

# Discussion about Formulations and Resolution Techniques of Electrical Machine Design Problems

Sonia Cafieri\*, Leo Liberti\*\*, Frédéric Messine\*\*\*, Bertrand Nogarede\*\*\*\*  
 sonia.cafieri@enac.fr, liberti@lix.polytechnique.fr, messine@n7.fr, nogarede@n7.fr

**Abstract**—In this paper, we discuss, through a simple example, the impact of two different but equivalent formulations used by standard local optimization solvers (here *fmincon* of *MatLab*). We show that even if the two formulations are equivalent in a mathematical sense (no loss of global optima) it is not completely true in a numerical way; using 1000 starting points, we show that it is quite difficult for the designer to find a starting point yielding a convergence of the algorithm to a local minimum and to find better points yielding the global solution (previously found using a global optimization algorithm). Furthermore, we discuss how to deal with the insertion of the integer variable  $p$ , representing the number of pole pairs of the machine, inside the problem of design which uses a standard local continuous optimization code to be solved.

**Keywords:** analytical model, formulation, local optimization, inverse problem, design, electrical machine, mixed integer non-linear programming.

## I. INTRODUCTION

The interest of the *electromagnetical actuators design* combining *optimization algorithms* and *analytical models* has already been shown in a lot of works, [5]–[8], [10], [11], [13], [14]. The design of electromechanical actuators is understood as an *inverse problem*, i.e. from the characteristic values given by the schedule of conditions (for example the torque), obtain the *structure*, the *dimensions* and the *material compositions* of the actuator constitutive parts, [1], [2].

In [9], we have shown that even for *optimal dimensioning problems* which can be formulated as *continuous-constrained optimization problems*, it was really difficult to find the *global optima* using classical optimization algorithms, like Lagrangian Augmented Techniques [5], [14]. Moreover, we have shown that *exact global optimization* based on *interval analysis* can be perfectly adapted to solve these kinds of problem [2], [7]–[9].

Unfortunately, the exact global algorithm named IBBA (for Interval Branch and Bound Algorithm [3], [4], [7], [12]), is difficult to implement on a computer (about 10000 lines of a Fortran90 code), and nowadays, it does not exist an industrial version of IBBA or of a similar one. Furthermore, its application to solve such problems is not so easy and it often needs to develop or to adapt some parts of the code.

\*ENAC, 7 av. E. Belin, 31055 Toulouse, France

\*\*LIX, École Polytechnique, Palaiseau, France

\*\*\*ENSEEHT-IRIT, UMR-CNRS 5055, 2 rue Charles Camichel, BP 7122, 31071 Toulouse, Cedex 7, France

\*\*\*\*Electrodynamics-EM3 Research Group, ENSEEHT-LEEI, UMR-CNRS 5828, 2 rue Charles Camichel, BP 7122, 31071 Toulouse, Cedex 7, France

Concerning stochastic algorithms, such as genetic algorithms or simulated annealing, they are difficult to be efficiently used to solve this kind of problems due to the equality and inequality constraints which define many small realizable domains.

In fact, the *inverse problems* of the electromechanical actuators design are more general than *optimal dimensioning problems*, see [2]. Thus, these problems must be formulated as *mixed-constrained optimization problems*, see [1], [2] for more complete formulations.

In this paper, we discuss about the impact of different formulations on the efficient use of a deterministic local optimization solver (*fmincon* of *MatLab*), through a simple example of a slotless electrical rotating machine with magnetic effects. This example was first presented in [5] and also studied in a lot of papers such as [5], [8]–[11], [14].

The design problem can be formulated as follows:

$$\left\{ \begin{array}{l} \min \quad \pi \beta l_a \frac{D}{\lambda} (D - 2e - l_a) \\ \text{s.t.} \quad \Gamma_{em} = \frac{\pi}{2\lambda} (1 - K_f) \sqrt{k_r \beta E_{ch} E} D^2 (D + E) B_e \\ \quad E_{ch} = A J_{cu} = k_r E J_{cu}^2 \\ \quad K_f = 1.5 p \beta \frac{e + E}{D} \\ \quad B_e = \frac{2 l_a P}{D \ln \left( \frac{D + 2E}{D - 2(e + e)} \right)} \\ \quad C = \frac{\pi \beta B_e}{4 p B_{iron}} D \\ \quad p = \frac{\pi D}{\Delta_p} \end{array} \right. \quad (1)$$

where  $D(m)$  is the bore diameter,  $\lambda$  the diameter over length ratio,  $l_a(m)$  the thickness of the permanent magnets,  $E(m)$  the winding thickness,  $C(m)$  the thickness of yoke,  $\beta$  the polar arc factor,  $B_e(T)$  the magnetic field in the air-gap,  $J_{cu}(A/m^2)$  the current areal density,  $K_f$  a semi-empiric magnetic leakage coefficient (established by numerical simulations),  $e(m)$  the thickness of the mechanical air gap,  $p$  the number of pole pairs,  $k_r$  a coefficient of occupation,  $B_{iron}$  the magnetic field in the iron,  $\Gamma_{em}$  the electromagnetical torque,  $P$  the magnetic polarization and  $\Delta_p$  the polar step. For this study, we fix  $\Gamma_{em} = 10 N.m$ ,  $P = 0.9 T$ ,  $k_r = 0.7$ ,  $B_{iron} = 1.5 T$ ,  $E_{ch} = 10^{11} A/m$  and  $\Delta_p = 0.1 m$  the polar step as in [5]. The other parameters can vary inside the following intervals:  $D(m) \in [0.01, 0.5]$ ,  $\lambda \in [1, 2.5]$ ,  $l_a(m) \in [0.003, 0.05]$ ,  $E(m) \in [0.001, 0.05]$ ,  $C(m) \in [0.001, 0.05]$ ,  $\beta \in [0.8, 1]$ ,  $B_e(T) \in [0.1, 1]$ ,  $J_{cu}(A/m^2) \in [10^5, 10^7]$ ,  $K_f \in [0.01, 0.3]$ ,  $e(m) \in [0.001, 0.005]$  and  $p \in \{1, \dots, 10\}$ . Furthermore, the function to be minimized corresponds to the magnet volume ( $V_m = \pi \beta l_a \frac{D}{\lambda} (D - 2e - l_a)$ ).

This problem is mixed (the parameter  $p$  is an integer number) and also non-homogeneous (some parameters, such as  $J_{cu}$ , have high values and the others are quite small, such as  $D, e, \dots$ ). The designer must take care about this point.

In Section II, we present two different but equivalent formulations of the design problem and we discuss about their efficiency when the optimization problem is solved using the function 'fmincon' of *MatLab*<sup>7</sup>. In Section III, we discuss about the fact that the parameter  $p$  representing the number of pole pairs can become an integer variable of the optimization code. This yields a mixed-integer non-linear optimization problem which is difficult to solve using classical continuous local search algorithms such as *fmincon* of *MatLab*. Some concluding remarks are given in Section IV.

## II. FORMULATIONS AND STARTING POINTS

One of the main difficulty in the use of a deterministic local optimization algorithm is the introduction of a starting point to initialize it. Then, we have three possibilities: (i) the algorithm converges to a local solution; (ii) the algorithm works very slowly and, after a while, it reaches the maximum number of iterations or of function evaluations (here, both are fixed to 30000); (iii) the algorithm does not converge and fails.

If we are on stage (ii), we can restart the algorithm by increasing the maximum number of iterations and of function evaluations but it could not work (case (iii) or still case (ii)).

In this section,  $p$  is fixed to 5 and we discuss about the two following formulations:

$$\left\{ \begin{array}{l} \text{Formulation 1:} \\ - D, \lambda, l_a, E, C, \beta, J_{cu}, K_f, e: 9 \text{ variables.} \\ - B_e \text{ is a function depending on } D, \lambda, \dots \\ - \text{Equations in (1) yield 5 equalities constraints} \\ \text{and 2 inequality ones: } 0.1 \leq B_e(D, \lambda, \dots) \leq 1. \\ - \text{Add constraints to the bounds of the variables.} \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \text{Formulation 2:} \\ - D, \lambda, l_a, E, C, \beta, J_{cu}, K_f, e, B_e: 10 \text{ variables.} \\ - \text{Equations in (1) yield 6 equalities constraints.} \\ - \text{Add constraints to the bounds of the variables.} \end{array} \right. \quad (3)$$

These two explicit formulations are:

- Formulation 1:

$$\left\{ \begin{array}{l} \min_{(D, \lambda, \dots) \in \mathcal{D} \subset \mathbb{R}^9} V_m(D, \lambda, \dots) = \pi \beta l_a \frac{D}{\lambda} (D - 2e - l_a) \\ \Gamma_{em} = \frac{\pi}{2\lambda} (1 - K_f) \sqrt{k_r \beta E_{ch} E} D^2 \times \\ \quad (D + E) B_e(D, \lambda, \dots) \\ E_{ch} = A J_{cu} = k_r E J_{cu}^2 \\ K_f = 1.5 p \beta \frac{e + E}{D} \\ C = \frac{\pi \beta B_e}{4 p B_{iron}} D \\ p = \frac{\pi D}{\Delta_p} \\ B_e(D, \lambda, \dots) \geq 0.1 \\ B_e(D, \lambda, \dots) \leq 1 \end{array} \right. \quad (4)$$

Where  $B_e(D, \lambda, \dots)$  is defined as an auxiliary function and returns  $\frac{2l_a P}{D \ln\left(\frac{D+2E}{D-2(l_a+e)}\right)}$ .

- Formulation 2:

$$\left\{ \begin{array}{l} \min_{(D, \lambda, \dots) \in \mathcal{D} \subset \mathbb{R}^{10}} V_m(D, \lambda, \dots) = \pi \beta l_a \frac{D}{\lambda} (D - 2e - l_a) \\ \Gamma_{em} = \frac{\pi}{2\lambda} (1 - K_f) \sqrt{k_r \beta E_{ch} E} D^2 \times \\ \quad (D + E) B_e \\ E_{ch} = A J_{cu} = k_r E J_{cu}^2 \\ K_f = 1.5 p \beta \frac{e + E}{D} \\ B_e = \frac{2l_a P}{D \ln\left(\frac{D+2E}{D-2(l_a+e)}\right)} \\ C = \frac{\pi \beta B_e}{4 p B_{iron}} D \\ p = \frac{\pi D}{\Delta_p} \end{array} \right. \quad (5)$$

$\mathcal{D}$  define the initial domain where the solution is searched for; it defines lower and upper bounds for all the variables.

Mathematically this two formulations are equivalent because the only difference is the addition of a new variable  $B_e \in [0.1, 1]$  in Formulation 2 and the transformation of 2 inequality constraints into an equality one. Therefore, this reformulation involves nothing in the mathematical point of view: any solution (global minimum) of Formulation 1 is also a solution for Formulation 2 and conversely.

Table I presents numerical results obtained for the two described formulations, using the same starting points. These starting points are randomly generated between the bounds of the variables by using the function 'rand' of *MatLab* (for Formulation 2, one random number is necessary for  $B_e$ ). Some attentions must be paid to discard the fact that using some random numbers, some evaluations of negative logarithms could be performed.

TABLE I  
COMPARISON BETWEEN FORMULATIONS 1 AND 2:  $p = 5$

	Formulation 1	Formulation 2
% local min	44.4%	39.6%
% best min	3.5%	4.6%
best min value	$7.3581e^{-5}$	$7.3571e^{-5}$
worst local min	$1.1524e^{-3}$	$1.1679e^{-3}$
best CPU-time	0.00s	0.00s
worst CPU-time	25.97s	23.75s
$x_0 = mid$	$8.4162e^{-4}$	—

We performed 1000 iterations of 'fmincon' for each formulation using 1000 randomly generated starting points. In the first line of Table I we give the percentage of starting points yielding a local minimum (the percentage of failure is quite high). In the second line, we show the percentage yielding the best solution found during the 1000 iterations (comparison of the values at  $10^{-7}$ ). This percentage is very low, confirming that a lot of care must be paid in the choice of a good starting point. In the following of the table, the best and worst values of local minima and the best and worst CPU-times used for one iteration are reported. The last line indicates that taking the starting point in the middle of the hypercube defined by the bounds of the variables, Formulation 1 provides a bad solution and Formulation 2 does not converge; this proves the non-equivalence of the

two formulations in a numerical sense. However, we note that the two formulations provide similar results. The percentage of time when the two formulations provide equivalent local values (within a tolerance of  $10^{-7}$ ) at the same iteration is about 3.3%. We remark that Formulation 2 is more efficient providing best convergence to the best local minimum (which is the global one); it seems to be more efficient to use this formulation with the introduction of a new variable  $B_e$  because the expressions of the equations to be satisfied become less complicated.

The best found local solutions for Formulation 1 and Formulation 2 are the following:

Best local min	Formulation 1	Formulation 2
$D^* =$	0.159155	0.159155
$\lambda^* =$	2.5	2.5
$l_a^* =$	0.003	0.003
$E^* =$	0.002636	0.002809
$C^* =$	0.005033	0.004911
$\beta^* =$	0.8	0.8
$B_e^* =$	0.377488	0.368305
$J_{cu}^* =$	7361029.21	7130912.95
$K_f^* =$	0.153328	0.160244
$e^* =$	0.001431	0.001441

Thus, the two formulations provides two similar best local solutions.

Moreover, we note that the number of pole pairs fixed to  $p = 4$  yields a best local minimum equal to  $8.4272e^{-5}$  for Formulation 1 and  $8.4097e^{-5}$  for Formulation 2, which are about 15% larger than the value corresponding to the case  $p = 5$ .

In the next section, we discuss how to deal with  $p$  as an integer variable of the optimization problem yielding a difficult *mixed integer non-linear program*.

### III. INTEGER PARAMETER

In such a dimensioning problem, the number of pole pairs  $p$  was previously considered as a fixed parameter,  $p = 5$ . In fact, Quasi-Newton algorithms such as *fmincon* of MatLab needs continuous formulations and twice differentiable functions. As a consequence, integer parameters must be dealt with attention.

In the literature, the integer parameter  $p$  is usually considered as a continuous one. The user has to convert the obtained real value of  $p$  into an integer one, [5], [11]. Here, by considering  $p \in [1, \dots, 10]$ , we performed a multistart method with 1000 starting points (randomly generated) and the results are presented in Table II, where  $p^*$  corresponds to the best found local minimum and  $p_{mid}$  to the local minimum found using the middle of the the domain as a starting point.

The best found local minima for Formulation 1 and Formu-

TABLE II  
COMPARISON BETWEEN FORMULATIONS 1 AND 2 WITH  $p \in [1, 10]$

	Formulation 1	Formulation 2
% local min	51.7%	58.1%
% best min	1.4%	1.8%
best min value	$6.9894e^{-5}$	$6.9851e^{-5}$
$p^*$	4.8617	4.8603
worst local min	$1.3261e^{-3}$	$1.3740e^{-3}$
best CPU-time	0.00s	0.02s
worst CPU-time	15.02s	20.47s
$x_0 = mid$	$8.8957e^{-4}$	$2.2991e^{-4}$
$p_{mid}$	3.4728	3.7814

lation 2 are:

Best local min	Formulation 1	Formulation 2
$D^* =$	0.154754	0.154707
$\lambda^* =$	2.5	2.5
$l_a^* =$	0.003	0.003
$E^* =$	0.003066	0.002659
$C^* =$	0.00506	0.005356
$\beta^* =$	0.8	0.8
$B_e^* =$	0.37953	0.401686
$J_{cu}^* =$	6825814.56	7329490.86
$K_f^* =$	0.153290	0.137949
$e^* =$	0.001	0.001
$p^* =$	4.861748	4.8603

We can note that these two solutions are different even if their corresponding volume values are very close to each other.

From results in Table II, we remark that Formulation 2 gives again the best results in terms of the percentage of the found local minima and of the best found local minimum. The real values of the number of pole pairs  $p$  in both cases are very close and larger than 4.86, thus confirming that the integer value  $p = 5$  (used for experiments discussed in Section I) is a good choice.

Note that only 31 starting points over 1000 provide close optimal solutions where the values of the volume differ at most of  $10^{-7}$ .

If we fix  $p^* = 5$ , which is the integer value the most close to  $p^*$ , we obtain the following table containing the values of the equalities constraints. They must be close to 0 (less than  $10^{-8}$ ) to be considered as satisfied constraints:

Functions	Formulation 1	Formulation 2
$V_a =$	$6.9894e^{-5}$	$6.9851e^{-5}$
$eq_1 =$	$9.7964e^{-5}$	$3.7897e^{-4}$
$eq_2 =$	$-4.7982e^{-5}$	$-8.2906e^{-5}$
$eq_3 =$	$-4.3537e^{-3}$	$-3.9579e^{-3}$
$eq_4 =$	$1.3943e^{-4}$	$-1.2990e^{-5}$
$eq_5 =$	$1.3826e^{-1}$	$1.4987e^{-4}$
$eq_6 =$	—	$1.3974e^{-1}$

Of course, the equality constraints are now not satisfied and therefore one has to change some values of the solution previously found. In the following, we discuss some of these possibilities to deal with the integer parameter.

Another possibility to deal with the fact that  $p$  is integer, is to insert a new equality constraint which is satisfied when  $p \in \{1, \dots, 10\}$ :

$$(p - 1) \times (p - 2) \times \dots \times (p - 10) = 0.$$

Numerical results for both formulations are shown in Table III, where again  $p^*$  corresponds to the best found local minimum.

TABLE III  
COMPARISON BETWEEN FORMULATIONS 1 AND 2 WITH  
 $(p-1) \times \dots \times (p-10) = 0$

	Formulation 1	Formulation 2
% local min	31.0%	30.9%
% best min	0.2%	0.6%
best min value	$7.3587e^{-5}$	$7.3676e^{-5}$
$p^*$	5	5
worst local min	$1.3885e^{-3}$	$1.3885e^{-3}$
best CPU-time	0.02s	0.02s
worst CPU-time	10.84s	14.01s
$x_0 = mid$	—	—

The best found local solutions for Formulation 1 and Formulation 2 are:

Best local min	Formulation 1	Formulation 2
$D^* =$	0.159155	0.159155
$\lambda^* =$	2.5	2.5
$l_a^* =$	0.003	0.003
$E^* =$	0.003441	0.002053
$C^* =$	0.004545	0.005556
$\beta^* =$	0.8	0.8
$B_e^* =$	0.340905	0.416680
$J_{cu}^* =$	6443484.74	8342409.94
$K_f^* =$	0.183394	0.127555
$e^* =$	0.001424	0.001331
$p^* =$	5	5

As in the results previously discussed, the two solutions are different even if close values are found for the volume of magnet.

Note that in this case only 49 starting points over 1000 provide close optimal solutions where the values of the volume differ at most of  $10^{-7}$ .

Another strategy to deal with  $p$  consists in starting from the real solutions given in Table II, and changing the last added equality constraint to a lower degree polynomial expression as follows:

$$(p - ([p^*] - 1)) \times (p - [p^*]) \times (p - ([p^*] + 1)) = 0,$$

where  $[p^*]$  represents the most close integer value of  $p^*$ . In our experiments, we used for  $p^*$  the value 4.86 that was found by the two formulations for the continuous problem. Results are reported in Table IV.

TABLE IV  
COMPARISON BETWEEN FORMULATIONS 1 AND 2 WITH  
 $(p-4) \times (p-5) \times (p-6) = 0$

	Formulation 1	Formulation 2
% local min	25.9%	26.8%
% best min	1.3%	2.6%
best min value	$7.3572e^{-5}$	$7.3663e^{-5}$
$p^*$	5	5
worst local min	$1.3885e^{-3}$	$1.3885e^{-3}$
best CPU-time	0.02s	0.02s
worst CPU-time	10.75s	12.14s
$x_0 = mid$	$7.3482e-4$	—
$p_{mid}$	4	—

We note that the percentage to find a local minimum is a little lower than in the previous case, while the percentage

to find the best local minimum is significantly improved. Therefore, this seems to show that it is beneficial to reduce the zone of research, concentrating the efforts around  $p = 5$ , in order to improve the chances to find the global minimum.

The best found local minima for Formulation 1 and Formulation 2 are:

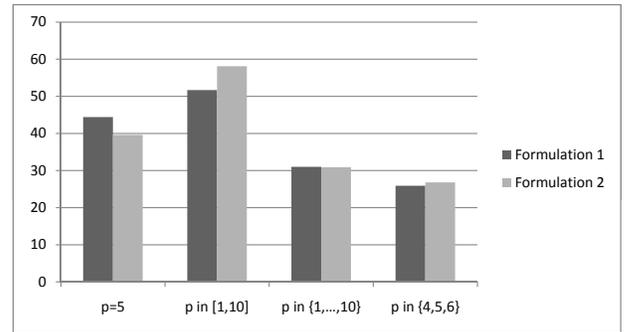
Best local min	Formulation 1	Formulation 2
$D^* =$	0.159155	0.159155
$\lambda^* =$	2.5	2.5
$l_a^* =$	0.003	0.003
$E^* =$	0.003201	0.002108
$C^* =$	0.004671	0.005497
$\beta^* =$	0.8	0.8
$B_e^* =$	0.340905	0.412257
$J_{cu}^* =$	6680741.53	8231696.33
$K_f^* =$	0.1749434	0.130198
$e^* =$	0.001440	0.001345
$p^* =$	5	5

Again the two solutions differ, despite of equivalent values of the volume of the magnet.

We remark that only 30 starting points over 1000 provide close optimal solutions, where the values of the volume differ at most of  $10^{-7}$ .

In Figures 1 and 2 we compare the two formulations in terms of the percentage of found local minima and the percentage of best found local minima respectively, considering the different choices of  $p$  discussed above. So, these figures summarize the results presented in Tables I to IV.

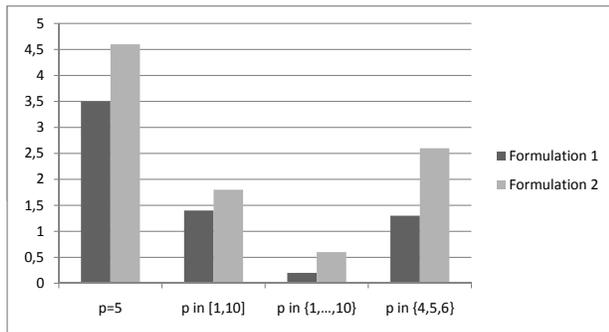
Fig. 1. Percentages of found local minima



From Figure 1, one can see that there is not a clearly most efficient method to find the largest number of local solutions. This actually depends on the considered problem. We can note that considering the parameter  $p$  as a variable, the second formulation seems to be more efficient.

In Figure 2, it is most clear that, using the second formulation, a higher number of starting points allows to obtain the best found local solution. Thus, as the values of the percentage of best found local solution are low, we can have more confidence on the local solutions found by the second formulation in order to provide the global solution.

Fig. 2. Percentages of best found local minima



#### IV. CONCLUSION

We have presented two reformulations of a design problem, showing that the designer must take care about the formulation of the problem in order to make more efficient the use of a local standard algorithm such as that proposed by MatLab in *fmincon*.

First, we have discussed the solutions found using the two formulations, where the number of pole pairs is fixed to 5 and 4. Formulation 2 provides the best percentage of local solutions found.

Second, we have discussed about the fact that  $p$  has to be an integer variable, providing a mixed integer non-linear problem. Using the two formulations and some different strategies to deal with the integer variable, we have remarked that Formulation 2 is again most efficient in providing the best local solution. However, the two best local solutions, found using one or the other formulation, are different but give the same value of the volume of the magnet. Furthermore, using the middle of the domain as a starting point, we found that Formulation 1 provides a solution which is in general far from the best one, while using Formulation 2 the process never converges.

#### REFERENCES

- [1] E. Fitani, F. Messine, B. Nogarede, *A General Analytical Model of Electrical Permanent Magnet Machine dedicated to Optimal Design*, COMPEL - International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Vol. 22, N. 4, 2003.
- [2] E. Fitani, F. Messine, B. Nogarede, *The Electromagnetical Actuators Design Problem: a General and Rational Approach*, IEEE Transactions on Magnetics, Vol. 40, N. 3, pp. 1579–1590, 2004.
- [3] E. Hansen, *Global Optimization using Interval Analysis*, Marcel Dekker, Inc. 270 Madison Avenue, New York 100016, 1992.
- [4] R.B. Keafott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1996.
- [5] A.D. Kone, B. Nogarede, M. Lajoie-Mazenc, *Le Dimensionnement des Actionneurs Electriques: un Problème de Programmation Non-Linéaire*, Journal de Physique III, France 3, pp. 285–301, 1993.
- [6] S. Huang, M. Aydin, T.A. Lipo, *A Direct Approach to Electrical Machine Performance Evaluation: Torque density Assessment and Sizing Optimisation*, ICEM 2002, Art. 235, 2002.

- [7] F. Messine, *A Deterministic Global Optimization Algorithm for Design Problems*, in C. Audet, P. Hansen, G. Savard (editors), *Essays and Surveys in Global Optimization*, Kluwer, pp. 267–294, 2005.
- [8] F. Messine, *Deterministic Global Optimization using Interval Constraint Propagation Techniques*, RAIRO-Operations Research, Vol. 38, N. 4, pp. 277–294, 2004.
- [9] F. Messine, B. Nogarede, J.L. Lagouanelle, *Optimal design of electromechanical actuators: a new method based on global optimization*, IEEE Transactions on Magnetics, Vol. 34, N. 1, pp 299–307, 1998.
- [10] B. Nogarede, *Machines Tournantes: principes et constitutions*, Techniques de l'Ingénieur, D3410 and D3411, France, 2001.
- [11] B. Nogarede, A.D. Kone, M. Lajoie-Mazenc, *Optimal Design of Permanent-Magnet Machines using Analytical Field Modeling*, Electromotion, Vol. 2, N 1, pp.25–34, 1995.
- [12] H. Ratschek, J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood Limited Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England, 1988.
- [13] G.R. Slemon, X. Liu, *Modeling and design optimization of permanent magnet motors*, Electric Machines and Power Systems, pp.71–92, 1992.
- [14] F. Wurtz, J. Bignon, C. Poirson, *A Methodology and a Tool for the Computer-Aided Design with Constraints of Electrical Devices*, IEEE Trans. Mag., Vol. 32, pp. 1429–1432, 1996.