



HAL
open science

UNE NOUVELLE APPROCHE POUR LA CONCEPTION SOUS CONTRAINTES DE MACHINES ELECTRIQUES

Frédéric Wurtz

► **To cite this version:**

Frédéric Wurtz. UNE NOUVELLE APPROCHE POUR LA CONCEPTION SOUS CONTRAINTES DE MACHINES ELECTRIQUES. Energie électrique. Institut National Polytechnique de Grenoble - INPG, 1996. Français. NNT : . tel-00729068

HAL Id: tel-00729068

<https://theses.hal.science/tel-00729068>

Submitted on 7 Sep 2012

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

Frédéric WURTZ
Ingénieur **E.N.S.I.E.G.**

pour obtenir le titre de DOCTEUR

de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 mars 1992)

Spécialité Génie Electrique

=====

**UNE NOUVELLE APPROCHE POUR LA CONCEPTION SOUS
CONTRAINTES DE MACHINES ELECTRIQUES**

=====

Le 28 mai 1996

	Composition du Jury:	
Monsieur:	H. MARTINOT	Président
Messieurs:	B. DUBUISSON	Rapporteur
	M. JUFER	Rapporteur
	J. BIGEON	
	A. FOGGIA	
	C. POIRSON	
	J.C. SABONNADIÈRE	

Thèse préparée au sein du Laboratoire d'Electrotechnique de Grenoble

A ma "petite dame"

Remerciements:

Je tiens à remercier,

Monsieur Henri Martinot, directeur de recherche au CNRS et directeur du LAAS qui m'a fait l'honneur de présider le jury de ma thèse,

Monsieur Bernard Dubuisson, directeur adjoint du département S.P.I. du CNRS et Professeur à l'Université de Technologie de Compiègne qui a accepté d'être rapporteur de cette thèse,

Monsieur Marcel Jufer, professeur à l'Ecole Polytechnique Fédérale de Lausanne qui a également accepté d'être rapporteur de cette thèse,

Monsieur Jean-Claude Sabonnadière, Professeur à l'ENSIEG et Directeur du Laboratoire d'Electrotechnique de Grenoble pour la confiance qu'il m'a accordée au cours de mes années passées en thèse et pour sa participation à ce jury de thèse,

Monsieur Claude Poirson, P.D.G. du groupe Radio-Energie pour avoir rendu possible la collaboration entre le LEG et le groupe Radio-Energie à l'origine de ma thèse et pour la confiance qu'il m'a accordée. Je veux aussi sincèrement le remercier pour sa participation à ce jury de thèse,

Monsieur Albert Foggia, professeur à l'ENSIEG pour les discussions scientifiques que nous n'avons jamais manqué d'avoir et pour le soutien et l'intérêt qu'il porte aux travaux que j'ai menés,

Monsieur Jean Bigeon, chargé de recherche au CNRS et directeur de l'équipe Conception et Diagnostic Intégrés du Laboratoire d'Electrotechnique de Grenoble. Je tiens particulièrement à le remercier pour le climat de confiance et d'amitié réciproque dans lequel mes travaux de thèse ont pu se dérouler et qui a permis une réflexion scientifique que je pense mutuelle et fructueuse. Je veux aussi le remercier pour tous les efforts qu'il fournit afin de m'aider à pouvoir poursuivre ma vocation de chercheur. Enfin je tiens à souligner que je lui dois beaucoup: c'est en effet grâce à ses capacités de visionnaire, à sa volonté et à son opiniâtreté que l'équipe Conception et Diagnostic Intégrés a vu le jour, ce qui m'a donné le cadre et les moyens de m'intéresser à cette problématique scientifique passionnante et pleine d'avenir qu'est l'étude et la mise au point d'outils actifs d'aide à la conception.

Je veux aussi témoigner mon amitié à toutes les personnes du groupe Radio-Energie avec lesquelles j'ai eu l'occasion de travailler.

Je pense particulièrement à Monsieur Pierre-André Gardic que, au delà de ses qualités humaines profondément appréciées, je veux remercier pour avoir compris l'intérêt de nos travaux et avoir été l'un des acteurs principaux ayant rendu mon travail de thèse possible.

Je veux aussi remercier personnellement Monsieur Téodorowicz, d'abord pour la confiance qu'il m'a accordée, mais aussi pour tout le plaisir que j'ai eu à travailler avec lui grâce à sa disponibilité, sa gentillesse, sa compétence et sa volonté sincère de partager son expérience et son savoir.

Je tiens à saluer aussi Philippe Rapiteau dont les talents font merveilles sur la plate forme d'essais du groupe Radio-Energie.

Je tiens aussi à remercier mes collègues du LEG qui ont bien voulu travailler avec les premières versions de PASCOSMA, m'aidant ainsi à valider et à perfectionner l'approche. Ces remerciements vont tout d'abord à Christian Chillet avec lequel j'ai eu la collaboration la plus ambitieuse et la plus fructueuse. Mais je n'oublie pas Stéphane Bergeon, Jean-Marc Bouche, Joël Conrad, Laurent Gerbaud, Christophe Lechevalier et James Roudet avec lesquels j'ai pu travailler avec efficacité et plaisir.

Je veux aussi saluer ceux qui par leur disponibilité, leur soutien et les services rendus m'ont beaucoup aidé lors des derniers mois de la thèse:

Jean Marc Bouche est assurément celui à qui je dois le plus. Qu'il soit remercié pour tous les services qu'il m'a rendus et qu'il sache que qu'il peut compter sur moi de la même manière que j'ai pu compter sur lui.

Mais je veux aussi saluer et remercier Alain Bolopion pour la disponibilité et la confiance que l'on peut avoir en lui qui n'ont d'égal que la pertinence de ses analyses et de ses remarques.

Je veux enfin remercier Laurent Gerbaud pour sa disponibilité, mais surtout pour le soutien et les conseils qu'il n'a jamais manqué de m'apporter.

Je tiens aussi à saluer mes amis de l'équipe Conception et Diagnostic intégrés que je n'ai pas encore eu l'occasion de citer: que Régis Darnault, Eric Escande, Florence François et Marc Richomme soient assurés de mon amitié et de ma sympathie.

Enfin, je veux dire combien l'ambiance de travail a été chaleureuse et agréable lors de ces années de thèse passées au LEG grâce aux amis que j'ai pu continuer à y côtoyer ou à y connaître. Je veux donc saluer au hasard, et de manière non exhaustive, Pascal Petin, Samuel Nughes, Patrick Eustache, Edit Clavel, Bruno Cogitore, Sébastien Adenot ...

TABLE DES MATIERES

Introduction:	2
I. Etat de l'art de la CAO dans le génie électrique et nécessité d'une nouvelle approche	6
I.1. La réalité de l'activité de conception pour l'électrotechnicien: une activité d'Analyse Assistée par Ordinateur plutôt qu'une activité de Conception Assistée par Ordinateur.....	6
I.2 Notre objectif: la définition et la réalisation d'un véritable outil d'aide à la conception adapté au Génie Electrique.....	6
I.3 La problématique de la réalisation de logiciels de CAO ayant des capacités de conception automatique.....	8
I.4 Vers la possibilité d'une indépendance entre la connaissance de l'objet à concevoir et le processus de conception.....	10
I.5 Etat de l'art des logiciels de CAO en génie électrique ayant une faculté de conception automatique.....	10
I.5.a Les logiciels de conception utilisant une approche procédurale.....	11
I.5.b. Les logiciels de conception s'appuyant sur les techniques d'optimisation.....	12
I.5.b.1. Les logiciels d'optimisation utilisant un programme d'analyse et un programme de calcul de sensibilité développés spécifiquement.....	14
I.5.b.2. Les logiciels d'optimisation utilisant un programme d'analyse numérique générique comme outil d'analyse et de calcul de sensibilité.....	16
I.5.c. La conception automatique en utilisant les techniques de l'Intelligence Artificielle: les systèmes experts.....	17
I.6. Une nouvelle approche pour faire de la CAO en génie électrique.....	21
I.6.a Vers une approche surpassant les freins et les limites des approches habituelles.....	21
I.6.b. Vue synthétique de la nouvelle approche proposée.....	24
I.6.c. Place de la nouvelle approche proposée dans la problématique de la Conception Assistée par Ordinateur en génie électrique.....	25
Conclusions du chapitre:.....	28
II. PASCOSMA: une nouvelle approche du Dimensionnement Assisté par Ordinateur sous contraintes de machines électriques	30
Introduction.....	30
II.1. Choix de l'algorithme d'optimisation au coeur du processus de dimensionnement.....	31
II.2. Méthodologie de génération automatique et d'utilisation d'un logiciel de CAO proposée par la nouvelle approche.....	34
II.2.a. 1 ^{er} niveau de la nouvelle approche: cadre méthodologique permettant d'assurer la génération automatique du logiciel de conception.....	34
II.2.a.1. Saisie des équations du modèle analytique de la machine à dimensionner.....	36
II.2.a.2. Recherche des paramètres d'entrée et de sortie du modèle.....	36
II.2.a.3. Calcul symbolique et automatique des paramètres de sortie en fonction des paramètres d'entrée.....	38
II.2.a.4. Programmation automatique des expressions des paramètres de sortie en fonction des paramètres d'entrée: <u>génération du programme d'analyse</u>	39
II.2.a.5. Calcul symbolique et automatique des dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée.....	39
II.2.a.6. Programmation automatique des dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée: <u>génération du programme de calcul de sensibilité</u>	40
II.2.a.7. Couplage du programme d'analyse et du programme de calcul de sensibilité avec un algorithme d'optimisation sous contraintes: création du logiciel de dimensionnement automatique.....	41

II.2.b. 2 ^{ème} niveau de la nouvelle approche: cadre méthodologique permettant une utilisation souple et conviviale du logiciel de conception généré	41
II.2.b.1. Notion de souplesse d'utilisation du logiciel de dimensionnement et sa mise en oeuvre.....	42
II.2.b.2. Notion d'utilisation transparente des algorithmes d'optimisation et sa mise en oeuvre.....	44
II.2.b.3. Notion d'utilisation conviviale du logiciel de dimensionnement et sa mise en oeuvre.....	50
II.3 Fonctionnement du logiciel de Dimensionnement Assisté par Ordinateur généré grâce à la nouvelle approche	53
II.3.a Fonctionnement du logiciel de dimensionnement automatique généré: le cycle d'exploration de l'espace des solutions utilisé	54
II.3.b. Fonctionnement capable de traiter tous les cahiers des charges contraints possibles.....	57
II.3.c. Limites de fonctionnement de l'application de dimensionnement.....	58
II.4 Prise en charge des modèles analytiques des machines à concevoir proposée par la nouvelle approche	59
II.4.a. Mise en forme du modèle analytique proposée par la méthodologie - Introduction de la notion de modèle structural orienté.....	60
II.4.b Importance de la séparation des paramètres du modèle en paramètres d'entrée et de sortie.....	63
II.4.c Gestion des variables implicites proposée dans la nouvelle approche.....	64
Conclusions du chapitre:	66

III. Un prototype informatique de mise en oeuvre de la nouvelle approche proposée: développements nécessaires, possibilités et mode d'emploi, enseignements apportés

Introduction:.....	68
III.1. Développements nécessaires à la mise en oeuvre de la nouvelle approche PASCOSMA proposée	68
III.1.a. Développement et implantation des fonctionnalités du niveau 1 de la méthodologie PASCOSMA	69
III.1.b. Algorithme d'optimisation utilisé	71
III.1.c. Développements et implantation des fonctionnalités du niveau 2 de la méthodologie PASCOSMA	72
III.1.c.1. Développement des parties incluses dans l'exécutable de dimensionnement	72
III.1.c.2. Développement de l'interface conviviale et autoconfigurable	73
III.2. Une session utilisateur complète avec le prototype réalisé.....	75
III.2.a. Description de l'exemple	75
III.2.b. Utilisation du générateur du logiciel de dimensionnement automatique	75
III.2.c Utilisation du logiciel généré de CAO sous contraintes	77
III.3. Problèmes mis en évidence grâce au prototype informatique et solutions apportées	89
III.3.a. Le risque d'explosion de la taille du code généré: problème, et solution apportée.....	90
III.3.b. Limites du fonctionnement du logiciel de dimensionnement mises en évidence	95
III.4. Démarche et philosophie d'utilisation de la nouvelle approche proposée pour faire du dimensionnement sous contraintes de machines électriques	97
III.4.a. Démarche d'utilisation optimale pour la mise au point du modèle analytique.....	97
III.4.b. Philosophie d'utilisation du logiciel de dimensionnement automatiquement généré.....	98
Conclusions du chapitre:	99
IV. Applications de l'outil et de la méthodologie PASCOSMA pour la conception en génie électrique	102
Introduction:.....	102
IV.1. Génération et utilisation d'applications de type PASCOSMA pour le dimensionnement de machines asynchrones	103
IV.1.a. Objectif	103

IV.1.b. Réalisation d'un modèle structural de machine asynchrone	103
IV.1.b.1. Principes mis en oeuvre dans la réalisation du modèle structural	103
IV.1.b.2. Possibilités de dimensionnement offertes par le modèle structural actuel de dimensionnement de machines asynchrones et améliorations possibles.....	106
IV.1.b.3. Démarche adoptée pour la mise au point du modèle structural.....	107
IV.1.c. Description d'une première application de dimensionnement pour machines asynchrones - démonstration de la capacité à dimensionner une machine complète.....	107
IV.1.d. Description d'une seconde application de dimensionnement pour machines asynchrones - démonstration de la capacité à fournir des résultats exploitables dans un bureau d'étude	110
IV. 2 Autres applications réalisées	112
IV.3. L'approche PASCOSMA: de nouvelles opportunités pour employer le patrimoine de modèles analytiques disponibles en électrotechnique.....	113
Conclusions du chapitre:	114
Perspectives	116
Conclusions	119
ANNEXE A : modèle structural permettant de dimensionner un électroaimant.....	121
ANNEXE B : un exemple de modèle structural permettant de dimensionner des machines asynchrones	127
ANNEXE C : une application de dimensionnement pour machines asynchrones au paramétrage exhaustif - capacité de l'approche PASCOSMA à dimensionner une machine complète	155
ANNEXE D : une application de dimensionnement pour machines asynchrones destinée à un usage en bureau d'études.....	165
BIBLIOGRAPHIE :.....	181
Récapitulatif des figures utilisées dans le manuscrit:	187
LISTE DES SYMBOLES EMPLOYES	190
Récapitulatif des tableaux utilisés dans le manuscrit:.....	200

Introduction

Introduction:

Tout concepteur de machines électriques se trouve en permanence confronté à la difficulté de devoir concevoir et dimensionner des machines:

- à la géométrie souvent complexe,
- dans lesquelles interviennent des phénomènes non linéaires et fortement couplés (*électromagnétique, thermiques, mécaniques, ...*),
- devant répondre à un cahier des charges induisant des contraintes d'égalité ou d'intervalle sur les paramètres,
- et devant encore, si possible, représenter un optimum.

Ces concepteurs ont à leur disposition des logiciels d'analyse dont l'usage s'est à présent banalisé. En revanche ils disposent de rien ou de peu de choses pour les aider dans les phases mêmes de conception telles que le dimensionnement ou l'optimisation.

En réponse à ce constat, ce mémoire présente *une nouvelle approche de la conception sous contraintes de machines électriques*.

Celle-ci permet d'utiliser la puissance toujours plus grande des ordinateurs afin de fournir aux électrotechniciens un outil d'aide à la conception qui soit plus qu'un outil de simulation ou d'analyse, mais bien un outil qui a une capacité d'exploration automatique de l'espace des solutions.

Cette approche offre de grands avantages comme:

- la gestion des contraintes d'un cahier des charges,
- l'interdépendance des phénomènes physiques intervenant dans une machine,
- ou encore l'optimisation des solutions trouvées.

Sa grande originalité est qu'elle apporte en plus:

la possibilité de générer automatiquement le logiciel de conception.

Pour cela elle utilise:

- *massivement des techniques de calcul symbolique et de programmation automatique,*
- *comme connaissance de base, un modèle analytique de la machine à laquelle on s'intéresse.*

Afin de guider le processus de conception, elle emploie des algorithmes génériques d'optimisation numérique sous contraintes.

Dans un premier chapitre intitulé "*Etat de l'art de la CAO dans le génie électrique et nécessité d'une nouvelle approche*", nous montrerons que les solutions classiques disponibles pour réaliser des logiciels de CAO ayant de réelles facultés d'exploration de l'espace des solutions souffrent d'inconvénients majeurs comme des temps de développement importants, des difficultés de maintenance non négligeables ou des temps de réponse lents. C'est pourquoi nous avons introduit notre approche qui doit permettre de surmonter ces limites.

Dans un second chapitre intitulé "*PASCOSMA: une nouvelle approche du Dimensionnement Assisté par Ordinateur sous contraintes de machines électriques*", nous formaliserons tous les aspects théoriques de l'approche proposée et désignée sous le nom de PASCOSMA (*Programme d'Analyse, de Synthèse, de Conception et d'Optimisation de Systèmes Modélisables Analytiquement*). Les points suivants seront ainsi abordés:

- le type d'algorithme d'optimisation dont nous préconisons l'emploi,
- la méthodologie de génération automatique du logiciel de conception que nous avons imaginée,
- le fonctionnement du logiciel de conception automatique obtenu,
- la manipulation des modèles analytiques qui est réalisée.

Dans le troisième chapitre intitulé "*Mise en oeuvre concrète de la méthodologie de conception PASCOSMA*", nous détaillerons le prototype informatique que nous avons développé afin de mettre en oeuvre l'approche formalisée. On évoquera ainsi:

- les choix techniques ainsi que le travail d'implantation nécessaire,
- le mode d'emploi et les fonctionnalités offertes,
- les problèmes mis en évidence par l'implantation concrète et les solutions apportées,
- la démarche et la philosophie d'utilisation optimale que nous permet de recommander l'expérience d'un usage important du prototype résultant.

Enfin, dans le quatrième chapitre intitulé "*Applications de l'outil et de la méthodologie PASCOSMA pour la conception de machines électrotechniques*", nous indiquerons comment nous avons employé notre approche et son prototype pour la résolution de problèmes concrets et complexes de dimensionnement de machines électriques, et quels sont ses apports essentiels.

Ainsi, nous détaillerons:

- le cas concret du dimensionnement de machines asynchrones sur la base duquel:
 - nous exposerons la manière dont nous avons réalisé les modèles analytiques et électrotechniques nécessaires,
 - nous montrerons que notre approche est en mesure à la fois de dimensionner des machines complètes et de fournir des résultats fiables et exploitables dans un contexte de bureau d'études industriel,

- l'ensemble des problèmes de dimensionnement de machines électriques que nous avons abordé et qui démontrera l'ampleur du terrain d'application de ce que nous proposons,
- les perspectives nouvelles que nous apportons.

- CHAPITRE I -

**Etat de l'art de la CAO dans le génie électrique
et nécessité d'une nouvelle approche**

I. Etat de l'art de la CAO dans le génie électrique et nécessité d'une nouvelle approche

I.1. La réalité de l'activité de conception pour l'électrotechnicien: une activité d'Analyse Assistée par Ordinateur plutôt qu'une activité de Conception Assistée par Ordinateur

Les outils disponibles actuellement pour faire de la *Conception Assistée par Ordinateur* (CAO) en génie électrique sont essentiellement des outils d'analyse: ils permettent de simuler les performances d'une machine à partir de la donnée de ses spécifications de construction et de ses conditions d'utilisation [22], [18], [19], [20], [53], [54].

Si ces logiciels résolvent de gros problèmes inhérents à l'activité de conception (*prédiction des performances, limitation du nombre de prototypes,...*), c'est toujours l'utilisateur qui assure le processus de conception en proposant et en modifiant les différentes structures de machines qui vont permettre d'aboutir à la machine finale.

Ceci permet d'arriver aux deux conclusions suivantes:

1° D'une part, et comme certains auteurs l'ont souligné [62], les outils informatiques que nous venons d'évoquer devraient plus correctement s'appeler outils d'Analyse Assistée par Ordinateur, plutôt qu'outils de Conception Assistée par Ordinateur.

2° D'autre part, offrir aux électrotechniciens des outils apportant une réelle aide dans le processus de conception lui-même, et pas simplement dans le processus d'analyse, aussi amplement diffusés et facilement utilisables que les outils d'analyse actuellement disponibles, est une réalité qui demande encore largement à se concrétiser.

I.2 Notre objectif: la définition et la réalisation d'un véritable outil d'aide à la conception adapté au Génie Electrique

Notre objectif est de contribuer à ce que la disponibilité de véritables outils d'aide au processus de conception lui-même devienne une réalité dans un domaine comme le génie électrique.

Dans ce but, notre objectif est:

La définition d'outils de CAO ayant une capacité d'exploration automatique de l'espace des solutions face à un cahier des charges

Dans la suite de ce manuscrit, on appellera outils de CAO des outils répondant à l'objectif que nous cherchons à atteindre. C'est pourquoi les outils de simulation, communément désignés sous le terme d'outils de CAO, seront appelés outils d'analyse.

Par ailleurs, *afin que les outils de CAO que nous nous proposons de réaliser soient aussi largement utilisés que peuvent l'être les logiciels d'analyse dans un domaine comme le génie électrique, nous montrerons qu'ils doivent vérifier les cinq critères suivants:*

* Critère N°1: ils devront être *utilisables par tout électrotechnicien* et, en particulier, les électrotechniciens n'ayant pas de compétences particulières en informatique.

* Critère N°2: ils devront *offrir des facilités de mise en oeuvre et des temps de développement rapides*. Cette mise en oeuvre et ce développement devront pouvoir être assurés par un électrotechnicien non informaticien.

* Critère N°3: ils devront être *facilement maintenables et évolutifs*, et ceci devra pouvoir être assuré par un électrotechnicien non informaticien. On précise que:

- la maintenabilité correspond à la possibilité de corriger les éventuelles imperfections de l'outil de CAO existant,
- l'évolutivité correspond à la possibilité de pouvoir intégrer les évolutions concernant les machines à concevoir.

* Critère N°4: ils devront avoir *des temps de réponse rapides* c'est-à-dire fournir des solutions au cahier des charges posé en un temps raisonnable.

* Critère N°5: Ils devront avoir *une faculté à concevoir une machine électrique complète* en intégrant tous les phénomènes qui peuvent intervenir dans cette machine et qui peuvent influencer sa conception.

I.3 La problématique de la réalisation de logiciels de CAO ayant des capacités de conception automatique

Réaliser des logiciels de CAO ayant une réelle faculté d'explorer seuls l'espace des solutions nécessite de trouver des voies pour automatiser la résolution du problème suivant que nous appellerons problème inverse:

Problème inverse:

Pour des performances exigées, trouver les caractéristiques d'une machine (structure, dimensions et conditions d'utilisation).

Nous appellerons problème direct, la réciproque du problème inverse, dont une formulation pourrait être:

Problème direct:

Pour des caractéristiques données d'une machine (structure, dimensions et conditions d'utilisation), déterminer les performances.

Le problème direct est en général *directement* résolu avec un ordinateur en utilisant des logiciels d'analyse, d'où son nom.

Le problème inverse ne peut *pas être résolu de manière directe*, et ceci d'autant moins, si l'on intègre les deux considérations suivantes:

1° Pour un ensemble de performances exigées, on peut souvent déterminer plus d'une machine, voire une infinité de machines associées. Le problème inverse ne connaît donc pas, la plupart du temps, de solution unique, alors qu'en revanche, le problème direct ne connaît habituellement qu'une seule solution physique possible.

2° Le problème inverse est beaucoup plus complexe que le problème direct car les performances et les caractéristiques de la machine à obtenir sont le plus souvent soumises à des contraintes [66], l'ensemble de ces contraintes constituant le cahier des charges. Ainsi, l'encombrement maximum de la machine pourra être imposé en vue de fournir une puissance minimale, tout en ne dépassant pas des échauffements eux-mêmes limités par des valeurs maximales,...

Ces difficultés impliquent que le problème inverse, implanté sur un ordinateur, sera nécessairement résolu sous forme d'un processus itératif [13] composé d'une suite d'essais et d'erreurs qui doivent permettre d'aboutir à une ou plusieurs solutions (on parlera de processus de type "cut-and-try" [67] ou "trial-and-error" [26]). Ce problème, représentatif de l'activité

de conception d'une machine, ne se résout que de manière indirecte. C'est pourquoi on parlera de "*problème inverse*" ou de "*problème indirect*".

Ceci étant établi, *toute la difficulté de la réalisation de véritables logiciels de CAO est de mener correctement le processus itératif*. Le succès et la bonne marche de celui-ci sont en effet constamment menacés par le phénomène de *l'explosion combinatoire de l'espace des solutions possibles*. Ainsi, pour définir complètement la structure, les dimensions et les conditions d'utilisation d'une machine susceptible de répondre aux performances exigées, il n'est pas rare qu'en génie électrique, 30 paramètres soient nécessaires. Si chacun de ces paramètres a 10 valeurs possibles, l'espace des combinaisons à explorer a une taille de 10^{30} . Avec un temps de traitement moyen d'une seconde par combinaison, l'évaluation de toutes les combinaisons possibles exigerait 317×10^{22} ... années !!!

La conception de véritables logiciels de CAO doit donc se faire en ayant à l'esprit les deux éléments suivants:

1° L'exploration systématique de tout l'espace des solutions est à proscrire car le plus souvent impossible à mener dans des délais raisonnables (*cela a été souligné dès la mise au point des premiers outils de conception automatique [26]*).

2° Les combinaisons qui seront examinées au cours du processus itératif devront être judicieusement choisies car la proportion de combinaisons pouvant représenter une solution possible est souvent infime face à l'ensemble de toutes les combinaisons possibles.

Face à ces contraintes, on est tenté de répondre qu'une certaine connaissance et un certain savoir-faire sont nécessaires afin de guider le processus de conception automatique. Toute la problématique est alors de définir quelle connaissance, quel savoir-faire et quelle forme ces éléments peuvent avoir.

Nous allons par la suite détailler les diverses réponses qui ont pu être apportées à cette question. Puis nous détaillerons la réponse que nous prétendons y apporter en incorporant les cinq critères de facilité d'emploi, de simplicité de mise oeuvre, de maintenabilité, d'évolutivité et de temps de réponse que nous nous imposons d'atteindre. Malgré tout, on peut déjà affirmer qu'il faut s'intéresser à deux aspects différents et pourtant complémentaires qui sont:

* La connaissance de l'objet susceptible de répondre au cahier des charges. En effet, pour concevoir la machine susceptible de répondre au cahier des charges, il est conseillé de savoir de quoi cette machine est constituée et quelles peuvent être les influences mutuelles des différents paramètres de celle-ci.

* Le processus utilisant cette connaissance afin de s'orienter vers la ou les bonnes solutions. En d'autres termes, il s'agit de disposer d'une stratégie de résolution.

I.4 Vers la possibilité d'une indépendance entre la connaissance de l'objet à concevoir et le processus de conception

Un examen attentif des deux éléments nécessaires pour assurer un processus de conception efficace nous a amenés à la réflexion suivante:

la connaissance de l'objet à concevoir dépend de cet objet et n'est donc pas générique. Mais, comme nous le montrerons par la suite, le processus de conception utilisant cette connaissance peut être très fortement générique, c'est-à-dire largement indépendant de la connaissance qu'il va utiliser, et donc réutilisable pour chaque nouveau type de machine.

C'est pourquoi, à la liste des critères (*donnée dans le paragraphe I.2*) que devra atteindre notre outil de conception, nous ajouterons encore le sixième critère suivant:

* Critère N°6: *L'indépendance la plus totale possible du mécanisme assurant le processus d'exploration de l'espace des solutions par rapport à la connaissance de l'objet à concevoir.*

Comme nous le verrons par la suite, la satisfaction de ce critère est primordiale car elle rend réalisable la satisfaction d'autres critères que nous avons définis (*critère N°2 = la facilité de mise en oeuvre et de développement et critère N°3 = facilité de maintenance et d'évolutivité*).

I.5 Etat de l'art des logiciels de CAO en génie électrique ayant une faculté de conception automatique

Dans ce paragraphe, nous allons passer en revue succinctement les solutions qui existent actuellement en électrotechnique pour aboutir à des logiciels de CAO ayant une faculté intrinsèque de conception. Nous avons scindé ces solutions en trois catégories:

- 1° Les logiciels de conception utilisant une approche procédurale,
- 2° Les logiciels de conception s'appuyant sur des techniques d'optimisation,
- 3° Les logiciels de type système expert.

Une fois ces solutions passées en revue, nous analyserons les freins à l'emploi de ces approches qui, selon nous, expliquent que la disponibilité d'outils ayant des facultés de conception est loin d'être une réalité aussi concrète que la disponibilité d'outils d'analyse.

I.5.a Les logiciels de conception utilisant une approche procédurale

Le principe de cette approche consiste à mettre au point une procédure de conception visant à faire séquentiellement les différentes étapes nécessaires. Souvent, cette procédure va comporter des tests et des retours en arrière pour remettre en cause des choix non judicieux. Cette procédure est ensuite implantée grâce à un langage de programmation quelconque ou en utilisant éventuellement des outils de plus haut niveau tels que des tableurs [41]. Ce type d'outils d'aide au processus de conception par ordinateur a été mis au point, en génie électrique, dès que les ordinateurs ont commencé à être disponibles [66], [67], [13] et reste assurément l'un des plus employés à ce jour. Les outils obtenus permettent incontestablement d'offrir une aide pour les raisons suivantes:

- * Une fois développés, ils sont généralement utilisables par un électrotechnicien non informaticien, ce qui leur permet de satisfaire au critère N°1 que nous avons défini.
- * Leurs temps de réponse sont parmi les plus rapides dans la gamme d'outils ayant des facultés de conception automatique que nous passerons en revue. Ils satisfont ainsi au critère N°4.
- * La capacité de dimensionner une machine complète peut devenir une réalité dès lors qu'on est prêt à assumer les développements en conséquence. Ils satisfont ainsi le critère N°5.

Il n'en reste pas moins que les inconvénients inhérents à cette approche font qu'elle est éloignée de l'idéal que nous souhaitons atteindre. Ainsi:

- * Cette approche ne satisfait pas le critère N°2 que nous avons défini, puisqu'elle *n'est ni facile, ni rapide à mettre en oeuvre*, surtout si l'on veut réaliser un logiciel assurant le dimensionnement complet d'une machine. Nous ne comptons pas ici le temps de formalisation de la connaissance, propre à la machine à dimensionner, mais le temps de développement informatique de l'outil, le temps de mise au point de la procédure de conception et le temps de débogage. Par ailleurs, la définition d'une procédure qui

prenne en compte l'interdépendance des phénomènes physiques intervenant dans une machine et les aspects itératifs rallonge le temps de formalisation de la connaissance. Enfin, par rapport au critère N°2, cette approche n'est *pas accessible à un électrotechnicien sans compétence informatique* puisque, dans la phase d'obtention de l'outil de conception la programmation, doit être réalisée.

* Cette approche ne satisfait pas le critère N°3 étant donné que *la maintenabilité et l'évolutivité ne sont pas aisées et ne peuvent pas être assurées sans compétence informatique*. En effet, toute maintenance ou toute évolution passent par le remaniement de procédures ce qui est source d'erreurs. Par ailleurs, ces procédures étant le plus souvent écrites dans un langage de programmation, cela exige les compétences informatiques relatives à l'emploi de ce langage.

* Cette approche ne satisfait pas le critère N°6 puisque *l'indépendance du mécanisme assurant le processus de conception par rapport à la connaissance de l'objet à concevoir n'existe pas*. En effet, la procédure de conception dépend complètement de la connaissance que l'on a de la machine et reste spécifique à cette machine. Ainsi, la procédure de conception utilisée pour concevoir une machine asynchrone ([67], [13]) sera très différente de celle employée pour une machine autosynchrone à aimants permanents [38]. Cette non indépendance explique en grande partie que ce type de logiciel puisse être très difficile à mettre en oeuvre (*non satisfaction du critère N°2*) et difficile à maintenir et à faire évoluer (*non satisfaction du critère N°3*).

I.5.b. Les logiciels de conception s'appuyant sur les techniques d'optimisation

Une autre solution, largement employée en génie électrique pour réaliser des logiciels de conception automatique, a été la réalisation de programmes de dimensionnement utilisant des algorithmes génériques d'optimisation.

Les algorithmes d'optimisation sont des techniques d'exploration automatique d'espaces mathématiques permettant de résoudre le problème P1 suivant:

$$P1 \Rightarrow \left[\begin{array}{ll} \min_{\mathbf{p}} f_{ob}(\mathbf{p}) & \\ \text{avec } g_i(\mathbf{p}) \leq 0 & i=1, l \\ g_i(\mathbf{p}) = 0 & i=l+1, m \\ p_{j\min} \leq p_j \leq p_{j\max} & j=1, k \end{array} \right.$$

où:

* \mathbf{p} est un vecteur de dimension k , contenant les paramètres de l'espace à explorer

- * $fob(p)$ est la fonction objectif dépendant des paramètres
- * $g_i(p)$ sont des fonctions qui dépendent aussi de ces paramètres et qui peuvent être soumises à *des contraintes d'égalité ou d'inégalité*.
- * Les paramètres sont eux-mêmes soumis à des *contraintes d'inégalité*.

Or, comme l'ont démontré certains auteurs [32], [33], la conception, et plus particulièrement le dimensionnement d'une machine, sont équivalents à la résolution du problème P1. Dans le cadre de cette équivalence on peut assimiler, en première approximation, l'ensemble des paramètres p à l'ensemble des paramètres de construction et des conditions d'utilisation de la machine (*dimensions de cette machine, tension d'alimentation...*), l'ensemble des fonctions $g_i(p)$ aux performances et aux contraintes physiques caractéristiques de cette machine (*courant absorbé, puissance utile fournie, rendement, densité de courant, inductions maximales ...*) et $fob(p)$ à une fonction qui pourrait représenter le coût de cette machine. Pour résoudre ce problème, les numériciens ont développé des algorithmes génériques d'ordre mathématique. Les plus efficaces semblent être ceux du type gradient [33], [65], [13], [31]. Ils ont besoin, pour pouvoir fonctionner, des deux éléments suivants:

- * *Un programme d'analyse*. Un programme d'analyse est un programme de calcul qui, pour l'ensemble des paramètres de construction et d'utilisation d'une machine (*donc l'ensemble p*), est en mesure de calculer les performances, les contraintes physiques et le coût de cette machine (*donc l'ensemble constitué des fonctions $g_i(p)$ et de $fob(p)$*).
- * *Un programme de calcul de sensibilité*. Un programme de calcul de sensibilité est un programme qui indique comment vont évoluer les performances et les contraintes physiques lorsque l'on va faire évoluer les paramètres de construction et d'utilisation de la machine. Cette indication est autant qualitative (*quelles sont les performances et les contraintes physiques qui vont augmenter ou diminuer ou rester inchangées suite à l'augmentation d'un paramètre de construction ou d'utilisation*) que quantitative (*de combien vont varier les différentes performances et contraintes physiques*). Le support mathématique de cette information passe par *les dérivées partielles* de toutes les performances et les contraintes physiques par rapport aux paramètres de construction et d'utilisation.

En vue d'utiliser ce genre de techniques, deux approches ont été utilisées en génie électrique pour obtenir un programme d'analyse et un programme de calcul de sensibilité:

- * Dans la première approche, le programme d'analyse et le programme de calcul de sensibilité sont développés chacun de manière spécifique pour la machine à concevoir à partir d'un modèle analytique de cette machine.

* Dans la seconde approche, des outils d'analyse numérique fins, tels que ceux s'appuyant sur la méthode des éléments finis, ont été employés afin de fournir des outils de calculs d'analyse et de calculs de sensibilité génériques.

Nous allons détailler brièvement ces deux approches en mettant en évidence les avantages et les inconvénients qu'elles possèdent par rapport aux critères que nous nous sommes imposés de respecter.

I.5.b.1. Les logiciels d'optimisation utilisant un programme d'analyse et un programme de calcul de sensibilité développés spécifiquement

Cette approche a été beaucoup employée dès 1971 [50] pour dimensionner des machines asynchrones. Une longue série de travaux a été menée toujours sur la machine asynchrone [43], [5], [6], [31], [58], ou d'autres types de dispositifs électrotechniques comme la machine synchrone [33], [32] ou les convertisseurs statiques [68]. Dans les références que nous indiquons ici, les programmes de calculs d'analyse et de sensibilité ont été réalisés sur la base d'un modèle analytique de la machine à concevoir.

L'approche permet assurément de plus s'approcher mieux de l'idéal que nous voulons atteindre:

* Elle répond au critère N°1 que nous avons défini, puisque les programmes développés selon cette approche sont utilisables par un électrotechnicien non informaticien.

* Elle satisfait au critère N°4, dans la mesure où leurs temps de réponse sont parmi les plus rapides dans la gamme d'outils ayant des facultés de conception automatique que nous passerons en revue.

* Elle rend possible le dimensionnement d'une machine complète et répond donc au critère N°5.

* Elle respecte, et ce n'est pas le moindre de ses avantages, le critère N°6. En effet, l'indépendance du processus d'exploration de l'espace des solutions vis-à-vis de la connaissance de l'objet à concevoir y est largement assurée puisque:

-l'algorithme d'optimisation, de nature générique, constitue le processus de conception,

-les programmes d'analyse et de calcul de sensibilité, spécifiques à la machine à concevoir, sont développés de manière indépendante.

* On notera encore, comme intérêt supplémentaire de cette approche, que le mécanisme d'exploration de l'espace des solutions sait en plus gérer les contraintes qui font partie intégrante de la formulation d'un problème inverse de conception.

Cependant, cette approche ne satisfait pas un certain nombre de critères que nous avons définis:

* Ainsi le critère N°2 n'est pas satisfait car cette approche *n'est ni facile, ni rapide à mettre en oeuvre* en raison:

- du temps d'écriture du programme d'analyse,
- du temps de familiarisation et de mise en oeuvre des techniques d'optimisation,
- du temps de débogage du logiciel réalisé,
- du temps d'écriture du programme de calcul de sensibilité. En toute rigueur, cette étape exige de calculer symboliquement, puis de programmer, un nombre considérable de dérivées partielles. Ce travail est si immense qu'il n'est jamais réalisé, du moins dans les travaux menés en électrotechnique dont nous avons rappelé les références précédemment. Les concepteurs essaient alors d'évaluer numériquement ces dérivées partielles ce qui peut être source d'erreurs numériques et rallonge les temps de calcul. Ils s'efforcent aussi souvent de se reporter sur des algorithmes d'optimisation qui n'ont pas besoin des dérivées partielles, mais qui sont moins performants en terme de précision et de rapidité de convergence [33], [65], ...

Par ailleurs, cette mise en oeuvre n'est *pas accessible à un électrotechnicien non informaticien* car la phase de réalisation de l'outil de conception repose sur la programmation dans un langage informatique des logiciels d'analyse et de calcul de sensibilité.

* Le critère N°3 n'est pas respecté, *car la maintenabilité et l'évolutivité ne sont pas aisées*. Comme pour l'approche procédurale, toute maintenance ou toute évolution passent par le remaniement de lignes d'instructions du programme d'analyse ou du programme de calcul de sensibilité écrites dans un langage informatique, ce qui reste source d'erreurs et nécessite des compétences particulières.

I.5.b.2. Les logiciels d'optimisation utilisant un programme d'analyse numérique générique comme outil d'analyse et de calcul de sensibilité

Pour essayer de pallier les inconvénients de l'approche précédente tout en essayant d'en conserver les avantages, des travaux ont été menés afin d'utiliser des programmes de calcul numériques fins comme outil d'analyse et comme outil de calcul de sensibilité. Ainsi, se sont développés des couplages entre des algorithmes d'optimisation et des programmes d'analyse utilisant la méthode des éléments finis [34], [25] ou la méthode des intégrales de frontière [65]. On profite ainsi du niveau de généralité de ces outils d'analyse numérique fine, sachant faire un calcul d'analyse d'une machine, simplement à partir de la donnée de ses spécifications et de ses conditions d'utilisation. Par ailleurs, ils peuvent éventuellement réaliser un calcul de sensibilité, pour peu qu'on leur ajoute les fonctionnalités nécessaires [65]. Le premier avantage de cette approche est donc qu'il n'est pas nécessaire de développer un modèle de la structure de la machine à dimensionner.

Cette approche bénéficie aussi des avantages suivants faisant partie de la liste des critères que nous avons définis:

* Elle satisfait le critère N°1 car les types d'outils qu'elle propose sont utilisables par un électrotechnicien non informaticien puisqu'ils nécessitent la seule saisie des structures des machines à dimensionner. De plus, cette saisie est assurée par l'intermédiaire de l'outil d'analyse numérique qui possède souvent le double avantage:

- d'avoir un niveau de convivialité élevé,
- d'être d'un emploi, à présent, entré dans la culture des électrotechniciens.

* Elle répond au critère N°2 puisque ne nécessitant pas de programmation, elle est rapide à mettre en oeuvre.

* Elle respecte aussi le critère N°3 car on peut considérer qu'elle résout le problème de maintenabilité et d'évolutivité sans compétence informatique particulière. En effet l'utilisateur n'a plus à intervenir dans des lignes de programmes mais fournit les caractéristiques de chacun de ses problèmes par l'intermédiaire de l'interface conviviale de l'outil d'analyse.

* Elle garde l'avantage du critère N°6 stipulant une indépendance aussi large que possible entre le processus de conception de la machine (= *l'algorithme d'optimisation sous contraintes en l'occurrence*) et la connaissance de la machine (*fournie par le logiciel d'analyse*).

* Elle conserve en plus l'avantage de gérer les contraintes par l'intermédiaire de l'utilisation de l'algorithme d'optimisation sous contraintes.

Malheureusement, l'emploi d'outils d'analyse numérique est extrêmement lourd et coûteux en terme de temps de calcul. Il en résulte que cette approche ne satisfait pas deux des critères que nous souhaiterions atteindre, ce qui n'en fait pas l'outil de CAO dont nous voudrions disposer. En effet:

* Le critère N°4 n'est pas satisfait car *les temps de réponses peuvent devenir prohibitifs avant d'obtenir une solution au cahier des charges posé*. Ceci provient bien du fait que le calcul d'analyse et de sensibilité sont tous deux menés grâce à des procédures numériques coûteuses en terme de temps de calcul [65].

* Le critère N°5 n'est pas respecté car il est *impossible d'utiliser cette démarche pour prévoir le dimensionnement complet d'une machine*. En effet, le nombre de paramètres sur lequel on peut jouer dans cette approche est relativement faible (*5 à 10 semble être un maximum dans l'état actuel des connaissances*). Par ailleurs, les logiciels d'analyse numérique fins utilisés ne prennent souvent en compte qu'un type de phénomène intervenant dans la machine. Ainsi, avec un logiciel de calculs par éléments finis, tel que FLUX2D [18], on ne prendra en compte que les phénomènes électromagnétiques, alors que le dimensionnement complet de la machine exige d'intégrer aussi les phénomènes thermiques, aérauliques et mécaniques. Aussi cette approche est-elle adaptée à l'optimisation et aux dimensionnements des problèmes tels que des optimisations de forme locales, et ne semble pas se prêter au dimensionnement ou au prédimensionnement complet d'une machine.

I.5.c. La conception automatique en utilisant les techniques de l'Intelligence Artificielle: les systèmes experts

La dernière catégorie de logiciels ayant des facultés de conception automatique qui ait été explorée en électrotechnique est celle du type système expert. Ils ont pour caractéristique essentielle de mettre en oeuvre des techniques provenant de la recherche en intelligence artificielle. Cette approche, et les différentes méthodes qu'elle met en oeuvre, mériteraient une étude complète et en profondeur. Nous allons ici simplement en faire un passage en revue rapide pour essayer d'en cerner les avantages et les inconvénients.

On notera d'abord à son propos qu'elle permet une large représentation de toutes les connaissances nécessaires à un processus de conception. Ceci concerne:

1° *Les connaissances de l'objet à concevoir*, et notamment de sa structure, en utilisant les paradigmes de la représentation orientée objet [23], [63].

2° *Les connaissances concernant les règles de conception de cet objet*. Pour cela, le formalisme des règles de production (*Si condition1 et condition2 et ... alors action*) est

largement employé. Celui-ci permet de modéliser des règles d'hypothèses, de calcul, d'optimisation et de vérification de la cohérence des solutions trouvées [23], [63]. Ces règles proviennent en général de l'expérience de concepteurs habitués à concevoir les machines auxquelles on s'intéresse, ce qui doit contribuer à s'orienter rapidement vers les bonnes solutions possibles. Par ailleurs, ces approches offrent des mécanismes de manipulation de ces règles, appelés moteurs d'inférences.

L'ensemble vise à modéliser et à reproduire le processus de conception tel qu'il est pratiqué par un expert du domaine, en utilisant à la fois ses connaissances et son processus de résolution du problème. Ceci suppose de pouvoir en modéliser les aspects les plus complexes dont des exemples sont:

- * la dynamique et l'adaptation du raisonnement aux résultats déjà obtenus. Afin d'y parvenir, on utilise notamment des moteurs d'inférences dirigés par les faits [23],
- * la capacité de modéliser et de raisonner sur des connaissances imprécises et incertaines. La solution proposée est de s'orienter vers des systèmes experts utilisant les paradigmes de la logique floue [21].

Par rapport aux critères définis que nous cherchons à atteindre, cette approche en satisfait quelques-uns:

* Le critère N°1: les programmes développés selon cette approche sont en général directement utilisables par des électrotechniciens, d'autant qu'un soin particulier est souvent apporté à la réalisation de leurs interfaces graphiques [21], [23], [63].

* Le critère N°3: les outils de conception de type système expert offrent souvent de grandes possibilités de maintenance et d'évolution par un non informaticien. D'ailleurs, ceci fait généralement partie du cahier des charges de ces systèmes dans lequel l'électrotechnicien, en tant qu'utilisateur, doit pouvoir en permanence rajouter ses connaissances afin de compléter celles du système [63]. Cette dernière opération est facilitée par les langages déclaratifs de haut niveau qu'utilisent ces systèmes et qui permettent une saisie de la connaissance dans un formalisme proche du langage naturel. On évite ainsi le passage par un langage de programmation "ésotérique".

* Le critère N°5: les systèmes experts ont de réelles facultés à concevoir une machine complète. Utilisant des langages symboliques de haut niveau, ils vont même jusqu'à offrir des possibilités de prise en compte des contraintes d'ordre sociologique, économique ou humain de l'environnement dans lequel la machine est produite et dont on sait qu'elles peuvent influencer la conception finale de la machine [37].

* Le critère N°6: les systèmes experts offrent des mécanismes de découplage de la connaissance de l'objet à concevoir et de son processus de conception. Ils permettent ainsi de gérer d'une part les connaissances sur la structure de l'objet à concevoir ainsi que ses règles de conception et d'offrir d'autre part des moteurs d'inférence manipulant toutes ces informations.

Néanmoins, cette approche ne satisfait pas d'autres critères que nous cherchons à atteindre:

* Le critère N°2: *les temps de mise en oeuvre et de développement sont souvent très longs.* Dans ce cadre, la première difficulté à résoudre est la collecte de la connaissance à mettre dans ces systèmes. Celle-ci est généralement recueillie auprès d'experts du domaine par l'intermédiaire d'entrevues, ce qui est un travail long et pour lequel aucune procédure vraiment rigoureuse n'existe. Ensuite, cette connaissance doit être mise en forme afin d'être intégrée dans le système informatique. Ce travail est d'autant plus long que pour devenir efficace, un système expert doit habituellement comporter un nombre important de règles [37]. Par ailleurs *cette mise en oeuvre et ce développement ne sont pas accessibles à l'électrotechnicien non spécialiste des techniques complexes et difficiles qu'il faut mettre en oeuvre.*

* En partie le critère N°3: *même si nous avons souligné précédemment que la maintenance et l'évolutivité sont normalement accessibles à un non informaticien par l'intermédiaire de langages déclaratifs de haut niveau, elle est en fait complexifiée par le problème de cohérence de la connaissance.* Il ne faut pas rentrer de nouvelles données qui pourraient contredire celles existantes et venir ainsi bloquer le raisonnement du système [36]. Ceci devient d'autant plus difficile que le nombre de règles dans le système est élevé. Comme par ailleurs le système ne devient efficace que lorsque ce nombre est élevé, on réalise les difficultés auxquelles on peut être confronté.

* Le critère N°4: *le temps de réponse à un cahier des charges est en effet souvent long.* Celui-ci peut se compter en heures, voire en jours. Ces systèmes payent en fait leur haut niveau de généricité et leur utilisation de langages symboliques par un temps de traitement long.

I.5.d Conclusion à propos des solutions apportées à ce jour: les raisons de leur développement limité

Ce tour d'horizon succinct, mais que nous pensons représentatif de ce qui a été fait en électrotechnique, nous permet de comprendre pourquoi la disponibilité d'outils de CAO ayant

une réelle faculté de conception, c'est-à-dire d'exploration de l'espace des solutions, est loin d'être aussi répandue que la disponibilité d'outils d'analyse.

La raison principale est que toutes les approches qui ont été explorées à ce jour ne satisfont pas l'un, au moins, des critères que nous avons définis. Il ne s'agit pas du critère N°1 imposant la possibilité d'utilisation par tout électrotechnicien, tant ce critère est incontournable. En fait, les approches passées en revue sont handicapées par les éléments suivants:

* Elles exigent *des temps de mise en oeuvre et de développement importants (non satisfaction du critère N°2)* ce qui impose des investissements beaucoup plus élevés que le fait d'acheter un programme d'analyse qui permet de simuler une machine très rapidement. On notera que la seule approche satisfaisant le critère N°2 est celle qui satisfait aussi le critère N°6 qui lui-même impose une indépendance aussi large que possible entre le processus de conception et la connaissance de l'objet à concevoir (*cf. les algorithmes d'optimisation couplés à des outils d'analyse*).

* Souvent elles posent *des problèmes de maintenance et d'évolutivité non simples (non satisfaction du critère N°3)* ce qui incite encore moins à se lancer dans des investissements élevés nécessaires. Ceci est d'autant plus vrai que, par ailleurs, l'évolutivité et la maintenance sont des caractéristiques indispensables dans un domaine en constante évolution comme peut l'être la conception de machines. On notera que les approches ne satisfaisant pas ce critère N°3 ont souvent en commun de pas satisfaire non plus le critère N°6 (*cf. l'approche procédurale*).

* Les approches qui respectent le critère N°6 et ont ainsi des capacités à respecter le critère N°2 de facilité de mise en oeuvre (*cf. les algorithmes d'optimisation couplés à des outils d'analyse*), ou le critère N°3 de facilité de maintenance et d'évolutivité (*cf. les algorithmes d'optimisation couplés à des outils d'analyse, les systèmes experts*) ont d'autres inconvénients comme:

* *des temps de réponse trop longs (cf. les systèmes experts et les algorithmes d'optimisation couplés à des outils d'analyse qui ainsi ne satisfont pas le critère N°4)*

* *l'incapacité de prendre en charge le dimensionnement complet d'une machine (cf. les algorithmes d'optimisation couplés à des outils d'analyse qui ne répondent pas au critère N°5).*

Il en résulte que toutes ces approches, malgré les intérêts certains dont chacune fait preuve et qui n'ont pas tous été nécessairement passés en revue (*cf. la gestion des paramètres discrets que savent prendre en compte les systèmes experts*), connaissent des lourdeurs importantes qui gênent leur emploi.

C'est pourquoi nous pensons que seule une approche respectant à la fois les six critères que nous avons définis permettra d'apporter une solution au fait que l'électrotechnicien est beaucoup moins aidé par l'ordinateur dans la résolution du problème inverse de conception qu'il ne l'est dans la résolution du problème direct d'analyse.

I.6. Une nouvelle approche pour faire de la CAO en génie électrique

I.6.a Vers une approche surpassant les freins et les limites des approches habituelles

Les conclusions du paragraphe précédent expliquent que:

- La première phase du travail de thèse que nous allons exposer dans ce manuel a consisté en la recherche des concepts de l'électrotechnique et de l'informatique qui pourraient être utilisés afin de définir une nouvelle approche pour faire de la CAO.
- L'objectif de cette approche est de satisfaire l'ensemble des six critères que nous avons définis.

Aussi, nous allons reprendre ces critères un à un afin de détailler quelle solution, provenant de l'un ou l'autre domaine, nous proposons afin d'y apporter une réponse.

** Face au critère N°1 d'utilisation du logiciel de conception par un électrotechnicien n'ayant pas de compétence informatique particulière, nous proposons d'apporter une solution classique de type informatique. Il s'agit de mettre en place des mécanismes et des interfaces permettant une utilisation conviviale et souple de ce logiciel et des mécanismes d'exploration de l'espace des solutions qu'il emploiera.*

** Face au critère N°2 imposant d'offrir des facilités de mise en oeuvre et des temps de développement rapides accessibles à un électrotechnicien non nécessairement informaticien, la réponse que nous proposons possède plusieurs facettes:*

** Premièrement, face à la nécessité d'une mise en oeuvre et de procédures de développement rapides et faciles, la solution que nous apportons est d'abord d'ordre informatique. Elle consiste en la recherche d'une possibilité d'automatisation de l'obtention du logiciel de conception. Nous montrerons que ceci est possible en utilisant des techniques développées par l'informatique et qui sont maintenant arrivées à maturité. Il s'agit:*

- * des techniques de calcul symbolique,*

* des techniques de programmation automatique.

Il est en effet clair que si l'obtention du logiciel de conception est automatisée, la mise en oeuvre et le développement seront considérablement accélérés et facilités.

* Deuxièmement, face à la nécessité de rendre la mise en oeuvre et le développement accessibles à un non informaticien, la solution que nous proposons est de limiter son intervention. Nous montrerons qu'il est possible de la réduire à la seule introduction déclarative de la connaissance nécessaire de l'objet à concevoir dans un formalisme auquel il est familiarisé. Ce formalisme est couramment manipulé par lui et ne provient pas de l'informatique comme cela est trop souvent le cas. Il s'agit *du formalisme des équations mathématiques* que l'on peut trouver dans *un modèle analytique de la machine à dimensionner*. Celui-ci présente non seulement l'avantage d'être universellement connu de tous les électrotechniciens mais, en plus, il représente une connaissance de la machine à concevoir.

En effet, les modèles analytiques ont comme caractéristique de relier explicitement les uns aux autres, par les équations qu'ils contiennent, les grandeurs intervenant dans la machine [32], [36]. Grâce à eux, on peut connaître l'interdépendance des différents phénomènes physiques qui interviennent. Ceci est particulièrement important pour les dispositifs électrotechniques dans lesquels ce sont à la fois des phénomènes d'ordre électriques, thermiques, mécaniques et aérauliques qui vont conditionner le fonctionnement et la conception. Ces modèles vont aussi donner la dépendance de tous ces phénomènes avec les paramètres de construction et les conditions d'utilisation de la machine. Or cette connaissance de la machine est primordiale, même si elle est fournie par des lois simples et parfois peu précises dans l'absolu.

On notera aussi que l'on rend cette solution possible en saisissant l'opportunité que nous offre l'électrotechnique, en mettant à notre disposition un grand nombre de modèles analytiques, développés depuis plus de cent ans, et qui permettent de dimensionner ou de prédimensionner un grand nombre de machines comme la machine asynchrone [3], [12], la machine asynchrone linéaire [24], la machine synchrone à aimants permanents [38], [8], [49], ... En effet, comme nous le démontrerons, ces modèles se prêtent à une manipulation par les techniques de calcul et de programmation automatique utilisées pour obtenir le logiciel de dimensionnement. D'autre part, l'utilisation de ces modèles ne peut que contribuer à accélérer le temps de développement car c'est un type de connaissance facilement disponible et très bien formalisé dans les articles et les livres du domaine.

* *Face au critère N°3 imposant des possibilités de maintenance et d'évolutivité sans compétence informatique particulière*, l'automatisation de l'obtention du logiciel de conception que nous avons déjà proposée est la voie que nous envisageons. En effet, celle-ci résout à la fois le problème de maintenance et d'évolutivité:

- Les besoins de maintenance sont réduits, voire annihilés, puisque l'obtention automatique du logiciel de conception est la meilleure possibilité d'obtenir un logiciel sans erreur de programmation, ce qui contribue à sa qualité et à la réduction de ses coûts de maintenance.

- L'évolutivité est grandement facilitée car, si l'on a besoin d'une nouvelle version, on la fait générer sans avoir à modifier des lignes de codes dans une version précédente.

L'ensemble est accessible sans compétence informatique particulière.

* *Face au critère N°4 imposant à l'outil de conception d'avoir un temps de réponse rapide face à un cahier des charges*, la solution est autant d'ordre informatique et mathématique que d'ordre électrotechnique. En effet, le logiciel de dimensionnement que nous nous proposons de faire générer automatiquement fonctionnera en utilisant un algorithme d'optimisation sous contraintes en association avec un programme d'analyse et un programme de calcul de sensibilité, qui eux-mêmes, s'appuient sur un modèle analytique. Or, on a pu évoquer le fait que cette association permettait d'obtenir des logiciels de conception aux temps de réponses qui sont parmi les plus rapides existant.

* *Face au critère N°5 imposant la faculté de concevoir une machine électrique complète*, la solution que nous proposons d'apporter est d'ordre électrotechnique. Elle consiste en l'utilisation des modèles analytiques dont on a déjà eu l'occasion de souligner qu'ils permettent de connaître l'interdépendance de toutes les grandeurs qui interviennent dans une machine électrique.

* *Face au critère N°6 de déconnexion de la connaissance de l'objet à concevoir et du processus de conception*, la solution que nous avons apportée est d'ordre informatique et mathématique. Nous avons déjà eu l'occasion de signaler que les algorithmes d'optimisation sous contraintes sont un mécanisme d'exploration de l'espace tout à fait générique. Ils permettent, bien entendu, la gestion des contraintes, dont on a vu qu'elles font partie intégrante d'un cahier des charges. La connaissance de l'objet à concevoir sera, quant à elle, contenue dans les modèles analytiques que l'on transformera automatiquement en programme d'analyse et de calcul de sensibilité par les techniques de calcul symbolique et de programmation automatique.

I.6.b. Vue synthétique de la nouvelle approche proposée

Si l'on réalise la synthèse de toutes les solutions que nous avons mises en évidence face aux critères que nous voulons atteindre, il apparaît que nous avons posé les bases d'une nouvelle approche pour faire de la CAO sous contraintes.

Celle-ci est caractérisée par deux niveaux de fonctionnement:

Le 1^{er} niveau consiste en la génération automatique du logiciel de conception à partir d'un modèle analytique de la machine à concevoir.

Le principe mis en oeuvre pour générer automatiquement ce logiciel consiste à utiliser des techniques de calcul symbolique et de programmation automatique pour générer, à partir du modèle analytique, un programme de calcul d'analyse et un programme de calcul de sensibilité.

Ces deux programmes sont ensuite liés à un algorithme d'optimisation sous contraintes. L'ensemble permet alors d'obtenir un logiciel ayant la faculté d'explorer automatiquement l'espace des solutions pour résoudre un cahier des charges contraint.

Le 2^{ème} niveau consiste en l'utilisation du logiciel généré afin d'explorer efficacement l'espace des solutions dans le but de trouver des réponses optimales au cahier des charges.

Outre la formalisation et la définition de cette approche pour surmonter les freins et les limites du développement d'outils de CAO qui existent dans le domaine du génie électrique, le travail de thèse que nous allons exposer dans ce mémoire a consisté:

* Pour le 1^{er} niveau, à mettre au point et à formaliser des mécanismes permettant d'assurer le processus de génération de code automatique nécessaire. Ceci a dû se faire en intégrant toutes les contraintes apportées par les capacités des outils informatiques disponibles actuellement, et les contraintes apportées par la modélisation analytique des machines électriques que nous prétendons utiliser. L'objectif final est de démontrer que l'approche est effectivement réalisable et permet de générer des applications de conception pour ces machines électriques.

* Pour le 2^{ème} niveau, à définir des mécanismes afin que le logiciel généré au cours du niveau précédent puisse être utilisé de manière souple et conviviale. Nous verrons notamment que la difficulté a été de mettre en place une interface graphique et conviviale ayant la capacité de se reconfigurer automatiquement pour chaque logiciel de conception que l'on veut générer. Ensuite, nous avons vérifié que les applications de

conception générées étaient effectivement en mesure de trouver des solutions dans des problèmes complexes, typiques du dimensionnement complet de machines électriques.

I.6.c. Place de la nouvelle approche proposée dans la problématique de la Conception Assistée par Ordinateur en génie électrique

Comme nous allons le confirmer dans la suite de ce mémoire, la nouvelle approche proposée permet d'atteindre les six critères que nous nous sommes fixés. Elle devrait donc significativement contribuer à ce que la disponibilité d'outils de CAO ayant des facultés d'exploration automatique de l'espace des solutions devienne une réalité beaucoup plus concrète pour l'électrotechnicien.

Avant d'aborder les détails de cette approche, il reste à en cerner deux aspects qui sont:

- 1° La place qu'elle prend exactement dans la problématique de la conception en général et dans la conception assistée par ordinateur en particulier.
- 2° La manière dont elle s'inscrit dans la lignée des travaux d'élaboration de logiciel de CAO qui ont été menés à ce jour en électrotechnique.

La place de cette approche dans la problématique de la conception en général, et dans la conception assistée par ordinateur en particulier:

*L'approche dont nous venons de proposer les grandes lignes doit permettre de réaliser un type de conception que l'on qualifie de conception routinière [36]. Il s'agit d'une conception dans laquelle on recherche des solutions à un cahier des charges sur la base de structures et de choix bien maîtrisés. C'est un type de conception très pratiqué en électrotechnique où la plupart des machines que l'on construit chaque année en des millions d'exemplaires (*les moteurs synchrones, asynchrones, les contacteurs, les disjoncteurs,...*) ne sont que le résultat de variantes de conception autour de structures et de procédures de dimensionnement pratiquées depuis de longues années. En fait, cette conception s'oppose à la conception innovante [10], [36], non assurée par notre approche et dans laquelle des solutions inédites sont imaginées en vue de répondre au cahier des charges.*

Par ailleurs cette conception routinière peut elle-même se décomposer en différentes phases qui pourraient être:

- * le choix d'une structure possible,
- * le premier dimensionnement de cette structure,

- * le dimensionnement correctif (*afin de trouver un dimensionnement qui respecte les spécifications du cahier des charges*),
- * l'optimisation (*qui consiste à améliorer le dimensionnement, tout en restant dans les limites de spécifications du cahier des charges*),

Par rapport à cette décomposition, on peut d'ores et déjà préciser que:

Notre approche saura peu, ou ne saura pas, prendre en charge le choix de la structure possible face au cahier des charges. A cause des mécanismes d'optimisation que nous avons l'intention d'utiliser et qui se servent d'informations comme les dérivées partielles, le fait de devoir se déplacer dans un espace continu et dérivable est obligatoire. A priori, cela semble incompatible avec le fait de changer de structure au cours du dimensionnement. En effet, ceci s'accompagnera inévitablement d'une discontinuité de l'espace mathématique dans lequel on cherche une solution. On notera que d'autres approches, plus lourdes, telle que l'approche système expert, sont en mesure de prendre en charge ce choix du type de structure.

Notre approche n'a pas la faculté, non plus, de trouver un premier dimensionnement de la structure. Les algorithmes d'optimisation sous contraintes que nous comptons utiliser comme mécanisme d'exploration de l'espace des solutions ont besoin d'un point de départ pour amorcer l'exploration. Celui-ci peut être assimilé au premier dimensionnement de la structure et doit leur être fourni. Là encore, d'autres approches plus lourdes, comme les systèmes experts, sont par contre en mesure de prendre cet aspect en charge.

Cependant, notre approche est parfaitement adaptée au dimensionnement correctif d'une structure puisque les algorithmes d'optimisation sous contraintes possèdent les deux propriétés suivantes:

- 1° La faculté de prendre en charge les contraintes pour trouver une solution finale satisfaisant ces contraintes,
- 2° La faculté de se déplacer dans la partie de l'espace des solutions ne respectant pas les contraintes pour se ramener à la partie de l'espace les respectant.

Enfin, notre approche excelle aussi pour ce qui concerne l'optimisation du dimensionnement puisque le mécanisme d'exploration de l'espace des solutions est constitué par un algorithme dont la vocation première est l'optimisation.

Il résulte de cette analyse que notre approche n'est pas en mesure de couvrir tous les aspects relatifs à la conception et à la conception automatique complète par ordinateur. Il serait donc abusif de prétendre que celle-ci est un outil de CAO complet. Cependant, à la vue des fonctions de dimensionnement et d'optimisation qu'elle sait parfaitement prendre en compte, il est légitime de parler d'approche pour le **D**imensionnement et l'**O**ptimisation **A**utomatique par **O**rdinateur.

Rajoutons que, si l'approche ne couvre pas tous les aspects de la conception, son intérêt pour le génie électrique reste encore énorme car le dimensionnement et l'optimisation de structures de machines bien connues représentent incontestablement la plus grande partie du volume de l'activité de conception pratiquée en génie électrique chaque jour dans le monde.

Enfin, il semble que, si l'on veut s'orienter vers des systèmes assurant toutes les phases d'un processus de conception, des couplages avec des systèmes plus lourds restent nécessaires. En ce sens, notre approche, qui possède de grands avantages par rapport à une approche lourde à mettre en oeuvre comme les systèmes experts, reste cependant complémentaire et non concurrente de celle-ci.

Positionnement de l'approche par rapport aux travaux existants

Le travail réalisé s'inscrit dans la lignée de travaux qui ont été menés dès les années 1960 et qui ont essayé d'utiliser les techniques d'optimisation et les modèles analytiques de machines en vue d'obtenir des logiciels de dimensionnement automatique [50], [51], [43], [5], [6], [31] [32], [33],... Cependant, l'objectif de ces travaux était avant tout:

- de tester et de comparer les capacités des techniques d'optimisation (*performances des différents algorithmes, manière de prendre en compte les contraintes, critères d'arrêt*),
- de commenter les résultats obtenus avec les logiciels de dimensionnement et d'optimisation,
- éventuellement de proposer des modèles de machines adaptés à être utilisés afin de faire de l'optimisation.

Notre travail utilise les résultats de ces études.

Cependant, son originalité est qu'il explore un aspect jamais abordé à notre connaissance:

dès lors que les modèles sont réalisés, que les techniques d'optimisation existent pour utiliser ces modèles en vue de dimensionner, il définit une voie pour mettre en oeuvre facilement, voire automatiquement, ces techniques, ce qui sans cela exige du temps de programmation, des compétences informatiques, et des compétences dans les techniques d'optimisation.

Conclusions du chapitre:

Ce chapitre nous a permis de montrer que la disponibilité d'outils de CAO offrant une réelle aide dans le processus de conception se heurte, avec les approches proposées à ce jour, à de nombreuses difficultés dont:

- des temps de mise en oeuvre importants,
- des problèmes de maintenance et d'évolutivité,
- des temps de réponse longs,
- ou l'incapacité à prendre en charge le dimensionnement complet d'une machine.

C'est pourquoi nous avons imaginé une nouvelle approche qui permet de surmonter ces obstacles tout en offrant des capacités de dimensionnement et d'optimisation automatiques.

Celle-ci propose de:

- résoudre le problème de dimensionnement sous forme d'un problème d'optimisation sous contraintes,
- de générer automatiquement le logiciel de conception en utilisant des techniques de calcul symbolique et de programmation automatique,
- d'utiliser comme connaissance de base un modèle analytique de la machine à concevoir.

L'objet du prochain chapitre est d'exposer en détail tous les mécanismes et aspects théoriques de cette nouvelle approche que nous proposons.

- CHAPITRE II -

**PASCOSMA: une nouvelle approche
du Dimensionnement Assisté par Ordinateur sous contraintes de
machines électriques**

II. PASCOSMA: une nouvelle approche du Dimensionnement Assisté par Ordinateur sous contraintes de machines électriques

Introduction

L'objectif de ce chapitre est de formaliser tous les aspects théoriques de la nouvelle approche que nous proposons pour réaliser le dimensionnement par ordinateur et sous contraintes de machines électriques.

Cette approche sera par la suite désignée sous le terme de PASCOSMA qui est le sigle de: **P**rogramme d'**A**nalyse, de **S**ynthèse, de **C**onception et d'**O**ptimisation de **S**ystèmes **M**odélisables **A**nytiquement.

Rappelons brièvement, à propos de cette approche introduite dans le chapitre précédent, les éléments suivants:

- * Elle résout le problème de dimensionnement sous contraintes sous la forme du problème d'optimisation contraint P1:

$$P1 \Rightarrow \left[\begin{array}{ll} \min_p fob(p) & \\ \text{avec } g_i(p) \leq 0 & i=1,l \\ & g_i(p) = 0 \quad i=l+1,m \\ p_{jmin} \leq p_j \leq p_{jmax} & j=1,k \end{array} \right.$$

- * Elle assure la génération automatique des éléments logiciels nécessaires pour résoudre le problème de dimensionnement mis sous cette forme et ceci à partir d'un modèle analytique de la machine à dimensionner.
- * Elle utilise pour cela les modèles analytiques des machines à concevoir comme connaissance de base.

Afin d'en formaliser tous les aspects théoriques que nous avons développés, nous allons procéder en quatre étapes:

- 1° Nous allons commencer par mener une discussion devant nous *permettre de choisir convenablement le type d'algorithme d'optimisation sous contraintes* que nous utiliserons.
- 2° Puis, nous détaillerons la méthodologie et les mécanismes que nous avons créés afin d'assurer:

- d'une part la *génération automatique du programme de dimensionnement*,
- d'autre part *un emploi souple et convivial du logiciel dès lors qu'il est généré.*

3° Nous aborderons ensuite *une description du fonctionnement du logiciel de dimensionnement généré.* Dans ce cadre, on s'intéressera aux points suivants:

- le cycle grâce auquel le logiciel explore l'espace des solutions,
- les possibilités de traitement des cahiers des charges contraints que ce logiciel offre,
- les limites du fonctionnement.

4° Enfin, dans une dernière partie, *nous prendrons du recul sur la manière dont les modèles analytiques sont pris en charge par notre approche en redétaillant certains aspects.*

II.1. Choix de l'algorithme d'optimisation au coeur du processus de dimensionnement

Notre approche utilise dans son fonctionnement les algorithmes d'optimisation sous contraintes comme mécanisme d'exploration de l'espace des solutions.

C'est pourquoi nous allons faire un passage en revue succinct de ces techniques pour indiquer celles que nous emploierons de manière privilégiée.

Précisons que notre ambition n'est pas d'améliorer ces méthodes d'optimisation, ce type de travaux ayant déjà été entrepris par de nombreux chercheurs [56], [34], [33], [65].

Notre ambition n'est pas non plus de faire une analyse exhaustive de ces techniques, que l'on pourra d'ailleurs trouver dans les mêmes références.

L'approche que nous adoptons délibérément est d'utiliser ces techniques sous forme d'algorithmes "prêts à l'emploi" et disponibles dans des bibliothèques d'algorithmes d'analyse numérique telles que OSL [44] ou HARWELL [29]. Nous les considérons donc comme des "boîtes noires":

- dont on connaît les caractéristiques principales que nous allons rappeler,
- auxquelles il faut fournir un certain nombre d'informations en entrée,
- qui nous fournissent en retour un d'autre type d'informations.

Grâce à un processus itératif, l'ensemble doit aboutir à une solution optimale au problème de dimensionnement sous contraintes.

Le problème que nous avons, dès lors, à résoudre est de trouver la boîte noire ayant les caractéristiques les plus intéressantes dans le cadre de notre approche.

Ces algorithmes d'optimisation sous contraintes se scindent en deux grandes catégories [33]:

1° *Les méthodes de type direct* qui n'utilisent que la fonction objectif et les valeurs des fonctions soumises à des contraintes sans avoir besoin de leurs dérivées partielles.

Ces méthodes se scindent elles-mêmes en trois catégories:

* *Les méthodes stochastiques*, basées sur des mécanismes d'exploration aléatoire de l'espace des solutions. Des exemples de telles méthodes sont:

- les algorithmes génétiques fondés sur les lois de l'évolution naturelle [27], [65], [64], [52],
- les algorithmes de type recuit simulé utilisant les lois de la thermodynamique du recuit d'un solide [65].

* *Les méthodes de type heuristique* qui font des hypothèses sur les nouvelles directions de recherche à partir des expériences des points déjà explorés. Des exemples de méthodes de ce type sont la méthode de Hooke et Jeeves [33], [51] ou la méthode de Rosenbrock [33], [51].

* *Les méthodes à base théorique* qui offrent un processus d'exploration basé sur des propriétés mathématiques que l'espace des solutions devrait vérifier. On citera la méthode des directions conjuguées de Powell [33] qui suppose que l'espace des solutions a une forme quadratique autour de l'optimum.

2° *Les méthodes de type gradient* qui cherchent la solution du problème d'optimisation sous contraintes en se basant sur la connaissance d'une direction de recherche, trouvée en utilisant l'information fournie par les dérivées partielles de la fonction objectif et des fonctions soumises à des contraintes.

Concernant *les méthodes de type gradient*, il en existe plusieurs versions. On citera la méthode de Cauchy (*encore appelée méthode de la plus grande pente*), la méthode de Newton, la méthode du gradient conjugué, ou encore la méthode de type quasi Newton [33].

A titre indicatif, on précisera encore que toutes les méthodes que nous venons de citer sont des algorithmes d'optimisation sans contrainte. Des mécanismes de prise en compte de ces contraintes existent cependant et sont intégrés lorsque ces algorithmes sont fournis sous forme de boîtes noires dans les bibliothèques d'analyse numérique. Il s'agit de méthodes de transformation telles que les méthodes des fonctions de pénalité [33], [65]: elles permettent de transformer un problème avec contraintes en un problème, ou une séquence de problèmes sans contrainte, qui deviennent dès lors solubles par les techniques précédemment citées. Ces

méthodes sont considérées comme intégrées dans les algorithmes disponibles sous forme de boîtes noires.

Dès lors que les différents types d'algorithmes ont été passés en revue, nous pouvons indiquer ceux dont nous préconisons l'emploi dans le cadre de notre approche.

Notre choix va vers les algorithmes de type gradient pour les trois raisons suivantes:

1° Comme ils utilisent la connaissance des dérivées partielles pour trouver une direction de recherche, *ils doivent théoriquement se montrer plus performants que les algorithmes directs en terme de rapidité et de précision de convergence.* Ceci est confirmé par de nombreux auteurs [13], [43], [33], [31].

2° *Ils emploient des critères rigoureux d'optimalité (= critères permettant de savoir si on a atteint l'optimum),* dont les bases théoriques s'appuient sur des développements en série de Taylor des fonctions que l'on étudie. Cet emploi est possible grâce à la disponibilité des dérivées partielles [33], [31].

3° *La méthodologie de génération du logiciel de dimensionnement PASCOSMA, que nous allons présenter par la suite, permet d'obtenir les expressions des dérivées partielles sous forme symbolique exacte. Ceci va nous permettre de profiter de toute la puissance des algorithmes du type gradient car,*

s'il est vrai que ces derniers sont plus performants que les algorithmes de type direct, ceci n'est vrai que si l'on dispose d'un moyen d'évaluation précis des dérivées partielles [13], [43].

Malheureusement, l'une des grandes difficultés en électrotechnique est que l'évaluation exacte des expressions symboliques des dérivées partielles est impossible à réaliser manuellement, tant les équations sont la plupart du temps nombreuses et complexes [51].

C'est pourquoi tous les travaux de ce domaine dont nous avons trouvé les références et qui utilisent les algorithmes d'optimisation couplés avec des modèles de type analytique évaluent numériquement ces dérivées partielles de manière approchée en effectuant par exemple un calcul du type:

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x)}{h} \quad (1)$$

avec h une valeur suffisamment petite à déterminer.

Cette démarche amène souvent à une mauvaise approximation de la valeur de la dérivée du fait d'un mauvais choix du pas h [58]. Par ailleurs, il ne semble pas exister de critère ou d'algorithme en mesure de fournir la bonne valeur de h.

Ceci peut apparemment avoir des conséquences dramatiques comme tend à le montrer l'étude menée dans [51]:

celle-ci compare les performances de différents type d'algorithmes d'optimisation en vue de dimensionner et d'optimiser des machines asynchrones. Or, les auteurs y constatent des performances meilleures de la part des algorithmes de type direct par rapport aux algorithmes de type gradient. Pour eux, ces résultats, à priori paradoxaux, s'expliquent par l'obligation de calculer numériquement les dérivées partielles, ce qui entraînait inévitablement des erreurs contribuant à tromper l'algorithme.

II.2. Méthodologie de génération automatique et d'utilisation d'un logiciel de CAO proposée par la nouvelle approche

Le premier objectif de l'approche PASCOSMA que nous proposons est d'assurer la génération *automatique d'un logiciel de dimensionnement convivial* à partir d'un modèle analytique de la machine à concevoir. Nous allons ici décrire la méthodologie qui permet d'atteindre cet objectif. Celle-ci se scinde en deux niveaux:

1^{er} niveau: le cadre méthodologique qui assure la génération du logiciel de dimensionnement,

2^{ème} niveau: le cadre méthodologique qui met en place les mécanismes génériques assurant une utilisation conviviale et souple du logiciel généré.

II.2.a. 1^{er} niveau de la nouvelle approche: cadre méthodologique permettant d'assurer la génération automatique du logiciel de conception

Le principe général selon lequel nous obtenons automatiquement le logiciel de dimensionnement à partir du modèle analytique peut se formuler ainsi:

nous générons, en utilisant les équations du modèle analytique, un programme d'analyse et un programme de calcul de sensibilité. Nous lions ensuite ces deux programmes avec un algorithme d'optimisation sous contraintes, le tout constituant le logiciel de dimensionnement auquel nous voulons parvenir.

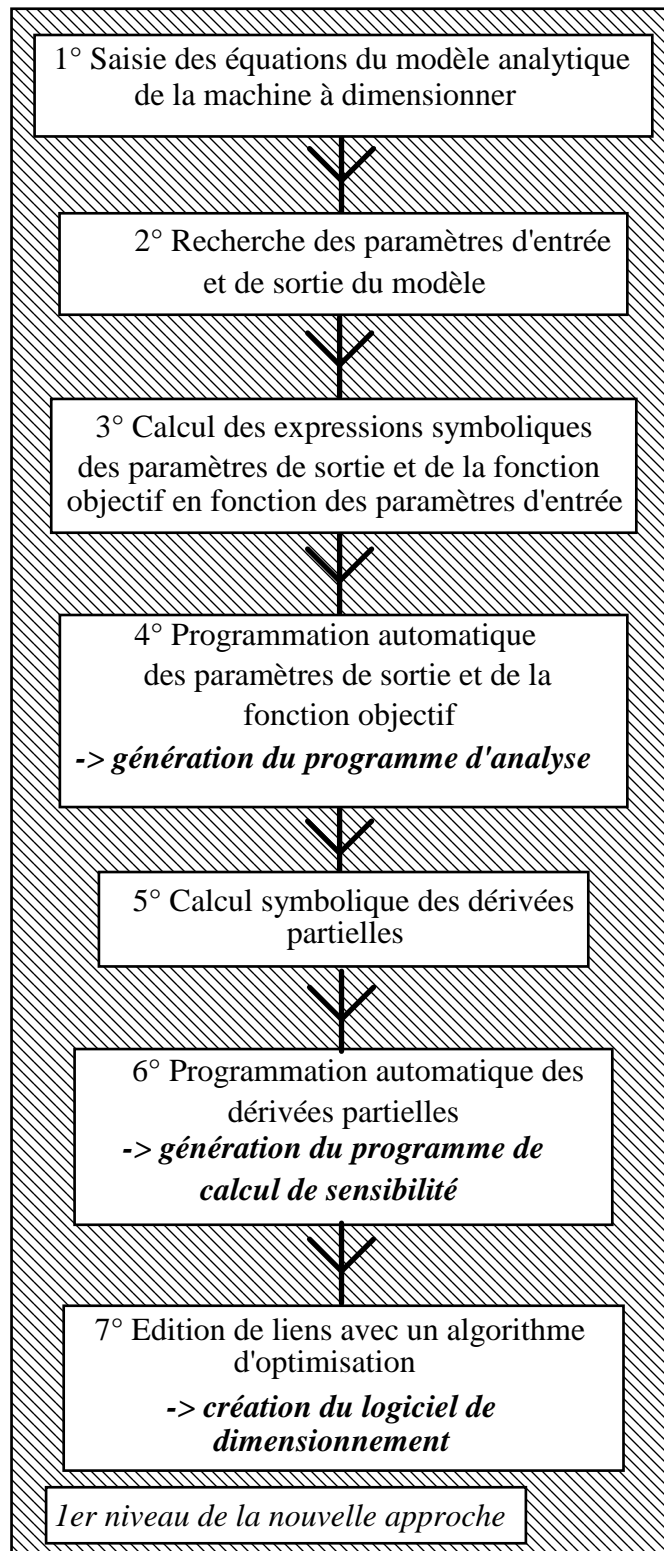


Figure 1: les sept étapes de la méthodologie de génération automatique du logiciel de dimensionnement = 1^{er} niveau de la nouvelle approche

La méthodologie que nous avons mise au point et qui permet de mettre en oeuvre ce principe se compose de sept étapes (*cf. figure 1*) que nous allons détailler dans les paragraphes qui vont suivre.

II.2.a.1. Saisie des équations du modèle analytique de la machine à dimensionner

Notre démarche utilise comme connaissance de base les équations du modèle analytique de la machine à dimensionner. Ces équations doivent donc être fournies. On notera que cette étape est la seule dans laquelle l'utilisateur de notre méthodologie intervient, les autres se déroulant de manière automatique.

Les équations à saisir doivent permettre de relier explicitement, les unes aux autres, les grandeurs physiques intervenant dans la machine. Ces grandeurs sont les paramètres de construction (*longueur, diamètre, ...*), les paramètres d'utilisation (*tension d'alimentation, courant fourni, ...*) ainsi les contraintes et les performances physiques qui en résultent (*échauffement, puissance fournie,...*).

L'ensemble de ces grandeurs sera désigné par la suite sous le terme générique de variables.

Les équations doivent être données sous la forme générique suivante:

$$1 \text{ variable} = 1 \text{ expression dépendant d'autres variables}^1 \quad (2)$$

exemple: $bcn = \frac{\text{fluair}}{hn^2}$

L'ensemble des équations constitue le modèle analytique de la machine à dimensionner.

La figure 4 donne un exemple de modèle permettant de dimensionner un électroaimant. Toutes les explications relatives à ce modèle et aux équations qu'il contient sont données dans l'annexe A.

En se contentant de cette connaissance saisie sous la forme déclarative précédemment indiquée, notre approche atteint son objectif d'être accessible à tout électrotechnicien.

II.2.a.2. Recherche des paramètres d'entrée et de sortie du modèle

Dans cette phase, nous préconisons le classement de l'ensemble des variables du modèle dans deux groupes distincts que nous appelons:

les paramètres d'entrée

les paramètres de sortie

Nous réalisons ce classement grâce à la procédure générique et automatisable suivante:

¹: Il est interdit de rentrer une variable dépendant d'elle-même, car sinon on rentre dans le cadre de la gestion de variables implicites que nous détaillons dans le paragraphe II.4.c.

$\text{lind} := 2 \cdot c + b$	$h := d + \text{ent} + a + \text{hn}$
$\text{atair} := 2 \cdot \frac{\text{bair}}{4 \cdot \pi \cdot 10^{-7}} \cdot \text{ent}$	$\text{bcm} := \frac{\text{fluair}}{\text{hn} \cdot d}$
$\text{fluair} := \text{bair} \cdot c \cdot \text{hn}$	$\text{atcm} := \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcm}^2} + \text{kc} \right) \cdot \text{bcm} \cdot (b + c)$
$\text{atcf} := \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcf}^2} + \text{kc} \right) \cdot \text{bcf} \cdot (2 \cdot a + \text{hn})$	$\text{atcn} := \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcn}^2} + \text{kc} \right) \cdot \text{bcn} \cdot (\text{lind} - c)$
$\text{attot} := \text{atair} + \text{atcm} + \text{atcf} + \text{atcn}$	$m := (d \cdot \text{lind} + 2 \cdot a \cdot c + \text{lind} \cdot \text{hn}) \cdot \text{hn} \cdot 7800 + 0.003$
$i := \frac{\text{attot}}{10}$	$f := \frac{\text{bair}^2}{4 \cdot \pi \cdot 10^{-7} \cdot 2} \cdot c \cdot \text{hn} \cdot 2$
$\text{bcf} := \text{bair}$	$\text{bcn} := \frac{\text{fluair}}{\text{hn}^2}$
<p><u>Constantes du problème:</u></p>	
$\text{ka} := 76.1$	$\text{kb} := 1.26$
	$\text{kc} := 129.5$

Figure 4: équations du modèle de dimensionnement d'un électroaimant

* On prend les équations du modèle telles qu'elles sont données dans la phase 1, c'est-à-dire sous la forme générique donnée par (2).

* On met dans l'ensemble des paramètres d'entrée les variables du modèle qui, dans les équations, apparaissent au moins une fois dans le terme de droite sans jamais apparaître dans le terme de gauche.

Cela définit l'ensemble des variables auxquelles aucune équation n'a été associée, donc des variables qui ne sont pas substituables par une expression dépendant d'autres variables.

On notera que l'ensemble des paramètres d'entrée ainsi déterminé est entièrement dépendant de la manière dont les équations ont été saisies.

* L'ensemble des paramètres de sortie du modèle sera constitué par toutes les autres variables du modèle.

Ainsi, si on applique cette démarche au modèle de la figure 4, l'ensemble Ω_e des paramètres d'entrée du modèle et l'ensemble Ω_s des paramètres de sortie du modèle seront respectivement égaux à:

$$\Omega_e = \{a, b, \text{bair}, c, d, \text{ent}, \text{hn}\}$$

$$\Omega_s = \{\text{atair}, \text{atcf}, \text{atcm}, \text{atcn}, \text{attot}, \text{bcf}, \text{bcm}, \text{bcn}, f, \text{fluair}, h, i, \text{ka}, \text{kb}, \text{kc}, \text{lind}, m\}$$

Le classement des variables du modèle ainsi réalisé vérifie les deux propriétés fondamentales que notre démarche va exploiter par la suite:

1° Tous les paramètres de sortie sont exprimables symboliquement en fonction des paramètres d'entrée par des substitutions symboliques.

2° Dès lors qu'une valeur est affectée à chacun des paramètres d'entrée, la valeur de tout paramètre de sortie est calculable.

II.2.a.3. Calcul symbolique et automatique des paramètres de sortie en fonction des paramètres d'entrée

Dans cette étape, nous calculons symboliquement les paramètres de sortie du modèle en fonction de ses seuls paramètres d'entrée.

Si on désigne par:

$C_i, i=1,n$ l'ensemble des paramètres de sortie du modèle

$P_j, j=1,m$ l'ensemble des paramètres d'entrée du modèle,

l'expression (2) peut aussi s'écrire sous la forme:

$$C_i = f_i(P_1, P_2, P_3, \dots, P_m, C_1, C_2, \dots, C_{i-1}, C_{i+1}, \dots, C_n)^2 \quad (3)$$

Le principe générique et automatisable que nous mettons en oeuvre dans cette étape consiste à substituer, grâce à un algorithme récursif, les expressions des $C_k, k \in [1,n]$ et $k \neq i$, jusqu'à ce que C_i ait une expression symbolique qui ne s'exprime plus qu'en fonction des $P_j, j=1,m$.

Exemple:

Prenons dans le modèle de la figure 4 l'exemple du paramètre de sortie i qui désigne le courant à fournir à la bobine de électroaimant.

Ce paramètre de sortie est le terme gauche de l'équation suivante:

$$i = \frac{\text{attot}}{10}$$

Dans cette expression, le paramètre de sortie dont elle dépend, à savoir attot , est substitué en

utilisant l'équation $\text{attot} = \text{atair} + \text{atcm} + \text{atcf} + \text{atcn}$. Ceci permet d'aboutir à:

$$i = \frac{\text{atair} + \text{atcm} + \text{atcf} + \text{atcn}}{10}$$

Cette nouvelle expression dépend encore des paramètres de sortie (*cf. atair, atcm, atcf, atcn*) qui sont à leur tour substitués ...

Ce processus récursif est poursuivi jusqu'à ce que l'expression de i ne dépende plus que des seuls paramètres d'entrée du modèle.

²: Il est interdit de rentrer une variable dépendant d'elle-même, d'où C_i ne s'exprime pas en fonction de C_i , car sinon on rentre dans le cadre de la gestion de variables implicites que nous détaillons dans le paragraphe II.4.c.

En l'occurrence, on aboutit pour i à l'expression suivante:

$$i = \frac{\frac{\text{bair} \times c \times (c + b) \times (76.1 \times e^{\frac{1.26 \times \text{bair}^2 \times c^2}{\text{hn}^2}} + 129.5)}{\text{hn}} + \text{bair} \times (76.1 \times e^{1.26 \times \text{bair}^2} + 129.5) \times (\text{hn} + 2 \times a)}{10} + \frac{\frac{5000000 \times \text{bair} \times \text{ent}}{\pi} + \frac{\text{bair} \times c \times (c + b) \times (76.1 \times e^{\frac{1.26 \times \text{bair}^2 \times c^2}{d^2}} + 129.5)}{d}}{10}$$

II.2.a.4. Programmation automatique des expressions des paramètres de sortie en fonction des paramètres d'entrée: génération du programme d'analyse

Dans cette étape, nous programmons dans un langage informatique les expressions symboliques des paramètres de sortie calculés précédemment ce qui est opération parfaitement automatisable.

Exemple:

L'expression de i calculée précédemment sera traduite, en langage FORTRAN [7], sous la forme suivante:

```
DOUBLE PRECISION U10,U11,U12
U10=C+B
U11=BAIR**2
U12=C**2
I=((BAIR*C*U10*(76.1*EXP(1.26*U11*U12*1.0d0/HN**2))+129.5)/HN+BAIR
.*(76.1*EXP(1.26*U11)+129.5)*(HN+2*A)+(5000000*BAIR*ENT)/3.14159+(
.BAIR*C*U10*(76.1*EXP(1.26*U11*U12*1.0d0/D**2)+129.5))/D)/10.0
```

On génère ainsi automatiquement les lignes de programme du *logiciel d'analyse* qui est le premier élément nécessaire à l'obtention du logiciel de dimensionnement automatique final.

II.2.a.5. Calcul symbolique et automatique des dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée

Dans cette étape, nous calculons symboliquement et automatiquement les dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée.

Ceci signifie qu'on détermine les $n \times m$ expressions symboliques suivantes:

$$\frac{\partial C_i}{\partial P_j} = \frac{\partial F_i(P_1, P_2, P_3, \dots, P_m)}{\partial P_j} \text{ pour } i=1, n \text{ et } j=1, m$$

Cette opération, complexe et longue, voire impossible à réaliser "à la main" à partir d'un certain niveau de complexité, est en revanche tout à fait automatisable. En effet, pour dériver n'importe quelle expression symbolique il suffit d'utiliser, autant de fois que cela est nécessaire:

- les lois de dérivation de toutes les fonctions élémentaires, à savoir:
 $\{+, -, *, /, \text{sqrt}, \text{exp}, \text{log}, \text{log}10, \text{sin}, \text{cos}, \text{tan}, \text{asin}, \text{acos}, \text{atan}, \text{sinh}, \text{cosh}, \text{tanh}\}$
- ainsi que la loi de dérivation d'une fonction composée.

Exemple:

Si l'on reprend l'exemple du paramètre de sortie *i* issu du modèle de la figure 4, le travail à réaliser dans cette étape consistera à calculer symboliquement toutes ses dérivées partielles par rapport aux paramètres d'entrée, soit l'ensemble des expressions suivantes:

$$\frac{\partial i}{\partial a}, \frac{\partial i}{\partial b}, \frac{\partial i}{\partial \text{bair}}, \frac{\partial i}{\partial c}, \frac{\partial i}{\partial d}, \frac{\partial i}{\partial \text{ent}}, \frac{\partial i}{\partial \text{hn}}$$

A titre d'exemple, l'environnement de manipulation symbolique MACSYMA [39] nous permet d'obtenir instantanément l'expression de $\frac{\partial i}{\partial \text{hn}}$ suivante:

$$\frac{\partial i}{\partial \text{hn}} = - \frac{\text{bair} \times c \times (c + b) \times (76.1 \times e^{\frac{1.26 \times \text{bair}^2 \times c^2}{\text{hn}^2}} + 129.5)}{10 \times \text{hn}^2} + \frac{191.772 \times \text{bair}^3 \times c^3 \times (c + b) \times e^{\frac{1.26 \times \text{bair}^2 \times c^2}{\text{hn}^2}}}{10 \times \text{hn}^4} + \frac{\text{bair} \times (76.1 \times e^{\frac{1.26 \times \text{bair}^2}{10}} + 129.5)}{10}$$

II.2.a.6. Programmation automatique des dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée: génération du programme de calcul de sensibilité

Dans cette étape, nous programmons les expressions symboliques des dérivées partielles des paramètres de sortie calculées précédemment dans un langage informatique.

Les mécanismes mis en oeuvre dans cette étape seront les mêmes que ceux qui ont été nécessaires à la génération du logiciel d'analyse (*voir paragraphe II.2.a.4.*).

On génère ainsi automatiquement les lignes de programme du *logiciel de calcul de sensibilité* qui est le second élément nécessaire à l'obtention du logiciel de dimensionnement automatique final.

II.2.a.7. Couplage du programme d'analyse et du programme de calcul de sensibilité avec un algorithme d'optimisation sous contraintes: création du logiciel de dimensionnement automatique

Dans cette dernière étape, nous réalisons une édition de liens des lignes de code du programme d'analyse et du programme de calcul de sensibilité avec l'algorithme d'optimisation sous contraintes.

On obtient ainsi automatiquement *le logiciel de dimensionnement automatique sous contraintes* comportant:

- la connaissance relative à la machine à dimensionner, matérialisée par le programme d'analyse et le programme de calcul de sensibilité,
- le processus d'exploration de l'espace des solutions utilisant cette connaissance représenté par l'algorithme d'optimisation sous contraintes.

II.2.b. 2^{ème} niveau de la nouvelle approche: cadre méthodologique permettant une utilisation souple et conviviale du logiciel de conception généré

En plus de définir les mécanismes permettant d'assurer la génération automatique du logiciel de dimensionnement, notre approche formalise un second niveau. Celui-ci contient les mécanismes génériques à mettre en place afin d'assurer une utilisation souple et conviviale du logiciel de dimensionnement généré.

Ce niveau ne doit pas être négligé, car le degré de souplesse et de convivialité avec lequel on arrivera à utiliser les applications de dimensionnement générées conditionne la viabilité et l'intérêt de ces applications. Ceci est d'autant plus vrai si l'on veut atteindre l'objectif d'utilisation du logiciel de dimensionnement par un électrotechnicien non informaticien et non spécialiste des techniques d'optimisation.

Notre approche définit en fait quels sont exactement les besoins que l'utilisateur électrotechnicien peut avoir en terme de souplesse et de convivialité . Puis, elle précise les solutions génériques à mettre en oeuvre.

Les besoins et les solutions formalisées concernent trois aspects:

- * *une utilisation souple,*
- * *une utilisation transparente des techniques d'optimisation,*
- * *une utilisation conviviale.*

Nous allons, dans les paragraphes qui vont suivre, formaliser précisément ces aspects et les solutions génériques que notre approche préconise.

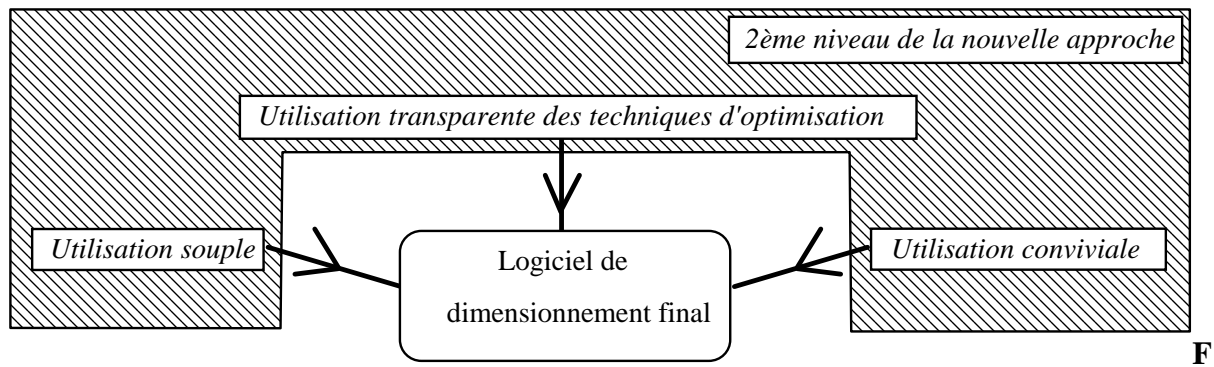


Figure 2: cadre méthodologique à définir et à mettre en oeuvre devant permettre une utilisation souple et conviviale du logiciel de conception généré = 2^{ème} niveau de la nouvelle approche

II.2.b.1. Notion de souplesse d'utilisation du logiciel de dimensionnement et sa mise en oeuvre

Le logiciel de dimensionnement automatique généré doit permettre de prendre en compte:

- Des contraintes d'égalité: $A = \text{une valeur fixe}$
 - Des contraintes d'inégalité: $\text{une valeur min.} \leq A \leq \text{une valeur max.}$
- où A pourra être soit un paramètre de sortie soit un paramètre d'entrée du modèle.

Ces contraintes représentent en effet celles que l'on peut poser dans un cahier des charges.

La souplesse du logiciel à mettre en place est relative à ces contraintes. Elle doit offrir à l'utilisateur les possibilités suivantes:

** La possibilité de modifier les limites des contraintes.*

Considérons à nouveau l'exemple du dimensionnement d'un contacteur (*voir annexe A*) dont les équations du modèle sont rappelées sur la figure 4.

Si l'utilisateur souhaite dimensionner ce contacteur en essayant deux valeurs différentes pour la hauteur maximale 'h' de l'appareil qui pourraient être 0,2 m puis 0,3 m, il doit pouvoir d'abord introduire la contrainte:

$$0,008 \text{ m} \leq h \leq 0,2 \text{ m}$$

puis éventuellement la contrainte:

$$0,008 \text{ m} \leq h \leq 0,3 \text{ m}$$

De même, s'il souhaite dimensionner son contacteur en essayant deux valeurs différentes pour l'entrefer 'ent' qui pourraient être 0,0005 m puis 0,001m, il doit pouvoir passer de la contrainte d'égalité:

ent=0,0005 m
à la contrainte d'égalité:
ent=0,001 m

Cette souplesse est mise en place en assurant la génération du logiciel de dimensionnement avec des limites de contraintes gérées non pas comme des constantes fixes, mais comme des variables, dont la valeur sera fournie par l'utilisateur.

**La possibilité de modifier la nature des contraintes posées sur les paramètres d'entrée et de sortie:*

L'utilisateur doit pouvoir imposer, à un paramètre d'entrée ou de sortie, d'être soumis à une contrainte d'égalité pour un dimensionnement donné puis de pouvoir être soumis à une contrainte d'inégalité pour un autre dimensionnement. Il doit aussi avoir la possibilité de ne pas imposer de contrainte.

Ainsi, dans l'exemple du contacteur, l'utilisateur peut dans un premier temps vouloir imposer la valeur de la hauteur de l'électroaimant en posant la contrainte suivante:

$h=0,2$ m

Selon les résultats qu'il obtiendra, ou parce qu'un autre cahier des charges lui est imposé, il peut décider ensuite d'explorer l'espace des solutions en laissant varier cette hauteur entre 0,2 m et 0,3 m et il voudra alors poser la contrainte suivante:

$0,2 \text{ m} \leq h \leq 0,3 \text{ m}$

Enfin, il peut aussi être confronté à un cahier des charges où cette hauteur n'est pas contrainte et, dans ce cas, la laisser évoluer sans restriction.

Pour mettre en place cette souplesse, il faut prévoir une saisie du statut de la contrainte que l'utilisateur voudra affecter au paramètre d'entrée ou de sortie. Puis, il faudra assurer la gestion de ce paramètre d'entrée ou de sortie en fonction de ce statut.

Les possibilités d'un tel fonctionnement doivent donc se traduire par des mécanismes de paramétrage des procédures du programme de dimensionnement en fonction du statut et des valeurs limites des contraintes que l'utilisateur voudra affecter à chacun des paramètres d'entrée ou de sortie.

Aussi, ce souci de paramétrage fera-t-il partie de tous les mécanismes mis en oeuvre dans notre approche et que nous allons détailler par la suite.

II.2.b.2. Notion d'utilisation transparente des algorithmes d'optimisation et sa mise en oeuvre

L'emploi des algorithmes d'optimisation sous contraintes que nous utilisons comme mécanisme d'exploration de l'espace des solutions, nécessite un certain nombre de manipulations et de précautions. Notre objectif est de définir les mécanismes génériques les prenant automatiquement en compte afin que l'utilisateur du logiciel de dimensionnement n'ait plus à s'en soucier. L'utilisation des techniques d'optimisation devient ainsi "transparente", autrement dit accessible à un non spécialiste.

Les mécanismes à mettre en oeuvre dans ce cadre sont au nombre de trois. Il s'agit respectivement:

- * d'une normalisation des valeurs des paramètres de sortie et de leurs dérivées partielles,*
- * d'une reformulation des contraintes entre leur formulation dans le cahier des charges et celle employée par l'algorithme d'optimisation,*
- * d'un tri des valeurs des contraintes et de leurs dérivées partielles en fonction du type de la contrainte qui leur est appliquée.*

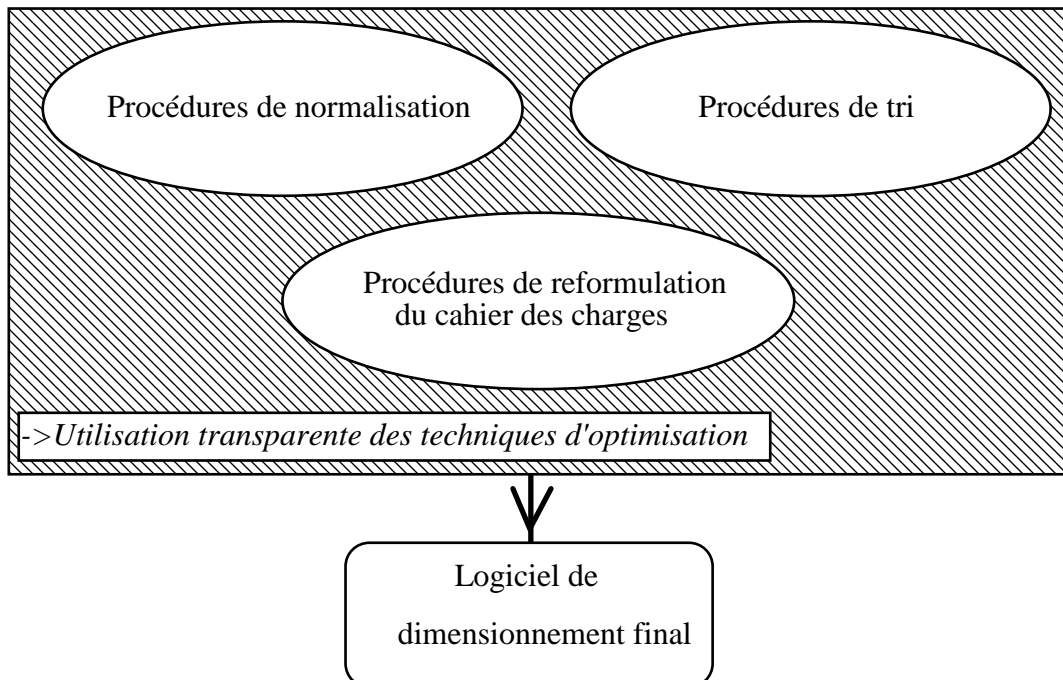


Figure 3: procédures à définir afin d'assurer une utilisation transparente des techniques d'optimisation

La normalisation des valeurs des paramètres de sortie et de leurs dérivées partielles:

Cette normalisation est conseillée voire nécessaire pour assurer un bon conditionnement numérique du problème tel qu'il sera fourni à l'algorithme d'optimisation [29], [33]. Il est en effet très défavorable que des valeurs de fonctions manipulées par lui aient des ordres de grandeurs très différents.

Or, dans une machine électrique, on peut manipuler des grandeurs variant dans des plages très différentes. Ainsi, pour une machine asynchrone:

- la puissance a un ordre de grandeur du kw (*1000 à 10000W par exemple*),
- l'entrefer a un ordre de grandeur du dixième de millimètre (*typiquement de 0,0005 à 0,001 m*).

Il en résulte qu'en unité S.I., on a un facteur de $\frac{1000}{0,001}$, soit un million entre ces deux grandeurs.

Pour améliorer le conditionnement numérique du problème, une solution efficace est de ramener toutes les grandeurs dans un espace réduit dans lequel celles-ci varient autour de 1.

Notre travail consiste alors à définir et à mettre en place les mécanismes qui vont permettre à l'électrotechnicien de manipuler les paramètres d'entrée et de sortie avec leurs valeurs réelles et physiques tout en laissant l'algorithme d'optimisation travailler dans l'espace réduit. D'où la nécessité de définir la transformation Tr définie sur la figure 5.

Le tableau 1 donne un exemple de transformation Tr que nous avons définie et qui permet de passer de l'espace réel à un espace réduit. On notera que cette transformation dépend du type de la contrainte (*contrainte d'égalité ou d'intervalle*) et des valeurs limites posées sur les paramètres d'entrée et de sortie. On prend ainsi en compte le souci de paramétrage des procédures en fonction des statuts et des valeurs limites des contraintes évoqué dans le paragraphe II.2.b.1.

On constate aussi que la variable du modèle qui sera utilisée comme fonction objectif fait l'objet d'un traitement particulier. En effet, la fonction objectif n'étant soumise à aucune contrainte, la normalisation est réalisée à partir de valeurs minimales et maximales estimées (*ces valeurs estimées seront fournies par un algorithme ou pourront même être demandées à l'utilisateur*).

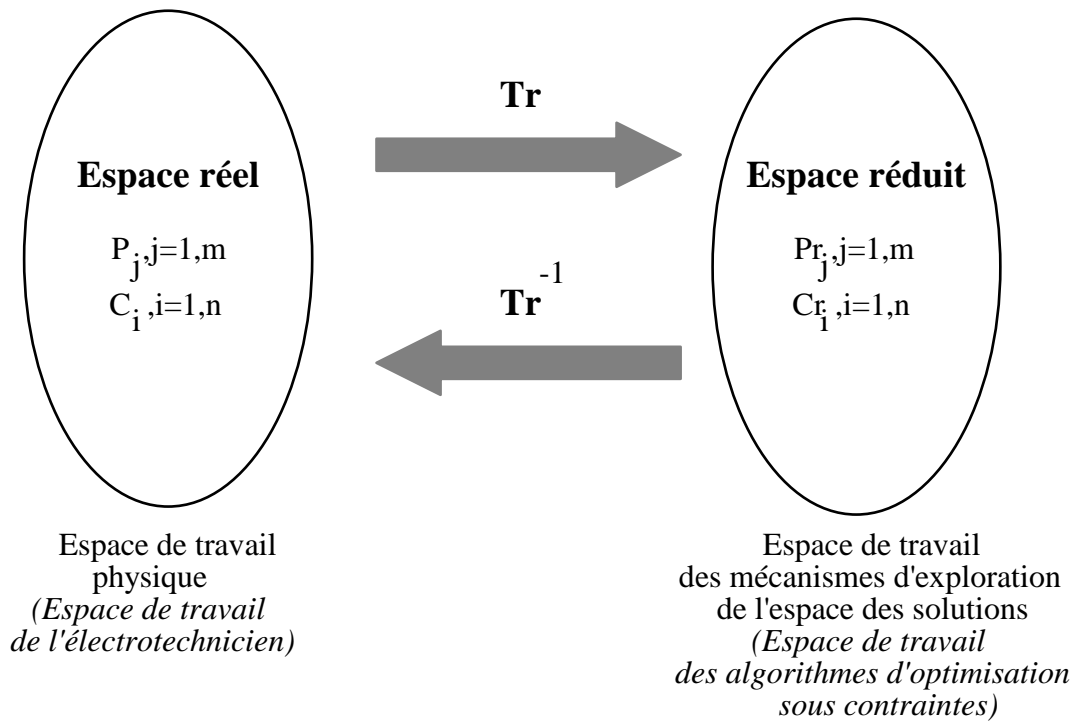


Figure 5: Processus de changement d'espace de travail

Type du paramètre	Type de la contrainte posée sur le paramètre dans le cahier des charges	Informations utilisées pour faire la normalisation	Tr Fonction de passage de l'espace réel à l'espace normalisé	Tr⁻¹ Fonction de passage de l'espace normalisé à l'espace réel
Paramètre d'entrée	Egalité: $P_j = P_{const_j}$	P_{const_j}	$Pr_j = Tr(P_j) = \frac{P_j}{P_{const_j}}$	$P_j = Tr^{-1}(Pr_j) = Pr_j \times P_{const_j}$
$P_j, j=1,m$	Intervalle: $P_{min_j} \leq P_j \leq P_{max_j}$	P_{min_j}, P_{max_j}	$Pr_j = Tr(P_j) = \frac{P_j - P_{min_j}}{P_{max_j} - P_{min_j}}$	$P_j = Tr^{-1}(Pr_j) = Pr_j \times (P_{max_j} - P_{min_j}) + P_{min_j}$
Paramètre de sortie	Egalité: $C_i = C_{const_i}$	C_{const_i}	$Cr_i = Tr(C_i) = \frac{C_i}{C_{const_i}}$	$C_i = Tr^{-1}(Cr_i) = Cr_i \times C_{const_i}$
$C_i, i=1,n$	Intervalle: $C_{min_i} \leq C_i \leq C_{max_i}$	C_{min_i}, C_{max_i}	$Cr_i = Tr(C_i) = \frac{C_i - C_{min_i}}{C_{max_i} - C_{min_i}}$	$C_i = Tr^{-1}(Cr_i) = Cr_i \times (C_{max_i} - C_{min_i}) + C_{min_i}$
Fonction objectif fob	<i>On n'impose pas de contrainte sur fob</i>	Fobmin, Fobmax <i>valeurs estimées</i>	$fobr = Tr(fob) = \frac{fob - fob_{min}}{fob_{max} - fob_{min}}$	$fob = Tr^{-1}(fobr) = fobr \times (fob_{max} - fob_{min}) + fob_{min}$

Tableau 1: exemple de fonction Tr et Tr⁻¹ de passage de l'espace réel à l'espace réduit en fonction du type du paramètre et du type de la contrainte appliquée sur ce paramètre

Ce tableau définit aussi la fonction réciproque Tr^{-1} permettant de passer de l'espace réduit à l'espace réel. Cette fonction est notamment utilisée lorsque qu'il faut ramener les solutions trouvées par l'algorithme d'optimisation de l'espace réduit à l'espace réel afin de les communiquer à l'utilisateur.

Nous devons aussi définir et assurer le passage de l'espace réel à l'espace réduit pour les dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée.

En effet, l'algorithme d'optimisation qui résout le problème dans l'espace réduit a besoin des dérivées partielles exprimées dans cet espace.

Les expressions des dérivées partielles, que nous avons calculées symboliquement et programmées automatiquement dans le 1^{er} niveau de notre méthodologie (*voir le paragraphe II.2.a.5.*) sont exprimées dans l'espace réel. Autrement dit, nous disposons des expressions:

$$\frac{\partial C_i}{\partial P_j} = \frac{\partial F_i(P_1, P_2, P_3, \dots, P_m)}{\partial P_j} \text{ pour } i=1, n \text{ et } j=1, m$$

Il s'agit de définir et d'assurer automatiquement le calcul de:

$$\frac{\partial Cr_i}{\partial Pr_j} = \text{Tr}\left(\frac{\partial C_i}{\partial P_j}\right) = \frac{\partial Fr_i(Pr_1, Pr_2, Pr_3, \dots, Pr_m)}{\partial Pr_j} \text{ pour } i=1, n \text{ et } j=1, m$$

La formule générique de calcul des $\frac{\partial Cr_i}{\partial Pr_j}$ en fonction des $\frac{\partial C_i}{\partial P_j}$ se trouve en appliquant le

raisonnement suivant:

Sachant que:

$$Cr_i = \text{Tr}(C_i)$$

on en déduit que:

$$\frac{\partial Cr_i}{\partial Pr_j} = \frac{\partial \text{Tr}(C_i)}{\partial C_i} \times \frac{\partial C_i}{\partial P_j}$$

or:

$$\frac{\partial C_i}{\partial Pr_j} = \frac{\partial C_i}{\partial P_j} \times \frac{\partial P_j}{\partial Pr_j}$$

comme:

$$P_j = \text{Tr}^{-1}(Pr_j)$$

il vient que:

$$\frac{\partial C_i}{\partial Pr_j} = \frac{\partial C_i}{\partial P_j} \times \frac{\partial \text{Tr}^{-1}(Pr_j)}{\partial Pr_j}$$

Soit enfin:

$$\frac{\partial Cr_i}{\partial Pr_j} = \text{Tr}\left(\frac{\partial C_i}{\partial P_j}\right) = \frac{\partial \text{Tr}(C_i)}{\partial C_i} \times \frac{\partial C_i}{\partial P_j} \times \frac{\partial \text{Tr}^{-1}(Pr_j)}{\partial Pr_j}$$

Contrainte sur C_i	Contrainte sur P_j	$\text{Tr}(C_i)$	$\frac{\partial \text{Tr}(C_i)}{\partial C_i}$	$\text{Tr}^{-1}(P_j)$	$\frac{\partial \text{Tr}^{-1}(P_j)}{\partial P_j}$	$\frac{\partial C_i}{\partial P_j} = \frac{\partial \text{Tr}(C_i)}{\partial C_i} \times \frac{\partial C_i}{\partial P_j} \times \frac{\partial \text{Tr}^{-1}(P_j)}{\partial P_j}$
Egalité	Egalité	$\frac{C_i}{C_{\text{const}_i}}$	$\frac{1}{C_{\text{const}_i}}$	$P_j \times P_{\text{const}_j}$	P_{const_j}	$\frac{1}{C_{\text{const}_i}} \times \frac{\partial C_i}{\partial P_j} \times P_{\text{const}_i}$
Intervalle	Egalité	$\frac{C_i}{C_{\text{max}_i} - C_{\text{min}_i}}$	$\frac{1}{C_{\text{max}_i} - C_{\text{min}_i}}$	$P_j \times P_{\text{const}_j}$	P_{const_j}	$\frac{1}{C_{\text{max}_i} - C_{\text{min}_i}} \times \frac{\partial C_i}{\partial P_j} \times P_{\text{const}_i}$
Intervalle	Intervalle	$\frac{C_i}{C_{\text{max}_i} - C_{\text{min}_i}}$	$\frac{1}{C_{\text{max}_i} - C_{\text{min}_i}}$	$P_j \times (P_{\text{max}_j} - P_{\text{min}_j})$	$P_{\text{max}_j} - P_{\text{min}_j}$	$\frac{1}{C_{\text{max}_i} - C_{\text{min}_i}} \times \frac{\partial C_i}{\partial P_j} \times (P_{\text{max}_j} - P_{\text{min}_j})$
Egalité	Intervalle	$\frac{C_i}{C_{\text{const}_i}}$	$\frac{1}{C_{\text{const}_i}}$	$P_j \times (P_{\text{max}_j} - P_{\text{min}_j})$	$P_{\text{max}_j} - P_{\text{min}_j}$	$\frac{1}{C_{\text{const}_i}} \times \frac{\partial C_i}{\partial P_j} \times (P_{\text{max}_j} - P_{\text{min}_j})$

Tableau 2: calcul des dérivées partielles des paramètres de sortie dans l'espace réduit

Nous utilisons ce résultat dans le tableau 2 pour calculer les expressions des dérivées partielles des paramètres de sortie dans l'espace réduit sur la base de la transformation Tr^3 proposée dans le tableau 1.

Ce tableau permet de voir que l'expression $\frac{\partial Cr_i}{\partial Pr_j}$ de ces dérivées partielles dépend du statut et des valeurs limites de la contrainte qui a été posée sur C_i et de celle sur P_j . On prend ainsi à nouveau en compte le souci de paramétrage des procédures en fonction des statuts et des valeurs limites des contraintes évoqué dans le paragraphe II.2.b.1.

La reformulation des contraintes entre leur formulation dans le cahier des charges et celle employée par l'algorithme d'optimisation:

La différence de formulation des contraintes entre le cahier des charges et l'algorithme d'optimisation résulte du fait que les algorithmes d'optimisation ne savent souvent résoudre qu'un seul type de contraintes, à savoir des contraintes positives " ≥ 0 ", ou des contraintes négatives " ≤ 0 ".

Une autre configuration classique est de trouver des algorithmes qui, en plus des contraintes de type " ≤ 0 ", ou de type " ≥ 0 ", gèrent des contraintes d'égalité, donc du type " $= 0$ ".

Or, dans un cahier des charges, on donne des contraintes d'égalité et des intervalles de variation. Ces contraintes peuvent fort heureusement toujours se ramener à un ensemble de contraintes de type exclusivement " ≤ 0 " ou de type exclusivement " ≥ 0 " ou encore de type " ≤ 0 " et " $= 0$ " ou enfin de type " ≥ 0 " et " $= 0$ ".

Notre travail consiste encore une fois à définir les procédures nécessaires (cf. tableau 3), puis à les assurer de manière transparente dans le logiciel qui sera généré.

Le tri des valeurs des paramètres de sortie et de leurs dérivées partielles en fonction du type de la contrainte qui leur est appliquée:

Nous devons assurer ce type de mécanisme lorsque l'algorithme d'optimisation gère à la fois des contraintes d'inégalité (de type " ≤ 0 " ou " ≥ 0 ") et des contraintes d'égalité (de type " $= 0$ "). Dans ce cas, cet algorithme peut demander qu'on lui fournisse, dans des vecteurs différents, les valeurs des contraintes d'égalité et les valeurs des contraintes d'inégalité. De même, il peut demander qu'on lui fournisse, dans des matrices différentes, les valeurs des dérivées partielles

³: **Remarque:** L'exemple de fonction Tr que nous avons développé assure essentiellement une normalisation de la valeur des paramètres de sortie. On notera que certains auteurs préconisent de normaliser aussi les valeurs des gradients de ces paramètres [33]. Cette opération peut être réalisée grâce à des algorithmes faisant de multiples itérations. On citera notamment la procédure de mise à l'échelle systématique proposée par Root et Ragsdell [48], [33].

des contraintes d'égalité et des contraintes d'inégalité par rapport aux paramètres d'entrée. Ceci explique donc qu'en fonction du type de contrainte que l'utilisateur a posé sur les paramètres de sortie, nous devons assurer un travail de tri dans le programme de dimensionnement que nous allons générer.

Type de contraintes gérées par l'algorithme d'optimisation	Type de contrainte du cahier des charges = Contraintes fournies par l'utilisateur) (sur les paramètres d'entrée ou sur les paramètres de sortie)	Reformulation de la contrainte du cahier des charges sous la forme de contrainte compréhensible par l'algorithme d'optimisation
$x \geq 0$	Egalité: $y = V_{\text{fixe}}$	$y - V_{\text{fixe}} \geq 0$ et $-y + V_{\text{fixe}} \geq 0$
	Intervalle: $V_{\text{min}} \leq y \leq V_{\text{max}}$	$V_{\text{max}} - y \geq 0$ et $-V_{\text{min}} + y \geq 0$
$x \leq 0$	Egalité: $y = V_{\text{fixe}}$	$-y + V_{\text{fixe}} \leq 0$ et $y - V_{\text{fixe}} \leq 0$
	Intervalle: $V_{\text{min}} \leq y \leq V_{\text{max}}$	$-V_{\text{max}} + y \leq 0$ et $V_{\text{min}} - y \leq 0$
$x \geq 0$ et $x = 0$	Egalité: $y = V_{\text{fixe}}$	$y - V_{\text{fixe}} = 0$
	Intervalle: $V_{\text{min}} \leq y \leq V_{\text{max}}$	$V_{\text{max}} - y \geq 0$ et $-V_{\text{min}} + y \geq 0$
$x \leq 0$ et $x = 0$	Egalité: $y = V_{\text{fixe}}$	$y - V_{\text{fixe}} = 0$
	Intervalle: $V_{\text{min}} \leq y \leq V_{\text{max}}$	$-V_{\text{max}} + y \leq 0$ et $V_{\text{min}} - y \leq 0$

Tableau 3: table de reformulation des contraintes d'égalité ou d'intervalle d'un cahier des charges

II.2.b.3. Notion d'utilisation conviviale du logiciel de dimensionnement et sa mise en oeuvre

Une des contraintes que nous nous fixons, afin que le logiciel de dimensionnement puisse être utilisé par tout électrotechnicien, est que celui-ci soit convivial à utiliser.

En pratique cela passera par la définition d'une interface d'utilisation de ce logiciel qui devra posséder les propriétés suivantes:

-L'utilisation de cette interface devra être simple et intuitive ce qui signifie notamment que l'utilisateur devra pouvoir accéder facilement aux fonctions dont il a besoin. Pour cela, nous préconisons l'emploi d'une interface de type graphique utilisant, entre autres, les concepts à présent vulgarisés de fenêtres, de boîtes de dialogues ou de menus déroulants.

-L'utilisateur devra disposer de toutes les fonctionnalités dont il peut avoir besoin. Ainsi, dans le cadre que nous étudions, les fonctionnalités les plus importantes à prévoir sont celles pour:

- faire du calcul d'analyse,
- faire du calcul de sensibilité,
- tracer l'évolution d'un paramètre de sortie en fonction d'un paramètre d'entrée,
- saisir et visualiser les contraintes du cahier des charges,
- lancer un dimensionnement automatique
- visualiser les solutions trouvées par le système.

-L'utilisateur devra enfin disposer de fonctionnalités d'aide.

Le problème est ensuite de définir et d'assurer les mécanismes afin de mettre en place ces propriétés sachant que, d'un modèle à l'autre, les paramètres d'entrée et de sortie ne seront pas les mêmes. Il en résulte que la difficulté importante que nous nous devons de surmonter est la suivante:

définir une structure et les moyens d'implantation d'interface graphique qui aient la capacité de s'adapter à toutes les configurations possibles.

Face à cette difficulté, nos réflexions nous ont amenés à considérer que nous pouvions mettre en oeuvre les deux solutions suivantes:

1° Nous générons les lignes d'instructions de l'interface graphique de manière automatique et spécifique au modèle analytique que nous utilisons de la même manière que le sont les lignes d'instructions du programme d'analyse ou du programme de calcul de sensibilité. Ces lignes d'instructions peuvent alors être liées aux autres lignes d'instruction du logiciel de dimensionnement (*cf. le programme de calcul d'analyse, de calcul de sensibilité et l'algorithme d'optimisation*) pour former un seul et même programme. La figure 6 schématise cette première approche.

2° Nous créons une interface graphique qui ont la capacité de s'autoconfigurer automatiquement en fonction des caractéristiques du logiciel de dimensionnement. L'interface et le logiciel de dimensionnement se présentent alors sous forme de deux programmes exécutables indépendants dont l'un (*l'interface graphique*) a pour vocation de piloter l'autre (*le programme de dimensionnement*).

Cette solution est plus élégante que la précédente car elle découple le problème de réalisation de l'interface graphique de celui du logiciel de dimensionnement. *C'est donc celle que nous privilégierons par la suite.*

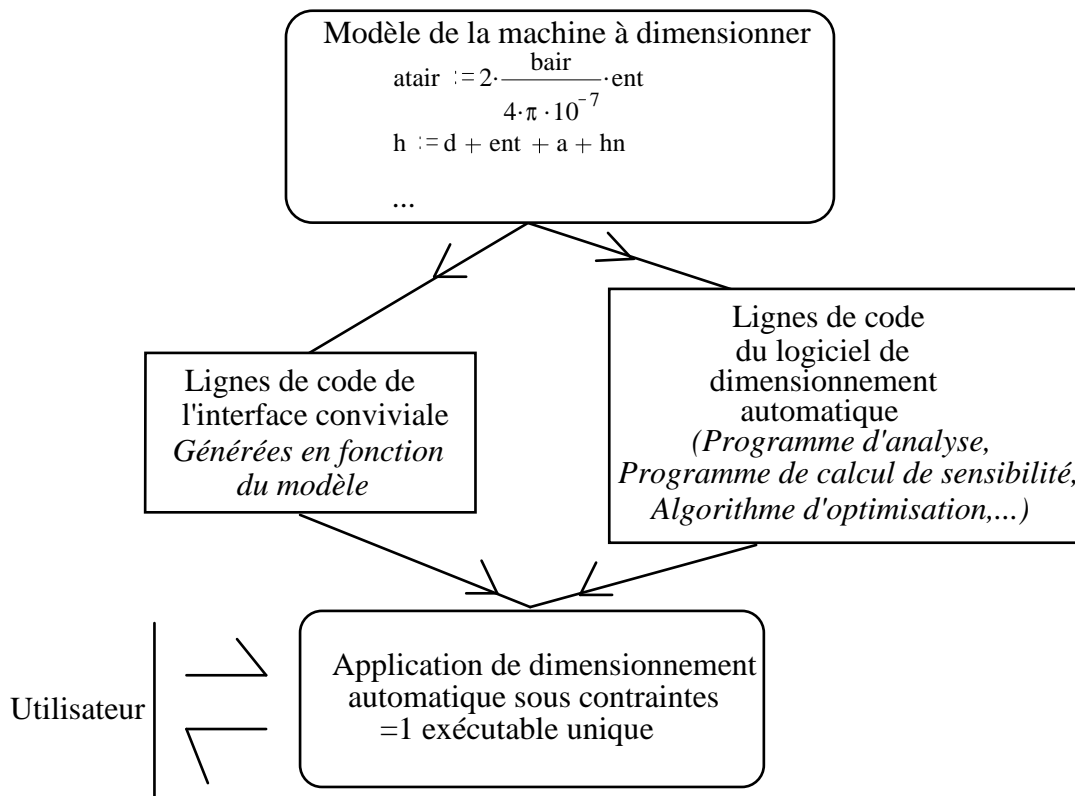


Figure 6: première approche en vue de disposer d'une interface conviviale pour utiliser le logiciel de dimensionnement automatique sous contraintes généré

Nous pouvons la mettre en place en utilisant:

- La *dynamicité* de certains langages de programmation de haut niveau, tels que C++ [17] ou lisp [35] qui permettent de définir des interfaces graphiques dans lesquelles la création dynamique (*c'est-à-dire en cours de fonctionnement du programme*) des objets qu'elles utilisent (*cf. les boutons, les fenêtres, les menus déroulants, les boîtes de dialogues,...*) est possible.

Ceci revient donc à paramétrer complètement le fonctionnement de l'interface graphique en fonction de toutes les configurations de données qu'elle devra manipuler. Ceci représente un travail complexe et long pour le développeur de cette interface graphique.

Nous devons aussi dans cette configuration,

définir un protocole de communication complet entre l'interface graphique et l'exécutable de dimensionnement permettant de gérer des échanges d'informations aussi complexes et divers que:

- des cahiers des charges = ensemble de contraintes sur les paramètres d'entrées et de sortie,
- des solutions au cahier des charges = évolutions des paramètres d'entrée et de sortie au cours du dimensionnement,
- des demandes de calcul d'analyse,

- des résultats de calcul d'analyse,
- des demandes de calcul de sensibilité,
- ...

Nous gérons ces échanges d'informations par l'intermédiaire de fichiers.

La figure 7 schématise cette seconde approche.

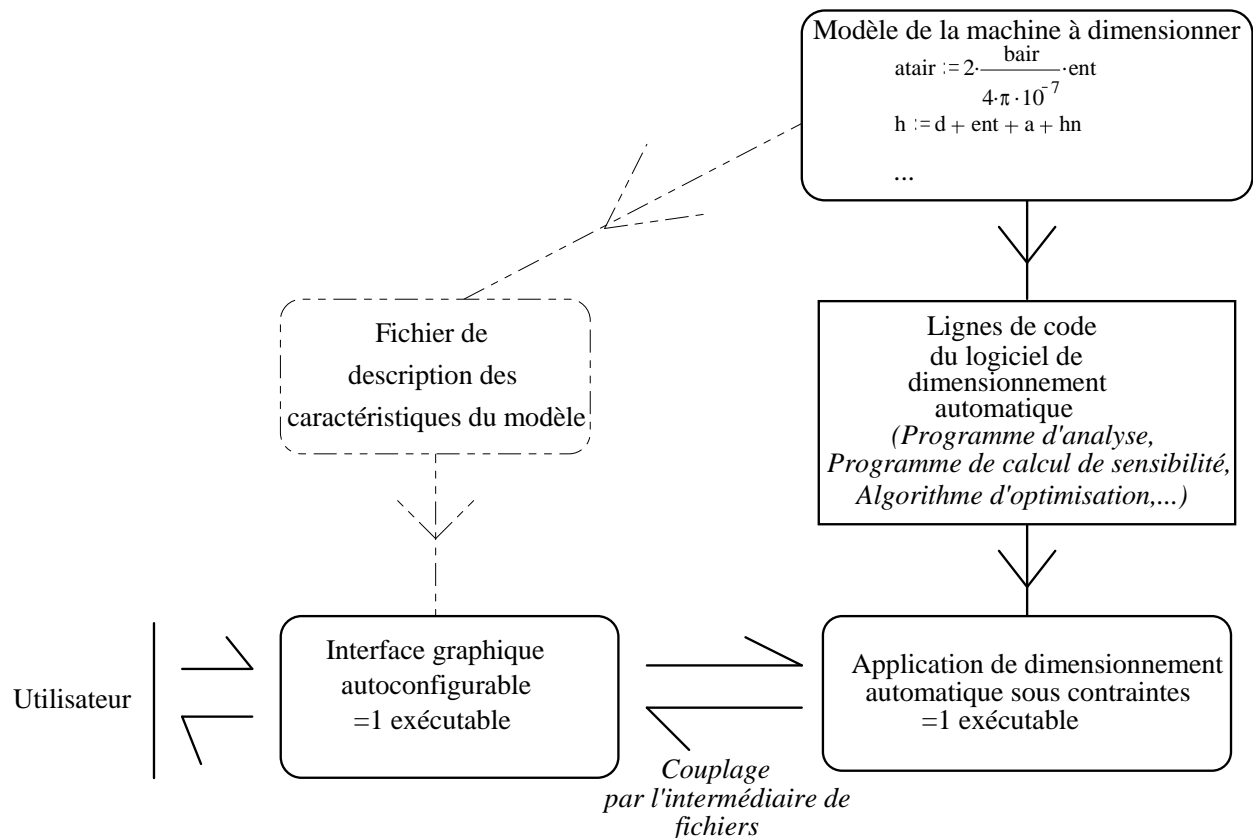


Figure 7: seconde approche en vue de disposer d'une interface conviviale pour utiliser le logiciel de dimensionnement automatique sous contraintes généré

II.3 Fonctionnement du logiciel de Dimensionnement Assisté par Ordinateur généré grâce à la nouvelle approche

Après avoir formalisé les mécanismes de génération automatique du logiciel de dimensionnement ainsi que les mécanismes qui vont permettre de rendre son utilisation souple et conviviale, nous allons à présent:

- * détailler le fonctionnement même de ce logiciel de dimensionnement,
- ainsi que deux aspects connexes à ce fonctionnement qui sont:

- * *la capacité du logiciel de dimensionnement généré à traiter toutes les configurations de cahiers des charges possibles,*
- * *et les limites de fonctionnement de ce logiciel.*

II.3.a Fonctionnement du logiciel de dimensionnement automatique généré: le cycle d'exploration de l'espace des solutions utilisé

Dès lors que nous avons généré l'exécutable de dimensionnement automatique, celui-ci nous permet de dimensionner des machines selon un mécanisme itératif, comportant sept étapes.

Nous avons représenté, sur la figure 8, l'enchaînement de ces étapes grâce à un organigramme très détaillé.

Cette figure met les points suivants en évidence:

- * Les informations à fournir avant de pouvoir lancer le cycle de dimensionnement sont:

1° *Un cahier des charges*, constitué des trois ensembles d'informations suivants:

- le type de la contrainte à appliquer sur chaque paramètre d'entrée et chaque paramètre de sortie du modèle.
- les valeurs limites pour chaque paramètre d'entrée ou de sortie, en fonction du type de la contrainte qui lui est appliqué (*une valeur finale à atteindre pour une contrainte de type égalité, une valeur minimale et une valeur maximale pour une contrainte de type inégalité*).
- une valeur prévisionnelle de la valeur minimale et maximale de la fonction objectif.

2° *Un point de départ* qui est en fait un ensemble de valeurs initiales pour tous les paramètres d'entrée du modèle. Nous reparlerons de cette notion de point de départ.

- * Dès lors que les informations nécessaires sont fournies, le processus de dimensionnement automatique peut s'enclencher. Celui-ci s'apparente à un cycle qui comprend les étapes suivantes:

- * Etape 0: *initialisation*.

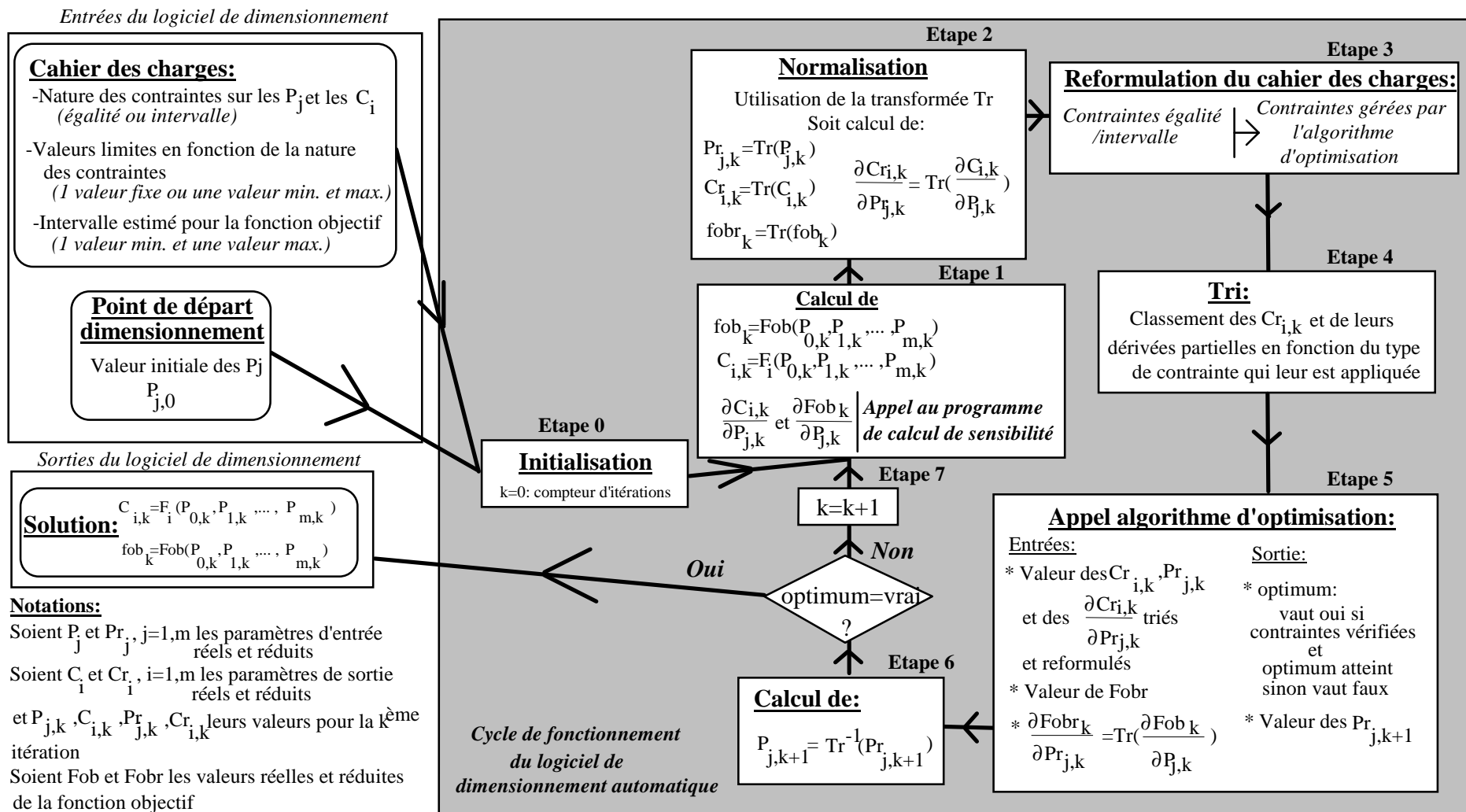


Figure 8: fonctionnement général du logiciel de dimensionnement généré par la méthodologie PASCOSMA

* Etape 1: *appel aux procédures de calcul de la fonction objectif (qui n'est qu'un paramètre de sortie particulier), des paramètres de sortie et de leurs dérivées partielles.* Les lignes de programme de calcul correspondantes sont celles générées lors du niveau 1 de la méthodologie de génération de code. Elles fournissent leurs résultats dans l'espace réel.

Nous faisons ensuite intervenir une suite d'étapes qui correspondent à l'emploi des procédures du niveau 2 de la méthodologie, qui assurent une reformulation des informations en vue de leur utilisation par l'algorithme d'optimisation:

* Etape 2: *normalisation* automatisée des valeurs des paramètres de sortie et de leurs dérivées partielles en fonction des paramètres d'entrée.

* Etape 3: *reformulation* automatisée du cahier des charges.

* Etape 4: *tri* des valeurs des paramètres de sortie et de leurs dérivées partielles soumis à des contraintes.

Nous utilisons ensuite l'algorithme d'optimisation sous contraintes:

* Etape 5: *appel à l'algorithme d'optimisation* pour assurer les deux rôles suivants:

- A partir du jeu Pr_j , $j=1, m$ de paramètres réduits pour l'itération courante, *déterminer un nouveau jeu de paramètres* visant à satisfaire toutes les contraintes du problème et à fournir une solution optimale.
- *Vérifier si on se trouve dans une situation où toutes les contraintes sont respectées, tout en ayant une fonction objectif optimale.*

En conséquence, l'algorithme d'optimisation fournit deux informations qui sont:

- le jeu de paramètres d'entrée Pr_j , $j=1, m$ pour la nouvelle itération,
- un indicateur permettant à l'algorithme d'optimisation d'indiquer s'il considère qu'un optimum vérifiant toutes les contraintes est atteint ou pas.

* Etape 6: *appel aux procédures de transformation des valeurs des paramètres de l'espace réduit vers l'espace réel*

* Etape 7: test pour savoir si le cycle d'exploration doit être poursuivi sur la base de l'information fournie par l'algorithme d'optimisation dans l'étape précédente.

En résumé, le fonctionnement du logiciel de dimensionnement est basé sur un cycle itératif d'exploration de l'espace des solutions guidé par l'algorithme d'optimisation sous contraintes. Cet algorithme d'optimisation se sert lui-même, pour déterminer une direction de recherche, des informations fournies par le programme d'analyse et le programme de calcul de sensibilité. Ces informations prennent la forme de valeurs de paramètres de sortie et de dérivées partielles de ces paramètres de sortie en fonction des paramètres d'entrée. Elles doivent être recalculées à chaque itération et mises en forme afin de pouvoir être employées efficacement par l'algorithme d'optimisation.

II.3.b. Fonctionnement capable de traiter tous les cahiers des charges contraints possibles

Une propriété du fonctionnement du logiciel de dimensionnement évoqué précédemment se doit d'être mise en évidence:

Le processus d'exploration de l'espace des solutions est non orienté, à savoir non basé sur un processus de déduction de nouvelles valeurs sur la base de valeurs déjà connues.

Un processus d'exploration orienté de l'espace des solutions est typiquement celui qui est utilisé dans un programme de dimensionnement procédural: à partir d'un certain nombre de valeurs initiales, on utilise une procédure déduisant séquentiellement d'autres valeurs. Il en résulte:

- que le cheminement dans l'espace des solutions est figé et n'autorise pas la possibilité d'optimisation,
- qu'un seul cahier des charges peut être résolu par le logiciel.

Or ici, la démarche est tout autre: elle ne consiste pas à déterminer de nouvelles valeurs par déduction de valeurs déjà connues au sein d'une procédure prédéterminée et imposée.

Au contraire, elle laisse tous les paramètres d'entrée libres d'évoluer tout au long du processus, l'objectif étant d'en trouver un jeu final qui respecte:

- les contraintes du cahier des charges,
- les contraintes de dépendance des paramètres de sortie par rapport aux paramètres d'entrée.

Pour cela, l'algorithme d'optimisation sous contraintes est utilisé.

Cette philosophie de fonctionnement autorise à la fois:

- l'optimisation,
- la prise en charge de n'importe quelle combinaison de contraintes sur les paramètres d'entrée et de sortie, donc n'importe quelle combinaison de cahier des charges contraint.

On notera enfin que l'objectif visé de souplesse d'utilisation du logiciel de dimensionnement a en fait rendu nécessaire la définition et l'utilisation d'un tel mécanisme d'exploration non orienté de l'espace des solutions.

II.3.c. Limites de fonctionnement de l'application de dimensionnement

Le dernier point que nous allons évoquer concerne les limites de fonctionnement du logiciel que nous générons.

Ces limites sont très exactement celles des algorithmes d'optimisation de type gradient employés:

- 1°. *la nécessité de fournir un point de départ,*
- 2°. *la sensibilité au point de départ qui est fourni,*
- 3°. *l'impossibilité de gérer les paramètres discrets,*
- 4°. *l'impossibilité de gérer des discontinuités dans l'espace de recherche.*

Pour amorcer le processus d'exploration de l'espace des solutions, il faut d'abord *fournir une valeur initiale à l'ensemble des paramètres d'entrée.*

Ceci peut constituer une limite et un frein à l'utilisation du logiciel de dimensionnement généré car ce point de départ peut ne pas être évident à déterminer. Il peut éventuellement être constitué du jeu de paramètres d'entrée correspondant à une précédente machine dimensionnée mais cela exige de disposer d'un tel dimensionnement. Ce point de départ pourra être aussi généré de manière aléatoire.

Cette dernière solution est cependant déconseillée dans la mesure où les algorithmes d'optimisation du type gradient possèdent *une grande sensibilité au point de départ choisi.*

Cela signifie qu'en fonction de ce point de départ:

- Une solution qui n'est pas nécessairement la meilleure solution possible, peut être trouvée.
- Aucune solution n'est trouvée alors qu'il en existe.

Ces phénomènes se produisent dans des espaces de recherche dits multimodaux (*plusieurs optima relatifs existent*). En effet, les techniques d'optimisation de type gradient s'appliquent,

théoriquement, dans un domaine où la fonction est unimodale (*il n'existe qu'un optimum*). Aussi, lorsqu'on les emploie dans des espaces multimodaux, elles s'orientent vers le premier optimum relatif qui se présente et qui peut:

- ne pas être l'optimum absolu,
- ne pas satisfaire aux exigences des contraintes [33].

Les algorithmes d'optimisation sous contraintes de type gradient *n'ont pas la faculté de gérer les paramètres discrets*. En utilisant ces algorithmes, on se déplace forcément dans un espace mathématique réel et continu. Ceci est un handicap pour la conception en génie électrique où de nombreux paramètres sont discrets pour des raisons de coûts ou de réalisabilité. Ainsi, le nombre de spires qu'un bobinier peut mettre dans une encoche est forcément entier. De même, les diamètres possibles de fils pour faire une encoche sont souvent limités à ceux existant dans une liste de fils normalisés.

Enfin, le fait de devoir se déplacer dans un espace continu explique la quatrième limite qui est *l'impossibilité de gérer des discontinuités dans l'espace de recherche*.

Ceci est source de limitation dans la conception en génie électrique car cela ne permet pas de changer de structure de machine en cours d'exploration de l'espace des solutions. Ceci induirait une modification de la nature de l'espace, source de discontinuité.

II.4 Prise en charge des modèles analytiques des machines à concevoir proposée par la nouvelle approche

Après avoir détaillé les mécanismes méthodologiques permettant de générer automatiquement le logiciel de dimensionnement, puis étudié le fonctionnement du logiciel généré, l'objectif est à présent de:

formaliser tous les aspects théoriques, de notre approche, relatifs à la gestion des modèles analytiques.

Vis-à-vis de cet objectif, l'intérêt de ce qui va suivre est double:

- faire prendre du recul sur la manière dont ces modèles sont gérés,
- définir les solutions pour surmonter les difficultés que cette gestion peut impliquer.

C'est pourquoi, on détaillera les trois aspects suivants:

1° Nous montrerons *que notre approche propose en fait une gestion originale du modèle analytique sous une forme orientée.*

2° Nous discuterons brièvement *l'importance et les implications de la séparation des variables du modèle en un groupe de paramètres d'entrée et un groupe de paramètres de sortie.*

3° Nous proposerons *un moyen pour gérer les paramètres implicites d'un modèle.*

II.4.a. Mise en forme du modèle analytique proposée par la méthodologie - Introduction de la notion de modèle structural orienté

Dans ce paragraphe nous allons analyser la gestion des modèles analytiques proposée par notre approche.

Cette gestion se doit en effet d'être expliquée avec soin car elle constitue un point fondamental.

Rappelons que notre approche impose la saisie des équations du modèle sous la forme générique suivante:

1 variable = 1 expression dépendant d'autres variables. (2)

soit encore:

$C_i = f_i(P_1, P_2, P_3, \dots, P_m, C_1, C_2, \dots, C_{i-1}, C_{i+1}, \dots, C_n)^4$ (3)

Comme nous l'avons indiqué dans le paragraphe II.2.a.2., cette manière de procéder permet de séparer les variables du modèle en deux catégories:

1° *Les paramètres de sortie, $C_i, i=1, n$.* Ce sont les variables qui sont le terme de gauche d'une équation, autrement dit les variables qui s'expriment symboliquement en fonction d'autres variables.

2° *Les paramètres d'entrée, $P_j, j=1, m$.* Ce sont les variables du modèle qui ne sont jamais le terme gauche d'équation et qui ne s'expriment donc pas en fonction d'autres variables.

Afin de bien comprendre ce qu'implique cette manière de procéder, on va utiliser une représentation imagée des équations telles qu'elles sont utilisées. Ainsi, on va représenter l'équation (3) sous forme d'un arbre tel qu'il est visualisé sur la figure 9:

⁴: Il est interdit de rentrer une variable dépendant d'elle-même, d'où C_i ne s'exprime pas en fonction de C_i , car sinon on rentre dans le cadre de la gestion de variables implicites que nous détaillons dans le paragraphe II.4°c.

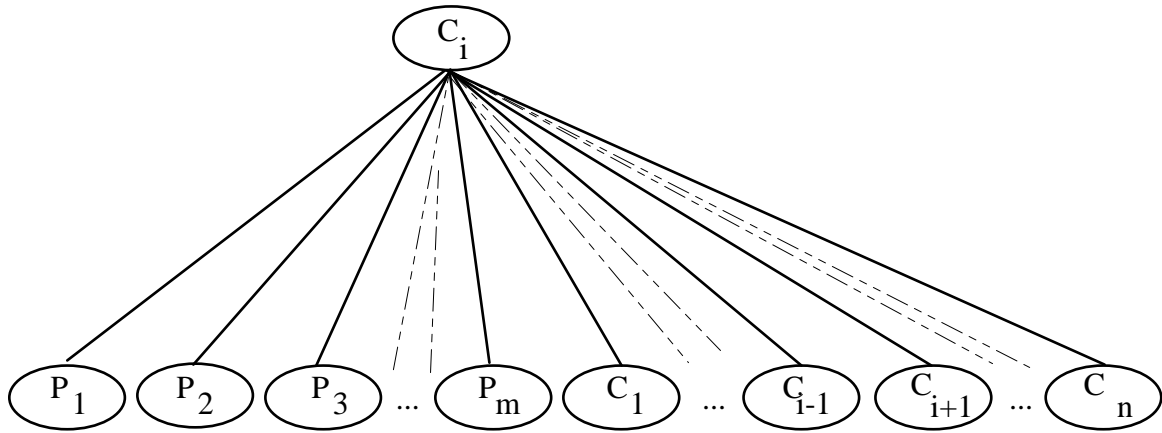


Figure 9: arbre de dépendance élémentaire d'une équation

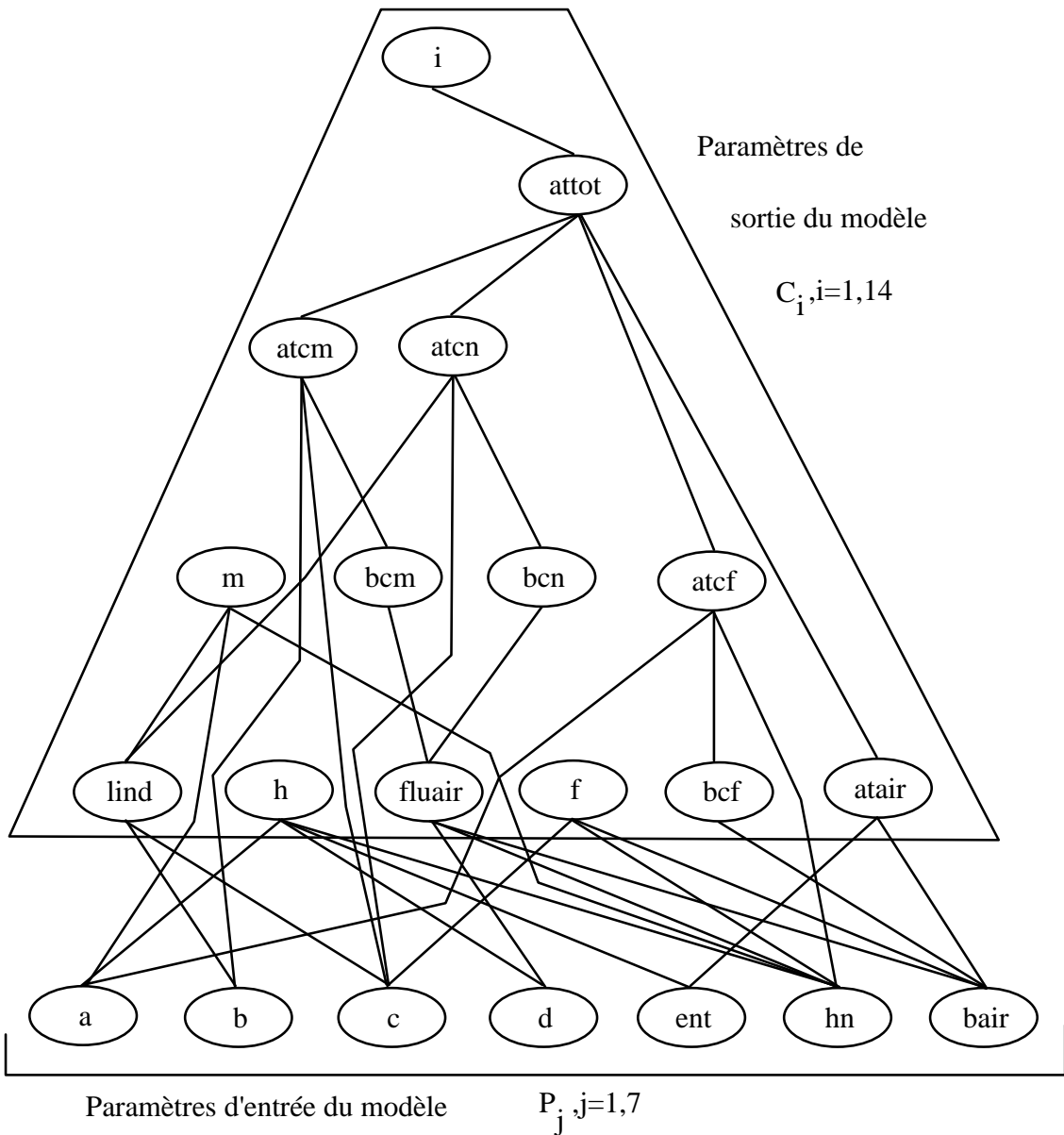


Figure 10: arbre de dépendance complet de l'exemple de électroaimant

Dans la phase de génération de code, tous les C_k $k \in [1, n]$ et $k \neq i$ de l'équation (3), sont substitués par leur équation jusqu'à ce C_k ne soit plus qu'une expression dépendant des paramètres d'entrée. Cela revient à assembler les arbres de dépendance élémentaires de chaque équation les uns aux autres pour construire l'*arbre de dépendance* complet du modèle.

Cet *arbre de dépendance* complet a été construit sur la figure 10 pour l'exemple de l'électroaimant dont le modèle a été fourni sur la figure 4.

La construction d'un tel arbre, pour tout modèle que notre approche est susceptible de manipuler, permet de donner une représentation des phénomènes fondamentaux suivants:

* Cet arbre permet de visualiser, grâce aux branches qu'il contient, les relations d'interdépendance entre les paramètres d'entrée et sortie fournies par le modèle analytique. Ces derniers sont des variables de construction, de conditions d'utilisation et de performances de la machine réalisée.

On visualise ainsi le type de contraintes particulières fournies par le modèle analytique. Il s'agit des contraintes d'interdépendance entre les paramètres de construction et d'utilisation de la machine avec les phénomènes physiques qui s'y produisent. Celles-ci sont inhérentes à la structure de la machine et aux phénomènes physiques qui s'y produisent. C'est pourquoi, on reprendra, pour ces contraintes, *la notion de contraintes structurales* introduites dans , [32], [33]

Ainsi, on désignera par la suite le modèle analytique sous le terme *de modèle structural de la machine à dimensionner*.

* La représentation de cet arbre visualise la forme orientée avec laquelle le modèle analytique est employé. Il comporte en effet des entrées constituées par les paramètres d'entrée et des sorties constituées par tous les paramètres de sortie.

De fait, on parlera par la suite *de modèle structural orienté de la machine à dimensionner*.

Cette mise en forme du modèle que nous préconisons permet:

1° d'exprimer symboliquement tout paramètre de sortie en fonction de l'ensemble des paramètres d'entrée,

2° d'enchaîner toutes les étapes de notre approche en disposant de ces expressions symboliques.

II.4.b Importance de la séparation des paramètres du modèle en paramètres d'entrée et de sortie

Dans notre approche, le fait de gérer le modèle analytique sous forme d'un modèle structural orienté, séparant les variables en paramètres d'entrée et de sortie peut paraître "gênant" a priori. Ceci est d'autant plus vrai que l'appartenance des différentes variables à chacun de ces groupes dépend uniquement de la manière dont les équations sont saisies.

En effet, ne vaudrait-il pas mieux mettre en place des algorithmes de maniement symbolique des équations du modèle, ayant pour objet de les transformer, afin d'imposer le jeu de paramètres d'entrée et de sortie? Ainsi, l'utilisateur peut souhaiter que le jeu de paramètres d'entrée corresponde aux paramètres du cahier des charges qu'on a l'habitude de lui soumettre, et que toutes les autres variables du modèle correspondent aux paramètres de sortie.

Or, dans le cadre du concept de logiciel de dimensionnement que notre méthodologie génère automatiquement, on peut dire que ceci est inutile car le *fait d'être paramètre d'entrée ou de sortie n'a pas grande importance* à la vue des raisons suivantes:

1° *Pour la gestion des contraintes, le logiciel de dimensionnement autorise à poser les mêmes contraintes d'égalité ou d'intervalle sur toutes les variables du modèle* et ceci indépendamment du fait qu'elles se retrouvent dans l'ensemble des paramètres d'entrée ou de sortie.

2° *Pour le processus de dimensionnement, être paramètre d'entrée ou de sortie n'a, a priori, aucune incidence.* Mettre certaines variables du modèle dans l'ensemble des paramètres d'entrée et d'autres dans l'ensemble des paramètres de sortie permet simplement la génération du logiciel de dimensionnement. Dès lors que ce logiciel est généré, rappelons que l'une de ses grandes originalités, est qu'il ne possède pas un mécanisme de fonctionnement orienté. En effet, il ne part pas des paramètres d'entrée pour déduire les paramètres de sortie mais il fait varier tous les paramètres simultanément dans l'objectif d'en trouver un jeu optimal compatible avec toutes les contraintes du problème:

- les contraintes d'égalité et d'intervalle posées sur les paramètres d'entrée et de sortie du modèle et qui proviennent du cahier des charges,
- les contraintes structurelles de la machine à dimensionner exprimées explicitement par les expressions symboliques des paramètres de sortie en fonction des paramètres d'entrée.

L'essentiel dans la gestion du modèle est en fait de formuler les relations structurelles qui existent entre les variables du modèle et qui resteront

structurellement les mêmes, indépendamment de la manière dont ces variables sont distribuées dans les ensembles de paramètres d'entrée et de sortie.

En marge de *cette inutilité à chercher à modifier les équations du modèle, on peut mettre en avant la difficulté d'une telle opération*. Il faut en effet mettre en oeuvre des concepts de calcul symbolique beaucoup plus complexes que les simples opérations de substitution ou de dérivation symbolique que nous utilisons, tels que des recherches de racines symboliques. Par ailleurs, il faut mettre au point et utiliser des algorithmes génériques de reformulation du modèle basés sur la théorie des graphes. Des recherches ont été menées en vue d'automatiser ce genre d'opérations, notamment pour déduire dynamiquement de nouveaux paramètres de la machine en fonction de paramètres initiaux qui auraient été imposés [57]. Cependant la complexité et la forte non linéarité des modèles que nous pouvons rencontrer en électrotechnique font que cette voie est assurément difficile à mettre en oeuvre.

En définitive, l'un des autres grands avantages de notre méthodologie est qu'elle utilise ce qui est "facilement" automatisable avec des techniques de calcul symbolique (*des dérivations et des substitutions*) et ce qui est performant avec des techniques numériques (*de l'optimisation sous contraintes avec des algorithmes de type gradient*) pour obtenir un ensemble au fonctionnement efficace.

Pour être complet au sujet de l'implication de la séparation des variables du modèle en un groupe de paramètres d'entrée et de sortie, on signalera la seule différence notable qui caractérise les deux groupes:

une valeur de départ doit être donnée pour chaque paramètre d'entrée afin d'amorcer le processus de dimensionnement alors que les évolutions d'un paramètre de sortie peuvent être tout à fait ignorées.

II.4.c Gestion des variables implicites proposée dans la nouvelle approche

Le dernier point important à traiter, quant à la gestion des modèles analytiques telle qu'elle est proposée par notre approche, concerne la manière dont celle-ci permet de gérer les variables implicites que l'on peut rencontrer.

On appelle variable implicite V d'un modèle une variable de ce modèle possédant les caractéristiques suivantes:

- Elle vérifie une équation du type $V=F(V)$.
- L'équation $V-F(V)=0$ ne connaît pas de racine exprimable sous forme symbolique.

Les voies habituelles pour gérer ce genre d'équation que l'on rencontre couramment en électrotechnique, sont de deux types:

1° on peut chercher numériquement la valeur de V en résolvant l'équation $V-F(V)=0$ grâce à des algorithmes de recherche des racines d'une équation du type Newton-Raphson.

2° on peut chercher itérativement une racine V grâce à des algorithmes dont le principe est rappelé sur la figure 11

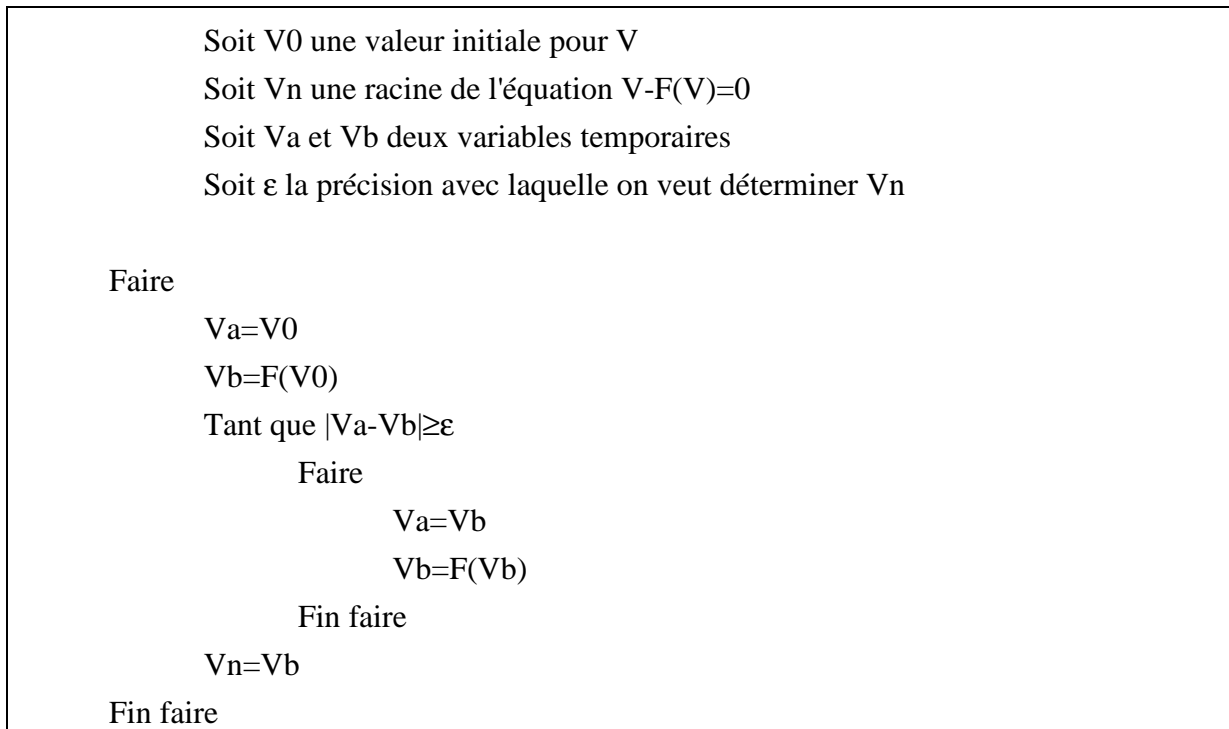


Figure 11: principe d'un algorithme itératif de détermination d'une variable implicite

Le problème d'une équation définissant une variable implicite est qu'elle n'est pas gérable sous la forme $V=F(V)$ par notre approche. On peut constater qu'une équation de ce type va bloquer la deuxième étape du processus de génération du code dans laquelle nous effectuons les substitutions symboliques afin d'exprimer les paramètres de sortie en fonction des paramètres d'entrée. En effet, si l'on part de $F(V)$, dans laquelle on substitue V par son expression, on se retrouve avec la fonction suivante $F(F(V))$, dans laquelle on doit à nouveau substituer V par son expression, et l'on se retrouve avec $F(F(F(V)))$,... On est entré dans une boucle infinie !

Une équation de ce type ne doit pas être fournie sous cette forme dans le jeu d'équations du modèle servant à faire générer le code du logiciel de dimensionnement. C'est pourquoi nous avons clairement indiqué que les équations qui devaient être fournies devaient vérifier la forme générique suivante:

$$C_i = f_i(P_1, P_2, P_3, \dots, P_m, C_1, C_2, \dots, C_{i-1}, C_{i+1}, \dots, C_n)^5$$

où l'on insiste bien sur le fait que C_i ne doit pas dépendre de C_i .

En fait, la solution que nous préconisons est de gérer l'équation définissant le paramètre implicite $V=F(V)$ sous la forme suivante:

$CV=V-F(V)$ où CV est un paramètre de sortie que l'on crée "artificiellement".

On résout ainsi le problème de substitution symbolique et on rend la génération de code à nouveau possible.

En contrepartie, dans l'exécutable final, lors du dimensionnement, on posera sur CV la contrainte d'égalité: $CV=0$.

Ainsi le problème de gestion des équations implicites non résolu sur le plan symbolique, est-il reporté sur le plan numérique, où il devient résoluble.

Conclusions du chapitre:

Les aspects théoriques de notre approche PASCOSMA ont été formalisés dans ce chapitre. Ils concernent:

- * la motivation du choix des algorithmes d'optimisation dont nous préconisons l'emploi,
- * la méthodologie que nous avons développée afin de générer automatiquement un logiciel convivial et souple à partir d'un modèle analytique,
- * le fonctionnement du logiciel de dimensionnement obtenu,
- * la gestion des modèles analytiques que nous avons définie

Nous pouvons, à présent, détailler dans le chapitre suivant:

- * le travail nécessaire pour réaliser un prototype informatique mettant en oeuvre tous les concepts de notre approche,
- * la manière dont ce prototype peut être utilisé afin d'effectuer du dimensionnement sous contraintes.

⁵: Il est à noter qu'aucun des C_x , pour $x \in [1, 2, \dots, i-1, i+1, \dots, C_n]$ ne doit dépendre non plus de C_i . Cela revient à dire que les équations du modèle ne doivent pas comporter de cycle de dépendance.

- CHAPITRE III -

Un prototype informatique de mise en oeuvre de la nouvelle approche proposée: développements nécessaires, possibilités et mode d'emploi, enseignements apportés

III. Un prototype informatique de mise en oeuvre de la nouvelle approche proposée: développements nécessaires, possibilités et mode d'emploi, enseignements apportés

Introduction:

Après avoir formalisé les aspects théoriques de notre approche PASCOSMA dans le chapitre précédent, l'objectif de ce chapitre est de détailler la réalisation d'un prototype informatique ainsi que quelques considérations connexes à celle-ci.

C'est pourquoi ce chapitre sera composé de trois parties:

1° La première détaillera *le travail de réalisation du prototype qui a été nécessaire ainsi que les solutions d'ordre informatiques et techniques qui ont été choisies.*

2° La seconde donnera *un aperçu du mode d'emploi et des fonctionnalités du prototype obtenu.*

3° La dernière abordera *les enseignements que ce prototype nous a apportés.* Ainsi nous détaillerons:

* *Les problèmes qu'il nous a permis de mettre en évidence et les solutions que nous avons imaginées pour les résoudre.* Cela concerne des points comme:

-le risque d'explosion de la taille du code généré,

-certaines limites du processus d'exploration de l'espace des solutions.

* *L'expérience en terme de démarche d'utilisation optimale et de "philosophie d'emploi"* que tout utilisateur de notre approche se devra d'adopter.

III.1. Développements nécessaires à la mise en oeuvre de la nouvelle approche PASCOSMA proposée

Le paragraphe qui suit va détailler les développements nécessaires et les solutions techniques adoptées pour l'implantation de notre approche.

III.1.a. Développement et implantation des fonctionnalités du niveau 1 de la méthodologie PASCOSMA

Rappelons que *les fonctionnalités du niveau 1 de la méthodologie sont celles qui, par le biais de calculs symboliques et de programmations automatiques massives, assurent la génération du programme de calcul d'analyse et du programme de calcul de sensibilité.*

Pour les développer nous nous sommes appuyés sur l'environnement de manipulation symbolique MACSYMA [39] qui, malgré une certaine lenteur due certainement à son langage de programmation de type interprété, offre les fonctions et opérateurs élémentaires pour faire du calcul symbolique et de la programmation automatique.

L'implantation de ce niveau 1 de notre méthodologie a nécessité le développement de plus de sept bibliothèques de fonctions. Celles-ci sont destinées à être chargées dans l'environnement MACSYMA dès lors qu'on veut les employer. Elles mettent à la disposition de l'utilisateur plus de 72 nouvelles instructions que l'on peut regrouper en deux catégories:

- * des instructions prenant en charges des fonctions de base comme:
 - la recherche des paramètres d'entrée et de sortie du modèle,
 - le calcul symbolique de tout paramètre de sortie en fonction des paramètres d'entrée,
 - la programmation en langage FORTRAN de tout paramètre de sortie,
 - le calcul symbolique de la dérivée partielle de tout paramètre de sortie en fonction de n'importe quel paramètre d'entrée,
 - ...
- * des instructions de haut niveau, en nombre très réduit, qui permettent:
 - de déclarer le contenu final du logiciel de dimensionnement, autrement dit:
 - les paramètres de sortie à prendre en compte,
 - les paramètres intermédiaires à utiliser⁶,
 - le paramètre d'entrée ou de sortie qui sera la fonction objectif,
 - de lancer la génération complète du programme d'analyse et du programme de sensibilité.

L'utilisateur final du système n'a pas besoin d'appeler les instructions de base. En effet, il peut se contenter d'appeler les instructions de haut niveau qui se chargent d'appeler celles de bas niveau nécessaires. L'essentiel des instructions de haut niveau sera détaillé dans le paragraphe III.2.b. lorsque nous montrerons un exemple de génération d'application.

⁶: cette notion de paramètres intermédiaires sera introduite dans le paragraphe III.3.a.

L'ensemble des bibliothèques développées représente plus de 5100 lignes de code (*soit plus de 300 ko de code source commenté*) dans le langage de programmation de l'environnement MACSYMA.

Par ailleurs, afin que ce générateur de code soit un produit opérationnel, nous avons fourni un travail constant afin d'assurer les points suivants:

1° *La fiabilité du code généré.* Pour cela, nous avons défini des procédures de test rigoureuses du code généré. Celles-ci consistent en une comparaison des valeurs de paramètres de sortie et de dérivées partielles calculées grâce à ce code avec des valeurs fournies par d'autres environnements tel MATHCAD [40]. Ceci est fait sur un certain nombre d'exemples suffisamment simples pour pouvoir être implantés dans ces autres environnements mais significatifs des situations auxquelles le générateur peut être confronté. Ces procédures ont été utilisées à chaque fois que des évolutions ont été réalisées garantissant ainsi la qualité du code généré.

2° *La robustesse du générateur de code.* Pour cela, toutes les sources d'erreurs recensées de la part de l'utilisateur font l'objet de garde-fous. Ainsi, il est impossible parmi d'autres choses:

- d'associer à un paramètre de sortie deux équations différentes,
 - de vouloir générer un paramètre de sortie qui n'appartient pas au modèle,
 - ou encore d'employer des noms de variables trop longs,
- sans voir apparaître un message d'erreur diagnostiquant précisément le problème afin que l'utilisateur puisse le corriger.

On notera enfin que, du fait de la lenteur de l'environnement MACSYMA, toute la programmation automatique des codes à générer n'est pas faite dans cet environnement. Nous avons développé la génération des parties de code ne nécessitant pas de calcul symbolique, ou de retranscription dans le langage final de programmation d'une formule symbolique, dans un autre langage de programmation, notablement plus rapide, qui est en l'occurrence Le-lisp [35]. Ce module de génération de code indépendant représente de l'ordre de 2000 lignes d'instructions (*soit près de 100 ko de code source commenté*): cette taille est non négligeable car nous devons y assurer la génération de toutes les parties du logiciel qui gèrent des aspects tels que:

- la gestion des variables nécessaires pour stocker les résultats des calculs d'analyse et de sensibilité,
- les lignes de codes qui font l'interface entre la partie du programme générée spécifiquement pour chaque modèle et les lignes de codes figées (*cf. le programme d'optimisation ou les routines du niveau 2 de la méthodologie*).

Par ailleurs, dans ce module, nous assurons encore:

- la génération des fichiers de commande qui vont permettre la compilation puis l'édition de liens,
- la supervision de la compilation et de l'édition de liens,
- la génération d'un fichier qui va permettre l'autoconfiguration de l'interface graphique (*voir paragraphe III.1.c*).

Les lignes de codes du programme d'analyse et du programme de calcul de sensibilité sont générées en FORTRAN [7].

III.1.b. Algorithme d'optimisation utilisé

Pour mettre en oeuvre notre méthodologie, nous avons dû choisir un algorithme d'optimisation. Conformément à nos intentions, nous n'avons pas développé cet algorithme mais nous l'avons choisi dans une bibliothèque d'algorithmes numériques existante.

Notre choix final s'est porté sur l'algorithme VF13AD de la bibliothèque HARWELL [29]. Celui-ci avait en effet les qualités principales requises:

- 1° *il s'agit d'un algorithme du type gradient* donc utilisant les dérivées partielles de la fonction objectif et des paramètres de sortie pour trouver une direction de recherche,
- 2° *cet algorithme prend en charge la gestion de contraintes sur des paramètres non linéaires.*

Mais, au delà de ces caractéristiques, cet algorithme offre des garanties de performances par les mécanismes perfectionnés de fonctionnement qu'il emploie:

- * La méthode d'optimisation sous contraintes utilisée est, bien sûr, itérative.
- * Dans chaque itération, cet algorithme minimise une approximation quadratique de la fonction Lagrangienne⁷. Il tient compte des contraintes en les approximant linéairement. La dérivée seconde de la fonction Lagrangienne est estimée automatiquement.
- * Concernant la gestion des contraintes, une technique, appelée 'watchdog technique' est spécialement utilisée pour accélérer la convergence lorsque les valeurs initiales des variables sont loin de la solution dans un espace où les paramètres contraints sont non linéaires. Cette technique autorise, au cours des itérations, des directions de recherche pouvant initialement violer largement les contraintes ce qui peut permettre, paradoxalement, de converger ensuite plus rapidement vers la solution. Ce comportement s'explique dans des espaces non linéaires. En effet, respecter à chaque

⁷: une fonction Lagrangienne est une fonction de reformulation de la fonction objectif permettant de prendre en compte les contraintes. Pour plus de précisions on pourra se reporter avec profit aux références [33], [46], [65].

itération les frontières des contraintes qui peuvent avoir une très forte courbure, peut fortement ralentir la convergence. Cette technique gère aussi les éventuels retours en arrière si les nouvelles directions de recherche mènent à des échecs.

On trouvera la description complète de la méthode utilisée dans cet algorithme d'optimisation dans [46]. Concernant une description détaillée de la 'watchdog technique', on pourra se reporter à la référence [15]. Enfin, la procédure de minimisation d'une fonction quadratique utilisée est détaillée dans [47].

III.1.c. Développements et implantation des fonctionnalités du niveau 2 de la méthodologie PASCOSMA

Rappelons que les fonctionnalités du niveau 2 de la méthodologie sont celles qui doivent permettre d'utiliser le logiciel de conception généré de manière souple et conviviale.

III.1.c.1. Développement des parties incluses dans l'exécutable de dimensionnement

Les fonctionnalités à inclure dans l'exécutable de dimensionnement généré doivent servir à rendre l'utilisation de l'algorithme d'optimisation transparente et souple. Mais surtout, elles doivent permettre une adaptation dynamique de l'exécutable généré à tout cahier des charges que l'utilisateur peut saisir.

Dans ce cadre, il a fallu assurer les fonctions du 2^{ème} niveau de notre méthodologie exposées dans le chapitre précédent (*cf. paragraphe II.2.b.2.*) et qui sont:

1° La reformulation des contraintes du cahier des charges sous forme de contraintes compréhensibles par l'algorithme d'optimisation. Sachant que cet algorithme sait gérer les contraintes du type " $=0$ " et les contraintes de type " ≥ 0 ", les procédures de transformation que nous avons implantées peuvent être retrouvées dans le tableau 3.

2° La normalisation donc le passage de l'espace réel à l'espace réduit et vice-versa. Les procédures implantées afin d'assurer cette fonction sont conformes à celles décrites dans le tableau 1 (*pour les valeurs des fonctions*) et dans le tableau 2 (*pour les valeurs des dérivées partielles*). Il est à signaler que nous n'avons pas implanté de fonctionnalités permettant d'évaluer automatiquement un ordre de grandeur de la valeur minimale et maximale de la fonction objectif (*valeurs qui sont nécessaires pour normaliser cette fonction objectif*). C'est pourquoi une estimation de ces valeurs doit être communiquée par l'utilisateur au programme, au même titre que le sont des valeurs minimales et maximales de contraintes de type intervalle.

3° Le tri des valeurs des paramètres et des valeurs des dérivées partielles des paramètres en fonction du type de contraintes qui sont affectées aux paramètres.

L'ensemble de toutes ces fonctionnalités a été développé sous la forme de bibliothèques de routines FORTRAN qui sont liées avec les lignes de code du programme de calcul de sensibilité, les lignes de codes du programme de calcul d'analyse et l'algorithme d'optimisation. L'ensemble représente environ 500 lignes d'instructions (*soit 25 ko de code source commenté*). Il est à noter qu'une partie du code nécessaire pour assurer ces fonctionnalités doit être fabriqué spécifiquement pour chaque application de dimensionnement générée. Ce travail est assuré par le module écrit en langage Le-Lisp qui assure la génération du code non pris en charge par les bibliothèques développées sous MACSYMA.

III.1.c.2. Développement de l'interface conviviale et autoconfigurable

L'interface graphique du prototype que nous avons réalisée a demandé un soin et un travail important car nous voulions que celle-ci ait:

la capacité de s'adapter dynamiquement à l'application de dimensionnement à piloter,

tout en offrant les fonctionnalités et la convivialité nécessaires. Ainsi, il nous a semblé indispensable:

- d'utiliser des objets de haut niveau du type menus déroulants, boîtes de dialogues ou boutons qui sont les composants de base permettant de développer des interfaces aux normes actuelles de convivialité,
- d'offrir des possibilités aussi complexes que des tracés de courbes avec une fonction de zoom, afin de visualiser:
 - les évolutions d'un paramètre de sortie en fonction d'un paramètre d'entrée,
 - les évolutions des paramètres d'entrée et de sortie au cours du dimensionnement.

Pour pouvoir atteindre ces objectifs, nous avons dû trouver des solutions aux problèmes suivants:

1° *Définir la forme que doit prendre cette interface à l'écran* afin d'être adaptée à toutes les situations auxquelles elle peut être confrontée.

Un exemple de difficulté de ce type qu'il a fallu surmonter est le nombre de paramètres d'entrée et de sortie qui varie fortement d'un modèle à l'autre. En l'occurrence, ce problème a été résolu en utilisant des listes avec ascenseur permettant:

- de gérer un nombre non limité de boutons, chacun de ces boutons étant associé à un paramètre d'entrée ou de sortie
- d'accéder à tout bouton de la liste en déplaçant l'ascenseur.

Ces listes avec ascenseurs sont visibles sur la figure 12 (*cf. paragraphe III.2.c*).

2° *Définir le type de langage de programmation adapté*. Il fallait, grâce à ce langage, surmonter des difficultés telles que:

- la création dynamique des objets de l'interface (*ascenseurs, boutons,...*),
- créer de manière simultanée:
 - les comportements associés à chacun de ces objets (*cf. l'ouverture d'une fenêtre de gestion du paramètre d'entrée ou de sortie sur lequel on vient de cliquer,...*),
 - les variables nécessaires à la gestion de chacun de ces objets,
 - les liens entre ces différents objets (*cf. l'objet liste avec ascenseur qui doit être lié à la liste de boutons dont il doit assurer le défilement*)

Face à toutes ces contraintes, nous avons utilisé un langage de programmation orienté objet qui offre les concepts de base nécessaires tels que:

- la possibilité de définir des classes d'objets avec leurs comportements et leurs variables associées,
- la possibilité de créer dynamiquement ces objets.

Notre choix final s'est porté vers Le-Lisp [35] et nous avons utilisé les classes d'objet Aida [2] (*boutons, listes, fenêtres,...*) développées dans ce langage.

Dès lors que les éléments de base étaient disponibles, nous avons pu assurer le développement de l'interface graphique, dont la vocation est de:

- * saisir toutes les données et toutes les demandes provenant de l'utilisateur (*type des contraintes, valeurs limites des contraintes, demandes de calcul d'analyse, demande de dimensionnement...*) pour les fournir ensuite au logiciel de dimensionnement,
- * visualiser les résultats des calculs et du dimensionnements, notamment grâce à des courbes, que le logiciel de dimensionnement lui fournira,
- * enfin, de contrôler les actions que l'utilisateur fait, ou peut demander. Ainsi, elle va imposer qu'un cahier des charges soit effectivement saisi avant de lancer un dimensionnement ou elle va empêcher que l'utilisateur puisse entrer une valeur minimale de contrainte d'intervalle qui soit supérieure à la valeur maximale.

Ceci a nécessité de notre part des développements importants et rigoureux, puisque toute l'interface est conçue pour fonctionner sur la base d'une structure variable.

La taille du code résultant est de l'ordre de 6000 lignes d'instructions réparties en huit bibliothèques (*ce qui représente de l'ordre de 400 ko de code source commenté*).

Nous avons aussi dû définir un protocole de communication entre l'interface graphique et le logiciel de dimensionnement. Celui-ci se fait par l'intermédiaire de fichiers.

On notera enfin que, pour permettre l'autoconfiguration de l'interface, un fichier de description des caractéristiques de l'application de dimensionnement à piloter doit lui être fourni. Ce fichier est généré en même temps que le code du programme d'analyse et du programme de calcul de sensibilité (*voir paragraphe III.1.a*).

III.2. Une session utilisateur complète avec le prototype réalisé

III.2.a. Description de l'exemple

Pour donner un aperçu du fonctionnement du prototype logiciel réalisé, nous allons reprendre l'exemple du dimensionnement de l'électroaimant détaillé dans l'annexe A et dont le modèle structural employé est rappelé aussi sur la figure 4 (*cf. paragraphe II.2.a.2*).

III.2.b. Utilisation du générateur du logiciel de dimensionnement automatique

Dès lors que l'on dispose du modèle de la structure à dimensionner, il faut le fournir au générateur de code qui va assurer la génération du logiciel de dimensionnement.

Dans la version actuelle du prototype de générateur de code, les équations du modèle structural doivent être fournies dans un fichier contenant ces équations ramenées au format MACSYMA [39].

Ainsi, pour générer l'application de dimensionnement de l'électroaimant (*voir annexe A*) que nous prenons ici en exemple, le contenu de ce fichier sera le suivant, :

```

eqka:ka=76.1;
eqkb:kb=1.26;
eqkc:kc=129.5;
eqlind:lind=2*c+b;
eqh:h=d+ent+a+hn;
eqatair:atair=2*(bair/(4*%pi*10^-7)*ent);
eqfluair:fluair=bair*c*hn;
eqbcm:bcm=fluair/(hn*d);
eqatcm:atcm=(ka*exp(kb*bcm^2)+kc)*bcm*(b+c);
eqbcf:bcf=bair;
eqatcf:atcf=(ka*exp(kb*bcf^2)+kc)*bcf*(2*a+hn);
eqbcn:bcn=fluair/hn^2;
eqatcn:atcn=(ka*exp(kb*bcn^2)+kc)*bcn*(lind-c);
eqattot:attot=atair+atcm+atcf+atcn;
eqi:i=attot/10;
eqf:f=bair^2/(4*%pi*10^-7*2)*c*hn^2;
eqm:m=(d*lind+2*a*c+lind*hn)*hn*7800+0.071;

```

On constatera que chaque équation est précédée d'un double point et d'une chaîne de caractères. Cette chaîne de caractères sert simplement à donner un nom aux équations afin de pouvoir les manipuler plus simplement par la suite.

Il faut ensuite fournir au générateur un deuxième fichier, contenant les instructions qui vont permettre d'assurer la génération automatique du code. Ainsi, pour l'exemple de l'électroaimant, ce fichier est le suivant:

```
fichier_equations("eqtest.mac");

cree_liste_des_equations();

verif_var_mod(liste_des_equations);

fixe_nom_ddl([a, b, bair, c, d, ent, hn],[a, b, bai, c, d, ent, hn]);

ajoute_contrainte(lind,lin);
ajoute_contrainte(h,h);
ajoute_contrainte(atair,ata);
ajoute_contrainte(fluair,flu);
ajoute_contrainte(bcm,bcm);
ajoute_contrainte(atcm,atc);
ajoute_contrainte(bcf,bcf);
ajoute_contrainte(atcf,atf);
ajoute_contrainte(bcn,bcn);
ajoute_contrainte(atcn,atn);
ajoute_contrainte(atot,ato);
ajoute_contrainte(i,i);
ajoute_contrainte(f,f);
ajoute_contrainte(m,m);

fixe_fonction_objectif(m,mob);

genere();
```

Ce fichier contient des exemples des principales instructions de haut niveau que nous avons développées dans l'environnement MACSYMA pour assurer la génération du code:

- * La première instruction "fichier_equations("eqtest.mac")" sert à donner le nom du fichier contenant les équations du modèle. Ce fichier s'appelle en l'occurrence "eqtest.mac".
- * La deuxième instruction "cree_liste_des_equations()" ordonne au système de recenser les équations se trouvant dans le fichier indiqué précédemment.
- * L'instruction "verif_var_mod(liste_des_equations)" réalise un certain nombre de vérifications concernant les équations du modèle (*syntaxe des variables utilisées, vérification qu'un paramètre de sortie n'est défini que par une équation et une seule,...*).
- * L'instruction "fixe_nom_ddl" sert à affecter, à chaque paramètre d'entrée du modèle (*qui sont en l'occurrence "[a, b, bair, c, d, ent, hn]"*), une chaîne de caractères qui sera utilisée pour gérer ces paramètres d'entrée dans le programme FORTRAN⁸.

⁸: toutes les instructions que nous développées permettent à l'utilisateur d'imposer les noms de symboles à utiliser dans le programme FORTRAN. Ceci doit lui permettre de faciliter une relecture éventuelle des codes générés.

* Les instructions "ajoute_contrainte" servent ensuite à déclarer au système les paramètres de sortie que l'utilisateur désirera contraindre dans le logiciel de dimensionnement final. Ce sont donc les paramètres de sortie pour lesquels le système va calculer l'expression symbolique en fonction des paramètres d'entrée, ainsi que les expressions symboliques des dérivées partielles en fonction des paramètres d'entrée, pour les programmer ensuite en FORTRAN. Cette instruction possède deux arguments: le premier indique le nom du paramètre de sortie à prendre en compte; le second donne un symbole qui servira à désigner ce paramètre de sortie dans le programme FORTRAN.

* Vient ensuite la fonction "fixe_fonction_objectif(m,mob)" qui sert à désigner quel est le paramètre de sortie qui fera office de fonction objectif. Cette instruction possède deux arguments: le premier désigne le paramètre d'entrée ou de sortie du modèle qui sera la fonction objectif, le second donne une chaîne de caractères désignant cette fonction objectif dans le programme FORTRAN.

* Enfin la dernière instruction, "genere()" sert à lancer la génération du code.

Dès lors que l'instruction "genere()" est lancée, les instructions de base nécessaires, développées sous MACSYMA, sont appelées pour prendre en charge la génération de lignes de code du programme d'analyse et du programme de calcul de sensibilité.

Lorsque ce travail a été exécuté, un modèle indépendant, développé en Le-lisp, assure encore la génération d'un certain nombre de ligne de codes. Puis il supervise la compilation et l'édition de liens entre tous les éléments nécessaires pour arriver au logiciel de dimensionnement final.

III.2.c Utilisation du logiciel généré de CAO sous contraintes

Dès lors que le logiciel de dimensionnement est généré, on peut lancer l'interface d'utilisation. Celle-ci, après avoir lu un fichier d'informations sur celui-ci qu'elle doit piloter, se configure automatiquement en fonction de celui-ci. Ceci dure moins d'une à deux secondes et on se retrouve devant l'interface qui est reproduite sur la figure 12 pour le cas du dimensionnement de notre électroaimant.

Cette fenêtre se compose essentiellement de trois lignes de boutons que l'on fait peut défiler avec un ascenseur:

-la première ligne, intitulée "Menu", contient les boutons permettant d'accéder à toutes les fonctionnalités de l'interface graphique (*lecture et écriture de données dans des fichiers, lancement de calcul d'analyse et de calcul de sensibilité, lancement du dimensionnement,...*).

Cependant cette possibilité est appelée à devenir optionnelle, voire à disparaître, puisque l'utilisateur du générateur de code n'est pas censé accéder aux lignes du programme.

-la deuxième ligne, intitulée "Liste des PARAMETRE(s) D'ENTREE", contient un bouton pour chaque paramètre d'entrée du modèle. Ces boutons donnent accès à une fenêtre qui va permettre de gérer les contraintes relatives à chaque paramètre d'entrée.

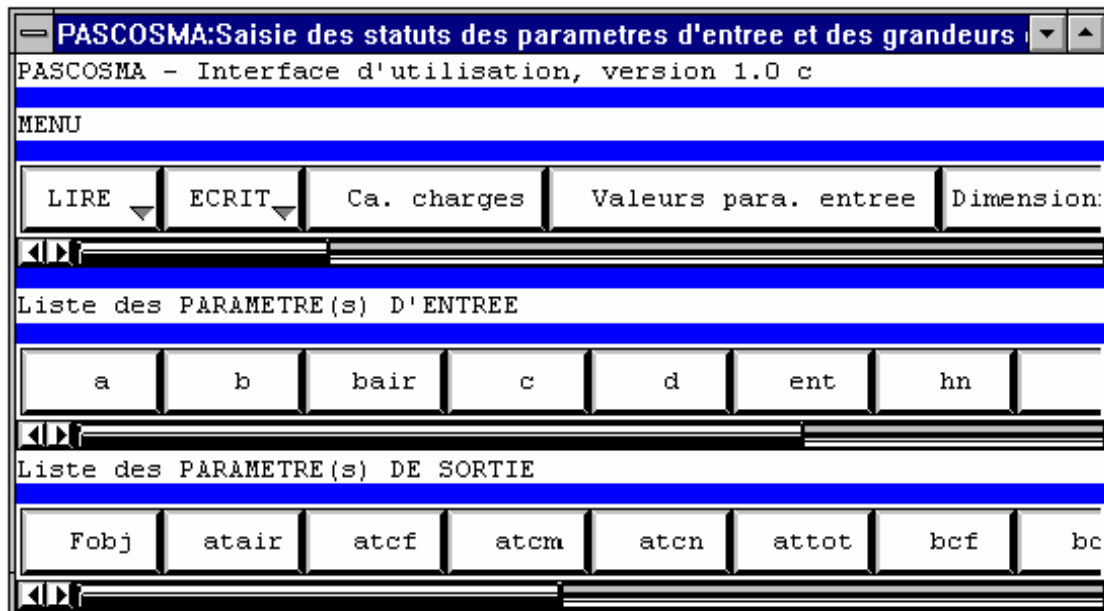


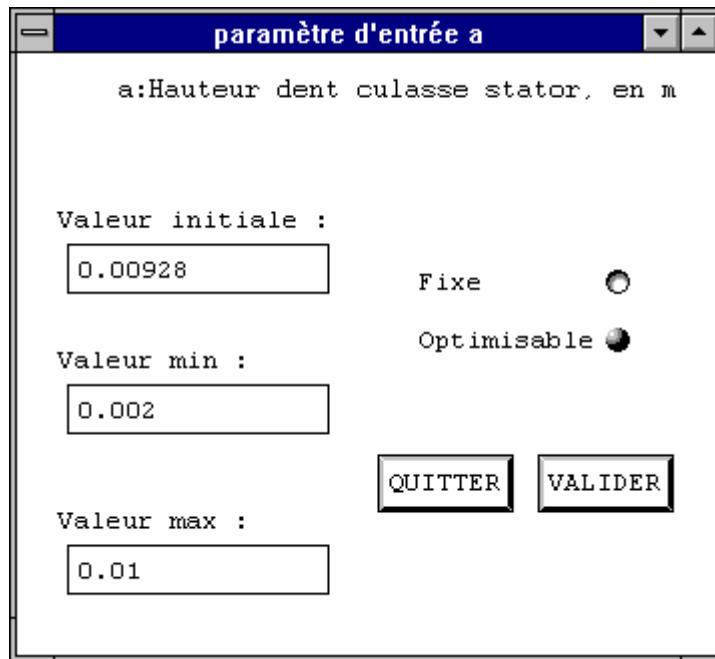
Figure 12: Fenêtre principale de l'interface d'utilisation des applications de type PASCOSMA (exemple de l'électroaimant)

-la troisième ligne, intitulée "Liste des PARAMETRE(s) DE SORTIE", contient un bouton pour chaque paramètre de sortie pour lequel on a demandé la génération du code nécessaire (cf. instruction "ajoute_contrainte(...) dans le paragraphe précédent). Ces boutons ouvrent une fenêtre pour la gestion des contraintes relatives à chaque paramètre de sortie.

Nous allons donner ici un aperçu des principales des fonctions assurées grâce à cette interface graphique, ce qui illustre des possibilités en matière de conception qu'offre le logiciel automatiquement généré.

La première fonctionnalité assurée par cette interface graphique est bien sûr *la saisie du cahier des charges* donc des contraintes sur les paramètres d'entrée et les paramètres de sortie.

Pour les contraintes sur les paramètres d'entrée, il faut cliquer sur le bouton portant le nom de ce paramètre d'entrée. Une fenêtre du style de celle visualisée sur la figure 13 apparaîtra alors.



**Figure 13: fenêtre de gestion d'un paramètre d'entrée
(exemple de l'électroaimant)**

Cette fenêtre permet:

* *D'imposer une contrainte de type "Fixe" sur le paramètre d'entrée.* Dans ce cas, il faut renseigner le champ "Valeur initiale" et cela revient alors à poser une contrainte de type égalité sur le paramètre d'entrée:

$$\text{paramètre d'entrée} = \text{"Valeur initiale"}$$

* *D'imposer une contrainte de type "Optimisable" sur le paramètre d'entrée.* Dans ce cas, il faut renseigner les champs "Valeur min." et "Valeur max." et cela revient alors à poser la contrainte d'intervalle suivante sur le paramètre d'entrée:

$$\text{"Valeur min."} \leq \text{paramètre d'entrée} \leq \text{"Valeur max."}$$

La valeur saisie grâce au champ "Valeur initiale" servira à affecter une valeur initiale au paramètre d'entrée.

On notera donc, qu'en plus du type de la contrainte, cette fenêtre permet de saisir la valeur initiale des paramètres d'entrée c'est-à-dire *la saisie du point de départ du dimensionnement.*

Pour les contraintes sur les paramètres de sortie, il faut cliquer sur le bouton portant le nom de ce paramètre de sortie. Une fenêtre du style de celle visualisée sur la figure 14 apparaîtra alors.

Outre les contraintes "Fixe" et "Optimisable" analogues aux précédentes, cette fenêtre permet:

* *d'imposer un statut de type "Ignoré" sur le paramètre de sortie.* Dans ce cas, on indique au système que le paramètre est libre de prendre n'importe quelle valeur, autrement dit on ne pose aucune contrainte sur son évolution.

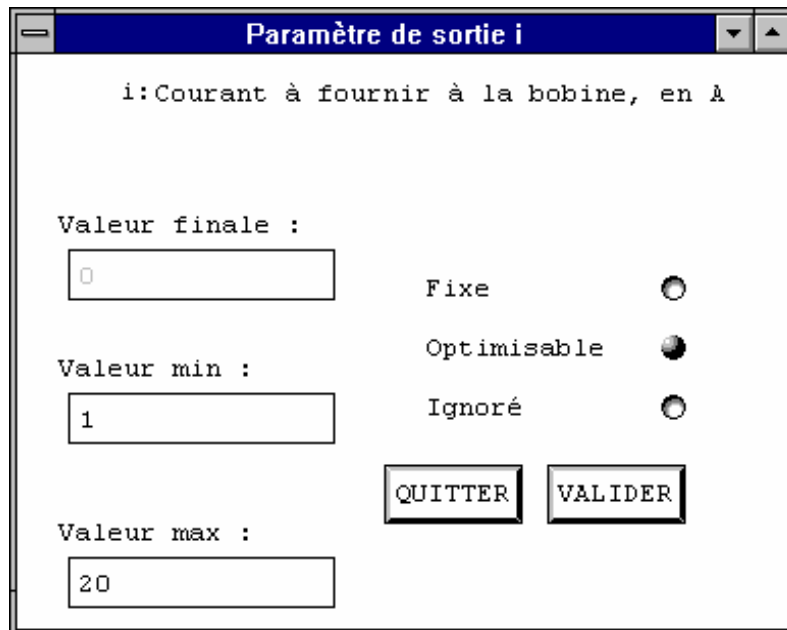


Figure 14: fenêtre de gestion d'un paramètre de sortie
(exemple de l'électroaimant)

Le bouton "Fobj", quant à lui, provoque l'ouverture d'une fenêtre représentée sur la figure 15 pour la saisie d'un intervalle de variation estimé de la fonction objectif. Les valeurs limites de cet intervalle sont nécessaires pour des besoins de normalisation du problème.

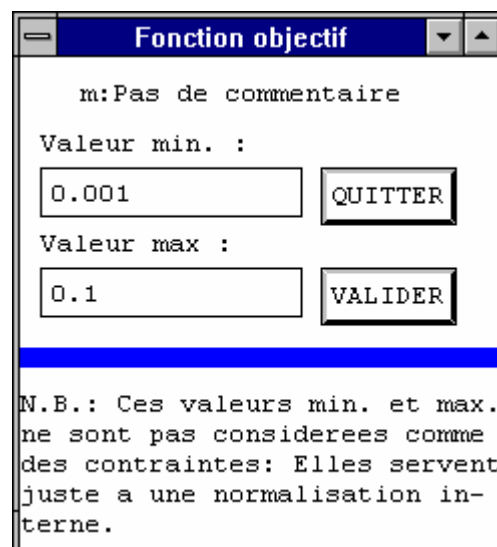


Figure 15: fenêtre de gestion de la fonction objectif
(exemple de l'électroaimant)

Dès lors que le cahier des charges est saisi, celui-ci peut-être visualisé grâce à une fenêtre dont on a donné une représentation sur la figure 16.

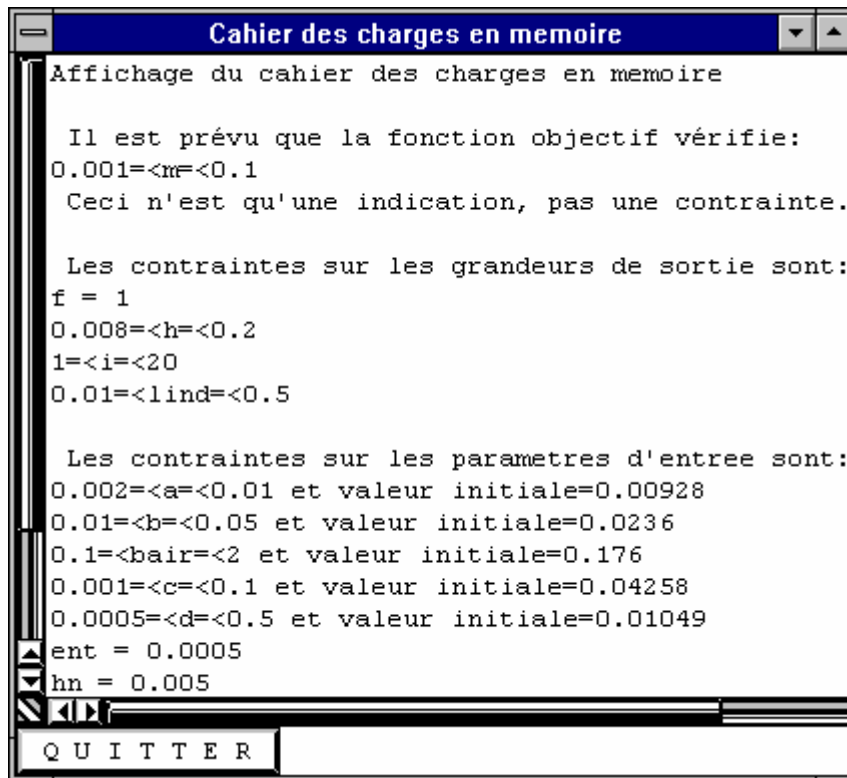
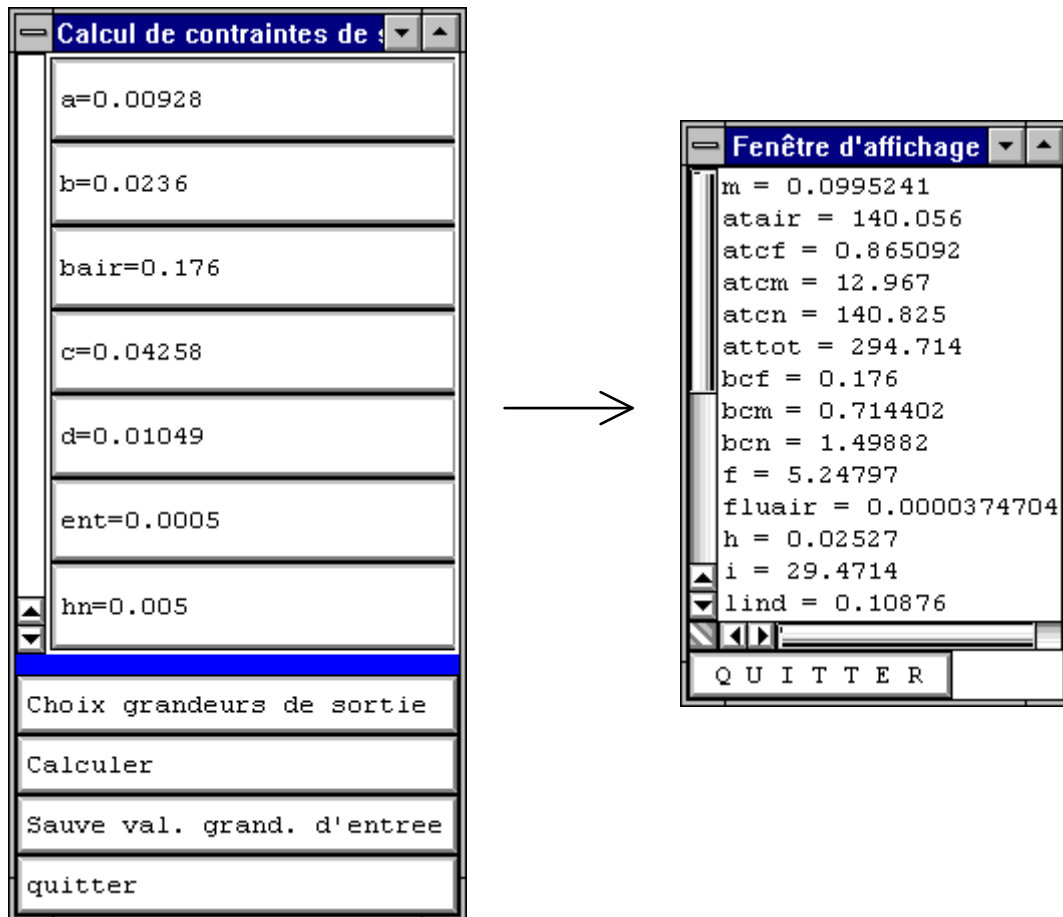


Figure 16: exemple de cahier des charges saisi
(exemple de l'électroaimant)

Cette interface donne la possibilité de mener:

- * Des *calculs d'analyse*, comme cela est indiqué sur la figure 17, c'est à dire des calculs qui déterminent la valeur des paramètres de sortie du modèle pour un jeu de paramètres d'entrée.
- * De mener des *calculs de sensibilités* (c'est-à-dire plus précisément des calculs de dérivées partielles de paramètres de sortie en fonction de paramètres d'entrée qui pourront servir au calcul d'analyse) comme cela est indiqué sur la figure 18.
- * Enfin, de tracer l'évolution d'un paramètre d'entrée en fonction d'un paramètre de sortie, comme cela est indiqué sur la figure 19.



**Figure 17: exemple de calcul d'analyse (=calcul des paramètres de sortie pour un jeu de paramètres d'entrée)
(exemple de l'électroaimant)**

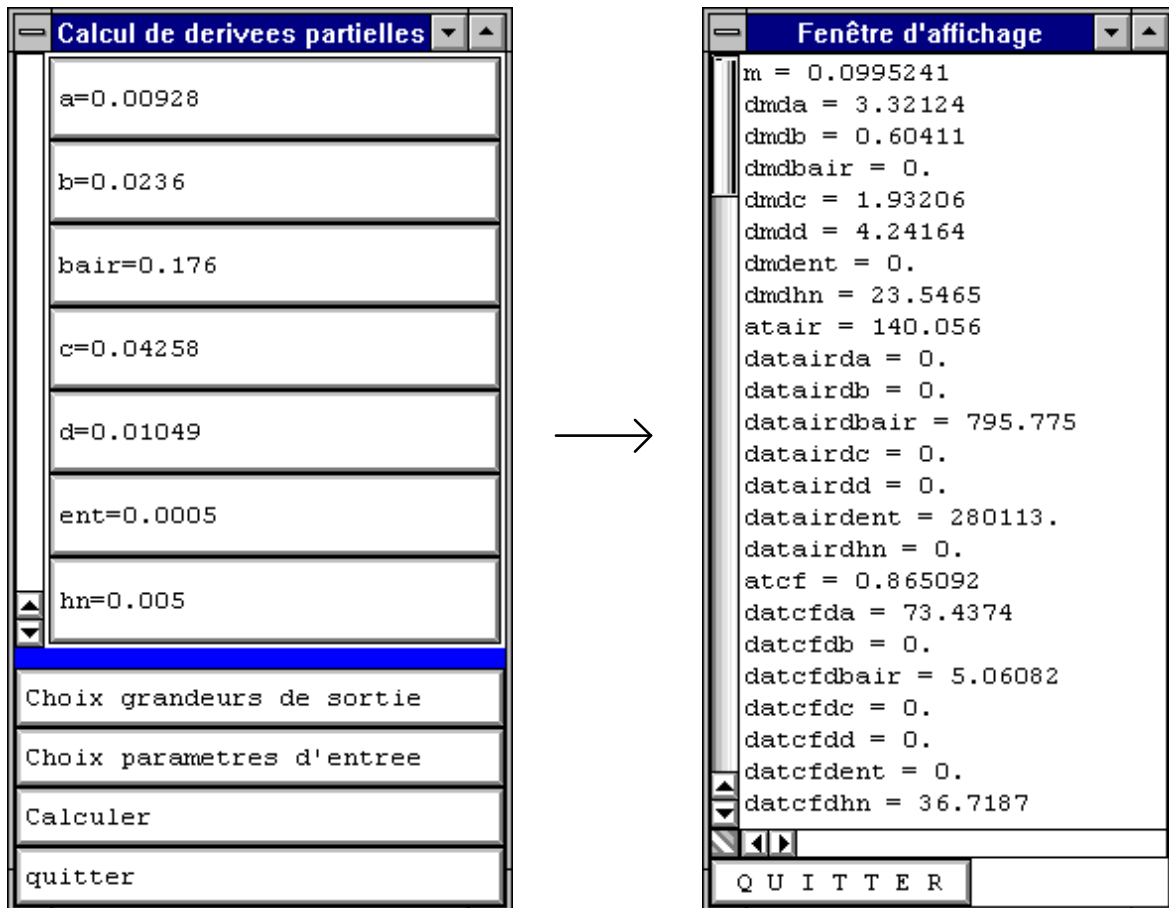


Figure 18: exemple de calcul de sensibilité (=calcul des dérivées partielles des paramètres de sortie pour un jeu de paramètres d'entrée)
(exemple de l'électroaimant)

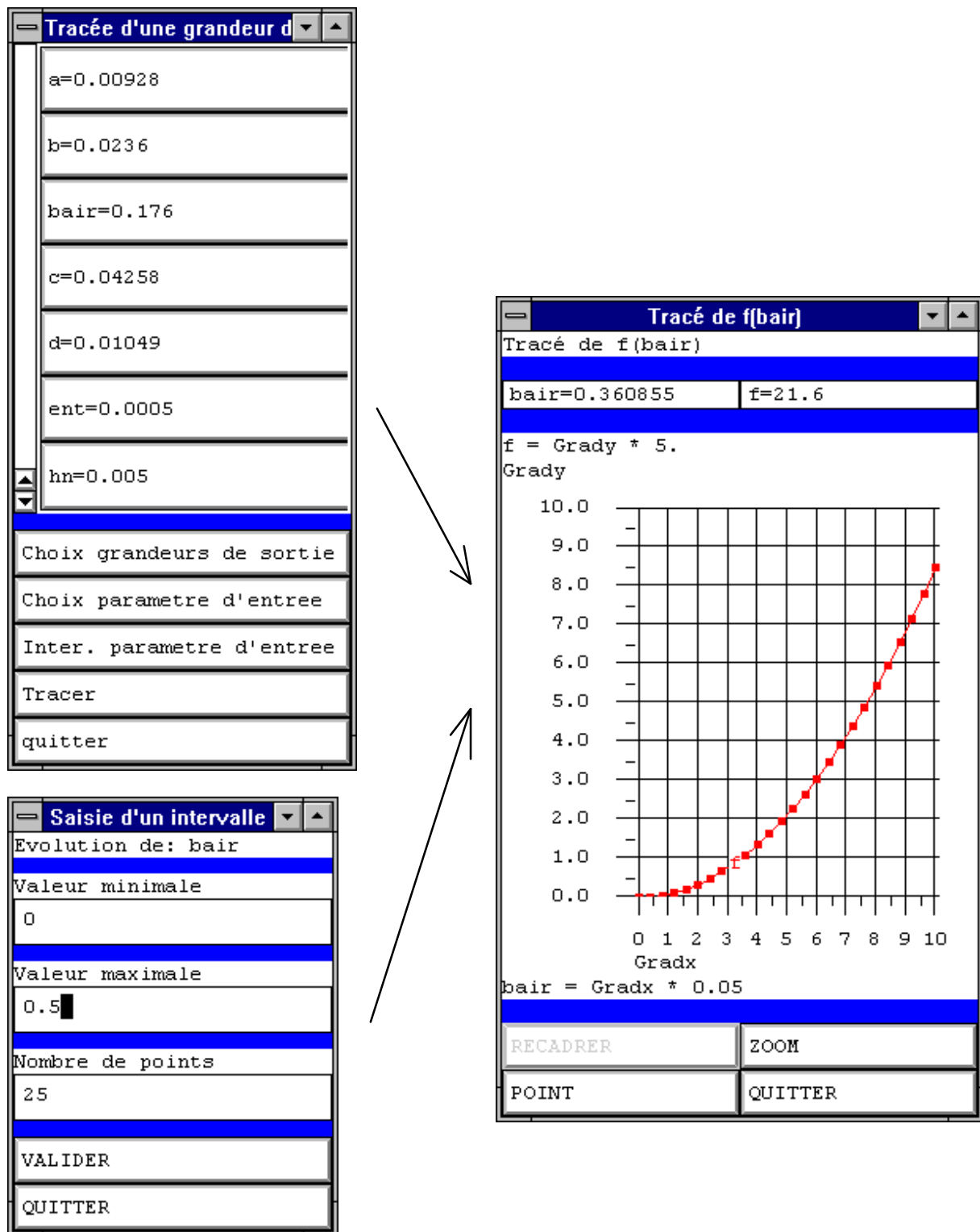


Figure 19: exemple de tracé de paramètre d'entrée en fonction de paramètre de sortie (exemple de l'électroaimant)⁹

⁹: Attention aux échelles du graphe qui est visualisé sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais doivent être multipliées par un coefficient comme cela est précisé à côté de ces échelles.

Mais le plus important reste bien sûr que grâce à l'interface, on peut *lancer des dimensionnements automatiques*.

Dès lors que ces dimensionnements ont été effectués, l'interface donne d'abord un compte-rendu du dimensionnement réalisé sous la forme de la fenêtre reproduite sur la figure 20.

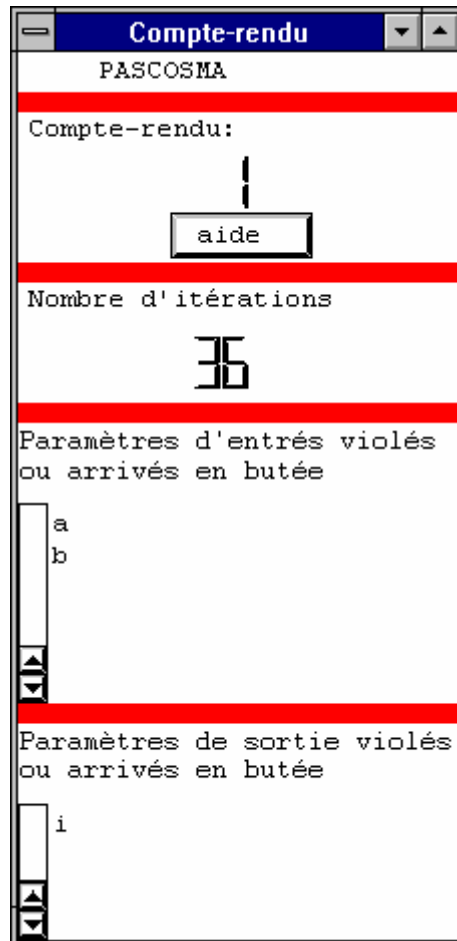
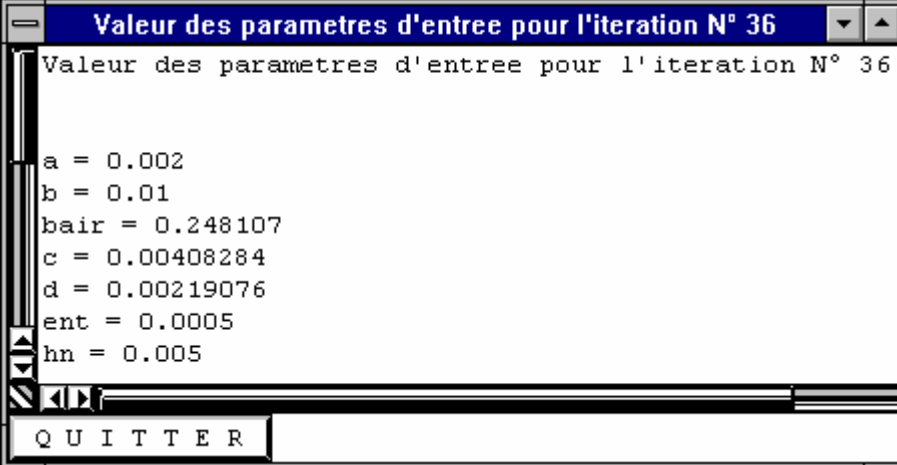


Figure 20: compte rendu de dimensionnement
(exemple de l'électroaimant)

Cette fenêtre offre les divers renseignements suivants:

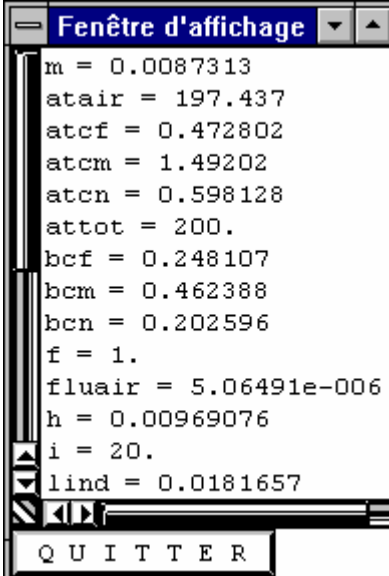
- * le résultat du dimensionnement, qui est résumé par un numéro dans la rubrique compte-rendu (*ainsi le chiffre 1 signifie qu'une solution a été trouvée avec la précision souhaitée*),
- * le nombre d'itérations que le programme a été amené à faire,
- * la liste des paramètres d'entrée et de sortie qui ont violé leurs contraintes ou atteint la limite de leurs contraintes pour l'itération finale. En l'occurrence, lorsque le compte-rendu N° 1 de succès est renvoyé, ces listes correspondent à des paramètres qui ont simplement atteint une de leurs valeurs limites.

On peut ensuite visualiser la solution finale trouvée ce qui est illustré sur la figure 21 (*cette solution correspond au cahier des charges de la figure 16*).



```
Valeur des paramètres d'entrée pour l'iteration N° 36  
  
a = 0.002  
b = 0.01  
bair = 0.248107  
c = 0.00408284  
d = 0.00219076  
ent = 0.0005  
hn = 0.005  
  
Q U I T T E R
```

Valeur des paramètres d'entrée pour l'itération finale



```
Fenêtre d'affichage  
  
m = 0.0087313  
atair = 197.437  
atcf = 0.472802  
atcm = 1.49202  
atcn = 0.598128  
attot = 200.  
bcf = 0.248107  
bcm = 0.462388  
bcn = 0.202596  
f = 1.  
fluair = 5.06491e-006  
h = 0.00969076  
i = 20.  
lind = 0.0181657  
  
Q U I T T E R
```

Valeur des paramètres de sortie pour l'itération finale

Figure 21: valeurs des paramètres d'entrée et de sortie trouvées pour le cahier des charges de la figure 16 (exemple de l'électroaimant)

En plus de la solution numérique même, l'interface permet d'étudier les évolutions des différents paramètres du problème comme::

- la fonction objectif (cf. figure 22),
- l'évolution de paramètres d'entrée tels que 'c' soumis à une contrainte d'intervalle (cf. figure 23),
- ou encore l'évolution de paramètres de sortie tels que f soumis à une contrainte d'égalité (cf. figure 23).

Ces courbes présentent de l'intérêt:

* Lorsque le système n'a pas réussi à trouver de solution car elles peuvent aider à déterminer les causes de l'échec. Ainsi, elles permettent souvent de se rendre compte que la limite d'une contrainte sur un paramètre empêche un autre paramètre d'atteindre ses propres limites et donc que ces deux contraintes sont incompatibles entre elles.

* Mais aussi lorsque le système a réussi à déterminer une solution car elles donnent les moyens d'étudier l'évolution entre la solution initiale et la solution finale. Ainsi, dans le cadre de notre exemple, on constate bien que la masse de la structure, fonction objectif à minimiser, passe de 99.5 g grammes à 8.7 grammes (cf figure 22), et que dans le même temps la structure finale respecte les contraintes imposées (cf. figure 23).

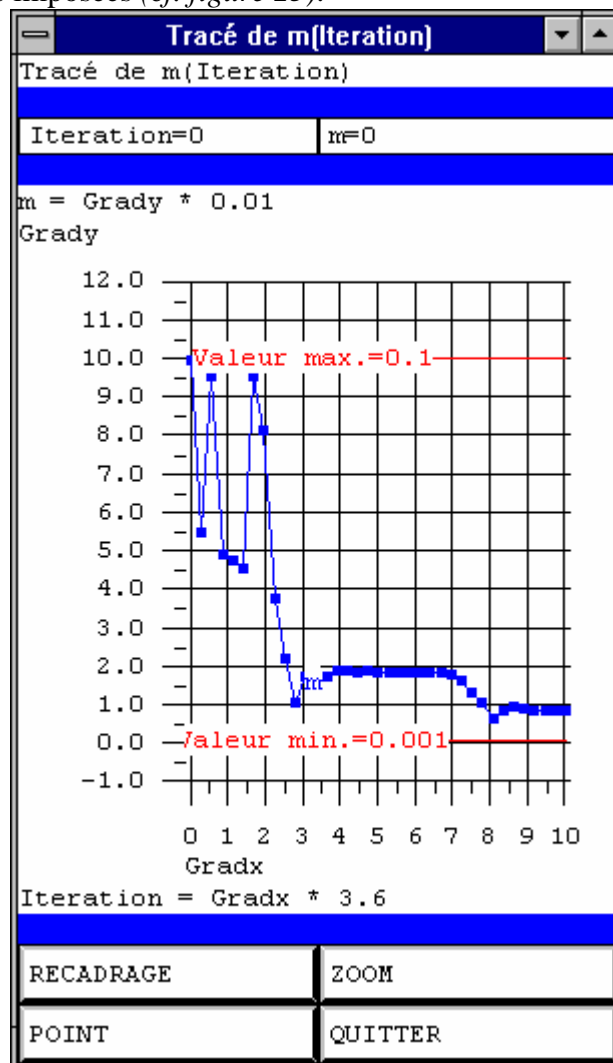


Figure 22: évolution de la fonction objectif au cours du dimensionnement (*masse de la structure - exemple de l'électroaimant*)¹⁰

¹⁰: Attention aux échelles du graphe qui est visualisé sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais doivent être multipliées par un coefficient comme cela est précisé à côté de ces échelles.

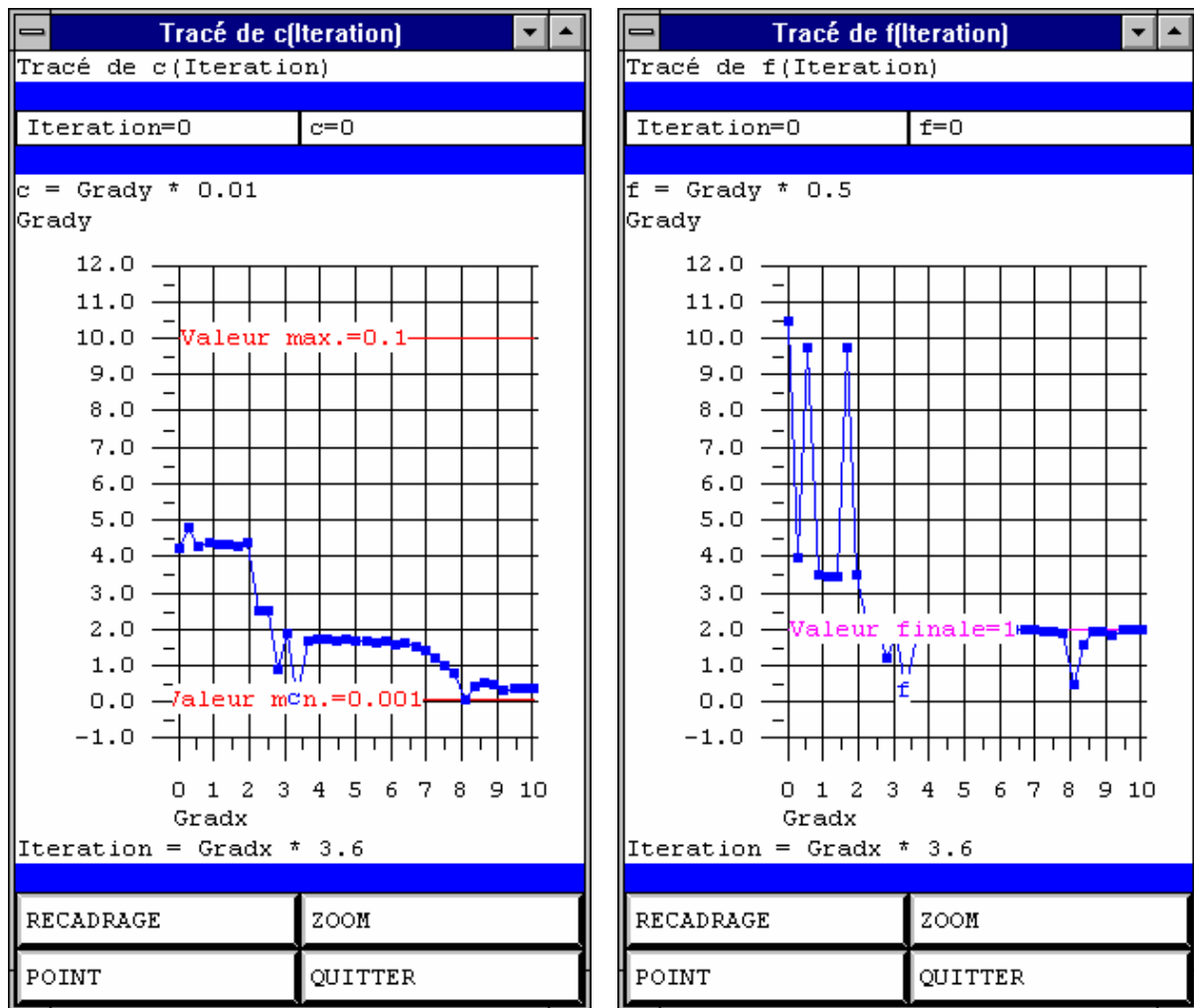


Figure 23: évolution au cours du dimensionnement du paramètre d'entrée 'c' (largeur de la dent de la culasse fixe) et du paramètre de sortie 'f' (force en N fournie) (exemple de l'électroaimant)¹¹

Pour clore ce passage en revue des possibilités du logiciel généré sur notre exemple d'électroaimant, on donnera encore, sur la figure 24, la géométrie initiale du circuit magnétique de l'électroaimant (donc le point de départ du dimensionnement) et la solution finale trouvée par le logiciel.

On constatera que la structure finale du circuit magnétique est effectivement bien moins volumineuse, donc nettement moins lourde. Le point de départ était donc relativement éloigné de la structure finale atteinte. Il faut dire que ce point de départ a été déterminé de manière aléatoire (en utilisant une fonction de génération de point de départ aléatoire accessible par l'interface graphique) ce qui explique son poids excessif et sa forme visiblement surproportionnée par rapport à la taille de la bobine.

¹¹: Attention aux échelles des graphes qui sont visualisés sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais doivent être multipliées par un coefficient comme cela est précisé à côté de ces échelles.

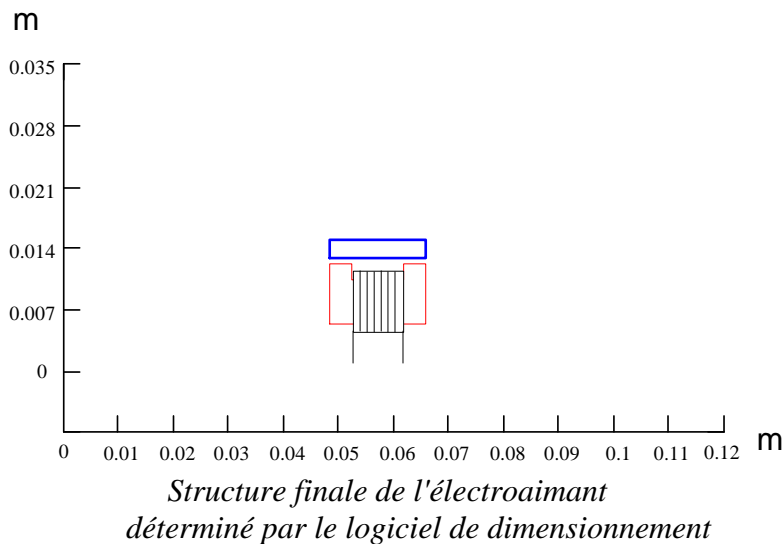
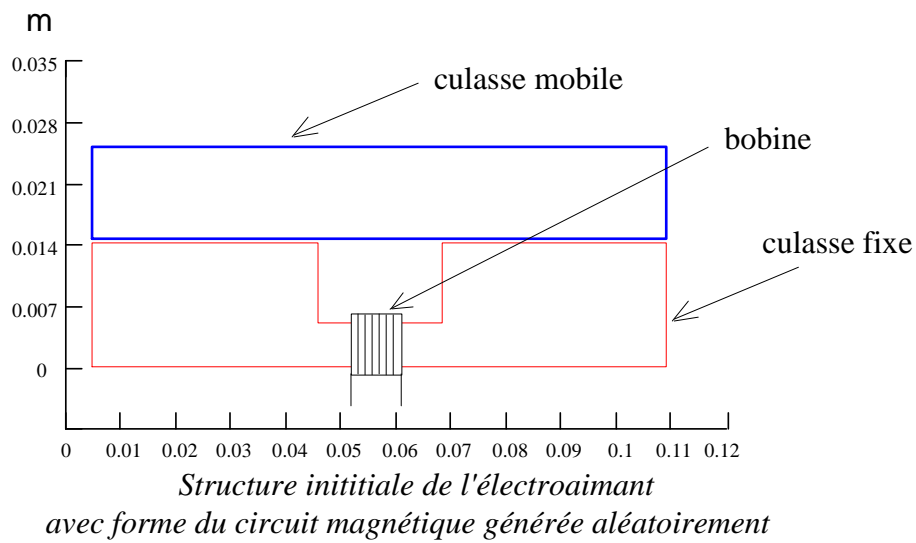


Figure 24: comparaison entre les structures initiales et finales du circuit magnétique (exemple de l'électroaimant)

III.3. Problèmes mis en évidence grâce au prototype informatique et solutions apportées

La réalisation du prototype, dont nous venons de donner un aperçu du fonctionnement, nous a aussi permis de cerner et de quantifier un certain nombre de problèmes inhérents à l'approche que nous proposons.

Les paragraphes suivants exposeront ces problèmes, et détailleront les solutions que nous avons apportées à la plupart d'entre eux.

Ces développements vont concerner deux aspects:

- 1° les risques d'explosion de la taille du code généré,
- 2° les limites du fonctionnement du mécanisme d'exploration de l'espace des solutions.

III.3.a. Le risque d'explosion de la taille du code généré: problème, et solution apportée

Le premier problème auquel nous nous sommes rapidement heurtés, est le risque d'explosion de la taille du code généré pour le programme d'analyse et pour le programme de calcul de sensibilité.

En fait, la taille du code généré dépend essentiellement de trois facteurs qui sont:

1° *Le nombre de paramètres d'entrée et de sortie du modèle.* On établit en effet que si N_{pe} est le nombre de paramètres d'entrée du modèle, et si N_{ps} est le nombre de paramètres de sortie du modèle, et si l'on considère qu'il y a encore une fonction objectif, il y aura:

N_{ps} expressions de paramètres de sortie à calculer et à programmer

$N_{ps} \times N_{pe}$ expressions de dérivées partielles de paramètres de sortie en fonction de paramètres d'entrée

1 expression pour la fonction objectif

N_{pe} expressions pour les dérivées partielles de la fonction objectif en fonction des paramètres d'entrée

Soit en tout: $N_{ps} + N_{ps} \times N_{pe} + 1 + N_{pe} = (1 + N_{pe}) \times (N_{ps} + 1)$ expressions à programmer et à calculer.

2° *La complexité des équations du modèle.*

3° *La manière de générer le code.*

Sachant que N_{ps} peut atteindre des valeurs de l'ordre de 50 à 100 pour une machine électrique, et qu'il en est de même pour N_{pe} , on constate donc que le nombre total d'expressions mathématiques symboliques à calculer peut-être de l'ordre de 2500 à 10000.

On comprend alors tout l'intérêt de notre méthodologie qui vise à automatiser l'obtention de ce grand nombre d'expressions. Mais on comprend aussi quel est le risque d'explosion de taille de l'exécutable à générer.

En pratique, pour limiter cette taille, on ne peut jouer que sur le troisième facteur, à savoir la manière de générer le code. Les deux autres facteurs sont en effet liés à la complexité du modèle structural que nous employons et que notre approche se doit de gérer pour avoir tout son intérêt.

Par ailleurs, l'expérience recueillie grâce au prototype nous a montré que ce facteur pouvait être très sensible (*au point d'aboutir à des tailles de code qui ne sont plus gérables faute de place mémoire*) et que l'approche que nous proposons pour générer le code possède intrinsèquement une source de divergence possible de la taille de celui-ci. Fort heureusement, cette divergence peut être évitée au prix de l'introduction du concept de paramètres intermédiaires.

Nous allons définir ce concept après avoir expliqué les causes de la divergence de la taille du code.

Les raisons de la divergence possible de la taille du code généré par notre approche:

Rappelons que les paramètres de sortie s'écrivent sous la forme générique:

$$C_i = f_i(P_1, P_2, P_3, \dots, P_m, C_1, C_2, \dots, C_{i-1}, C_{i+1}, \dots, C_n)^{12} \quad (3)$$

La méthodologie PASCOSMA préconise de substituer chaque C_k , $k \in [1, n]$ et $k \neq i$ par son expression symbolique, jusqu'à ce que C_i ne dépende plus que des seuls paramètres d'entrée P_j , $j=1, m$.

Or, il arrive fréquemment que ces C_k vérifient les deux propriétés suivantes:

- 1° Ils interviennent très souvent dans l'expression symbolique f_i .
- 2° Leur expression en fonction des paramètres d'entrée P_j est complexe et longue.

Il en résulte que l'expression finale de f_i peut être symboliquement très longue.

Par suite, les expressions symboliques des dérivées partielles de f_i , dont la taille augmente toujours considérablement, peuvent ne plus être calculables symboliquement, faute de place mémoire.

On notera que cette situation se produit très souvent en électrotechnique où l'on a coutume d'évaluer les performances d'une machine par l'intermédiaire de schémas électriques équivalents. Il en résulte alors qu'effectivement:

- * Les performances (*puissance, courant absorbé, ...*) s'expriment symboliquement en fonction des paramètres du schéma équivalent selon des expressions qui peuvent être très élaborées et dans lesquelles les paramètres du schéma équivalent apparaissent très

¹²: Il est interdit de rentrer une variable dépendant d'elle-même, d'où C_i ne s'exprime pas en fonction de C_i , car sinon on rentre dans le cadre de la gestion de variables implicites que nous détaillons dans le paragraphe II.4.c

souvent. Ceci est d'autant plus vrai que, lorsque l'on utilise ces modèles, les calculs se font dans l'espace des complexes, ce qui conduit toujours à des expressions symboliques très longues.

* Les paramètres du schéma équivalent s'expriment par ailleurs eux-mêmes en fonction des paramètres de construction et des conditions d'utilisation de la machine selon des expressions qui peuvent aussi être très complexes.

Dès lors, si on remplace systématiquement les paramètres du schéma équivalent dans les expressions des performances, les expressions symboliques résultantes divergent en taille

Une solution au risque de divergence de la taille du code généré - l'introduction de la notion de paramètres intermédiaires:

Dans la situation évoquée précédemment, la taille du code peut être considérablement limitée en introduisant la notion de paramètres intermédiaires.

Soit P_{int_k} , $k=1,t$ l'ensemble des paramètres de sortie qui seront gérés comme des paramètres intermédiaires.

Le principe consiste à faire les substitutions symboliques pour exprimer les paramètres de sortie C_i , $i=1,n$, non plus simplement en fonction des seuls paramètres d'entrée P_j , $j=1,m$, mais en fonction des paramètres d'entrée et des paramètres intermédiaires. On obtient ainsi des expressions du type:

$$C_i=f'_i(P_1, P_2, \dots, P_m, P_{int_1}, P_{int_2}, \dots, P_{int_t}) \quad (4)$$

au lieu des expressions:

$$C_i=f_i(P_1, P_2, \dots, P_m) \quad (5)$$

Les paramètres intermédiaires sont d'autre part exprimés symboliquement en fonction des seuls paramètres d'entrée. D'où:

$$P_{int_k}=f_k(P_1, P_2, \dots, P_m) \quad (6)$$

Pour évaluer la valeur des paramètres de sortie en fonction des paramètres d'entrée, les lignes du programme d'analyse doivent être générées pour procéder en deux temps:

1° Les valeurs numériques des paramètres intermédiaires sont calculées grâce à (6) à partir des valeurs des paramètres d'entrée.

2° Les valeurs numériques des paramètres de sortie peuvent alors calculées grâce à (4) en utilisant les valeurs numériques des paramètres d'entrée et des paramètres intermédiaires.

Lorsque les paramètres intermédiaires sont ceux évoqués précédemment, à savoir des paramètres de sortie à l'expression symbolique de grande taille, et qui interviennent souvent dans l'expression symbolique d'autres paramètres de sortie, la taille du programme pour générer les expressions (4) et (6) peut être considérablement plus faible que celle du code associé aux expressions (5).

Par ailleurs, pour assurer la génération du programme de calcul de sensibilité, le calcul symbolique est lui aussi réalisé en deux étapes:

1° A partir de (4), on calcule les expressions symboliques des dérivées partielles suivantes:

$$\frac{\partial C_i}{\partial P_j} \text{ et } \frac{\partial C_i}{\partial P_{int_k}} \text{ pour } i=1,n, j=1,m \text{ et } k=1,t \quad (7)$$

2° A partir de (6), on calcule les expressions symboliques des dérivées partielles suivantes:

$$\frac{\partial P_{int_k}}{\partial P_j} \text{ pour } k=1,t \text{ et } j=1,m \quad (8)$$

Pour évaluer la valeur des dérivées partielles des paramètres de sortie en fonction des paramètres d'entrée, soit l'ensemble des $\frac{dC_i}{dP_j}$, les lignes du programme de calcul de sensibilité

doivent être générées pour procéder en deux temps:

1° Les valeurs numériques des dérivées partielles qui se calculent en fonction des valeurs des paramètres d'entrée et des paramètres intermédiaires sont évaluées grâce aux expressions (7) et (8).

2° Pour calculer les valeurs numériques des $\frac{dC_i}{dP_j}$, on applique ensuite la formule

suivante:

$$\frac{dC_i}{dP_j} = \frac{\partial C_i}{\partial P_j} + \sum_{k=1}^t \frac{\partial C_i}{\partial P_{int_k}} \times \frac{\partial P_{int_k}}{\partial P_j} \quad (9)$$

Là encore, en employant les mêmes paramètres intermédiaires que précédemment, la taille du programme de calcul de sensibilité nécessaire pour les expressions (7) et (8) peut être considérablement plus faible que celle du programme qui code les expressions symboliques $\frac{dC_i}{dP_j}$ à partir de (5).

Forts de cet enseignement, nous avons introduit dans notre prototype la gestion automatique des paramètres intermédiaires selon les procédures que nous venons de détailler.

Ainsi, dans les instructions de génération du code, a été ajoutée la fonction:

```
ajoute_ddl_int(argument1, argument2)
```

dont le premier argument sert à désigner le paramètre de sortie qui devra être utilisé comme paramètre intermédiaire, et le deuxième argument est un symbole qui sera utilisé pour désigner ce paramètre dans le programme FORTRAN. Cette instruction doit être employée autant de fois qu'il y a de paramètres intermédiaires à désigner. Tout le reste est ensuite pris en charge de manière automatique conformément à la philosophie de notre approche.

Remarques à propos de la notion de paramètres intermédiaires:

* On notera que le choix des paramètres intermédiaires est à faire avec soin car, si leur introduction peut contribuer à diminuer sensiblement la taille et le temps de génération symbolique, elle a cependant un effet pervers: l'augmentation du nombre de routines à générer.

On peut dénombrer le nombre de routines supplémentaires nécessaires en constatant que pour un problème avec 't' paramètres intermédiaires, m paramètres d'entrée et n paramètres de sortie, il faut prévoir:

- t routines de calcul des valeurs de ces paramètres intermédiaires
 - $n \times t$ routines de calcul des dérivées partielles des paramètres de sortie en fonction des paramètres intermédiaires
 - $m \times t$ routines de calcul des dérivées partielles des paramètres intermédiaires en fonction des paramètres d'entrées
- soit en tout $t \times (m+n+1)$ routines.

Un mauvais choix de paramètres intermédiaires, est celui pour lequel le gain (*en taille et en temps de génération du code*) apporté par le fait de disposer d'expressions symboliques plus compactes sera détruit par la gestion du nombre plus grand de routines nécessaires.

A l'heure actuelle, nous n'avons pas mis au point de routine ni d'algorithme permettant de trouver automatiquement "les bons paramètres intermédiaires". C'est donc à l'utilisateur de les désigner et d'établir ainsi ce qu'on pourrait appeler une stratégie de génération de code.

* L'introduction de cette notion de paramètres intermédiaires ne remet pas en cause le principe de notre approche qui consiste à mettre le modèle analytique sous forme orienté pour pouvoir calculer les paramètres de sortie en fonction des paramètres d'entrée. Il ne s'agit en fait que d'une astuce d'ordre méthodologique pour accélérer les temps de génération de code et raccourcir la taille du code généré. Cette introduction remet d'autant moins en cause notre approche que son usage reste optionnel. Ainsi, pour le problème de l'électroaimant détaillé précédemment, nous n'avons pas utilisé de paramètres intermédiaires.

* Cette introduction peut être vue comme une solution alternative à la simplification des expressions symboliques des paramètres de sortie exprimées en fonction des seuls paramètres d'entrée. Nous ne nous sommes pas orientés vers la voie de la simplification symbolique, car nous pensons que celle-ci est beaucoup plus difficile à mettre en oeuvre de manière générique, surtout dans le domaine du génie électrique où les équations manipulées sont souvent très fortement non linéaires.

* On notera enfin que nous n'avons qu'un niveau de paramètres intermédiaires. Il s'est avéré que cela était suffisant pour les problèmes électrotechniques que nous avons traités à ce jour. Cependant, rien n'empêche l'introduction de plusieurs niveaux de paramètres intermédiaires si cela devait s'avérer nécessaire.

III.3.b. Limites du fonctionnement du logiciel de dimensionnement mises en évidence

Il était prévisible, du fait des routines d'optimisation du type gradient utilisées, que le logiciel de dimensionnement obtenu serait sensible au point de départ, à savoir que la solution vers laquelle le système peut converger peut dépendre de ce point.

Cependant, le prototype informatique réalisé nous a aussi permis de mettre en évidence d'autres limites de fonctionnement qui sont:

* *Une influence de la valeur limite des contraintes d'intervalles.* Il semble que, de la même manière que le point de départ peut influencer la solution finale trouvée par le système, des valeurs différentes de limites de contraintes d'intervalles puissent avoir le même effet. Concrètement, cela signifie que pour un paramètre Pmod du modèle devant vérifier:

$$P_{\text{modmin}} \leq P_{\text{mod}} \leq P_{\text{modmax}}$$

il peut se produire qu'en changeant la valeur supérieure de cette contrainte, le système converge vers une autre solution du problème pour laquelle la valeur finale de Pmod est pourtant comprise dans le premier intervalle posé.

De même, il peut se produire que le système trouve une solution avec un certain jeu de contraintes et qu'il ne trouve pas de solution pour un autre jeu de contraintes qui pourtant englobe le premier.

Nous n'avons pas fait une étude exhaustive de la raison de ces comportements mais ils s'expliquent vraisemblablement par le fait qu'on évolue dans des espaces non linéaires.

Dans ce cadre, le fait de changer des valeurs de limites de contraintes doit empêcher de passer par certaines zones de l'espace des solutions pouvant conduire à des solutions meilleures ou pouvant tout simplement permettre d'accéder à une solution.

Par ailleurs, n'oublions pas que dans les algorithmes d'optimisation de type gradient, les contraintes sont prises en compte lors de la détermination d'une direction de recherche¹³. Il n'est donc pas absurde qu'en changeant de limite de contrainte, la direction de recherche soit modifiée et, qu'au bout du compte, on converge vers des solutions différentes.

Ces phénomènes ont été soulignés par d'autres travaux [58]. Les auteurs recommandent, en conséquence, de poser progressivement les contraintes finales à atteindre car le fait de mettre d'emblée toutes les contraintes peut restreindre le "chemin" menant à la solution.

* *Un risque d'exploration de zones de l'espace des solutions non calculables.* Ceci signifie qu'il peut se produire que le processus de dimensionnement aboutisse à un point de l'espace des solutions qui ne soit pas calculable, tout simplement parce que l'évaluation de ce point, lors du calcul d'analyse ou du calcul de sensibilité, nécessite l'évaluation d'une racine carrée négative ou d'un argument de valeur de fonction "arcsin" supérieur à 1.

Dans ces situations, le fait de partir d'un autre point de départ permet souvent d'éviter de rencontrer à nouveau le même écueil.

Par ailleurs, l'expérience a montré que si l'on était en mesure de définir l'espace calculable par des contraintes sur des paramètres du problème, cela pouvait contribuer à éviter cette difficulté.

Ainsi, si le paramètre A du modèle est un argument de la fonction "arcsin", le fait de poser: $-1 \leq A \leq 1$ est un exemple de contrainte de ce type. Il est à noter que ceci est particulièrement efficace dans le cas où A est un paramètre dépendant linéairement des paramètres d'entrée du modèle. Ceci n'est pas étonnant dans le cas de l'algorithme que nous utilisons dans notre prototype (*voir paragraphe III.1.b.*), puisque celui-ci gère les contraintes sur les paramètres en les linéarisant.

¹³: Dans les algorithmes d'optimisation sous contraintes, les contraintes sont prises en compte grâce à une transformation de la fonction objectif. La direction de recherche étant recherchée en minimisant cette fonction objectif transformée, il semble donc plausible de penser que des contraintes différentes peuvent amener à prendre des directions de recherche différentes qui peuvent elles-mêmes conduire à des solutions différentes.

III.4.Démarche et philosophie d'utilisation de la nouvelle approche proposée pour faire du dimensionnement sous contraintes de machines électriques

III.4.a. Démarche d'utilisation optimale pour la mise au point du modèle analytique

L'usage intensif du prototype détaillé dans ce chapitre nous a permis d'avoir une idée précise de la démarche d'utilisation optimale que l'utilisateur devait avoir afin de mettre au point le modèle analytique que nous utilisons comme connaissance de base.

Il est clair que cette phase est la plus importante pour l'utilisateur qui voudra utiliser notre approche, car c'est celle qui va demander le plus d'efforts de sa part.

Celle-ci pourrait éventuellement être faite avec le générateur de code que nous proposons, sachant que celui-ci génère un programme d'analyse qui peut permettre de vérifier que le modèle est correct.

Cependant, cette démarche n'est pas optimale, car les phases de génération de code nécessaires induisent une certaine lourdeur incompatible avec un processus de mise au point de modèle qui sera nécessairement itératif.

Elle est d'autant moins optimale que de nouvelles catégories d'outils informatiques existent actuellement, qui offrent infiniment plus de souplesse pour cette phase. Il s'agit d'outils, dont MATHCAD [40] est un exemple, qui donnent la possibilité d'utiliser l'ordinateur tout à la fois comme un traitement de texte scientifique et comme une feuille de calcul. Ils permettent de saisir les équations sous la forme avec laquelle on est habitué à les écrire sur une feuille de papier, puis ils les évaluent. La figure 25 donne un aperçu du travail que l'on peut réaliser avec ce genre d'outils. Si on ajoute les possibilités extrêmement simples de tracé de graphes offertes, on dispose d'un type d'outil idéal pour la mise au point d'un modèle analytique, en étant dégagé des contraintes:

- de calculs manuels,
- de calcul à la machine à calculer,
- ou même de programmation dans un langage quelconque,

tout en pouvant analyser en temps réel le comportement du modèle qu'on est en train d'élaborer.

On peut être certain que ce nouveau type d'outils induira une nouvelle manière de travailler et constitue, de notre point de vue, le complément idéal de notre méthodologie pour la phase de mise au point des modèles analytiques qu'elle utilise.

Dès lors que le modèle est au point, notre méthodologie reprend tout son sens car les outils évoqués précédemment n'offrent pas les possibilités de dimensionnement automatique avec prise en compte des contraintes¹⁴.

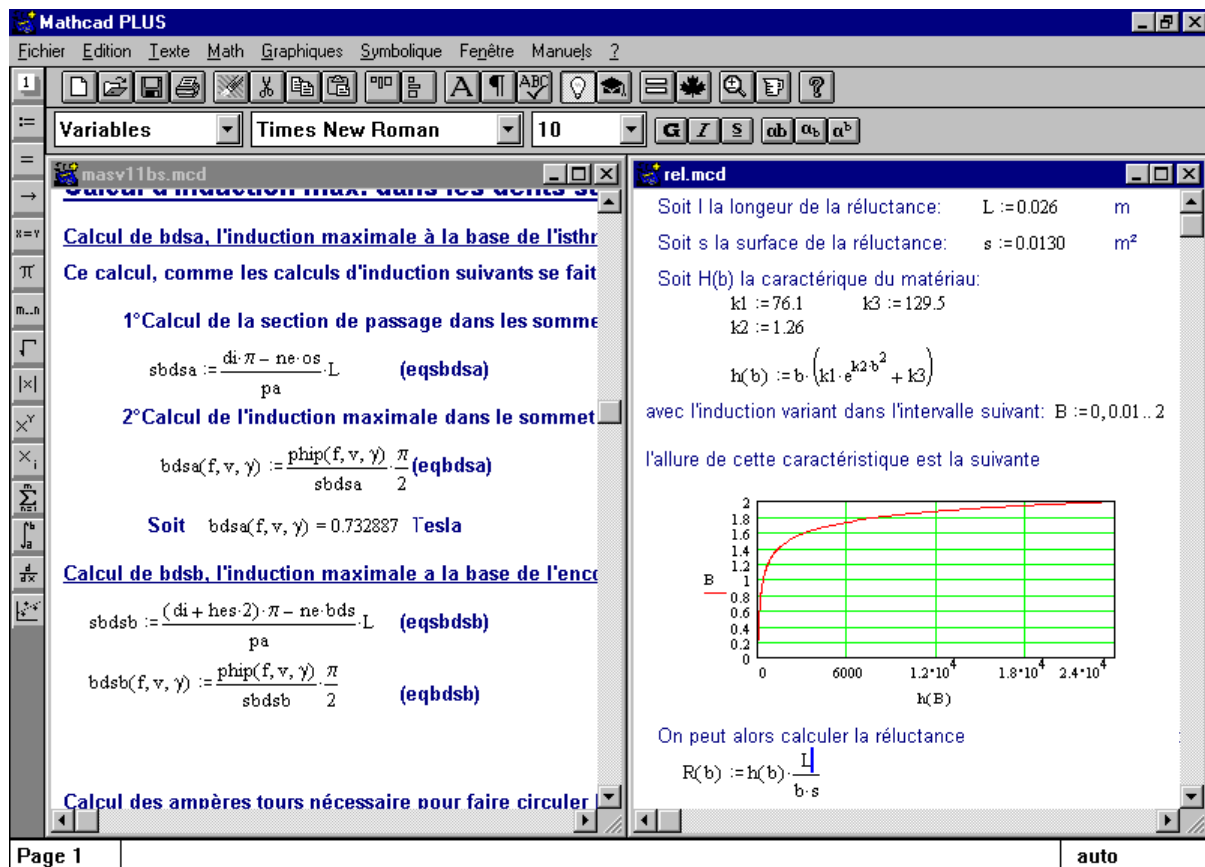


figure 25: MATHCAD, une illustration du nouveau type d'outil mis à la disposition des concepteurs permettant une mise au point de modèles analytiques

III.4.b. Philosophie d'utilisation du logiciel de dimensionnement automatiquement généré

L'expérience acquise grâce au prototype réalisé nous a aussi montré que l'usage du logiciel de dimensionnement automatiquement généré devait se faire avec une certaine philosophie d'utilisation.

Celle-ci doit intégrer le fait que, si le logiciel généré possède effectivement des capacités de dimensionnement automatique, il ne reste qu'un outil pouvant d'ailleurs être mis en défaut par

¹⁴: Précisons que ce type d'outils offre aussi la possibilité d'utiliser des algorithmes d'optimisation avec prise en compte des contraintes. Cependant, il n'est possible de traiter que des problèmes de très petite envergure et qui sont sans commune mesure avec le dimensionnement complet d'une machine.

un mauvais point de départ ou l'exploration d'un point de l'espace des solutions non calculable.

Aussi, l'ingénieur ou le chercheur utilisateur du système devra aborder l'utilisation de l'outil en étant conscient de ces limites afin d'être sensibilisé au fait que tout résultat proposé par le système nécessite une analyse critique ou un diagnostic:

* Ainsi, en cas d'échec du logiciel, il ne faut surtout pas penser que le problème est irrémédiablement sans solution. Peut-être qu'un autre point de départ ou une contrainte plus lâche permettront de converger vers la solution, à moins que des contraintes du cahier des charges ne soient effectivement incompatibles entre elles, ce qu'une analyse judicieuse des graphes d'évolution de chaque paramètre du problème doit mettre en évidence.

* De même, en cas de solution trouvée, une analyse de la part de l'utilisateur, en fonction de son expérience ou en prenant éventuellement d'autres points de départ, doit permettre de confirmer que cette solution correspond réellement à une solution intéressante.

* Des points, comme la gestion des paramètres discrets, reste à la charge de l'utilisateur: à lui de définir quand, et sur quelles valeurs, il faut figer ces paramètres discrets.

En résumé, comme cela est souligné dans [58], un programme de dimensionnement s'appuyant sur des techniques d'optimisation sous contraintes comporte trois composantes:

1° L'algorithme d'optimisation sous contraintes.

2° Les procédures d'évaluations des fonctions, soit pour nous le programme d'analyse et le programme de calcul de sensibilité.

3° ***L'utilisateur du programme d'optimisation.***

Notre approche permet de faciliter et d'automatiser l'accès aux deux premières composantes.

La troisième composante reste fort heureusement à la charge de l'utilisateur avec la philosophie d'approche nécessaire de sa part.

Conclusions du chapitre:

Dans ce chapitre, nous avons détaillé les développements et les solutions informatiques retenues pour réaliser un prototype mettant en oeuvre notre méthodologie PASCOSMA.

Ceci nous a permis d'aboutir à un prototype au fonctionnement souple, convivial et fiable, dont nous avons donné un aperçu du mode d'emploi et des possibilités.

Grâce à ce prototype, nous avons pu mettre en évidence:

- * Les problèmes que pouvait soulever l'implantation concrète de notre approche tels que les risques d'explosion de la taille du code généré. Ce problème a été résolu en introduisant le concept de paramètres intermédiaires.
- * Certaines limites du mécanisme d'exploration des solutions telle que la sensibilité à la valeur limite des contraintes.
- * La philosophie d'utilisation de notre approche à employer.

Dès lors que ces aspects d'ordre général sur la mise en oeuvre, le perfectionnement ou l'emploi de notre approche PASCOSMA et de son prototype informatique associé ont été évoqués, nous pouvons nous intéresser plus particulièrement dans le chapitre suivant à:

l'application de l'outil et de la méthodologie PASCOSMA pour la conception de machines électrotechniques.

- CHAPITRE IV -

**Applications de l'outil et de la méthodologie PASCOSMA pour la
conception en génie électrique**

IV. Applications de l'outil et de la méthodologie PASCOSMA pour la conception en génie électrique

Introduction:

Après avoir présenté dans le chapitre III des aspects relatifs au prototype informatique tels que:

- sa mise en oeuvre,
- son mode d'emploi,
- les fonctionnalités qu'il offre,
- et les enseignements d'ordre généraux qu'il nous a permis de mettre en évidence,

nous allons, dans ce chapitre, montrer comment ce prototype, et donc notre approche, a été utilisé en vue de faire du dimensionnement sous contraintes en génie électrique. Nous mettrons aussi en relief les apports que nous avons mis en évidence dans le cadre particulier de cette activité.

Dans ce but, le développement qui va suivre sera composé de trois parties:

1° Dans la première d'entre elles, nous allons indiquer comment nous avons traité *un problème aussi complexe que le dimensionnement de machines asynchrones*. Cette partie se décomposera elle-même en trois sous parties ayant comme objectifs respectifs de montrer:

- 1° *comment nous avons réalisé le modèle structural nécessaire avec les connaissances du domaine,*
- 2° *que notre approche permet de dimensionner une machine complète,*
- 3° *qu'elle peut apporter une aide très efficace dans un contexte de bureau d'étude industriel.*

2° La seconde donnera *un aperçu d'autres problèmes de dimensionnement que nous avons traités, ce qui montrera combien le champ d'application de PASCOSMA est vaste dans le génie électrique.*

3° La dernière évoquera *les opportunités nouvelles que notre approche offre en ce qui concerne l'utilisation du patrimoine de connaissances analytiques disponible en électrotechnique en vue de faire du prédimensionnement et du dimensionnement.*

IV.1. Génération et utilisation d'applications de type PASCOSMA pour le dimensionnement de machines asynchrones

IV.1.a. Objectif

L'objectif de cette première partie de chapitre est de donner un aperçu du travail que nous avons réalisé afin de:

montrer que notre approche est en mesure de prendre en charge un problème tel que le dimensionnement de machines asynchrones

Pour cela, nous détaillerons:

1° La manière dont nous avons appréhendé la réalisation du modèle structural nécessaire. Ainsi, nous évoquerons:

- * les principes de modélisation mis en oeuvre,
- * les possibilités et les performances offertes par le modèle réalisé,
- * la démarche adoptée pour la réalisation de ce modèle.

2° La réalisation de deux applications de dimensionnement de machines asynchrones.

La première démontrera que le dimensionnement complet d'une machine est possible avec notre approche en autorisant l'introduction et la variation d'un grand nombre de paramètres.

Cependant, le modèle employé n'ayant pas fait l'objet d'une confrontation avec des résultats d'essais, nous détaillerons la réalisation d'une seconde application de dimensionnement, moins ambitieuse en terme de paramétrage, mais utilisant un modèle ayant fait l'objet de comparaisons avec des résultats d'essai. Celle-ci permettra de prouver que des résultats fiables, rapides et intéressants peuvent être obtenus dans un contexte de dimensionnement industriel réel.

IV.1.b. Réalisation d'un modèle structural de machine asynchrone

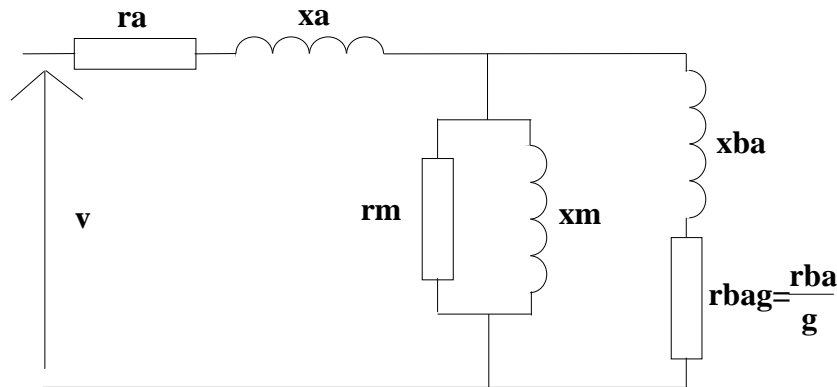
IV.1.b.1. Principes mis en oeuvre dans la réalisation du modèle structural

Les modèles structuraux de machines asynchrones utilisés (*plusieurs versions ont été définies*) ont été réalisés en utilisant une modélisation par schéma équivalent [31], [50] (*cf. figure 26*). Il en résulte que ces modèles possèdent deux niveaux:

1° Les performances de la machine sont évaluées à partir des paramètres du schéma équivalent.

2° Les paramètres du schéma équivalent sont par ailleurs calculés en fonction des paramètres de construction de la machine.

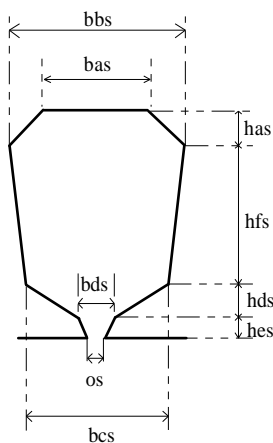
En substituant les expressions des paramètres du schéma équivalent dans les expressions calculant les performances, ils permettent ainsi de relier les paramètres de construction de la machine aux performances.



- ra: Résistance du bobinage statorique
- xa: Réactance de fuite du bobinage statorique
- rm: Résistance représentative des pertes fer
- xm: Inductance magnétisante du moteur
- xba: Réactance de fuite équivalente du rotor ramenée au stator
- rbag: Réactance équivalente du rotor ramenée au stator divisée par le glissement
- g: Glissement
- v: Tension d'alimentation par phase

Figure 26: schéma équivalent d'une phase de machine asynchrone

Une représentation du modèle structural typique de ces modèles est donnée à la figure 28 sous forme d'un arbre de dépendance condensé (*toutes les variables du modèle ne sont pas représentées, mais simplement les groupes principaux*). La dernière version de modèle que nous avons réalisée est détaillée de manière complète dans l'annexe B.



**Figure 27:
Encoche stator et
son paramétrage
utilisé dans le
modèle structural**

On indiquera encore quelques principes de modélisation qui y sont mis en oeuvre:

* *Le principe d'un paramétrage exhaustif*. Ainsi la figure 27 donne le schéma d'une encoche stator avec son paramétrage.

* *Le calcul d'un facteur de saturation*. Celui-ci a été introduit dès lors que notre modèle structural a été confronté à des résultats d'essais. Ce facteur permet de *modéliser le phénomène de saturation magnétique à vide de la tôle*.

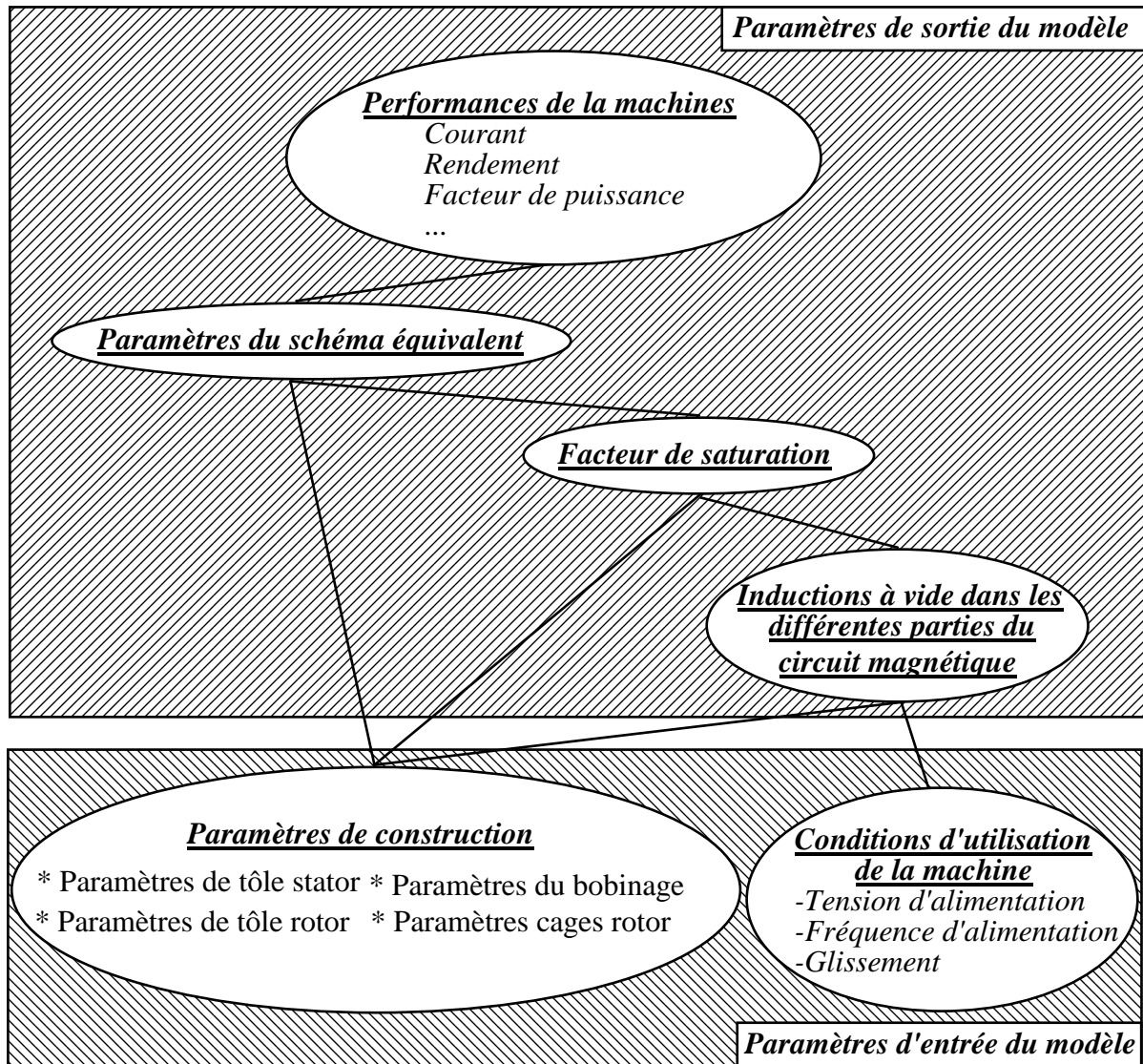


Figure 28: arbre de dépendance condensé typique des modèles structuraux utilisés pour dimensionner des machines asynchrones

* Une dépendance des paramètres du schéma équivalent envers les paramètres de construction (comme les dimensions de l'anneau de court-circuit, les dimensions des tôles magnétiques,...), mais aussi envers les conditions d'emploi de la machine (tension d'alimentation, fréquence d'alimentation, glissement,...). Ainsi, des coefficients correctifs multiplient par exemple la résistance r_{ba} et l'inductance x_{ba} du rotor pour tenir compte de l'effet de peau qui varie avec le glissement.

* Un calcul des inductions à vide dans les différentes parties de la tôle. Outre son intérêt pour calculer le facteur de saturation, le fait de disposer des inductions va

permettre de contraindre ces grandeurs afin que leurs valeurs restent raisonnables (*ainsi l'induction maximale dans les dents ne doit pas dépasser 1.8 T par exemple*).

* *Le calcul d'un coefficient d'inclinaison qui modélise les effets de l'inclinaison des barres du rotor.*

* *La variation de perméance de l'entrefer due aux encoches est prise en compte de façon moyenne en multipliant l'entrefer réel de la machine par le coefficient de Carter [3].*

IV.1.b.2. Possibilités de dimensionnement offertes par le modèle structural actuel de dimensionnement de machines asynchrones et améliorations possibles

Tous les modèles que nous avons développés ont été exclusivement utilisés afin de dimensionner des machines asynchrones dans le cadre d'un fonctionnement autour de leur point nominal (*glissement g compris entre 0 et 0.3*).

En effet, ces modèles risquent d'être insuffisants dès lors qu'on les utilisera en dehors de cette plage car certains aspects comme les inductances de fuite des têtes de bobines sont certainement très mal modélisés. Par ailleurs, d'autres phénomènes comme:

- la saturation des inductances de fuite au démarrage,
- la présence de couples harmoniques,

ne sont pas modélisés du tout. Or ces phénomènes, s'ils ne viennent pas perturber le fonctionnement du moteur autour du point nominal, peuvent devenir prépondérants lors du démarrage (*soit pour des glissements compris entre 0.5 et 1*) [63]. Enfin, notre modèle actuel ne modélise pas le comportement thermique de la machine.

Notre objectif n'était pas de faire un modèle exhaustif de la machine asynchrone mais d'utiliser notre démarche sur un exemple suffisamment complexe pour valider notre approche par rapport à certains points que nous allons détailler par la suite. Aussi, nous n'avons pas cherché à surmonter ces limites.

Cependant il est important de noter que ceci semble pouvoir être envisagé avec des modèles analytiques que notre approche saura prendre en compte. Ainsi, on se reportera:

- aux publications [4], [14] en vue de modéliser la saturation des inductances de fuite au démarrage,
- aux publications [1], [3], [16], [55] en vue de prendre en compte les couples harmoniques,
- à la publication [42] pour faire une modélisation thermique grâce à un schéma électrique équivalent,

- ou encore aux publications [60], [61] pour faire une modélisation rigoureuse de l'effet de peau en fonction de la forme de l'encoche.

IV.1.b.3. Démarche adoptée pour la mise au point du modèle structural

La démarche que nous avons adoptée pour réaliser le modèle structural a essentiellement consisté en un travail de compilation de modèles qui ont déjà pu être réalisés. Concernant la machine asynchrone, le premier schéma équivalent pour modéliser cette machine a été proposé dès 1897 [59]. Ensuite, un nombre considérable de documents ont été publiés qui prennent en compte des phénomènes de plus en plus complexes. Cependant, lors de ce travail, nous nous sommes heurtés aux difficultés suivantes:

* *La difficulté de trouver la bonne information en fonction de ce que l'on veut modéliser.* La masse d'information disponible est si grande qu'il est parfois difficile de trouver rapidement ce dont on a besoin.

* *Il n'existe pas de schéma équivalent unique pour modéliser une machine asynchrone* [45]. Ceci se traduit par le fait que, pour incorporer des phénomènes qui mathématiquement se formalisent très bien, plusieurs versions de schémas équivalents sont possibles. On peut citer en exemple la modélisation de l'inclinaison des barres du rotor [11], [30].

* *Certains phénomènes ne connaissent pas de modèle analytique correct à ce jour.* La modélisation des fuites au niveau des têtes de bobines en est un exemple. Ceci se traduit par le fait que, d'un auteur à l'autre, les formules employées peuvent varier substantiellement.

Malgré ces difficultés, nous pensons que la connaissance analytique reste l'une des plus faciles à utiliser car c'est celle qui est la plus et le mieux formalisée dans la littérature du domaine. Par ailleurs, il est souvent possible de réaliser très rapidement des modèles analytiques simples, utiles au moins dans un cadre de prédimensionnement et dont l'élaboration ne se heurtera pas aux difficultés précédentes.

IV.1.c. Description d'une première application de dimensionnement pour machines asynchrones - démonstration de la capacité à dimensionner une machine complète

Dans le cadre du dimensionnement de machines asynchrones nous avons réalisé une première application dont l'objectif était de vérifier que notre approche reste robuste face à un problème très fortement paramétré, typique du dimensionnement complet d'une machine.

Dans ce but, nous avons pris un modèle structural de machine asynchrone réalisé selon les principes énoncés précédemment. Cependant, celui-ci ne prend pas en compte la saturation à vide de la machine par l'intermédiaire du facteur de saturation. Dans l'application de dimensionnement qui a été générée sur la base de ce modèle, aucun paramètre d'entrée n'a été figé¹⁵ et un grand nombre de paramètres de sortie ont été générés. Il en résulte que la taille du code à programmer automatiquement devenait très grande de par le nombre de routines à générer.

Les caractéristiques exactes pour la génération du code de cette application étaient:

Nombre de paramètres d'entrée du modèle:	57
Nombre de paramètres de sortie du modèle:	42
Nombre de paramètres intermédiaires:	6

Ce sont les six paramètres du schéma équivalent (*voir Figure 26*) qui ont été pris comme paramètres intermédiaires, ce qui s'est avéré nécessaire pour rendre la génération du code possible.

Selon la formule de dénombrement des routines à générer, mise en évidence dans le paragraphe III.3.a., on obtient:

$$42 \times 57 + (57 + 42) \times 6 = 2988 \text{ routines}$$

Ce nombre, associé au fait que certaines routines ont des expressions très complexes, permet de réaliser l'ampleur du travail nécessaire pour obtenir une application de dimensionnement.

Notre prototype s'est parfaitement acquitté de ce travail dans un laps de temps de l'ordre de 10 heures sur un ordinateur de type PC pentium 75 Mhz. Ce temps inclut les manipulations symboliques, la génération du code en FORTRAN, la compilation et l'édition de lien.

Ce temps, qui peut a priori paraître élevé, doit être relativisé par les deux considérations suivantes:

* *Le temps de génération est un investissement*, puisqu'une fois l'application de dimensionnement générée, elle peut traiter n'importe quel cahier des charges en un temps de dimensionnement très rapide comme nous allons le voir par la suite.

* *Le temps de génération actuellement nécessaire peut encore être optimisé* dans la mesure où notre prototype actuel de générateur de code possède quelques lourdeurs (*l'emploi d'un environnement lent comme MACSYMA pour faire les opérations de calcul symbolique et de génération de code en est un exemple*).

¹⁵: figer un paramètre d'entrée 'A' signifie introduire une équation $A=\lambda$, où ' λ ' est une constante réelle dans le modèle. Le paramètre disparaît alors de la liste des paramètres d'entrée. Ceci allège le code puisqu'un symbole est substitué par une valeur numérique ce qui a notamment pour conséquence que les dérivées partielles des paramètres de sortie en fonction de ce paramètre d'entrée n'ont plus à être générées.

La taille du logiciel de dimensionnement final est de l'ordre de 3 Mo.

Cet exemple démontre la capacité de notre approche à générer un code de dimensionnement pour des problèmes très fortement paramétrés, typiques des cas où l'on souhaite dimensionner une machine complète.

L'annexe C détaille un exemple de dimensionnement réalisé avec cette application. Celui-ci montre:

* *La capacité du logiciel généré à explorer l'espace des solutions avec un grand nombre de paramètres variables.* En effet, le cahier des charges est traité autorise la variation de 16 paramètres d'entrée et fixe des contraintes pour 21 paramètres de sortie.

Par ailleurs, cet exemple montre d'autres aspects importants en vue de l'utilisation de notre approche dans une perspective de dimensionnement largement automatisée en électrotechnique. Ainsi, sont mises en évidence:

* *La rapidité avec laquelle le cahier des charges est traité,* puisque la solution est fournie en 3,75 s.

* *La capacité du logiciel à trouver automatiquement ce qu'il faut faire en fonction du cahier des charges qui est posé.* En effet, on y montre qu'au fur et à mesure des itérations, le système prend des initiatives qu'aurait prises un homme de l'art. Ainsi le système s'est orienté vers une machine à l'entrefer minimum dans l'intervalle autorisé: ce choix est cohérent avec la pratique d'un expert électrotechnicien qui s'oriente toujours vers la valeur d'entrefer minimale autorisée car:

- cela contribue à diminuer le courant nécessaire pour magnétiser la machine,
- par suite cela entraîne une diminution des pertes Joule statoriques,
- ce qui permet d'améliorer le rendement qui était en l'occurrence fortement contraint.

* *La gestion de l'interdépendance des grandeurs les unes avec les autres.* Ainsi, il est demandé de minimiser le coefficient de remplissage des encoches tout en restant dans un intervalle raisonnable (*afin que la machine obtenue soit rapide à bobiner*). Le système a bien minimisé ce coefficient de remplissage, mais il n'a pas atteint la limite minimale autorisée car:

- une valeur trop faible aurait diminué la section de cuivre disponible pour faire passer le courant,
- ce qui aurait contribué à faire augmenter les pertes Joule,
- dégradant ainsi le rendement,
- qui n'aurait plus été dans l'intervalle autorisé.

Concernant les deux derniers points, il est important de réaliser que:

- * Le logiciel a effectué un processus d'ordre mathématique qui relève de l'algorithme d'optimisation qu'il utilise et qui n'est pas un raisonnement d'ordre électrotechnique. Il n'en reste pas moins que, comme nous venons de le montrer, chaque résultat s'explique et aurait pu être obtenu par des raisonnements tenus par un homme de l'art.
- * Cependant, pour arriver à ces résultats, nous n'avons eu besoin, ni de coder un algorithme, ni de rentrer des règles pour indiquer "ce qu'il faut faire" où "comment gérer l'interdépendance des phénomènes": le savoir faire nécessaire est retrouvé implicitement et automatiquement à partir de la seule donnée du modèle structural.

IV.1.d. Description d'une seconde application de dimensionnement pour machines asynchrones - démonstration de la capacité à fournir des résultats exploitables dans un bureau d'étude

L'application précédente a permis de démontrer que l'approche que nous proposons est en mesure de prendre en charge et de résoudre des problèmes représentatifs du dimensionnement complet d'une machine. Cependant, le modèle employé restait théorique car non confronté à des résultats d'essais.

C'est pourquoi au cours du travail de thèse, nous avons réalisé une application de dimensionnement, au paramétrage plus modeste, mais ayant la vocation d'être utilisée dans un bureau d'études industriel afin de fournir rapidement des dimensionnements optimaux et fiables. Cette application a donc été réalisée en ayant à l'esprit les deux préoccupations suivantes:

- 1° *La fiabilisation du modèle,*
- 2° *La comparaison des dimensionnements proposés par notre logiciel avec des dimensionnements concrets.*

Nous nous sommes intéressés au *dimensionnement de machines appartenant toutes à une même gamme*. Il s'agit de machines asynchrones, de hauteur d'axe fixe, de petite puissance (*de 1000 à 5000 W environ*) et utilisées dans des machines de découpe du bois.

Dans l'ensemble de ces machines, beaucoup de paramètres tels que la géométrie de la tôle et de la carcasse restent figés. Seuls changent des paramètres comme le bobinage ou la longueur de fer du circuit magnétique.

C'est pourquoi nous avons fiabilisé le modèle analytique grâce à une machine de référence de la gamme.

Celle-ci a fait l'objet d'une étude importante consistant à définir:

-les capteurs dont elle devait être équipée, tels que:

- une génératrice tachymétrique,
- les sondes thermiques pour lesquelles nous avons dû définir le nombre et la localisation. Ainsi, 18 sondes ont été implantées dans des endroits aussi divers que:

- les têtes de bobines,
- la culasse stator,
- une dent stator,
- le milieu de l'arbre rotor,
- ...

-l'ensemble des essais dont elle devait faire l'objet (*à vide, en transitoires, en court-circuit, en régime permanent avec différents points nominaux,...*)

Les solutions techniques quant à la réalisation de cette machine ont été définies avec le bureau d'étude du groupe Radio-Energie. La machine a été fabriquée dans les ateliers de cette société.

Elle a alors fait l'objet des essais définis. Leur exploitation a permis:

- * D'identifier certains paramètres, tels qu'un coefficient de frottement visqueux ou des valeurs de pertes fer que l'on fait évoluer par des lois d'extrapolation.
- * De vérifier que le modèle structural employé est suffisamment fiable pour dimensionner ces machines pour leur point de fonctionnement nominal. Dans celui-ci, le calcul du coefficient de saturation a été introduit. On obtient ainsi exactement le modèle structural dont l'arbre de dépendance condensé est donné sur la figure 28 et dont tous les détails sont fournis dans l'annexe B.
- * D'établir des valeurs limites de pertes autorisées dans ces machines pour ne pas dépasser les échauffements maximaux autorisés (*en attendant de mettre au point un modèle thermique plus élaboré*).

L'annexe D donne des informations plus précises sur ces points.

Elle décrit aussi:

- * Les caractéristiques exactes de l'application générée sur la base du modèle,
- * Deux cahiers des charges qui ont été résolus avec cette application:

** Le premier a pour objectif d'optimiser le fonctionnement du prototype instrumenté en cherchant les conditions de fonctionnement dans lesquelles il va fournir le maximum de puissance en régime nominal sans dommage.*

On y montre que le logiciel trouve pratiquement instantanément un point de fonctionnement, fournissant 75% de puissance de plus que la puissance nominale que le constructeur avait initialement prévue, tout en restant dans les limites d'échauffement autorisées. Il est vrai que le constructeur n'avait pas réalisé d'optimisation particulière de cette machine. Notre application a cependant montré qu'elle est l'outil adéquat pour réaliser cette optimisation car le point de fonctionnement trouvé a fait l'objet d'une vérification expérimentale confirmant le résultat fourni (*cela est détaillé précisément dans l'annexe D*).

** Le second cahier des charges a eu pour vocation de dimensionner une machine de même puissance que celle qui avait été annoncée pour le prototype. Le logiciel trouve en quelques secondes une machine nécessitant 50% de cuivre et de fer en moins. Là encore, tous les détails sont disponibles dans l'annexe D.*

Cette étude avait pour objectif d'ancrer notre approche dans la réalité du dimensionnement qu'on peut rencontrer dans un bureau d'études et dans la réalité du fonctionnement d'une machine asynchrone. La conclusion est que, dans ce contexte, le type de logiciel de dimensionnement que nous proposons est en mesure de fournir des machines réalisables et optimales en des temps rapides.

Par ailleurs, on insistera aussi sur le fait que cette étude reste assurément timide par rapport aux perspectives que l'on peut entrevoir. En effet, la réalisation d'un modèle structural plus élaboré, modélisant des phénomènes électromagnétiques complexes (*couples harmoniques, saturation des inductances de fuites,...*) et modélisant parfaitement le comportement thermique devrait permettre de dimensionner des machines:

- en intégrant des contraintes sur toute la plage de fonctionnement,
- en faisant varier fidèlement un grand nombre de paramètres.

IV. 2 Autres applications réalisées

Indépendamment du travail mené autour de la machine asynchrone, nous avons effectué des études ayant abouti à la réalisation d'autres applications de dimensionnement.

Ainsi, des applications de type PASCOSMA ont permis de dimensionner ou de prédimensionner des dispositifs électriques aussi divers que:

- des machines synchrones à aimants permanents de structure classique,

- des machines synchrones à FEM trapézoïdales,
- des machines synchrones sans encoches [70],
- des actionneurs linéaires,
- des convertisseurs statiques [9],
- des disjoncteurs différentiels.

La diversité des dispositifs abordés confirme la généralité de notre approche et son intérêt pour le génie électrique. Dans tous les cas:

- * les temps de mise en oeuvre de l'application de dimensionnement ont été considérablement accélérés,
- * les temps de dimensionnement ont été tout aussi considérablement diminués,
- * des solutions plus optimales ont été obtenues, que celles déterminées sans notre approche et le prototype qui la met en oeuvre.

IV.3. L'approche PASCOSMA: de nouvelles opportunités pour employer le patrimoine de modèles analytiques disponibles en électrotechnique

Indépendamment des avantages de l'approche PASCOSMA que nous avons déjà pu mettre en évidence (*rapidité de mise en oeuvre, gestion de l'interdépendance des phénomènes physiques, gestion des contraintes du cahier des charges, capacité à trouver automatiquement "ce qu'il faut faire",...*), nous pensons que celle-ci offre réellement de nouvelles opportunités pour l'activité de dimensionnement et de prédimensionnement dans le génie électriques. Ces opportunités sont les suivantes:

1° *L'emploi du patrimoine important de modèles analytiques disponibles dans ce domaine, issu de plus d'un siècle de recherches. Ceci est d'autant plus crucial que, faute d'outil et de méthodologie pour les mettre en oeuvre facilement, ces modèles avaient peut-être tendance à être abandonnés au profit d'outils d'analyse numériques plus faciles d'emploi. Or, même dans les cas où ils sont moins fins, ces modèles analytiques vont pouvoir être utilisés plus rapidement et plus efficacement que jamais pour accélérer, si ce n'est les phases de dimensionnement, au moins les phases de prédimensionnement.*

2° *La manipulation de ces modèles sans la nécessité de faire des approximations, et ceci en allant jusqu'à offrir un calcul de sensibilité rigoureux. On évite ainsi:*

- les simplifications souvent indispensables qui avaient pour conséquence de limiter leur domaine de validité et l'intérêt de ces modèles.

- la nécessité d'employer des abaques qui synthétisent le comportement de modèles et d'équations analytiques lourds et difficiles à manipuler.

Conclusions du chapitre:

Ce chapitre a synthétisé les aspects de l'emploi de notre approche PASCOSMA pour le dimensionnement et le prédimensionnement dans le domaine du génie électrique.

Ainsi, la manière de réaliser des modèles structuraux pour dimensionner des machines asynchrones a été évoquée. Nous avons détaillé des études montrant que l'approche proposée est en mesure:

- d'assurer le dimensionnement complet d'une machine,
- de fournir des solutions performantes et exploitables dans un bureau d'études.

Nous avons aussi brièvement évoqué la diversité des applications de dimensionnement que nous avons réalisées à ce jour et dont le nombre témoigne de l'importance du champ d'application de ce que nous proposons.

Enfin, nous avons mis en lumière les opportunités nouvelles que nous offrons en terme d'utilisation du savoir analytique disponible en électrotechnique en vue de son emploi pour le prédimensionnement et le dimensionnement.

Ainsi, outre l'originalité de l'approche et sa mise en oeuvre, notre travail aura permis de dégager son intérêt pour le génie électrique.

Perspectives et conclusions

Perspectives

Grâce au travail présenté dans ce mémoire, une nouvelle approche permettant de disposer facilement d'un outil d'aide active au dimensionnement des dispositifs électrotechniques a été définie, mise en oeuvre et validée.

Bien que les résultats obtenus et les propriétés mises en évidence soient à ce jour très satisfaisants, nous pensons qu'un grand nombre de voies de recherche peuvent être explorées afin d'améliorer et de développer le nouveau concept d'outil proposé.

Ces voies de recherche se scindent en quatre catégories:

1° Les recherches visant à ***améliorer et à perfectionner les mécanismes employés dans le fonctionnement de l'approche PASCOSMA.***

Dans ce cadre, on pourra envisager:

- a) *Une amélioration du processus de génération de code.* Ainsi, il serait intéressant de disposer d'algorithmes génériques de détermination automatique des "bons" paramètres intermédiaires.
- b) *Une amélioration du processus d'exploration de l'espace des solutions.* Ceci pourra consister en la recherche et la mise au point de processus surpassant les limites actuelles telles que:
 - la non prise en compte des paramètres discrets,
 - la sensibilité aux valeurs du point de départ ou aux valeurs limites des contraintes.

Dans cet objectif, on a déjà montré dans un travail de DEA encadré au cours de la thèse [52], [71] que les algorithmes génétiques [27], [64], [65], avaient des capacités très intéressantes, dont la prise en charge des paramètres discrets ou la capacité à ne pas se laisser piéger par des optimums locaux. Aussi, leur emploi, par couplage avec des algorithmes du type gradient, semble pertinent.

c) *Une mise au point de techniques d'aide au diagnostic* lorsque le système n'a pas trouvé de solution ce qui pour le moment n'existe pas.

2° Les recherches visant à ***développer les aspects méthodologiques permettant un couplage avec d'autres types de logiciels employés dans un cadre de conception.***

Dans ce cadre, on pourra chercher à définir les moyens permettant:

- a) d'utiliser de manière souple des outils graphiques de visualisation,

- b) de réaliser des couplages avec des bases de données allant directement chercher les points de départ possibles ou les limites de contraintes en fonction des pièces disponibles dans un stock,
- c) de réaliser des couplages avec des outils d'analyse numérique fine s'appuyant par exemple sur la méthode des éléments finis dans un but de simulation précise des solutions déterminées.

3° Les recherches *électrotechniques visant à développer tous les aspects relatifs à l'utilisation de modèles analytiques.*

Ainsi:

- a) On pourra chercher à réaliser un travail rigoureux de recensement et de classement de la connaissance analytique disponible dont les objectifs seraient notamment:
 - un recensement des modèles les plus adaptés à une utilisation avec notre approche,
 - un classement de ces modèles par type de machine que l'on souhaite dimensionner (*machine synchrone, machine asynchrone, ...*).
- b) Il serait intéressant de disposer par ailleurs d'outils méthodologiques et informatiques d'aide à la gestion de cette masse de connaissances électrotechnique en vue de faciliter la recherche des modèles analytiques existants en fonction de critères tels que:
 - la machine que l'on souhaite dimensionner,
 - les phénomènes physiques qu'on veut y modéliser.
- c) Enfin, il serait avantageux de formaliser selon quels mécanismes des couplages peuvent être définis entre les modèles analytiques et les outils d'analyse numérique fine. Ces derniers pourront être utilisés notamment dans une perspective de mise au point de ces modèles analytiques en permettant des interpolation de formules analytiques.

4° Les *recherches sur le plan de la problématique de la résolution du problème de conception par ordinateur.*

Nous avons déjà eu l'occasion d'évoquer dans le premier chapitre que notre approche n'était pas en mesure de couvrir tous les aspects de la conception.

Ainsi, notre approche n'est peu, ou pas en mesure de faire:

- le choix de la structure de l'objet à concevoir,
- le premier dimensionnement de cette structure (= *le choix du point de départ*)

Aussi faudra-t-il étudier quelles sont les voies de recherche possibles afin de pouvoir assurer ces fonctionnalités tout en continuant à employer notre approche, qui, par ailleurs, sait fort bien s'acquitter des autres aspects de la problématique de la conception que sont:

- le dimensionnement correctif (*= action de trouver un dimensionnement vérifiant toutes les contraintes du cahier des charges*),
- et l'optimisation.

Ceci passera certainement par un couplage avec des techniques telles que les systèmes experts (*pour le problème de choix de structure*) ou les algorithmes génétiques (*pour le problème du choix du point de départ*) [27], [52], [64], [65].

Conclusions

Nous avons défini une nouvelle approche de la conception, et plus précisément du dimensionnement sous contraintes de machines électriques. Celle-ci a été formalisée dans tous ses aspects; puis implantée et son intérêt ainsi que ses possibilités pour le dimensionnement de machines électriques ont été démontrés et validés.

L'une des caractéristiques les plus importantes de cette approche est de surpasser les limitations que l'on rencontre habituellement au développement d'outils ayant une réelle faculté de conception: le temps de développement et de maintenance de ces outils.

Ces difficultés ont en effet été surpassées en mettant au point une approche qui assure la génération automatique du logiciel de dimensionnement à partir d'un simple modèle analytique de la machine à concevoir.

Grâce à cette souplesse d'obtention du logiciel, mais grâce aussi aux possibilités:

- de traitement de tous les cahiers des charges possibles pour une structure de machines,
- de prise en compte des contraintes du cahier des charges,
- d'optimisation,
- d'utilisation de modèles analytiques sans avoir à faire de simplifications,
- d'obtention de valeurs de dérivées partielles rigoureusement justes,
- de gestion de l'interdépendance des grandeurs intervenant dans une machine,
- de dimensionnement d'une machine complète,
- d'obtention de résultats fiables et utilisables dans un bureau d'études,

dont nous avons donné un aperçu dans ce mémoire, on peut penser que l'approche a les moyens de ses ambitions:

Offrir une voie pour changer la manière de prédimensionner et de dimensionner les machines en électrotechnique.

L'électrotechnicien pourra ainsi profiter de la puissance des ordinateurs afin d'avoir une aide dans les phases de conception, tels que le dimensionnement, et non plus simplement dans la phase d'analyse.

Ces ambitions sont d'autant plus réalistes que l'approche proposée est particulièrement pertinente en électrotechnique: elle permet en effet d'utiliser de manière particulièrement efficace le patrimoine de modèles analytiques, issu d'un siècle de recherche, dont on dispose dans ce domaine.

**ANNEXE A: modèle structural permettant de dimensionner un
électroaimant**

ANNEXE A: modèle structural permettant de dimensionner un électroaimant

La méthodologie PASCOSMA exposée dans ce mémoire offre une nouvelle approche pour faire de la conception sous contraintes de machines électriques.

Celle-ci comporte 2 niveaux. D'abord, elle définit comment générer automatiquement les éléments fondamentaux d'un logiciel de conception sous contraintes. Puis, elle fixe un cadre et les mécanismes permettant d'utiliser ce logiciel de manière conviviale afin de réaliser des *calculs d'analyse*, des *calculs de sensibilité* mais surtout un *dimensionnement automatique* sous contraintes.

Afin d'expliquer cette méthodologie nous nous sommes, tout au long de ce mémoire, appuyés sur un exemple de dimensionnement électroaimant.

Cette annexe a pour but d'indiquer quels objectifs de dimensionnement on veut atteindre et quel est, pour cela, le modèle structural à employer.

Objectifs de dimensionnement:

Concrètement, disposant d'une bobine électrique ayant la géométrie détaillée sur la figure A.1, on veut dimensionner le circuit magnétique de électroaimant représenté sur la figure A.2.

Un exemple typique pourrait être le suivant:

Dimensionner électroaimant pour qu'il soit le plus léger possible tout en fournissant une force d'appel de 1 N. La valeur d'entrefer entre la culasse mobile et la culasse fixe est figée à 0,005 mm. Les paramètres qui décrivent la structure du circuit magnétique doivent respecter un certain nombre de contraintes. Ces dernières visent:

- à limiter l'encombrement maximal final de électroaimant,
- à faire en sorte que la bobine puisse effectivement être insérée sur le noyau magnétique.

Par ailleurs, la valeur maximale du courant alimentant la bobine est limitée.

Le cahier des charges correspondant sera le suivant:

Trouver des valeurs de a, b, c, d et h_n telles que:

$$f=1 \text{ N}$$

et $e_n=0,0005 \text{ m}$

$$0,002 \text{ m} \leq a \leq 0,01 \text{ m}$$

$$0,01 \text{ m} \leq b \leq 0,05 \text{ m}$$

$$0,001 \text{ m} \leq c \leq 0,1 \text{ m}$$

$$0,0005 \text{ m} \leq d \leq 0,5 \text{ m}$$

$$0,008 \text{ m} \leq h \leq 0,2 \text{ m}$$

$$h_n = 0,005 \text{ m}$$

$$0,01 \text{ m} \leq l_{ind} \leq 0,5 \text{ m}$$

Contrainte sur le courant: $1 \text{ A} \leq i \leq 20 \text{ A}$

Contrainte magnétique: $0,1 \text{ T} \leq b_{air} \leq 2 \text{ T}$

La fonction objectif à minimiser est constituée par la masse m de la structure.

C'est exactement le cahier des charges que nous avons traité en guise d'exemple dans le chapitre III.

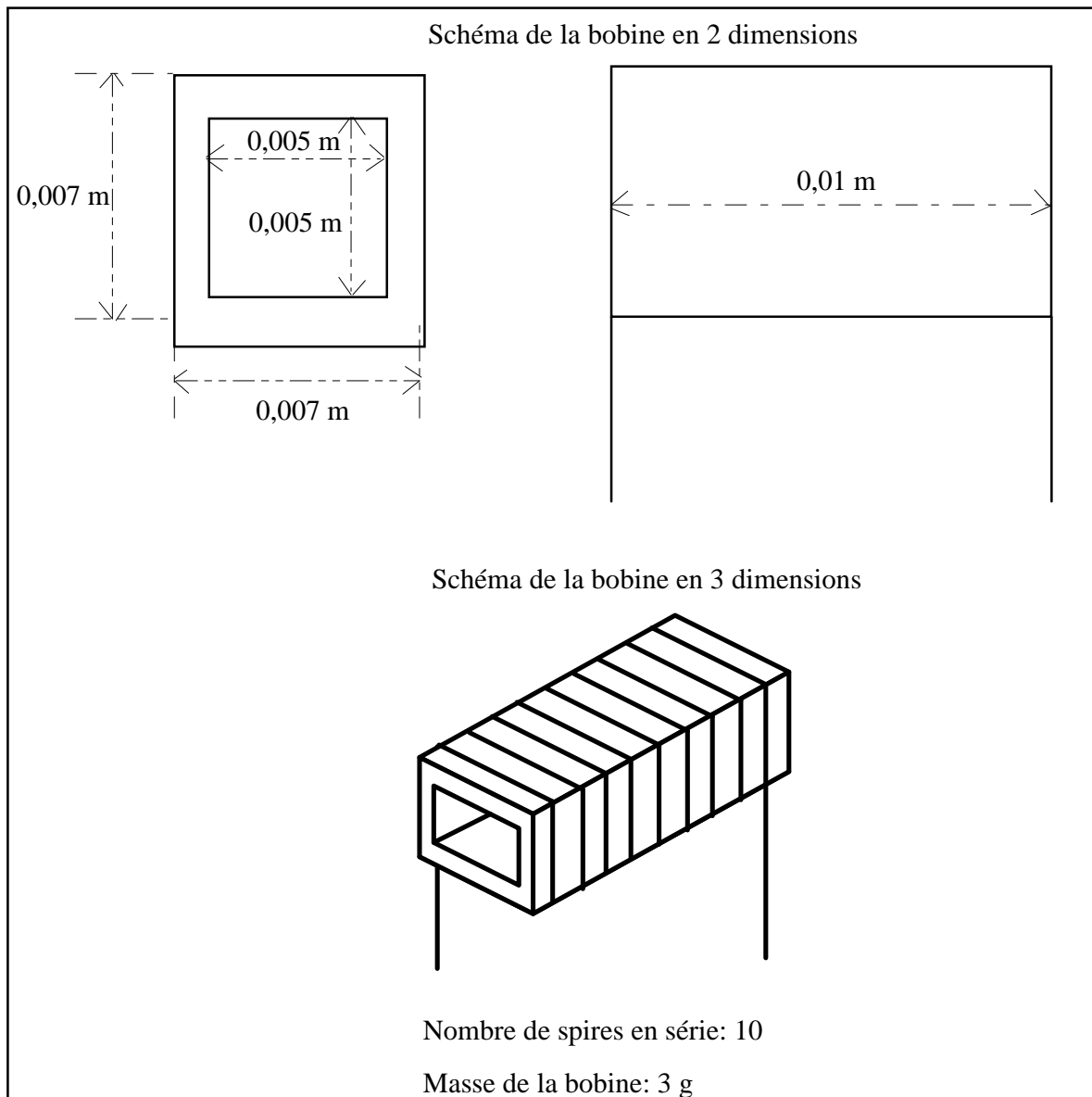


Figure A.1: schéma de la bobine utilisée pour faire un électroaimant

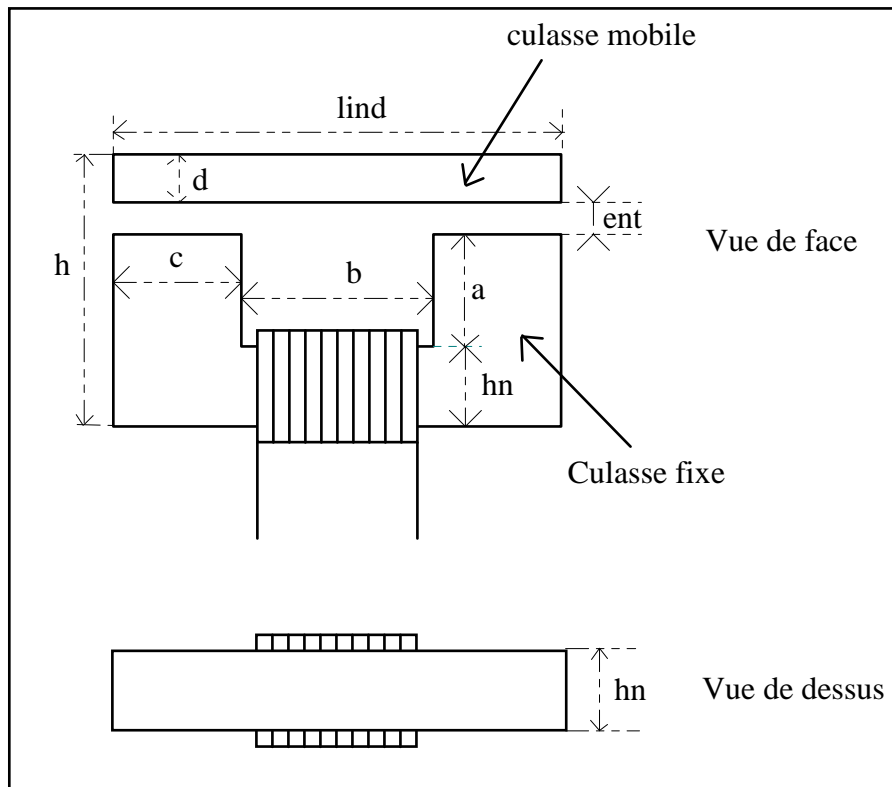


Figure A.2: Schéma de l'électroaimant avec sa bobine

Modèle structural employé:

Pour que le système soit en mesure de trouver une solution réaliste, les inductions maximales dans les différentes parties du circuit magnétique sont calculées par des lois de conservation de flux (*le trajet du flux est idéalisé: on suppose qu'il est normal aux sections de fer qu'il traverse. Dans l'entrefer, le flux est supposé perpendiculaire à la section de la dent dont il sort et sa section de passage est supposée constante dans l'air. Les fuites sont négligées*). Les ampères-tours consommés dans le fer sont calculés en utilisant la fonction suivante donnant le comportement magnétique du utilisé pour le noyau:

soient H la valeur maximale du champ magnétique et B la valeur de l'induction magnétique, on a:

$$H(B) = B \times (k_a \times e^{k_b \times B^2} + k_c) \text{ avec } k_a = 76,1, k_b = 1,26 \text{ et } k_c = 129,5$$

ce qui correspond à l'allure de la courbe donnée sur la figure A.3.

Le dictionnaire des variables employées dans le modèle structural est donné sur la figure A.4.

La figure A.5 donne enfin le détail des équations utilisées.

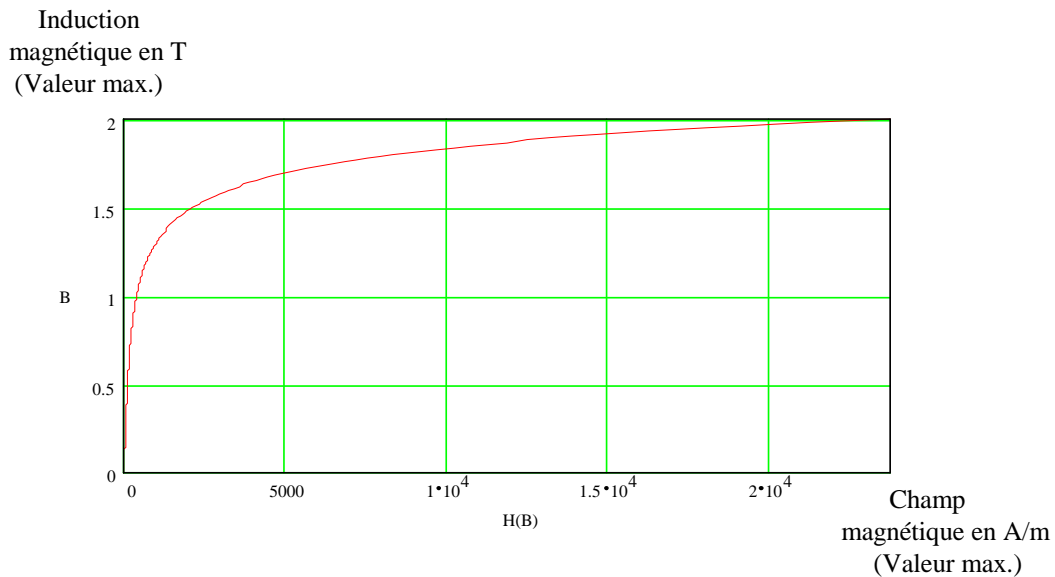


Figure A.3: courbe "B(H)" du matériau utilisé pour construire l'électroaimant.

a: Hauteur de la dent de la culasse fixe, voir figure A.2
 atair: Ampères tours consommés par la circulation du flux dans l'air
 atcf: Ampères tours consommés dans les dents de la culasse fixe
 atcm: Ampères tours consommés dans la culasse mobile
 atcn: Ampères tours consommés dans le noyau de la culasse fixe
 attot: Ampères tours totaux consommés sur la ligne de circulation de flux moyenne
 b: Largeur de l'entraxe de la culasse fixe, voir figure A.2
 bair: Valeur de l'induction dans l'air
 bcf: Valeur de l'induction dans la dent de la culasse fixe
 bcm: Valeur de l'induction dans la culasse mobile
 bcn: Valeur de l'induction dans le noyau de la culasse fixe
 c: Largeur de la dent de la culasse fixe, voir figure A.2
 d: Hauteur de la culasse mobile, voir figure A.2
 ent: Largeur de l'entrefer, voir figure A.2
 f: Force d'appel entre la culasse mobile et la culasse fixe
 fluair: Valeur du flux total dans l'entrefer
 h: Hauteur totale du circuit magnétique
 hn: Hauteur du noyau de la culasse fixe et profondeur du circuit magnétique, voir figure A.2
 lind: Largeur totale du circuit magnétique, voir figure A.2
 m: Masse totale de électroaimant

Figure A.4: liste et signification des variables employées dans l'exemple de l'électroaimant

$$\begin{aligned}
\text{lind} &:= 2 \cdot c + b & h &:= d + \text{ent} + a + \text{hn} \\
\text{atair} &:= 2 \cdot \frac{\text{bair}}{4 \cdot \pi \cdot 10^{-7}} \cdot \text{ent} & \text{bcm} &:= \frac{\text{fluair}}{\text{hn} \cdot d} \\
\text{fluair} &:= \text{bair} \cdot c \cdot \text{hn} & \text{atcm} &:= \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcm}^2} + \text{kc} \right) \cdot \text{bcm} \cdot (b + c) \\
\text{atcf} &:= \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcf}^2} + \text{kc} \right) \cdot \text{bcf} \cdot (2 \cdot a + \text{hn}) & \text{atcn} &:= \left(\text{ka} \cdot e^{\text{kb} \cdot \text{bcn}^2} + \text{kc} \right) \cdot \text{bcn} \cdot (\text{lind} - c) \\
\text{attot} &:= \text{atair} + \text{atcm} + \text{atcf} + \text{atcn} & m &:= (d \cdot \text{lind} + 2 \cdot a \cdot c + \text{lind} \cdot \text{hn}) \cdot \text{hn} \cdot 7800 + 0.003 \\
i &:= \frac{\text{attot}}{10} & f &:= \frac{\text{bair}^2}{4 \cdot \pi \cdot 10^{-7} \cdot 2} \cdot c \cdot \text{hn} \cdot 2 \\
\text{bcf} &:= \text{bair} & \text{bcn} &:= \frac{\text{fluair}}{\text{hn}^2}
\end{aligned}$$

Constantes du problème:

$$\text{ka} := 76.1$$

$$\text{kb} := 1.26$$

$$\text{kc} := 129.5$$

Figure A.5: équations utilisées dans le modèle structural de électroaimant

ANNEXE B: un exemple de modèle structural permettant de dimensionner
des machines asynchrones

ANNEXE B: un exemple de modèle structural permettant de dimensionner des machines asynchrones

Cette annexe décrit un modèle structural permettant de dimensionner des machines asynchrones.

Ce modèle a été employé pour créer l'application de dimensionnement évoquée dans le paragraphe IV.1.d. Par ailleurs l'annexe D détaille deux exemples de cahier des charges traités avec cette application.

Le modèle est décrit en neuf étapes:

- 1° le principe du modèle structural employé,
- 2° les caractéristiques du modèle structural présenté,
- 3° la liste et la signification des paramètres d'entrée du modèle,
- 4° le calcul de grandeurs caractéristiques,
- 5° la prise en compte de l'effet de peau,
- 6° la prise en compte des pertes mécaniques,
- 7° le calcul des paramètres du schéma équivalent,
- 8° la prise en compte de la saturation à vide de la machine,
- 9° le calcul des performances du moteur.

Cette annexe contient aussi les deux types d'informations suivantes:

- * des informations précises sur les références d'où proviennent les équations (*cette annexe possède d'ailleurs sa propre bibliographie indépendante de la bibliographie générale du mémoire*),
- * des informations sur la manière dont ces équations ont été manipulées pour générer l'application de dimensionnement avec l'approche PASCOSMA.

Modèle structural de dimensionnement d'une machine asynchrone

Modèle employé pour dimensionner des machines d'une même gamme

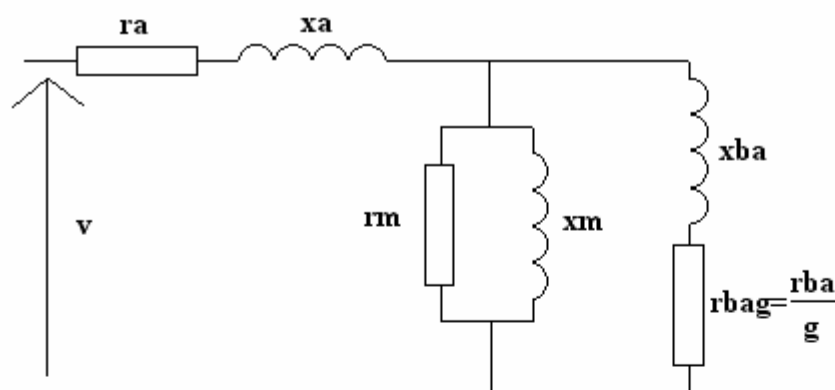
Remarque préalable:

Dans le cadre restreint de dimensionnement de machine pour lequel le modèle suivant est employé, toutes les machines utilisent la même géométrie et la même découpe de tôles: c'est pourquoi beaucoup de paramètres ont été figés dans l'application PASCOSMA (cf. *paramètres suivis de la mention figé*), ce qui signifie qu'une valeur constante leur a été attribuée, ce qui les fait disparaître de la liste des paramètres d'entrée.

B.1°. Principe du modèle structural:

Le principe du modèle utilisé est de calculer les performances du moteur en fonction des paramètres d'un schéma équivalent. Ces paramètres du schéma équivalent sont eux mêmes calculés à partir des paramètres de construction de la machines, et l'utilisation de l'ensemble permet de relier les performances de la machine à ses paramètres de construction.

Dans ce cadre, le schéma équivalent utilisé pour modéliser une phase d'une machine asynchrone triphasée est le suivant:



ra: Résistance du bobinage statorique

xa: Réactance de fuite du bobinage statorique

rm: Résistance représentative des pertes fer

xm: Inductance magnétisante du moteur

xba: Réactance de fuite équivalente du rotor ramenée au stator

rbag: Réactance équivalente du rotor ramenée au stator divisée par le glissement

g: Glissement

v: Tension d'alimentation par phase

Figure B.1

Ce schéma ne tient compte que du flux fondamental dans l'entrefer.

B.2°. Caractéristiques du modèle structural:

- Domaine d'utilisation: fonctionnement autour du point nominal ($g=0$ a $g=+0.3$).
- Modèle non-linéaire tenant compte de la saturation de la machine à vide.
- Modèle prenant en compte l'inclinaison des barres du rotor.
- Modèle ne tenant compte que de l'harmonique fondamentale (i.e. $\gamma=1$ par la suite).
- Au stator, on a un bobinage composé de 2 fils ronds au maximum. Les raccourcissements et les spires en parallèles sont prises en compte.
- Au rotor, on a une cage d'écuréuil. Cette cage est soit coulée, soit faite d'1 barre.

Le modèle offre aussi les possibilités suivantes qui ont cependant été figées dans l'application PASCOSMA qui a été générée:

- Encoches stator paramétrables, permettant de schématiser toute encoche.
- Encoches rotor paramétrables. Mais pour faire les calculs tenant compte de l'effet de peau au démarrage, on se ramène à une encoche équivalente carrée (cf. paragraphe B.5°).

B.3°. Liste des paramètres d'entrée: signification et valeurs

Variables diverses utilisées pour le fonctionnement du programme Mathcad:

(ma est la limite mathématique pour designer 0:	$ma := 1 \cdot 10^{-6}$)
(mm permet de passer des mm au m:	$mm := 10^{-3}$)
(mm ² permet de passer des mm ² au m ² :	$mm^2 := mm \cdot mm$)

Gamme de machine prise en exemple pour les valeurs numériques:

Machine pour découpe de bois, hauteur d'axe 75 mm

Soit m le nombre de phase	$m := 3$	(Figé)
Soit pa le nombre de pôles	$pa := 2$	(Figé)
Soit L la longueur de fer (en m)	$L := 150 \cdot mm$	
Soit ne le nombre d'encoches au stator	$ne := 24$	(Figé)
Soit nr le nombre de barres au rotor	$nr := 29$	(Figé)
Soit f la fréquence d'alimentation (en Hz)	$f := 50$	
Soit e la longueur de l'entrefer simple (en m)	$e := 0.40 \cdot mm$	
Soit v la tension d'alimentation par bobinage (en V)	$v := \frac{400}{\sqrt{3}}$	
Soit g le glissement de la machine au point de fonctionnement nominal:	$gm := 0.03$	
Soit nsspe le nombre de spires par encoche	$nsspe := 50$	
Soit nb le nombre de voies du bobinage en parallèle	$nb := 1$	
Soit nfa le nombre de fils en parallèle du diamètre de fil 1 utilisé	$nfa := 1$	
Soit dfila le diamètre du fil 1 utilisé (en m)	$dfila := 0.8 \cdot mm$	
Soit nfb le nombre de fils en parallèle du diamètre de fil 2 utilisé	$nfb := 1$	
Soit dfilb le diamètre du fil 2 utilisé (en m)	$dfilb := 0.5 \cdot mm$	
Soit γ le rang de l'harmonique pris en compte:	$\gamma := 1$	

Soit *renc* le nombre d'encoches de raccourcissement. *renc* := 0
 (Si le bobinage est de type section pleine, *renc*=0.
 Si le bobinage est de type demi-section, *renc*=le nombre d'encoche ou 2 phases différentes se superposent.)

Soit *tsta* la température des bobinages stator (en °C) *tsta* := 135

Soit *trot* la température des conducteurs rotor (en °C) *trot* := 140

Données provenant du plan de la tôle stator:

Soit *di* le diamètre intérieur du stator (en m), voir figure B.3 *di* := 70·mm (Figé)

Soit *deext* le diamètre extérieur de la machine (en m), voir figure B.3 *deext* := 130·mm (Figé)

Données du plan de l'encoche stator:

Le modèle d'encoche adopté est visible sur la figure B.2:

- | | |
|-------------------------------|------------------------------|
| <i>bas</i> := 5.52·mm (Figé) | <i>has</i> := 1.5·mm (Figé) |
| <i>bbs</i> := 7.52·mm (Figé) | <i>hfs</i> := 6.93·mm (Figé) |
| <i>bcs</i> := 5.876·mm (Figé) | <i>hds</i> := 2·mm (Figé) |
| <i>bds</i> := 2.7·mm (Figé) | <i>hes</i> := 0.8·mm (Figé) |
| <i>os</i> := 2.7·mm (Figé) | |

Figure B.2

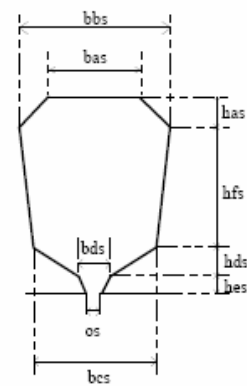


Schéma de l'encoche stator

D'où l'aire en mm² de l'encoche modélisé est:

$$ae := has \cdot \frac{bas + bbs}{2} + hfs \cdot \frac{bbs + bcs}{2} + hds \cdot \frac{bcs + bds}{2} \quad (\text{eqae})$$

La hauteur totale de l'encoche est:

$$hts := has + hfs + hds + hes \quad (\text{eqhts})$$

Données du plan de l'encoche rotor:

Soit *dir* le diamètre intérieur de la tôle rotor: $dir := \frac{40}{3} \cdot \text{mm}$

Remarque: on divise la valeur réelle du diamètre intérieur de la tôle rotor par 3, pour tenir compte de la règle qui dit que, pour un moteur 2 pôles, et pour un arbre non amagnétique, on peut considérer que le flux passe dans les 2/3 de l'arbre (cette grandeur ne sert, en effet, qu'à calculer le flux dans la culasse rotor).

Si on représente les différents diamètres qui interviennent dans le modèle, on obtient le schéma de la figure B.3

Figure B.3

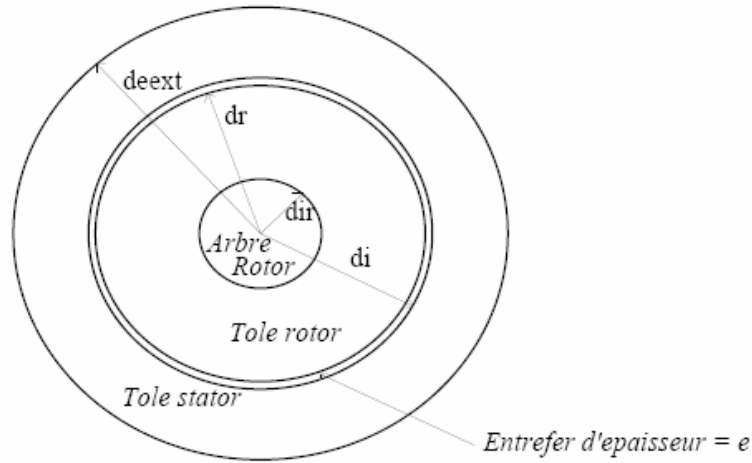


Schéma des différents diamètres utilisés dans le modèle

Le modèle d'encoche rotor adopté est le suivant:

$bar := 2.5\text{-mm}$ (Figé)	$har := 0\text{-mm}$ (Figé)
$bbr := 2.5\text{-mm}$ (Figé)	$hfr := 8.4\text{-mm}$ (Figé)
$bcr := 2.5\text{-mm}$ (Figé)	$hdr := 0\text{-mm}$ (Figé)
$bdr := 0\text{-mm}$ (Figé)	$her := 0\text{-mm}$ (Figé)
$orr := 0$ (Figé)	

Remarque: Dans le cas du moteur étudié, l'encoche est carrée (cf. les valeurs ci dessus).

Figure B.4

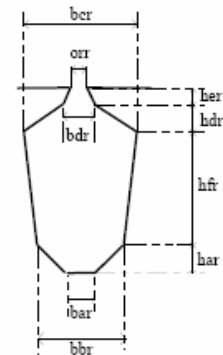


Schéma de l'encoche rotor

D'où l'aire en mm^2 de l'encoche modélisé est:

$$aer := har \cdot \frac{bar + bbr}{2} + hfr \cdot \frac{bbr + bcr}{2} + hdr \cdot \frac{bcr + bdr}{2} \quad (\text{eqaer})$$

La hauteur totale de l'encoche est:

$$htr := har + hfr + hdr + her \quad (\text{eqhtr})$$

Données concernant les conducteurs du rotor (Dans ce modèle on a soit une cage coulée au rotor, soit 1 barre rectangulaire de conducteur insérée dans l'encoche):

S'il s'agit d'une cage coulee, coulee doit valoir 1, sinon coulee doit valoir 0 et $coulee := 0$ (Figé)
dans ce cas il s'agit d'une cage a barres.

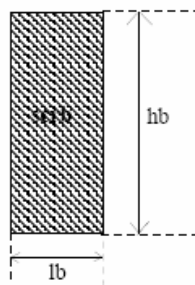
Soit scr_b la surface de conducteur dans l'encoche rotorique (en m^2) $scr_b := 16 \cdot mm^2$ (Figé)
(Cette valeur doit être impérativement donnée pour une cage avec 1 barreau non coulé. Par contre pour des cages coulées, cette valeur n'est pas utilisée, car la surface de conducteur est alors calculée à partir de la forme de l'encoche).

Soit hb la hauteur de la barre rotor. $hb := 8 \cdot mm$ (Figé)
(Cette valeur doit être impérativement donnée pour une cage avec 1 barreau non coulé. Par contre pour des cages coulées, cette valeur n'a pas de sens et n'est pas utilisée).

Soit lb la largeur de la barre $lb := 2 \cdot mm$ (Figé)
(Cette valeur doit être impérativement donnée pour une cage avec 1 barreau non coulé. Par contre pour des cages coulées, cette n'a pas de sens et n'est pas utilisée. Elle peut donc être affectée à zero ici).

Donc en fait, si le rotor est fait d'une barre, il faut rentrer les grandeurs suivantes:

Figure B.5



Dimensions de la barre rotor gérées dans le modèle lorsqu la cage rotor n'est pas une cage coulée

Données provenant des spécifications de la cage rotor:

Soit a la surface de l'anneau de court-circuit (en m^2)	$a := 125 \cdot mm^2$ (Figé)
Soit kr la valeur de l'inclinaison de la cage rotor (en pas dentaire)	$kr := 1.5$
Soit ρ_{al} la résistivité de la cage rotor (en Ohm.m)	$\rho_{al} := 0.017 \cdot 10^{-6}$
Soit α_{rot} le coefficient d'évolution de la résistivité rotor (En $1/K$)	$\alpha_{rot} := 0.00380$
Soit dam le diamètre moyen de la cage rotor (en m)	$dam := 57 \cdot mm$ (Figé)
Soit han la hauteur de l'anneau de court-circuit (en m)	$han := 10 \cdot mm$ (Figé)
Soit ban la largeur de l'anneau de court-circuit (en m)	$ban := 25 \cdot mm$ (Figé)

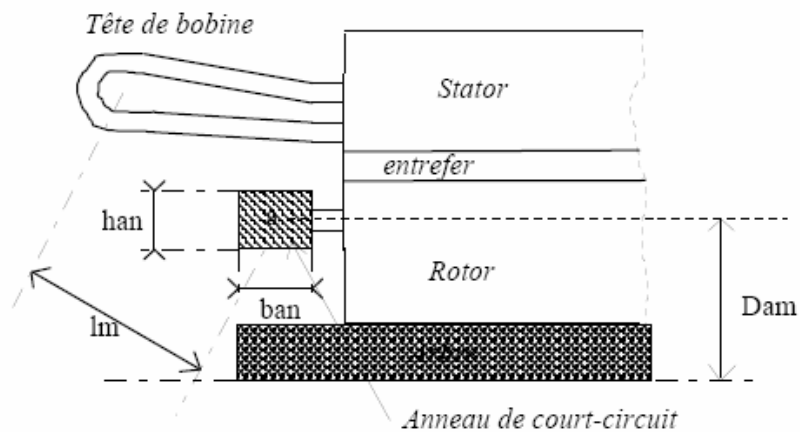
Soit l_m la distance entre le centre de la tête de bobine et l'anneau de court-circuit

$l_m := 50 \cdot \text{mm}$ (Figé)

(L'évaluation plus objective de cette grandeur est à travailler avec le constructeur de machine.
L'idéal serait de pouvoir évaluer cette grandeur en fonction d'autres grandeurs comme le polarité, la hauteur d'axe, le type du bobinage ...)

Pour comprendre plus précisément ce que représentent han, ban et l_m , on pourra se reporter au schéma de la figure B.6.

Figure B.6



Quelques dimensions nécessaires au modèle

B.4°. Calcul de grandeurs caractéristiques:

Calcul de grandeurs d'ordre géométrique:

Soit p le nombre de paires de pôles $p := \frac{pa}{2}$ (eqp)

Soit q le nombre d'encoches par pôle et par phase $q := \frac{ne}{(m \cdot p \cdot 2)}$ (eqq)

Soit n_s le nombre de spires en série par phase $n_s := \frac{n_{sspe} \cdot q \cdot p}{nb}$ (eqns)

Soit d_{enc} le nombre d'encoches par pôle:

$$d_{enc} := \frac{ne}{pa} \quad (\text{eqdenc})$$

Soit pas le pas de bobinage électromagnétique en nombre de dents: $pas := d_{enc} - r_{enc}$ (eqpas)

On calcule la pulsation: $\omega := 2 \cdot \pi \cdot f$ (eqw)

Soit l_{ds} la largeur d'une dent statorique: $l_{ds} := \pi \cdot \frac{di}{ne} - o_s$ (eqlds)

Soit d_r le diamètre extérieur du rotor (en m) $d_r := d_i - 2 \cdot e$ (eqdr)
 (C'est le diamètre intérieur stator - 2 * l'entrefer)

Calcul du coefficient c_{rs} de remplissage des encoches stator:

Pour cela on calcule d'abord la section utile de cuivre s_{ce} :

$$s_{cf} := \frac{\pi}{4} \cdot (n_{fa} \cdot d_{fila}^2 + n_{fb} \cdot d_{filb}^2) \quad (\text{eqscf})$$

$$s_{ce} := s_{cf} \cdot n_{sspe} \quad (\text{eqsce})$$

$$c_{rs} := \frac{s_{ce}}{a_e} \quad (\text{eqcrs})$$

Calcul du coefficient de bobinage:

Pour cela on calcule d'abord les 2 paramètres suivants:

On calcule l'ouverture d'une bobine (en m): $c := \frac{(\pi \cdot d_i \cdot p_{as})}{n_e}$ (eqc)

On calcule le pas polaire (en m): $\tau := \frac{(\pi \cdot d_i \cdot n_e)}{(n_e \cdot 2 \cdot p)}$ (eqtau)

On calcule le coefficient de bobinage:

Ce coefficient tient compte de la distribution et du raccourcissement

$$k_b := \frac{\sin\left[\frac{(\gamma \cdot \pi)}{(2 \cdot m)}\right]}{\left[q \cdot \sin\left[\frac{(\gamma \cdot \pi)}{(2 \cdot m \cdot q)}\right] \right]} \cdot \sin\left(\gamma \cdot \frac{c}{\tau} \cdot \frac{\pi}{2}\right) \quad (\text{eqkb})$$

Calcul de l'induction dans l'entrefer à vide qui sera utilisé pour calculer les pertes fer et pour calculer le coefficient de magnétisation:

Pour cela on définit d'abord:

Calcul du flux mutuel par pôle: $\phi_{hip} := \frac{v}{2 \cdot \pi \cdot f \cdot k_b \cdot n_s} \cdot \sqrt{2}$ (eqphip)

Dans cette équation, on part de $v = r_a \cdot i + n_s \cdot k_b(1) \cdot d\phi_{hip}/dt$ et on néglige le terme résistif.

Périmètre d'un pôle: $s_p := \pi \cdot \frac{d_i}{p_a}$ (eqsp) Surface de l'entrefer par pôle: $a_g := s_p \cdot L$ (eqag)

d'où le calcul de la valeur maximum du champ magnétique dans l'entrefer:

$$b_{eent} := \frac{\phi_{hip}}{a_g} \cdot \frac{\pi}{2} \quad (\text{eqbeent})$$

Calcul du coefficient d'inclinaison:

Calculs issus de [TAIEB] page 52 et [Alger] page 228 et [Cochran] formule 10.151.

Ce coefficient va être utilisé à plusieurs reprises afin de modéliser l'effet de l'inclinaison des barres du rotor.

Ce coefficient dépend du rang γ de l'harmonique considéré:

$$k_i := \frac{\sin\left(\gamma \cdot p \cdot \pi \cdot \frac{k_r}{n_r}\right)}{\left(\gamma \cdot p \cdot \pi \cdot \frac{k_r}{n_r}\right)} \quad (\text{eqki})$$

Calcul du coefficient permettant de ramener les impédances rotor au stator:

Calculs issus de [KOSTENKO] page 444

$$k := \frac{m \cdot n_s^2 \cdot k_b^2}{n_r \cdot n_{sr}^2 \cdot k_{br}^2 \cdot k_i^2} \quad (\text{eqk}) \quad \text{avec} \quad n_{sr} := \frac{1}{2}(\text{eqn}_{sr}) \quad \text{et} \quad k_{br} := 1 (\text{eqk}_{br})$$

Voir formule (18-23) de [Kostenko], ou page 53 de [TAIEB].

Où:

Nsr: Nombre de spires en série par phase. Nsr vaut 1/2 dans le cas de rotors à cage.

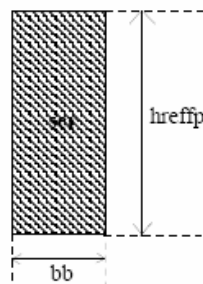
Kbr: représente le coefficient de bobinage du rotor Vaut 1 dans le cas du rotor à cage.

Ce rapport de transformation tient compte de l'inclinaison des barres des rotor [Cochran], [Jong], [Alger].

B.5°. Prise en compte de l'effet de peau:

Les formules analytiques permettant de prendre en compte l'effet de peau n'existent que pour des conducteurs de forme particulière. La formule utilisée dans notre modèle (voir paragraphe B.7°.4 et B.7°.6) est valable pour une encoche de forme suivante:

Figure B.7



Forme de conducteur considérée lors de la prise en compte de l'effet de peau au rotor.

Si au rotor, les conducteurs sont des barres de formes rectangulaires (cf. figure B.5), il n'y a aucun problème, puisqu'alors on a directement: $bb=lb$, $hreffp=hb$ et $scr=scrib$

Par contre, si au rotor, on a une cage coulée du type de celle de la figure B.4, alors pour tenir compte de l'effet de peau, on calcule une encoche carrée équivalente. Cette encoche carrée équivalente a:

- la même surface aer que l'encoche originale,
- et la même hauteur htr,

seule une largeur équivalente est calculé par la division de la surface de aer par htr.

On a donc: $scr=ser$, $hreffp=htr$ et $bb=ser/htr$.

Cette approximation sera d'autant plus fausse que l'encoche coulée s'éloignera de la forme carrée.

La gestion de ces deux configuration se fait avec les 'si' que l'on trouve ci-après, qui sont gérées dans le programme PASCOSMA par les 6 paramètres de sortie: $scrca$, $scricb$, $hreffpa$, $hreffpb$, bba , et bbb .

Calcul de la surface scr de l'encoche rotor:

```

si coulee = 1 alors   scr := aer
sinon                scr := scrib

```

Pour gérer la condition 'si' dans PASCOSMA, on introduit les deux paramètres de sortie 'scrca' et 'scricb'. Ils sont à gérer, par l'utilisateur du système, de la manière suivante:

Si cage coulée
alors poser la contrainte $scrca=0$ et ignorer $scricb$
sinon ignorer $scrca$ et poser la contrainte $scricb=0$
avec $scrca=scr-aer$ (eqscrca)
et $scricb=scr-scrib$ (eqscrib)

Calcul de la hauteur $hreffp$ du conducteur pour la prise en compte de l'effet de peau:

```

si coulee = 1 alors   hreffp := htr
sinon                hreffp := hb

```

Pour gérer la condition 'si' dans PASCOSMA, on introduit les deux paramètres de sortie 'hreffpa' et 'hreffpb'. Ils sont à gérer, par l'utilisateur du système, de la manière suivante:

Si cage coulée
alors poser la contrainte $hreffpa=0$ et ignorer $hreffpb$
sinon ignorer $hreffpa$ et poser la contrainte $hreffpb=0$
avec $hreffpa=hreffp-htr$ (eqhreffpa)
et $hreffpb=hreffp-hb$ (eqhreffpb)

Calcul de la largeur moyenne bb des conducteurs:

```

si coulee = 1 alors   bb := aer/htr
sinon                bb := lb

```

Pour gérer la condition 'si' dans PASCOSMA, on introduit les deux paramètres de sortie 'bba' et 'bbb'. Ils sont à gérer, par l'utilisateur du système, de la manière suivante:

Si cage coulée
alors poser la contrainte $bba=0$ et ignorer bbb
sinon ignorer bba et poser la contrainte $bbb=0$
avec $bba=bb-aer/htr$ (eqbba)
et $bbb=bb-lb$ (eqbbb)

Soit be la largeur moyenne de l'encoche: $be := \frac{aer}{htr}$ (eqbe)

B.6°. Prise en compte des pertes mécaniques:

Ces pertes mécaniques sont prises en compte grâce à un coefficient de frottement visqueux.

Soit $smecav$ le coefficient dont la valeur suivante a été identifiée sur une machine de référence:

$$smecav := 0.00109$$

D'un point de vue pratique, on peut, pour identifier ce coefficient, mesurer la puissance qu'il faut pour entraîner la machine alors qu'elle n'est pas alimentée. Si $Pmec$ est cette puissance, $smecav$ est défini par:

$$smecav = Pmec/w^2 \text{ ou } w \text{ est la vitesse en rad/s à laquelle l'essai est fait.}$$

B.7°. Calcul des paramètres du schéma équivalent:

B.7°1. Calcul de la résistance de l'enroulement primaire r_a

Calculs issus de [TAIEB] page 43

On commence par définir la résistivité du cuivre: $\rho := 0.017 \cdot 10^{-6}$

Ainsi que le coefficient d'évolution de cette résistivité avec la température: $\alpha := 0.0038$

On calcule la surface de cuivre pour le fil 1: $s_{ca} := \frac{\pi \cdot d_{fila}^2}{4}$ (eqsca)

On calcule la surface de cuivre pour le fil 2: $s_{cb} := \frac{\pi \cdot d_{filb}^2}{4}$ (eqscb)

D'où, la surface utile de cuivre totale est: $s_{cu} := n_{spe} \cdot (n_{fa} \cdot s_{ca} + n_{fb} \cdot s_{cb})$ (eqscu)

On calcule la longueur d'une spire, sans la longueur des têtes de bobines (en m):

$$l_{spi} := 2 \cdot L \quad (eqlspi)$$

On estime la longueur des têtes de bobine selon la formule suivante (cf. [VASSENT] page 72):

$$l_{tb} := \frac{\pi}{(2 \cdot p)} \cdot (d_i + 2 \cdot h_{ts}) + 2 \cdot h_{ts} \quad (eqltb)$$

Et la longueur totale d'une spire est:

$$l_s := l_{spi} + 2 \cdot l_{tb} \quad (eqls)$$

Soit ρ_{hot} la résistivité du bobinage en fonction de la température:

$$\rho_{hot} := \rho \cdot [1 + \alpha \cdot (t_{sta} - 20)] \quad (eqrho)$$

Soit r_{as} la résistance pour le fil 1:

$$r_{as} := \rho_{hot} \cdot \frac{n_s \cdot l_s}{s_{ca} \cdot n_{fa}} \quad (eqras)$$

Soit r_{bs} la résistance pour le fil 2:

$$r_{bs} := \rho_{hot} \cdot \frac{n_s \cdot l_s}{s_{cb} \cdot n_{fb}} \quad (eqrbs)$$

Et la résistance finale équivalente est:

$$r_a := \frac{1}{n_b} \cdot \frac{r_{bs} \cdot r_{as}}{r_{as} + r_{bs}} \quad (eqra)$$

B.7°.2. Calcul de la réactance magnétisante x_m

Calculs issus de [TAIEB] page 48

On calcule l'ouverture d'une bobine (en m): $c := \frac{(\pi \cdot di \cdot pas)}{ne}$ (eqc)

On calcule le pas polaire (en m): $\tau := \frac{(\pi \cdot di \cdot ne)}{(ne \cdot 2 \cdot p)}$ (eqtau)

On calcule le coefficient de carter qui permet de modéliser de façon moyenne les effets de l'encoche stator et rotor sur la valeur de l'inductance magnétisante [ALGER].

Pour cela, on commence par calculer le coefficient de carter du stator:

$$k_{cs} := \frac{ts}{\left[ts - \frac{os^2}{(5 \cdot e + os)} \right]} \quad \text{avec } ts := \pi \cdot \frac{di}{ne} \quad \text{le pas dentaire au stator} \quad (\text{eqts})$$

(eqkcs)

Puis on calcule le coefficient de carter du rotor, en réutilisant la formule de Carter:

$$k_{cr} := \frac{tr}{\left[tr - \frac{orr^2}{(5 \cdot e + orr)} \right]} \quad \text{avec } tr := \pi \cdot \frac{dr}{nr} \quad \text{le pas dentaire au rotor} \quad (\text{eqtr})$$

(eqkcr)

Le coefficient de carter final est obtenu par:

$$k_c := k_{cr} \cdot k_{cs} \quad (\text{eqkc})$$

On définit la perméabilité du vide: $\mu_{uz} := 4 \cdot \pi \cdot 10^{-7}$ (eqmuz)

La réactance magnétisante peut alors être calculée par la formule suivante:

$$x_m := \frac{2 \cdot m \cdot f \cdot kb^2 \cdot ns^2 \cdot L \cdot di \cdot \mu_{uz}}{p^2 \cdot e \cdot kc \cdot \gamma^2} \quad (\text{eqxm})$$

B.7°.3. Calcul de la résistance modélisant les pertes fer r_m

La valeur de la résistance représentative des pertes fer à été identifiée sur une machine de référence de la gamme à laquelle nous nous intéressons.

La valeur de la résistance identifiée est: $r_m=3000$ Ohms.
Cette machine de référence avait comme caractéristiques:

- Longueur de fer: $L1=150$ mm
- Une fréquence d'alimentation: $f1=50$ Hz
- Un coefficient de bobinage: $kb1=0.9576$
- Un nombre de spires en série par phase: $ns1=200$
- Un diamètre intérieur de tôle stator: $di1=70$ mm
- Un nombre de pôles: $pa1=2$
- Une induction maximale dans l'entrefer: 0.51 T

Principe du calcul d'extrapolation de la valeur de la résistance des pertes fers:

- * On identifie la valeur de la résistance représentative des pertes fer sur une machine de base.
- * On suppose que les pertes fers évoluent proportionnellement à la longueur de fer.
- * On suppose, que les pertes fer par unité de masse de fer, sont proportionnelles:

- à l'induction maximum locale dans le fer au carré .

Il en résulte que les pertes fer totales sont proportionnelles:

- à l'induction maximale dans l'entrefer au carré. En effet, comme on est à géométrie de tôle fixée on peut supposer que les inductions locales dans les différentes parties de la tôle sont:

- proportionnelles à l'induction dans l'entrefer,

- il en résulte bien que les pertes totales seront aussi proportionnelles à l'induction dans l'entrefer au carré.

On peut alors montrer que la résistance r_m représentant les pertes fers évolue selon la loi suivante:

Soient: $a_{un} := \frac{\sqrt{2}}{4 \cdot \pi} \cdot \frac{2}{0.07 \cdot 50 \cdot 0.9576 \cdot 200 \cdot 0.15}$ (eqaun) et $a_{deux} := \frac{\sqrt{2}}{4 \cdot \pi} \cdot \frac{p_a}{d_i \cdot f \cdot k_b \cdot n_s \cdot L}$ (eqadeux)

$$r_m := 3000 \cdot \frac{0.15}{L} \cdot \frac{2500}{f^2} \cdot \frac{a_{un}^2}{a_{deux}^2} \text{ (eqrm)}$$

Remarques:

* Cette loi extrapole l'évolution des pertes fer en fonction de la fréquence. Cette possibilité à été introduite pour autoriser des variations prospectives en fonction de ce paramètre. Elle ne doit pas être utilisé dans un autre esprit, car elle ne pas compte de l'évolution des pertes fer par hystérésis.

* Cette loi permet essentiellement d'avoir une bonne idée des pertes à vide de la machine, vu les hypothèses faites et le fait que la résistance sur la machine de base soit identifiée sur la base d'un essai à vide.

* Avec cette loi d'extrapolation, le nombre de paires de pôles, la qualité et l'épaisseur des tôles doivent rester les mêmes entre la machine de base et la machine extrapolée.

B.7°.4. Calcul de la résistance rotorique r_{ba}

Calculs issus de [Kostenko].

On commence par calculer la résistance de la barre rotorique:

$$r_b := \rho_{al} \cdot [1 + \alpha_{rot} \cdot (t_{rot} - 20)] \cdot \frac{L}{s_{cr}} \text{ (eqrb)}$$

On calcule ensuite la résistance de la portion d'anneau de court-circuit entre 2 barres. Cette résistance est ramenée à une résistance en série avec la barre.

$$r_{eqan} := \pi \cdot \rho_{al} \cdot [1 + \alpha_{rot} \cdot (t_{rot} - 20)] \cdot \frac{d_{am}}{(nr \cdot a)} \cdot \frac{1}{\left(2 \cdot \sin\left(\frac{\pi \cdot p}{nr} \cdot \gamma\right)\right)^2} \text{ (eqreqan)}$$

Voir formule 18-24 de [Kostenko].

Prise en compte de l'effet de peau:

Soit:

$$\epsilon_p := h_{reffp} \cdot \sqrt{\left(\pi \cdot \mu_{uz} \cdot \frac{b_b \cdot \gamma \cdot f \cdot |g_m|}{b_e \cdot \rho_{al}}\right)} \text{ (eqepsp) } \text{ Voir formule 23-19 de [Kostenko] ou formule 13.204 de [Cochran].}$$

$$k_{rr} := \epsilon_p \cdot \frac{\text{sh}(2 \cdot \epsilon_p) + \sin(2 \cdot \epsilon_p)}{\text{ch}(2 \cdot \epsilon_p) - \cos(2 \cdot \epsilon_p)} \quad (\text{eqkrr}) \quad \text{Voir formule 23-18 de [Kostenko] ou formule 10.64 de [Cochran].}$$

Finalement, la résistance rotorique est:

$$r_{bbs} := r_b \cdot k_{rr} + r_{eqan} \cdot 2 \quad (\text{eqrbbs}) \quad (\text{Voir formule 18-22 de [Kostenko]}).$$

Ramenée au stator, la résistance rotorique vaut: $r_{ba} := k \cdot r_{bbs} \quad (\text{eqrba})$

B.7°.5. Calcul de l'inductance de fuite du bobinage stator xa

k_s représente le coefficient par lequel il faut multiplier l'inductance de fuite en cas de raccourcissement du bobinage.

$$k_s := \frac{1}{2} \cdot \left(1 + \cos \left(\text{renc} \cdot \frac{\pi}{\text{denc}} \right) \right) \quad (\text{eqks}) \quad (\text{Voir page d 440-6 de [T1]})$$

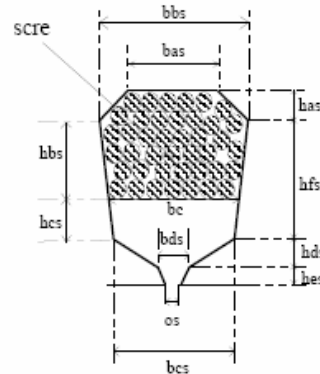
k_{xs} représente le coefficient, qui multiplié par les coefficients de perméance va permettre de calculer les réactances:

$$k_{xs} := \mu_{uz} \cdot m \cdot \omega \cdot n_s^2 \cdot 4 \cdot \frac{L}{n_e} \cdot k_s \quad (\text{eqkxs}) \quad \begin{matrix} (\text{Voir formule 5.14 de [Kostenko]}) \\ (\text{Voir formule 38 de [T1]}) \end{matrix}$$

On commence par calculer la réactance de fuite d'encoche statorique:

Nous travaillons sur le type d'encoche suivant:

Figure B.8



Encoche stator avec ses conducteurs

Commençons par calculer les grandeurs h_{bs} , h_{cs} et bc introduites sur ce schémas:

Soit $scrc$ la surface réelle occupée par les conducteurs:

$$scrc := scru \cdot \frac{4}{\pi} \quad (\text{eqscrc})$$

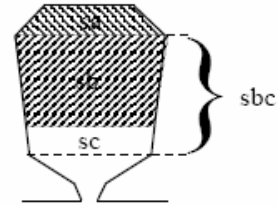
Car on montre, que pour mettre des fils ronds sur une surface carrée, il faut que la surface carrée soit $4/\pi$ fois grande que la surface des fils.

$$sa := has \cdot \frac{bas + bbs}{2} \quad (\text{eqsa}) \quad \quad \quad sbc := hfs \cdot \frac{bbs + bcs}{2} \quad (\text{eqsbc})$$

$$sb := scre - sa \quad (\text{eqsb})$$

$$sc := sbc - sb \quad (\text{eqsc})$$

Figure B.9



Avec sa, sb, sc étant les aires définies sur la figure B.9. avec $sbc = sb + sc$

Découpage de l'encoche pour le calcul de la hauteur équivalente de conducteur

Et on peut démontrer pour bc que, à condition de respecter le fait que $scre > sa$ et $scre < sa + sbc$. (cela revient à dire aussi qu'il faut imposer $hbs > 0$ et $hcs > 0$, ce qu'il est conseillé de systématiquement faire dans PASCOSMA):

$$bc := \sqrt{2 \cdot \frac{sb \cdot bcs}{hfs} + 2 \cdot \frac{sc \cdot bbs}{hfs} - bbs \cdot bcs} \quad (\text{eqbc})$$

$$hbs := 2 \cdot \frac{sb}{bbs + bc} \quad (\text{eqhbs}) \quad \quad \quad hcs := 2 \cdot \frac{sc}{bcs + bc} \quad (\text{eqhcs})$$

Pour cette encoche, on peut alors calculer le coefficient de perméance suivant :

$$\lambda_{as} := \frac{2 \cdot has}{3 \cdot (bas + bbs)} + \frac{2 \cdot hbs}{3 \cdot (bbs + bc)} + \frac{2 \cdot hcs}{bc + bcs} + \frac{2 \cdot hds}{bcs + bds} + \frac{hes}{bds} \quad (\text{eqlandaas})$$

(Voir formule 5-14 de [Kostenko], ou formule 11 de [TI]).

d'où le calcul de la réactance d'encoche statorique: $x_{es} := k_{xs} \cdot \lambda_{as} \quad (\text{eqxes})$

On calcule la réactance due aux flux de fuite dent à dent passant dans l'entrefer=Flux de dispersion dans l'entrefer:

$$\lambda_{bs} := \left(1 + \frac{orr}{orr + ma} \right) \cdot \frac{5 \cdot e}{5 \cdot os + 4 \cdot e} \quad (\text{eqlandabs})$$

Cette valeur est issue de [TI], formule 18. Le facteur qui dépend de 'orr' et de 'ma', permet simplement de tenir compte de la remarque se trouvant dans cette référence et disant que: si les 2 parties de la machines sont dentées, la formule dépendant de 'e' et de 'os' doit être multipliée par 2. On constate que si 'orr' différent de 0, on obtient bien le résultat voulu car ma vérifiera toujours: $ma \ll orr$.

d'où: $x_{diss} := k_{xs} \cdot \lambda_{bs} \quad (\text{eqxdiss})$

On calcule la réactance de fuite différentielle statorique:

D'après les techniques de l'ingénieur (Page D-440-10) formule 28, on calcule le coefficient de perméance correspondant au flux différentiel de phase est:

$$\lambda_{cs} := \frac{\tau^2}{\left(\pi^2 \cdot e \cdot ts \right)} \cdot K_{diff} \quad (\text{eqlandacs}) \quad \quad \quad \text{avec} \quad K_{diff} := 0.0025 \quad (\text{eqkdiff})$$

d'où $x_{ds} := k_{xs} \cdot \lambda_{cs} \quad (\text{eqxds})$

On calcule la réactance de fuite zig-zag statorique:

D'après les techniques de l'ingénieur (Page D 440-11) formule 34, on calcule le coefficient de perméance tenant compte du flux zig-zag):

$$\lambda_{ds} := \frac{(t_s - o_s)^2}{(8 \cdot e \cdot t_s)} \quad (\text{eqlandads})$$

d'où $x_{zs} := k_{xs} \cdot \lambda_{ds} \quad (\text{eqxzs})$

On calcule la réactance des têtes de bobines statorique:

On estime cette réactance avec une méthode approximative (cf. [Kostenko] page 136)

Puis pour les bobinages concentriques à pôles non conséquents, on calcule le terme suivant:

$$p_{va} := 0.47 \cdot l_{tb} - 0.3 \cdot \tau \quad (\text{eqpva})$$

Pour les bobinages concentriques à pôles conséquents, il faut calculer le terme suivant:

$$p_{vb} := 0.67 \cdot l_{tb} - 0.43 \cdot \tau \quad (\text{eqpvb})$$

Pour avoir un ordre de grandeur général, nous faisons la moyenne des deux:

$$v_a := \frac{(p_{va} + p_{vb})}{2} \quad (\text{eqpv})$$

d'où, (cf. [kostenko] page 131 formule 5-14):

$$x_{tb} := q \cdot v_a \cdot \frac{l_{tb}}{L} \cdot k_{xs} \quad (\text{eqxtb})$$

Il est à noter que les formules pour calculer les inductances de tête de bobines sont très nombreuses et variées dans la littérature et qu'il faudrait faire une étude comparative.

On peut alors calculer l'inductance de fuite statorique par la formule:

$$x_a := x_{es} + x_{ds} + x_{zs} + x_{diss} + x_{tb} \quad (\text{eqxa})$$

B.7°.6. Calcul de l'inductance de fuite du rotor x_{ba}

On commence par définir le coefficient, qui multiplié par les coefficients de perméance va permettre de calculer les réactances:

$$k_{xr} := \mu_{uz} \cdot \omega \cdot L \quad (\text{eqkxr})$$

On commence par calculer la réactance de fuite d'encoche rotorique.

Nous travaillons sur le type d'encoche représentée sur la figure B.4.

Pour cette encoche, on peut alors calculer le coefficient de perméance (voir *eqlandaar*), en supposant qu'elle est complètement pleine de conducteur, sauf dans l'isthme de hauteur h_{er} (ce qui est vrai si on a une cage coulée, mais reste plus approximatif lorsqu'on utilise des barres):

Prise en compte de l'effet de peau:

Pour calculer cette perméance, on introduit la variation de perméance au niveau des conducteurs due à l'effet de peau grâce au coefficient suivant:

(cf. [TAIEB], [Kostenko], formule 23-18), [Cochran] figure 10.9 il parle du rapport "of bar portion of slot").

$$k_x := \frac{3}{(2 \cdot \varepsilon p)} \cdot \frac{\text{sh}(2 \cdot \varepsilon p) - \sin(2 \cdot \varepsilon p)}{\text{ch}(2 \cdot \varepsilon p) - \cos(2 \cdot \varepsilon p)} \quad (\text{eqkx})$$

$$\lambda_{ar} := \left[\frac{2 \cdot \text{har}}{3 \cdot (\text{bar} + \text{bbr})} + \frac{2 \cdot \text{hfr}}{3 \cdot (\text{bbr} + \text{bcr})} + \frac{2 \cdot \text{hdr}}{3 \cdot (\text{bcr} + \text{bdr})} \right] \cdot k_x + \frac{\text{her}}{\text{bdr} + \text{ma}} \quad (\text{eqlandaar})$$

N.B: dans cette formule, on divise her par bdr+ma et non pas simplement par bdr pour éviter une indétermination lorsque her=0 et bdr=0

d'où $x_{er} := k_x \cdot \lambda_{ar}$ (eqxer) (Voir formule 5-14 de [Kostenko] ou formule 11 de [T])

On calcule la réactance due aux flux de fuite dent à dent passant dans l'entrefer=Flux de dispersion dans l'entrefer:

$$\lambda_{br} := \left(1 + \frac{\text{os}}{\text{os} + \text{ma}} \right) \cdot \frac{5 \cdot e}{5 \cdot \text{orr} + 4 \cdot e} \quad (\text{eqlandabr})$$

Cette valeur est issue de [T], formule 18.
Le facteur qui dépend de 'os' et de 'ma', permet simplement de tenir compte de la remarque se trouvant dans cette référence et disant que: si les 2 parties de la machines sont dentées, la formule dépendant de 'e' et de 'orr' doit être multipliée par 2.
On constate que si 'os' différent de 0, on obtient bien le résultat voulu car ma vérifiera toujours: ma << os.

d'où: $x_{dir} := k_x \cdot \lambda_{br}$ (eqxdir)

On calcule la réactance de fuite différentielle rotorique:

D'après les techniques de l'ingénieur (Page D-440-10), on calcule aussi le coefficient de perméance tenant compte du flux différentiel de phase

$$\lambda_{cr} := \frac{t_r^2}{\left(\pi^2 \cdot e \cdot t_r \right)} \cdot K_{diff} \quad (\text{eqlandacr})$$

d'où $x_{dr} := k_x \cdot \lambda_{cr}$ (eqxdr)

On calcule la réactance de fuite zig-zag rotorique:

D'après les techniques de l'ingénieur (Page D 440-11), on calcule le coefficient de perméance tenant compte du flux zig-zag):

$$\lambda_{dr} := \frac{(t_r - \text{orr})^2}{(8 \cdot e \cdot t_r)} \quad (\text{eqlandadr})$$

d'où $x_{zr} := k_x \cdot \lambda_{dr}$ (eqxzr)

On calcule la réactance de fuite de l'anneau de court-circuit:

On calcule le coefficient de perméance correspondant à l'anneau de court-circuit. Pour cela on utilise la formule 18.25.a de [Kostenko], ainsi qu'une approximation analytique de la courbe 18-6 du même ouvrage:

L'approximation analytique a été réalisée selon la formule suivante:

$$a_k := 142.222 \quad (\text{eqak}) \quad \varepsilon := 0.223 \cdot (\text{han} + \text{ban}) \quad (\text{eqeps})$$

$$b_k := 0.444 \quad (\text{eqbk})$$

$$c_k := 1 \quad (\text{eqck})$$

$$g_f := \frac{1}{2 \cdot a_k} \left[-b_k + \sqrt{b_k^2 - 4 \cdot a_k \cdot \left(c_k - \frac{\text{lm}}{\varepsilon} \right)} \right] \quad (\text{eqgf})$$

On a alors, d'après la formule 18.25a de [Kostenko]:

$$\lambda_{ep} := \frac{n_r}{2 \cdot p \cdot m} \cdot \frac{\tau}{L} \cdot g_f \quad (\text{eqlandaer})$$

$$\text{d'où} \quad x_{acr} := k_{xr} \cdot \lambda_{ep} \quad (\text{eqxacr})$$

On calcule la réactance de fuite due à l'inclinaison des barres du rotor:

Cette inductance se calcule selon la formule suivante: (Cf. [Alger] formule 7.60, [Cochran] formule 13.159, [Jong], [Christofides 65] formule 7).

$$x_{skew} := \left(\frac{1}{k_i^2} - 1 \right) \cdot x_m \quad (\text{eqxskew}) \quad \text{N.B: Il s'agit donc là d'une valeur de réactance déjà ramenée au primaire.}$$

On peut alors calculer l'inductance de fuite statorique par la formule:

La somme des réactance non ramenées au primaire vaut:

$$x_b := x_{er} + x_{dr} + x_{zr} + x_{dir} + x_{acr} \quad (\text{eqxb})$$

La réactance totale vue au primaire vaut:

$$x_{ba} := k \cdot x_b + x_{skew} \quad (\text{eqxba})$$

B.8°. Prise en compte de la saturation à vide de la machine :

Méthode issue de [Cochran]

Cette prise en compte est réalisée grâce au calcul d'un facteur de saturation. Celui est égal au rapport suivant:

somme des ampères nécessaires pour faire circuler le flux magnétique dans l'entrefer et dans la tôle magnétique

divisé par

les ampères tours nécessaires pour faire circuler le flux magnétique dans l'entrefer.

Ce coefficient sert ensuite à diviser l'inductance de magnétisation x_m du schéma équivalent.

Pour effectuer ce calcul, nous avons besoin des trois caractéristiques suivantes:

* Caractéristique $H(B)$ de la tôle utilisée:

$$h_t(b) := 175.673 \cdot b^{0.506} + \frac{\exp(b - 2.385)^{4.606}}{1 + \exp(b - 2.385)^{4.606}} \cdot (3359 \cdot b^{5.839}) \cdot h_t(x)$$

* Caractéristique $H(B)$ équivalente dans la culasse:

$$h_c(b) := 142.426 \cdot b^{0.49} + \frac{\exp(b - 2.748)^{4.055}}{1 + \exp(b - 2.748)^{4.055}} \cdot (3280 \cdot b^{5.652}) \cdot h_c(x)$$

Cette caractéristique équivalente est une interpolation mathématique de $\frac{1}{\pi} \cdot \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} h_{cc}(b \cdot \cos(\alpha)) d\alpha$

qui représente la courbe de magnétisation équivalente de la culasse dans les conditions suivantes:

- l'induction dans la culasse est purement orthoradiale,
- son module varie sinusoidalement ($\alpha=0$ sur l'axe interpolaire).

Dans cette formule b représente l'induction maximale dans la culasse qui se situe sur l'axe interpolaire

Pour plus de précision, on pourra se reporter à [Cochran].

Détail du calcul du facteur de saturation:

Calcul du flux mutuel par pôle: $\phi_{hip} := \frac{v}{2 \cdot \pi \cdot f \cdot k_b \cdot n_s} \cdot \sqrt{2}$ (eqhip)

Dans cette équation, on part de $v = r_a \cdot i + n_s \cdot k_b(1) \cdot d\phi_{hip}/dt$ et on néglige le terme résistif.

Périmètre d'un pôle: $s_p := \pi \cdot \frac{d_i}{p_a}$ (eqsp)

Surface de l'entrefer par pôle: $a_g := s_p \cdot L$ (eqag)

Calcul de la valeur maximum du champ magnétique dans l'entrefer:

$b_{eent} := \frac{\phi_{hip}}{a_g} \cdot \frac{\pi}{2}$ (eqbeent)

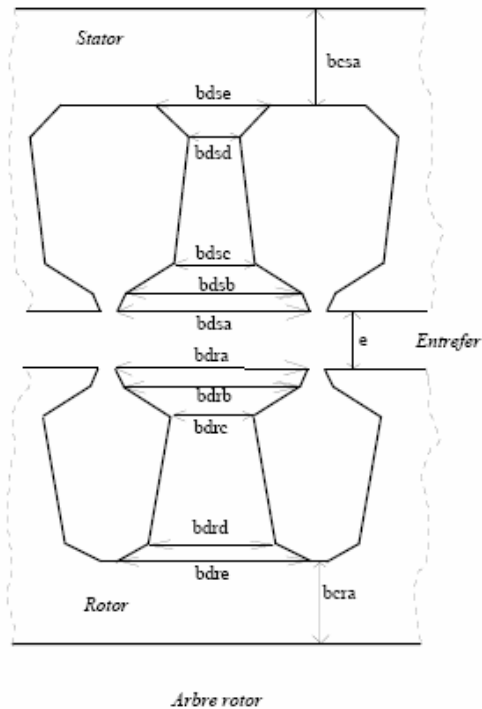
Calcul du nombre d'ampères tours nécessaires pour faire circuler le champ magnétique dans l'entrefer simple:

$n_{ig} := \frac{k_c \cdot e \cdot b_{eent}}{4 \cdot \pi \cdot 10^{-7}}$ (eqnig)

Calcul d'inductions dans la tôle stator et calcul des ampères tours nécessaire pour faire circuler ces inductions:

Dans les pages suivantes, on calcule les inductions représentées sur la figure suivante. On fait l'hypothèse que tout le flux passe dans le fer.

Figure B.10



Projection linéaire de la tôle rotor et stator sur 2 encoches pour visualiser les inductions maximales calculées.

Calcul des inductions et ampères tours consommés dans les dents stator:

Calcul de bdsa, l'induction maximale à la base de l'isthme de l'encoche:

Ce calcul, comme les calculs d'induction suivants se fait en 2 étapes:

1° Calcul de la section de passage:

$$sbdsa := \frac{di \cdot \pi - ne \cdot os}{pa} \cdot L \quad (eqsbdsa)$$

2° Calcul de l'induction maximale:

$$bdsa := \frac{phip}{sbdsa} \cdot \frac{\pi}{2} \quad (eqbdsa)$$

Calcul de bdsb, l'induction maximale à la base de l'encoche elle même:

$$sbdsb := \frac{(di + hes \cdot 2) \cdot \pi - ne \cdot bds}{pa} \cdot L \quad (eqsbdsb)$$

$$bdsb := \frac{phip}{sbdsb} \cdot \frac{\pi}{2} \quad (eqbdsb)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre bdsa et bdsb:

On calcule l'induction moyenne entre bdsa et bdsb:

$$bdmsab := \frac{bdsa + bdsb}{2} \quad (eqbdmsab)$$

On calcule les ampères tours totaux nécessaires pour faire circuler le flux sur la hauteur h_{es} grâce à la formule de Simpson:

$$a_{thes} := h_{es} \cdot \frac{ht(b_{dsa}) + 4 \cdot ht(b_{dmsab}) + ht(b_{dsb})}{6} \quad (eqathes)$$

Calcul de b_{dsc} :

$$s_{bdsc} := \frac{[di + (h_{es} + h_{ds}) \cdot 2] \cdot \pi - n_e \cdot b_{cs}}{p_a} \cdot L \quad (eqsbdsc)$$

$$b_{dsc} := \frac{\phi_{hip}}{s_{bdsc}} \cdot \frac{\pi}{2} \quad (eqbdsc)$$

On calcule l'induction moyenne entre b_{dsb} et b_{dsc} :

$$b_{dmsbc} := \frac{b_{dsb} + b_{dsc}}{2} \quad (eqbdmsbc)$$

puis les ampères tours nécessaire pour faire circuler l'induction entre b_{dsb} et b_{dsc} :

$$a_{thds} := h_{ds} \cdot \frac{ht(b_{dsb}) + 4 \cdot ht(b_{dmsbc}) + ht(b_{dsc})}{6} \quad (eqathds)$$

Calcul de b_{dsd} :

$$s_{bdsd} := \frac{[di + (h_{es} + h_{ds} + h_{fs}) \cdot 2] \cdot \pi - n_e \cdot b_{bs}}{p_a} \cdot L \quad (eqsbdsd)$$

$$b_{dsd} := \frac{\phi_{hip}}{s_{bdsd}} \cdot \frac{\pi}{2} \quad (eqbdsd)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre b_{dsc} et b_{dsd} :

$$b_{dmscd} := \frac{b_{dsc} + b_{dsd}}{2} \quad (eqbdmscd)$$

$$a_{thfs} := h_{fs} \cdot \frac{ht(b_{dsc}) + 4 \cdot ht(b_{dmscd}) + ht(b_{dsd})}{6} \quad (eqathfs)$$

Calcul de b_{dse} :

$$s_{bdse} := \frac{[di + (h_{es} + h_{ds} + h_{fs} + h_{as}) \cdot 2] \cdot \pi - n_e \cdot b_{as}}{p_a} \cdot L \quad (eqsbdse)$$

$$b_{dse} := \frac{\phi_{hip}}{s_{bdse}} \cdot \frac{\pi}{2} \quad (eqbdse)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre b_{dsd} et b_{dse} :

$$b_{dmsde} := \frac{b_{dsd} + b_{dse}}{2} \quad (eqbdmsde)$$

$$a_{thas} := h_{as} \cdot \frac{ht(b_{dsd}) + 4 \cdot ht(b_{dmsde}) + ht(b_{dse})}{6} \quad (eqathas)$$

D'où le nombre total d'ampères tours nécessaires pour faire circuler le flux dans les dents stator est:

$$n_{ids} := a_{thes} + a_{thds} + a_{thfs} + a_{thas} \quad (eqnids)$$

Calcul de l'induction max. et des ampères tours consommés dans la culasse stator:

Calcul de la section de passage dans la culasse stator:

$$s_{bc\text{sa}} := \frac{de_{\text{ext}} - [di + 2 \cdot (hes + hds + hfs + has)]}{2} \cdot L \quad (\text{eqsbcsa})$$

$$b_{c\text{sa}} := \frac{\phi_{\text{hip}}}{s_{bc\text{sa}} \cdot 2} \quad (\text{eqbc\text{sa}}) \quad \text{On divise par 2, car seule la moitié du flux présent dans l'entrefer passe dans la culasse sur l'axe interpolaire.}$$

Calcul des ampères tours nécessaires pour faire circuler le flux dans la culasse stator:

Calcul de la longueur moyenne de la ligne de flux dans la culasse stator:

$$l_{c\text{s}} := \frac{de_{\text{ext}} - \frac{de_{\text{ext}} - [di + 2 \cdot (hes + hds + hfs + has)]}{2}}{2 \cdot \rho_a} \cdot \pi + \frac{\frac{de_{\text{ext}} - [di + 2 \cdot (hes + hds + hfs + has)]}{2}}{4} \cdot \pi \quad (\text{eqlcs})$$

Formule proposée par [Cochran]. Le second terme de cette formule permet de tenir compte du flux radial dans la culasse

Calcul des ampères tours nécessaires pour faire passer le flux dans la culasse stator:

$$n_{i\text{cs}} := l_{c\text{s}} \cdot hc(b_{c\text{sa}}) \quad (\text{eqnics})$$

Calcul des inductions et ampères tours consommés dans les dents rotor:

Calcul de b_{dra} , l'induction maximale à la base de l'isthme de l'encoche:

Ce calcul, comme les calculs d'induction suivants se fait en 2 étapes:

1° Calcul de la section de passage dans les sommet de la dent:

$$s_{b\text{dra}} := \frac{\pi \cdot dr - nr \cdot orr}{\rho_a} \cdot L \quad (\text{eqsbdra})$$

2° Calcul de l'induction max. dans le sommet de la dent:

$$b_{dra} := \frac{\phi_{\text{hip}}}{s_{b\text{dra}}} \cdot \frac{\pi}{2} \quad (\text{eqbdra})$$

Calcul de b_{drb} , l'induction maximale à la base de l'encoche elle même:

$$s_{b\text{drb}} := \frac{(di - 2 \cdot her) \cdot \pi - nr \cdot bdr}{\rho_a} \cdot L \quad (\text{eqsbdrb})$$

$$b_{drb} := \frac{\phi_{\text{hip}}}{s_{b\text{drb}}} \cdot \frac{\pi}{2} \quad (\text{eqbdrb})$$

Calcul des ampères tours nécessaires pour faire circuler l'induction entre b_{dra} et b_{drb} :

On calcule l'induction moyenne entre b_{dra} et b_{drb} :

$$b_{dmr\text{ab}} := \frac{b_{dra} + b_{drb}}{2} \quad (\text{eqbdmrab})$$

$$a_{\text{ther}} := her \cdot \frac{ht(b_{dra}) + 4 \cdot ht(b_{dmr\text{ab}}) + ht(b_{drb})}{6} \quad (\text{eqather})$$

Calcul de bdr̄c:

$$sbdr̄c := \frac{[di - 2 \cdot (her + hdr)] \cdot \pi - nr \cdot bcr}{pa} \cdot L \quad (eqsbdr̄c)$$

$$bdr̄c := \frac{p_{hip}}{sbdr̄c} \cdot \frac{\pi}{2} \quad (eqbdr̄c)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre bdr̄b et bdr̄c:

$$bdr̄mrc := \frac{bdr̄b + bdr̄c}{2} \quad (eqbdr̄mrc)$$

$$athdr := hdr \cdot \frac{ht(bdr̄b) + 4 \cdot ht(bdr̄mrc) + ht(bdr̄c)}{6} \quad (eqathdr)$$

Calcul de bdr̄d:

$$sbdr̄d := \frac{[di - 2 \cdot (her + hdr + hfr)] \cdot \pi - nr \cdot bbr}{pa} \cdot L \quad (eqsbdr̄d)$$

$$bdr̄d := \frac{p_{hip}}{sbdr̄d} \cdot \frac{\pi}{2} \quad (eqbdr̄d)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre bdr̄c et bdr̄d:

$$bdr̄mrcd := \frac{bdr̄c + bdr̄d}{2} \quad (eqbdr̄mrcd)$$

$$athfr := hfr \cdot \frac{ht(bdr̄c) + 4 \cdot ht(bdr̄mrcd) + ht(bdr̄d)}{6} \quad (eqathfr)$$

Calcul de bdr̄e:

$$sbdr̄e := \frac{[di - 2 \cdot (her + hdr + hfr + har)] \cdot \pi - ne \cdot bar}{pa} \cdot L \quad (eqsbdr̄e)$$

$$bdr̄e := \frac{p_{hip}}{sbdr̄e} \cdot \frac{\pi}{2} \quad (eqbdr̄e)$$

Calcul des ampères tours nécessaire pour faire circuler l'induction entre bdr̄d et bdr̄e:

$$bdr̄mrde := \frac{bdr̄d + bdr̄e}{2} \quad (eqbdr̄mrde)$$

$$athar := har \cdot \frac{ht(bdr̄d) + 4 \cdot ht(bdr̄mrde) + ht(bdr̄e)}{6} \quad (eqathar)$$

D'où le nombre total d'ampères-tours nécessaires pour faire circuler le flux dans les dents rotor est:

$$nidr := ather + athdr + athfr + athar \quad (eqnidr)$$

Calcul de l'induction max. et des ampères-tours consommés dans la culasse rotor:

Calcul de la section de passage dans la culasse rotor:

$$sbcra := \frac{di - 2 \cdot (her + hdr + hfr + har) - dir}{2} \cdot L \quad (eqsbcra)$$

$$bcra := \frac{\phi_{hip}}{sbcra \cdot 2} \quad (eqbcra)$$

Calcul des ampères tours nécessaires pour faire circuler le flux dans la culasse rotor:

Calcul de la longueur moyenne de la ligne de flux dans la culasse rotor:

$$lcr := \frac{di - [dir + 2 \cdot (her + hdr + hfr + har)]}{2 \cdot pa} \cdot \pi + \frac{di - [dir + 2 \cdot (her + hdr + hfr + har)]}{4} \cdot \pi \quad (eqlcr)$$

Formule proposée par [Cochran]. Le second terme de cette formule permet de tenir compte du flux radial dans la culasse

Calcul des ampères-tours nécessaires pour faire passer le flux dans la culasse rotor:

$$nicr := lcr \cdot hc(bcra) \quad (eqnicr)$$

D'où on peut à présent calculer le coefficient de saturation:

$$\text{On a:} \quad k_{sat} = \frac{2 \cdot nig + 2 \cdot nids + nics + 2 \cdot nidr + nicr}{2 \cdot nig}$$

Mais on va calculer k_{sat} selon les 2 équations suivantes

$$kic := 1 + \frac{nics + nicr}{2 \cdot nig} \quad (eqkic)$$

$$k_{sat} := \frac{nids + nidr}{nig} + kic \quad (eqksat)$$

On introduit donc la variable kic . Cette manoeuvre a pour objectif de rendre le calcul symbolique et la génération de code dans PASCOSMA plus simple:

En effet l'expression de k_{sat} va apparaître dans toutes les expressions macroscopiques calculées à partir du schéma équivalent.

Ces expressions sont $cosphi$, pu , $pabs$,... . Il faut savoir que le calcul de ces grandeurs symboliques devient vite très gros. D'où la nécessité d'introduire des paramètres intermédiaires, qui seront notamment les paramètres du schéma équivalent. A première vue, il aurait été intéressant d'introduire k_{sat} comme paramètre intermédiaire. Le problème est que tout paramètre intermédiaire est calculé symboliquement en fonction exclusivement des paramètres d'entrées du modèle, et il s'avère que l'expression de k_{sat} résultante aussi très grande.

D'où la nécessité de prendre des paramètres intermédiaires dont dépend k_{sat} .

La première idée a été de prendre $nics$, $nicr$, $nids$, $nidr$ et nig comme paramètres intermédiaires. Le problème était alors que les expressions de $cosphi$, pu , $pabs$,... étaient symboliquement plus grosses. Aussi, on a réduit le nombre des paramètres qui intervenaient dans ces équations, en introduisant kic comme paramètre intermédiaire.

B.9°. Calcul des performances du moteur:

On pose:

$$z_a := r_a + i \cdot x_a \quad (\text{eqza})$$

$$r_{bag} := \frac{r_{ba}}{g_m} \quad (\text{eqrbag})$$

$$x_{mk} := \frac{x_m}{k_{sat}} \quad (\text{eqxmk})$$

$$z_b := r_{bag} + i \cdot x_{ba} \quad (\text{eqzb})$$

$$z_m := \frac{i \cdot x_{mk} \cdot r_m}{x_{mk} \cdot i + r_m} \quad (\text{eqzm})$$

On peut alors calculer l'impédance du circuit équivalent:

$$z := z_a + \frac{z_b \cdot z_m}{z_b + z_m} \quad (\text{eqz})$$

qui admet comme norme:

$$n_z := \frac{|z_b \cdot z_m + z_a \cdot (z_m + z_b)|}{|z_m + z_b|} \quad (\text{eqnz})$$

Et le courant d'alimentation par phase:

$$i_a := \frac{v}{n_z} \quad (\text{eqia})$$

On calcule le courant dans la branche représentant le rotor:

$$i_b := \frac{|z_m|}{|z_b \cdot z_m + z_a \cdot (z_m + z_b)|} \cdot v \quad (\text{eqib})$$

A l'origine on avait écrit: $i_b = \left| \frac{z_m}{z_m + z_b} \right| \cdot i_a$

qui s'obtient en appliquant dans le schéma équivalent de la figure B.1 la règle du diviseur du courant. Mais la forme finale de eqib a été choisie, car elle semble aboutir à un code symbolique généré plus simple.

On calcule le couple électromagnétique total que fournit la machine:

$$c_{em} := 3 \cdot r_{bag} \cdot i_b^2 \cdot \frac{p}{(2 \cdot \pi \cdot f)} \quad (\text{eqcem})$$

On calcule la puissance mécanique totale que fournit la machine:

$$p_t := \left[3 \cdot (1 - g_m) \cdot r_{bag} \cdot i_b^2 \right] \quad (\text{eqpt})$$

On calcule le facteur de puissance du moteur:

$$\cos\phi_i := \frac{\text{Re}(z)}{n_z} \quad (\text{eqcosphi})$$

Et le courant absorbé à vide (On considère que $g_0 = m_a$):

$$I_{0\text{vide}} := \frac{v}{nz}$$

Calcul du couple de résistance mécanique visqueux:

$$c_{\text{mecav}} := s_{\text{mecav}} \cdot \omega \cdot \frac{1 - g_m}{p} \quad (\text{eqcmecav})$$

Calcul des pertes mécaniques dues aux forces de frottement visqueuses:

$$p_{\text{mecav}} := c_{\text{mecav}} \cdot \frac{1 - g_m}{p} \cdot \omega \quad (\text{eqpmecav})$$

Puis on calcule le couple utile de la machine:

$$c_u := 3 \cdot r_{\text{bag}} \cdot i_a^2 \cdot \frac{p}{2 \cdot \pi \cdot f} - c_{\text{mecav}} \quad (\text{eqcu})$$

Puis on calcule la puissance utile que fournit la machine:

$$p_u := p_t - p_{\text{mecav}} \quad (\text{eqpu})$$

On calcule la puissance active absorbée:

$$p_{\text{abs}} := 3 \cdot v \cdot i_a \cdot \cos \phi_i \quad (\text{eqpabs})$$

On calcule le rendement:

$$\text{rend} := \frac{p_u}{p_{\text{abs}}} \quad (\text{eqrend})$$

Calcul de la densité de courant:

$$j_s := \frac{i_a}{s_{\text{cf}} \cdot n_b} \quad (\text{eqjs})$$

Equation calculant la fonction objectif. Cette fonction objectif permet d'optimiser la longueur de fer, le coefficient de remplissage des encoches ou la puissance utile ou une combinaison de ces paramètres de sortie en fonction des valeurs qu'on donnera à coefl , coefcrs ou coefpu .

$$f_{\text{ob}} := \text{coefl} \cdot L + \text{coefcrs} \cdot \text{crs} + \text{coefpu} \cdot p_u \quad (\text{eqfob})$$

Calcul de la densité de courant au rotor:

$$k_{\text{ci}} := \frac{n_s \cdot k_b \cdot m}{n_{\text{sr}} \cdot k_{\text{br}} \cdot n_r} \quad (\text{eqkci}) \quad (\text{Voir formule 19-12c de [Kostenko]. Cette formule permet de ramener les courants absorbé au stator à leur valeur dans le rotor.})$$

$$j_{\text{rs}} := \frac{i_b}{s_{\text{cr}}} \cdot k_{\text{ci}} \quad (\text{eqjrs})$$

Calcul des pertes joules au stator:

$$p_{\text{jous}} := 3 \cdot r_a \cdot i_a^2 \quad (\text{eqpjous})$$

Calcul des pertes joules au rotor:

$$p_{\text{jour}} := 3 \cdot g_m \cdot r_{\text{bag}} \cdot i_b^2 \quad (\text{eqpjour})$$

Calcul des pertes fer:

$$z_s := \frac{i \cdot x_{mk} \cdot z_b}{x_{mk} \cdot i + z_b} \quad (\text{eqzs})$$

$$i_m := \frac{|z_s|}{|z_s + r_m|} \cdot i_a \quad (\text{eqim})$$

$$p_{fer} := 3 \cdot r_m \cdot i_m^2 \quad (\text{eqpfer})$$

Bibliographie:

[Alger] - Philip L. Alger, "Induction Machines", ed. Gordon and Breach Science Publishers - 150 Fifth Avenue, New-York, 518 pages, 1970.

[Cochran] - Paul L. Cochran, "Polyphase Induction Motors - Analysis, Design, and Application", ed. MARCEL DEKKER, 675 pages, 1989, ISBN 0-8247-4.

[Christofides 65] - Christofides, "Origins of load losses in induction motors with cast aluminium rotors", IEE Proceedings, vol. 112, december 1965, pp. 2317-2332.

[Jong] - H. C. J. de Jong, "Skew Leakage in Induction Machines", ETEP Vol. 4, No. 1, January/February 1994, pp. 43 - 46.

[Kostenko] - M. Kostenko et L. Piotrovski, "Machines Electriques - TOME II", éditions Mir - Moscou 1969.

[TAIEB] - Taieb Brahim Abdelhalim, "Contribution à la modélisation de la machine asynchrone à cage par logiciel d'éléments finis 2D et 3D", Thèse de Docteur, Institut National Polytechnique de Grenoble, juin 1992.

[TI] - André Belot, "Calcul des fuites et inductances de fuites de l'induit", issu des Techniques de l'ingénieur, page D440-1 D440-16.

[VASSENT] - Vassent Eric, "Contribution à la modélisation des moteurs asynchrones par la méthode des éléments finis", Thèse de Docteur, Institut National Polytechnique de Grenoble, Novembre 1990.

**ANNEXE C: une application de dimensionnement pour machines
asynchrones au paramétrage exhaustif - capacité de l'approche
PASCOSMA à dimensionner une machine complète**

ANNEXE C: une application de dimensionnement pour machines asynchrones au paramétrage exhaustif - capacité de l'approche PASCOSMA à dimensionner une machine complète

Dans le cadre du dimensionnement de machines asynchrones nous avons réalisé une première application dont l'objectif a été de vérifier que notre approche reste robuste face à un problème très fortement paramétré, typique du dimensionnement complet d'une machine.

Cette application est évoquée dans le paragraphe IV.1.c. où l'on montre les aspects relatifs à la génération. On rappellera le plus important d'entre eux: malgré le nombre important de paramètres (*57 paramètres d'entrée et 42 paramètres de sortie*), le prototype PASCOSMA a assuré la programmation automatique.

La présente annexe va servir:

- à détailler un exemple de dimensionnement qui a été traité avec elle afin de montrer que une fois générée, une application aussi fortement paramétrée est en mesure d'assurer un dimensionnement en jouant sur un nombre représentatif de paramètres,
- à mettre en évidence certaines de ses propriétés particulièrement intéressantes dans le cadre de la problématique du dimensionnement assisté par ordinateur.

La fonction objectif utilisée:

La fonction objectif est égale à la somme réduite du coefficient de remplissage des encoches ajouté au volume réduit de la machine. On va donc chercher à diminuer simultanément ces deux grandeurs. Ceci peut représenter ce que peut être une machine peu chère pour un fabricant:

une machine nécessitant le minimum de matière première, avec un coefficient de remplissage minimisé afin de raccourcir le temps de bobinage.

Le coefficient de bobinage est par ailleurs contraint dans un intervalle (*à savoir de 0,35 à 0,55*). Aussi le fait de vouloir le minimiser n'avait jamais pour effet de le faire tendre vers des valeurs absurdes (*tendant vers 0 notamment*), mais simplement de le rapprocher de la valeur minimale de 0,35.

Le cahier des charges traité:

Nous n'allons pas rentrer dans le détail des valeurs numériques et des contraintes posées sur tous les paramètres d'entrée et de sortie, ce qui serait long et non fondamental pour notre propos. Nous allons simplement évoquer les aspects les plus intéressants en terme de conception.

Dans le cahier des charges traité, nous avons autorisé un certain nombre de paramètres d'entrée à varier. Ce sont:

* *Des paramètres de définition de la tôle tels que:*

- 'deext', le diamètre extérieur de la machine,
- et 'di' le diamètre intérieur du stator (cf. figure C.1).

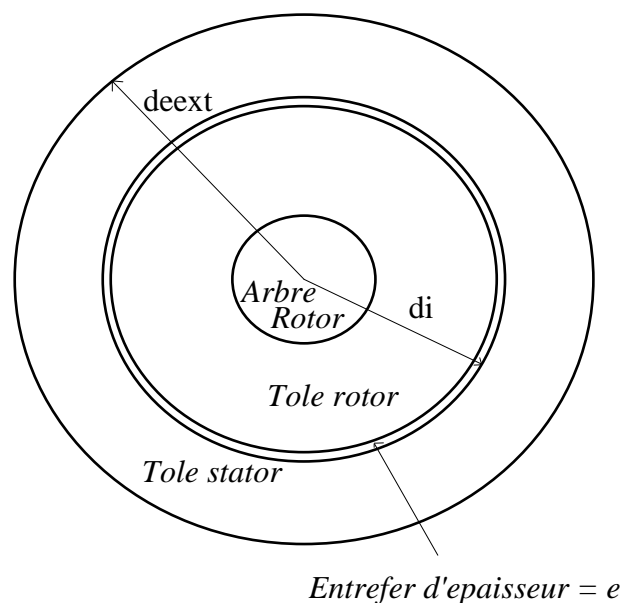


Figure C.1: diamètres autorisés à varier au cours du dimensionnement

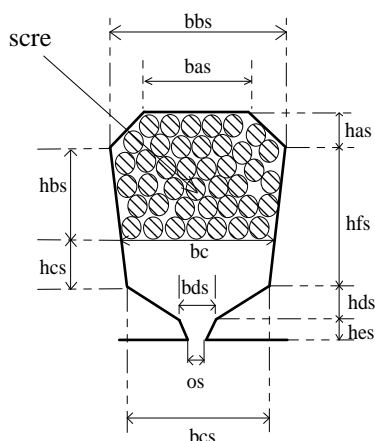


Figure C.2: encoche stator avec ses conducteurs

* *Tous les paramètres géométriques de définition d'une encoche stator,*

soit 'bas', 'bbs', 'bcs', 'bds', 'os', 'has', 'hfs', 'hds', 'hes' (cf. figure C.2).

* *L'entrefer 'e' de la machine.*

* *La longueur de fer 'l' de la machine.*

* *Des paramètres relatifs au bobinage:*

- 'dfila', le diamètre de fil utilisé pour le bobinage,
- 'nsspe' le nombre de spires par encoche.

* *Un paramètre d'utilisation nominale de la machine:*

- 'g' le glissement.

Soit en tout 16 paramètres d'entrée autorisés à varier sur les 57 possibles.

Les paramètres d'entrée qui ont été laissés fixes sont, entre autres, les paramètres de la géométrie de l'encoche rotor. Ont été aussi figés des paramètres pour lesquels des valeurs entières sont obligatoires tels que:

- le nombre de paires de pôles,
- ou encore le nombre d'encoches.

Les paramètres de sortie dont l'évolution a été contrainte dans un intervalle sont:

* *Tous les paramètres de sortie représentant une valeur d'induction dans les différentes parties de la tôle de la machine.*

Sont ainsi contraints à varier dans un intervalle 'bcsa', 'bdsa', 'bdsb', 'bdsc', 'bdse', 'bcra', 'bdra', 'bdrb', 'bdrc', 'bdre'. La figure C.3 indique la section de la tôle sur laquelle ces différentes inductions ont été calculées.

* *L'induction dans l'entrefer 'beent'.*

* *Les densités de courant.*

Sont ainsi contraintes 'js' représentant la densité de courant au stator et 'jrs' représentant la densité de courant au rotor.

* *Des paramètres de sortie représentatifs des performances de la machine:*

Sont ainsi contraints:

- 'pu' la puissance utile,
- 'cosphi' qui représente le facteur de puissance,
- 'crs' qui représente le coefficient de remplissage des encoches,
- 'r' qui représente le rendement.

* *Les hauteurs équivalentes de conducteur dans l'encoche:*

- ces grandeurs sont 'hbs' et 'hcs'. Elles représentent respectivement la hauteur équivalente de conducteur et de vide conformément au schéma de la figure C.2. Ces grandeurs sont contraintes dans un intervalle compris entre 0 et 1 m, simplement pour éviter que le système ne trouve des solutions avec des hauteurs négatives qui seraient physiquement absurdes.

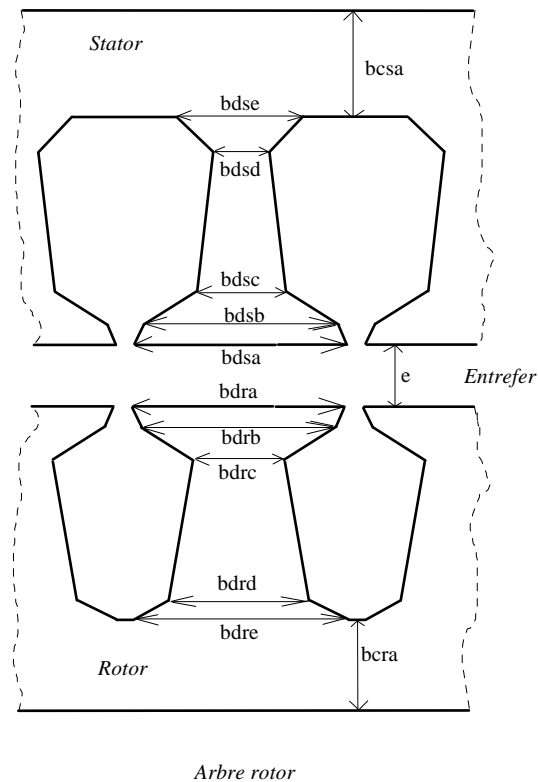


Figure C.3: représentation schématique et linéaire de la tôle rotor et stator sur 2 encoches ; section de passage des différentes inductions calculées dans le logiciel de dimensionnement

21 paramètres de sortie sur les 42 possibles sont ainsi contraints.

Les contraintes exactes qui ont été posées sur chacun de ses paramètres de sortie sont reproduites sur la figure C.4.

On notera que l'objectif du cahier des charges posé est de trouver une machine de puissance utile $p_u=750$ W. La machine qui a servi de point de départ, et qui a donc servi à fournir une valeur initiale aux paramètres d'entrée, est une machine existante dont la caractéristique principale était de fournir une puissance utile aux alentours de 1000 W.

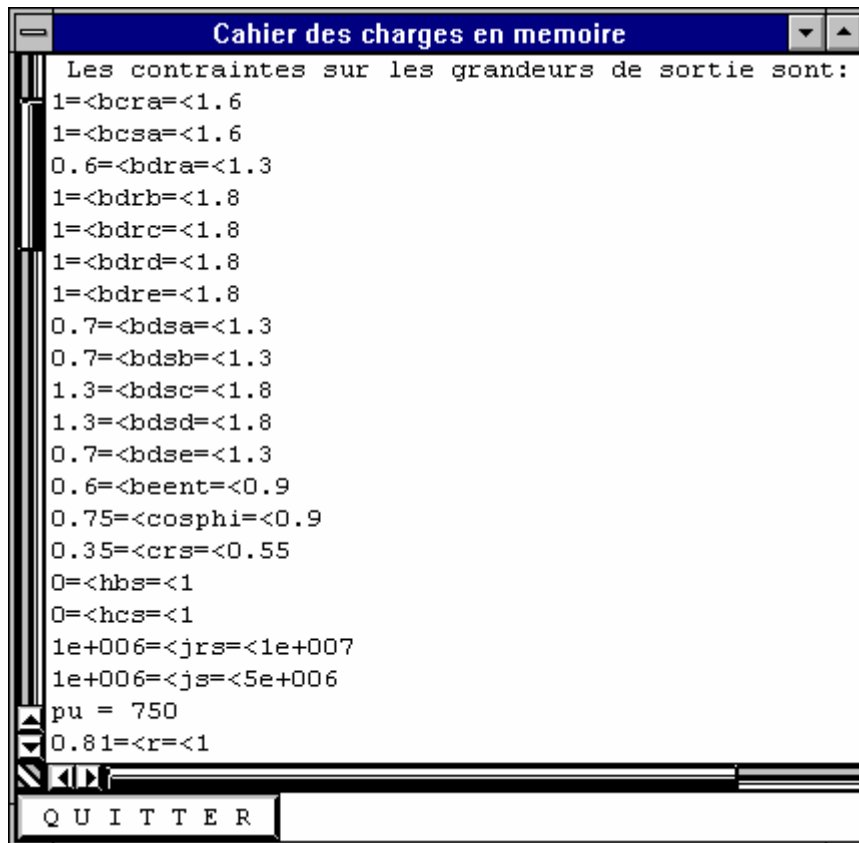


Figure C.4: contraintes sur les paramètres de sortie

Le résultat obtenu:

Face à un cahier des charges tel que celui-ci, nous noterons trois propriétés intéressantes en terme de dimensionnement, dont le logiciel de dimensionnement fait preuve:

1° *La rapidité pour traiter un tel cahier des charges* puisque, sur un ordinateur de type PC pentium 75 Mhz, le temps nécessaire pour trouver une solution peut être variable en fonction du nombre d'itérations que le système est amené à faire, mais il ne dépassera jamais 2 à 3 dizaines de secondes. Ceci est d'ailleurs un maximum rarement atteint. Ainsi, pour le cahier des charges précédent, le temps de dimensionnement est exactement de 3,75 s pour 9 itérations réalisées.

2° *La capacité de trouver automatiquement ce qu'il faut faire en fonction du cahier des charges qui est posé.*

Pour trouver une solution qui soit compatible avec toutes les contraintes (*cf. entre autres la puissance utile 'pu' et le rendement 'r' dont les évolutions au cours du dimensionnement sont représentées sur la figure C.5*) le système a, au fur et à mesure des itérations, pris les initiatives qu'aurait prises un expert de ces machines.

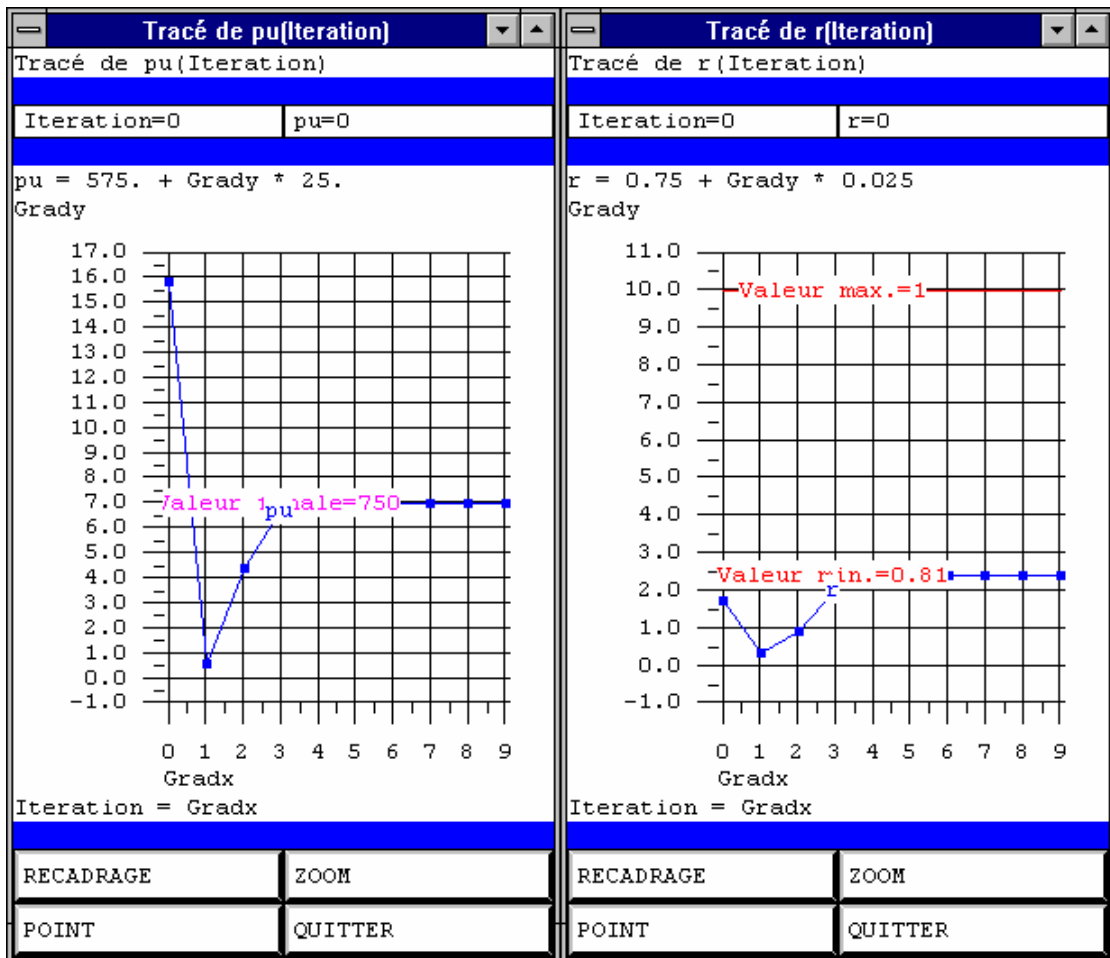


Figure C.5: évolution de la puissance utile 'pu' et du rendement 'r' au cours du dimensionnement¹⁶

Ainsi, comme l'illustre la figure C.6, le système s'est orienté vers une machine ayant un entrefer 'e' minimal dans l'intervalle autorisé. Ce choix est cohérent avec la pratique d'un expert électrotechnicien qui s'oriente toujours vers la valeur d'entrefer minimale autorisée car:

- cela contribue à diminuer le courant nécessaire pour magnétiser la machine,
- ce qui permet de diminuer les pertes Joule,
- entraînant une amélioration du rendement qui en l'occurrence était contraint à être au dessus de 0.81.

Des exemples de ce type pourraient être multipliés à propos de cette application et contribueraient à montrer que le logiciel de dimensionnement trouve des solutions qui sont cohérentes avec les règles de l'art.

¹⁶: Attention aux échelles des graphes qui sont visualisés sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais un coefficient multiplicateur et un décalage doivent être pris en compte comme cela est précisé à côté de ces échelles.

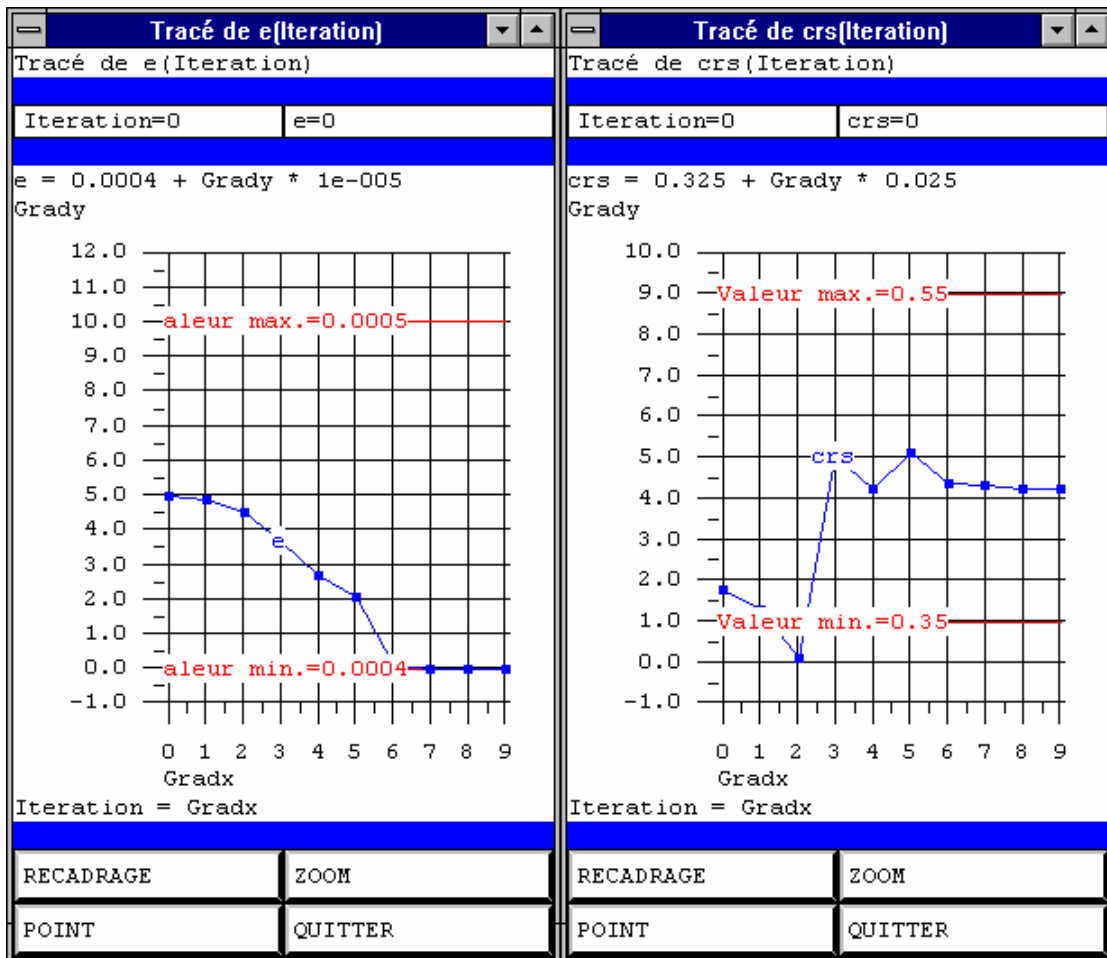


Figure C.6: évolution de la valeur de l'entrefer 'e' et du coefficient de remplissage des encoches 'crs' au cours du dimensionnement¹⁷

3° La gestion de l'interdépendance des grandeurs les unes avec les autres: Le comportement du coefficient 'crs' de remplissage des encoches, représenté à la figure C.6, est tout à fait typique de cette propriété. Rappelons que ce coefficient fait partie de la fonction objectif et qu'une mission du système est de le diminuer autant que possible. Pourtant, on voit que pour l'itération finale, il ne se stabilise pas sur sa valeur minimale de 0,35, mais se stabilise, au contraire, à une valeur plus élevée qui vaut 0,43. En fait, le système a bien minimisé ce coefficient de remplissage mais à une valeur qui soit compatible avec le rendement exigé. En effet, une valeur faible du coefficient de remplissage revient à diminuer la section de cuivre disponible pour faire passer le courant, ce qui augmente les pertes Joule et dégrade le rendement.

Concernant les deux derniers aspects soulignés, on insistera sur le fait qu'il n'y a pas eu besoin de mettre dans le système des règles afin d'indiquer ce qu'il faut modifier, ou quels

¹⁷: Attention aux échelles des graphes qui sont visualisés sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais un coefficient multiplicateur et un offset doivent être pris en compte comme cela est précisé à côté de ces échelles.

sont les paramètres qui vont brider l'évolution d'autres paramètres en fonction du cahier des charges à atteindre. Le savoir-faire nécessaire est retrouvé automatiquement et implicitement.

Conclusions de l'étude:

Cet exemple permet de montrer ce à quoi nous voulions arriver: *notre approche est parfaitement en mesure d'assurer la génération du code, puis la recherche de solutions dans le cadre d'un problème représentatif de ce que peut être le dimensionnement complet d'une machine électrique.* En effet, cet exemple nous a permis de vérifier que ni un fort paramétrage, ni le fait d'utiliser un modèle complexe, n'ont été des obstacles au bon fonctionnement.

Pour conclure sur cet exemple, et à titre indicatif, on donnera encore sur la figure C.7, des schémas de l'évolution de la forme de l'encoche et du circuit magnétique entre la machine fournie comme point de départ et la machine finale trouvée dans le problème détaillé précédemment.

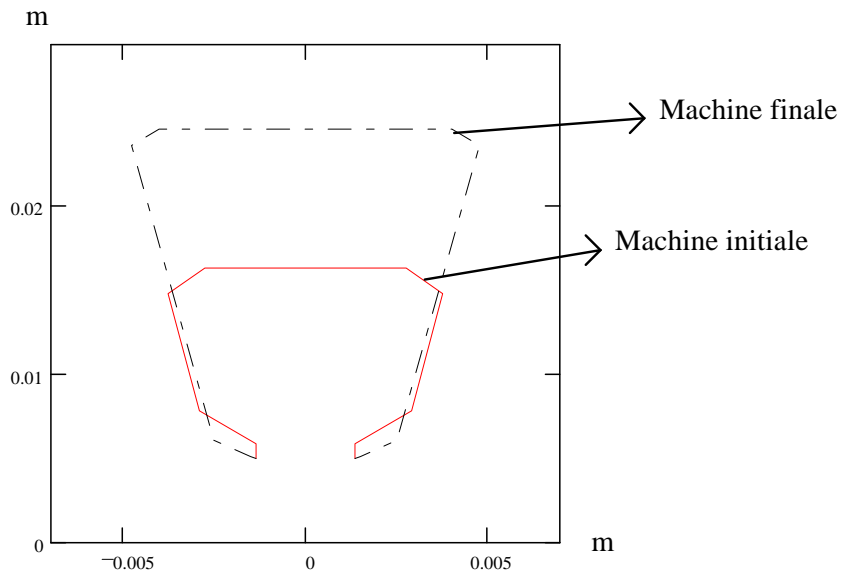


Schéma de l'évolution de la forme de l'encoche stator

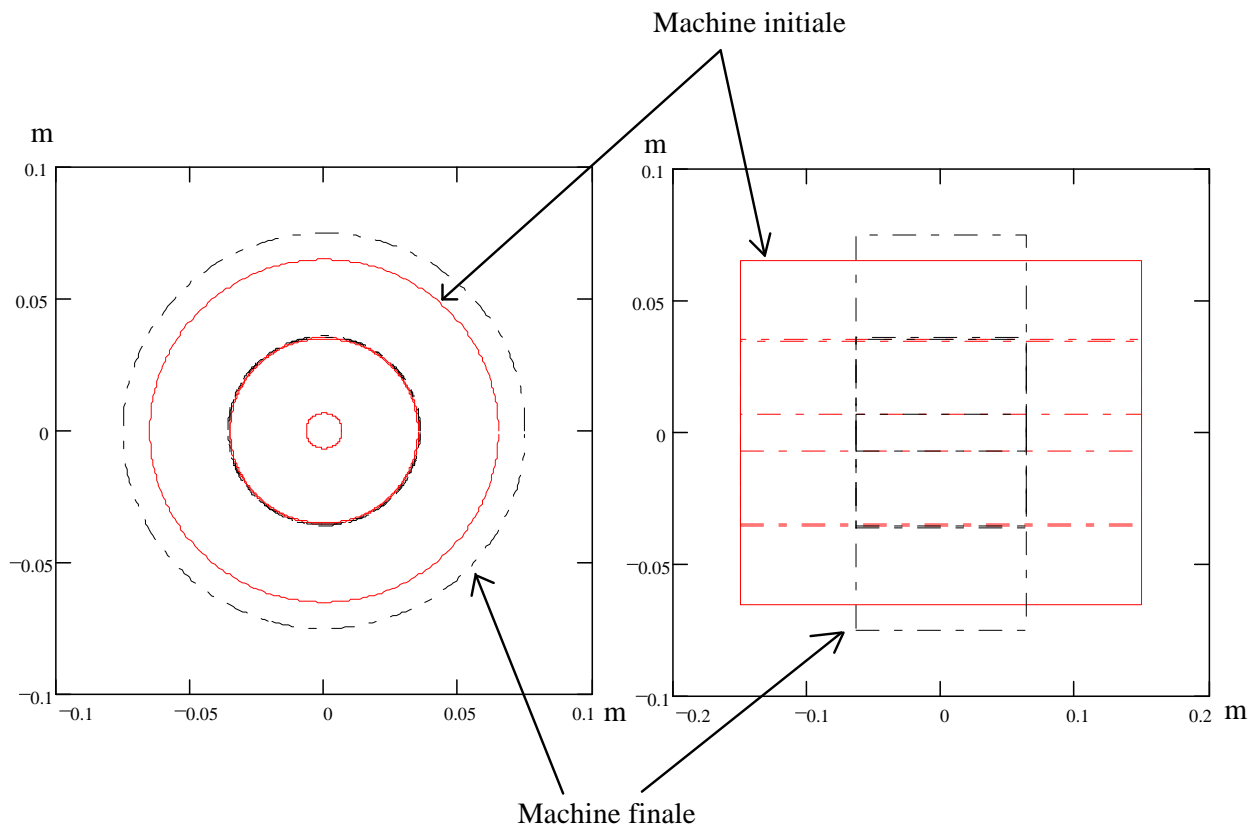


Schéma de l'évolution de la forme du circuit magnétique
(représenté sans les encoches)

Figure C.7: évolution entre la machine initiale et la machine finale dans le cadre du cahier des charges de la figure C.4

**ANNEXE D: une application de dimensionnement pour machines
asynchrones destinée à un usage en bureau d'études**

ANNEXE D: une application de dimensionnement pour machines asynchrones destinée à un usage en bureau d'études

Une de nos préoccupations, lors de notre travail, a été de démontrer que notre approche peut contribuer à améliorer et à accélérer sensiblement le dimensionnement dans un bureau d'études.

C'est pourquoi nous avons réalisé une application de dimensionnement de machines asynchrones avec les objectifs suivants:

- une vérification et une fiabilisation du modèle analytique employé avec une machine de référence,
- une vérification de l'intérêt et de la validité des solutions que notre système peut trouver.

C'est cette étude de faisabilité et ses conclusions que nous allons détailler dans les paragraphes qui vont suivre. Celle-ci est constituée de cinq parties:

- 1° la fiabilisation du modèle structural,
- 2° la génération de l'application de dimensionnement,
- 3° la fonction objectif employée,
- 4° l'utilisation de l'application de dimensionnement en vue de déterminer un point de fonctionnement optimal pour le prototype,
- 5° le dimensionnement d'une nouvelle machine de même puissance que celle annoncée pour le prototype.

On rappellera que nous donnons des informations supplémentaires sur cette application dans le paragraphe IV.1.d. (*gamme de machines à laquelle nous nous intéressons, instrumentation du prototype utilisé,...*) et que, par ailleurs, l'annexe B détaille le modèle structural complet employé.

La fiabilisation du modèle:

La fiabilisation expérimentale a été faite en utilisant des résultats d'essais provenant d'une machine appartenant à la gamme de machines que nous voulons dimensionner. Cette machine possède les caractéristiques suivantes:

Type de machine:	moteur asynchrone triphasé fermé
Ventilation:	sans ventilation
Hauteur d'axe:	75 mm
Longueur de fer du circuit magnétique:	150 mm
Fréquence d'alimentation:	50 Hz

Nombre de pôles:	2
Vitesse de synchronisme:	3000 tr/mn

Le constructeur avait prévu que cette machine était en mesure de fournir:

une puissance utile de 1000W en régime permanent sous une tension d'alimentation de
230 V par phase

Cette machine a fait l'objet de toute une série d'essais afin d'en caractériser complètement le fonctionnement, autant d'un point de vue thermique que d'un point de vue électromagnétique. Dans une perspective de dimensionnement de gamme, ces essais ont d'abord permis l'identification d'un certain nombre de valeurs qu'il est très difficile d'évaluer a priori. Nous avons ainsi identifié:

- un coefficient de frottements pour les pertes mécaniques,
- une valeur de résistance pour modéliser les pertes fer à vide. Celle-ci fait ensuite l'objet d'une loi d'extrapolation pour disposer de l'évolution de ces pertes.

L'annexe B détaille ces éléments.

Nous avons ensuite utilisé ces essais pour vérifier que le modèle structural fournissait des résultats suffisamment fiables afin de permettre le dimensionnement.

Ainsi, la figure D.1 compare la caractéristique du couple en fonction du glissement du moteur, telle que nous l'avons mesurée sur le prototype et telle qu'elle est donnée par le modèle structural. Ces résultats sont parfaitement en accord avec ce que nous pouvions espérer avec notre modèle:

- la zone de fonctionnement autour du point nominal (*à savoir pour $g=0$ à $g=0,2$*) est correctement prévue,
- par contre la caractéristique pour des points de fonctionnement près du démarrage du moteur est nettement moins bien prédite (*à savoir pour $g=0,3$ à $g=1$*). Or, comme nous l'avons évoqué dans le paragraphe IV.1.b.1., le modèle que nous avons pris pour faire notre étude ne prend pas en compte certains phénomènes qui deviennent prépondérants lors du démarrage, tels que l'apparition de couples harmoniques ou la saturation des inductances de fuite.

La figure D.2 compare la caractéristique du courant absorbé à vide par la machine en fonction de la tension d'alimentation, telle qu'elle est mesurée sur le prototype et telle qu'elle est donnée par le modèle structural. Ce courant représente essentiellement le courant de magnétisation et l'on voit que le modèle reproduit de manière satisfaisante le phénomène de saturation (*ceci grâce au calcul du coefficient de saturation que nous y avons introduit*).

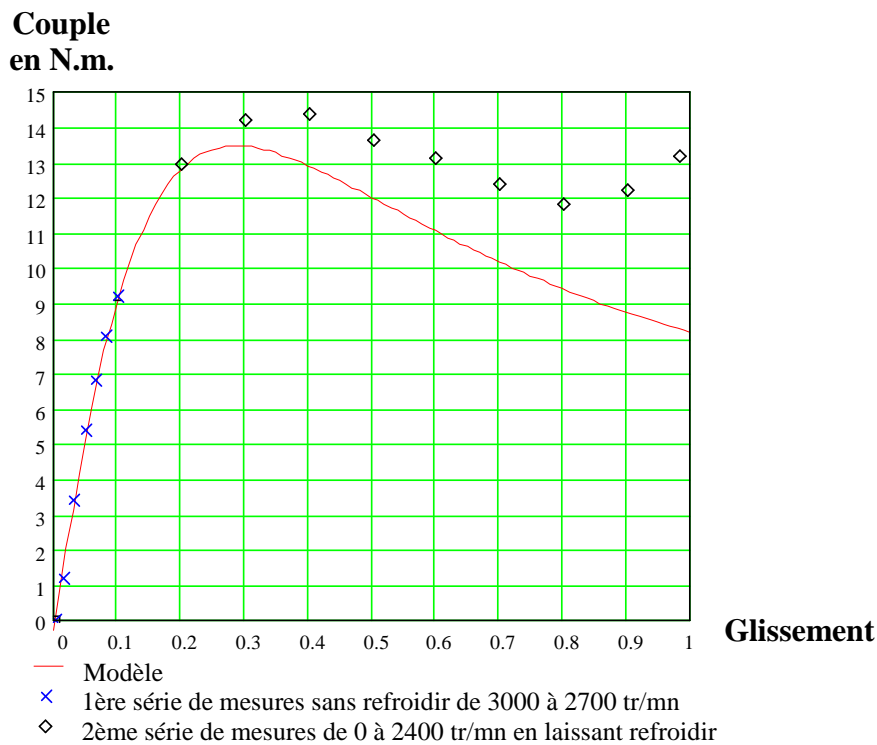


Figure D.1: caractéristique du couple en fonction du glissement, $f=50$ Hz, couplage triangle, $V=235$ V entre 2 phases - comparaison entre modèle structural et résultat d'essais

Toutes les autres confrontations entre résultats d'essais et résultats provenant du prototype, que nous avons faites, mais que nous ne détaillerons pas ici, vont dans le même sens que ce que nous montrent les figures D.1 et D.2:

le modèle prévoit correctement le comportement du moteur pour les glissements autour du point nominal et pour le fonctionnement à vide, mais est nettement moins fiable pour les glissements proches du démarrage.

On en déduit que le modèle est suffisant pour ce que voulons réaliser dans le cadre de notre étude:

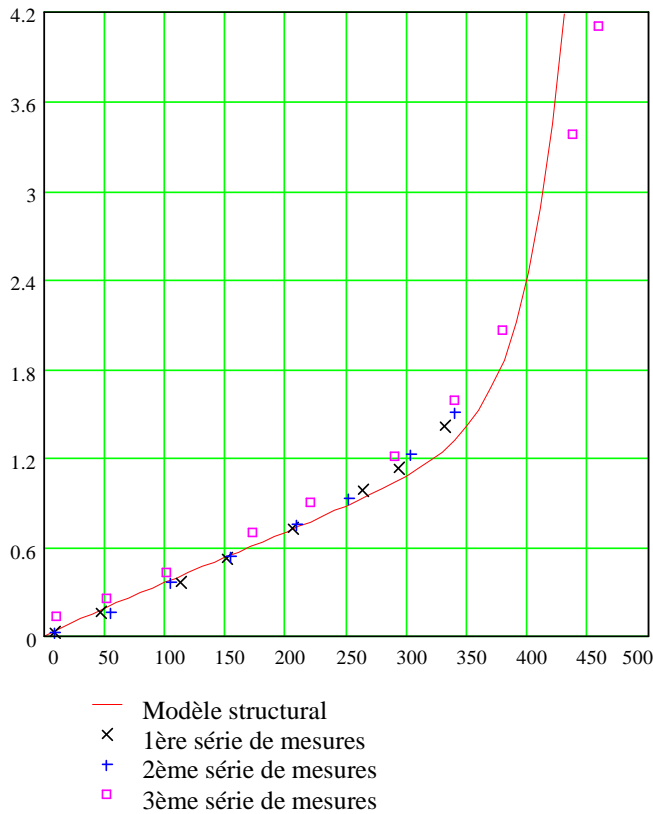
dimensionner des machines appelées à fonctionner autour de leur point nominal.

Remarque concernant l'amélioration du modèle structural:

L'objet de l'exemple qui est détaillé ici n'est pas d'améliorer le comportement du modèle au démarrage. Cependant, le jour où l'on s'intéressera à cette question, on pourra utiliser les possibilités de calcul de sensibilité qu'offre le logiciel de dimensionnement du type PASCOSMA. Celles-ci peuvent en effet aider à déterminer les valeurs qui ont la plus grande

influence sur les valeurs mal calculées et peuvent permettre de déterminer les éléments du modèle à améliorer.

**Courant efficace
absorbé par bobinage
en A**



**Tension d'alimentation
entre 2 phases
en V**

Figure D.2: caractéristique du courant absorbée à vide ($g=0$) en fonction de la tension d'alimentation, couplage triangle - comparaison entre modèle structural et résultat d'essais

Génération de l'application de dimensionnement:

La génération de l'application s'est faite avec un nombre restreint de paramètres d'entrée et de sortie. Nous nous intéressons ici, en effet, au dimensionnement dans le cadre d'une gamme de moteurs. Il en résulte que:

- * d'un moteur à l'autre, beaucoup de paramètres sont figés,
- * ces paramètres ont aussi été figés dans l'application de dimensionnement générée.

Ceci signifie que:

- pour chacun de ces paramètres, on a introduit une équation du type $A=\lambda$, où λ est une constante réelle dans le modèle (cf. annexe B),
- le paramètre disparaît alors de la liste des paramètres d'entrée,

- ce qui permet d'alléger le code, puisqu'un symbole est substitué par une valeur numérique ce qui a notamment pour conséquence que les dérivées partielles des paramètres de sortie en fonction de ce paramètre d'entrée n'ont plus à être générées.

L'application de dimensionnement perd en généralité mais gagne en simplicité d'emploi, ce qui est important dans un contexte industriel. Par ailleurs, si on veut s'intéresser à une autre gamme de machines, notre méthodologie permettra de générer l'application spécialisée nécessaire.

Les caractéristiques exactes pour la génération du code de cette application sont:

Nombre de paramètres d'entrée du modèle:	22
Nombre de paramètres de sortie du modèle:	28
Nombre de paramètres intermédiaires:	10

Selon la formule de dénombrement des routines à générer, mise en évidence dans le paragraphe III.3.a., on trouve:

$$22 \times 28 + (22 + 28) \times 10 = 1116 \text{ routines}$$

Dans cette application, on a repris les paramètres du schéma équivalent (*à savoir 'ra', 'xa', 'xm', 'xba' et 'rbag'*) comme paramètres intermédiaires. Cependant, le fait d'introduire le coefficient de saturation dans le modèle nous a obligé à introduire quatre paramètres intermédiaires supplémentaires qui sont 'nig', 'nids', 'nidr' et 'kic'. Ces paramètres interviennent dans le calcul du coefficient de saturation (*cf. annexe B*).

Le temps de génération de l'application de dimensionnement est de l'ordre de 7 heures sur un ordinateur de type PC pentium 75 Mhz. Ce temps, qui inclut les manipulations symboliques, la génération du code en FORTRAN, la compilation et l'édition de lien, n'est pas beaucoup plus court que le temps nécessaire pour générer l'application de dimensionnement dans laquelle aucun paramètre n'avait été figé. (*dans laquelle on a généré 2988 routines en 10 heures cf. annexe C et paragraphe IV.1.c*). Ceci s'explique par le fait que dans le modèle employé ici, nous avons introduit le calcul du coefficient de saturation. Il en résulte que les routines à générer en nombre plus restreint sont malgré tout plus longues et plus difficiles à programmer.

La taille de l'exécutable final de dimensionnement est de l'ordre de 1,5 Mo.

Fonction objectif employée:

L'application de dimensionnement générée possède la fonction objectif suivante:

$$fob = \text{coefl} \times L + \text{coefcrs} \times \text{crs} + \text{coefpu} \times \text{pu}$$

avec:

'L': la longueur de fer du circuit magnétique de la machine,

'crs': le coefficient de remplissage des encoches,

'pu': la puissance utile.

Les coefficients 'coefl', 'coefcrs' et 'coefpu' ont pour rôle de pondérer l'importance que l'utilisateur voudra donner à 'L', 'crs' ou 'pu'.

Ainsi, s'il choisit:

- coefl=0, coefcrs=0 et coefpu=-1, sachant que le logiciel cherche à minimiser la fonction objectif, celui-ci va chercher une machine fournissant le plus de puissance possible,
- coefl=1, coefcrs=0 et coefpu=0, il cherchera à trouver une machine qui sera optimisée du point de vue de sa longueur de fer. Comme dans la gamme de machines sur laquelle nous travaillons, le diamètre de la tôle est figé, cela revient à chercher une machine de volume optimisé.

Utilisation de l'application de dimensionnement en vue de déterminer un point de fonctionnement optimal pour le prototype:

Le premier cahier des charges que nous allons détailler ici a consisté en la recherche d'un point de fonctionnement optimal pour le prototype instrumenté dont nous disposons.

L'objectif est de vérifier que l'application est en mesure:

- de trouver un point de fonctionnement fiable, ce que nous pourrions vérifier expérimentalement puisque nous disposons de la machine,
- de trouver un point de fonctionnement qui optimise de manière intéressante la puissance initiale que l'on pensait pouvoir fournir avec cette machine.

Le point de départ utilisé correspond donc au point de fonctionnement déterminé par le constructeur, à savoir:

une puissance utile de 1000 W, sous une tension de 230/400V en couplage triangle/étoile à une fréquence de 50 Hz avec un circuit magnétique d'une longueur de 150 mm.

On notera que ces caractéristiques nominales affichées n'ont pas du tout été optimisées. Par ailleurs, le prototype dont nous disposons est la première machine de la gamme réalisée sans ventilation. Aussi a-t-elle été dimensionnée avec un coefficient de sécurité important.

Pour optimiser le point de fonctionnement, les paramètres sur lesquels on peut jouer ne sont pas des paramètres de construction, mais des paramètres caractéristiques des conditions dans lesquelles la machine est utilisée. Or, dans notre cas, les seules conditions sur lesquelles on peut jouer sont:

- 'g' le glissement
- 'v' la tension d'alimentation

C'est pourquoi nous avons entré dans le logiciel de dimensionnement le cahier des charges reproduit à la figure D.3 (*concernant la signification de tous ces paramètres, on pourra se reporter à l'annexe B et à la liste des symboles employés*).

Concernant les paramètres de sortie, des contraintes ont été posées sur eux pour:

1° Que le logiciel trouve un point de fonctionnement qui soit intéressant. Ainsi, on impose au facteur de puissance 'cosphi' et au rendement 'r' de rester tous deux dans un intervalle compris entre 0,8 et 1.

2° Que le système trouve un point de fonctionnement non destructif pour la machine. C'est pourquoi les niveaux de pertes Joule au stator et au rotor ont été respectivement limités entre 0 et 83W et 0 et 50W. L'ensemble des essais que nous avons menés nous ont en effet appris que ce sont principalement ces deux paramètres qui conditionnent l'échauffement des points sensibles de la machine. Les valeurs limites que nous leur imposons ici sont celles dont on a pu établir expérimentalement que:

- Elles provoquent un échauffement proche de la limite maximale de 115 °C que l'on s'autorise pour le point d'échauffement le plus élevé du bobinage stator, à savoir les têtes de bobines. Avec une température ambiante de 20°C, on estime ainsi, de manière un peu pessimiste, la valeur de la température du bobinage stator à 135°C (*cf. valeur de 'tsta' sur la figure D.3*).
- Elles provoquent un échauffement proche de la limite maximale de 120°C que l'on s'autorise pour la température du rotor soit une température de 140°C avec une température ambiante de 20°C (*cf. valeur de 'trot' dans figure D.3*).

3° Que le programme trouve un point de fonctionnement qui ne sature pas trop la machine. C'est pourquoi on impose au coefficient de saturation de la machine 'ksat' d'être compris entre 1 et 1,5, et à l'induction dans l'entrefer 'beent' d'être comprise entre 0,5 et 0,9 Tesla. Il faut éviter que le point de fonctionnement déterminé par le système ne sature trop la machine et crée ainsi un niveau de pertes fer qui viendrait perturber nos limites de pertes Joule, stator et rotor, autorisées.

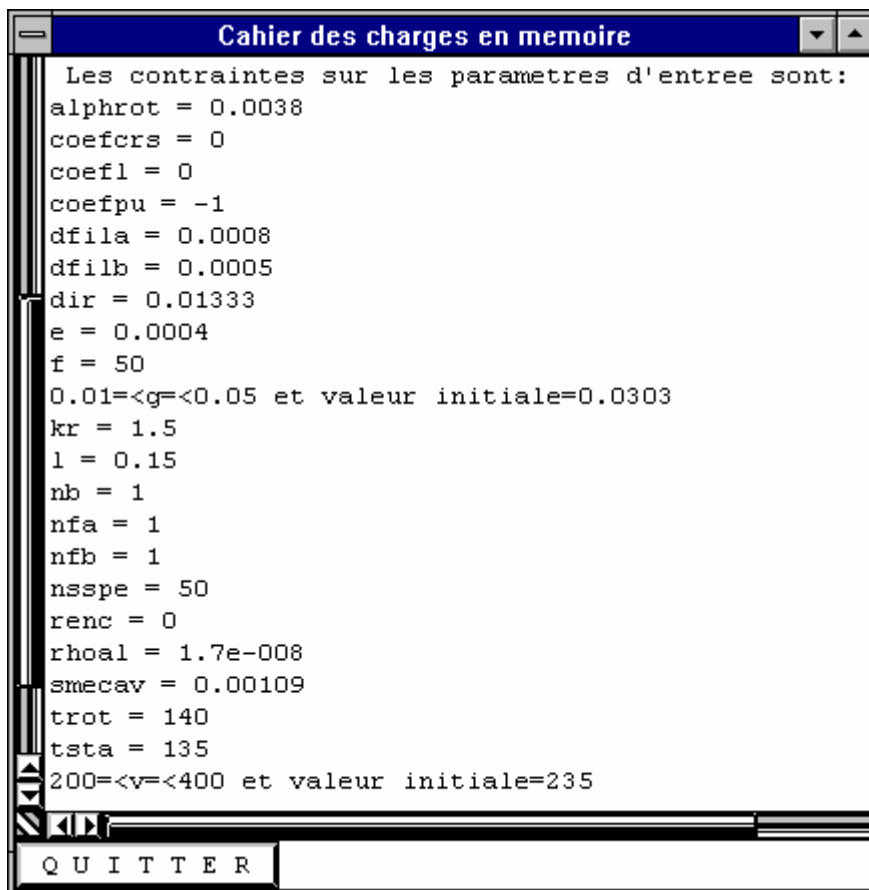
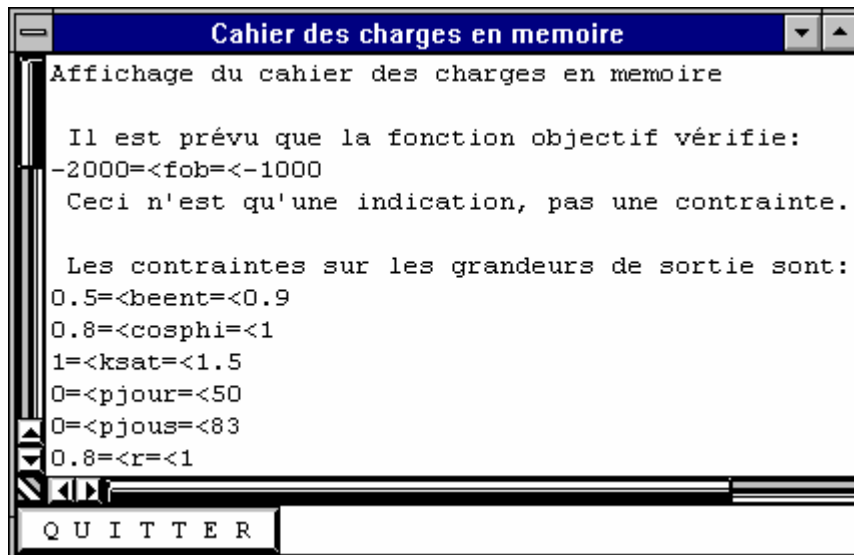


Figure D.3: cahier des charges pour chercher un point de fonctionnement optimal pour la machine prototype

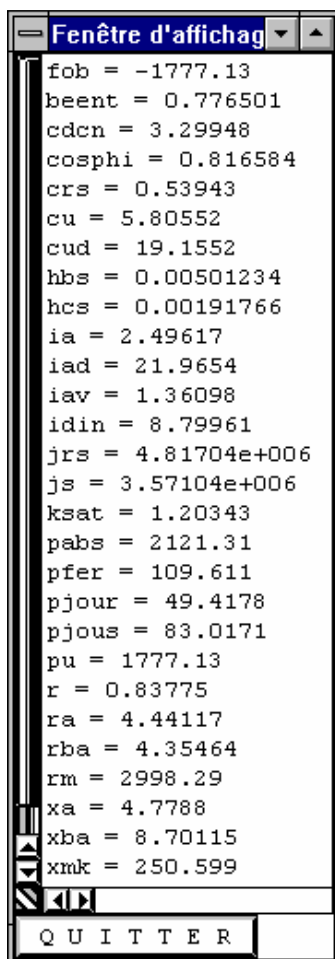
Dans ces conditions, le logiciel de dimensionnement détermine en 5 itérations, soit une à deux secondes de calcul, le point de fonctionnement dont les caractéristiques principales sont les suivantes:

$g=0,025$ soit une vitesse de 2925 tr/mn

$v=346$ V (qu'on arrondira par la suite à 350 V)

$pu=1777$ W (qu'on arrondira par la suite à 1750 W)

Les valeurs complètes des paramètres de sortie déterminées par le logiciel pour ce point de fonctionnement sont données sur la figure D.4. On peut vérifier que toutes les contraintes du cahier des charges sont respectées.



```
Fenêtre d'affichag
fob = -1777.13
beent = 0.776501
cdcn = 3.29948
cosphi = 0.816584
crs = 0.53943
cu = 5.80552
cud = 19.1552
hbs = 0.00501234
hcs = 0.00191766
ia = 2.49617
iad = 21.9654
iav = 1.36098
idin = 8.79961
jrs = 4.81704e+006
js = 3.57104e+006
ksat = 1.20343
pabs = 2121.31
pfer = 109.611
pjour = 49.4178
pjous = 83.0171
pu = 1777.13
r = 0.83775
ra = 4.44117
rba = 4.35464
rm = 2998.29
xa = 4.7788
xba = 8.70115
xmk = 250.599
QUITTER
```

Le point de fonctionnement déterminé fournit une puissance de 1750 W, soit 75 % de plus que la puissance initiale prévue pour cette machine.

Il est vrai cette dernière était sous-évaluée et devait faire l'objet d'une optimisation. Notre outil a montré qu'il était parfaitement adapté pour faire cela, d'autant plus que la validité du point de fonctionnement déterminée a été vérifiée expérimentalement.

Cette confrontation est parfaitement concluante comme l'indiquent:

- le tableau D.1 qui fournit la comparaison entre les performances prévues par le logiciel et les performances mesurées au cours de l'essai,
- le tableau D.2 qui montre que les échauffements mesurés pour le fonctionnement en régime permanent sont, conformes à ce que nous souhaitions atteindre: être proche des limites permises sans les dépasser.

Figure D.4: valeur des paramètres de sortie du modèle pour le point de fonctionnement déterminé

	Essai	Performances prévues par le logiciel de dimensionnement
Couple utile (En N.m)	5.6	5,8
Tension d'alimentation par phase	350 V (<i>Imposée</i>)	346 V
Glissement	0,024	0.025
Courant par phase (en A)	2,62	2,49
Puissance utile (en W)	≈ 1750 (<i>Imposée</i>)	1777
Puissance absorbée (en W)	2121	2121
Facteur de puissance	0,785	0,81
Rendement (en %)	82	83

Tableau D.1: comparaison des performances mesurées et celles prévues par le logiciel de dimensionnement pour le point de fonctionnement optimal déterminé

	Température mesurée	Echauffement résultant (Température ambiante = 21 °C)	Echauffement maximal autorisé
Têtes de bobines stator <i>=point le plus chaud du bobinage stator</i>	133,9 °C (='tsta')	112,9°C	115°C
Rotor	136 °C (='trot')	115°C	120°C

Tableau D.2: températures des points sensibles de la machine atteintes en régime permanent pour le point de fonctionnement optimal déterminé

Commentaire à propos du résultat obtenu:

Il est clair que pour trouver ce point de fonctionnement optimal, le logiciel a réalisé un processus d'ordre mathématique qui relève de l'algorithme d'optimisation qu'il utilise et qui n'est pas un raisonnement d'ordre électrotechnique. Il n'en reste pas moins, qu'encore une fois, la solution trouvée est parfaitement cohérente, d'un point de vue électrotechnique, et s'interprète à partir des considérations suivantes:

* Toute augmentation de la tension d'alimentation permet d'augmenter la puissance que va fournir la machine. Cependant cette augmentation se traduit aussi:

- par une augmentation du courant magnétisant qui va circuler dans le bobinage,
- qui va elle-même se traduire par une augmentation des pertes Joule générées.

Ce phénomène sera en plus fortement accentué lorsque l'on rentrera dans la zone de saturation de la machine qui, pour notre prototype, se situe aux alentours de 350 V selon la figure D.2. (*dans ce raisonnement, on néglige l'effet des pertes fer car les essais que nous avons réalisés nous montrent que pour les niveaux d'induction que l'on s'autorise dans l'entrefer, i.e. $0,5 \leq beent \leq 0,9$, celles-ci n'influent pas de manière prépondérante sur les températures des points sensibles de la machine que sont les têtes de bobines ou le rotor*).

* Toute augmentation du glissement se traduit aussi par une augmentation de la puissance utile que va fournir la machine. Cependant elle induit:

- une circulation de courant plus élevée, à la fois dans le bobinage stator et dans la cage rotor,
- qui entraînera une augmentation des pertes Joule à la fois au rotor et au stator.

Sachant que les pertes Joule sont limitées au rotor et au stator, le problème consiste à déterminer le bon couple de valeurs (*tension, glissement*) qui va fournir le maximum de puissance utile en utilisant au mieux le "crédit" de pertes autorisées.

C'est très exactement ce dosage optimal que le logiciel a déterminé. On notera que pour le point déterminé:

- la tension d'alimentation est de de l'ordre de 350 V, c'est à dire la tension limite avant la zone de saturation,
- le rendement du moteur est amélioré comme l'illustre la figure D.5, ce qui ajouté au fait que la puissance fournie est 1,75 fois plus grande rend la solution déterminée d'autant plus intéressante.

La précision (*confirmée par l'expérience*) et la vitesse avec laquelle le logiciel a trouvé ce point de fonctionnement optimal montre incontestablement le potentiel de notre approche.

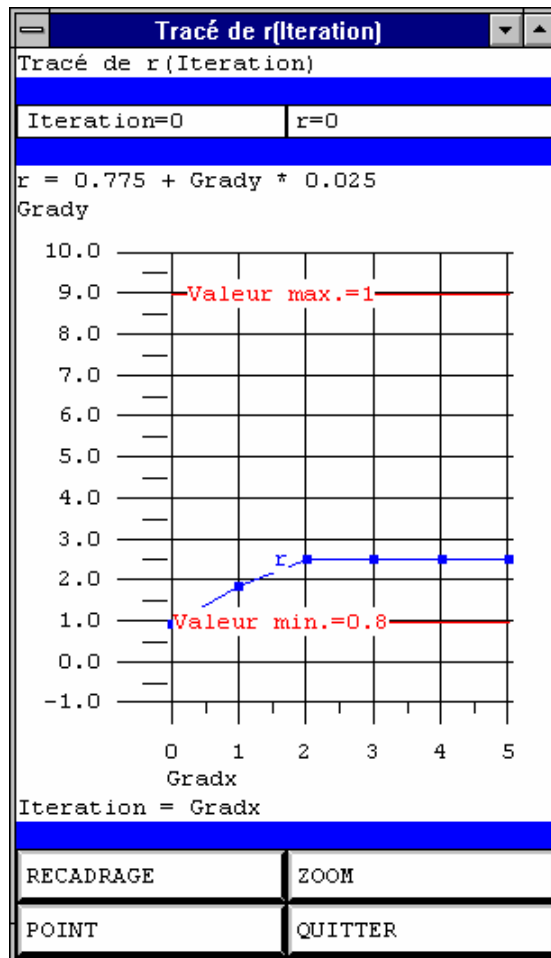


Figure D.5: évolution du rendement lors de la recherche du point de fonctionnement optimal¹⁸

Dimensionnement d'une nouvelle machine de même puissance que celle annoncée pour le prototype:

L'objectif est à présent de dimensionner une machine qui a les mêmes performances que celles initialement annoncées pour notre prototype instrumenté. Sachant que ce prototype est en mesure de fournir plus de puissance que celle qu'on lui attribuait, il semble légitime de penser que cette puissance peut-être fournie par une machine plus compacte.

Dans le cadre d'un processus de dimensionnement de machines d'une même gamme, les seuls paramètres de construction que l'on s'autorise à modifier sont:

- 'nsspe' le nombre de spires par encoche,
- 'dfila' le diamètre de fils utilisé,
- et 'l' la longueur de fer.

¹⁸: Attention aux échelles du graphe qui sont visualisés sur cette figure: les valeurs des échelles ne correspondent pas directement aux valeurs, mais un coefficient multiplicateur et un décalage doivent être pris en compte comme cela est précisé à côté de ces échelles.

Il en résulte que la machine la plus compacte sera celle ayant la longueur de fer la plus petite (*la carcasse reste la même entre toutes les machines de la gamme*).

Les paramètres d'utilisation de la machine que l'on autorise à varier est 'g', le glissement,

Quant à 'v', la tension d'alimentation; on va la figer à 230V afin de pouvoir utiliser le moteur sur un réseau 230/400V.

Tous ces paramètres sont des paramètres d'entrée.

Concernant les paramètres de sortie, des contraintes sont posées:

* Sur les paramètres suivants afin d'obtenir un point de fonctionnement intéressant:

- l'induction dans l'entrefer '*beent*',
- le facteur de puissance '*cosphi*',
- le facteur de saturation '*ksat*',
- ou les pertes Joule '*pjous*' et '*pjour*' au stator et au rotor.

* Sur les paramètres relatifs à la mise des conducteurs dans les encoches tels que:

- le coefficient de remplissage des encoches *crs*,
- la hauteur de conducteur '*hbs*' et de vide '*hcs*' dans l'encoche (*voir figure B.8 annexe B*)

Il faut en effet que le logiciel trouve un nombre de spires et un diamètre de fils qui fassent en sorte que le coefficient de remplissage reste dans un intervalle raisonnable ($0,35 \leq crs \leq 0,55$) alors que '*hcs*' et '*hbs*' doivent surtout éviter de devenir négatif, ce qui ferait converger le système vers des valeurs non physiques (*d'où les contraintes $0 \leq hbs \leq 0,01$ et $0 \leq hcs \leq 0,01$*).

* Sur la puissance '*pu*' et le rendement '*r*' afin que le système trouve une machine aux performances similaires à celles qui étaient annoncées pour le prototype. C'est pourquoi on impose à la puissance *pu* et au rendement *r* les contraintes suivantes: $pu=1000$ et $0,79 \leq r \leq 1$.

La fonction objectif est égale à la longueur de fer la machine (*d'où $coefcrs=0$, $coefl=1$ et $coefpu=0$*) qui pour notre cas peut varier entre 0,05 et 0,15 m.

Aussi, le cahier des charges représenté à la figure D.6 a été saisi dans le logiciel de dimensionnement (*concernant la signification de tous ces paramètres, on pourra se reporter à l'annexe B et à la liste des symboles*):

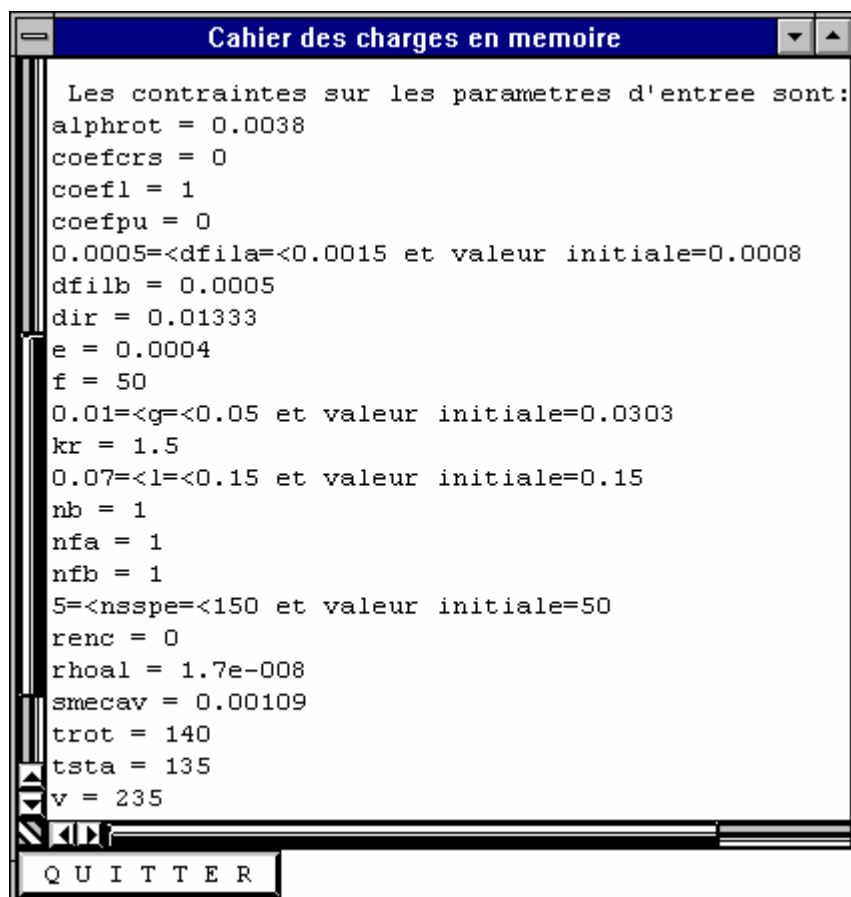
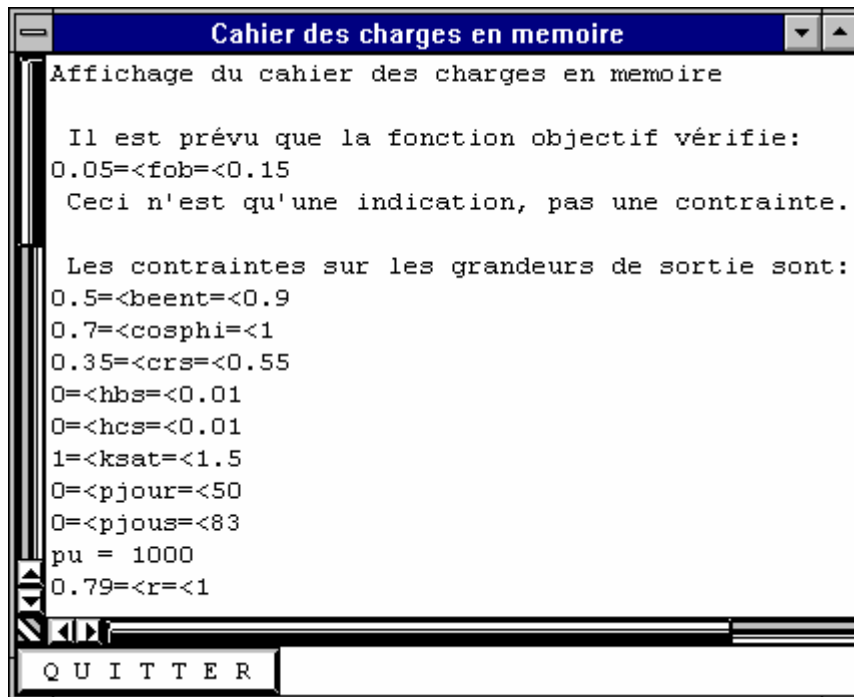


Figure D.6: cahier des charges en vue de dimensionner une machine de mêmes performances nominales que celles annoncées pour le prototype

Avec ce cahier des charges le logiciel de dimensionnement converge en 9 itérations vers une solution et ceci en quelques secondes.

```

Fenêtre d'affichag
fob = 0.0846021
beent = 0.753268
cdcn = 2.80383
cosphi = 0.851733
crs = 0.549976
cu = 3.29379
cud = 9.23525
hbs = 0.00515004
hcs = 0.00177996
ia = 2.10803
iad = 15.3839
iav = 1.02779
idin = 7.29774
jrs = 5.28359e+006
js = 3.66228e+006
ksat = 1.16054
pabs = 1265.81
pfer = 56.38
pjour = 38.3008
pjous = 70.6956
pu = 999.971
r = 0.789984
ra = 5.30296
rba = 4.30037
rm = 2592.33
xa = 4.31314
xba = 7.93837
xmk = 224.678
QUITTER

```

Les valeurs déterminées pour les paramètres d'entrées sont:

dfila=0,00069 m,

g=0,033,

l=84 mm,

nsspe=62 spires.

Les performances calculées par le logiciel pour cette machine sont données à la figure D.7.

On constate que la longueur de fer est 84 mm, alors qu'une longueur de fer de 150 mm avait été initialement prévue. Le gain de matière première, autant en cuivre qu'en fer, de l'ordre de 44% est donc substantiel pour une machine qui offrira les mêmes performances.

Le gain et la vitesse avec lesquels la nouvelle solution a été obtenue confirment le potentiel qu'un bureau d'études peut espérer tirer du type de logiciel de dimensionnement que nous proposons.

Par ailleurs, le fait d'avoir traité deux cahiers des charges aux objectifs bien différents, montre aussi la souplesse que les applications de dimensionnement du type PASCOSMA permettent d'obtenir.

Figure D.7: valeur des paramètres de sortie pour la machine de dimensions optimales déterminée

Conclusions de l'étude:

Cette étude avait pour objectif d'ancrer notre approche dans la réalité du dimensionnement qu'on peut rencontrer dans un bureau d'études et dans la réalité du fonctionnement d'une machine asynchrone. La conclusion que l'on peut tirer est que, dans ce contexte, le type de logiciel de dimensionnement que nous proposons devrait fournir des machines réalisables et optimales en des temps de dimensionnement extrêmement rapides.

BIBLIOGRAPHIE

BIBLIOGRAPHIE:

- [1] A. ABOU AKAR, G. GRELLET, A. JAMMAL, "Space Harmonics in Saturated Induction Motors - Effect of Geometric and Winding Parameters", International Conference on Electrical Machines, 13-15 August 1990, Massachusetts Institute of Technology, Cambridge, Massachusetts USA, pp. 239-244
- [2] AIDA, Version 1.65, Manuel de référence, ILOG S.A., Mars 1992
- [3] Philip L. ALGER, *Induction Machines*, ed. Gordon and Breach Science Publishers, 150 Fifth Avenue, New-York, 518 pages, 1970
- [4] G. ANGST, "Saturation Factors for Leakage Reactance of Induction Motors with Skewed Rotor", Trans. Amer. Inst. Elect. Engrs, 1963, 82, pp. 716-725
- [5] J. APPELBAUM, E.F. FUCHS, J.C. WHITE, "Optimization of Three-Phase Induction Motor Design - PART I: Formulation of the Optimization Technique", IEEE Transactions on Energy Conversion, Vol. EC-2, No. 3, September 1987, pp.407-414
- [6] J. APPELBAUM, I.A. KHAN, E.F. FUCHS, "Optimization of Three-Phase Motor Design - PART II: The Efficiency and Cost of an Optimal Design", IEEE Transactions on Energy Conversion, Vol. EC-2, No. 3, September 1987, pp. 415-421
- [7] M. BERNHEIM, FORTRAN mode d'emploi, interéditions, Paris, 1991, ISBN 2-7296-0303-4
- [8] NADY BOULES, "Design of Permanent Magnet DC Motors", IEEE Transactions on Industry, Vol. 26, No 4, July/August 1990, pp. 786-792
- [9] J. M. BOUCHE, F. WURTZ, L. GERBAUD, "Automatic Generation of Softwares for the Design of Static Converters", 6th European Conference on Power, Electronics and Applications (EPE'95), 19-21 September 1995, Sevilla Spain
- [10] P. BRUNOX, *Etude de différentes représentations du contacteur électromagnétique dans un but de conception innovante*, D.E.A. du LEG (Laboratoire d'Electrotechnique de Grenoble), Institut National Polytechnique de Grenoble, septembre 1994
- [11] O. I. BUTLER, T. S. BIRCH, "Comparison of alternative skew-effect parameters of cage induction motors", Proceedings IEE, Vol. 118, No 7, July 1971, pp. 879-883
- [12] Paul L. COCHRAN, *Polyphase Induction Motors- Analysis, Design, and Application*, ed. MARCEL DEKKER, 675 pages, 1989, ISBN 0-8247-4
- [13] B. J. CHALMERS, B. J. BENNINGTON, "Digital-computer program for design synthesis of large squirrel-cage induction motors", proceedings IEE, vol 114, No 2, february 1967, pp. 261-267

- [14] B. J. CHALMERS, R. DODGSON, "Saturated leakage reactance of cage induction motors", Proceedings IEE, Vol. 116, No 8, August 1969, pp. 1395-1404
- [15] R. M. CHAMBERLAIN, C. LEMARECHAL, H.C. PEDERSEN and M.J.D. POWELL, "The watchdog technique for forcing convergence in algorithms for constrained optimization", Mathematical Programming Study 16 (1982), pp.1-7.
- [16] N. CHRISTOFIDES, "Origins of load losses in induction motors with cast aluminium rotors", Proceedings IEE, vol. 112, No 12, december 1965, pp. 2317-2332
- [17] C. DELANNOY, *Programmer Windows avec TURBO C++*, Editions Eyrolles, 1994, ISBN2-212-08792-6
- [18] FLUX2D, Manuel d'utilisation, Version 7.1, CEDRAT S.A., 10 Chemin du Pré Carré, Zirst 38240 Meylan
- [19] FLUX3D, Manuel d'utilisation, Version 2.01, CEDRAT S.A., 10 Chemin du Pré Carré, Zirst 38240 Meylan
- [20] FLUXEXPERT, Manuel d'Utilisation, DT2I, 8 chemin des Prêles, 38240 MEYLAN ZIRST
- [21] F. FRANCOIS, *Contribution de la modélisation floue à la conception en génie électrique*, Thèse de docteur, Institut National Polytechnique de Grenoble, 1994
- [22] CIRCUIT, Manuel d'utilisation, CEDRAT S.A., 10 Chemin du Pré Carré, Zirst 38240 Meylan
- [23] A. GENTILHOMME, *C.O.C.A.S.E un système d'aide à la conception d'appareillage électriques*, Thèse de docteur, Institut National Polytechnique de Grenoble, 1991
- [24] J. F. GIERAS, "Sizing Equations and Magnetic Circuit Design for Single-Sided Linear Induction Motors", International Conference on Electrical Machines, 13-15 August 1990, Massachusetts Institute of Technology, Cambridge Massachusetts USA, pp.498-503
- [25] S. GITOSUSASTRO, J. L. COULOMB, "Finite Element Optimization of the Magnetic Field in a Quadrupole", ELECTRICAL AND POWER SYSTEMS modelling and simulation, J. Robert, W. Midvidy (editors), J.C. Baltzer AG, Scientific Publishing Co.© IMACS, 1989, pp.205-207
- [26] C. G. GODWIN, "Optimum Machine Design by Digital Computer", AIEE Transactions, August 1959, pp. 478-487
- [27] D. E. GOLDBERG, Algorithmes Génétiques, Exploration, Optimisation et Apprentissage Automatique, Ed. Addison-Wesley France, 1994, ISBN 2-87908-054-1
- [28] R. GORDON, XIAN LIU, "Modeling And Design Optimization of Permanent Magnet Motors", Electric Machines and Power systems, Volume 20, No 20, March-April 1992, pp. 71-92

- [29] Harwell Subroutine Library, 10th October 1987
- [30] H. C. J. DE JONG, "Skew Leakage in Induction Machines", ETEP Vol. 4, No 1, January/February 1994, pp. 43-46
- [31] W. JAZDZYNSKI, "Multicriterial optimisation of squirrel-cage induction motor design", proceedings IEE, Vol. 136, Pt. B, No 6, november 1989, pp 299-307
- [32] A.D. KONE, B. NOGAREDE, M. LAJOIE MAZENC, "Le dimensionnement des actionneurs électriques: un problème de programmation non linéaire", Journal de Physique III, Février 1993, pp. 285-301
- [33] A. D. KONE, *Contribution à la conception des actionneurs électriques par formulation en termes d'optimisation*, Thèse de docteur, Institut National Polytechnique de Toulouse, 1993
- [34] K. KADDED, *Optimisation de formes de machines électriques à l'aide d'un logiciel éléments finis et de la méthode des pénalités intérieures étendues*, Thèse de docteur, Institut National Polytechnique de Grenoble, 1993
- [35] LE LISP, Manuel de référence, Version 15.26, 1^{er} décembre 1993, ILOG, GENTILLY FRANCE
- [36] D. LOWTHER, "The Automated Design of Electromagnetic Devices - Fact or Fiction?", International Compumag Society Newsletter, Vol. 2, No. 2, pp. 6-9, 12 June 1995
- [37] D. LOWTHER, "Knowledge Based system I", Workshop on Applications of Intelligent Systems Techniques to Optimal Design of Electromagnetics Devices, July 9, 1995 (0900-1700), Berlin
- [38] A. MIRAOUI, *Dimensionnement optimal d'un moteur sans balais à aimants Néodyme-Fer-Bore*, Thèse de docteur, Université de Franche-Comté en sciences pour l'ingénieur, 1992
- [39] Macsyma reference manual version 13, Computer Aided Mathematics Group of Symbolics, INC., Nov. 1988
- [40] MATHCAD 5.0, Manuel de l'utilisateur, Mathsoft Europe, P.O. Box 58, Livingston EH547AE, Grande Bretagne
- [41] F. PEABODY, D. W. NYBERG, W.G. DUNFORD, "The Use of a Spreadsheet Program to Design Motors on a Personal Computer", IEEE Transaction on Industry Applications, Vol. IA-23, No. 3, May/June 1987, pp. 520-525
- [42] P. H. MELLOR, D. ROBERTS, D.R. TURNER, "Lumped parameter thermal model for electrical machines of TEFC design", IEE PROCEEDINGS-B, Vol. 138, No. 5, september 1991, pp. 205-218

- [43] R.W. MENZIES, G.W. NEAL, "Optimisation program for large-induction-motor design", proceedings IEE, Vol. 122, No. 6, june 1975, pp. 643-646
- [44] OSL, IBM, Point de contact: E_mail@osl.fr.ibm.com, FAX: 49-28-08-60, TEL: 40-01-55-45
- [45] M. POLOUJADOFF, M. IVANES, "Comparaison des schémas équivalents au moteur asynchrone polyphasé", Revue Générale de l'Electricité, Tome 76, No 1, janvier 1967, pages 1 à 6
- [46] M. J. D. POWELL, "Extension to subroutine VF02D", in System Modeling and Optimization, Lecture Notes in Control and Information Science 38, eds. R. F. Drenick and F. Kozin, Springer-Verlag (Berlin), 1982, pp. 529-538.
- [47] M. J. D. POWELL, "On the quadratic programming algorithm of Goldfarb and Idnani", Mathematical Programming Study 25 (1985), pp. 46-61.
- [48] G. V. REKLAITIS, A. RAVINDRAN and K. M. RAGSDELL, *Engineering Optimization - Methods and Applications* John Willey and Sons, Inc., 1979
- [49] M.M. RADULESCU, "Analytically modelling of a small slotless electronically-commutated neodymium-magnet motor", ELECTROMOTION 1, 1994, pp. 45-51
- [50] R. RAMARATHNAM, B.G. DESAI, "Optimization of polyphase induction motor design - a nonlinear programming approach", IEEE Transactions on Power Apparatus and Systems, VOL. PAS-90, No 2, MARCH/APRIL 1971, pp. 570-578
- [51] R. RAMARATHNAM, B.G. DESAI, and V. SUBBA RAO, "A comparative study of minimization techniques for optimization of induction motor design", IEEE Transactions on Power Apparatus and Systems, VOL. PAS 92, No 5, SEPTEMBER/OCTOBER 1973, pp. 1448-1454
- [52] M. RICHOMME, Utilisation d'un algorithme génétique en vue du dimensionnement d'une machine électrique, D.E.A. du LEG (Laboratoire d'Electrotechnique de Grenoble), Institut National Polytechnique de Grenoble, 19 septembre 1995
- [53] A. RIVAT, *Logiciel de simulation Analogique SPICE 5.30*, Editions DUNOD, Paris, 1994, ISBN 2 10 001981 3
- [54] Saber User's Guide, Release 3.1 a, Analogy Inc., P.O. Box. 1669, Beaverton, OR97075-1669
- [55] SABIR A. ELDHEMY, "Effect of Skewing on the Cogging Torque in Induction Machines", Electric Machines and Power Systems, Vol. 22, No 5, September-October 1994, pp. 545-555
- [56] R. R. SALDANHA, *Optimisation en Electromagnétisme par application conjointe des Méthodes de Programmation Non linéaire et de la Méthode des Eléments Finis*, Thèse de docteur, Institut National Polytechnique de Grenoble, 1988

- [57] C. M. SALDANHA, *An Algebraic Constraint System for Computer-Aided Design in Magnetics*, thesis, Computational Analysis and Design Laboratory, Department of Electrical Engineering, Mc Gill University, Montreal, Canada, 1988
- [58] C. SINGH, D. SARKAR, "Practical considerations in the optimisation of induction motor design", IEE PROCEEDINGS-B, Vol. 139, No. 4, July 1992
- [59] C.P. STEINMETZ, "The alternating current induction motor", AIEE Trans., 1897, vol. 14, pp. 185-217.
- [60] S. A. SWANN, J. W. SALMON, " Effective Resistance and Reactance of a solid Cylindrical Conductor Placed in a Semi-closed Slot", The Institution of Electrical Engineers, Monograph No. 527U, Vol. 109, Part C, No. 16, July 1962, pp. 611-615
- [61] S. A. SWANN, J. W. SALMON, "Effective resistance and reactance of a rectangular conductor placed in a semi closed slot", proceedings IEE, Vol. 110, No 9, September 1963, pp. 1656-1662
- [62] F. TRICHON, *Systèmes experts pour la conception de moteurs asynchrones*, rapport de fin d'étude, Ecole Nationale Supérieure d'Ingénieurs Electriciens de Grenoble (ENSIEG), Institut National Polytechnique de Grenoble, 1988
- [63] F. TRICHON, *Modélisation du processus de conception des machines électriques*, Thèse de docteur, Institut National Polytechnique de Grenoble, 1991
- [64] G. F. Üler, O.A. Mohammed, and C.S. Koh, "Utilizing genetic algorithms for the optimal desing of electromagnetic devices", *IEEE Transaction on Magnetics*, vol. 30, pp. 4296-4298, April 1994
- [65] J.A. DE VASCONCELOS, *Optimisation de forme de structures électromagnétiques*, Thèse de docteur, Ecole Centrale de Lyon, 1994
- [66] C. G. VEINOTT, "Induction Machinery Design being Revolutionized by the Digital Computer", AIEE Transactions, Power Apparatus and systems, vol. 75, pp. 1509-1517, 1956
- [67] C. G. VEINOTT, "Synthesis of Induction Motor Designs on a Digital Computer"; AIEE Transactions, Power Apparatus and systems, vol. 79, pp. 12-18, 1960
- [68] P. VIAROUGE, F. TOURKHANI, I. KAMWA, H. LE-HUY," Nonlinear Optimization Techniques for the Design of Static Converters", Proceedings of IMACS'TC1'93, Ecole Polytechnique de Montréal, July 7th to 9th 1993, pp. 543-547
- [69] S. WOLFRAM, MATHEMATICA, *A System for doing Mathematics by Computers*, second Edition, Addison Wesley PUBLISHING COMPANY, ISBN 0-201-51502-4;0-201-51507-5 (pbk)

- [70] F. WURTZ, J. BIGEON, J.C. POIRSON, "A Methodology and a Tool for the Computer Aided Design with Constraints of Electrical Devices", Compumag, Berlin, Germany, July 10-13, 1995
A paraitre aussi dans le numéro du mois de mai 1996 de IEEE Transactions on Magnetics
- [71] F. WURTZ, M. RICHOMME, J. BIGEON, "A Few Results for Using Genetic Algorithms in the Design of Electrical Machines", the Seventh Biennial Conference on Electromagnetic Field Computation, (CEFC'96), Okayama, March 18-20, 1996

Récapitulatif des figures utilisées dans le manuscrit:

Figure 1: les sept étapes de la méthodologie de génération automatique du logiciel de dimensionnement

Figure 2: cadre méthodologique à définir permettant une utilisation souple et conviviale du logiciel de conception généré = 2^{ème} niveau de la nouvelle approche

Figure 3: procédure à définir afin d'assurer une utilisation transparente des techniques d'optimisation

Figure 4: équations du modèle de dimensionnement d'un électro-aimant

Figure 5: Processus de changement d'espace de travail

Figure 6: première approche en vue de disposer d'une interface conviviale pour utiliser le logiciel de dimensionnement automatique sous contrainte généré

Figure 7: seconde approche en vue de disposer d'une interface conviviale pour utiliser le logiciel de dimensionnement automatique sous contrainte généré

Figure 8: fonctionnement général du logiciel de dimensionnement généré par la méthodologie PASCOSMA

Figure 9: arbre de dépendance élémentaire d'une équation

Figure 10: arbre de dépendance complet de l'électro-aimant

Figure 11: principe d'un algorithme itératif de détermination d'un variable implicite

Figure 12: Fenêtre principale de l'interface d'utilisation des application de type PASCOSMA (*exemple de l'électro-aimant*)

Figure 13: fenêtre de gestion d'un paramètre d'entrée (*exemple de l'électro-aimant*)

Figure 14: fenêtre de gestion d'un paramètre de sortie (*exemple de l'électroaimant*)

Figure 15: fenêtre de gestion de la fonction objectif (*exemple de l'électroaimant*)

Figure 16: exemple de cahier des charges saisi (*exemple de l'électro-aimant*)

Figure 17: exemple de calcul d'analyse (=calcul des paramètres de sortie pour un jeu de paramètres d'entrée) (*exemple de l'électro-aimant*)

Figure 18: exemple de calcul de sensibilité (=calcul des dérivées partielles des paramètres de sortie pour un jeu de paramètres d'entrée) (*exemple de l'électro-aimant*)

Figure 19: exemple de tracé de paramètre d'entrée en fonction de paramètre de sortie (*exemple de l'électro-aimant*)

Figure 20: compte rendu de dimensionnement (*exemple de l'électro-aimant*)

Figure 21: valeurs des paramètres d'entrée et de sortie trouvées pour le cahier des charges de la figure 16 (*exemple de l'électro-aimant*)

Figure 22: évolution de la fonction objectif au cours du dimensionnement qui est en l'occurrence la masse de la structure (*exemple de l'électro-aimant*)

Figure 23: évolution au cours du dimensionnement du paramètre d'entrée a (*hauteur de la dent de la culasse fixe*) et du paramètre de sortie f (*force en N fournie*) (*exemple de l'électro-aimant*)

Figure 24: comparaison entre la structure initiale et la structure finale du circuit magnétique (*exemple de l'électro-aimant*)

Figure 25: mathcad, une illustration du nouveau type d'outil mis à la disposition des concepteurs permettant une mise au point de modèles analytiques

Figure 26: schéma équivalent d'une phase de machine asynchrone

Figure 27: encoche stator et son paramétrage utilisé dans le modèle structural

Figure 28 arbre de dépendance condensé typique des modèles structuraux utilisés pour dimensionner des machines asynchrones

LISTE DES SYMBOLES EMPLOYES

LISTE DES SYMBOLES EMPLOYES

Symboles employés pour l'exemple de l'électro-aimant:

- a: Hauteur de la dent de la culasse fixe, voir figure A.2, en m
- atair: Ampères tours consommés par la circulation du flux dans l'air, en A
- atcf: Ampères tours consommés dans les dents de la culasse fixe, en A
- atcm: Ampères tours consommés dans la culasse mobile, en A
- atcn: Ampères tours consommés dans le noyau de la culasse fixe, en A
- attot: Ampères tours totaux consommés sur la ligne de circulation de flux moyenne, en A
- b: Largeur de l'entraxe de la culasse fixe, en m, voir figure A.2
- bair: Valeur de l'induction dans l'air, en T
- bcf: Valeur de l'induction dans la dent de la culasse fixe, en T
- bcm: Valeur de l'induction dans la culasse mobile, en T
- bcn: Valeur de l'induction dans le noyau de la culasse fixe, en T
- c: Largeur de la dent de la culasse fixe, en m, voir figure A.2
- d: Hauteur de la culasse mobile, en m, voir figure A.2
- ent: Largeur de l'entrefer, en m, voir figure A.2
- f: Force d'appel entre la culasse mobile et la culasse fixe, en N
- fluair: Valeur du flux total dans l'entrefer, en Wb
- h: Hauteur totale du circuit magnétique, en m, voir figure A.2
- hn: Hauteur du noyau de la culasse fixe et profondeur du circuit magnétique, en m, voir figure A.2
- lind: Largeur totale du circuit magnétique, en m, voir figure A.2
- m: Masse totale de électroaimant, en kg

Symboles employés pour l'exemple de la machine asynchrone:

- α_{rot} : Noté encore alphrot, coefficient d'évolution de la résistivité rotor, en 1/K, voir page 139
- α_{sta} : Noté encore alphasta, coefficient d'évolution de la résistivité stator, en 1/K, voir page 137
- ε : Noté encore eps, coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- ε_p : Noté encore epsp, paramètre intervenant dans la prise en compte de l'effet de peau au rotor, voir page 139
- γ : Noté encore landa, rang de l'harmonique considéré
- λ_{ar} : Noté encore landaar, coefficient de perméance de l'encoche rotor, unité S.I., voir page 143

- λ_s : Noté encore λ_{daas} , coefficient de perméance de l'encoche stator, unité S.I., voir page 141
- λ_{br} : Noté encore λ_{dabr} , coefficient de perméance dû au flux de dispersion dans l'entrefer au rotor, unité S.I., voir page 143
- λ_{bs} : Noté encore λ_{dabs} , coefficient de perméance dû au flux de dispersion dans l'entrefer au stator, unité S.I., voir page 141
- λ_{cr} : Noté encore λ_{dacr} , coefficient de perméance dû au flux de fuite différentielle au rotor, unité S.I., voir page 143
- λ_{cs} : Noté encore λ_{dacs} , coefficient de perméance dû au flux de fuite différentielle au stator, unité S.I., voir page 141
- λ_{dr} : Noté encore λ_{dadr} , coefficient de perméance dû au flux de fuite zig-zag au rotor, unité S.I., voir page 143
- λ_{ds} : Noté encore λ_{dads} , coefficient de perméance dû au flux de fuite zig-zag au stator, unité S.I., voir page 142
- λ_{ep} : Noté encore λ_{daer} , coefficient de perméance du flux de fuite de l'anneau de court-circuit, unité S.I., voir page 144
- ρ_{al} : Noté encore ρ_{oal} , résistivité du métal composant la cage rotor, en $\Omega.m$
- τ : Noté encore tau, pas polaire, en m, voir page 138
- ω : Noté encore w, pulsation statorique, en rad/s, voir page 133
- a: Surface de l'anneau de court-circuit, voir figure B.6, en m^2
- adeux: Coefficient intervenant dans le calcul de r_m , voir page 139
- ae: Aire de l'encoche stator, en m^2 , voir page 130
- aer: Aire de l'encoche rotor, en m^2 , voir page 131
- ag: Surface de l'entrefer par pôle, en m^2 , voir page 145
- ak: Coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- alphrot: Noté encore α_{rot} , coefficient d'évolution de la résistivité rotor, en $1/K$, voir page 139
- alphsta: Noté encore α_{sta} , coefficient d'évolution de la résistivité stator, en $1/K$, voir page 137
- athar: Ampères tours nécessaires pour faire circuler l'induction entre b_{drd} et b_{dre} , voir page 149
- athas: Ampères tours nécessaires pour faire circuler l'induction entre b_{dsd} et b_{dse} , voir page 147
- athdr: Ampères tours nécessaires pour faire circuler l'induction entre b_{drb} et b_{drc} , voir page 149
- athds: Ampères tours nécessaires pour faire circuler l'induction entre b_{dsb} et b_{dsc} , voir page 147
- ather: Ampères tours nécessaires pour faire circuler l'induction entre b_{dra} et b_{drb} , voir page 148

athes: Ampères tours nécessaires pour faire circuler l'induction entre bdsa et bdsb, voir page 147
 athfr: Ampères tours nécessaires pour faire circuler l'induction entre bdrc et bdrd, voir page 149
 athfs: Ampères tours nécessaires pour faire circuler l'induction entre bdsc et bdsd, voir page 147
 aun: Coefficient intervenant dans le calcul de r_m , voir page 139
 ban: Largeur de l'anneau de court-circuit, en m, voir figure B.6
 bar: Dimension d'une encoche rotor, en m, voir figure B.4
 bas: Dimension d'une encoche stator, en m, voir figure B.2
 bb: Dimension de conducteur équivalent pour prendre en compte l'effet de peau au rotor, en m, voir figure B.7
 bba: Contrainte créée pour PASCOSMA, voir page 136
 bbb: Contrainte créée pour PASCOSMA, voir page 136
 bbr: Dimension d'une encoche rotor, en m, voir figure B.4
 bbs: Dimension d'une encoche stator, en m, voir figure B.2
 bc: Largeur de conducteurs dans une encoche stator, en m, voir figure B.8
 bcr: Dimension d'une encoche rotor, en m, voir figure B.4
 bcra: Induction maximale dans la culasse rotor, en T, voir figure B.10
 bcs: Dimension d'une encoche stator, en m, voir figure B.2
 bcsa: Induction maximale dans la culasse stator, en T, voir figure B.10
 bdmrab: Induction moyenne entre bdra et bdrb, en T, voir page 148
 bdmrbc: Induction moyenne entre bdrb et bdrc, en T, voir page 149
 bdmrbd: Induction moyenne entre bdrc et bdrd, en T, voir page 149
 bdmrde: Induction moyenne entre bdrd et bdre, en T, voir page 149
 bdmsab: Induction moyenne entre bdsa et bdsb, en T, voir page 146
 bdmsbc: Induction moyenne entre bdsb et bdsc, en T, voir page 147
 bdmscd: Induction moyenne entre bdsc et bdsd, en T, voir page 147
 bdmsde: Induction moyenne entre bdsd et bdse, en T, voir page 147
 bdr: Dimension d'une encoche rotor, en m, voir figure B.4
 bdra: Induction maximale dans la dent rotor, en T, voir figure B.10
 bdrb: Induction maximale dans la dent rotor, en T, voir figure B.10
 bdrc: Induction maximale dans la dent rotor, en T, voir figure B.10
 bdrd: Induction maximale dans la dent rotor, en T, voir figure B.10
 bdre: Induction maximale dans la dent rotor, en T, voir figure B.10
 bds: Dimension d'une encoche stator, en m, voir figure B.2
 bdsa: Induction maximale dans la dent stator, en T, voir figure B.10
 bdsb: Induction maximale dans la dent stator, en T, voir figure B.10

- bdsc: Induction maximale dans la dent stator, en T, voir figure B.10
- bdsd: Induction maximale dans la dent stator, en T, voir figure B.10
- bdse: Induction maximale dans la dent stator, en T, voir figure B.10
- be: Largeur moyenne de l'encoche rotor, en m, voir page 136
- beent: Induction maximale dans l'entrefer, en T, voir page 145
- bk: Coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- c: Ouverture d'une bobine, en m, voir page 134
- cem: Couple électromagnétique total que fournit la machine, en N, voir page 151
- ck: Coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- cmecav: Couple de résistance mécanique visqueux, en N.m., voir page 152
- coefers: Coefficient intervenant dans la fonction objectif et pondérant le coefficient de remplissage des encoches, voir page 152
- coefl: Coefficient intervenant dans la fonction objectif et pondérant la longueur de fer, voir page 152
- coefpu: Coefficient intervenant dans la fonction objectif et pondérant la puissance utile, voir page 152
- cosphi: Facteur de puissance, sans unité, voir page 151
- coulee: Variable valant 1 si la cage rotor est coulée, 0 si la cage est à barres.
- crs: Coefficient de remplissage des encoches, sans unité, voir page 134
- cu: Couple utile, en N.m., voir 152
- dam: Diamètre moyen de la cage rotor, en m, voir figure B.6
- deext: Diamètre extérieur de la machine, en m, voir figure B.3
- denc: Nombre d'encoches par pôle, sans unité, voir page 133
- dfila: Diamètre de fil 1 utilisé pour faire une spire, en m
- dfilb: Diamètre de fil 2 utilisé pour faire une spire, en m
- di: Diamètre intérieur du stator, en m, voir figure B.3
- dir: Diamètre intérieur du rotor, en m, voir figure B.3
- dr: Diamètre extérieur du rotor, en m, voir figure B.3
- e: Entrefer simple, en m, voir figure B.3
- eps: Noté encore ϵ , coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- epsp: Noté encore ϵ_p , paramètre intervenant dans la prise en compte de l'effet de peau au rotor, voir page 139
- f: Fréquence d'alimentation du stator, en Hz
- fob: Fonction objectif, voir page 152
- g: Glissement de la machine, sans unité

- gf: Coefficient intervenant dans le calcul de la réactance de fuite de l'anneau de court circuit, voir page 144
- han: Hauteur de l'anneau de court-circuit, en m, voir figure B.6
- har: Dimension d'une encoche rotor, en m, voir figure B.4
- has: Dimension d'une encoche stator, en m, voir figure B.2
- hb: Hauteur de la barre rotor, en m, voir figure B.5
- hbs: Hauteur de conducteur dans une encoche stator, en m, voir figure B.8
- hcs: Hauteur de vide dans une encoche stator, en m, voir figure B.8
- hdr: Dimension d'une encoche rotor, en m, voir figure B.4
- hds: Dimension d'une encoche stator, en m, voir figure B.2
- her: Dimension d'une encoche rotor, en m, voir figure B.4
- hes: Dimension d'une encoche stator, en m, voir figure B.2
- hfr: Dimension d'une encoche rotor, en m, voir figure B.4
- hfs: Dimension d'une encoche stator, en m, voir figure B.2
- hreffp: Hauteur de conducteur rotor pour la prise en compte de l'effet de peau, en m, voir figure B.7 et page 135
- hreffpa: Contrainte créée pour PASCOSMA, voir page 136
- hreffpb: Contrainte créée pour PASCOSMA, voir page 136
- htr: Hauteur totale de l'encoche rotor, en m, voir page 131
- hts: Hauteur totale de l'encoche stator, en m, voir page 130
- ia: Module du courant stator, en A, voir page 151
- ib: Module du courant rotor, en A, voir page 151
- im: Module du courant passant dans la résistance représentant les pertes fers, en m, voir page 153
- jrs: Densité de courant dans une barre rotor, en A/m², voir page 152
- js: Densité de courant dans le bobinage stator, en A/m², voir page 152
- k: Coefficient permettant de ramener les impédances rotor au stator, sans unité, voir page 135
- kb: Coefficient de bobinage, sans unité, voir page 134
- kbr: Coefficient intervenant dans le calcul de k, voir page 135
- kc: Coefficient de Carter total de la machine, sans unité, voir page 138
- kci: Coefficient permettant de ramener les courants rotor du schéma équivalent (ramené au stator) à leurs valeurs réelles, sans unité, voir page 152
- kcr: Coefficient de Carter du rotor, sans unité, voir page 138
- kcs: Coefficient de Carter du stator, sans unité, voir page 138
- kdifff: Coefficient intervenant dans le calcul de la réactance de fuite différentielle statorique, voir page 141
- ki: Calcul du coefficient d'inclinaison des barres du rotor, sans unité, voir page 135

- kic: Paramètre intermédiaire intervenant dans le calcul de k_{sat} , sans unité, voir page 150
 kr: Valeur de l'inclinaison de la cage rotor, en pas dentaire, voir page 132
 krr: Coefficient de prise en compte de l'effet de peau dans le calcul de la résistance rotorique, voir page 140
 ks: Coefficient de prise en compte du raccourcissement dans le calcul de l'inductance de fuite du bobinage stator, sans unité, voir page 140
 ksat: Coefficient de saturation, sans unité, voir page 150
 kx: Coefficient prenant en compte l'effet de peau dans le calcul de l'inductance de fuite du rotor, sans unité, voir page 143
 kxr: Coefficient permettant à partir des perméances d'obtenir les inductances pour le rotor, unité S.I., voir page 142
 kxs: Coefficient permettant à partir des perméances d'obtenir les inductances pour le stator, unité S.I., voir page 140
 L: Longueur de fer de la machine, en m
 landa: Noté encore γ , rang de l'harmonique considéré
 landaar: Noté encore λ_{ar} , coefficient de perméance de l'encoche rotor, unité S.I., voir page 143
 landaas: Noté encore λ_{as} , coefficient de perméance de l'encoche stator, unité S.I., voir page 141
 landabr: Noté encore λ_{br} , coefficient de perméance dû au flux de dispersion dans l'entrefer au rotor, unité S.I., voir page 143
 landabs: Noté encore λ_{bs} , coefficient de perméance dû au flux de dispersion dans l'entrefer au stator, unité S.I., voir page 141
 landacr: Noté encore λ_{cr} , coefficient de perméance dû au flux de fuite différentielle au rotor, unité S.I., voir page 143
 landacs: Noté encore λ_{cs} , coefficient de perméance dû au flux de fuite différentielle au stator, unité S.I., voir page 141
 landadr: Noté encore λ_{dr} , coefficient de perméance dû au flux de fuite zig-zag au rotor, unité S.I., voir page 143
 landads: Noté encore λ_{ds} , coefficient de perméance dû au flux de fuite zig-zag au stator, unité S.I., voir page 142
 landaer: Noté encore λ_{er} , coefficient de perméance du flux de fuite de l'anneau de court-circuit, unité S.I., voir page 144
 lb: Largeur de la barre rotor, en m, voir figure B.5
 lcr: Longueur moyenne de la ligne de flux dans la culasse rotor, en m, voir page 150
 lcs: Longueur moyenne de la ligne de flux dans la culasse stator, en m, voir page 148
 lds: Largeur d'une dent statorique, en m, voir page 133

lm:	Distance entre le centre de la tête de bobine et l'anneau de court-circuit, en m, voir figure B.6
ls:	Longueur totale d'une spire, en m, voir page 137
lspi:	Longueur d'une spire sans les têtes de bobines, en m, voir page 137
ltb:	Longueur d'une tête de bobine, en m, voir page 137
m:	Nombre de phases de la machine
ma:	Limite mathématique pour désigner 0
muz:	Perméabilité du vide, unité S.I.
nb:	Nombre de voies du bobinage stator en parallèle
ne:	Nombre d'encoches au stator
nfa:	Nombre de fils en parallèle du diamètre de fil 1 utilisé
nfb:	Nombre de fils en parallèle du diamètre de fil 2 utilisé
nicr:	Ampères tours nécessaires pour faire passer le flux dans la culasse rotor, voir page 150
nicr:	Ampères tours nécessaires pour faire passer le flux dans la culasse stator, voir page 148
nidr:	Ampères tours nécessaires pour faire passer le flux dans les dents rotor, voir page 149
nids:	Ampères tours nécessaires pour faire passer le flux dans les dents stator, voir page 147
nig:	Ampères tours nécessaires pour faire circuler le champ magnétique dans l'entrefer simple, voir page 145
nr:	Nombre de barres au rotor
ns:	Nombre de spires en série par phase, voir page 133
nsr:	Coefficient intervenant dans le calcul de k, voir page 135
nsspe:	Nombre de spires en série par encoche
nz:	Norme de l'impédance du circuit équivalent qui modélise une phase stator, en Ω , voir page 151
orr:	Dimension d'une encoche rotor, en m, voir figure B.4
os:	Dimension d'une encoche stator, en m, voir figure B.2
p:	Nombre de paires de pôles, voir page 133
pa:	Nombre de pôles
pabs:	Puissance active totale absorbée, en W, voir page 152
pas:	Pas du bobinage en nombre de dents, voir page 133
pfer:	Pertes fer totales de la machine, en W, voir page 153
phip:	Flux mutuel par pôle, en Wb, voir page 145
pjour:	Pertes Joules au rotor, en W, voir page 152
pjours:	Pertes Joules au stator, en W, voir page 152

pmecav: Pertes mécaniques, en W, voir page 152
 pt: Puissance mécanique totale que fournit la machine, en W, voir page 151
 pu: Puissance utile que fournit la machine, en W, voir page 152
 pv: Coefficient intervenant dans le calcul de la réactance des têtes de bobines, voir page 142
 pva: Coefficient intervenant dans le calcul de la réactance des têtes de bobines, voir page 142
 pvb: Coefficient intervenant dans le calcul de la réactance des têtes de bobines, voir page 142
 q: Nombre d'encoches par pôles et par phase, voir page 133
 ra: Résistance du bobinage stator, en Ω , voir page 137
 ras: Résistance du fil de diamètre 1 composant le bobinage stator, en Ω , voir page 137
 rb: Résistance d'une barre rotorique, en Ω , voir page 139
 rba: Résistance rotorique ramenée au stator, en Ω , voir page 140
 rbag: rba divisé par g, voir page 151
 rbbs: Résistance rotorique, en en Ω , voir page 140
 rbs: Résistance du fil de diamètre 2 composant le bobinage stator, en Ω , voir page 137
 renc: Nombre d'encoche de raccourcissement pour un bobinage raccourci
 reqan: Résistance de la portion d'anneau de court-circuit, voir page 139
 rho: Résistivité du cuivre à 20°C, en $\Omega.m$
 rhoal: Noté encore pal, résistivité du métal composant la cage rotor, en $\Omega.m$
 rhot: Résistivité du bobinage stator à la température tsta, en $\Omega.m$, voir page 137
 rm: Résistance représentant les pertes fer, en Ω , voir page 139
 sa: Coefficient intervenant dans le calcul de hbs et hcs, voir page 141 et figure B.9
 sb: Coefficient intervenant dans le calcul de hbs et hcs, voir page 141 et figure B.9
 sbc: Coefficient intervenant dans le calcul de hbs et hcs, voir page 141 et figure B.9
 sbcra: Section de passage du flux dans la culasse rotor, en m^2 , voir page 150
 sbcsa: Section de passage du flux dans la culasse stator, en m^2 , voir page 148
 sbdra: Section de passage du flux dans la dent rotor, en m^2 , voir page 148
 sbdrb: Section de passage du flux dans la dent rotor, en m^2 , voir page 148
 sbdrc: Section de passage du flux dans la dent rotor, en m^2 , voir page 149
 sbdrd: Section de passage du flux dans la dent rotor, en m^2 , voir page 149
 sbdre: Section de passage du flux dans la dent rotor, en m^2 , voir page 149
 sbdsa: Section de passage du flux dans la dent stator, en m^2 , voir page 146
 sbdsb: Section de passage du flux dans la dent stator, en m^2 , voir page 146
 sbdsc: Section de passage du flux dans la dent stator, en m^2 , voir page 147
 sbdsd: Section de passage du flux dans la dent stator, en m^2 , voir page 147
 sbdse: Section de passage du flux dans la dent stator, en m^2 , voir page 147

- sc: Coefficient intervenant dans le calcul de hbs et hcs, voir page 141 et figure B.9
- sca: Surface de cuivre pour le fil de diamètre 1 composant le bobinage stator, en m², voir page 137
- scb: Surface de cuivre pour le fil de diamètre 2 composant le bobinage stator, en m², voir page 137
- sce: Surface utile de cuivre par encoche, en m², voir page 134
- scf: Surface utile de cuivre par spire, en m², voir page 134
- scr: Surface de l'encoche rotor, en m², voir page 136
- scrib: Surface de la barre rotor, en m², voir figure B.5
- scrca: Contrainte créée pour PASCOSMA, voir page 136
- scricb: Contrainte créée pour PASCOSMA, voir page 136
- scre: Surface réelle occupée par les conducteurs, en m², voir page 140
- scu: Surface utile de cuivre par spire, en m², voir page 137
- smecav: Coefficient de frottement visqueux des pertes mécaniques, unité S.I., voir page 137
- sp: Périmètre d'un pôle, en m, voir page 134
- tau: Noté encore τ , pas polaire, en m, voir page 138
- tr: Pas dentaire au rotor, en m, voir page 138
- trot: Température du rotor, en °C
- ts: Pas dentaire au stator, en m, voir page 138
- tsta: Température du stator en °C
- v: Tension d'alimentation par bobinage, en V
- w: Noté encore ω , pulsation statorique, en rad/s, voir page 133
- xa: Inductance de fuite statorique, en Ω , voir page 142
- xacr: Réactance de fuite de l'anneau de court-circuit, en Ω , voir page 144
- xb: Somme des réactances de fuite du rotor non ramenées au primaire, en Ω , voir page 144
- xba: Réactance de fuite du rotor ramenée au stator, en Ω , voir page 144
- xdisr: Réactance du flux de dispersion dans l'entrefer au rotor, en Ω , voir page 143
- xdiss: Réactance du flux de dispersion dans l'entrefer au stator, en Ω , voir page 141
- xdr: Réactance de fuite différentielle rotorique, en Ω , voir page 143
- xds: Réactance de fuite différentielle statorique, en Ω , voir page 141
- xer: Réactance de fuite d'encoche rotorique, en Ω , voir page 143
- xes: Réactance de fuite d'encoche statorique, en Ω , voir page 141
- xm: Inductance magnétisante, en Ω , voir page 138
- xmk: xm divisé par ksat, voir page 151
- xskew: Réactance de fuite due à l'inclinaison des barres du rotor, en Ω , voir page 144
- xtb: Réactance des têtes de bobines statoriques, en Ω , voir page 142
- xzr: Réactance de fuite zig-zag rotorique, en Ω , voir page 143

- xzs: Réactance de fuite zig-zag statorique, en Ω , voir page 142
- z: Impédance du schéma équivalent, en Ω , voir page 151
- za: Impédance de la branche du schéma équivalent représentant le bobinage stator, en Ω , voir page 151
- zb: Impédance de la branche du schéma équivalent représentant le rotor, en Ω , voir page 151
- zm: Impédance équivalente à x_{mk} et r_m en parallèle, en Ω , voir page 151
- zs: Impédance équivalente à x_{mk} et z_b en parallèle, voir page 153

Récapitulatif des tableaux utilisés dans le manuscrit:

Tableau 1: exemple de fonction Tr et Tr^{-1} de passage de l'espace réel à l'espace réduit en fonction du type du paramètre et de type de la contrainte appliquée sur ce paramètre

Tableau 2: calcul des dérivées partielles des paramètres de sortie dans l'espace réduit

Tableau 3: table de reformulation des contraintes d'égalité ou d'intervalle d'un cahier des charges

Résumé:

L'objectif de ce travail de thèse est de définir et de présenter une nouvelle approche de la conception sous contraintes de machines électriques. Celle-ci permet aux électrotechniciens d'utiliser l'ordinateur non pas seulement comme un outil d'analyse mais aussi comme un outil ayant de réelles facultés de dimensionnement automatique. Cette approche offre de grands avantages comme:

- la gestion des contraintes d'un cahier des charges,
- la gestion de l'interdépendance des phénomènes physiques intervenant dans une machine,
- ou encore l'optimisation des solutions trouvées.

Cependant sa grande originalité est qu'elle assure automatiquement la génération du logiciel de conception. Pour cela elle utilise comme connaissance de base, un modèle analytique de la machine à laquelle on s'intéresse, et comme moyen, des techniques de calcul symbolique et de programmation automatique. Afin de guider le processus de conception, elle emploie des algorithmes génériques d'optimisation numérique sous contraintes.

Cette approche est notamment appliquée, dans ce mémoire, au dimensionnement de machines asynchrones.

Mots clés:

conception automatique, optimisation, dimensionnement automatique, modèle analytique, calcul symbolique, machine asynchrone, programmation automatique

Abstract:

The aim of this work is to define and to present a new approach to the design with constraints of electrical devices. It gives to electrical engineers and searchers the opportunity to use computers not only as analysis tools but also as tools that have real capabilities to make automatic designs. This approach offers great advantages like the ability:

- to take into account constraints on the specifications of the device,
- to take into account the interdependence of all the physical phenomenon that occur,
- to find optimal solutions.

However its main originality is that it automatically generates the design software by using techniques of symbolic calculation and automatic programming. For this an analytical model of the device is used as basic knowledge. For the design process, constrained optimisation algorithms are used.

In this work, the approach is particularly used for the design of induction machines.

Keywords:

automatic design, optimisation, analytical models, symbolic calculus, induction machine, automatic programming