The electromagnetic actuator design problem: An adapted interval global optimization algorithm using model reformulation and constraint propagation

I. Mazhoud[1], K. Hadj-Hamou[1], J. Bigeon[1] and G. Remy[2]

[1] G-SCOP - CNRS, Grenoble-INP-UJF, 46 Av. Félix Viallet, 38000 Grenoble, France
[2]Laboratoire de Génie Electrique de Paris (LGEP) / SPEE-Labs, CNRS UMR 8507; SUPELEC; Université Pierre et Marie Curie P6; Université Paris-Sud 11; 11 rue Joliot Curie, Plateau de Moulon F91192 Gif sur Yvette CEDEX, France
issam.mazhoud@grenoble-inp.fr, khaled.hadj-hamou@grenoble-inp.fr,

This paper presents a deterministic optimization algorithm applied to the optimal design of electromagnetic actuators. The algorithm is based on interval arithmetic and constraint propagation, and aims at solving nonlinear optimization problems by enclosing the global optimum. A new reformulation step is introduced in order to accelerate the convergence of the algorithm and increase the solutions accuracy. The tests have been performed according to three performance criteria: convergence, precision and number of iterations. The reformulation has been validated on the optimal design of a second electromagnetic device: a transformer.

***Index Terms*—analytical model, design optimization, nonlinear equation**

I. Introduction

NOWADAYS, the design optimization topic is generating more and more interest especially with growing environmental issues. Design optimization may lead to substantial savings in material and energy consumption. In many cases, a local optimum is no longer sufficient and the global optimum is required.

In the preliminary design of electromagnetic machines, a design model is usually dealt with. There are principally two types of models: exact models from physico-mathematical modeling and approximated models from RSM, Kriging…[1] A design model is the aggregation of an exact or approximated model and the specifications (see figure 1). The aim of the preliminary design phase is to propose a first quantification of the design parameters that will be the basis of the future prototypes. Consequently, proposing the best solution that respects the design constraints (customer, physics, economic…) reduces the prototyping costs. F. Messine has proved in [2] the interest of using global optimization methods in the design of electromagnetic actuator; the use of such methods allows a gain of about 10%.

The interval-based method is one of the most promising deterministic global optimization methods. In fact, the Interval Branch and Bound Algorithm (IBBA) has the property to exactly enclose, with a fixed precision specified by the user, the global optimum and all the corresponding optimizers. Nevertheless, the IBBA, like all deterministic algorithms, may be time-consuming. In addition, as it is based on interval arithmetics, the operators must be adapted to this algebra.

The aim of this paper is to propose a new reformulation adapted to the IBBA and to the preliminary design context that allows to reduce the number of iterations.

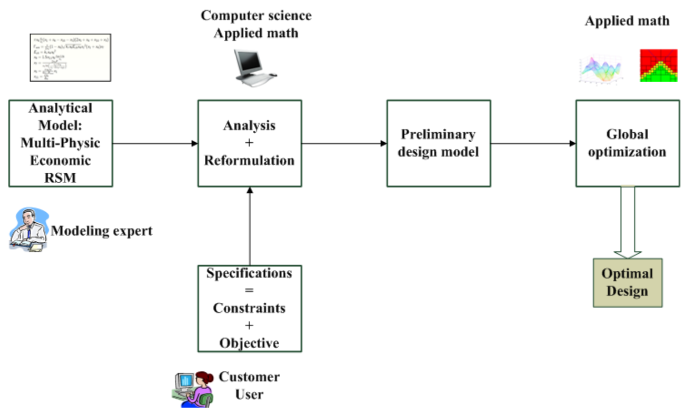

FIG I. PRELIMINARY DESIGN APPROACH: REFORMULATION AND OPTIMIZATION

In this paper, we introduce an adapted global optimization algorithm that aims at providing exact solutions for the optimal design of electromagnetic actuators if the preliminary design model is feasible, and proof of non-existence if not. This method is based on extended interval analysis and constraint propagation [3] coupled with a new reformulation. The reformulation aims at increasing the algorithm performance so as to obtain a more accurate solution in fewer iterations. The algorithm and the reformulation will also be tested on the optimal design of transformers.

II. Interval Analysis and Contractors

Interval analysis has been introduced by R.E Moore in 1966 [4] as an approach to overcome bounding errors. All the usual arithmetic operators have been extended to the interval analysis. Before introducing some interval operators, two essential concepts should be defined:

Definition 1: (Interval) An interval is a connected ([1,2]U[3,5] is not an interval), closed ([1,2[is not an *interval) subset of ℝ.The set of all intervals of ℝ will be denoted* $\mathbb{IR}$ *.*

Definition 2: (Box) A box is the Cartesian product of n intervals i.e. an interval vector: ([1,2],[4,5],[4,10]) is a tridimensional box. A n-dimension box is an element of $\mathbb{IR}^n$*.*

Definition 3: (elementary operators) Let X and Y be two intervals and $\circ \in \{+,-,\times,/\}$*:*

$$X \circ Y = \{x \circ y \mid x \in X, y \in Y\} \quad (1)$$

Thereby:
[a, b]+[c, d] = [a+c, b+d]
[a, b]-[c, d] = [a-d, b-c]
etc…

The other elementary operations have been adapted (Cos, Sin, log …). For more details, see [5].

For the Cos function, let i be an interval $[a..b]$. We define $e_1 = E(a/\pi)$ and $e_2 = E(b/\pi)$ where E(x) is the floor of x. We have:

$$cos(i) = \begin{cases} [-1,1] & if\ e_1 < e_2 - 1 \\ [min\{cos(a), cos(b)\}, 1] & if\ e_1 = e_2 - 1\ and\ e_1\ is\ odd \\ [-1, max\{cos(a), cos(b)\}] & if\ e_1 = e_2 - 1\ and\ e_1\ is\ even \\ [cos(a), cos(b)] & if\ e_1 = e_2\ and\ e_1\ is\ odd \\ [cos(b), cos(a)] & if\ e_1 = e_2\ and\ e_1\ is\ even \end{cases}$$

We also define the following notations for intervals and boxes. Let i be an interval [a, b]:

- $left([a,b]) = a$
- $right([a,b]) = b$
- $mid(i) = \frac{a+b}{2}$ the middle (center) of the interval $[a,b]$. For a box, the vector middle of the box intervals. For example, for a box $X = ([1,2],[4,5],[4,10])$, $mid(X) = (\frac{3}{2}, \frac{9}{2}, 7)$.
- $w(i) = (b-a)$ the width of the interval $[a,b]$. For a box, it is defined by the width of the widest interval. For example, for a box $X = ([1,2],[4,5],[4,10])$, $w(X) = max(1,1,6) = 6$.

A particularly important operator is the bisection. In fact, it is the operator that permits to explore smaller parts of the search domain. This operator aims at splitting an interval i (respectively a box B through a specific direction) in order to generate two intervals i_1 and i_2 such verifying $i = i_1 \cup i_2$ (respectively two boxes B_1 and B_2 such that $B = B_1 \cup B_2$).

The bisection of an interval $[a..b]$ generates the two equal intervals $\left[a, \frac{a+b}{2}\right]$ and $\left[\frac{a+b}{2}, b\right]$. The bisection of the box ([1,2],[4,5],[4,10]) by its largest side generates the two boxes ([1,2],[4,5],[4,7]) and ([1,2],[4,5],[7,10]).

The operators that deal with the constraints are called contractors. The contractors allow reducing the search space for optimization problems by eliminating non-consistent parts. They are issued from the constraint satisfaction problems solving (CSP) and adapt its constraint propagation techniques. The aim of the contractors is to make projections of the constraints on the domains of the variables in order to eliminate the non-feasible parts. Consider the atomic constraint $C: x_1 = x_2 + x_3$ and the intervals D_1, D_2 and D_3 respectively for x_1, x_2 and x_3, a possible contractor can be computed as follows:

$$\begin{cases} D_1 = D_1 \cap (D_2 + D_3) \\ D_2 = D_2 \cap (D_1 - D_3) \\ D_3 = D_3 \cap (D_1 - D_2) \end{cases} \quad (2)$$

Of course, this contractor was possible assuming that the constraint has a simple form. For more complex constraints, two solutions exist:

1. A method based on linearization using centered forms such as Taylor first order extension [6]. For a non linear constraint $C(x) = 0$ defined on a box X, it is possible to express Taylor's first order interval extension:

$$\forall y \in X, C(X) \subseteq C(y) + G(X)(X - y) \quad (3)$$

where $G(X)$is the gradient's enclosure of C. Then, we can express each variable X_i using the others:

$$X_i = X_i \cap \left(y_i + \frac{-C(y) - \sum_{j=1, j\neq i}^{n} G_j(X)(X_j - Y_j)}{C_i'(X)}\right) \quad (4)$$

where X_i the i^{th} component of X, $C_i'(X)$ the i^{th} component of $G(X)$. Therefore, to use this contraction technique, it is necessary to compute all the first order derivatives of the constraints. Hand computing of all the derivatives is not an easy task for real engineering design problems. To handle this issue, techniques and software for automatic differentiation are available, such as Matlab and Pro@Design (DPT, 2010) [7].

2. A method based on the calculus tree commonly named Hull consistency [8. The central idea is to create intermediate constraints in order to transform a complex constraint into a set of elementary constraints (atomic constraints). This technique has been tested and has shown good performances [9].

Compared to the unconstrained IBBA, three main adjustments are made to implement a constrained IBBA (also called Branch & Prune):

- each box taken from the $WorkList$ is systematically pruned (contracted),
- during the updating step, a feasibility test is added (the upper bound must be feasible)

The stopping test contains an additional condition $\left(\tilde{f} - b_{inf} < \varepsilon_{bound}\right)$. This test reflects a requirement on the quality of the solution. Through this test, we are certain that the global minimum is greater than $(\tilde{f} - \varepsilon_{bound})$.

For more details about interval analysis see (references), and about contractors see (references).

III. Interval Branch and Bound Algorithm and Reformulation

The Interval Branch and Bound Algorithm (IBBA) works by bisecting the search space into small boxes, and discarding the boxes that cannot contain the global minimum. This algorithm is assisted by contractors; operators that enable to propagate the constraints in order to reduce the search space by eliminating the non-consistent regions.

During the execution of the classic IBBA, the algorithm maintains two lists of boxes; the $WorkList$ and the $ResultList$. The $WorkList$ contains the boxes that have not been discarded yet and that must be treated, and the $ResultList$ contains the boxes that have not been discarded, have a maximum width of ε_{width} and in which the lower bound of f (the objective function) is higher than the current upper bound plus an ε_{bound} precision . The boxes of the $ResultList$ necessarily contain the global minimum if it exists.

The algorithm maintains through the iterations an upper bound of the global minimum. This upper bound is necessarily feasible. An important test that permits to discard boxes is the upper bound test (exclusion operator). Consequently, updating the upper bound more often permits to make the algorithm converge faster. In the classical algorithm, the upper bound is updated with the center of the current box. In fact, the design parameters are selected independently of the constraints, and independently of each other. We noticed that for the problems with tight constraints, such as equality constraints, the updating step is very difficult. This is explained by the tightness of the constraints and the selection of the potential upper bound. In fact, the upper bound must respect all the constraints simultaneously. Moreover, design engineers always relax the equality constraints in order to make the updating step possible.

The reformulation aims at making the updating step possible and easier, even without relaxing the constraints. The reformulation introduced in this paper takes advantage of specific properties of the analytical models. Currently, we suppose that the constraints of the analytical model are explicit. Knowing that, the constraints have the following form:

$$x_k = g_k(x) \tag{5}$$

where x_k is a design parameter and x the design parameter vector. The whole optimization problem may be expressed as follows:

$$\begin{cases} min_{x \in X \subseteq \mathbb{R}^n} f(x) \\ x_k = g_k(x), k = 1, \dots, \mathrm{r} \end{cases} \tag{6}$$

where $X = \{[l_i, u_i], i = 1, \dots, n\}$ are the bound constraints, $G = \{g_k, k = 1, \dots, r\}$ is the set of r analytic constraints and $x = (x_1, \dots, x_n)$ is a realization of all the variables x_i. It must be noted that the inequality constraints may be reformulated into equality ones by adding new variables. The design parameters may be classified in two categories: inputs and outputs. Following this model, the output variables are $x_1, \dots, x_r$. The inputs are the remaining variables $x_{r+1}, \dots, x_n$. Indeed, for a given input set, the output variables can be computed using the analytic constraints. In order to make this possible, it is necessary to address the equations in a specific order. The reformulation will be detailed in the examples.

This property will be used in the classical IBBA, in order to improve the updating step. The optimization problem variables will be separated into two categories: the inputs and the outputs. As the outputs can be computed via the inputs using the constraints, the IBBA will only seek the optimal inputs. Consequently, the reformulation reduces the problem's dimension. In the updating step, only inputs will be selected independently, the outputs will be then computed. The feasibility test will be replaced by a belonging test. In fact, knowing that, by construction, the combination of the inputs and outputs respects the analytical constraints, the only required condition in order to have a feasible point is to ensure that the outputs belong to their bounds.

In the adapted algorithm (see Algorithm I), we no longer consider the whole search space, but only the input box. This allows us to reach the specified precision in fewer iterations. Also, the feasibility test consists in a belonging test. Consequently, this test would be easier if the output variables vary in large intervals.

Algorithm I
Constrained global optimization algorithm with reformulation

1: Initialize $X_0^I \leftarrow$ the initial inputs box
2: Initialize $\tilde{f} \leftarrow +\infty$
3: Initialize the list $WorkList \leftarrow (+\infty, X_0^I)$: The elements of the $WorkList$ have the form: (lower bound of f, Box).
4: Initialize the list $ResultList$: $empty$

while ($WorkList$ not empty)
  5: take an element (v_i, X_i^I) from $WorkList$
  6: bisect X_i^I into two boxes: X_{i1}^I and X_{i2}^I
for j from 1 to 2
  7: contract the box using contractors
  8: compute the lower bound of f in X_{ij}^I : b_{inf}
  if ($b_{inf} > \tilde{f}$)
    9: discard X_{ij}^I
  else
    10: $C_{ij}^I \leftarrow Center(X_{ij}^I)$
    ***11:** compute the corresponding outputs: C_{ij}^O*
    if ($f(C) < \tilde{f}$ and $X_{ij}^O \in X_0^O$)
      12: $\tilde{f} \leftarrow f(C)$
    end if
    if ($w(X_{ij}^I) < \varepsilon_{width}$ and $\tilde{f} - b_{inf} < \varepsilon_{bound}$)
      13: $ResultList \leftarrow ResultList + (b_{inf}, X_{ij}^I)$
    else
      14: $WorkList \leftarrow WorkList + (b_{inf}, X_{ij}^I)$
    end if
  end if
end for
end while

IV. NUMERICAL RESULTS

A. Electromagnetic Actuator

The structure of the actuator and its behavior are modeled by the relations that are the expression of the electromagnetic conversion and the flux conservation. The significations of the design parameters are detailed in tables I and II.

The electromagnetic torque is given by the following relation:

$$\Gamma_{em} = \frac{\pi}{2\lambda}(1 - K_f)\sqrt{k_r \beta E_{ch} E D^2 (D + E)} B_e \qquad (7)$$

E_{ch} represents the machine heating, and is defined by the relation:

$$E_{ch} = k_r E J_{cu}^2 \qquad (8)$$

An empirical relation between the magnetic leakage coefficient and the geometric dimensions has been established (p stands for the number of pole pairs):

$$K_f \cong 1.5 p \beta \frac{e+E}{D} \qquad (9)$$

The magnetic field in the air gap, supposed exclusively radial, is given by:

$$B_e = \frac{2 l_a M}{D \log\left[\frac{D+2E}{D-2(l_a+e)}\right]} \qquad (10)$$

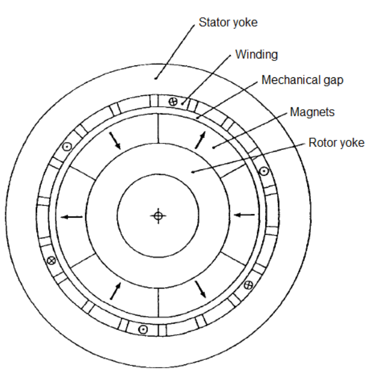

FIG II. THE ELECTROMAGNETIC ACTUATOR STRUCTURE

We give here the expression of the yoke thickness, in which we assume that the inter-polar leakage is negligible:

$$C = \frac{\pi \beta B_e}{4 p B_{fer}} D \qquad (11)$$

The number of pole pairs is obtained by:

$$p = \frac{\pi D}{\Delta_p} \qquad (12)$$

The structural model is formed exclusively by equality constraints. Other constraints related to the variation domains of the parameters are given; between a small value and a large value on the scale of the considered parameter. Theses domains are given in table I.

In the model considered, the machine must be dimensioned in order to minimize the volume of the active parts:

$$V_u = \pi \frac{D}{\lambda}(D + E - e - l_a)(2C + E + e + l_a) \qquad (13)$$

The constraints of the electromagnetic actuator model are explicit. Initially, without reformulation, the output parameters are $\Gamma_{em}, E_{ch}, K_f, B_e, C$ and p. In order to ensure that the outputs vary in intervals so as to make the feasibility test easier, we have rearranged the concerned constraints.

TABLE I
THE VARIABLE DESIGN PARAMETERS: SIGNIFICATION AND INTERVALS

Symbol	Signification	Interval
$D(m)$	bore diameter	[0.001,0.5]
$B_e(T)$	magnetic field in the air gap	[0.1,1]
K_f	semi-empiric magnetic leakage coefficient	[0.01,0.3]
$J_{cu}(A/m^2)$	current areal density	$[10^5,10^7]$
$e(m)$	mechanical air gap thickness	[0.001,0.005]
$l_a(m)$	permanent magnets thickness	[0.001,0.05]
$E(m)$	winding thickness	[0.001,0.05]
$C(m)$	yoke thickness	[0.001,0.05]
β	polar arc factor	[0.8,1]
λ	length	[1,0.5]

TABLE II
THE FIXED DESIGN PARAMETERS: SIGNIFICATION AND VALUES

Symbol	Signification	Value
k_r	coil winding filling factor	0.7
$B_{fer}(T)$	magnetic field in the iron	1.5
$E_{ch}(A/m)$	machine warm-up	10^{11}
$\Gamma_{em}(N.m)$	electromagnetic torque	10
$M(T)$	magnetic polarization	0.9
$\Delta_p(m)$	polar step	0.1

The reformulated design model has the following form:

$$\min \ V_u = \pi \frac{D}{\lambda}(D + E - e - l_a)(2C + E + e + l_a)$$

$$s.t \begin{cases} D = \frac{p\Delta_p}{\pi} \\ J_{cu} = \sqrt{\frac{E_{ch}}{k_r E}} \\ B_e = \frac{2 l_a M}{D \log\left[\frac{D+2E}{D-2(l_a+e)}\right]} \\ K_f = 1.5 p \beta \frac{e+E}{D} \\ \lambda = \frac{\pi}{2\Gamma_{em}}\left(1 - K_f\right)\sqrt{k_r \beta E_{ch} E D^2 (D+E)} B_e \\ C = \frac{\pi \beta B_e}{4 p B_{fer}} D \end{cases} \qquad (14)$$

In order to assess the reformulation, we compared the numerical results given by the IBBA with and without the reformulation. Tables III and IV show that without the reformulation, we relaxed the equality constraints by $\epsilon_t = 10{-}3$. Consequently, an equality constraint pi=f(P) is replaced by $p_i \in [f(P) - \epsilon_t .. f(P) + \epsilon_t]$.

TABLE III
NUMERICAL RESULTS: ACTUATOR OPTIMIZATION 1

	Solution (10^{-4})	Relaxation	Iterations	CPU time (s)
Non formulated model	6.0743	10^{-3}	60 151 649	1123
Formulated model	6.0736	0	470 960	13

TABLE IV
NUMERICAL RESULTS: ACTUATOR OPTIMIZATION 2

	Total updates	Feasible solutions found
Non formulated model	12	124 434
Formulated model	34	705 461

Compared to the non-reformulated model, the use of reformulation considerably reduces the total number of iterations necessary to reach the global optimum. Also, we noted an increase in the number of updates (from 12 to 34), explained by the increase of the number of feasible solutions found (from 124 434 solutions in 60 151 649 iterations to 705 461 solutions in 470 960 iterations). This observation confirms that the reformulation makes the updating step easier. This allows discarding more boxes by the bound test, and thus making the convergence faster.

B. Transformer

The second design example is the design of a transformer. The design model and details can be. found in [10]. It consists of 21 equality constraints and 53 variables. 21 are fixed variables, 30 are free variables (vary in $[0,+\infty]$), and 2 variables vary respectively in [0.4,100] and [100,600]. The same procedure has been used to reformulate the model. The tables V and VI summarize the principal results.

Without reformulation, we consider that the variables are independent. This considerably affects the updating step. In the case of the transformer design, the updating step consists in fixing 32 design parameters independently. In order to have an upper bound of the global minimum, the selected point must be feasible i.e. all the 21 equality constraints must be fulfilled. Knowing that the selected parameters are fixed independently, the probability is negligible. This explains why the problem has not been solved without reformulation. The reformulation reduces the problem to its freedom degrees (2 freedom degrees) which made the updating step much more frequent. In fact, without reformulation, no feasible solution has been found after 180 millions iterations (even with relaxations). With the reformulation, one or two feasible solutions are found on average at each iteration (1 673 951 feasible points in 1 237 187 iterations). Consequently, the algorithm converges in a relatively short time (5.2 seconds).

TABLE V
NUMERICAL RESULTS: TRANSFORMER OPTIMIZATION 1

	Solution	Relaxation	Iterations	CPU time (s)
Non formulated model		Not solved		
Formulated model	2085130.462	0	1 237 187	33

TABLE VI
NUMERICAL RESULTS: TRANSFORMER OPTIMIZATION 2

	Total updates	Feasible solutions found
Non formulated model	0	0
Formulated model	463	1 673 951

REFERENCES

[1] P. Alotto, M. Gaggero, G. Molinari, and M. Nervi, “A ’design of experiment’ and statictical approach to enhance the ‘generalized response surface’ method in the optimization of multiminima problems,” *IEEE Trans. Magn.*, vol. 33, no. 2, pp. 1896-1899, 1997.

[2] F. Messine, B. Nogarede, and J. L. Lagouanelle, “Optimal design of electromechanical actuators: A new method based on global optimization,” *IEEE Trans. Magn.*, vol.34, pp. 299-308, 1998.

[3] E. Hansen, and G. W. WALSTER, “Global Optimization Using Interval Analysis,” *Marcel Dekker*, 2004.

[4] MOORE, R. E. Interval Analysis. Englewood Cliffs N. J., 1966.

[5] KEARFOTT, R. B. Interval Computations: Introduction, Uses and Ressources. Euromath Bulletin, v. 2, p. 95-112, 1996.

[6] WALSTER, G. W. Global Optimization Using Interval Analysis. Marcel Dekker, 2004.

[7] http://www.designprocessing.com/

[8] F. Benhamou, F. Goualard, L. Granvilliers and J. F. Puget, “Revising Hull and Box Consistency,” *International Conference on Logic Programming/Joint International Conference and Symposium on Logic Programming*, pp. 230-244, 1999.

[9] MESSINE, F. Deterministic Global Optimization using Interval Constraint Propagation Techniques. RAIRO-OR, v. 38, p. 277-294, 2004.

[10] M. Poloujadoff and R.D. Findlay, “A procedure for illustrating the effect of variation of parameters on optimal transformer design,” *IEEE Trans. Pow. Sys.*, vol. 1, no. 4, pp. 202-205, 1986.