



**HAL**  
open science

# Graphical predetermination of optimal machine designs by iso-performance configuration modeling

Stephane Vivier

► **To cite this version:**

Stephane Vivier. Graphical predetermination of optimal machine designs by iso-performance configuration modeling. *Mathematics and Computers in Simulation*, 2021, 184, pp.165-183. 10.1016/j.matcom.2020.02.025 . hal-04288636

**HAL Id: hal-04288636**

**<https://hal.science/hal-04288636>**

Submitted on 22 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Graphical predetermination of optimal machine designs by iso-performance configuration modeling

Stephane VIVIER

Sorbonne Université – Université de Technologie de Compiègne – Laboratoire UTC - CNRS FRE2012 Roberval

Centre de Recherche de Royallieu – CS 60 319

60203 Compiègne Cedex, FRANCE

[stephane.vivier@utc.fr](mailto:stephane.vivier@utc.fr)

---

**Abstract** – When using "conventional" approaches for the optimal sizing of electrical machines, the application of optimization algorithms makes it possible to search for the values of their input characteristics (dimensions, power supply, materials, etc.) giving the desired performances (torque, efficiency, mass, etc.), provided that a certain number of constraints are satisfied.

*This (one-way) approach has several weaknesses, the main one being that it is not clearly apparent whether this sizing problem has at least one solution. If sets of solutions exist, it becomes interesting to determine the characteristics of all these valid machines. On the other hand, if no machine can be defined by exactly meeting the specifications, it becomes more useful to look for the constraints to be relaxed to reveal solutions, or to determine and then choose among all the quasi-optimal configurations those that are the most interesting.*

*In this context, this work aims to respond to these issues by introducing a methodological procedure giving assistance in the optimal sizing of electrical machines in particular, and in optimization in general.*

*In this procedure, a number of numerical processing tools are implemented, to offer the opportunity to understand more finely, typically through 2D graphical representations, the complexity of optimization design problems. In other words, this work is an attempt to find an intermediate configuration between the pure optimization approach and the one based solely on modeling, in the sense that these models are first represented graphically in order to understand the complexity of the corresponding optimization problem.*

*The application of this approach is illustrated by the presentation of the main steps of the optimal design of a Permanent Magnet Synchronous Machine.*

**Keywords** – Optimization, Isovalue surfaces, Design of Experiments, Electrical machines, Graphical representations

## 1 Introduction

### 1.1. General background

The optimal design of electromechanical actuators has been a long-standing research discipline. Indeed, since the knowledge of the fundamental principles of electromagnetism, engineers have naturally sought to develop machines that operate according to these principles and offer the best performance. For example, for a rotating machine, it is natural to look for an actuator design such that the torque is the largest, mass and volume are the smallest (for integration and cost reasons) and which generates the lowest losses (for energy consumption reasons, and in a related way for thermal reasons).

Given the development of computer tools, the use of numerical simulations has become logical, as they typically make it possible to test new machines (virtual prototypes) with new structures and dimensions and/or using new materials.

Optimal design is based on the dual aspect of modeling and optimization. In the first place, it is necessary to establish a reliable model of the machine, that is to say a mathematical object, based on physical considerations, and evaluating the outputs or performances of the machine (like the torque, its internal temperatures, ...) as a function of input values (such as geometrical dimensions, values of electrical supply quantities, material characteristics, ...). From there, in a second step, this modeling is used by an optimization algorithm to find the right input values giving the best outputs (performance).

A large number of scientific contributions deal with the development of techniques aiming at finding optimal conditions for all types of optimization problems (linear/non-linear, uni/multimodal, mono/multiobjective, cheap or expensive models, etc.) [1].

### 1.2. Specific issues in optimal sizing problems

Given the great complexity of the models used (these models themselves being composed of sub-models, often non-linear) and given their often numerical calculation mode (by Finite Element Analyses for example), the model of machine becomes

a black box for the designer. The latter can no longer directly and therefore simply understand the links between inputs and outputs (for example, how to modify inputs to minimize a given output?). Thus, simply modeling is generally not enough for the designer: additional tools for understanding these models become necessary. This issue is often solved by carrying out sensitivity analyses, often based on the preliminary realization of experimental designs [2]. These tools provide results that are simple to analyze (effects of factors and interactions between factors) but which are necessarily global (i.e. over the entire domain of study) and do not describe local behaviors. To remedy this, one solution is to build and use a surrogate model, replacing the initial model, using the same inputs and calculating the same outputs, but by assumption much faster to evaluate ([the construction of such surrogate models is a problem in itself, which will be discussed below](#)). Thanks to this very lightweight alternative model, it is then possible to access less coarse and more local information. Finally, yet importantly, these surrogate models can be used to achieve efficient and comprehensive optimizations [3]. This indirect approach has been favored in this work.

On the other hand, it is possible to achieve optimizations by directly using the initial modeling. To limit costs, it is preferable to use cost-effective optimization algorithms, typically such as the *Sequential Quadratic Programming (SQP)* method. However, at the end of such approaches, the designer only has one set of input values for which performances are *a fortiori* improved. However, the optimization algorithm alone does not provide information on variations of outputs around the optimal point, nor does it say whether there are other optimal configurations (if any).

To summarize, having a machine modeling does not allow to understand how to modify its characteristics to improve its performances, which is the goal of optimization. And applying an optimization algorithm to this modeling does not give access to an understandable and accessible description of the variations of outputs (performance) as a function of inputs (characteristics), which is the role of the modeling itself.

In addition, from an engineer's point of view, running an optimization procedure is usually not enough. Thus, it may be useful to know, even qualitatively at first, whether a problem of machine design with constraints presents solutions or not. If there is no solution, it becomes interesting to identify the constraints to relax in order to reach a solution. If several design solutions exist, the useful information concerns their location and distribution in the domain (for example, sometimes, for the same specification, quite different topologies of machines may be appropriate: thus, solutions are defined by disjointed sets in the (design) space).

It is also conceivable that one can meet the situation in which specifications evolve. This can happen for example when one has to size a range of actuators, involving distinct specifications for each machine in this range. In this situation, an interesting sizing approach would be to evaluate the influence of the change of specifications on the definition of optimal solutions; this knowledge should result from a minimum of additional calculations.

Another important remark concerns the placement of optimal solutions in the domain of study. Indeed, if an optimization problem under constraints is well posed, then optimal solutions are necessarily located at least on a constraint and / or on a bound. This observation implies that it is necessary to model the constraints defined by the specifications relatively accurately. And in most cases, a constraint corresponds to an isovalue set (for instance designing a machine with a maximum mass of 10 kg requires knowing all the (input) configurations defining the machines of this mass: this set is the isovalue surface whose all points satisfy  $M_{tot}=10$  kg).

In addition, knowing the location of the points satisfying the constraints in the domain can make it possible to illustrate and thus to better understand the specificities of the optimization problem.

Finally, the usual approaches to solving optimization problems very often lead to the distinction between the objective function (the quantity to be minimized) and the constraint functions. This data partition is most of the time performed a priori, i.e. without knowing the variations of outputs according to inputs. This choice by the optimizer can influence the efficiency of the optimization methods applied. In general, it would be more accurate not to make this distinction (for example, to consider all outputs as constraints).

### 1.3. Assumptions and solution adopted

By hypothesis, this study considers optimal sizing problems (typically of electrical machines) for which the number of inputs can be quite high (between 2 and 20), the outputs numerous (between 2 and 10) and that are based on non-linear and implicit models.

These initial models, known as reference models, are potentially complex because they support the description of several significant physical phenomena, such as electrical, magnetic, thermal and mechanical aspects, as well as the couplings between these different physics. Typically, magnetic modeling, or even thermal modeling, is numerical, i.e. performed by the Finite Element method.

Values of input variables are defined in known limits (bounds). Sizing is always achieved in order to satisfy the content of pre-established specifications, in which the minimum performance to reach are specified (for example, design a machine developing at least 140 N.m and weighing less than 10 kg...).

The approach proposed by this work consists in dealing with these optimization problems in a graphical way, using 2D representations of isovalue sets. These sets, or (hyper-)isosurfaces, are built from fast surrogate models of the reference model of the machine to be optimized.

It is important to note that this procedure explicitly creates representations of the variations of the outputs as a function of the inputs, which is not done by the Pareto front which simply links output values together [4]. In general, Pareto sets are of little interest in solving an optimization problem, except to reveal the compromises from which the designer must choose. Some references try to match the space of the optimal outputs (in the Pareto sense) with that of the corresponding inputs, through visualization tools [5].

To sum up, this work tries to implement a relatively formatted procedure to address some weaknesses appearing in traditional approaches for the design of machines. This new approach aims to introduce some graphical tools in order to understand more thoroughly all kinds of configurations involving input and output variables as well as constraints defined in these optimization problems.

## 2 Description of the methodology

### 2.1. Hypotheses

Let us consider a dimensioning problem (typically of a machine), treated by an optimization approach. This problem counts  $k$  input variables (factors)  $x_i$  and  $r$  output variables (responses)  $y_j$ . Each factor  $x_i$  is normalized so that its values reside in the interval  $[0 \ 1]$ . It is further assumed that it is always possible to remove the constraints linking the factors to each other, thanks to substitutions of variables.

This last point is important because it allows the search of optimal solutions in a unit  $k$ -dimensional hypercube domain, which is called “domain of study”.

The machine to be designed is modeled by a reference model (typically thanks to Finite Element Analyses) which makes it possible to calculate the  $r$  outputs from any combination of the  $k$  factor values.

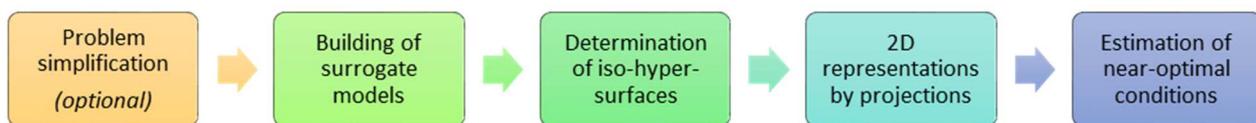
The objectives of this design problem are specified a priori, in specifications. Hence, each output  $y_j$  is associated with a value  $y_{refj}$  to reach.

Insofar as, by convention, any optimization problem seeks to find the minimum values of the objectives, one must eventually check the relation  $y_j \leq y_{refj}$  for all  $j$  in  $[1 \ r]$  at the end of the optimization procedure. For example, one can try to minimize the total mass of the actuator ( $M_{tot} \leq M_{totref}$ ) and/or to maximize its electromagnetic torque ( $-C_{em} \leq -C_{emref}$ : in this case  $y = -C_{em}$  and  $y_{ref} = -C_{emref}$ ).

Therefore, this problem consists in looking for the values of the factors that make it possible to satisfy all these objectives at once (if they exist).

### 2.2. Overview

The process put in place is presented in detail in the following sub-paragraphs. Its main steps are given by the flowchart in Fig. 1.



**Fig. 1** Main steps in the procedure

A first step consists in modeling the variations in performance (outputs) according to the characteristics (inputs) of the machine. By assumption, the reference modeling is accurate and therefore probably expensive to compute. To achieve good precision, this reference modeling must take into account all significant physical phenomena and consider only a minimum of simplifying hypotheses. To do this, it must be based on powerful numerical calculation tools, such as Finite Element models (typically for magnetic and thermal aspects).

In order to gain in efficiency, there is an interest here in replacing this fine modeling by another one that is much faster to evaluate. Thus, the aim becomes the construction of a mathematical model that is quick to calculate and can replace the reference model. By choice, this substitute model is deduced from the results of  $N$  previous uses of the reference modeling. In order to limit the construction cost of this surrogate model, while ensuring its quality, these  $N$  experiments (i.e. calls to the reference modeling) are defined using the design of experiments (DoE) methodology.

These “substitute” functions can then be used to determine iso-value surfaces (or isosurfaces), i.e. approximations of all the values of the inputs for which each output remains constant.

The use of isosurfaces is interesting because they can be easily represented graphically, particularly thanks to projections on 2D subspaces.

Finally, the superposition of these projections allows to graphically deduce intersections corresponding to sets of configurations for which the constraints and optimization objectives are satisfied or near. If certain constraints are violated, this "visual" approach naturally provides information on how to relax these constraints and eventually lead to almost optimal solutions.

### 2.3. Step 1: Calculation of the basis DoE

In order to build the surrogate model for the reference model, this fine modeling must first be evaluated at different points in the domain of study, i.e. for multiple combinations of values of the input variables.

This amounts to exploring this domain of study as widely as possible. The cost of this exploration, that is to say the number  $N$  of points, must remain limited.

For that, a design of experiments (DoE) with imposed budget is calculated. Space Filling (SF) designs are perfect tools for this purpose [6]. Indeed, they make it possible to define the positioning of a predefined number  $N$  of experiment points in the domain of study, according to mathematical criteria. One approach consists in spreading the design points out as far from each other as possible (while staying inside the experimental boundaries). Another approach is to space the points out evenly over the region of interest.

It is clear that the choice of the number  $N$  of experiments and the type of experimental design can have an influence on future results. If  $N$  is too small, the surrogate models will necessarily be of poor quality, regardless of the interpolation method used. Also, the position of the experiment points distributed in the domain may play a role. However, these choices are made a priori, i.e. without knowing the variations in the output variables of the reference model.

There is therefore a compromise to be found concerning the number of experiment points: enough to ensure that the substitute model is of good quality and not too many so that it does not cost too much. Of course, it is often the cost criterion that can indicate the authorized budget, i.e. the number of fine model evaluations that can be carried out. It should be noted that these calculations can be carried out in parallel because DoEs define experiments that are independent of each other.

So, for each experiment point, the  $r$  output variables  $y_j$  are calculated by evaluating the reference model of the machine. The simple examination of the  $N \times r$  calculated values makes it possible to deduce the lower and upper bounds for each output  $y_j$ :  $[\min(y_j) \max(y_j)]$ .

### 2.4. Step 2: Definition of $r$ interpolating models

For each output variable  $y_j$ , an interpolation model  $M_j$  can be defined from the  $N$  experiments of the DoE and the  $N$  corresponding values of  $y_j$  calculated by the reference model.

Among all the interpolation construction techniques, models based on RBF (Radial Basis Functions) were chosen [7]. Interpolation (or approximation) by RBF can be seen as a particular application of Neural Networks [8].

In addition to the simplicity of their construction, the RBF models maintain a very good numerical prediction behavior when the number of dimensions ( $k$ ) becomes large (typically greater than 10).

An RBF interpolation function is actually the weighted sum of a finite number  $n$  of identical radial functions:

$$y(\mathbf{x}) = \mu(\mathbf{x}) + \sum_{i=1}^n \lambda_i \cdot \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (1)$$

Where  $\mu(\mathbf{x})$  is a polynomial function or a simple constant,  $\lambda_i$  is the coefficient associated with the RBF function  $\phi_i$ . It should be remembered that the value of an RBF function  $\phi_i(d)$  at a point  $\mathbf{x}$  is calculated only as a function of the distance (or radius  $d$ ) from a base point  $\mathbf{x}_i$ . This gives us  $d = \|\mathbf{x} - \mathbf{x}_i\|$ . There are as many RBF functions ( $n$ ) as there are basis points. In our case, the basis points correspond to the points of the design of experiments and therefore  $n=N$ .

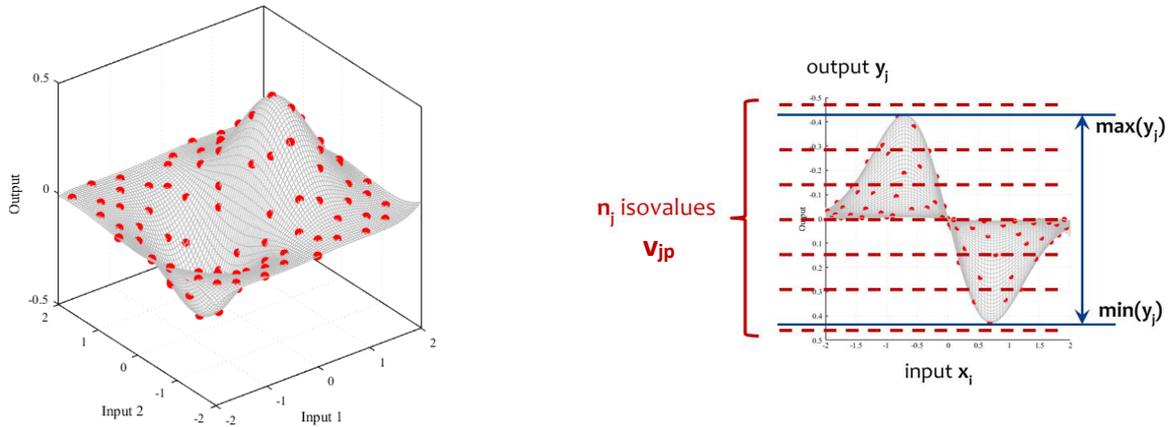
The values of the coefficients  $\lambda_i$  as well as the definition of the function  $\mu(\mathbf{x})$  can be easily deduced from the resolution of a linear system of equations constructed from the coordinates  $\mathbf{x}$  of the experimental points and the values  $y$  of the outputs of the fine model for each of these points.

There are several possible choices for the definition of RBF kernel functions  $\phi$  (linear, cubic, Gaussian, etc.) as well as for the type of norm to be applied for the evaluation of  $d$ . These choices modify the interpolation functions, particularly in the parts of the domain of study where the density of experiment points is lowest. These choices are often made through practice.

### 2.5. Step 3: Determination of isovalue definitions

The intention is now to identify, as far as possible, those sub-spaces of the domain of study in which the objectives are better than their respective reference values. The optimal conditions sought necessarily belong to the intersection of these sub-spaces. Therefore, it is necessary to determine them as accurately as possible; this mainly involves special work on their boundaries, which are iso-(hyper)-surfaces. This paragraph and the next one deal with these operations.

For the output  $y_j$ ,  $n_j$  of values in the interval  $[\min(y_j), \max(y_j)]$  are chosen:  $\{v_{j1}, v_{j2}, \dots, v_{jn_j}\}$ . Then, for each value  $v_{jp}$  (with index  $p$  in  $[1, n_j]$ ), and thanks to the model  $M_j$ , one must identify new points of coordinates  $\mathbf{x} = [x_1, x_2, \dots, x_k]$  such that  $|M_j(\mathbf{x}) - v_{jp}| \leq \varepsilon$ . In other words, new points are being sought on the iso-(hyper)-surface of value  $v_{jp}$ , with the tolerance  $\varepsilon$ . Of course, the values  $v_{jp}$  of each output  $y_j$  can be deduced, at least in part, from the specifications (from the values  $y_{refj}$ ). This step is performed for each of the  $r$  models  $M_j$ . Figure 2 illustrates this step.



**Fig. 2** Left: Example of 2D surrogate RBF model built from a Space Filling DoE  
Right: Principle of determination of levels of isosurfaces from extremum values of output variables

### 2.6. Step 4: Search for iso-performance configurations

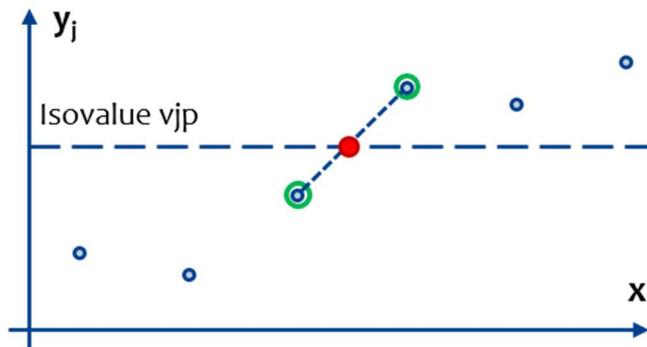
For each isovalue  $v_{jp}$ , one must find the configurations of input variables defining a machine whose output  $y_j$  is equal to  $v_{jp}$ . This set of configurations corresponds to a cloud of (experiment) points in the domain of study.

Many methods can be conceived to find the coordinates of points placed on isosurfaces. The two most effective ones are described in the following.

#### 2.6.1. Point-to-point interpolations

From the set of the  $N$  calculated points of the DoE, for an output  $y_j$  and a given isovalue  $v_{jp}$ , the nearest points of the corresponding isosurface are first isolated. Thus, a point in the DoE for which  $y_j < v_{jp}$  is retained only if its nearest neighbor in this same DoE is such that  $y_j > v_{jp}$ , and vice versa. Therefore, in this subset, all the points are coupled with another point placed on the other side of the isosurface to be constructed.

From there, for each of these pairs of points, and under the assumption that the values of the output  $y_j$  vary locally in a linear way, it is easy to deduce the coordinates of the intermediate point for which  $y_j \approx v_{jp}$ . As clearly shown in Fig. 3, this approach allows such points to be found within the domain, but will fail to define new points near its boundaries.



**Fig. 3** Principle of point-to-point interpolation

### 2.6.2. Pre-modeling and optimizations

As shown in Fig. 4, for a given output  $y_j$ , the associated model  $M_j$  can be directly used to find the points  $\mathbf{x}$  such that  $|M_j(\mathbf{x}) - v_{jp}| \leq \varepsilon$ , where  $\varepsilon$  is a very small tolerance. This constitutes therefore a (new) optimization problem, aiming here at finding as many points as possible belonging to an isovalue set, by exploiting the benefit of the speed of the substitute model.

This optimization problem can be solved by any classical algorithms (Sequential Simplex, SQP ...). However, to increase the chances of finding different solution points, this optimization sub-problem can be started from a multitude of initial points distributed in the domain; for example, the experiments of the previous DoE can be used as  $N$  initial points.

It is clear that this approach derives its validity from the quality of prediction of the interpolation model  $M_j$ . As a result, if the quality of the RBF model is not sufficient, this approach may introduce misinformation that may hinder understanding of the actual variations of the studied outputs. In brief, this second approach allows more new points to be created, but with a lower probability that they will be placed precisely on the isosurface to be built.

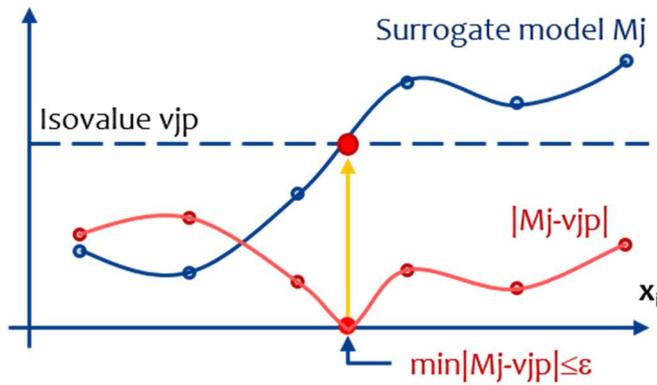


Fig. 4 Use of interpolation models for the determination of isovalues

### 2.7. Step 5 : Deduction of points better than a threshold

For a given objective, each experience point calculated so far falls into one of the following three sets:

- $S_{bd}$ : the "bad" points (worse than the reference value in the specifications);
- $S_{gd}$ : the "interesting" points (whose objective is better than that indicated by the specification);
- $S_{iso}$ : the border points (placed approximately on the corresponding iso-surface).

Logically, in a context of searching for optimal conditions, the number of points in  $S_{gd}$  must be much lower than in  $S_{bd}$ . Therefore, for the purpose of graphical representation, it will certainly be simpler and clearer to represent the smallest set, i.e.  $S_{gd}$ . In all cases, the  $S_{iso}$  set will also be useful to represent, in particular in order to restore the tolerance information taken into account by  $\varepsilon$ .

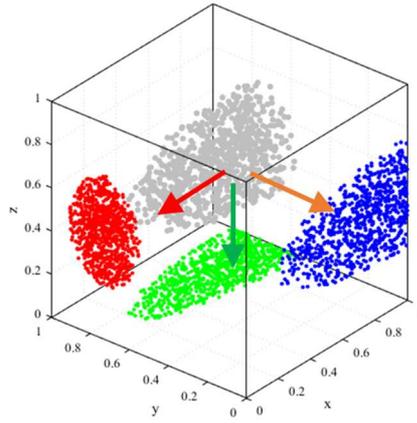
Then, the operations described in the preceding paragraph directly make it possible to deduce the coordinates  $\mathbf{x}$  of the points such that  $M_j(\mathbf{x}) \leq v_{jp} \cdot (1 + \varepsilon)$ , that is to say all the points for which the value of the model  $M_j$  of the output  $y_j$  is better than the value  $v_{jp}$ .

### 2.8. Step 6: Projection of boundary points

This step proposes to arrive at a graphical representation of the sets of points defined during the step 5.

It is well known that it is all the more delicate and complex to represent functions whose number of dimensions involved is high. To overcome partially this difficulty, it is necessary to abandon the representation of continuous variations of these functions, but rather to concentrate on the display of their *isovalue surfaces*. Moreover, representations can only be useful if they are easily understood; in this context, only 2D graphs (i.e. linking two factors to each other) or even 3D plots are exploitable.

In this work, for each output variable  $y_j$ , it was made the choice to represent the point clouds, established in the previous step, by their 2D-projections that is to say onto the  $\binom{k}{2}$  two-dimensional subspaces linking two distinct factors among  $k$ .



**Fig. 5** Illustration of the principle of projection of a point cloud on its two-dimensional subspaces

### 2.9. Step 7: Determination of enclosing 2D contours

Each 2D projection of the point cloud then forms another (2D) cloud included in a square domain. Depending on the volume of calculations made during step 4 for the search for new isosurface points, the distribution of points in this 2D cloud can be more or less homogeneous (holes may appear).

To take into account the theoretical continuity of these points between them (since they are supposed to belong to the same **continuous** (hyper-) surface), these points **must be** grouped by the construction of contours around them.

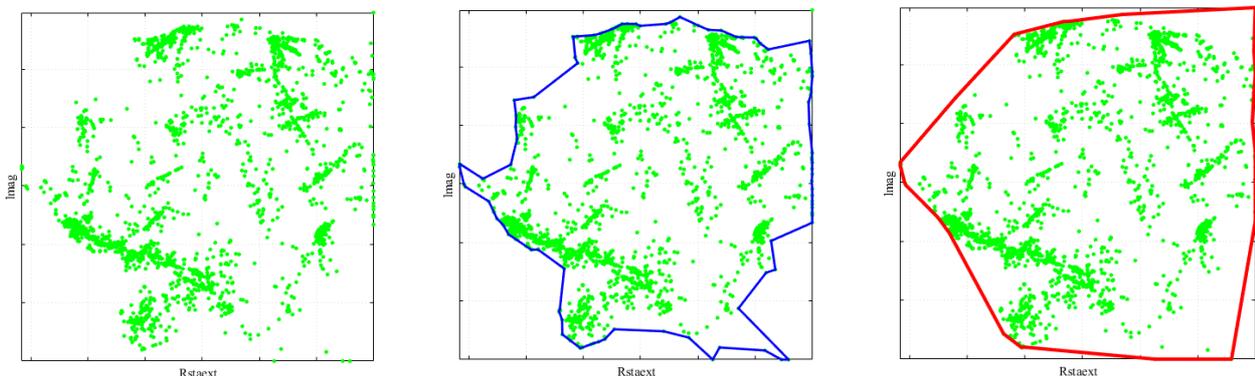
This grouping operation simply corresponds to the construction of *Alpha Shapes* [9] around these points, in the plane of their projection. This algorithm also gives the possibility to surround a cloud of points more or less roughly, by adjusting a radius parameter.

It determines the points in the cloud that alone define the enveloping shell of this point cloud. It is constructed implicitly as an approximation of the contour path taken by a virtual sphere of radius  $r$  held in contact with the point cloud. The smaller  $r$  is, the smaller the sphere is and the more it can "sink" into the cloud; this then defines a contour with a smaller perimeter and a cut-out shape. Conversely, for a large  $r$ , the sphere will tangent the point cloud without penetrating it, and the contour will be "more convex" and of larger perimeter.

Figure 6 illustrates the application and configuration of this method.

In our context, the dimensions of the domain of study are fixed (it is a hypercube of unit side length). The parameter  $r$  can then also be fixed. The tested value of  $r=0.05$  was chosen because it generates Alpha shape contours that follow the point distributions in the plane fairly closely.

The determination of these contours is carried out for each of the  $r$  outputs.



**Fig. 6** Left: example of 2D projection of a  $kD$  scatter plot - Center: corresponding AlphaShape contour line with  $r=0.05$ ; Right: corresponding AlphaShape with  $r=0.3$  – ( $l_{mag}$ - $R_{staext}$  2D projection plane)

### 2.10. Step 8: Intersection of contours

The next step consists in superimposing these contour lines. This operation can thus be used to visually:

- understand the evolution of the placement of points in the domain for different values  $v_{jp}$  of an output  $y_j$ ;
- identify the intersection areas between several contours, each characterizing an output  $y_j$ .

The second objective is the most important, as it allows to characterize the existence (or not) and the distribution of the points satisfying several output objectives at the same time. For that, the level  $v_{jp}$  to consider for each output  $y_j$  is chosen. Then, for each pair of factors (among  $k$ ), the contour curves associated with the previously selected levels are drawn. Finally, for each of these 2D projections, the intersections of the contours can be deduced.

In the case where there is a non-empty intersection between these contours for all pairs of factors (i.e. in all 2D projections), then it is possible that some points could satisfy all the output objectives. However, it is not certain. Indeed, the graphic display mode based on 2D projections is not able to represent correctly concave hypersurfaces. On the other hand, if at least one 2D projection has no intersection between the contours, then it is almost certain that no solution to the optimization problem exists. Still, this certainty cannot be complete because the definition of the isovalue surfaces is necessarily imperfect.

The difficulty of correctly handling non-convex sets of points can possibly be circumvented in several ways. The first one is to use the complementary set of points, i.e.  $S_{bd}$ . Thus, if  $S_{gd}$  is non-convex,  $S_{bd}$  may be (but it is still not sure). A second solution is to cut the non-convex set ( $S_{gd}$ ) into several parts; if these parts are well chosen and small enough, they are likely to be each convex. However, this strategy of splitting the initial set makes the subsequent graphical representation steps more complex.

### 2.11. Step 9: Estimation of near-optimal conditions

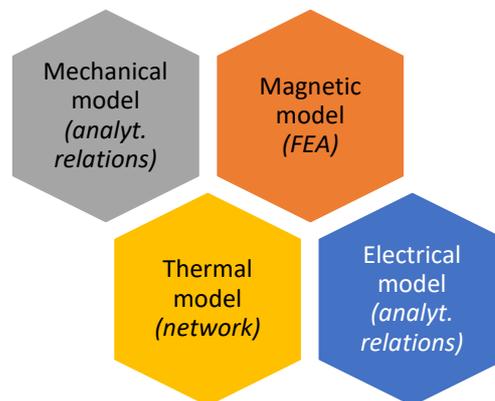
In the case where all the 2D projections have one (or more) intersections between contours, it becomes possible to define the bounds for the coordinates of the possible solution points, by defining the largest  $k$ -dimensional hyperrectangle including completely all the intersection zones.

## 3 Applicative example

As an illustration, the approach presented in the previous section is applied for the optimal sizing of an Internal Permanent Magnet Synchronous Machine (IPMSM), used as a Starter-Generator in an automotive embedded system [10], as shown by Figure 7.



**Fig. 7** Exploded view of the IPMSM (stator and rotor on the left) and the clutch system



**Fig. 8** Reference modeling of the IPMSM used for the optimization problem, composed of four coupled physical submodels

### 3.1. Definition of the optimization problem

The IPMSM machine is reliably computed by a multiphysical model composed of four coupled sub models (Fig. 8), which describe the electrical, magnetic, thermal and mechanical aspects. These different physics are coupled together in pairs (except between the electrical and mechanical models).

However, the machine is optimized for starter operation here. For this mode, the currents, and therefore the losses, are important, but imposed in a short way (a few seconds). Their impact on the rise of temperatures within the machine is real, but is insufficient to make temperatures exceed the limits supported by the insulating materials (during cold starts). In this context, thermal output variables will not be considered afterwards.

It is the same for mechanical quantities, such as those relating to vibration and noise, which will not be exploited in this context.

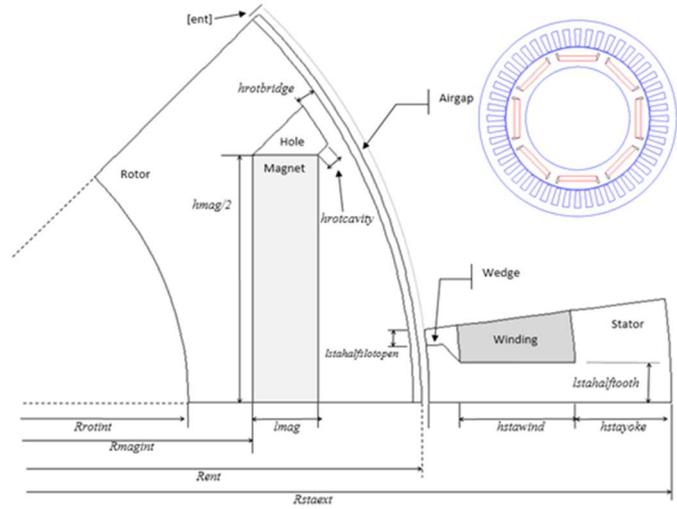
Hence, only the electromagnetic aspects are actually involved in further developments.

This model is parameterized according to 14 geometric factors  $x_i$  (Table 1). The power supply is fixed.

Figure 9. gives a cross section of the machine and the description of all the (geometrical) factors.

**Table 1** Input variables (factors –  $k=14$ )

Parameter $x_i$	Description
$R_{stext}$	Outer radius of stator
$R_{ag}$	Outer radius of rotor (inner radius of airgap)
$R_{rotint}$	Inner radius of rotor
$h_{rotbridge}$	Height of rotor magnetic bridge
$l_{mag}$	Width of rotor magnet
$h_{mag}$	Height of rotor magnet
$h_{rotcavity}$	Height of air magnetic deflector
$R_{magint}$	Inner radius of rotor magnet
$l_{stahalftooth}$	Half of stator tooth width
$h_{stayoke}$	Height of stator yoke
$h_{stawind}$	Height of stator slot part with windings
$l_{stahalfstotopen}$	Half of stator slot opening
$h_{staffootooth}$	Width of tooth base widening
$L_z$	Axial length of the machine



**Fig. 9** Cross section of the IPMSM, with the indications of the 13 geometrical parameters (factors) (+  $L_z$  the axial length)

The output quantities (responses  $y_j$ ) considered in this study are listed in Tab. 2, as well as their associated reference values  $y_{refj}$ . It is recalled that these are target values, typically specified by the specifications associated with the optimization problem.

It is clear here that there is no longer any distinction between constraints (to be satisfied) and objective function (to be minimized): all output variables must be close to and ideally exceed their respective reference values.

To simplify, only the starter function of the machine is considered here. In this mode, the efficiency  $\eta$  is necessarily low (transient mode at very high currents and low average speed). To sum up, the global objective is the conception of an IPMS motor with a high torque-to-weight ratio.

**Table 2** Output variables (responses –  $r=3$ ) defined in specifications

Parameter $y_j$	Names	Ref. value $y_{refj}$
Total mass	$M_{tot}$	( $\leq$ ) 15 kg
Efficiency	$\eta$	( $\geq$ ) 30%
Torque (electromagnetic)	$T$	( $\geq$ ) 90 Nm

### 3.2. Initial DoE & Interpolation models

In the first place, two experimental designs are calculated successively, over the entire field of study, to carry out the following two operations:

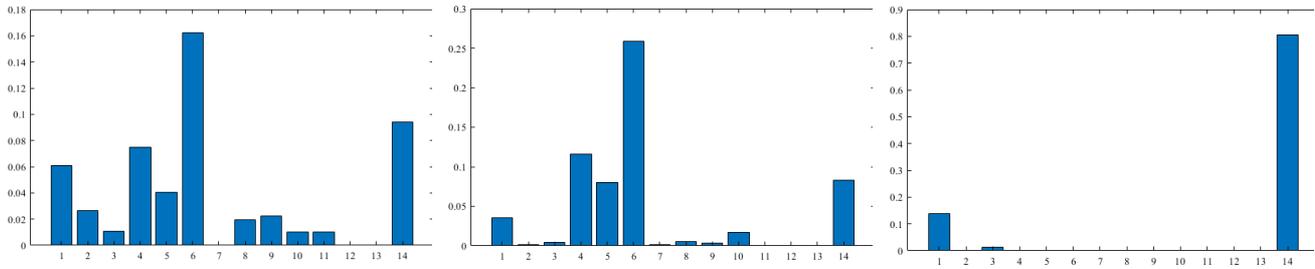
1. Conduct a sensitivity study on factors with respect to responses.
2. Calculate interpolation models of responses, based only on the factors previously considered as significant.

It was chosen to make a first design  $P_1$  with  $N = 1,024$  experiments and for  $k=14$  dimensions. This first design  $P_1$  is a Space Filling design, built from the Hammersley's low-discrepancy sampling method [11]. The calculation of such a design is relatively fast because all calculations can be parallelized.

The influence of the 14 factors was measured globally. For that, an *approximation* model of the RS-HDMR type (Random Sampling - High Dimensional Modeling Representation [12-13]) was calculated, for each output, from the  $N$  points of experiments.

The high dimensional model representation (HDMR) technique is a procedure for representing high dimensional functions efficiently. A practical form of the technique, random sampling-high dimensional model representation (RS-HDMR), is based on randomly sampling the overall function (such as the one given by a Space-Filling design). The HDMR method proposes to construct interpolations as a (finite) weighted sum of monovariate, bi-variate, etc. subfunctions. It can then be shown that these weighting coefficients can be used to calculate Sobol's sensitivity coefficients [14].

Fig. 10. represents the Sobol's first order indices (in which the interactions between factors are not integrated) relative to the three output variables.



**Fig. 10** First order Sobol's sensitivity coefficients of the  $k=14$  factors on the torque  $T$  (left), the efficiency  $\eta$  (center) and the total mass  $M_{tot}$  (right) variations

These histograms show that some factors are not very significant with respect to one or several outputs. From there, and in order to significantly reduce the size of the problem, only the factors 1, 2, 4, 5, 6 and 14 have been retained in the rest of the study.

Thanks to this sensitivity study, the number of factors could be reduced from 14 to 6, thus substantially simplifying the complexity of the optimization problem.

A new DoE  $P_2$  is then computed considering these  $k=6$  factors only.  $P_2$  is also a Hammersley design with  $N=1,200$  experiments. The calculation of these  $N$  points then makes it possible to evaluate the  $r$  responses  $y_j$  in the whole field of study.

It should be noted that the points from the previous  $P_1$  design of experiments could have been reused, since the deleted factors (factors 3, 7, 8, 9, 10, 11, 12 and 13) necessarily have negligible effects (this is why they have been deleted).

This leads to the definition of  $r$  interpolation models, one per output. All these models are built from RBF (Radial Basis Functions). The RBF kernel used is the thin plate function [15]:

$$\phi(r) = r^2 \cdot \ln(1 + r) \quad (2)$$

Its reliability and superiority over other kernel function have been proven by experience.

### 3.3. Building of isosurfaces

Each interpolation model is now improved by the addition of new base points placed closest to isovalue surfaces. These isosurfaces are defined for predetermined values  $v_{jp}$  which are reported in Tab. 3.

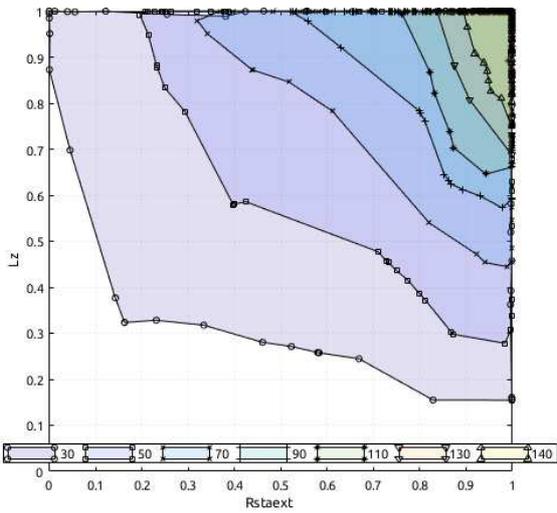
**Table 3** Isovalues for each output variables

Parameter	min( $y_j$ )	max( $y_j$ )	Isovalues $v_{jp}$
Total mass	2.05	22.98	4, 6, 8, 10, 12, 13, 15
Efficiency	$\approx 0$	32.2	20, 25, 27.5, 30, 32.5, 35
Torque	$\approx 0$	99.5	30, 50, 70, 90, 110, 130, 140

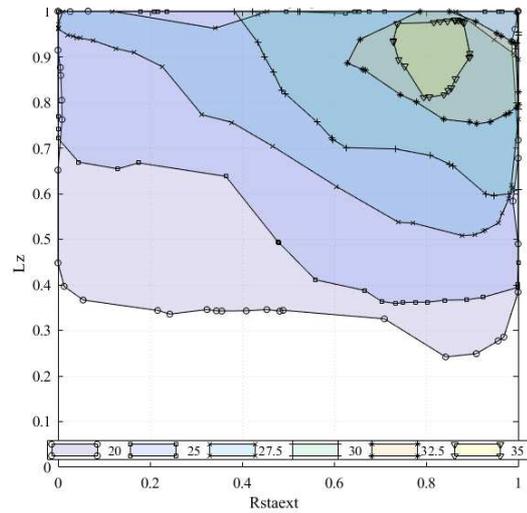
It should be noted here that some isosurface values have been chosen beyond the extreme values observed in the domain by the previous calculation of the two experimental designs. Indeed, it is quite clear that the positioning of the points of these designs does not allow to know exactly the **actual** minimum and maximum values of each output (otherwise, optimization would be greatly facilitated). This is why it is advisable to look for more ambitious values of output variables (corresponding to better performance), but still achievable (otherwise, no isosurface can be built). This makes this step of defining the isosurface values to be found quite tricky.

### 3.4. Graphical representations

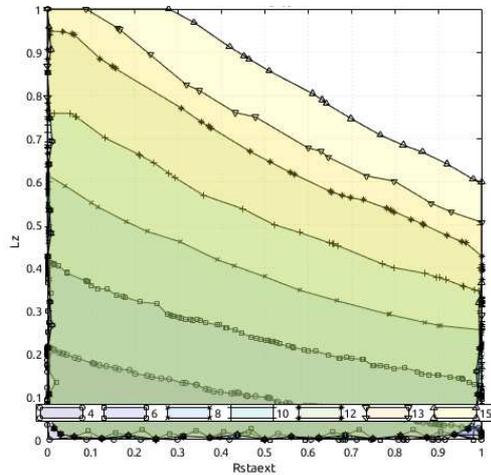
To be represented graphically, isosurfaces are projected on all the  $\binom{6}{2} = 15$  two-dimensional subspaces that can be defined from the  $k=6$  initial factors. Thus for each projection, it is the "silhouette" of the isosurface that is representable, simply by estimating the contour surrounding the projected points. The Alphashape algorithm is used for this purpose. For example, Fig. 11, 12 and 13. give the representation of the isosurfaces projected on the subspace  $L_z$ - $R_{staext}$  for the three responses  $T$ ,  $\eta$  and  $M_{tot}$  respectively.



**Fig. 11** Projection of torque  $T$  isosurfaces onto the  $L_z$ - $R_{staext}$  subspace



**Fig. 12** Projection of efficiency  $\eta$  isosurfaces onto the  $L_z$ - $R_{staext}$  subspace



**Fig. 13** Projection of total weight  $M_{tot}$  isosurfaces onto the  $L_z$ - $R_{staext}$  subspace

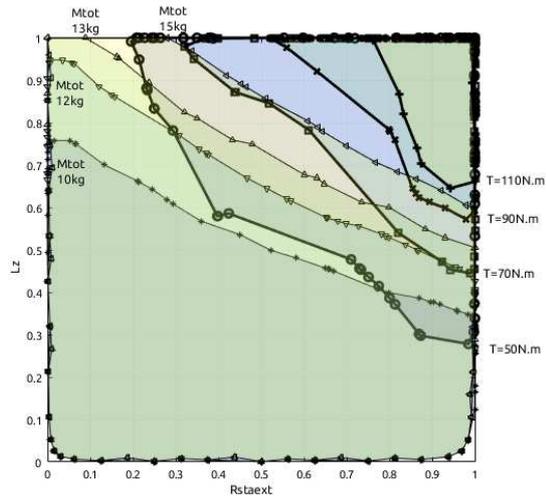
### 3.5. Determination of possible optimal configurations

Once all the projections of the  $n_j$  considered values of the  $r$  responses calculated, they can be superimposed. If there are common areas (that is non-empty intersections), then they define possible configurations of the input values for which all the  $r$  responses satisfy the specifications.

An illustration of this procedure is given by Fig. 14. Four isosurfaces of the total mass of the machine ( $M_{tot}=\{10, 12, 13, 15\}$  kg) as well as four isosurfaces of the electromagnetic torque ( $T=\{50, 70, 90, 110\}$  Nm) are reported. This representation then shows the following probable facts:

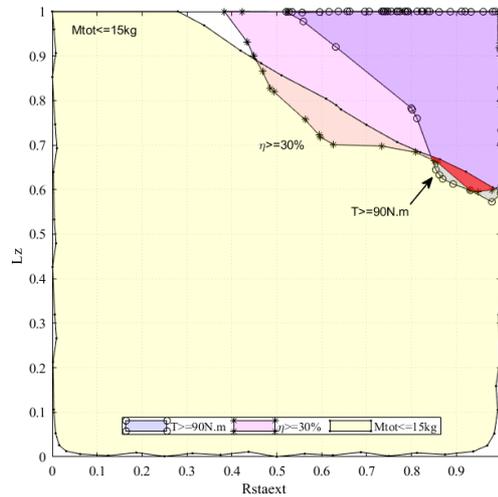
1. The machines (10 kg-50 Nm) and (12 kg-70 Nm) can be defined (because intersections between corresponding isosurfaces exist).
2. The configuration (14 kg-90 Nm) may exist (the corresponding isosurfaces are very close, and one must take into account the uncertainties on their definition).
3. It is not possible to design a machine of this type developing 110 Nm and weighing less than 14 kg.
4. The machine developing 50 Nm can weigh less than 10 kg.
5. All these machines are only defined for large values of the outer radius of the stator ( $R_{stae\text{xt}} \geq 0.8$ ).

One must not forget that these conclusions are only valid if they are not tempered or invalidated by the examination of other 2D projections.



**Fig. 14** Intersection of multiple torque and total mass isosurfaces in the  $L_z$ - $R_{stae\text{xt}}$  subspace

To involve all the outputs taken into account by specifications, efficiency values must be included. Fig. 15. plots the limits of isosurfaces of the torque (greater than 90 Nm), the total mass of the machine (less than 15 kg) and the efficiency  $\eta$  (greater than 30%). Then, a very restricted common area appears, inside which or near which optimal conditions can be found.



**Fig. 15** Intersection of different responses isosurfaces in the  $L_z$ - $R_{stae\text{xt}}$  subspace – the intersecting (common) surface in red

### 3.6. Definition of bounds of probable optimal conditions

The different intersection zones of the  $r=3$  responses projected onto the 15 two-dimensional subspaces make it possible to bound as precisely as possible the values of the  $k=6$  factors for which the optimal conditions are probably located.

This operation is fairly quick, since it is sufficient to consider the 15 sub-spaces successively, retaining each time only the maximum of the minimum coordinates and the minimum of the maximum coordinates of the points within the intersection zone. Table 4. gives these intervals.

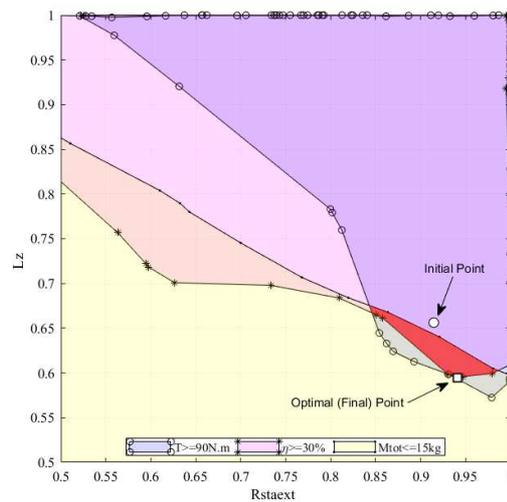
**Table 4** Intervals for the  $k$  input variables, corresponding to the intersection of the  $r$  output isosurfaces

Descriptions	Names	min	max
Outer stator radius	$R_{staext}$	0.843	0.98627
Air gap radius	$R_{ag}$	0	0.67074
Magnetic bridge height	$h_{rotbridge}$	0	0.54369
Magnet width	$l_{mag}$	0.012475	0.57419
Magnet height	$h_{mag}$	0.72147	1
Axial length	$L_z$	0.63704	0.67545

### 3.7. Direct optimization on models of output variables

To verify the relevance of this graphic and visual approach to the treatment of the design problem, a "classical" optimization is carried out here by applying the Sequential Quadratic Programming (SQP) algorithm to find the maximum torque  $T$  while imposing a total mass  $M_{tot}$  lower or equal to 15 kg and an efficiency  $\eta$  greater or equal to 30%. The starting point is chosen as the average of the columns min and max of previous Tab. 4. This optimization uses the surrogate model (based on RBF functions) to find these optimal conditions.

Figure 16. shows the location of the optimal point found by the optimization algorithm. It is integrated within the intersection area between the isosurfaces of  $T$ ,  $M_{tot}$  and  $\eta$ . Performances of the machine for this optimal configuration are given in Tab. 5.



**Fig. 16** Direct optimization using interpolation models  
Objective function:  $T$  – Constraint functions:  $\eta \geq 30\%$  and  $M_{tot} \leq 15\text{kg}$

The optimum value of the torque found is about 100 Nm, and not 90 Nm. This result is not at odds with the graphical representation of Fig. 15, because the isosurface (90 Nm) corresponds to the boundary points of the zone " $T \geq 90\text{Nm}$ ".

For information, Tab. 5. also gives the performance calculated by the reference (finite element) model of the machine, for these same optimal values of factors. These output values are quite close to those given by the interpolation models (indicating a good quality of these RBF interpolation models, at least locally).

**Table 5** Output performances at optimal point

Output values	$T$	$M_{tot}$	$\eta$
Target values (specifications)	90 Nm	15 kg	30 %
Interpolation RBF models	100.9 Nm	15 kg	30 %
Reference model (FEA)	85.4 Nm	15.1 kg	28.9 %

## 4 Remarks

### 4.1. Costs

Applying the approach presented in this article involves the completion of a series of steps. But not all of them represent the same cost in terms of time and/or volume of calculations.

Table 6 attempts to provide an estimate of these relative costs. It is clear that they are highly dependent on the problem under consideration, i.e. in particular on the number of input and output variables to be taken into account.

Here, only steps 2 to 9 are considered.

For step 1, since the calculation of experimental designs can be completely parallelised, its cost depends directly on the performance of the computing machine used and on the number of computing cores it possesses. In addition, the corresponding calculation time is also highly dependent on the cost of evaluating the reference model, which results from its complexity and the expected accuracy of its results. Therefore, estimating the cost of this step does not really make sense.

The next step (1a) is optional and has therefore not been taken into account in the overall counting.

Step 4 remains the most important step, in all respects. Also, its cost is difficult to evaluate because it depends on the level of precision expected by the modeller and/or the time budget he has set himself, in order to define the isovalue sets.

In the context of the study presented in this article, this column "Costs" gives approximately the costs in time expressed in minutes.

**Table 6** Estimation of the relative costs of each step of the proposed methodology.

Step	Output values	Costs [%]
1	Calculation of the basis DoE	-
1a	<i>Problem simplification (sensitivity analysis)</i>	-
2	Definition of $r$ interpolating models	8
3	Determination of isovalue definition	2
4	Search for iso-performance configurations	66
5	Deduction of points better than a threshold	1
6	Projection of boundary points	3
7	Determination of enclosing 2D contours	8
8	Intersection of contours	10
9	Estimation of near-optimal conditions	2

### 4.2. Strengths and Weaknesses

The proposed methodology presents the following advantages:

- The multidimensional and multiobjective optimization problem is simplified in a (visual) detection of the intersection of contours of 2D point clouds. In this sense, this approach can be viewed as a method of visualizing multidimensional data [16-17].
- It becomes very easy to know if a problem has at least one solution, for a given set of specifications.
- Isosurfaces definitions can be recorded in databases aiming at storing machine models.

However, the application of this method presents some difficulties:

- The consideration of a large number of factors can be problematic as the number of projections to be generated and analyzed becomes very large in this case.
- Sets of concave points cannot be correctly described by contours defined by projections in subspaces.
- This approach is interesting when contours are small with respect to the size of the domain; this only occurs when isovalues are close to extreme values.
- The definition of isosurfaces is an important step but is difficult to carry out; in particular, the determination of isosurface points relatively well distributed in space is not easy, especially when the number of dimensions  $k$  is high.
- Determining the isosurface values to be defined can be tricky, as one does not know a priori the "true" extreme values of the outputs to be optimized. In addition, their number must be kept small in order to limit the work necessary to find the points placed on each of these isosurfaces.

The main objective of the approach presented in this article is to graphically represent the complexity of an optimal dimensioning problem (of an electrical machine in particular). To achieve this, in a first step, only a limited number of values of the output quantities are considered, through the construction of isovalue sets. In a second step, this graphical approach can be used to determine optimal conditions, relative to the definition of the specifications.

Thus, if one must consider only the optimization approach, it is clearly indirect. Therefore, in terms of efficiency and speed in the search for optimal conditions, it cannot compete with direct optimization approaches (as with the use of SQP for instance).

Its contribution must be sought in everything that SQP does not give; i.e. an overview of the variations of all the outputs according to all the inputs, as well as the generation of a substitute model that can be reused afterwards, possibly with other hypotheses (specifications and/or different constraints...).

## 5 Conclusion

This article has focused on presenting a methodology for optimization design problems, essentially based on the construction and analysis of 2D graphic representations.

This approach uses all the interest represented by the isovalue surfaces, that is to say the continuous sets of the domain where output variables take particular values (typically the limits defined by specifications).

An applicative example of the application of this methodology was given. It concerned the optimization of a permanent magnet Synchronous Machine, with regard to criteria of mass, efficiency and electromagnetic torque. This case demonstrated the interest of this approach. Future work will complement this work by adding new objectives to the specifications. In particular, limit values will be considered for winding and permanent magnet temperatures.

This approach has been put in place to address many weak points encountered when directly applying "classic" optimization algorithms. In particular, this method gives the possibility of determining quite simply if at least one solution exists to the sizing problem. In the positive case, it also gives all the optimal values of input variables. If no solution appears, this approach also makes it possible to test easily different constraint relaxation scenarios, to reach nevertheless interesting optimization solutions.

By its nature, this visual approach does not replace conventional optimization algorithms, but rather complements them by providing additional information that can be directly understood by the designer.

This methodology will in the future benefit from many improvements, especially concerning the construction of isovalue surfaces. It may also integrate other concepts, such as the definition of Pareto sets and the consideration of uncertainty data on input and output variables.

## References

1. S. Shan, G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions", *Struct. Multidisc Optim.*, Vol. 41, pp. 219-241, 2010
2. F. Campolongo, A. Saltelli, J. Cariboni, "From screening to quantitative sensitivity analysis. A unified approach", *Computer Physics Communications*, Vol. 182, pp. 978–988, 2011
3. V. K. Ky, C. D'Ambrosio, Y. Hamadi, L. Liberti, "Surrogate-based methods for black-box optimization", *International Transactions in Operational Research*, 2016
4. O. Kurasova, T. Petkus, E. Filatov, "Visualization of Pareto Front Points when Solving Multi-objective Optimization Problems", *Information Technology and Control*, Vol.42, No.4, pp. 353-361, 2013
5. F. Kudo, T. Yoshikawa, T. Furuhashi, "A Study on Analysis of Design Variables in Pareto Solutions for Conceptual Design Optimization Problem of Hybrid Rocket Engine", *IEEE Congress of Evolutionary Computation (CEC)*, 2011
6. J. Santiago, M. Claeys-Bruno, M. Sergent, "Construction of space-filling designs using WSP algorithm for high dimensional spaces", *Chemometrics and Intelligent Laboratory Systems*, 2011
7. S. Bagheri, W. Konen, T. Bäck, "Comparing Kriging and Radial Basis Function Surrogates", *Workshop Computational Intelligence*, Dortmund, Nov. 2017
8. L. Jin-Yue, Z. Bao-Ling, "Research on the Non-linear Function Fitting of RBF Neural Network", *International Conference on Computational and Information Sciences*, 2013
9. Y. Li, G. Gao, B. Cao, L. Zhong, Y. Liu, "Building Boundaries Extraction from Point Clouds using Dual-Threshold Alpha Shapes", *23rd International Conference on Geoinformatics*, 2015
10. S. Vivier, G. Friedrich, "Comparison between Single-Model and Multimodel Optimization Methods for Multiphysical design of Electrical Machines", *IEEE Trans. on Industry Applications*, Vol. 54, Issue 2, pp. 1379-89, 2018
11. S.-H. Cheng, C.-H. Lo, T.-C. Lin, "Use of confidence region in the optimal design of a separation process in the presence of uncertainties", *6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, 2017
12. O. F. Alis, H. Rabitz, "Efficient implementation of high dimensional model representations", *Journal of mathematical chemistry*, vol. 29, N°2, 2001
13. Y. Liu, M. Y. Hussaini, G. Ökten, "Accurate construction of high dimensional model representation with applications to uncertainty quantification", *Reliability Engineering and System Safety*, 2016

14. M. Munoz Zuniga, S. Kucherenko, N. Shah, "Metamodelling with independent and dependent inputs", *Computer Physics Communications*, Vol. 184, pp. 1570-80, 2013
15. Y. Ji, S. Kim, "An Adaptive Radial Basis Function Method using Weighted Improvement", *Winter Simulation Conference*, 2013.
16. S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, V. Pascucci, "Visualizing High-Dimensional Data: Advances in the Past Decade", *Eurographics Conference on Visualization (EuroVis)*, 2015
17. A. K. A. Talukder, K. Deb, "A Topologically Consistent Visualization of High Dimensional Pareto-front for Multi-Criteria Decision Making", *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018