Fast optimization of an IPMSM with Space Mapping Technique

S. VIVIER, D. LEMOINE, G. FRIEDRICH *(IEEE Senior member)*
Laboratoire d'Électromécanique de Compiègne (LEC)
Université de Technologie de Compiègne (UTC)
Compiègne, FRANCE
stephane.vivier@utc.fr

Abstract— The work presented in this article relates to the design procedure of a permanent magnet synchronous machine, by the use of Output Space Mapping type methods. This type of optimization technique reduces significantly the time necessary to the search for optimal configurations, by the use of a fast modeling of the device to be optimized. At the same time, a second model, slower but more accurate, ensures the validity of the optimal configurations obtained. Two techniques belonging to this family of methods are applied in this study. Their results and performances are compared.

I. Introduction

This paper discusses the approach used for the optimization of an internal permanent magnet synchronous machine (IPMSM). The main idea of this work is based on the desire to reduce as much as possible the time required for the design of this machine with appropriate mathematical and computer tools.

The first step is achieved by building a finite element model of the machine to be optimized. As part of this work, only the magnetostatic aspects are considered. Dedicated tools have been developed in the laboratory to minimize the time required for the definition and the building of the finite element model. This model is set to be used in the next step.

The second step is the design process itself. This study is performed by solving a multidimensional constrained equivalent optimization problem.

To reduce the time required for design, the "Space Mapping" optimization approach is used. In this paper, two methods in this class are applied: the "Output Mapping" (OM) and the "Manifold Mapping" (MM). Both techniques use regular adjustments of output values of two models of the same system (in our case the synchronous machine). This common feature presents the advantage to benefit from the speed of a fast model, while exploiting the reference results of a second one.

Results of these two approaches are presented and compared. In both cases, it is clear that the optimization process using two models (rather than a single one traditionally) outperforms conventional approaches in terms of efficiency [1].

II. Space Mapping methods

A. Presentation

The method used for the research of the best building characteristics of the actuator, belongs to the "Space Mapping" (SM) method family [2] [3]. As will be shown in the following, these techniques have in common their functioning principle: the simultaneous and coordinated use of two models of the same system to be optimized. These two models differ by the precision of their results and by their evaluation speed.

In very general terms, the optimization problem to be solved is equivalent to find the definition of the vector of the n input values (collected in vector $\boldsymbol{x}$) that minimizes the "distance" between the m output values $\boldsymbol{f}(\boldsymbol{x})$ given by the modeling, and those one wants to reach, gathered in vector $\boldsymbol{y}$. This gives:

$$\boldsymbol{x}^* = \operatorname{argmin}\left\| \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y} \right\| \quad (1)$$

Asterisk indicates an optimal configuration. Values given by vector $\boldsymbol{y}$ can correspond to targets specified in specification books.

Function $\boldsymbol{f}(\boldsymbol{x})$ is a reference model, that is to say, a set of mathematical methods giving reliable and accurate values of the output entities (performances, size of the machine, etc.) based on the values of input variables (geometrical specifications, power supply, etc.). $\boldsymbol{f}(\boldsymbol{x})$ is called "*fine model*".

In most cases, the good accuracy of this model comes with high computing times, due to the complexity of the mathematical relationships employed, the amount of data generated, etc. Generally, this is a costly modeling (mainly in terms of time to get results). Its heaviness forbids its intensive use by design algorithms using the iterative optimization approach.

However, to avoid losing the knowledge represented by the fine reference model, while keeping the objective of solving the optimization problem above, it becomes necessary to use a second model, similar to the reference one, but much faster. This model is called "*coarse model*" and is denoted $\boldsymbol{c}(\boldsymbol{x})$.

Hence, the coarse model should be fast to be used intensively by any optimization algorithm, that is to say, to be called many times.

It will be assumed for the future that both fine and coarse models are defined using the same m output quantities, and the same n input variables (although this is not mandatory).

Results of this optimization can then be compared to that obtained for the same optimum conditions by a single evaluation of the fine model. If there is a difference between the outputs of the two models at this point, an adjustment is achieved. This process can be repeated several times to get even closer to better optimal conditions.

This adjustment is achieved through a transformation $\mathbf{S}$ applied to the outputs of the coarse model $\boldsymbol{c}(\boldsymbol{x})$, providing a new composed function $\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x}) = \mathbf{S}(\boldsymbol{c}(\boldsymbol{x}))$ which is then an approximation of the fine model $\boldsymbol{f}(\boldsymbol{x})$, at least locally. From there, the Space Mapping technique applied to outputs solves the modified and approximated problem:

$$\boldsymbol{x}_{SM}^{*} = \operatorname{argmin} \left\| \mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{y} \right\| \qquad (2)$$

This mode of operation is common to both Space Mapping methods used in the work presented in this paper. They differ only in the readjustment process between coarse and fine models, that is to say about the definition of the transformation $\mathbf{S}$.

B. Summary

Figure 1 below summarizes the overall functioning of Output Space Mapping methods.

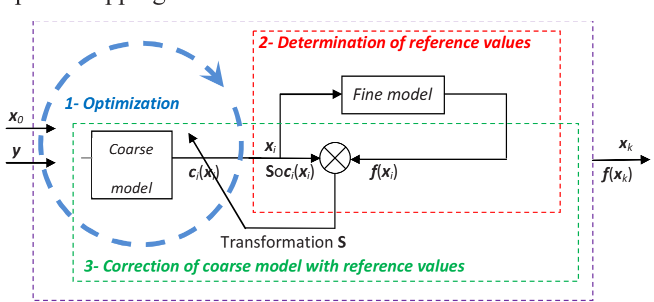

Figure 1. General principles of Output Space Mapping methods

The "coarse model" must be built in order to quickly compute the actuator performances (outputs). This model is used in an intensive way by a "traditional" optimization algorithm, based for example on the SQP ("Sequential Quadratic Programming") algorithm or on the Sequential Simplex algorithm [4]. Results given by this model do not have to be precise, but should remain representative of the general characteristics and physical behavior of the actuator.

The "fine model", giving reference results, is used to automatically correct, at regular intervals, the output values of the "traditional" optimizations carried out using the coarse model, introduced previously.

Hence, the "Space Mapping" approach corresponds to an automation of the optimization-correction procedures, which are generally "hand-made", by the judicious use of the resources: fast models are used for optimizations only, whereas fine models are used to readjust previously obtained optimization results. The additional effort initially required for the development of both models "is recovered" in a certain way, by the increase of research efficiency for optimal conditions.

C. Output Mapping method

The first algorithm retained in this study is called Output Mapping (OM) [5] [6]. Its operating mode is given in Figure 2.

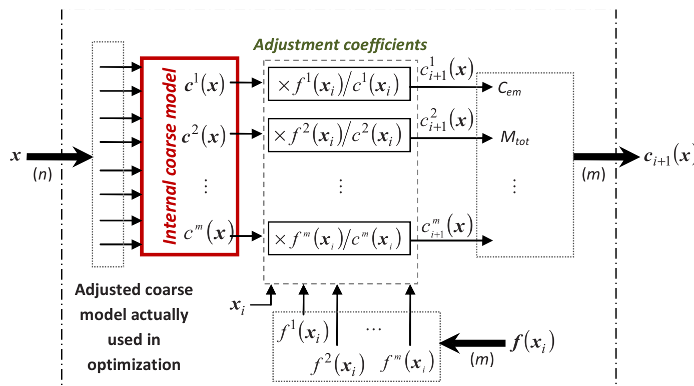

Figure 2. Adjustment technique in Output Mapping

The transformation used in the Output Mapping is very simple, as it is to correct the coarse model outputs by coefficients in order to find the corresponding output values given by the fine model. This procedure is detailed in the following.

At iteration i, from a vector of previous (or initial) input values $\boldsymbol{x}_{i-1}$, an optimization is achieved, during which only the coarse model is used (and not the fine one). The optimization result corresponds to vector $\boldsymbol{c}_i(\boldsymbol{x}_i)$ and is obtained for the corresponding optimal input vector $\boldsymbol{x}_i$. $\boldsymbol{c}_i(\boldsymbol{x}_i)$ is then compared to vector $\boldsymbol{f}(\boldsymbol{x}_i)$ given by the fine model, for the same optimal point $\boldsymbol{x}_i$ found previously. For the output j (among the m outputs defined), the lack of adjustment of the coarse model (which gives $c_i^j(\boldsymbol{x}_i)$) with respect to the fine model (giving $f^j(\boldsymbol{x}_i)$)) is defined by the ratio $\theta_i^j = f^j(\boldsymbol{x}_i) / c_i^j(\boldsymbol{x}_i)$. Hence, if one multiplies the output j of the coarse model $c_i^j(\boldsymbol{x}_i)$ by this ratio, then it will exactly coincide with the fine model for the point $\boldsymbol{x}_i$, and partially in the vicinity of this point. This corresponds to an adjustment of the outputs, that is to say an *output* space mapping. Therefore, this technique does not alter the writing of the coarse model $\boldsymbol{c}(\boldsymbol{x})$, but only modifies its m outputs (with different factors $\boldsymbol{\theta}_i$ at each iteration).

The ratio $f^j(\boldsymbol{x}_i)/c_i^j(\boldsymbol{x}_i)$ is obtained using results of iteration i, and allows to deduce the new values of the coarse model outputs, for iteration i+1. This one becomes $\boldsymbol{c}_{i+1}(\boldsymbol{x})$ and replaces $\boldsymbol{c}_i(\boldsymbol{x})$, through a simple linear "rule of three" :

$$c_{i+1}^j(\boldsymbol{x}) = \theta_i^j . c_i^j(\boldsymbol{x}) = \left(\frac{f^j(\boldsymbol{x}_i)}{c_i^j(\boldsymbol{x}_i)} \right) . c_i^j(\boldsymbol{x}) \tag{3}$$

This general procedure is repeated as long as it is possible to improve these values, with regard to the fixed objectives $\boldsymbol{y}$.

It should be noticed that no proof of convergence has been established for this algorithm.

D. Manifold Mapping method

The Manifold Mapping method (MM) is also a method of adjustment of the output values between coarse and fine models, that is to say a method of Output Space Mapping (OSM).

However, this new approach possesses, under certain circumstances proofs of convergence, which may make it more interesting from a mathematical point of view [7]. However, it is based on a more complex theoretical adjustment procedure, involving a more difficult algorithmic translation [8] [9].

Unlike the Output Mapping which makes the adjustment by a multiplication (i.e. a rule of three), the Manifold Mapping process performs two operations simultaneously, so the coarse model is:

- translated towards the fine model;
- rotated in order to better match (locally) the fine model.

From the mathematical point of view, this double transformation $\mathbf{S}$ applied to the model $\boldsymbol{c}(\boldsymbol{x})$, is:

$$\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_f^*) + O.(\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{c}(\boldsymbol{x}_f^*)) \tag{4}$$

With $\boldsymbol{x}_f^*$ the vector of input values for the best configuration found so far. Thus, if initially the O-term is omitted, this operator can be written $\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{c}(\boldsymbol{x}) + (\boldsymbol{f}(\boldsymbol{x}_f^*) - \boldsymbol{c}(\boldsymbol{x}_f^*))$. This corresponds to translating the output values of the coarse model $\boldsymbol{c}(\boldsymbol{x})$ by the vector describing the lack of fit $(\boldsymbol{f}(\boldsymbol{x}_f^*) - \boldsymbol{c}(\boldsymbol{x}_f^*))$ between the two models evaluated at the same best point $\boldsymbol{x}_f^*$.

Moreover, the Manifold Mapping method achieves the adjustment of the coarse model to the tangent plane of the fine model, at the point $\boldsymbol{x}_f^*$. This operation is realized thanks to jacobian matrices associated to fine and coarse functions, evaluated at the same point $\boldsymbol{x}_f^*$:

$$O = \boldsymbol{J}_f(\boldsymbol{x}_f^*).\boldsymbol{J}_c^+(\boldsymbol{x}_f^*) \tag{5}$$

With $\boldsymbol{J}_c^+$ the pseudo-inverse of the jacobian matrix. The normal inversion is not applicable because jacobian matrices are generally not square: they have m rows (the number of input variables) and n columns (the number of output variables). The operator O is a square matrix $m \times m$.

To sum up, $\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x})$ represents the formulation of an approximate model of the fine model defined by $\boldsymbol{f}(\boldsymbol{x})$. Its interest is the speed of its evaluation, as it is built from the fast (coarse) model. In this context, this fast function $\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x})$ can advantageously replace $\boldsymbol{f}(\boldsymbol{x})$.

The optimization problem therefore comes down to (2).

However, strictly speaking, the optimal conditions $\boldsymbol{x}_{SM}^*$ can match those $\boldsymbol{x}^*$ (for which $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{y}$) if and only if the transformation $\mathbf{S}$ is defined from the final optimum conditions $\boldsymbol{x}^*$, which are unknown a priori. So, the definition of the transformation $\mathbf{S}$ is not really defined. To address this problem, $\mathbf{S}$ is replaced by an approximation $\mathbf{S}_k$, updated at each iteration of the optimization algorithm:

$$\boldsymbol{x}_{k+1}^* = \operatorname{argmin} \left\| \mathbf{S}_k \circ \boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{y} \right\| \tag{7}$$

k is the iteration number of the OSM algorithm.

The expression of the transformation $\mathbf{S}_k$ is based on an approximation O_k of the operator O. For this, the jacobian matrices must themselves be approximated. For example, since the term $\boldsymbol{J}_{ij}$ is equal to df_i/dx_j, one can concede the classical approximation using a quotient of two differences: $\Delta f_i/\Delta x_j = (f_i(k+1) - f_i(k))/(x_j(k+1) - x_j(k))$.

Generalizing, one can write that $\boldsymbol{\Delta F}_k \approx \boldsymbol{J}_f(\boldsymbol{x}_f^*).\boldsymbol{\Delta X}_k$ and $\boldsymbol{\Delta C}_k \approx \boldsymbol{J}_c(\boldsymbol{x}_f^*).\boldsymbol{\Delta X}_k$. By inverting this last equation: $\boldsymbol{\Delta C}_k^+ \approx \boldsymbol{\Delta X}_k^{-1}.\boldsymbol{J}_c^+(\boldsymbol{x}_f^*)$. This allows to deduce an expression of O_k :

$$O_k \approx \boldsymbol{\Delta F}_k.\boldsymbol{\Delta C}_k^+ \left(= \boldsymbol{J}_f(\boldsymbol{x}_f^*).\boldsymbol{\Delta X}_k.\boldsymbol{\Delta X}_k^{-1}.\boldsymbol{J}_c^+(\boldsymbol{x}_f^*) = O \right) \tag{8}$$

At the first iteration of the algorithm, when k=1, the previous operator cannot be calculated, and one chooses the identity matrix $O_1 = \mathbf{I}_m$.

The algorithm described above has the disadvantage of requiring at each iteration a new calculation of the function $\mathbf{S} \circ \boldsymbol{c}(\boldsymbol{x})$. To avoid this, it may be more interesting to keep unchanged the definition of the coarse model $\boldsymbol{c}(\boldsymbol{x})$ and change the values of vector $\boldsymbol{y}$ at each iteration. Overall, this does not change the initial optimization problem, but only changes its expression:

$$\boldsymbol{x}_{k+1}^{*} = \operatorname{argmin}\left\| \boldsymbol{c}(\boldsymbol{x}) - \mathbf{T}_k \circ \boldsymbol{y} \right\| \tag{9}$$

Where $\mathbf{T}_k$ is a new transformation (approximated) applied to the constant vector $\boldsymbol{y}$, as:

$$\mathbf{T}_k \circ \boldsymbol{y} \approx \boldsymbol{c}\left(\boldsymbol{x}_f^{*}\right) - \left(\Delta \boldsymbol{C}_k . \Delta \boldsymbol{F}_k^{+}\right)\left(\boldsymbol{f}\left(\boldsymbol{x}_f^{*}\right) - \boldsymbol{y}\right) \tag{10}$$

At this level, small changes can be made at this writing, in order to improve the convergence behavior of the algorithm (by adding stabilization terms). Also, further developments can possibly include in the same kind of writing, constraints whose evaluation is very expensive [8] [10].

III. Optimization problem

A. Presentation

Space Mapping methods are applied here to optimize some performances and physical characteristics of an internal permanent magnet synchronous machine (IPMSM). This type of machine is commonly used nowadays, especially in embedded systems such as cars for example.

The structure considered here is a three-phase machine, with a small size (diameter and length of about 5 cm), 2 pairs of poles and an internal rotor.

The rotor position (i.e. its direct axis) is set to an angle of 0°. The stator power supply is fixed as well, and directed with an angle of 135 electrical degrees. This particular value corresponds to the most favorable torque angle available for this type of machine. For all three phases, a total of 250 A.t. is imposed in the stator windings. This magnetomotive force remains constant, knowing that the corresponding current density may evolve because of geometry changes resulting from the optimization of the dimensions of the actuator, including those of the slots. A filling factor equal to 0.6 is considered.

The same sheet FeV-800-50 is used for "iron" non-linear parts of the stator and rotor. Permanent magnets are NdFeB -type, linear, with $B_r = 1.13$ T.

The expected performances of this actuator are summarized in Table I. They concern the electromagnetic torque, the total mass and the mass of permanent magnets.

TABLE I. Output variables ans their target values

Descriptions	Names	Target values (y)
Torque	C_{em}	2.5 N.m
Global mass	M_{tot}	3 kg
Magnet mass	M_{mag}	250 g

Inputs of the optimization problem correspond to the probable variation sources of the output quantities. They are here solely geometric (Figure 3).

Table II gives their numerical specifications.

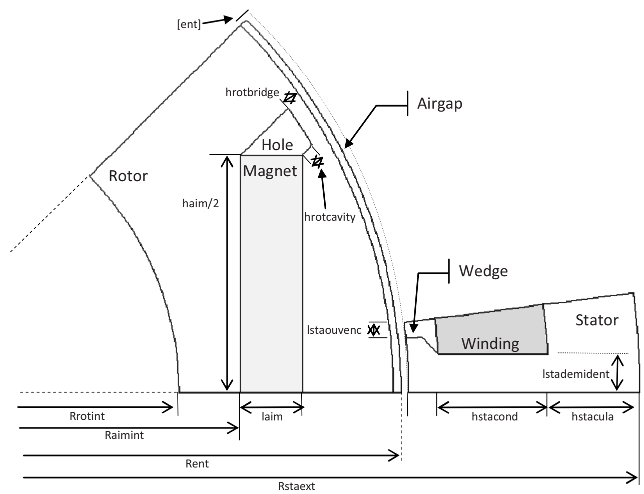

Figure 3. Parameterized geometry of IPMSM

TABLE II. Input variables (unit: mm)

Descriptions	Names	Lower bound	Upper bound
Exterior stator radius	R_{staext}	14	35
Air gap radius	R_{ent}	4.2	24.5
Interior rotor radius	R_{rotint}	5	10
Inner magnet radial position	R_{aimint}	10	14
Magnet width	l_{aim}	1	3
Magnet height	h_{aim}	10	20
Magnetic bridge height	$h_{rotbridge}$	0.2	0.7
Cavity height	$h_{rotcavity}$	0.1	1.4
Stator yoke height	$h_{stacula}$	1	5
Stator slot height	$h_{stacond}$	3	5
Half of tooth width	$l_{stademident}$	1	2
Half of slot opening height	$l_{stademiouvenc}$	0.2	0.7

B. Models

The operating procedures of "Space Mapping" optimization algorithms being well defined, the nature of coarse and fine models remains to be specified.

The choice of model types must conform to the requirements presented in the previous paragraphs; to summarize, one needs:

- a (very) fast "coarse" model, representative of the operating behavior of the actuator;
- a "fine" model giving reference results.

Typically, in the context of design and optimization of actuators by computer means, and within the framework of magnetic modeling, the fast model corresponds generally to a Magnetic Equivalent Circuit (MEC), more seldom to explicit and purely analytical relations, whereas the reference model consists of a finite element analysis (FEA).

This approach suffers from the major drawback to require the construction of two separate models of two different types, which must be evaluated thanks to two different dedicated programs. Thus, in addition to the "doubling" of the adjustment time of the models, this solution demands a greater diversity of resources.

For these reasons, it was decided to use the same modeling approach for the construction and the evaluation of both (coarse and fine) models. In this context, these two models correspond to finite element modelings.

At first, the fine model is built. Since its output values should be accurate and reliable, it is defined using a fine grid (made with a large number of elements) and high precision elements (quadratic), as well as taking into account the non-linearity of ferromagnetic materials.

The coarse model is then deducted from the fine one, by making successive simplifications of its characteristics. Several options are available to the modeler in order to implement such simplifying changes on the initial reference model; for instance, he may consider:

- a more or less fine consideration of the actuator geometry (simplified curves, etc.);
- the use of more or less precise techniques for the consideration of special boundary conditions (typically, in the case of open spaces for the consideration of infinity);
- the precision level requested, for the solution computed by finite element solver;
- the definition of material properties;
- the definition of the mesh.

As part of this work, the applied simplifications concern the last three points, that is to say the accuracy of the finite element solution, the definition of the mesh and the mathematical description of materials.

The most important difference between these two models concerns the characteristics of the material FeV-800-50. It describes the magnetic saturation of the ferromagnetic sheets that occurs around 1.5 T. The non-linearity represented by this curve is the chief difficulty for the solving of the finite element problem. In this case, the fine modeling uses the Newton-Raphson algorithm to approximate the solution iteratively. In the coarse model, to reduce as much as possible its evaluation time, it was decided to approach the non-linear magnetic characteristic by a linear evolution. Thus, the problem solving, initially non-linear, becomes linear and can be achieved directly (without iteration) and accurately.

This solution gives priority to the evaluation speed of the coarse model, with regard to the accuracy of the results (theoretically given by the fine model). The permeability of the material involved in the fast modeling is set at 150, about 1/20 of the permeability of the linear part of the initial magnetic saturation curve (considered by the fine model). Since this coarse model is used for optimization, this particular value can to some extent limit naturally the magnetic flux density in ferromagnetic parts, or equivalently increase the flow sections of the magnetic flux flowing in the machine. In addition, this value is not zero; therefore, permanent magnets are not short-circuited magnetically.

To complement these provisions limiting the magnetic flux density, a constraint is added to the definition of the problem. It consists in imposing the limit value of 2.5 T in the whole finite element model:

$$B_{\max} \leq 2.5\,\mathrm{T} \tag{11}$$

In parallel, a second constraint is defined so as to limit indirectly the thermal heating in the machine. For this, the current density in conductors is limited:

$$J_{\max} \leq 6\,\mathrm{A/mm^2} \tag{12}$$

Also, the definition of the mesh allows tuning the precision of the finite element model, and acts in the same time on the evaluation speed of this model. Indeed, changing the mesh modifies the number of nodes (and elements) and thus modifies the mathematical size of the problem.

The (per length and/or surface unit) node density of the mesh must be parameterized. Thus, a simple change of these parameter values makes it possible to modify the original fine mesh and obtain a "simplified" and faster model. Both models can therefore be defined from the same original data file.

In a general way, the simplification of the FE fine model is achieved in such a way that the representativeness of the model is preserved with respect to the torque. For this, the mesh is "lightened" everywhere, except inside the air gap (main location of magnetic energy).

Table III gives a comparison of the meshes of fine and coarse models, as well as their resolution times. The fine model, taking into account ferromagnetic material non-linearity, is 7 times longer to be solved than the coarse (fast) model. This difference is not in itself very important, but it may justify the use of Space Mapping methods to reduce significantly the time required for optimization.

TABLE III. COMPARISON OF THE TWO FE MODELS

Characteristics	Fine model	Coarse model	Ratio
Number of nodes	11613	978	1/12
Number of elements	5706	1748	1/3.2
Type of elements	T6 *(Triangular – 6 nodes - Quadratic)*	T3 *(Triangular – 3 nodes - Linear)*	-
FE solving time	0.22 s	0.031 s	1/7

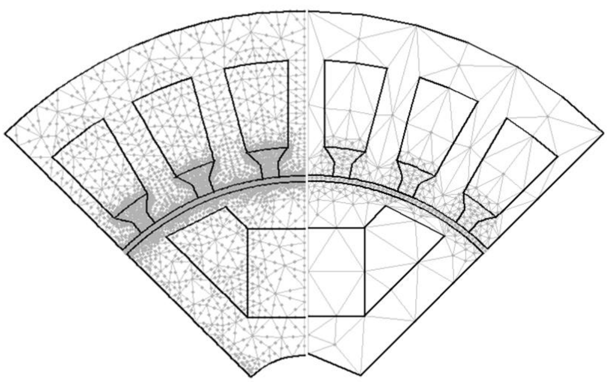

Figure 4. Meshes - Fine model *(left)* - Coarse model *(right)*

C. Management of multi-objective configuration

The design of the synchronous machine takes into account three outputs, which should, if possible, approach the respective values set a priori. This is the electromagnetic torque C_{em}, the total mass M_{tot} and the mass of permanent magnets M_{mag}. The corresponding optimization problem is therefore a multi- objective problem.

In the framework of this work, this problematic is not solved through the construction of the Pareto's set (which is too expensive to build), or by using an aggregate function describing the distance between the present state of the optimization algorithm and the target conditions ($\boldsymbol{y}$) [11].

The method only considers the electromagnetic torque C_{em} (which must be equal to about 2.5 N.m.) and transforms the two other output quantities M_{tot} and M_{mag} into constraints.

The optimization problem is then rewritten as follows:

$$\begin{cases} \boldsymbol{x}^* = \text{argmin}\left\| C_{em}(\boldsymbol{x}) - 2.5 \right\| / \delta\boldsymbol{y} \\ (M_{tot} - 3)/3 \le \tau_{cont} \\ (M_{mag} - 0.25)/0.25 \le \tau_{cont} \end{cases} \tag{13}$$

Where τ_{cont} is the tolerance on the respect of constraints and $\delta\boldsymbol{y}$ the tolerance values of the entities to optimize (i.e. to minimize). In the present case, $\delta\boldsymbol{y}$ is linked to C_{em}, and is taken as $\delta\boldsymbol{y} = 0.1$.

It may be noted that the Manifold Mapping algorithm can naturally take into account several outputs (that is, when m> 1). This option is not used here because its application with the Output Mapping method is tricky and can make convergence unstable in this case.

Strictly speaking, it is clear that this approach does not allow to find the “best” actuator (according to a criterion) but only a configuration of a machine whose characteristics are the closest to the ones predefined by (typically industrial) specifications (Table I). In the case of unreachable objectives, this problem becomes a real optimization problem (solved by the “distance-to-target” method), knowing that, before any calculation, only experience can tell whether objective values are realistic or not.

IV. OPTIMIZATION RESULTS

The whole optimization process is implemented under Matlab® by the author: this is a program that executes automatically OM and MM optimization algorithms and manages the building of the (unique) FE model according to characteristic dimensions (creation of meshes, solving of FE models and calculation of output values like torque, masses, volumes, etc.). The uniqueness of the scientific and numerical platform is advantageous, among others, in terms of data exchange, making them simpler, faster and safer.

Both optimizations start with the same initial point, which corresponds to the central values of the variation intervals of input variables, defined in Table I. The sequential simplex method is used for coarse model optimizations [4]. A limit of 350 iterations is imposed to these optimizations.

A. Output Mapping

With the OM algorithm, 5 evaluations of the fine model and 1748 evaluations of the coarse model are necessary to obtain the results. This optimization requires 10min 59s.

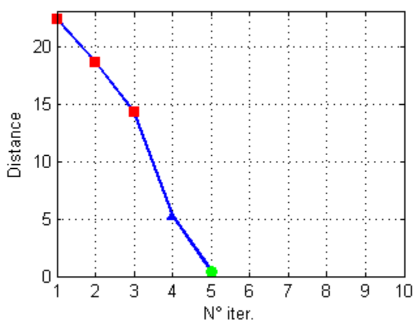

Figure 5. Convergence evolution (OM method)

The algorithm stops because the final conditions (at iteration 5 – green dot) correspond to the specifications defined by vector ***y***, through the consideration of tolerances on output values.

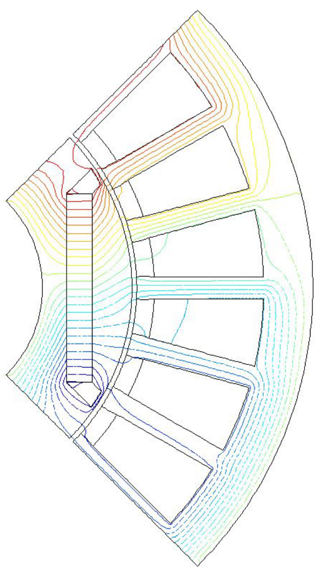

Figure 6. Optimized geometry (OM method)

B. *Manifold Mapping*

The Manifold Mapping method uses 6 evaluations of the fine model and 3237 evaluations of the coarse model. This optimization lasts 11min 39s.

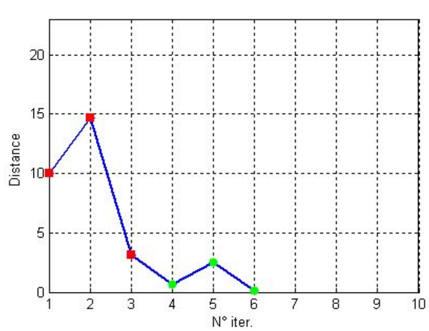

Figure 7. Convergence evolution (MM method)

In Figure 7 which gives the convergence history, green dots denote configurations for which good solutions of the design problem are found, that is to say configurations for which the specifications are met or even improved.

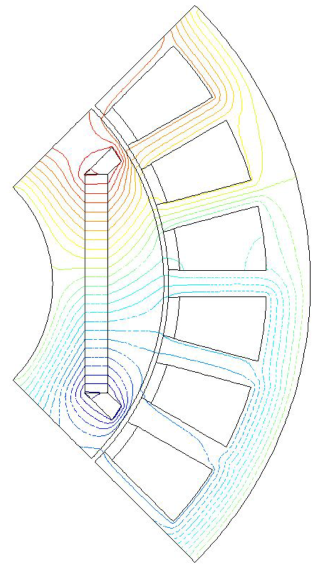

Figure 8. Optimized geometry (MM method)

C. *Comparison*

The following tables gather numerical data on the sizing of the synchronous machine, by the use of both Output Space Mapping methods.

TABLE IV. OPTIMIZATION RESULTS – OUTPUT VALUES COMPARISON BETWEEN OM AND MM APPROACHES

Descriptions	Names	OM	MM	Units
Torque	C_{em}	2.51	2.49	N.m
Global mass	M_{tot}	3.35	3.30	kg
Magnet mass	M_{mag}	0.142	0.149	kg

TABLE V. OPTIMIZATION RESULTS – CONSTRAINT VALUES COMPARISON BETWEEN OM AND MM APPROACHES

Descriptions	Names	OM	MM	Units
Max. field density	B_{max}	1.09	1.03	T
Max. current density	J_{max}	4.38	5.52	A/mm²

TABLE VI. OPTIMIZATION RESULTS – INPUT VALUES – (UNIT: MM) COMPARISON BETWEEN OM AND MM APPROACHES

Descriptions	Names	OM	MM
Exterior stator radius	R_{staext}	24.72	24.87
Air gap radius	R_{ent}	13.52	15.77
Interior rotor radius	R_{rotint}	7.74	8.59
Inner magnet radial position	R_{aimint}	9.27	10.64
Magnet width	l_{aim}	1.59	1.45
Magnet height	h_{aim}	11.87	13.79

Descriptions	Names	OM	MM
Magnetic bridge height	$h_{rotbridge}$	0.18	0.53
Cavity height	$h_{rotcavity}$	0.29	0.74
Stator yoke height	$h_{stacula}$	3.09	2.77
Stator slot height	$h_{stacond}$	6.86	5.47
Half of tooth width	$l_{stademident}$	0.68	0.84
Half of slot opening height	$l_{stademiouvenc}$	1.08	1.21

Optimal solutions found by the two optimization techniques are not strictly equivalent: they virtually do not describe exactly the same machine, even if their outer radii R_{staext} are very close.

The convergence history of these two algorithms is different. The Output Mapping was able to approach directly the desired configuration, never away from it. However, the Manifold Mapping approach did not converge monotonously, and not as fast as in the case of the OM method.

Conclusions provided by this comparison between the two methods, can be found in many other configurations. It gives the ability to describe the following general behavior:

- OM cannot guarantee convergence towards target values. In any case, there will be a rapprochement to these data, but sometimes oscillations may appear near them, without showing a real convergence.
- MM guarantees convergence (not necessarily monotonous), but may be slower than the approach by OM.

V. General remarks

It is clear that the main interest of Space Mapping methods holds in the will or the need to save as much as possible the number of fine model evaluations. For example, such a model may be a three-dimensional finite element model taking into account time or movement. In this case, each evaluation of this reference model may last several hours. It becomes obvious, even necessary to resort to a second model (much) faster, to be able to perform optimizations.

This *necessity* is not obvious here because the average evaluation time of the fine model is actually quite low. However, this article has attempted to present the specific Space Mapping methods as tools for a new efficient approach for the optimization of electromechanical systems.

Finally, it should be mentioned that the use of these two optimization techniques has been achieved here with the underlying objective to compare their mutual performances, rather than maximizing the efficiency of researches for optimal conditions. In this context, the definition of the models (that is to say that of the input variables, the output quantities and constraints) was made identical between the two performed optimizations. Thus, it is certainly possible to improve the effectiveness of these methods (for the optimization of the synchronous machine) by exploiting their intrinsic characteristics, and making the appropriate and corresponding changes in model definitions.

Conclusion

This article describes the techniques used for the design of an electromagnetic actuator. The central theme of this work focuses on the increase of the efficiency of the overall process optimization. Thus, novel solutions have been presented to reduce the development time of models (by using a single FE model with adjustable speed and precision), to reduce the evaluation time of the coarse model (using linear magnetic materials) and about the research for optimal conditions (by applying the Output Space Mapping techniques). This approach does not sacrifice the speed at the expense of the “quality” of the results, since it uses regular recalibrations thanks to the reference model.

Applied to the current design problem, the Output Mapping, although very simple (to understand and implement), clearly gives good results. The Manifold Mapping, using a more complicated adjustment technique, also gives interesting results. However, it does not outperform the approach by Output Mapping.

In both cases, these two optimization approaches offer new interesting perspectives for the efficient optimization and design of non-linear electromagnetic systems.

References

[1] Vivier S., Lemoine D., Friedrich G., "Optimal design a high dynamic actuator for diaphragm pumps", IEMDC 2009

[2] Sondergaard, “Optimisation using surrogate models – by Space Mapping technique”, PhD thesis, 2003

[3] Encica L., Echeverria D., Lomonova E.A., Vanderput A.J.A., Hemker P.W., Lahaye D., "Efficient optimal design of electromagnetic actuators using Space Mapping", Struct. Multidisc. Optim., 2006

[4] Ryan, P.B., Barr R.L., Todd H.P., “Simplex techniques for non linear optimization”, Analytical Chemistry, Vol 52, pp. 1460-1467, 1980

[5] Cheng, “Advances in implicit and output space mapping technology”, PhD thesis, 2004

[6] Tran T.V., Brisset S., Echeverria D., Lahaye D., Brochet P., "Space mapping techniques applied to the optimization of a safety isolating transformer", ISEF 2007

[7] D. Echeverria, P.W. Hewker, “Manifold mapping: A two-level optimization technique”, Computing and Visualization in Science, Vol. 11, N°4-6, pp. 193-206, 2008

[8] Echeverria D., "Two new variants of the manifold-mapping technique", COMPEL, Vol. 26, N°2, pp. 334-344, 2007

[9] D. Echeverria, D. Lahaye, L. Encica, E.A. Lomonova, A.J.A. Vandenput, “Manifold-mapping optimization applied to linear actuator design”, IEEE Transactions on Magnetics., Vol. 42, N°4, pp.1183-1186, 2006

[10] P.W. Hewker, D.Echeverria, “A trust-region stategy for manifold mapping optimization”, Journal of Computational Physics, 224, pp. 464-475, 2007

[11] Vivier S., Lemoine D., Friedrich G., "Improvements for multi objective optimizations using Output Space Mapping technique", OIPE 2010