

Du modèle mathématique à une représentation déclarative et structurée des problèmes de conception de système en génie électrique

Application à la configuration et au dimensionnement d'un moteur brushless

Pierre-Alain YVARS¹, Arnaud HUBERT², Laurent ZIMMER³

¹ ISAE-Supméca, QUARTZ EA7393, 3 rue Fernand Hainaut, 93407 Saint Ouen Cedex, France

² UTC, Roberval, Centre de recherche de Royallieu CS 60319, 60203 Compiègne, Cedex, France

³ Dassault Aviation, Direction de la Prospective, 78 quai Marcel Dassault, 92552 Saint Cloud, France.

RESUME – Cet article présente l'intérêt d'une approche déclarative et structurée pour modéliser et résoudre des problèmes de conception préliminaire. La démarche permet notamment de disposer de modèles structurés, réutilisables et extensibles plus riches qu'un modèle mathématique « à plat ». L'utilisation d'une technique de recherche de solution admissible voire d'optimisation d'objectif de type programmation par contraintes sur des domaines mixtes apporte une forme de garantie sur les solutions obtenues. La méthodologie de conception proposée est illustrée sur un cas d'étude de génération d'architecture pré-dimensionnée pour un actionneur électrique de type brushless. Les données et le cahier des charges de cette conception sont issus de la littérature.

Mots-clés— *conception préliminaire, synthèse de système à base de modèle, programmation par contrainte en variable mixte, modélisation de problème, formalisme déclaratif, architecture.*

1. INTRODUCTION

De nombreux problèmes de conception préliminaire en génie électrique sont formulés sous la forme d'un problème d'optimisation mathématique [1]. Le problème est alors résolu à l'aide d'outils de programmation linéaire (si le problème est linéaire), d'optimisation mathématique basé sur des méthodes de gradient et de lagrangien (si le problème est non linéaire et différentiable) ou de méta-heuristiques dans les autres cas (algorithme génétique, recuit simulé, essaim particulaire, ...). Les limites de ces approches apparaissent assez vite. En particulier du point de vue de la modélisation, un problème de conception n'est pas toujours réductible à un jeu d'équations et d'inéquations et il convient de pouvoir entre autres capturer la structure des systèmes à concevoir.

L'objectif principal de cet article est de montrer qu'il existe depuis quelques années des formalismes, méthodes et outils qui permettent de décrire et de résoudre des problèmes de conception à partir d'un langage de plus haut niveau que les langages *génériques* utilisés jusqu'à présent. Cette approche est dite « Model-Based System Synthesis » (MBSS) [4]. Ce changement de paradigme permet alors aux concepteurs de se focaliser d'élaborer un modèle "métier" du problème de conception plutôt que de se focaliser sur la mise en équation et la résolution de celui-ci

Pour illustrer cette nouvelle approche, l'article s'appuiera sur un problème de conception « benchmark » déjà traité dans la littérature par différentes approches. Ce problème a été proposé initialement dans [1] où les auteurs l'ont résolu à l'aide de la programmation non linéaire. Il a ensuite été repris dans [2] en utilisant des méthodes globales et déterministes d'optimisation (programmation par contrainte sur des intervalles de réels). [3] a ensuite proposé une amélioration des techniques de résolution utilisées par [2] en exploitant judicieusement certaines contraintes d'égalité. Celles-ci sont ainsi ré-organisées afin de pouvoir exprimer « explicitement » certaines variables en fonction des autres inconnues du problème, ce qui permet de réduire la dimension du problème et d'accélérer la vitesse de résolution. Ce problème de conception a également servi de base à [5] pour étendre la méthodologie à la gestion d'inconnues de conception de différents types : réels, entiers, variables logiques. Cette extension permet alors de traiter et de résoudre aussi bien des problèmes de dimensionnement que des problèmes de choix d'architecture (choix de matériaux, de configuration du rotor, de présence ou absence d'encoche, etc.).

La limite de ces différents travaux reste cependant l'utilisation d'un langage de modélisation d'assez bas niveau, c'est-à-dire dont les capacités de représentation de la structure des problèmes est limitée voire inexistante. Cette utilisation limite ainsi la ré-utilisabilité et la compréhension des modèles développés pour la description et la résolution de problèmes de conception proches mais tout de même différents.

Afin de palier à cette limite, des travaux ont ensuite été menés pour utiliser des langages de modélisation de plus haut niveau. Par exemple, [6] a repris le problème proposé par [5] à l'aide du langage CE [9]. Ce langage est basé sur un ensemble de modèles déclaratifs à plats qui sont à la fois plus lisibles et mieux structurés ce qui permet une ré-utilisabilité améliorée des modèles et problèmes développés. L'objectif de la suite de ce papier est de franchir un nouveau pas dans cette direction en proposant la reprise de ce problème mais en utilisant un langage d'un niveau encore plus élevé, en l'occurrence le langage DEPS développé récemment [7].

Le langage DEPS de description et de modélisation des problèmes de conception se base sur la méthodologie de conception MBSS [4]. Il est associé à un environnement de modélisation et de résolution qui possède un solveur basé sur la programmation par contraintes sur variables mixtes (intervalles d'entiers ou de réels et énumérés). Dans ce cadre, la reprise d'un benchmark de conception déjà largement utilisé permet de montrer l'intérêt des modèles et langages de haut niveau pour la conception. Il illustre aussi les avantages de l'approche MBSS par rapport aux méthodes de préconception plus classiques actuellement utilisées.

2. REPRESENTATION DE PROBLEME EN CONCEPTION PRELIMINAIRE DE SYSTEMES EN GENIE ELECTRIQUE

[1] définit quatre catégories de problèmes de conception : le dimensionnement, la configuration, l'allocation (ou déploiement), la génération d'architecture. L'exemple traité dans cet article relève à la fois de la catégorie du dimensionnement et de celle de la génération d'architecture. Si l'on veut capturer l'ensemble des connaissances et informations nécessaires pour l'aide à la conception, un modèle uniquement constitué d'un jeu de variables, d'équations et d'inéquations est trop restrictif car il ne permet pas de répondre aux critères, représentatifs de l'activité de conception préliminaire, énumérés ci-dessous :

- Il est important de pouvoir distinguer clairement ce qui relève de l'architecture du système (sa structure) de ce qui relève de ses comportements (qui peuvent être de nature multi-physique, environnemental, économique, etc.). Il n'est pas toujours aisé pour les concepteurs ayant une culture en sciences de l'ingénieur qui prône un usage systématique de la simulation de bien faire la distinction entre ces deux notions. La distinction est pourtant essentielle car les variables associées à chaque notion jouent un rôle différent dans les modèles *inverses* utilisés en conception. Les variables liées à l'architecture sont les variables qui permettent de *fabriquer* le système. Une fois qu'une solution de conception est choisie, ces variables sont définies *de manière unique* car un seul système est conçu au final (elles doivent être fixées pour lancer des simulations et certaines de ces inconnues de conception correspondent à ce qu'on appelle les paramètres des modèles *directs*). En revanche, les variables liées au comportement peuvent ne pas être définies de manière unique dans les modèles inverses, notamment lorsque le système doit être conçu pour

satisfaire plusieurs modes de fonctionnement différents (par exemple, le système doit satisfaire certaines performances à froid et d'autres performances à chaud). Cette distinction apparaît à la fois en terme de paramètres, de variables (les inconnues du problème) et de propriétés (les équations associées). Ainsi, pour un moteur électrique, ses dimensions, ses configurations (présence ou non d'encoche, forme d'onde des tensions, type de matériau, diamètre du rotor, etc.) seront des *inconnues d'architecture*. De même les relations entre ces variables seront qualifiées de propriétés d'architecture – ou de structure. En revanche, les températures, le couple électromagnétique, l'amplitude des courants ou des tensions, seront des *inconnues comportementales*. Elles seront par exemple relatives à un comportement thermique et/ou électromagnétique de ce moteur dans un mode de fonctionnement particulier (le cahier des charges pouvant en spécifier plusieurs, à satisfaire à la fois, pour que la solution soit admissible). Ainsi, leurs valeurs changeront en fonction des exigences à satisfaire (par exemple, le profil de route pour un actionneur embarqué dans un véhicule électrique). On notera de plus que les inconnues structurelles et/ou comportementales peuvent être à valeur dans des intervalles (de \mathbb{R} ou de \mathbb{Z}) ou bien être des énumérations de valeurs possibles (variables liées à des catalogues de choix possibles ou définies par des variables booléennes). Le formalisme devra donc être capable de prendre en charge ces différents domaines lors de la déclaration des variables et paramètres.

- Comme on s'adresse à des problèmes de l'ingénieur, il pourra être également utile de *typer* (au sens informatique du terme) les variables et paramètres à l'aide de *types physiques* prédéfinis (appelées *quantités* en DEPS).
- Il est aussi nécessaire de disposer d'un formalisme permettant d'encapsuler ces paramètres, variables et propriétés architecturales au sein d'un même modèle que nous appellerons le « modèle structurel » (ou architectural) du système à concevoir. Ces modèles seront de plus hiérarchisés afin de représenter la complexité architecturale du système. En effet, les dispositifs technologiques complexes sont généralement structurés et décomposables en parties et sous-systèmes de plus en plus simples, le composant *élémentaire* correspondant à la décomposition de plus bas niveau.
- Il est également essentiel de pouvoir encapsuler au sein de chaque modèle comportemental les paramètres, les variables et les propriétés caractéristiques de son comportement (selon plusieurs points de vue et plusieurs modes de fonctionnement possibles).

3. APPLICATION A LA CONCEPTION D'UN MOTEUR BRUSHLESS « BENCHMARK »

L'exemple choisi pour illustrer ce papier relève à la fois de la classe des problèmes de configuration et de celle du dimensionnement. Comme indiqué dans l'introduction, cet exemple a été proposé et résolu pour la première fois par [5] à l'aide d'un langage de programmation faisant appel à une méthode de programmation par contraintes sur des intervalles. Cependant [5] se limite alors à une formulation mathématique à *plat* du problème (sans structuration) et il utilise un programme informatique spécifiquement écrit pour sa résolution. [6] a repris ensuite ce problème en utilisant le langage de modélisation d'un solveur de contrainte dans lequel il est possible d'exprimer le problème sous une forme déclarative plus lisible et plus accessible pour les concepteurs métiers. Malheureusement ce langage propose encore une structuration très faible. Nous

proposons de montrer comment le formalisme DEPS permet d'aller plus loin pour l'aide à la conception et comment il permet d'incarner les points listés précédemment.

3.1. Description du problème

Le benchmark utilisé correspond à la conception d'un moteur brushless à aimants permanents à flux radial. En plus des relations géométriques, électromagnétiques et thermiques, le modèle de conception de [5] prend en compte :

- i les différentes configurations technologiques possibles : présence ou non d'encoches, forme d'onde sinusoïdale ou rectangulaire, rotor interne ou externe ;
- ii les différents matériaux constitutifs possibles : le choix dans les aimants NdFeB (à liant synthétique *ou* moderne) et le type de circuit magnétique utilisable (à base de poudre magnétique douce *ou* de matériaux emboutis).

L'objectif de la conception est alors de trouver une ou plusieurs solutions d'architecture dimensionnée de sorte que le moteur satisfasse un ensemble d'exigences thermiques et électromagnétiques spécifiées dans le cahier des charges de conception.

La figure 1 tirée de [2] représente un exemple possible d'architecture (moteur sans encoche et à rotor interne).

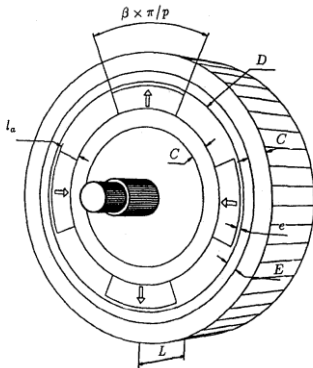


Fig. 1. Exemple d'une architecture possible pour le moteur brushless (moteur sans encoche et à rotor interne).[2]

3.2. Identification et classification des inconnues et paramètres du problème

L'ensemble des inconnues de ce problème est reporté dans le tableau 1 qui spécifie le type (entier ou réel) et le domaine (intervalle ou énuméré) de valeurs possibles de ces différentes variables. L'ensemble du détail des équations de modélisation peut être trouvé dans [2] pour la conception sans choix de configuration et dans [5] pour la conception avec choix de configuration (problème considéré dans ce papier).

Tab. 1. Tableau récapitulatif des variables définies et utilisées dans les modèles de conception du moteur brushless.

Nom	Unités	Type	Domaine	Définition
Système moteur				
M_a	kg	Réel positif	[1, 5]	Masse totale du moteur
Architecture Moteur (configuration + géométrie)				
Inconnues géométriques				
D	m	Réel positif	[0.001, 0.3]	Diamètre d'alésage intérieur du stator
L	m	Réel positif	[0.05, 0.15]	Longueur utile du moteur
a	m	Réel positif	[0.004, 0.4]	Longueur d'une encoche
d	m	Réel positif	[0.004, 0.4]	Longueur d'une dent
p	-	Entier positif	[1, 16]	Nombre de paires de pôles
l_a	m	Réel positif	[0.004, 0.4]	Épaisseur d'un aimant
C	m	Réel positif	[0.004, 0.4]	Épaisseur de la culasse stator ou rotor
E	m	Réel positif	[0.004, 0.4]	Épaisseur des bobinages
Inconnues de configuration				
σ_e	-	Entier positif	{0, 1}	Type d'armature du stator : avec encoches (=0), sans encoche (=1)
σ_f	-	Entier positif	{0, 1}	Type de forme d'onde : sinusoïdale (=0) ou rectangulaire (=1)
σ_r	-	Entier positif	{0, 1}	Type de configuration rotorique : externe (=0), interne (=1)
σ_m	-	Entier positif	{0, 1}	Type d'aimants permanents : modernes (=1), plastiques (=0)
σ_{mt}	-	Entier positif	{0, 1}	Type de matériaux magnétique : emboutis (=1), à poudre (=0)
Inconnues supplémentaires				
N_e	-	Entier positif	[1, 200]	Nombre d'encoches
R_{int}	m	Réel positif	[0.05, 0.08]	Rayon intérieur
R_{ext}	m	Réel positif	[0.05, 0.08]	Rayon extérieur
kd	-	Réel positif	[0.4, 0.6]	Ratio de correction si présence de denture
K_f	-	Réel positif	[0, 0.3]	Fuite magnétique
kT	-	Réel positif		Coefficient de couple
kc	-	Réel positif		Coefficient de Carter
S	-	Entier positif	{-1, 1}	Signe dépendant de la configuration rotor : rotor externe (=1), rotor interne (=1)
V_g	m ³	Réel positif	[0, +∞]	Volume global
V_c	m ³	Réel positif	[0, +∞]	Volume de la culasse stator et rotor
V_t	m ³	Réel positif	[0, +∞]	Volume des dents
V_{co}	m ³	Réel positif	[0, +∞]	Volume des conducteurs électriques
V_a	m ³	Réel positif	[0, +∞]	Volume des aimants
Comportement thermique				
R_e	K W ⁻¹	Réel positif	[0, +∞]	Résistance thermique équivalente entrefer
R_c	K W ⁻¹	Réel positif	[0, +∞]	Résistance thermique équivalente des culasses
Φ_{th}	W	Réel positif	[0, +∞]	Flux thermique
Comportement électromagnétique				
J	T	Réel positif	[0.6, 0.9]	Magnétisation des aimants permanents
ρ_{PM}	kg m ⁻³	Réel positif	{6000, 7900}	Masse volumique des aimants permanents
BM	T	Réel positif	{1.2, 1.5}	Champ max. des matériaux magnétiques
ρ_{CM}	kg m ⁻³	Réel positif	{6000, 7900}	Masse vol. des matériaux magnétiques
j	A mm ⁻²	Réel positif	[3, 6]	Densité de courant surfacique
B_e	T	Réel positif	[0.4, 0.9]	Densité de flux magnétique à vide entrefer
B_t	T	Réel positif	[0, +∞]	Densité de flux dans les dents
B_c	T	Réel positif	[0, +∞]	Densité de flux dans la culasse stator, rotor
K_s	kA m ⁻¹	Réel positif	[500, 1'5]	Densité de courant linéique
C_{em}	N . m	Réel positif	10	Couple électromagnétique exigé

3.3. Représentation de la structure du problème

Notre approche de conception repose sur l'utilisation de la méthodologie MBSS [4]. Le premier travail du concepteur est alors de *structurer* le problème global afin de faire apparaître explicitement ses différentes composantes et sous-systèmes. Cette décomposition permettra alors une réutilisation simplifiée des modèles pour des conception ultérieures légèrement différentes. Cette structuration repose sur des concepts issus de la modélisation orientée objet. Un diagramme de classes UML est alors pratique pour la représenter graphiquement : la figure 2 représente ainsi les principaux éléments de la structure utilisés pour modéliser ce problème de conception. On y visualise les différentes relations structurelles à prendre en compte entre les objets/modèles. Ainsi, ce problème est composé d'un ensemble d'exigences à satisfaire, d'un système moteur *sous-défini* et d'un modèle de matériaux, également *sous-défini*. Le système moteur est composé d'une architecture, de deux modèles de

comportement (électromagnétique et thermique) et d'un modèle de matériau. Le modèle d'architecture (unique) est partagé par ces deux modèles comportementaux.

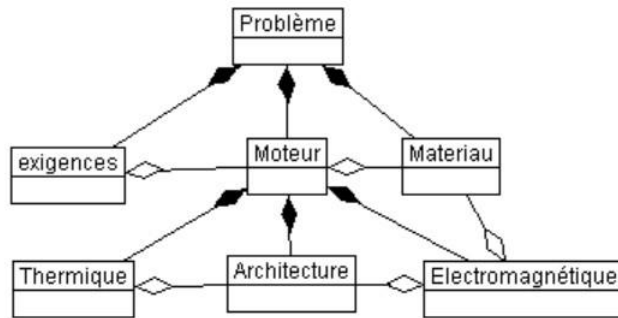


Fig. 2. Représentation graphique de la structure du problème

La figure 3 représente le *squelette* des trois principaux modèles DEPS de ce problème de conception : le problème lui-même (Problem BrushlessDesign), le modèle du moteur (Model Moteur) et le modèle des exigences. Chaque modèle permet une encapsulation de la représentation et des propriétés de chaque objet utilisé dans l'approche MBSS. Un modèle est constitué de constantes (fixés), de variables (inconnues à déterminer lors de la conception), d'éléments (les sous-systèmes ou les points de vue qui sont en association avec ce modèle par *agrégation* ou par *composition*) et de propriétés (l'ensemble des relations, équations, inéquations *définissant* tout objet instance de ce modèle). Dans le modèle du moteur, on voit par exemple qu'un moteur est composé de trois éléments : une architecture AS, un matériau électromagnétique MM, un comportement électromagnétique EM et un comportement thermique TH. On remarque de plus que ces deux comportements dépendent de l'architecture AS passée en argument des deux modèles et que le modèle de comportement électromagnétique EM dépend également du modèle de matériau MM passé en argument.

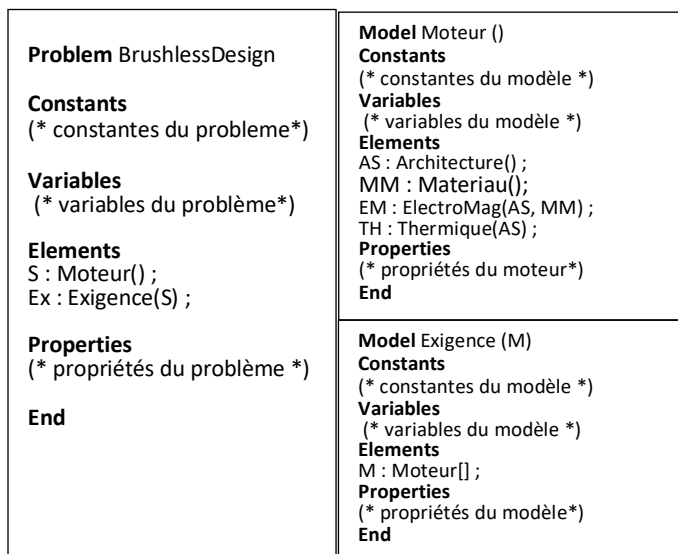


Fig. 3. Représentation en DEPS du sauelette de la structure du problème et des deux modèles associés (Moteur + Exigence)

Dans l'approche MBSS, il est important de toujours garder à l'esprit que tous les modèles sont *sous-définis*, c'est-à-dire qu'il sont déclarés et créés alors même que leurs variables ne sont pas encore *instanciées* : celles-ci appartiennent à un ensemble de départ – un intervalle ou une liste d'énumérés – qui sera ultérieurement *restreint* par le solveur durant le processus de

résolution. En effet, ce sera l'objet de ce processus d'instancier les valeurs *admissibles* de toutes les inconnues du problème de conception par propagation des contraintes encapsulées dans les propriétés de tous les modèles déclarés.

3.4. Enrichissement des modèles et réutilisabilité

Chaque Modèle peut-être enrichi au niveau de ses constantes, variables et propriétés. La figure 4 représente le modèle thermique complet enrichi des relations de définition de *Re*, *Rc* et *phiTh*. Néanmoins il y a une relation algébrique faisant intervenir des variables et des constantes des modèles électromagnétique, thermique et matériau. La pose de ce type de relation se fera au niveau englobant, celui du modèle du moteur (cf Figure 5).

```

Model Thermique(AS)
Constants
he : Real = 12;          ha : Real = 4;
thetaj : Real = 403.15;  thetae : Real = 328.15;
thetac : Real = 323.15;  rhocu : Real = 0.018e-6;

Variables
expr phiTh : Real;
expr Re : Real;
expr Rc : Real;

Elements
AS : Architecture[];

Properties
Re := 1/(Pi*AS.D*AS.L*he);
Rc := 1/(Pi*(AS.D+2*AS.S*(AS.E+AS.C))*AS.L*ha);
phiTh := (thetaj-thetae)/Re+(thetaj-thetac)/Rc;
End

```

Fig. 4. Représentation en DEPS du modèle du comportement thermique

```

Model Moteur ()
Constants
(* constantes du modèle *)

Variables
(* variables du modèle *)

Elements
AS : Architecture();
MM : Matériau();
EM : ElectroMag(AS, MM);
TH : Thermique(AS);

Properties
(* propriété inter-modèles*)
TH.phiTh=
TH.rhocu*Pi*AS.L*(AS.D+AS.S*AS.notsigmae*AS.E)*MM.Ks*EM.j;

End

```

Fig. 5. Pose d'une propriété inter-modèles au niveau du modèle de Moteur

La figure 6 représente un exemple de modèle associé à la définition des différents matériaux magnétiques disponibles. Un matériau est défini via deux grandeurs : sa densité de flux magnétique maximale *BM* et sa masse volumique *rhoCM*. Suivant la procédure proposée par [5], un choix de matériau est sélectionné au travers d'une variable *sigmat* qui peut prendre autant de valeur que de matériaux disponibles (deux dans notre cas). Dans [5], il n'y a que deux choix possibles de matériaux, ce qui lui permet de traiter ce choix à l'aide d'une variable

booléenne. DEPS étant un langage de plus haut niveau, il permet de gérer ce choix de manière plus puissante à l'aide de contraintes catalogues. Celles-ci sont implémentées dans les propriétés du modèle de matériaux via des tables prédéfinies. Ces tables contiennent les caractéristiques des matériaux disponibles et elles peuvent être facilement enrichies si de nouveaux matériaux sont disponibles ou si d'autres caractéristiques matériaux sont utilisées dans les modèles.

Model Materiau() Constants	Table ConducteurMag
Variables sigmat : Integer in [0, 1]; BM : Real in {1.2, 1.5}; rhoCM : Real in {6000, 7900};	Attributes sigmamnt: Integer; BM: Real; rhoCM: Real;
Elements Properties Catalog([sigmat, BM, rhoCM], ConducteurMag);	Tuples [0, 1.2, 6000], [1, 1.5, 7900]
End	End

Fig. 6. Représentation en DEPS du modèle de matériau magnétique

La structuration des modèles apportée par le formalisme DEPS permet de faciliter la réutilisabilité pour d'autres problèmes de conception. En effet, une fois conçu, le modèle de matériaux peut être réutilisé par d'autres problèmes de conception nécessitant l'utilisation de ce catalogue de matériaux disponibles. Inutile alors de le redéfinir, il suffit simplement de l'insérer à la manière d'un paquet. La table de données peut également être enrichi tant du point de vue de l'ajout d'attributs supplémentaires que de celui de l'insertion de *Tuples*.

3.5. Génération des solutions

L'ensemble du problème a été modélisé et résolu dans l'environnement de modélisation et de résolution DEPS Studio. DEPS Studio [8] intègre un solveur dédié développé spécifiquement pour résoudre des problèmes de conception modélisé en DEPS. Il est basé sur les techniques de programmation par contraintes en domaine mixte. Il traite naturellement les boucles algébriques et garantit la complétude et la cohérence des solutions obtenues. En cas d'échec de la résolution, cet événement est interprété comme une preuve d'absence de solution au problème posé (en cas de problème sur-constraint par exemple).

Tab. 2. Résultats de résolution sous DEPS Studio

Nom	Domaine initial	Domaine réduit	1ère solution admissible	Solution optimale
Système moteur				
<i>Ma</i>	[1, 5]	[0.1071, 5]	[4.6324, 4.7816]	[2, 1.777, 2.18]
Architecture Moteur (configuration + géométrie)				
Inconnues géométriques				
<i>D</i>	[1°-3, 0.3]	[0.108, 0.152]	[0.1256, 0.1265]	[0.1292, 0.13]
<i>L</i>	[0.05, 0.15]	[0.05, 0.15]	[0.0992, 0.1]	[0.0514, 0.0515]
<i>a</i>	[0.004, 0.44]	[0.004, 0.1279]	[0.0060, 0.0061]	[0.004, 0.004]
<i>d</i>	[0.004, 0.4]	[0.004, 0.1279]	[0.0041, 0.0041]	[0.004, 0.004]
<i>p</i>	[1, 16]	[1, 16]	[6, 6]	[8, 8]
<i>la</i>	[0.004, 0.1]	[0.004, 0.1]	[0.0055, 0.0055]	[0.0040, 0.0040]
<i>C</i>	[0.004, 0.1]	[0.004, 0.026]	[0.0047, 0.0047]	[0.0040, 0.0040]
<i>E</i>	[0.004, 0.1]	[0.004, 0.1]	[0.0074, 0.0075]	[0.0074, 0.0075]
Inconnues de configuration				
σ_c	{0, 1}	{1}	{1}	{1}
σ_f	{0, 1}	{0, 1}	{0}	{1}
σ_r	{0, 1}	{0, 1}	{0}	{0}
σ_m	{0, 1}	{0, 1}	{0}	{1}
σ_{mf}	{0, 1}	{0, 1}	{0}	{0}

Le tableau 2 synthétise les résultats obtenus pour les valeurs des variables de conception et la masse du système. Une première solution admissible est trouvée en 5,625 secondes sur une machine équipée d'un processeur Intel Core I7 cadencé à 2.8 Ghz et équipée de 16Go de RAM. Une solution optimale pour la masse du moteur est obtenue en 1 minute et 29 secondes.

4. CONCLUSIONS

Ce travail montre l'intérêt de disposer d'un formalisme déclaratif de haut niveau pour la représentation et la résolution de problème de conception préliminaire en génie électrique. D'autres fonctionnalités et mécanismes sont disponibles en DEPS et peuvent être utilisées pour modéliser des problèmes plus complexes. Citons l'héritage de modèles, les propriétés conditionnelles et les contraintes par morceaux. Nous travaillons à présent à la modélisation DEPS et à la résolution à l'aide de DEPS Studio de problèmes de type chaîne de traction électrique intégrant plusieurs sous-systèmes.

5. REFERENCES

- [1] Kone A., Nogarede B., Lajoie M., Le dimensionnement des actionneurs électriques : un problème de programmation non-linéaire. Journal de Physique III, EDP Sciences, Vol. 3(2), pp 285-301, 1993.
- [2] Messine F., Nogarede B., Lagouanelle J.-L., Optimal Design of Electromechanical Actuators: A new method based on global optimization, IEEE trans. on Magnetics, Vol. 34(1), Jan. 1998.
- [3] Mazhoud I., Hadj-Hamou K., Bigeon J., Remy G., The electromagnetic Actuator Design Problem: An adapted Interval Global Optimization Algorithm, IEEE trans. on Magnetics, Vol. 48(2), Feb. 2012.
- [4] Yvars P.A., Zimmer L., Toward a correct by construction design of complex systems : The MBSS approach, Procedia CIRP, Vol. 109, Elsevier, 2022.
- [5] E. Fitin, F. Messine, and B. Nogarède, « The electromagnetic actuator design problem : A general and rational approach », IEEE Trans. On Magnetics, Vol. 40(3), pp.1579-1590, May 2004.
- [6] S. Diampovesa, A. Hubert, and P. A. Yvars, "Nouveaux outils de synthèse pour la conception préliminaire et le choix d'architecture. application 'a un moteur brushless", Symposium de Génie Electrique (SGE 2018), 3-5 Juillet 2018, Nancy, 2018.
- [7] Yvars P.A., Zimmer L., Integration of Constraint Programming and Model-Based Approach for System Synthesis, proc of the IEEE International Systems Conference, SYSCON 2021, Vancouver, Canada.
- [8] Yvars P.A., Zimmer L. DEPS Studio : Un environnement intégré de modélisation et de résolution pour la synthèse de système, 20ème journées AFADL, Approche Formelle dans l'Assistance au Développement de Logiciel, Montpellier, 2021.
- [9] L. Zimmer, A. Anglada, M. Christie, and L. Granvilliers, Constraint Explore: a modelling and sizing tool for engineering design, invited session on Metamodelling and Constraint Based Problem Solving for Embodiment DesignSupport Systems in SCI, Orlando, 2004.