur demande à outil de fermer la boucle.**

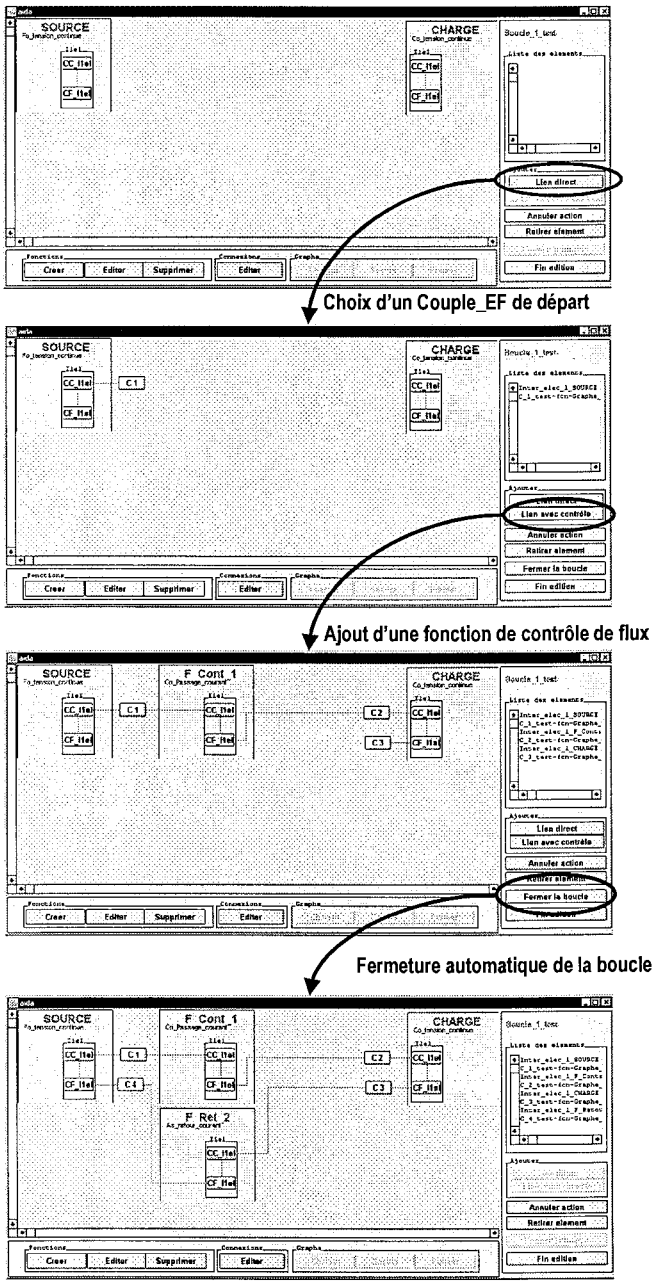



**Figure 5.21 : La construction d'une boucle avec notre prototype**

A partir de cette boucle, nous pouvons vérifier sa viabilité d'un point de vue causalité. L'interface graphique de notre outil signale un problème de causalité en affichant la fenêtre de la figure 5.22.

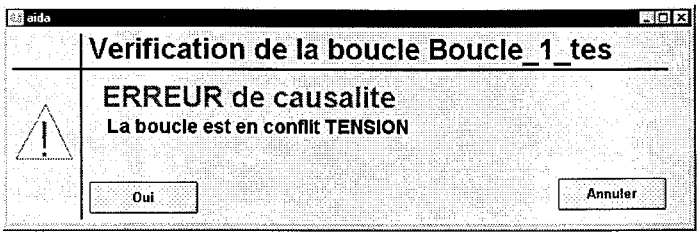



**Figure 5.22 : Une fenêtre signalant une erreur de causalité**

En effet, la boucle comporte deux fonctions avec une causalité tension. Il est possible de corriger ce problème en ajoutant une inertie dans la boucle. La figure 5.23 présente le résultat.

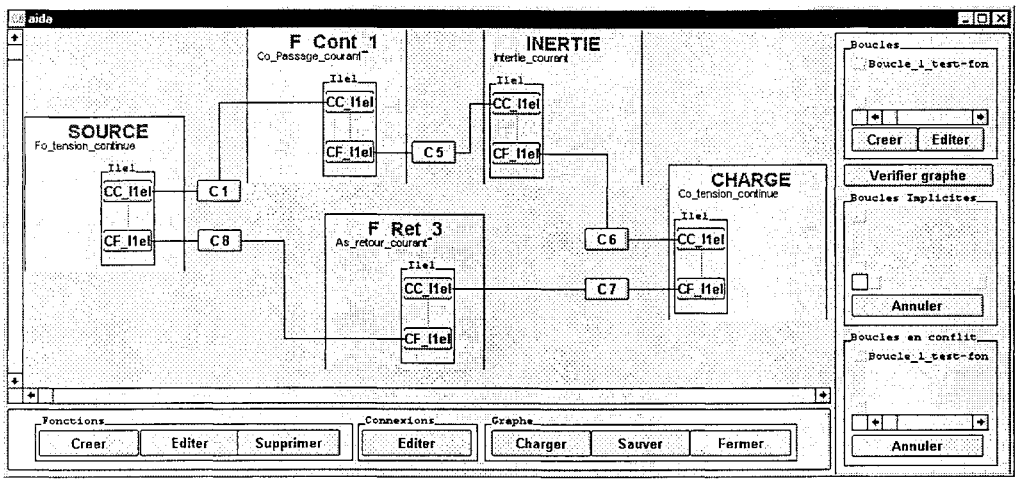



**Figure 5.23 : La boucle modifiée**

La nouvelle vérification de cohérence signale que la contrainte de continuité de la fonction inertie n'est pas respectée. Il est donc nécessaire de créer une seconde boucle (avec F Cont 3 et F ret 4) permettant d'assurer la continuité du courant (flux) dans l'inertie comme l'illustre la figure 5.24.

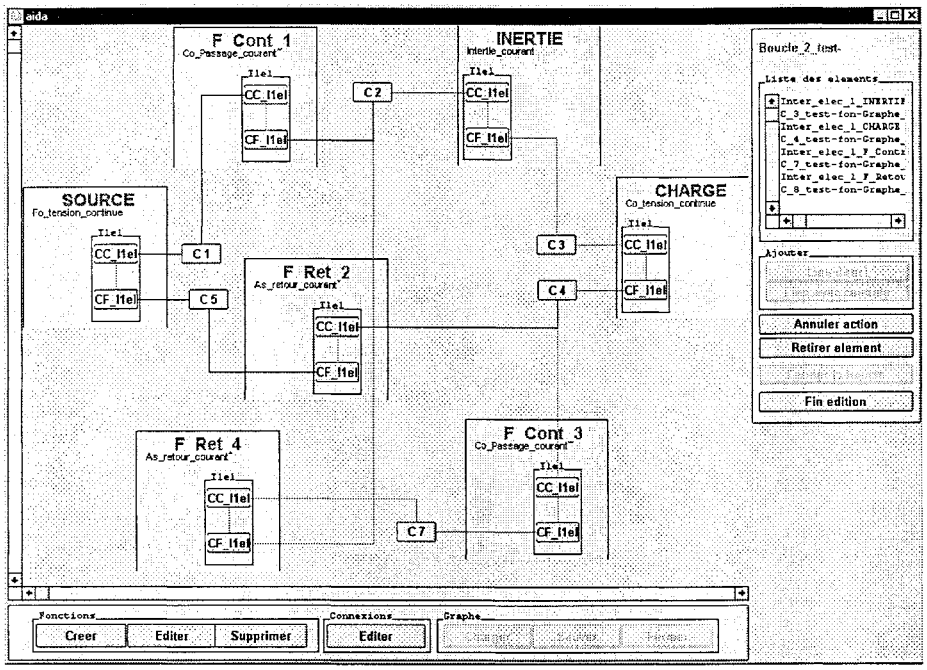



**Figure 5.24: Le graphe fonctionnel modifié**

La troisième phase de vérification de la cohérence réalise les opérations suivantes.

1. **Elle détermine automatiquement les boucles implicites.** Ici, le prototype construit une seule boucle.

**2. Elle propose de résoudre le problème de causalité sur la boucle implicite** en plaçant un lien de simultanéité (LNS) entre les fonctions F_CONT_3 (ou F_Ret_4) et F_CONT_1 (ou F_RET_2).

**3. Elle place un lien séquentiel** (LSEQ) entre la fonction F_CONT_3 (ou F_Ret_4) et F_CONT_1 (ou F_RET_2) pour garantir que la fonction inertie est toujours dans une boucle autorisant un flux.

La figure 5.25 montre le résultat de l'analyse de cohérence.

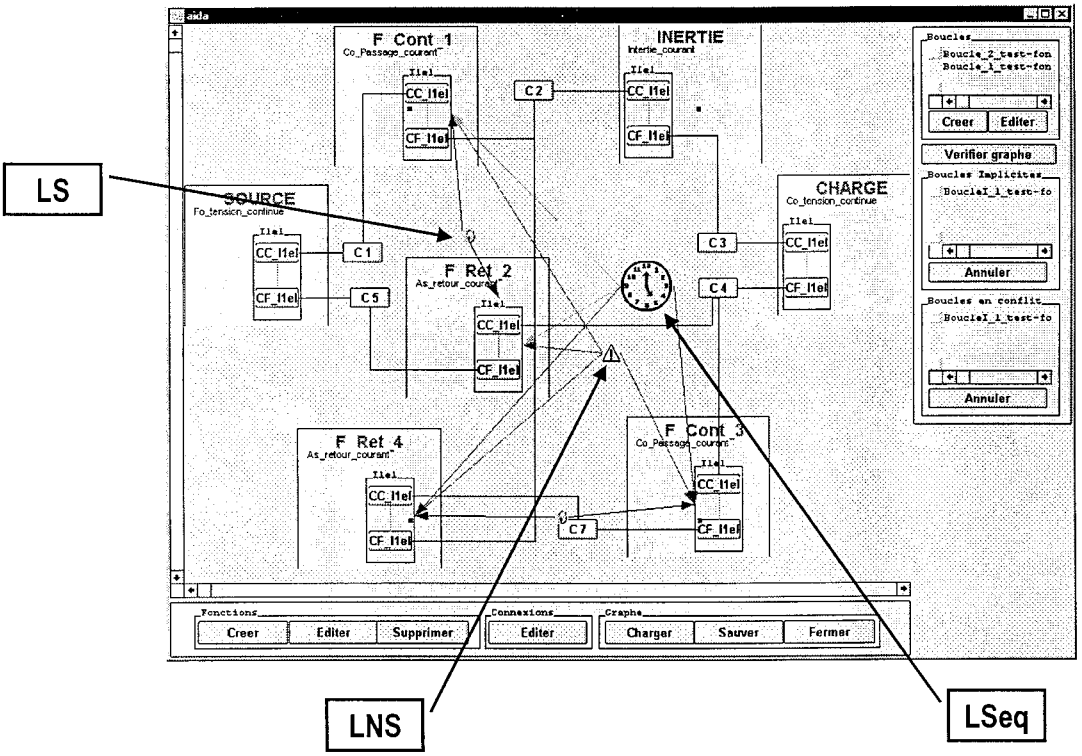



**Figure 5.25 : Le résultat de la vérification de cohérence**

Nous devons maintenant implanter dans notre outil toute l'expertise permettant d'effectuer les autres vérifications :

- les sens de flux,
- la comptabilité entre les niveaux,
- la vérification des connexions.

## III. Conclusion

Nous avons montré dans ce chapitre une utilisation possible de l'analyse fonctionnelle en vue de concevoir des structures de convertisseur statique. Nous avons présenté ici une structure très simple, traitée avec notre outil en cours de réalisation, et une autre beaucoup plus complexe pour monter l'intérêt de la démarche fonctionnelle. Il nous reste maintenant à continuer le développement de notre outil pour qu'il effectue complètement la vérification de cohérence des graphes afin de pouvoir traiter des exemples beaucoup plus complexes. L'étape suivante sera de bien formaliser le passage du fonctionnel vers le structurel pour construire les différentes solutions, et tendre vers un système d'aide à la conception de structures de convertisseurs statiques.

# Conclusion générale et perspectives

Dans ce rapport, nous avons proposé une démarche d'analyse fonctionnelle pour la spécification de la partie puissance des convertisseurs statiques.

Ainsi, celle-ci propose une modélisation orientée objet des fonctions et des graphes fonctionnels. L'analyse proposée pour ces derniers permet de vérifier les interactions énergétiques entre les différentes fonctions. Dans ce cadre, des solutions sont proposées pour éviter un certain nombre d'incohérences sur les échanges d'énergie. Les potentialités de l'approche sont illustrées autour d'un exemple.

A partir de cette modélisation, nous proposons un prototype informatique qui reprend en partie le modèle proposé pour l'analyse fonctionnelle. En effet, ce travail d'implantation n'est pas complet (faute de temps), mais permet d'évaluer l'approche proposée sur le papier. Un exemple est présenté afin d'illustrer l'utilisation de ce prototype.

Dans notre approche, le concepteur est un acteur très actif, notamment pour faire des choix afin d'orienter la conception. A l'avenir, si l'approche est introduite dans un système de conception utilisant de l'expertise, le concepteur pourra bénéficier de l'expérience des autres, tout en continuant à faire ses propres choix s'il le désire. Cependant, il aura aussi le rôle de compléter les bases de données : fonctions, structures et règles de conception.

Les travaux présentés dans cette thèse ouvrent une nouvelle voie pour la définition de la structure des convertisseurs statiques, et permettent d'envisager de tendre à long terme vers la conception innovante.

Cependant, différentes perspectives découlent de ce travail.

Nous devons maintenant continuer l'implantation informatique de notre modèle fonctionnel afin de pouvoir réaliser des structures complexes. Nous pouvons citer par exemple les structures multi-niveaux, les ensembles avec plusieurs fonctions Gyrateur, etc.

Dans ce travail, il faudra aussi envisager la migration du prototype vers des environnement de développement dont la pérennité n'est pas en cause. Ceci pourra se faire dans le cadre des produits de nouvelle génération d'ILOG et/ou des nouveaux langages objets (par exemple JAVA).

Actuellement la démarche proposée effectue les vérifications sur chaque boucle de façon indépendante sur des critères de causalité et de niveau de flux. Pour pouvoir prendre en compte les problèmes d'interaction entre les boucles ainsi que pour analyser les problèmes sur les niveaux efforts, il est nécessaire de mettre en place une arithmétique sur les objets Niveaux. Pour cela, l'utilisation de l'algèbre floue peut être une voie intéressante.

Afin de nous permettre de converger vers un environnement complet d'aide à la conception des structures, il est nécessaire de formaliser le lien fonction-structure. Nous procéderons en plusieurs étapes :

- Etape 1 : Il sera nécessaire d'implanter la base de données objet pour représenter les structures.

- Etape 2 : Il faudra définir et mettre en œuvre des liens entre les fonction et les structures. Ce travail est très délicat du fait que dans le cas général, ce lien peut être entre N structures et M fonctions, avec N et M pouvant être différents de un. Or, les imbrications de fonctions pour une structures donnée et réciproquement de structures pour une fonction donnée sont délicates à gérer. Ainsi, afin de limiter au mieux ces imbrications, et donc faciliter fortement la formulation des liens entre fonctions et structures, il sera important de travailler sur des graphes fonctionnels et structurels non réduits. Pour cela, on cherchera à mémoriser les réductions de ces graphes pour pouvoir revenir si besoin à des graphes plus clairs. Le mécanisme de passage d'un graphe structurel à un graphe fonctionnel, et réciproquement, devra être élaboré ainsi que celui qui permet de passer d'un graphe réduit à sa version initiale non réduite.

- Etape 3 : L'outil de conception devra manipuler les règles pour la préparation de la phase dimensionnement du convertisseur à partir d'informations issues de la représentation fonctionnelle. Il sera donc nécessaire de formuler et d'introduire des règles :

    - pour agencer les fonctions,

    - pour faire des choix technologiques,

- pour rejeter des solutions fonctionnelles ou structurelles trouvées par l'outil qui ne sont pas viables d'après l'expériences de concepteurs.

Ces règles devant être définies par les concepteurs de convertisseurs, nous devrons fournir un formalisme pour leur représentation qui soit suffisamment générique et permette l'ajout d'une règle indépendamment de celles existant dans le système [ESC96].

- Etape 4 : Nous devrons intégrer l'outil de conception de structures dans un processus complet de conception des convertisseurs statiques. Ainsi, nous devrons l'interfacer avec des outils d'analyse (simulation) et de dimensionnement.

A long terme, nous devrons tendre vers l'analyse fonctionnelle d'entraînement électromagnétique complet, c'est-à-dire prendre en compte les aspects commande et machine. Pour la partie puissance, nous pourrons utiliser la notion de fonction Gyrateur et l'étendre. Pour la partie commande, nous pourrons envisager un couplage de notre approche avec une approche fonctionnelle adaptée au traitement de la commande, par exemple SADT.

# Bibliographie


AID94 Ilog , **"AIDA, Manuel de référence, version 1.76 ",** 2 av. Galliéni, 94253 GENTILLY, 1994

ALL93 B. Allard, H. Morel et J.P. Chante, **"Power Electronic circuit simulation using bond graph and Pétri network techniques ",** PESC'93, 24th annual Power electronics Specialists Conference, University of Washington, Seattle, USA, June 20-24

BEL94 P. Belloy, **"Intégration de connaissances métier dans la conception : Un modèle pour les pièces mécaniques. Application à l'usinage et à l'estampage",** Thèse Université Joseph Fourier- Grenoble I, 1994, 192p

BIE95 J.D. Biedermann, D.E. Grierson, **"A generic model for building design",** Engineering with Computers Vol: 11 Iss: 3 , 1995, p. 173-184

BIG95 J. Bigeon, E. Escande, M. Tollenaere , **"Modèle de conception : place de l'utilisateur",** Actes de 01Design'95, Aspects Communication en Comception, Autrans - Isère, 11-13 Janvier 1995, p41-51

BRU94 P. BRUNOX, **"Etude de différentes représentations du contacteur électromagnétique dans un but de conception innovante",** Rapport de DEA de genie électrique, Septembre 1994, 30p

BUR95 J.M. Burdio, A. Martinez, **"A time-domaine simulator for power electronic converters based on discrete-time modeling of switches",** Proceedings of the EPE'95. Sixth European Conference on Power Electronics and Applications, Sevilla, Spain, september 19-21, 1995, p 1355,1360

BUR96 J. M. Burdio,A. Martinez, J. R. Garcia, **"A new ZVS resonant sine-wave inverter obtained by a synthesis method",** PESC '96 Record. 27th Annual IEEE Power Electronics Specialists Conference (Cat. No.96CH35962), Banevo, Italy, ISBN: 0 7803 3500 7, june 23-27, 1996, p245-251

CAM96 J.P. Cambronne, Ph. Le Moigne, J.P. Hautier, **"Synthèse de la commande d'un onduleur de tension",** Journal de physique III France n°6, juin 1996, p 757-778

CER96 M. favier, C. Raoul, **"Analyse fonctionnelle du besion & gestion de projet en environnement goupware",** GI5, 5th International Congress of Industrial Enfineering, 2-4 avril 1996

CHA93 B. Chandransekaran, A. K. Goel, Y; Iwasaki, **"Functional representation as design rationale",** IEEE Computer, ISSN 0018-9162, january 1993, p48-56

CHE89 Y. Chéron, **"La commutation douce dans la conversion statique de l'énergie électrique",** Technique et Documentation-Lavoisier, Avril 1989, 296p

CON92 D. Constant, **"Experimentation du modele de gestion de donnees SHOOD sur un exemple de conception mécanique",** Mémoire de DEA de Mécanique, Juin 1992, 40p

CON96 D. Constant, **"Contribution à la spécification d'un modèle fonctionnel de poduits pour la conception intégrée de systèmes mécaniques",** Thèse de doctorat spécilité Mécanique de l'Université Joseph Fourier, 1996, 158p

DAV97 P. Davancens, T. Meynard, **"Etude des convertisseurs multicellulaire parallèles : Modélsation",** Journal de physique III France n°7, 143-160

DAV97A P. Davancens, T. Meynard, **"Etude des convertisseurs multicellulaire parallèles : Analyse du modèle",** Journal de physique III France n°7, 161-177

DEB93 K. Debebe, V. Rajagopalan, T.S. Sankar, **"Expert system based instruction aid for power electronics",** IPEC '93. International Power Engineering Conference 1993. Conference Proceedings , 18-19 March 1993, p. 549-54 vol.2

DEL91 M. Delgado de nieto, **"Description et simulation de systemes lineaires representés par le Bond Graph",** These de doctorat spécailité Automatique de l'institut national des sciences appliquées de rennes, 1991

DIM92 Ilia Ivanov Dimitrov, **"Systems Based Knoledge Representation: Relations and Methods",** Artificial intelligence V: Methodology, Systems Applications. , Proceedings of the Fifth InternationalConerence (AIMSA '92), p. 203-12

DUC92 J.P. Ducreux, A. Castelain, C. Rombaut, G. Dauphin-Tanguy, **"Application de l'outil Bond-Graph à la modélisation des convertisseurs statiques de l'électronique de puissance",** Extrait du rapport des Journées EEA – électrotechnique, 26-27 Mars 1992, Conservatoire National des Arts et Métiers Electrotechnique, Ecole centrale de Lille

GER93 L. Gerbaud **"Aide à la conception des ensembles machine – convertisseur – commande : Apport d'une demarche système expert",** thèse de doctorat spécialité genie electrique de l'Institut National Polytechnique de Grenoble, 1993, 164p

ENS94 Marco Ensing, Ray Paton, Piet-Hein Speel, Roy Rada, **"An object-oriented approach to knowledge representation in a biomedical domain",** Artificial Intelligence in Medicine, 1994,Vol: 6 Iss: 6 p. 459-82

ESC96 E. Escande, **"Modélisation "objet" du processus de conception dans le domaine du génie electrique : application au cas de la machine asynchrone, le système OPUS"** thèse de doctorat spécialité genie electrique de l'Institut National Polytechnique de Grenoble, 1996, 237p

ESS90 J. Escamilla, V. Favier, P. Jean, G.T. Nguyen, D. Rieu, **"Representation de connaissances dynamiques dans SHEPA",** Rapports de Recherche, Avril 1990, 32p

FER94 J.P. Ferrieux, F. Forest, **"Alimentation à découpage, Convertisseur à résonnance",** ISBN 2-225-84555-7 Edition Masson, juin94, 296 p

FEZ93 D. Fezzani, H. Piquet,Y. Chéron, M. Metz, **"Design of static converters : an expert system approach",**EPE '93. The European Power Electronics Association, Brighton, UK, september 13-16, 1993, p47-52

FEZ96 D Fezzani , **"Système expert pour la conception en electronique de puissance",** Thèse spécialité genie electrique de l'Institut National Polytechnique de Toulouse, janvier 1996, 207p

FEZ96A D. Fezzani, H. Piquet, H. Foch, **"Systeme expert pour la conception en electronique de puissance",** Journal de physique III France n°6, Mars 1996, p349-361

FOC H. Foch, **"Principes fondamentaux : Commutation dans les convertisseurs statique",** Techniques de l'ingénieur, 21 rue Cassette, Paris, D3153

FRA94 Francois F.; Escande E.; Bigeon J, **"A methodology for knowledge representation in expert system, in the field of electrical engineering",** Compumag 93, IEEE Transactions on Magnetics, Vol: 30Iss: 5, pt.2, 1994 ,p. 3656-9

FRA96A B. François, J.P Cambronne, J. P. Hautier, **"Caractérisation des convertisseurs matriciels : I Structure de l'automate de commande rapprochée",** Journal de physique III France n°6, mai 1996, p625-639

FRA96B B. François, J.P Cambronne, J. P. Hautier, **"Caractérisation des convertisseurs matriciels : II Synthèse des fonctions de connexion",** Journal de physique III France n°6, mai 1996, p641-660

GEN91 A. Gentilhomme, **"C.O.C.A.S.E Un système expert d'aide à la conception d'appariellages electriques",** Thèse spécialité genie electrique de l'Institut National Polytechnique de Grenoble, 3 Mai 1991, 150p

GSI93 J. Bigeon, F. Bounass, D. Constant, C. Djeraba, J. Escamilla, E. Escande, F. Francois, J. F. Millasseau, G. T. Nguyen, D. Rieu, M. Tollenaere, **"Modelisation objet pour la C.F.A.O",** Rapport Interne du projet, janvier 93, 73p

GUI93 O. Guillaud, **"Modélisation shood du contacteur electromagnétique",** Rapport de DEA de Génie Industriel, juin93,

HSI92 Hsieh, J.; Liu, C.-C, **"An integrated knowledge- and algorithm-based method for power converter design",** PESC '92 Record. 23rd Annual IEEE Power Electronics Specialists Conference (Cat. No.92CH3163-3), ISBN: 0 7803 0695 3, 10-11 Sept. 1992, p. 1316-23 vol.2

JEY96 J.J. Jeyappragash, T.V. Sivakumar, V.V. Sastry, **"Object oriented modeling, simulation and optimisation of power electronic circuits",** PESC '96 Record. 27th Annual IEEE Power Electronics Specialists Conference (Cat. No.96CH35962), Banevo, Italy, ISBN: 0 7803 3500 7, june 23-27, 1996, p581-585

KAI94 Hermann Kaindl, **"Object-oriented approaches in software engineering and artificial intelligence",** Journal of Object-Oriented Programming, Jan 1994, Vol: 6Iss: 8p. 38-45

KUR93 Dhammika Kurumbalapitiya,S. Ratnajeevan H. Hoole, **"An object-oriented representation of electromagnetic knowledge",** Journal IEEE Transactions on Magnetics, Mars 1993, p1939-1942 Vol. 29 No 2

LAU98 P. Lautier, **"Modélisation des convertisseurs à découpage pour la conception et la commande : Application à l'ondeur",** Thèse de doctorat de l'Institut National des Sciences Appliquées de Lyon, Janvier 1998

LEL94 Ilog , **"LE-LISP, Manuel de référence, version 15.26 ",** 2 av. Galliéni, 94253 GENTILLY, 1994

LIO93 M. P. Liotard, **"Mécanisme de classification pour un système de réprésentation de connaissances",** Memoire Ingénieur C.N.A.M. en Informatique, 25 Mars 1993, 110p

LIS91 M. Lissandre, **"Maîtriser SADT",** Armand Colin, Février 1991- ISBN 2-200-42022-6, 219p

MAO96 H. Mao F. C. Lee, X. Zhou, D. Boroyevich, **"Improuved zero-current transition converter for high power applications",** IAS'96 : Conference record of 1996 IEEE Industry Applications Society (96CH35977), San Diego, California, USA, ISBN 0-7803-3544-9, October 6-10, 1996, p 1145-1151

MAR90 Marino, O.; Rechenmann, F.; Uvietta, P., **"Multiple perspectives and classification mechanism in object-oriented representation",** ECAI 90. Proceedings of the 9th European Conference on Artificial Intelligence, 6-10 Aug. 1990, p. 425-30

MAS92 B. Maschke, **"Approche systématique des systèmes continus, Graphes de liaison",** Extrait du rapport des Journées du Club EEA Electotechnique.

MAS94 Ilog , **"MASAI, Manuel de référence, version 1.76 ",** 2 av. Galliéni, 94253 GENTILLY, 1994

MIL92 Jean François MILLASEAU , **"Les méthodes sont aussi des objets",** DEA Informatique, septembre 92, 91p

MOH89 Mohan N.,Underland T. M., Robbins W.P. , **"POWER ELECTRONICS : Converters, Applications and Design: ",** 1ere Ed : ISBN: 0 471 61342 82eme :ISBN 0 471 50537 4, 1989

NGU91 Gia Toan NGUYEN, Dominique RIEU , **"Database issues in object-oriented design",** Rapport de Recherche, Février 1991, 16p

NGU92A Nguyen G.T, D. Rieu , **"Multiple objet representations",** 1992 ACM Computer Science Conference. Communications Proceedings, 1992, p197-204

NGU92B Nguyen G.T D.Rieu J.Escamilla , **"An Objet Model For Engineeering Design ",** ECOOP '92. European Conference on Object-Oriented, 1992, p233-251

OUS97 C. Oussalah, **"Ingénierie Objet",** Masson éditeur, mai 1997- ISBN 2-7296-0642-4, 424p

PIQ94 H. Piquet, Jp. Cambronne, S. Saadate, G. Glaize , **"Méthodologie et outils pour la conception en électronique de puissance",** EPF'94, 1994, p25-31

PON90 Anne Pons , **"Des points de vue comme solution simple et objective aux problemes d'heritage multiple dans les languages orientes objets",** Toulouse '90. Third International Workshop. Software Engineering and its Applications Proceedings, 1990 , p. 841-52 vol.2

PRI94 C. Prins , **"Algorithmes de graphes",**EYROLLES, Paris, ISBN 2-212-09020-xl, 1994, 4

RAB95 D. Rabasté, J. M. Maltartre, C. Glaize , **"Chargeur de batterie avec isolation galvanique et absorption sinusoïdale",** Journal de physique III France n°6, novembre 1995, p1887-1904

RAI98 C. Raimond, **"Apprenez JAVA 1.1 en 21 jours",** Simon & Schuster Macmillan (France), 1998,

REG95 J. Bigeon, F. Bounass, D. Constant, C. Djeraba, J. Escamilla, E. Escande, F. Francois, J. F. Millasseau, G. T. Nguyen, D. Rieu, M. Tollenaere, **"Projet SHOOD, Modélisation objet pour la CFAO : modèle de conception ",** Rapport de fin contrat, réponse à l'appel d'offres 1993 region Rhône-Alpes Pôle productique, Aout 1995, 107p

ROU95 L. Roucoules, **"Conception integrée de moteurs electriques. Influence de la fabrication sur les performances de la machine",** Mémoire de DEA de Mécanique, Juin 1995, 50p

REE94 L. H. Reeker, **"Tools for representing and managing knowledge : some pratical requiments and suggestions",** Proceedings. Sixth International Conference on Tools with Artificial Intelligence (Cat. No.95CH35727, ISBN: 0 8186 6785 0, 6-9 Nov. 1994, p240-244

SAA93 M. Saad, T Rafesthain, V. Rajagopalan, K. Al-Haddad , **"Conception assistée par ordinateur des convertisseurs statiques directs",** Canadian Journal of Electronics and Computer Engineering, Vol 18 No 3 1993, p97-103

SME94 Ilog , **"SMECI, Manuel de référence, version 1.76 ",** 2 av. Galliéni, 94253 GENTILLY, 1994

TAN90 Tang, W. ; Tabisz, W.A. ; Lotfi, A. ; Lee, F.C. ; Vorperian, V, **"DC analysis and design of forward multi-resonant converter",** PESC '90. Record. 21st Annual IEEE Power Electronics Specialists Conference (Cat. No.90CH2873-8),New York, NY, USA, ISSN: 0275-9306 , june 11-14, 1990, p. 862-869

TSA94 Curtis H.K.Tsang, Chris Bloor., **"An object-oriented automobile diagnosis expert system",** Applications and Innovations in Experts Systems II. Proceedings of Expert Systems 94, the Fourteenth AnnualTechnical Conference of the British Computer Society, 1994, p. 111-124

UEN92 Ueno, H.; Yamamoto, Y.; Fukuda, H., **"Knowledge modelling based on object model-an approach to multi-use engineering knowledge base",** Information Modelling and Knowledge Bases III: Foundations, Theory and Applications, Fujiyoshida, Japan , 1992, p. 75-96

UME97 Y. Umeda, T. Tomiyama, **" Functional Reasoning in Design "**, IEEE Expert, march-april 97, pp. 42-48

# Annexes

# Annexe I : Les graphes de liens

Le tableau A.1 présente les composants élémentaires utilisés dans les Bond Graph avec leurs équations de constitution suivant la causalité [LAU98], [DEL91].

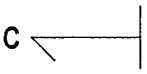
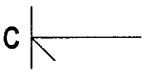
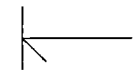
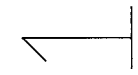
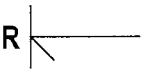
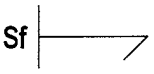
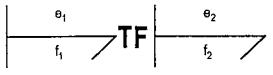
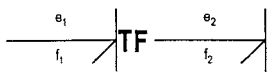
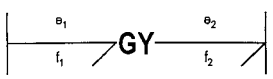
| Nom, Symbole | Equations | Variable d'entrée | Variable de sortie | Représentation graphique |
|---|---|---|---|---|
| Capacité, C | $q = \int_0^t f(t).dt$<br>$e = \Theta_c^{-1}(q)$ | $f$ | $e$ |  <br>Causalité intégrale (préférentielle) |
| | $q = \Theta_c(e)$<br>$f = \frac{dq}{dt}$ | $e$ | $f$ |  <br>Causalité différentielle |
| Inertie, I | $p = \int_0^t e(t).dt$<br>$f = \Theta_I^{-1}(p)$ | $e$ | $f$ |  <br>Causalité intégrale (préférentielle) |
| | $p = \Theta_I(f)$<br>$f = \frac{dp}{dt}$ | $f$ | $e$ |  <br>Causalité différentielle |
| Elément dissipatif | $e = \Theta_R(f)$ | $e$ | $f$ | |
| | $e = \Theta_R^{-1}(f)$ | $f$ | $e$ |  |
| Source d'effort Se | $e = e(t)$ | $e$ | $f$ | |
| Source de flux SF | $f = f(t)$ | $f$ | $e$ |  |
| Transformateur TF | $e_1 = m \cdot e_2$<br>$f_2 = m \cdot f_1$ | $e_2$, $f_1$ | $e_1$, $f_2$ |  |
| | $e_2 = m \cdot e_1$<br>$f_1 = m \cdot f_2$ | $e_1$, $f_2$ | $e_2$, $f_1$ |  |
| Transformateur TF | $e_1 = m \cdot f_2$<br>$e_2 = m \cdot f_1$ | $f_1$, $f_2$ | $e_1$, $e_2$ |  |
| Gyrateur GY | $f_1 = m \cdot e_2$<br>$f_2 = m \cdot e_1$ | $f_1$, $f_2$ | $e_1$, $e_2$ | |

**Tableau A. 1 : Composants élémentaires des graphes de lien [LAU98], [DEL91]**

# Annexe II : La décomposition structurelle

Cette annexe présente une partie (5% environ) de notre décomposition structurelle des convertisseurs. En particulier, nous présentons une classification possible des cellules de commutation. (voir figure A.1)

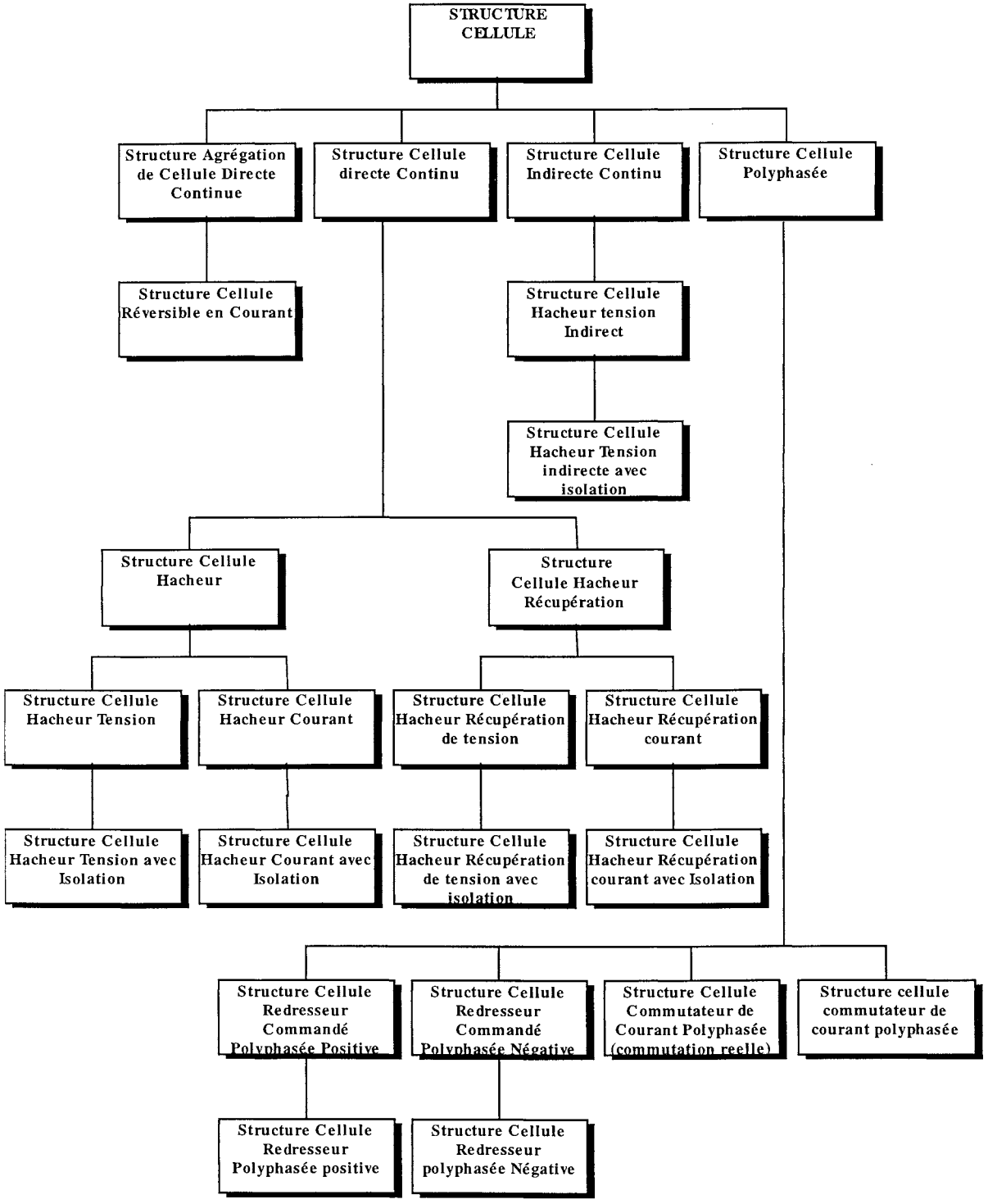



**Figure A. 1 : La classification de cellule de commutation.**

A titre d'exemple, nous allons développer des exemples de représentation de certaines structures. Dans un premier temps, nous présentons les éléments de base nécessaires pour cette modélisation, puis nous l'illustrerons par la structuration de cellules Hacheur.

## II.1. Un extrait des classes de service pour la caractérisation des classes de structures

### II.1.1. La classe Segment

**ROLE**

Cet objet permet de modéliser les segments de fonctionnement, c'est-à-dire les tolérances en terme de courant et tension sur les structures interrupteurs. La figure A.2 présente l'arbre de spécialisation de la classe segment.

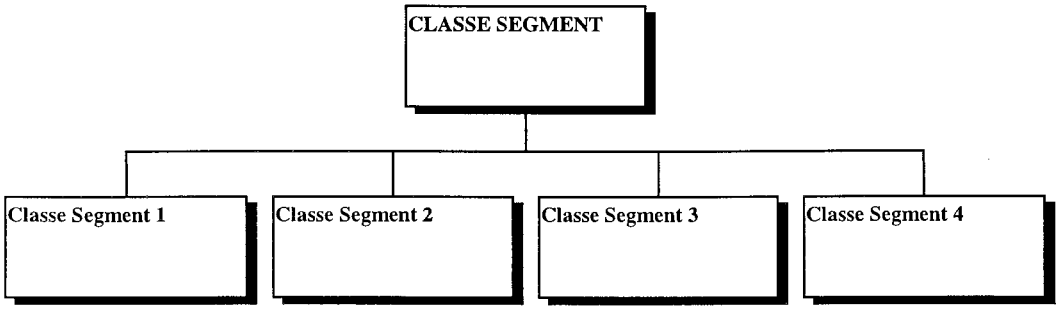



**Figure A. 2 : L'arbre des classes segments**

**CHAMPS DE LA CLASSE**

| | |
|---|---|
| **Tension** | ***Positive, Négative, Nulle*** |
| **Courant** | ***Positif, Négatif, Nul*** |

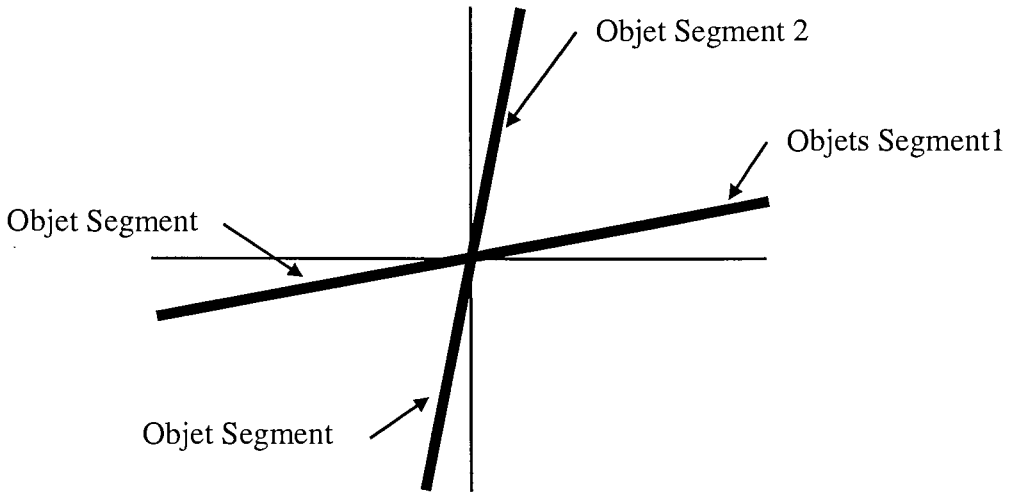



**Figure A. 3 : La représentation graphique des segments**

## II.1.2. La Classe Quadrant

La notion de quadrant permet de caractériser la nature énergétique des structures. Cette notion sera utile au moment de la connexion des structures entre elles, afin de vérifier la compatibilité électrique. Il est aussi nécessaire de définir une convention pour le calcul des puissances. Cette classe est organisée comme le montre la figure A. 4.

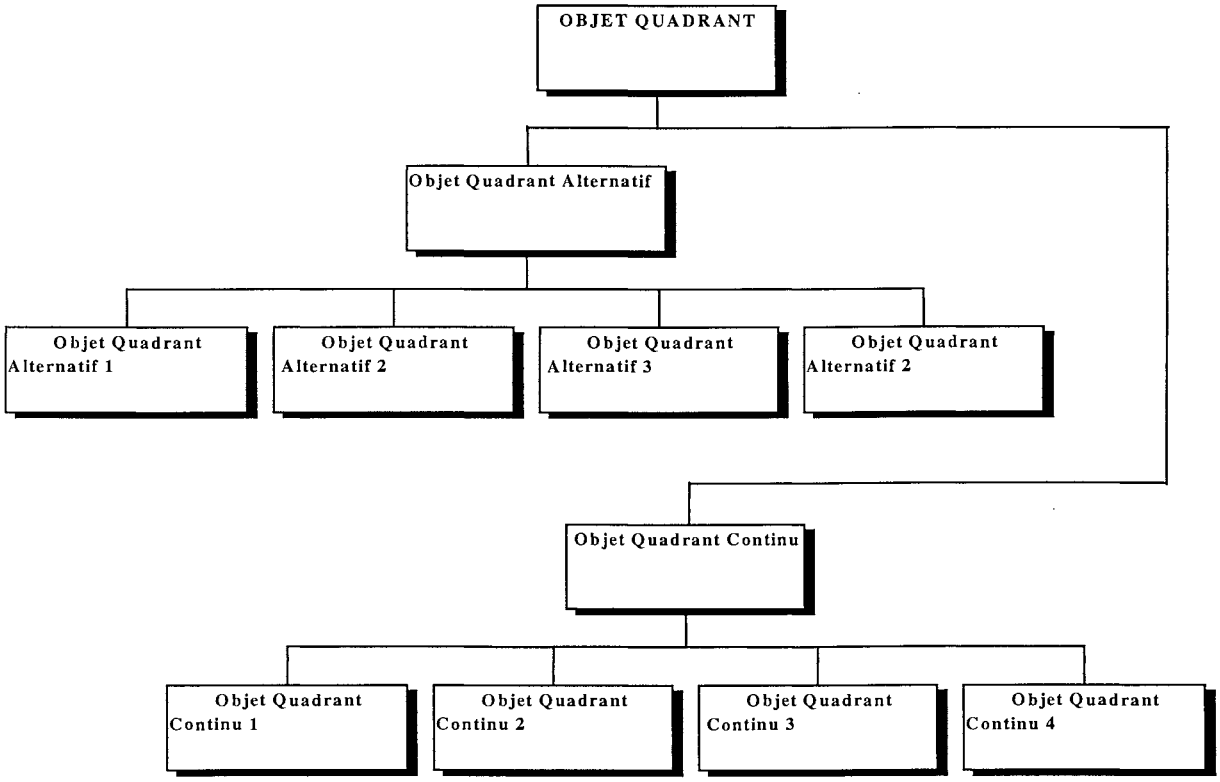



**Figure A. 4 : La classification des Quadrants**

**DEFINITION DE LA CLASSE**

| **Convention** | ***Générateur, Récepteur*** |
|---|---|
| **Puissance active** | ***Positive, Négative*** |

### *II.1.2.1. La Classe Quadrant Alternatif*

| Convention | *Générateur, Récepteur* |
|---|---|
| Puissance active | *Positive, Négative* |
| **Puissance réactive** | ***Positive, Négative*** |

Cette classe permet de caractériser le transfert d'énergie dans les structures. Les différents quadrants alternatifs se définissent par rapport aux puissances active et réactive comme le montre la Figure A. 5.

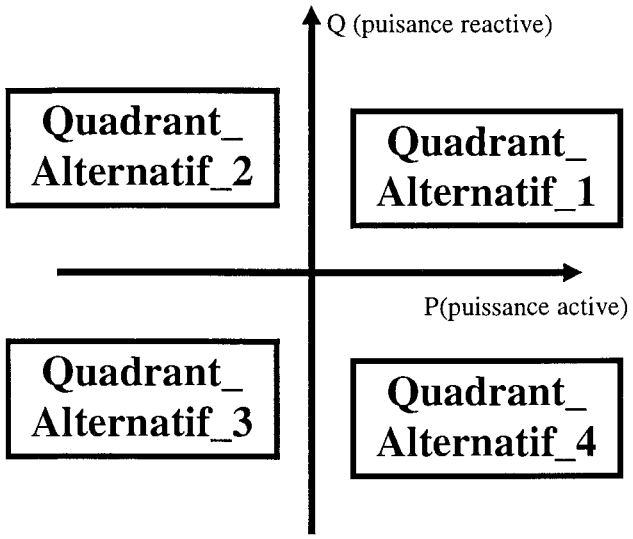



**Figure A. 5 : La définition des quadrants en alternatif**

### *II.1.2.2. La classe Quadrant Continu*

| Convention | *Générateur, Récepteur* |
|---|---|
| Puissance | *Positive, Négative* |
| **Tension** | ***Positive, Négative*** |
| **Courant** | ***Positif, Négatif*** |

**Note : Nous entendons par un signal continu, un signal à valeur moyenne non nulle.**

En continu, les quadrants se définissent par rapport aux grandeurs courant et tension comme le montre la Figure A. 6.

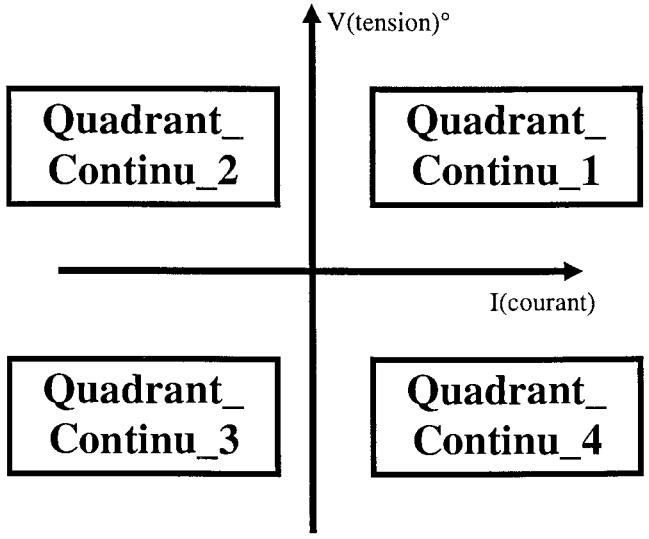



**Figure A. 6 : La définition des quadrants en continu**

## II.2. Structure Cellule Hacheur

### ROLE

Elle assure la connexion entre une source continue et une charge en courant (ou passive) avec le contrôle du transfert d'énergie.

### INFORMATIONS PROPRES

| | |
|---|---|
| Structure mère | *Structure cellule directe continu-continu* |
| Fréquence de commutation | ⇒ *Objet Valeur Fréquence* |
| Nature de la source | *Courant, Tension* |
| Quadrant d'entrée | *Quadrant Continu 1 (Récepteur)* |
| Nature de la charge | *Courant, Tension* |
| Quadrant de sortie | |
| **Interrupteur transfert source-charge** | ***Structure interrupteur*** |
| **Interrupteur continuité** | ***Structure interrupteur*** |
| Structure Commande | *Structure Commande* |
| Interface électrique source | ⇒ *Objet Interface Electrique Bipolaire* |
| Interface électrique charge | ⇒ *Objet Interface Electrique Bipolaire* |

### STRUCTURES COMPOSANTS

Structure Interrupteur
Structure diode

### II.2.1. Structure Cellule Hacheur Tension

### ROLE

Elle assure la connexion entre une source continue et une charge en courant (ou passive) avec le contrôle du transfert d'énergie.

### INFORMATIONS PROPRES

| | |
|---|---|
| Structure mère | *Structure Cellule Hacheur* |
| Fréquence de commutation | ⇒ *Objet Valeur Fréquence* |
| Nature de la source | *Tension* |
| Quadrant d'entrée | *Quadrant Continu 1 (Récepteur)* |
| Nature de la charge | *Courant* |
| Quadrant de sortie | *Quadrant Continu 1 (Générateur)* |
| Nœud courant | ⇒ *Objet nœud* |
| Nœud tension | ⇒ *Objet nœud* |
| Interrupteur transfert source-charge | *Structure interrupteur* |
| Interrupteur continuité | *Structure Diode* |
| Structure Commande | ⇒ *Structure Commande Mono Interrupteur* |
| Interface électrique source | ⇒ *Objet Interface Electrique Bipolaire Continu* |
| Interface électrique charge | ⇒ *Objet Interface Electrique Bipolaire Continu* |

## STUCTURES COMPOSANTS

Structure Interrupteur avec au moins les champs suivants

| | |
|---|---|
| **Structure mère** | |
| **Segments de fonctionnement** | ⇒ *Objet segment 1*<br>⇒ *Objet segment 2* |
| **Fréquence maximale** | ⇒ *Objet valeur fréquence fixe* |
| **Commutations nécessaires** | ⇒ *Objet commutation 1*<br>⇒ *Objet commutation 2* |
| **Commande** | ⇒ *Objet commande* |
| **Valeurs caractéristiques maximales** | |
| **Valeurs efficaces caractéristiques maximales** | |
| **Variation des valeurs caractéristiques maximale** | |
| **Interface thermique** | ⇒ *Objet interface thermique* |
| **Interface électrique** | ⇒ *Objet interface électrique* |

## CHAMPS DE L'INTERFACE ELECTRIQUE SOURCE

| | |
|---|---|
| **Convention** | ***Récepteur*** |
| **Type** | ***Tension*** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | ***Sans, Capacitive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ⇒ ***Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

## CHAMPS DE L'INTERFACE ELECTRIQUE CHARGE

| | |
|---|---|
| **Convention** | ***Générateur*** |
| **Type** | ***Courant*** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | ***Sans, Inductive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ⇒ ***Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

## SCHEMA EQUIVALENT

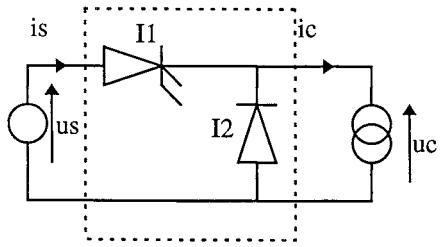



**Figure A. 7 : Schéma de la structure cellule Hacheur**

**REGLES**

- L'interrupteur I1 est commandé à l'ouverture et à la fermeture, et il contrôle le transfert d'énergie.
- L'interrupteur I1 peut avoir une seule commutation commandée, mais l'autre commutation doit se produire de façon naturelle ( => structure de blocage ou d'amorçage par résonnance).
- L'interrupteur I2 est à commutation naturelle.
- Le transfert d'énergie est caractérisé par la durée de conduction de I1.

=> commande à fréquence fixe, rapport cyclique variable

=> commande à fréquence variable, rapport cyclique fixe.

=> commande à fréquence et rapport cyclique variables. => commande MLI

régulation du courant ic.

régulation du courant is (absorption sinus)

### *II.2.1.1. Structure Cellule Hacheur Tension avec Isolation*

**ROLE**

Elle assure la connexion entre une source continue et une charge en courant (ou passive) avec le contrôle du transfert d'énergie.

**INFORMATIONS PROPRES**

| | |
|---|---|
| Structure mère | *Structure Cellule Hacheur* |
| Fréquence de commutation | ⇒ *Objet Valeur Fréquence* |
| Nature de la source | *Tension* |
| Quadrant d'entrée | *Quadrant Continu 1 (Récepteur)* |
| Nature de la charge | *Courant* |
| Nœud courant | ⇒ *Objet nœud* |
| Nœud tension | ⇒ *Objet nœud* |
| Interrupteur transfert source-charge | *Structure interrupteur Fonctionnel* |
| Interrupteur continuité | *Structure Diode* |
| Structure Commande | ⇒ *Structure Commande Mono Interrupteur* |
| **Structure isolation** | ***Structure Isolation magnétique régime impulsionnel asymétrique sans accumulation*** |
| Interface électrique source | ⇒ *Objet Interface Electrique Bipolaire Continue* |
| Interface électrique charge | ⇒ *Objet Interface Electrique Bipolaire Continue* |

**STUCTURES COMPOSANTS**

Structure Interrupteur avec au moins les champs suivants

| | |
|---|---|
| **Structure mère** | |
| **Segments de fonctionnement** | ⇒ ***Objet segment 1***<br>⇒ ***Objet segment 2*** |
| **Fréquence maximale** | ⇒ ***Objet valeur fréquence fixe*** |
| **Commutations nécessaires** | ⇒ ***Objet commutation 1***<br>⇒ ***Objet commutation 2*** |
| **Commande** | ⇒ ***Objet commande*** |

| | |
|---|---|
| **Valeurs caractéristiques maximales** | |
| **Valeurs efficaces caractéristiques maximales** | |
| **Variation des valeurs caractéristiques maximale** | |
| **Interface thermique** | ⇒ ***Objet interface thermique*** |
| **Interface électrique source** | ⇒ ***Objet interface électrique*** |
| **Interface électrique charge** | ⇒ ***Objet interface électrique*** |

**CHAMPS DE L'INTERFACE ELECTRIQUE SOURCE**

| | |
|---|---|
| **Convention** | ***Générateur*** |
| **Type** | ***Tension*** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | **Sans, Capacitive, Résistive** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | **Objet quadrant continu 1** |
| Nœud | *Lien avec un nœud* |

**CHAMPS DE L'INTERFACE ELECTRIQUE CHARGE**

| | |
|---|---|
| **Convention** | **Récepteur** |
| **Type** | **Courant** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | **Sans, Inductive, Résistive** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | **Objet quadrant continu 1** |
| Nœud | *Lien avec un nœud* |

**SCHEMA EQUIVALENT**

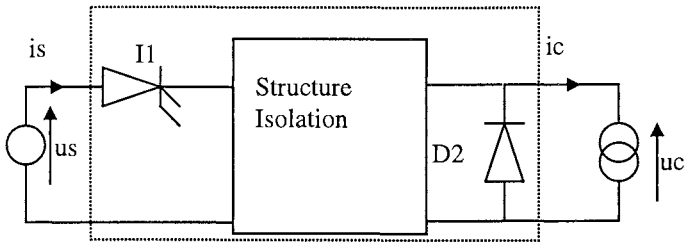



**Figure A. 8 : Schéma de la structure Cellule Hacheur tension avec Isolation**

**REGLES**

- L'interrupteur I1 est commandé à l'ouverture et à la fermeture, et il contrôle le transfert d'énergie.
- L'interrupteur I1 peut avoir une seule commutation commandée, mais l'autre commutation doit se produire de façon naturelle ( => structure de blocage ou d'amorçage par résonance ).
- L'interrupteur I2 est à commutation naturelle.
- Le transfert d'énergie est caractérisé par la durée de conduction de I1.

  => commande à fréquence fixe, rapport cyclique variable

  => commande à fréquence variable, rapport cyclique fixe.

=> commande à fréquence et rapport cyclique variables => commande MLI
régulation du courant ic.
régulation du courant is (absorption sinus)

## II.2.2. Structure Cellule Hacheur Courant

### ROLE

Elle assure la connexion entre une source de courant continu et une charge en tension continue (ou passive) avec le contrôle du transfert d'énergie.

### INFORMATIONS PROPRES

| | |
|---|---|
| Structure mère | *Structure Cellule Hacheur* |
| Fréquence de commutation | ⇒ *Objet Valeur Fréquence* |
| Nœud courant | ⇒ *Objet nœud* |
| tension | ⇒ *Objet nœud* |
| Nature de la source | *Courant* |
| Quadrant d'entrée | *Quadrant Continu 1 (Récepteur)* |
| Nature de la charge | *Tension* |
| Interrupteur transfert source-charge | *Structure Diode* |
| Interrupteur continuité | *Structure interrupteur* |
| Structure Commande | ⇒ *Structure Commande Mono Interrupteur* |
| Interface électrique source | ⇒ *Objet Interface Electrique Bipolaire Continue* |
| Interface électrique charge | ⇒ *Objet Interface Electrique Bipolaire Continue* |

### STUCTURES COMPOSANTS

Structure Interrupteur avec au moins les champs suivants

| | |
|---|---|
| **Structure mère** | |
| **Segments de fonctionnement** | ⇒ ***Objet segment 1***<br>⇒ ***Objet segment 2*** |
| **Fréquence maximale** | ⇒ ***Objet valeur fréquence fixe*** |
| **Commutations nécessaires** | ⇒ ***Objet commutation 1***<br>⇒ ***Objet commutation 2*** |
| **Commande** | ⇒ ***Objet commande*** |
| **Valeurs caractéristiques maximales** | |
| **Valeurs efficaces caractéristiques maximales** | |
| **Variation des valeurs caractéristiques maximale** | |
| **Interface thermique** | ⇒ ***Objet interface thermique*** |
| **Interface électrique** | ⇒ ***Objet interface électrique*** |

Structure diode

### CHAMPS DE L'INTERFACE ELECTRIQUE SOURCE

| | |
|---|---|
| **Convention** | ⇒ ***Récepteur*** |
| **Type** | ⇒ ***Courant*** |

| | |
|---|---|
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | ⇒ ***Sans, Inductive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ⇒ ***Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

**CHAMPS DE L'INTERFACE ELECTRIQUE CHARGE**

| | |
|---|---|
| **Convention** | ***Générateur*** |
| **Type** | ***Tension*** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | ***Sans, Capacitive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ⇒ ***Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

**SCHEMA EQUIVALENT**

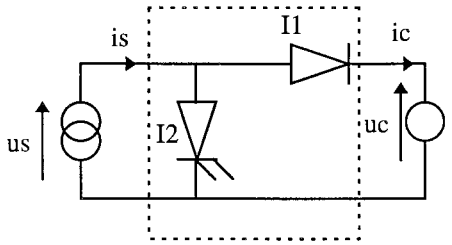



**Figure A. 9 : Schéma de la Structure Cellule Hacheur Courant**

**REGLES**

- L'interrupteur I2 est commandé à l'ouverture et à la fermeture, et il contrôle le transfert d'énergie.
- L'interrupteur I2 peut avoir une seule commutation commandée, mais l'autre commutation doit se produire de façon naturelle ( => structure de blocage ou d'amorçage par résonance).
- L'interrupteur I1 est à commutation naturelle.
- Le transfert d'énergie est caractérisé par la durée de conduction de I1.

=> commande à fréquence fixe, rapport cyclique variable

=> commande à fréquence variable, rapport cyclique fixe.

=> commande à fréquence et rapport cyclique variables

=> commande MLI

régulation du courant ic.

régulation du courant is (absorption sinus)

### *II.2.2.1. Structure Cellule Hacheur Courant avec Isolation*

**ROLE**

Elle assure la connexion entre une source de courant continu et une charge en tension continue (ou passive) avec le contrôle du transfert d'énergie.

**INFORMATIONS PROPRES**

| | |
|---|---|
| Structure mère | *Structure Cellule Hacheur* |
| Fréquence de commutation | ⇒ *Objet Valeur Fréquence* |
| Nœud courant | ⇒ *Objet nœud* |
| Nœud tension | ⇒ *Objet nœud* |
| Nature de la source | *Courant* |
| Quadrant d'entrée | *Quadrant Continu 1 (Récepteur)* |
| Nature de la charge | *Tension* |
| Interrupteur transfert source-charge | *Structure Diode* |
| Interrupteur continuité | *Structure interrupteur* |
| Structure Commande | ⇒ *Structure Commande Mono Interrupteur* |
| Interface électrique source | ⇒ *Objet Interface Electrique Bipolaire Continue* |
| Interface électrique charge | ⇒ *Objet Interface Electrique Bipolaire Continue* |

**STUCTURES COMPOSANTS**

Structure Interrupteur avec au moins les champs suivants

| | |
|---|---|
| **Structure mère** | |
| **Segments de fonctionnement** | ⇒ ***Objet segment 1***<br>⇒ ***Objet segment 2*** |
| **Fréquence maximale** | ⇒ ***Objet valeur fréquence fixe*** |
| **Commutations nécessaires** | ⇒ ***Objet commutation 1***<br>⇒ ***Objet commutation 2*** |
| **Commande** | ⇒ ***Objet commande*** |
| **Valeurs caractéristiques maximales** | |
| **Valeurs efficaces caractéristiques maximales** | |
| **Variation des valeurs caractéristiques maximale** | |
| **Interface thermique** | ⇒ ***Objet interface thermique*** |
| **Interface électrique** | ⇒ ***Objet interface électrique*** |

Structure diode

**CHAMPS DE L'INTERFACE ELECTRIQUE SOURCE**

| | |
|---|---|
| **Convention** | ***Récepteur*** |
| **Type** | ***Courant*** |
| Fréquence | ⇒ *Objet Valeur Fréquence* |
| **Imperfection** | ***Sans, Inductive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ⇒ ***Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

## CHAMPS DE L'INTERFACE ELECTRIQUE CHARGE

| | |
|---|---|
| **Convention** | ***Générateur*** |
| **Type** | ***Tension*** |
| Fréquence | *⇒ Objet Valeur Fréquence* |
| **Imperfection** | ***Sans, Capacitive, Résistive*** |
| **Loi d'évolution** | ***Quelconque, Continue.*** |
| **Quadrant de fonctionnement** | ***⇒ Objet quadrant continu 1*** |
| Nœud | *Lien avec un nœud* |

## SCHEMA EQUIVALENT

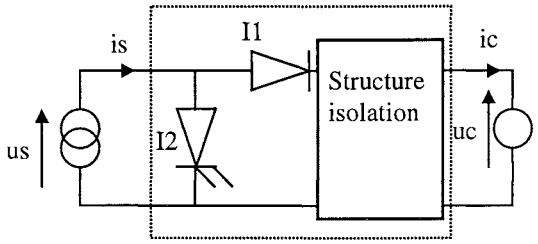



**Figure A. 10 : Le schéma de la Structure Cellule Hacheur Courant avec Isolation**

## REGLES

- L'interrupteur I2 est commandé à l'ouverture et à la fermeture, et il contrôle le transfert d'énergie.
- L'interrupteur I2 peut avoir une seule commutation commandée, mais l'autre commutation doit se produire de façon naturelle ( => structure de blocage ou d'amorçage par résonance).
- L'interrupteur I1 est à commutation naturelle.
- Le transfert d'énergie est caractérisé par la durée de conduction de I1.

=> commande à fréquence fixe, rapport cyclique variable

=> commande à fréquence variable, rapport cyclique fixe.

=> commande à fréquence et rapport cyclique variables

=> commande MLI

régulation du courant ic.

régulation du courant is (absorption sinus)

# Annexe III : Les classes participant à la modélisation des fonction

## III.1. Les classes élémentaires de notre modèle fonctionnel

Nous présentons ici les éléments de notre modélisation fonctionnelle. Cette dernière est orientée objet, c'est-à-dire qu'elle repose sur un modèle informatique à base de classe [OUS97], [NGU92]. Cependant, avant de développer la modélisation en objets des fonctions, nous devons présenter les classes élémentaires de notre modèle fonctionnel. Celles-ci serviront de classes mères pour la définition des classes avec une sémantique développée pour le modèle fonctionnel. La figure A.11 montre la racine de l'arbre des classes de notre modèle.

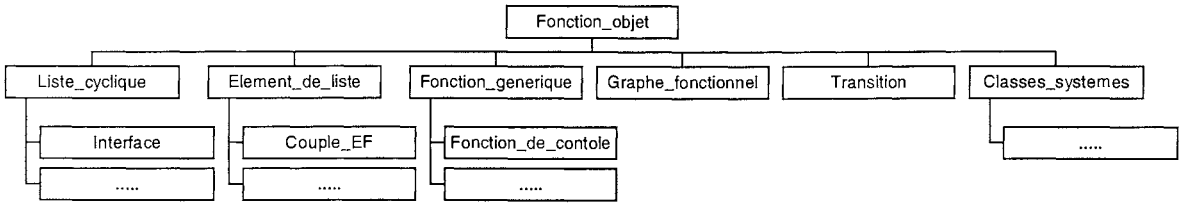



**Figure A. 11 : La racine de notre arbre des classes**

- La classe racine de cet arbre -Fonction_objet- ne comporte aucun champ, mais permet de définir des comportements communs à toutes les classes de notre modèle. Elle est utile notamment pour les interactions graphiques ou les opérations de sauvegarde de contexte.

- La classe Classes_systemes et ses descendantes permettent de définir des objets nécessaires au fonctionnement du modèle. Ces classes sont utilisées pour régler des problèmes techniques d'instanciation des autres classes et de manipulation des objets. Elles ne servent qu'à gérer des problèmes d'implantation informatique et sont donc sans intérêt direct pour le modèle fonctionnel même.

- Les classes Liste_cyclique et Element_de_liste, permettent de mettre en place une gestion de liste de façon indépendante par rapport à un environnement de programmation (ou langage). La modélisation informatique de celles-ci est illustrée par la figure A.12. L'objet de type Liste_cyclique constitue le corps de la liste et pointe sur le premier élément de la liste. Les objets de type Element_de_liste constituent alors chacun un maillon d'une chaîne fermée.

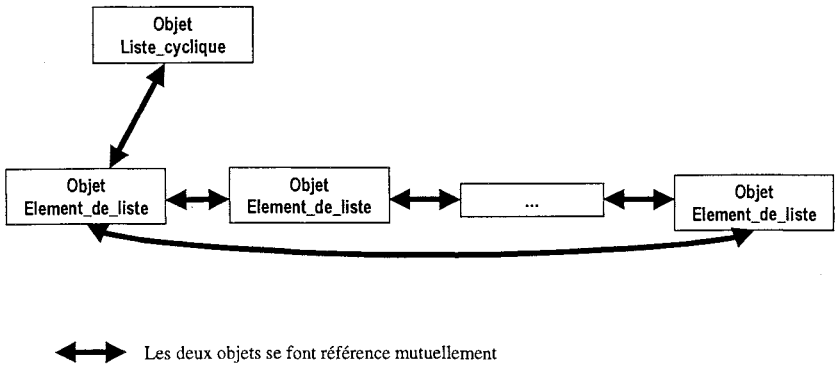



**Figure A. 12 : La modélisation informatique des listes**

D'un point du vue informatique, ces deux classes possèdent toutes les méthodes de gestions des listes à partir des structures données sur les tableaux A. 2 et A. 3 (pour la création de listes, l'ajout d'éléments, ...).

| Classe Liste_cyclique<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers l'objet de type Element_de_liste considéré comme le premier maillon de la chaîne définissant la liste. | *Champs propres* |
| **Elements** | Ce champ est rempli dynamiquement par une méthode à partir d'un parcours des maillons de la chaîne. | |

**Tableau A. 2 : La classe Liste_cyclique**

Avec cette structuration, toutes les sous classes de Liste_cyclique (par exemple la classe Interface) hérite des ses propriétés, c'est-à-dire ses champs et ses méthodes. Il en est de même pour la classe Element_de_liste.

| Classe Element_de_liste<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Liste_cyclique qui représente le corps de la liste. | *Champs propres* |
| **Element_suivant** | Lien vers l'objet Element_de_liste qui est le maillon suivant dans la liste. | |
| **Element_precedent** | Lien vers l'objet Element_de_liste qui est le maillon précédent dans la liste. | |

**Tableau A. 3 : La classe Element_de_liste**

## III.2. La classe Formes_d_onde

La classe Forme_d_onde est présentée sur le tableau A 4. Celle-ci hérite de la classe Liste_cyclique et les éléments de cette liste sont des objets Niveau_moyen. Ainsi, une forme d'onde est représentée par une liste cyclique d'objets Niveau_moyen. En complément, nous ajoutons des champs propres à la classe Forme_d_onde afin de la caractériser.

| Classe Forme_d_onde<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Niveau_moyen de la forme d'onde | *Champs hérités* |
| **Elements** | Objets Niveau_Moyen de la forme d'onde. | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | *Champs propres* |
| **Fréquence** | Valeur numérique pour caractériser la fréquence fondamentale de l'onde moyenne | |

**Tableau A. 4 : La classe Forme_d_onde**

Le paragraphe suivant présente la classe Niveau qui est dans les champs Elements et Tete_de_liste de la classe Forme_d_onde.

## III.3. La classe Niveau

Afin de modéliser une forme d'onde, nous venons de proposer de la décomposer en niveaux. Tous les niveaux possèdent des informations communes que l'on modélise dans une classe appelée Niveau et décrite dans le tableau A. 5. Cette classe hérite de la classe Element_de_liste et elle présente des renseignements sur l'amplitude du niveau, le signe, mais aussi sur la durée (par la différence entre l'angle de début et l'angle de fin). Cette durée est exprimée en angle pour obtenir une représentation relative du temps par rapport à la période de la forme d'onde.

*Remarque : Nous utilisons l'angle 0° pour indiquer le début de la période et 360° est la fin. Cette représentation est moins générique que la notion [0, 1], mais plus courante pour les raisonnements en électronique de puissance.*

| Classe Niveau<br>*Hérite de Element_de_liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Forme_d_onde auquel le Niveau est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Niveau suivant dans la forme d'onde | |
| **Element_precedent** | Lien vers l'objet Niveau précédent dans la forme d'onde | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | Champs propres |
| **Angle_début** | Valeur numérique de l'angle du début du niveau | |
| **Angle_fin** | Valeur numérique de l'angle de la fin du niveau | |
| **Signe** | Valeur pour déterminer le signe du niveau :<br>0 pour nul, 1 pour positif, -1 pour négatif | |

**Tableau A. 5 : La classe Niveau**

A partir de cette classe, nous pouvons définir les classes Niveau_moyen et Niveau_instantane ainsi que leurs différents sous-types. L'arbre partiel de classification des classes héritant de la classe Niveau est illustré sur la figure A. 13. Cette structuration à base d'héritages permet d'ajouter facilement un nouveau type de Niveau. Pour cela, il suffit d'ajouter une sous-classe à la classe Niveau_moyen ou Niveau_instantané.

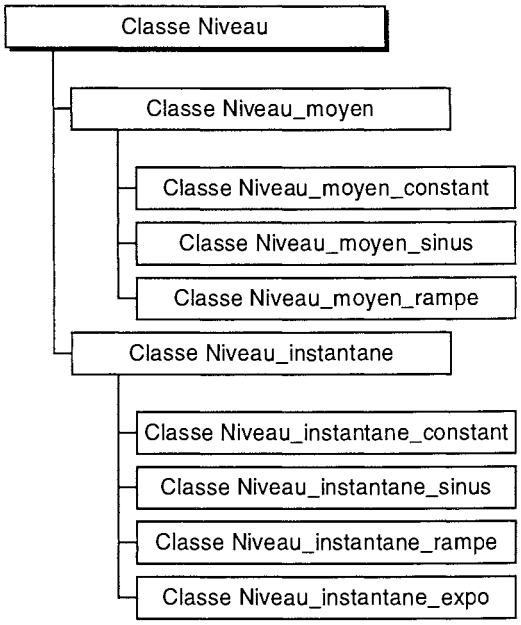



**Figure A. 13 : La classification des niveaux**

Les niveaux hérités de la classe Niveau_moyen (voir le tableau A. 6) permettent de décrire l'aspect moyen de la forme et peuvent être décomposés en objets Niveau_instantané. (voir le tableau A. 7). Ce sont ces derniers qui doivent être pris en compte pour le fonctionnement du convertisseur et la définition des contraintes fonctionnelles.

| Classe Niveau_moyen<br>*Hérite de Niveau* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Forme_d_onde auquel le Niveau_moyen est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Niveau_moyen suivant dans la forme d'onde | |
| **Element_precedent** | Lien vers l'objet Niveau_moyen précédent dans la forme d'onde | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | |
| **Angle_début** | Valeur numérique de l'angle du début du niveau | |
| **Angle_fin** | Valeur numérique de l'angle de la fin du niveau | |
| **Signe** | Valeur pour déterminer le signe du niveau :<br>0 pour nul, 1 pour positif, -1 pour négatif | |
| **Sous_niveau** | Lien vers un objet de la classe Liste_de_niveaux_instantanés dérivée de la classe Liste_cyclique (sans champ propre) et utilisée pour décomposer l'objet en objets Niveau_instantané | Champs propres |
| **Type_de_decomposition** | O : les sous niveaux représentent simplement le niveau moyen<br>1 : les sous niveaux constituent un motif de modulation | |

**Tableau A. 6 : La description de la classe Niveau_moyen**

Le champ sous_niveau permet d'attacher le niveau_moyen à un objet Liste_de_niveaux_ intantanés afin de construire la liste cyclique de d'objets Niveau_instantané.

| Classe Niveau_instantané<br>*Hérite de Niveau* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Liste_de_niveau_instantané auquel le Niveau_instantané est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Niveau_instantané suivant dans la forme d'onde | |
| **Element_precedent** | Lien vers l'objet Niveau_instantané précédent dans la forme d'onde | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | |
| **Angle_début** | Valeur numérique de l'angle du début du niveau | |
| **Angle_fin** | Valeur numérique de l'angle de la fin du niveau | |
| **Signe** | Valeur pour déterminer le signe du niveau :<br>0 pour nul, 1 pour positif, -1 pour négatif | |
| **Niveau_moyen_associe** | Lien vers l'objet Niveau_moyen auquel le Niveau_instantané est associé | Champs propres |

**Tableau A. 7 : La classe Niveau_instantane**

Afin de définir, la sémantique associée à la forme réelle de ces niveaux, nous utilisons en pratique une des sous-classes particularisées. Actuellement plusieurs sous-classes avec une sémantique différente sont définies. Cet aspect est développé en Annexe V.

## III.4. La classe Couple_EF

Afin de pouvoir travailler dans différents domaines énergétiques, différentes classes sont hérités de Couple_EF, comme l'illustre la figure A. 14. Cette organisation permet une extension facile du modèle. En effet pour définir un nouveau type de Couple_EF, il suffit de créer une nouvelle sous-classe par héritage.

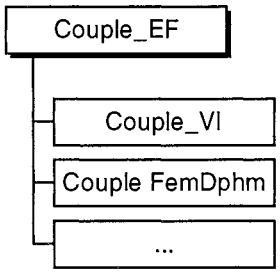



**Figure A. 14 : Hiérarchie de la classe Couple_EF**

Dans le graphe fonctionnel, cette classe Couple_EF permet d'établir les liens entre les interfaces et les connexions (via les liens de transfert d'énergie). La structure de la classe informatique correspondante est présentée dans le tableau A. 8.

**Remarque : pour l'effort, on parlera souvent de différence d'effort (dde), car l'effort d'un objet Couple_EF n'est souvent connu que par référence à l'effort d'un autre objet Couple_EF de la même interface.**

| **Classe Couple_EF**<br>*Hérite de Element_de_liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Interface auquel le Couple_EF est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Couple_EF suivant de l'Interface | |
| **Element_precedent** | Lien vers l'objet Couple_EF précédent de l'Interface | |
| **Connexion_externe** | Lien vers un objet connexion pour établir un lien de transfert d'énergie (voir chapitre 3. I.1) | Champs propres |
| **Appartient_a_elements _de_boucle** | Liens vers les éléments de boucle pour positionner le Couple_EF dans les différentes boucles | |
| **Appartient-a-boucles** | Ce champ est construit dynamiquement par une méthode pour référence les boucles auxquels appartient le Couple_EF (cf. § ch 3 : I.3.2.) | |
| **Liens_de_contrainte** | Liste des Lien_de_contrainte liant ce Couple_EF avec une autre (cf. § ch 2 :II.2.4.1) | |

**Tableau A. 8 : La classe Couple_EF**

## III.5. La classe Interface

La classe Interface est l'une des classes les plus importantes de notre modèle. Sa structure est présentée dans le tableau A.9.

| **Classe Interface**<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Couple_EF de l'interface | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Couple_EF de l'interface. | |
| **Contrainte_cyclique** | Objet Contrainte_cyclique qui mémorise les informations communes à tous les objets de Lien_de_contraintes | *Champs propres* |
| **Liste des contraintes** | Liste d'objets Lien_de_contrainte qui représentent les contraintes entre les objets Couple_EF de l'interface | |
| **Couple_neutre** | Permet de mémoriser l'objet Couple_EF représentant le point neutre de l'interface. | |

**Tableau A. 9 : La classe Interface**

Afin de gérer simplement l'extension du modèle, la classe Interface est organisée de façon hiérarchique comme le montre la figure A. 15. Ses sous-classes n'ont pas de champs propres, mais cette spécialisation permet de déterminer facilement le type de chaque objet Interface. L'ajout d'un nouveau type d'interface se fait seulement en ajoutant une nouvelle sous-classe. Les méthodes de gestion de la classe Interface étant écrites de façon générique, elles seront utilisables pour toutes les sous-classes de la classe Interface

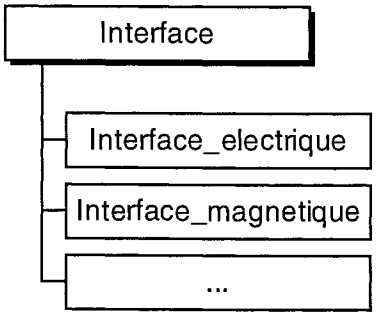



**Figure A. 15 : Organisation de la classe Interface**

Les paragraphes suivants présentent la description des contraintes énergétiques.

## III.6. La modélisation des contraintes

### III.6.1. La classe Contrainte_cyclique

Les objets Lien_de_contrainte d'une même interface sont souvent régis par les mêmes formes d'onde avec un décalage temporel entre elles. C'est notamment le cas lorsque que l'on modélise un système polyphasé équilibré. La classe Contrainte_Cyclique est alors utilisée pour mémoriser les informations communes à tous les liens_de_contrainte d'une même interface, c'est-à-dire portent sur les formes d'onde flux et effort. Sa structure est définie dans le tableau A. 10.

| **Classe Contrainte_Cyclique**<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Forme_d_onde_effort** | Forme effort permettant de fabriquer toutes les contraintes | *Champs propres* |
| **Forme_d_onde_flux** | Forme flux permettant de fabriquer toutes les contraintes | |
| **Dephasage_E/F** | Décalage angulaire entre la grandeur effort et la grandeur flux | |

**Tableau A. 10 : Définition de la classe Contrainte Cyclique**

En résumé, la classe Contrainte_cyclique sert pour caractériser des différents objets de la classe Lien_de_contrainte que nous allons présenter maintenant.

### III.6.2. La classe Lien_de_contrainte

La classe Lien_de_contriante est présentée par le tableau A.11. Le Couple_départ et le Couple_retour définissent un sens de parcours positif pour le flux. Afin de modéliser le fait que la classe Lien_de_Contrainte soit une liste d'objets Etat_de_contrainte, cette classe dérive de la classe Liste_cyclique.

| Classe Lien_de_contrainte<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Etat_de_contrainte du Lien_de_containte | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Etat_de_contrainte attachés. | |
| **Couple_depart** | Lien vers l'objet Couple_EF de départ du Lien_de_contrainte | *Champs propres* |
| **Couple_retour** | Lien vers l'objet Couple_EF de retour du Lien_de_contrainte | |
| **Type_variable_flux** | Type réel de la grandeur flux : ce champ est fixé au moment de la définition de la sous-classe de lien de contrainte Par exemple ce champs est fixé à « Courant » pour la classe Lien_de_contrainte_electrique | |
| **Type_variable_effort** | Type réel de la grandeur effort, par exemple ce champ est fixé à « Tension » pour la classe Lien_de_contrainte_electrique | |
| **Causalité** | Valeur de la causalité du lien de contrainte : une valeur parmi : Effort, Flux, Derivee_effort, Derivée_flux, Passif, Interrupteur, Quelconque | |

**Tableau A. 11 : La classe Lien_de_Contrainte**

D'une façon similaire aux classes héritées d'Interface et aux classes héritées de Couple_EF, la classe Lien_de_contrainte est organisée de façon hiérarchique. De plus, les deux champs propres Type_variable_flux et Type_variable_effort permettent la particularisation des contraintes en terme de nom de variable comme le montre la figure A. 16.

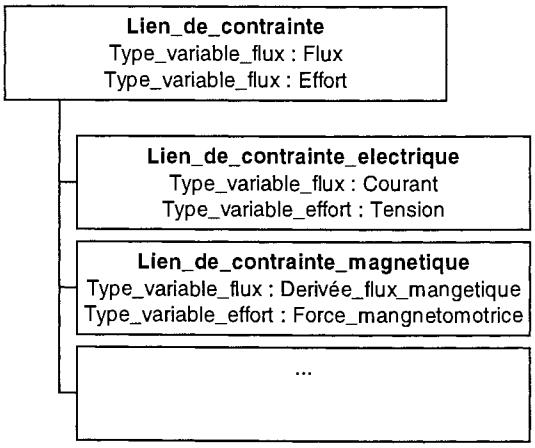



**Figure A. 16 : La hiérarchie des classes Lien_de_contraintre**

## III.6.3. La Classe Etat_de_contrainte

La classe Etat_de_contrainte permet d'étendre la notion de causalité des objets Lien_de_contrainte. Par les objets Etat_de_contrainte, il est possible de définir un niveau imposé (effort ou flux) et un niveau moule. S'il le désire, le concepteur peut aussi définir des contraintes sur le signe de la dérivée pour affiner la précision de son analyse.

| Classe Etat_de_contrainte<br>*Hérite de Element_de_liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Lien_de_containte décomposé | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Etat_de_contrainte suivant de l'Interface | |
| **Element_precedent** | Lien vers l'objet Etat_de_contrainte précédent de l'Interface | |
| **Contrainte_niveau_dde** | Niveau effort à respecter pour ce niveau | Champs propres |
| **Contrainte_niveau_flux** | Niveau flux à respecter pour ce niveau | |
| **Contrainte_valeur** | Valeur numérique avec trois valeurs possibles :<br>0 : contrainte_niveau_dde et contrainte_niveau_flux sont des moules (valeurs indicatives)<br>1 : l'effort est imposé dans contrainte_niveau_dde et ce niveau est alors un moule<br>–1 : le flux est imposé dans contrainte_niveau_flux et ce niveau est alors un moule | |
| **Contrainte_derivee** | Valeur numérique avec trois valeurs possibles :<br>0 : les dérivées ne sont imposées<br>1 : la dérivée de l'effort est imposée<br>–1 : La dérivée du flux est imposée | |
| **Contrainte_signe_derivee** | Valeur numérique avec quatre valeurs possibles :<br>() : le signe de la dérivée n'est pas connu<br>0 : la dérivée imposée est nulle<br>1 : la dérivée imposée est positive<br>–1 : La dérivée imposée est négative | |
| **Liste_de_transition_vers_suivant** | Liste d'objets Transition pour faire le lien vers les objets Etat_de_contrainte pouvant s'enchaîner après l'état courant | |
| **Liste_de_transition_vers_précedent** | Liste d'objets Transition pour faire le lien vers les objets Etat_de_contrainte pouvant intervenir avant l'état courant | |

**Tableau A. 12 :La classe Etat_de_Contrainte**

Les deux derniers champs d'Etat_de_containte permettent de définir les enchaînements entre les différents objets Etat_de_contrainte. Ces changements sont décrits à travers des objets Transition décrit dans le paragraphe suivant.

## III.7. La Classe Transition

La classe Transition permet de modéliser les conditions de changement d'état des fonctions de contrôle du flux. Un objet Transition mémorise un état de départ, un état d'arrivée ainsi que les conditions de passage (en terme de signe de flux, de commande) entre ces deux états.

| **Classe Transition**<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Etat_de_depart** | Etat_de_contrainte avant la transition | *Champs propres* |
| **Etat_d_arrivee** | Etat_de_contrainte après la transition | |
| **Niveau_de_transition_effort** | () : il n'y pas de condition de niveau<br>1 : la transition se produit à effort positif<br>0 : la transition se produit à effort nul<br>–1 : la transition se produit à effort négatif | |
| **Niveau_de_transition_flux** | () : il n'y pas de condition de niveau<br>1 : la transition se produit à flux positif<br>0: la transition se produit à flux nul<br>–1 : la transition se produit à flux négatif | |
| **Ordre_de_commande** | () : il n'y a pas d'ordre de commande<br>1 : la transition se produit sur un ordre de commande | |

**Tableau A. 13 : La classe Transition**

## III.8. La classe Fonction

Nous pouvons maintenant définir la classe Fonction_generique (Voir le tableau A.14). Celle-ci sert de base pour la définition de toutes les fonctions.

| **Classe Fonction_generique**<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Fonctionnalité** | Texte pour définir la Fonctionnalité | *Champs propres* |
| **Liste_des_interfaces** | Listes des interfaces externes de la fonction | |
| **Liste_des_interfaces_interne** | Listes des interfaces internes de la fonction ; cela sert pour gérer la composition de fonctions | |

**Tableau A. 14 : La classe Fonction_generique**

# Annexe IV : Les classes utiles à la modélisation du graphe fonctionnel

## IV.1. La classe Connexion

Une connexion est modélisée par un objet dont la classe est représentée dans le tableau A. 15. Cet objet permet de situer la connexion dans le graphe fonctionnel en mémorisant les LTE par les références aux Couples_EFs dans son champ Contient_les_couples ; et la position de la connexion dans les différentes boucles par son champ Appartient_a_element_de_boucle.

| **Classe Connexion**<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Etat_de_contrainte du Lien_de_containte | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Etat_de_contrainte attachés. | |
| **Contient_les_couples** | Liste des objets Couple_EF liés par cet objet Connexion | *Champs propres* |
| **Appartient_a_Element_de_boucle** | Listes des objets Element_de_boucle auxquels l'objet Connexion appartient | |

**Tableau A. 15 : La Classe Connexion**

Par le biais leurs interfaces des fonctions et à partir des connexions, nous pouvons maintenant construire les éléments fondamentaux des graphes fonctionnels : les boucles de flux.

## IV.2. Les Boucles

### IV.2.1. La classe Boucle

La classe Boucle est décrite par le tableau A. 16. Les informations sur les deux points de vue complémentaires de la boucle sont représentées par des champs spécifiques.

- Le point de vue topologique est mémorisé par le biais des champs Elements, Appartient_a_graphe_fonctionnel, boucle_fermee et Type_inerface.

- Un point de vue énergétique est réalisé à l'aide des champs liste_des_états, Type_de_conflit et Causalité_dominante.

| Classe Boucle<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Element_de_boucle attaché à l'objet Boucle | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Element_de_boucle attachés. | |
| **Appartient_a_graphe_fonctionnel** | Graphe fonctionnel qui contient la boucle | *Champs propres* |
| **Boucle_fermee** | 1 : la boucle est fermée et peut être analysée | |
| **Type_d_interface** | Classe des interfaces reliées par la boucle, pour connaître la nature de la boucle | |
| **Liste_des_etats** | Liste d'objets Etat_de_boucle pour le point de vue énergétique | |
| **Liens_temporels** | Liens fonctionnels attachés à la boucle. | |
| **Type_de_conflit** | () : la boucle n'a pas été vérifie<br>1 : la boucle est en conflit<br>0 : la boucle n'est pas en conflit | |
| **Causalite_dominante** | Causalité dominante de la boucle | |

**Tableau A. 16 : La classe Boucle**

## IV.2.2. Le point de vue topologique

Afin d'obtenir une manipulation simple des boucles, le point de vue topologique est réalisé par une liste d'objets de type Elément_de_boucle. Ceux-ci sont particularisés en deux sous-classes pour prendre en compte la nature hétérogène d'une boucle (les objets Couple_EF et les objets Connexion).

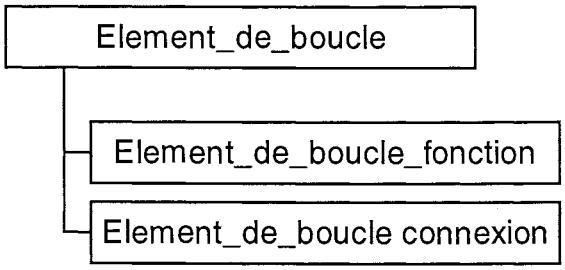



**Figure A. 17 : Hiérarchie de la classe Element_de_boucles**

La classe Element_de_boucle ne possède aucun champ propre, elle est définie pour des raisons purement sémantiques.

Les objets Element_de_boucle_fonctions (tableau A.16) décrivent les trajets du flux à travers une interface en mémorisant son point d'entrée, son point de sortie et dans le cas d'une interface multipolaire les Couple_EF rencontrés au cours du trajet interne (dans l'interface).

| Classe Element_de_boucle_fonction<br>*Hérite de Element_de_boucle* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Boucle auquel l'objet Element_de_boucle_fonction est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Element_de_boucle_connexion suivant dans la boucle | |
| **Element_precedent** | Lien vers l'objet Element_de_boucle_connexion précédent dans la boucle | |
| **Couple_EF_entrant** | Lien vers l'objet Couple_EF, point d'entrée du flux dans une interface | Champs propres |
| **Couple_EF_Intermedaire** | Liste des objets Couple EF rencontrés au cours du trajet du flux dans l'interface | |
| **Couple_EF_sortant** | Lien vers l'objet Couple_EF, point de sortie du flux dans l'interface | |

**Tableau A. 17 :La classe Element_de_boucle_fonction**

Les Objets Element_de_boucle_connexion sont utilisés pour référencer les connexions dans la boucle. Leur structure très simple est donnée dans le tableau A.17

| Classe Element_de_boucle_connexion<br>*Hérite de Element_de_boucle* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Boucle auquel l'objet Element_de_boucle_connexion est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Element_de_boucle_fonction suivant dans la boucle | |
| **Element_precedent** | Lien vers l'objet Element_de_boucle_fonction précédent suivant dans la boucle | |
| **Connexion** | Lien vers un objet Connexion | Champs propres |

**Tableau A. 18 : La classe Element_de_boucle_connexion**

Pour illustrer le fonctionnement de cette modélisation, reprenons l'exemple de la boucle simple de la figure A. 18.

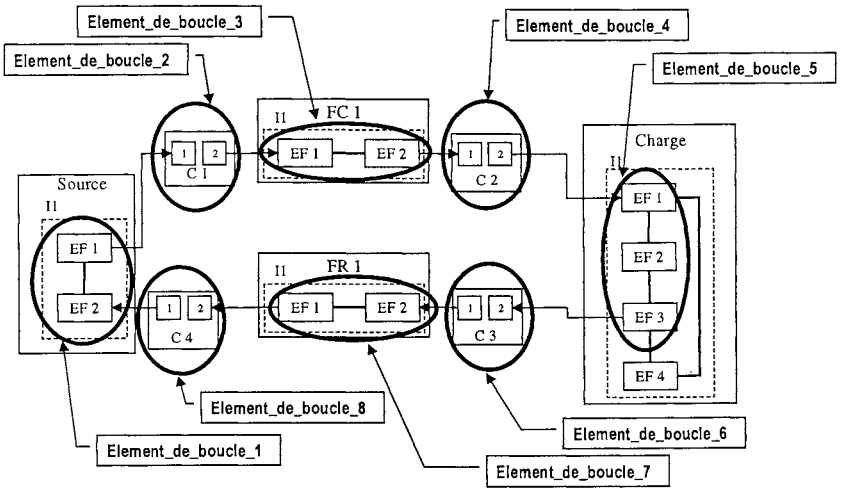



**Figure A. 18 : Un exemple de modélisation d'une boucle**

Celle-ci comporte une fonction source, une fonction charge, une fonction de contrôle et sa fonction de retour. En respectant notre modèle, la topologie de cette boucle est mémorisée comme une liste d'objets Element_de_boucle présenté par le tableau A. 18.

| Position de l'élément dans la boucle | Nom et Nature de l'élément | Intansciation de l'élément de boucle |
|---|---|---|
| 1 | Element_de_boucle_1<br>Classe Element_de_boucle_fonction | Couple_EF_entrant : EF2_I1_Source<br>Couple_EF_Intermediaire : ()<br>Couple_EF_sortant : EF1_I1_Source |
| 2 | Element_de_boucle_2<br>Classe Element_de_boucle_connexion | Connexion :C1 |
| 3 | Element_de_boucle_3<br>Classe Element_de_boucle_fonction | Couple_EF_entrant : EF1_I1_FC_1<br>Couple_EF_Intermediaire : ()<br>Couple_EF_sortant : EF2_I1_FC_1 |
| 4 | Element_de_boucle_4<br>Classe Element_de_boucle_connexion | Connexion :C2 |
| 5 | Element_de_boucle_5<br>Classe Element_de_boucle_fonction | Couple_EF_entrant : EF1_I1_Charge<br>Couple_EF_Intermediaire : EF2_I1_Charge<br>Couple_EF_sortant : EF3_I1_Charge |
| 6 | Element_de_boucle_6<br>Classe Element_de_boucle_connexion | Connexion :C3 |
| 7 | Element_de_boucle_7<br>Classe Element_de_boucle_fonction | Couple_EF_entrant : EF1_I1_FR_1<br>Couple_EF_Intermediaire : ()<br>Couple_EF_sortant : EF2_I1_FR_1 |
| 8 | Element_de_boucle_8<br>Classe Element_de_boucle_connexion | Connexion :C4 |

**Tableau A. 19 : Mémorisation de la topologie de la boucle de la figure 3.2**

## IV.2.3. Le point de vue énergétique

Afin de mémoriser de façon informatique les états de boucle, nous proposons la structuration objet illustrée par la figure A.19. Celle-ci présente deux dimensions pour le point de vue énergétique.

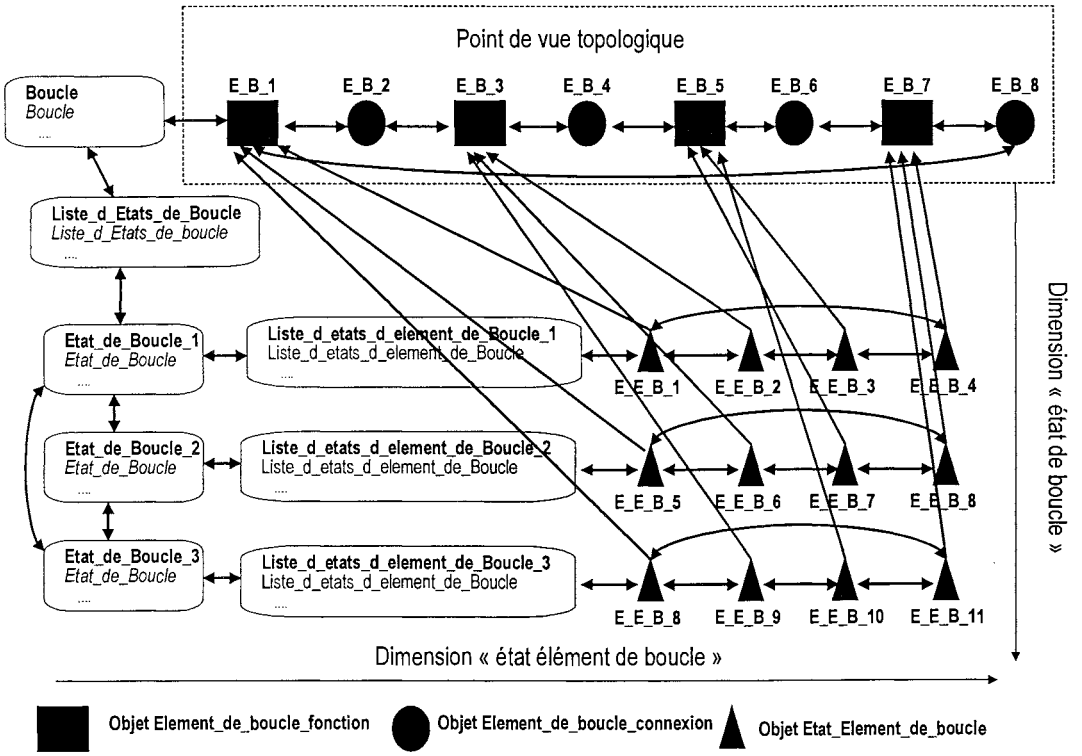



**Figure A. 19 : La modélisation objets des boucles dans la figure A.18**

Un premier objet de type Liste_d_etats_de_boucle (qui hérite de la Classe Liste_cyclique) permet de construire la dimension "état de boucle" par une liste d'objets de classe Etat_de_boucle. Ceux-ci font référence à un niveau de flux dans la boucle, ainsi qu'à des liens conditionnels vers les états suivants (ou précédents). Ils sont décrits par le tableau A. 19.

| Classe Etat_de_boucle<br>*Hérite de Element_de_liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Boucle auquel l'objet Element_de_boucle_connexion est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Element_de_boucle_fonction suivant dans la boucle | |
| **Element_precedent** | Lien vers l'objet Element_de_boucle_fonction précédent dans la boucle | |
| **Niveau_flux** | Objet de classe Niveau_instantané pour indiquer le niveau de flux de l'état | *Champs propres* |
| **Liste_des_etats_d_elements_de_boucle** | Liste_cyclique d'objets Etat_d_element_de_boucle pour mémoriser la cartographie énergétique de la boucle | |
| **Condition_vers_suivant** | Liste des objets Transition qui doivent être vérifiés pour passer à l'état suivant | |
| **Condition_vers_precedent** | Liste des objets Transition qui doivent être vérifiés pour passer à l'état précédent | |
| **Impose_par** | Fonction qui impose (ou qui a permit de déterminer) le niveau flux dans la boucle | |

**Tableau A. 20 : La classe Etat_de_boucle**

Les liens conditionnels sont matérialisés par des objets Transistion_de_boucle (cf. tableau A. 20) qui présente la même structure que les objets Transition attachés aux objets Etat_de_contrainte.

| Classe Transition_de_boucle<br>*Hérite de Fonct_objet* | | |
|---|---|---|
| **Etat_de_depart** | Etat_de_Boucle avant la transition | *Champs propres* |
| **Etat_d_arrivee** | Etat_de_Boucle après la transition | |
| **Niveau_de_transition_effort** | () : il n'y pas de condition de niveau<br>1 : la transition se produit à effort positif<br>0 : la transition se produit à effort nul<br>–1 : la transition se produit à effort négatif | |
| **Niveau_de_transition_flux** | () : il n'y pas de condition de niveau<br>1 : la transition se produit à flux positif<br>0 : la transition se produit à flux nul<br>–1 : la transition se produit à flux négatif | |
| **Ordre_de_commande** | () : il n'y a pas d'ordre de commande<br>1 : la transition se produit sur un ordre de commande | |

**Tableau A. 21 :La classe Transition_de_boucle**

La seconde dimension est construite à partir d'un objet de la classe Liste_des_etats_d_element_de_boucle (définie de façon similaire à la classe Liste_d_etats_de_boucle). Celle-ci est la photographie énergétique de la boucle pour l'objet Etat_de_boucle considéré. Un objet Etat_d_element_de_boucle (voir tableau A. 21) est construit pour chaque objet Lien_de_contrainte de la boucle et détermine les objets Etat_de_contrainte compatibles avec le niveau flux considéré.

| Classe Etat_d_element_de_boucle<br>*Hérite de Element_de_liste* | |
|---|---|
| **Etat_boucle** | Etat de boucle de rattachement de l'objet |
| **Lien_de_contrainte** | Lien de contrainte qui caractérise l'état |
| **Etat_de_contrainte_compatible** | Etat de contrainte compatible avec le niveau de flux |
| **Sens_de_parcours** | Champ pour déterminer si le sens de description topologique est le même que celui de la définition de la contrainte :<br>() : les deux sens sont identiques<br>–1 : les deux sens sont opposés |

**Tableau A. 22 : La classe Etat_d_element_de_boucle**

## IV.3. La Classe Lien_temporels

D'un point de vue informatique, tous les objets Lien_temporel ont la même structure présentée par le tableau A. 23.

| Classe Lien_temporel<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Fin_de_lien attaché à l'objet Lien_temporel | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Fin_de_lien attachés à l'objet Lien_temporel. | |
| **Liste_des_boucles_pointées** | Boucles concernées par le lien temporel :<br>si une seule boucle est référencée alors le lien porte sur cette boucle<br>si plusieurs boucles sont référencées alors le lien est entre ces boucles | *Champs propres* |
| **Liste_des_containtes** | Champ (rempli automatiquement) qui regroupe tous les objets Lien_de_contrainte pointés pour avoir un accès rapide. | |

**Tableau A. 23 : La classe Lien_temporel**

Cette classe est une liste cyclique d'objets Fin_de_lien (voir le tableau A. 24) et mémorise en plus les informations suivantes.

- **La liste des contraintes liées.** Cette liste est créée automatiquement (par une méthode) à partir des objets Fin_de_lien. Elle permet d'obtenir rapidement les informations caractéristiques du lien.

- **La liste des boucles liées.** Cela sert à informer rapidement le concepteur si le lien porte sur une boucle ou entre plusieurs boucles.

Le cœur de la modélisation des liens est la classe Fin_de_lien qui mémorise l'objet Lien_de_contrainte avec l'objet Boucle pour lequel il est considéré. En effet, les liens de contrainte peuvent appartenir à plusieurs boucles.

| Classe Fin_de_lien<br>*Hérite de Element_de_liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Lien_temporel auquel l'objet Fin_de_lien est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Fin_de_lien suivant | |
| **Element_precedent** | Lien vers l'objet Fin_de_lien précédent | |
| **Pointe_vers_liens_de_contrainte** | Objet Lien_de_contrainte pointé | Champs propres |
| **Pointe vers boucle** | Boucle pour laquelle le lien de contrainte doit être considéré | |

**Tableau A. 24 : La classe Fin_de_lien**

Pour pouvoir déterminer la nature les différents liens temporels, ceux-ci sont organisés de façon hiérarchique (voir la figure A. 20). Cette structuration permet d'ajouter un nouveau lien avec une sémantique différente sans avoir à redéfinir les méthodes de gestion.

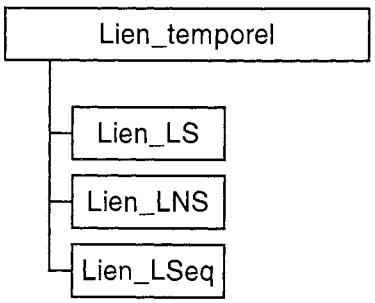



**Figure A. 20 : La hiérarchie des liens temporels**

Nous venons de présenter tous les éléments du graphe fonctionnel. Dans le paragraphe suivant, nous présentons comment nous gérons ces éléments dans leur cycle de vie (création, vérification, spécification).

## IV.4. La Classe Graphe_fonctionnel

Un graphe est modélisé à l'aide d'une classe présentée par le tableau A. 25. Celle ci comporte tous les champs nécessaires pour référencer les composants d'un graphe et des informations nécessaires pour le processus de vérification de la cohérence.

| Classe Graphe_fonctionnel<br>*Hérite de Liste_cyclique* | | |
|---|---|---|
| **Tete_de_liste** | Lien vers le premier objet Etat_de_graphe attaché à l'objet Graphe_fonctionnel | *Champs hérités* |
| **Elements** | Liste (construite dynamiquement) des objets Etat_de_graphe attachés à l'objet Graphe_fonctionnel. | |
| **Contient_les_fonctions** | Ensemble des fonctions du graphe, liste d'objets de type Fonction_generique | *Champs propres* |
| **Contient_les_boucles** | Ensemble des boucles construites par l'expert, liste d'objets de type Boucle | |
| **Contient_les_connexion** | Ensemble des connexions du graphe, liste d'objets de type Connexion | |
| **Liste_des_liens_temporels** | Ensemble des liens temporels du graphe | |
| **Liste_des_boucles_implicites** | Ensemble des boucles implicites construites automatiquement (voir § Chapitre 3 : II.3.1. | |
| **Liste_des_boucles_en_conflit** | Champ pour mémoriser le résultat de vérification a priori de la cohérence | |

**Tableau A. 25 : La classe Graphe fonctionnel**

La liste d'objets de type Etat_de_graphe (classe présentée dans le tableau A. 26) est utilisée pour représenter le mode de fonctionnement du graphe. Il appartiendra à l'expert de définir les méthodes de transition dans la phase spécification.

| Classe Etat_de_graphe<br>*Hérite de Element_de_Liste* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Graphe_fonctionnel auquel l'objet Etat_de_graphe est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Etat_de_graphe suivant | |
| **Element_precedent** | Lien vers l'objet Etat_de_graphe précédent | |
| **Liste_des_Etats_de_boucle** | Listes des objets Etat_de_boucle définissant cette phase | Champs propres |
| **Etat_cohérent** | Résultat du processus de vérification de cohérence complet | |

**Tableau A. 26 : La classe Etat de graphe**

# Annexe V : La définition de s différentes classes du type niveau

Afin de définir la sémantique associée à la forme réelle de ces niveaux, nous utilisons en pratique une des sous-classes particularisées. Actuellement plusieurs sous-classes avec une sémantique différente sont définies :

- **Le « type sinusoïdal ».** Ce niveau définit des morceaux de sinusoïde (qui ne changent pas de signe). La forme de ce niveau correspond à l'équation (A.1). La classe Niveau_sinusoïdale ne comporte pas de champ supplémentaire.

$$U = amplitude \cdot \sin(\frac{angle \cdot \pi}{180})$$
$$\text{pour } angle \in [angle_\min, angle_\max] \qquad \text{(A.1)}$$
$$\text{avec } angle_\min \in [0,180°] \text{ et } angle_\max \in [0,180°] \text{ si } signe = 1$$
$$\text{ou } angle_\min \in [180,360°] \text{ et } angle_\max \in [180,360°] \text{ si } signe = 0$$

- **Le « type constant ».** Ce niveau est utilisé pour représenter les parties constantes de la forme d'onde conformément à l'équation (A.2)

$$\forall angle \text{ on a } U = amplitude$$
$$\text{pour } angle \in [angle_\min, angle_\max] \qquad \text{(A.2)}$$
$$\text{avec } angle_\min \in [0,360°] \text{ et } angle_\max \in [0,360°]$$

- **Le « type rampe ».** Ce niveau sert pour obtenir une évolution linéaire caractérisée par l'équation (A.3). Le tableau A.27 présente les champs propres de la classe Niveau_moyen_rampe (la classe Niveau_instantané_rampe ajoute les mêmes champs) permettant de définir les variables supplémentaires de l'équation.

$$U = amplitude \cdot ((Ampl_fin - Ampl_debut) \cdot \frac{(angle - angle_\min)}{(angle_\max - angle_\min)} + Ampl_debut) \qquad \text{(A.3)}$$
$$\text{pour } angle \in [angle_\min, angle_\max]$$
$$\text{avec } angle_\min \in [0,360°] \text{ et } angle_\max \in [0,360°]$$

| Classe Niveau_moyen_rampe<br>*Hérite de Niveau_moyen* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Liste_de_niveau_instantané auquel le Niveau_instantané est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Niveau_instantané suivant dans la forme d'onde | |
| **Element_precedent** | Lien vers l'objet Niveau_instantané précédent dans la forme d'onde | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | |
| **Angle_début** | Valeur numérique de l'angle du début du niveau | |
| **Angle_fin** | Valeur numérique de l'angle de la fin du niveau | |
| **Signe** | Valeur pour déterminer le signe du niveau : 0 pour nul, 1 pour positif, -1 pour négatif | |
| **Niveau_moyen_associe** | Lien vers l'objet Niveau_moyen auquel le Niveau_instantané est associé | |
| **Ampl_debut** | Valeur numérique dans l'intervalle [0,100] indique en % de amplitude la valeur de l'amplitude au début du niveau | *Champs propres* |
| **Ampl_fin** | Valeur numérique dans l'intervalle [0,100] indique en % de amplitude la valeur de l'amplitude au début du niveau | |

**Tableau A. 27 : La classe Niveau_moyen_rampe**

- **Le "type exponentiel (ou expo)".** Ce niveau sert pour caractériser des évolutions de type exponentiel. L'équation (A.4) caractérise l'évolution de ce niveau. Le tableau A. 28 présente la classe Niveau_instantané_expo qui permet de mémoriser les trois informations supplémentaires qui sont nécessaires : l'amplitude de début, l'amplitude de fin, et la constante de temps de l'exponentielle.

$$U = \frac{1}{e^{\frac{-t0}{\tau}} - e^{\frac{-t1}{\tau}}} \left[ (U0 - U1) \cdot e^{\frac{-t}{\tau}} + \frac{(U0+U1) \cdot (e^{\frac{-t0}{\tau}} - e^{\frac{-t1}{\tau}}) + (U0-U1)(e^{\frac{-t0}{\tau}} + e^{\frac{-t1}{\tau}})}{2} \right]$$

$$\text{avec } t0 = \frac{angle_min}{360 \cdot frequence},\ t1 = \frac{angle_max}{360 \cdot frequence},\ \tau = constante_temps, \qquad \text{(A.4)}$$

$$U0 = Amplitude * Ampl_debut \text{ et } U1 = Amplitude * Ampl_fin$$

$$\text{pour } angle \in [angle_min, angle_max]$$

$$\text{avec } angle_min \in [0,360°] \text{ et } angle_max \in [0,360°]$$

| Classe Niveau_instantané_expo<br>*Hérite de Niveau_instantané* | | |
|---|---|---|
| **Objet_conteneur** | Lien vers l'objet Liste_de_niveau_instantané auquel le Niveau_instantané est attaché | *Champs hérités* |
| **Element_suivant** | Lien vers l'objet Niveau_instantané suivant dans la forme d'onde | |
| **Element_precedent** | Lien vers l'objet Niveau_instantané précédent dans la forme d'onde | |
| **Amplitude** | Valeur numérique de l'amplitude maximum de la forme | |
| **Angle_début** | Valeur numérique de l'angle du début du niveau | |
| **Angle_fin** | Valeur numérique de l'angle de la fin du niveau | |
| **Signe** | Valeur pour déterminer le signe du niveau : 0 pour nul, 1 pour positif, -1 pour négatif | |
| **Niveau_moyen_associe** | Lien vers l'objet Niveau_moyen auquel le Niveau_instantané est associé | |
| **Ampl_debut** | Valeur numérique dans l'intervalle [0,100] indique en % de l'amplitude la valeur de l'amplitude au début du niveau. | *Champs propres* |
| **Ampl_fin** | Valeur numérique dans l'intervalle [0,100] indique en % de l'amplitude la valeur de l'amplitude au début du niveau. | |
| **Constante_Temps** | Valeur numérique de la constante de temps du niveau caractéristique du niveau | |

**Tableau A. 28 : La classe Niveau_instantanée_expo**

**Résumé**

Ce travail de thèse propose d'utiliser l'analyse fonctionnelle dans le processus de conception des structures de convertisseurs statiques. L'analyse fonctionnelle permet d'obtenir un niveau de raisonnement relativement détaché d'une technologie de réalisation. Ainsi, elle permet de guider le concepteur vers l'obtention de plusieurs structures à partir d'un même cahier des charges. L'approche présentée ici est restreinte aux convertisseurs statiques mais peut être étendue aux entraînements électriques. Un modèle est proposé pour représenter les fonctions et les agencer sous forme de graphes fonctionnels. En complément, une analyse de cohérence de ces derniers a été élaborée afin d'aider le concepteur tout au long de sa démarche de conception, en lui signalant des problèmes éventuels de construction des graphes et en lui proposant même parfois des éléments de solution. A partir de là, nous avons mis en œuvre un prototype informatique afin de valider tous ces concepts. Cet outil ne réalise que la vérification de la cohérence et laisse à la charge du concepteur le choix des fonctions et le passage fonction-structure.

**Mots clefs :**

Analyse fonctionnelle, décomposition structurelle, C.A.O, structures de convertisseurs statiques, représentation orientée objet

**Abstract**

This work proposes to use a functional analysis in the design process of static converter structures. Such an analysis is detached from a given technology and aids the designer to obtain several solutions from the same specifications. The approach presented here is restrained to the design of static converters but it can be extended to the study of complete electric drives. In this way, a model is proposed for the functions in order to define and handle them. Add to this, a coherence analysis of the functional graphs is carried out in order to aid the designer during a functional graph building. It may indicate some working problems and even it may propose several solutions to solve them. A software has been made to test these concepts. This tool carries out only the coherence checking and asks the designer to choose the functions and their corresponding structures.

**Keywords :**

Functional analysis, structural decomposition, C.A.D, structure of static converter, object representation