Surrogate models with kriging and POD for the optimization of an intake port

Rajan Filomeno Coelho*, Piotr Breitkopf*, Pierre Villon*, Maryan Sidorkiewicz**

* Laboratoire Roberval, UTC-CNRS (UMR 6253), Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP 20529 – 60205 Compiègne, France
E-Mail: {rfilomen,piotr.breitkopf,cklv}@utc.fr
** Renault, Technocentre, Avenue du Golf 1, 78288 Guyancourt, France
E-Mail: maryan.sidorkiewicz@renault.com

1. Abstract
The automotive industry is facing increasing demands in terms of cost reduction and environment responsibility. To comply with new European emission standards, while satisfying the need for higher engine performances, the intake ports have to be optimized with respect to the mass flow rate and the aerodynamic turbulence inside the combustion chamber.

In this study, the shape of an intake port is optimized for a gazole engine. CFD calculations are post-processed to maximize two competing objectives: the mass flow and the tumble. Nevertheless, as the evaluation of these criteria is expensive (more than 2 hours of CFD computations for each design), the optimization is split into two phases: first, a series of refined computer experiments is designed and performed to get the velocity field inside the cylinder (from which the mass flow and tumble are calculated); then, the optimization is performed with surrogate models.

We investigate two families of reduced models for the multiobjective optimization of the intake port geometry: kriging surrogate surfaces, based directly on the post-processed values, and Proper Orthogonal Decomposition (POD) models to reduce the 3D velocity field.

2. Keywords: intake port, reduced models, surrogate models, kriging, proper orthogonal decomposition.

3. Introduction
In the development of a car engine, the design of intake ports is an important task since their geometry greatly influences the flow inside the combustion chamber, hence acting dramatically on the global performances of the engine. In particular, two competing criteria need to be carefully considered: the mass flow rate $Q$ and the tumble $T$ (i.e. the aerodynamic turbulence) inside the cylinder.

To estimate the responses of $T$ and $Q$ for a given intake port design, the full Navier-Stokes equations need to be solved for the cylinder and the intake port. This is usually performed by using CFD codes (e.g. implementing the finite volume method), which provide accurate behaviour of the 3D velocity field. However, each CFD run is expensive in terms of computational time and resources, which hinders the systematic use of accurate numerical simulations in a complete optimization process.

Therefore, in this study, the emphasis is put on the development of reliable surrogate models in order to perform the shape optimization of intake ports according to two conflicting objectives (the tumble $T$ and the mass flow rate $Q$). Beside the multiobjective optimization of the intake port geometry, the surrogate models can also be applied to estimate the robustness of the solutions with respect to the given tolerances of the design variables.

After the description of the CAD-CFD simulation process (§4), two different model reduction approaches will be discussed: kriging and Proper Orthogonal Decomposition (POD). In this paper, the emphasis is put on the surrogate models built upon the kriging method, which will be presented and applied for the robust multiobjective optimization of the intake port shape (§5). Then, due to lack of space, only the basics of a novel Proper Orthogonal Decomposition (POD) method are introduced (§6); their application to the intake port is left for the oral presentation.

4. Simulation process for an intake port – Motivation for surrogate models
The region of interest is depicted in Figure 1. While the cylinder geometry remains fixed, 6 geometrical parameters (radii of curvature, diameters and positioning of components) enable the modification of the intake port shape.

![Figure 1: Region of interest (intake port + cylinder).](image_url)
Practically, to estimate the quality of a given design, a 4-step process must be performed:

1. CAD: generation of the CAD geometry for the given set of parameters;
2. automatic 3D mesh generation of the fluid domain (constituted by the combustion chamber and the intake port);
3. CFD: Navier-Stokes steady simulation computed by means of the STAR-CD software (see Figure 2) [1];
4. post-processing of the CFD results: a cylindrical zone is defined in the combustion chamber (cf. Figure 3), in which the velocities are collected to calculate the mass flow rate \( Q \) and tumble \( T \).

Since each simulation demands more than 2 hours of computational time, approximate models are mandatory to complete a full optimization process. To reach this requirement, two options are available:

- approximating directly the global post-processed values (i.e. the tumble and the mass flow rate): this is done in §5, using the kriging method;
- reducing the whole 3D velocity field inside the combustion chamber, and calculating eventually the post-processed values from the reduced vector field; this is discussed in §6 using the Proper Orthogonal Decomposition (POD) method.
5. Surrogate models based on kriging for multiobjective optimization

5.1 Design of experiments
First, a series of numerical simulations has been performed for 164 designs, composed of a Latin Hypercube Sampling of 100 designs and a 2-point full-factorial (i.e. \(2^6 = 64\) designs). From these designs, 18 must be discarded, since the CAD does not generate a valid geometry for the automatic mesh generator. In the remainder of the paper, the database contains thus a set of 146 (feasible) points.

5.2 Surrogate models based on kriging
Based upon the 146 simulations or “snapshots” calculated by the accurate simulation process, a kriging response surface is built directly on the post-processed values of the tumble \(T\) and the mass flow rate \(Q\):

\[
\{ x_1, x_2, x_3, x_4, x_5, x_6 \} \rightarrow \{ T, Q \}_{\text{kriging}}
\]

Kriging is an interpolation method modelling the function as a realization of a stochastic process. The kriging predictor is defined in order to minimize the expected squared prediction error subject to being unbiased as well as being a linear function of the observed responses [2].

Now that an inexpensive surrogate model is available, different strategies are investigated in order to get fruitful information about the multiobjective optimization problem. First, some preliminary concepts are introduced in the next section.

5.3 Preliminary concepts on multiobjective optimization
When more than one criterion is pursued during the optimization, the concept of dominance is introduced: in a minimization problem, a vector of objectives \(u\) is said to dominate \(v\) if and only if all components \(u_i\) of \(u\) are equal to or better than \(v_i\) (\(\forall i: u_i \leq v_i\)), and \(u\) is strictly better than \(v\) for at least one component (\(\exists j: u_j < v_j\)).

Then, in the design space, the Pareto set is defined as the collection of points dominated by no other points; the image of the Pareto set in the objective space is called the Pareto front (cf. Figure 4).

Due to the presence of conflicting objectives, the solution is generally not unique in multiobjective optimization. The user has to provide additional information about his/her preferences in order to find the best compromise. Three different approaches are available in the literature (see Figure 5 [3]):

- preferences may be included since the beginning of the search process (\(a\ priori\) methods): the user has to assign a weight to each criterion, or at least a ranking of the \(m\) objectives;
- preferences may be used at the end, when the Pareto front has been completely determined (\(a\ posteriori\) methods);
- preferences may be used during the optimization process, in an interactive way (progressive methods).

![Figure 4: Pareto front \(PF\) (dotted line) in a 2-objective minimization example.](image)

![Figure 5: \(A\ priori\), progressive and \(a\ posteriori\) methods for multiobjective optimization [3].](image)

In this work, the first and second strategies are illustrated respectively in §5.3 and §5.4.
5.3 Multiobjective optimization by the weighted sum method

The most intuitive and widespread method consists in aggregating the objectives into a single cost function, for example through a weighted sum:

\[ f_{\text{cost}} = \sum w_i f_i. \]  

(2)

In this study, the same value is assigned for all weights, and the objectives are divided by a scaling factor:

\[ f_{\text{cost}} = \frac{T}{T_{\text{ref}}} + \frac{Q}{Q_{\text{ref}}}, \]

(3)

where \( Q_{\text{ref}} \) and \( T_{\text{ref}} \) are the maximum values found in the snapshot database (respectively for the mass flow rate \( Q \) and the tumble \( T \)). The values of \( T \) and \( Q \) are calculated with the surrogate response surfaces. By taking each point of the database as a starting point of a gradient-based sequential quadratic programming (SQP) optimizer, different solutions are obtained, as illustrated in Figure 6.

![Value of the objectives with respect to the starting point](image)

![Optimal variables with respect to the starting point](image)

Figure 6: Objectives and optimal variables with respect to the starting point of the optimizer.

From these results, 5 distinct solutions \( x_{\text{opt}}^{(k)} \) can be isolated, where \( k \) refers to the number of the design in the database used as the starting point for the optimization.

A further insight on the quality of the results may be found by considering additionally the tolerances on the variables, which are characterized by a Gaussian distribution around their nominal value, for variables \( x_3 \) and \( x_5 \) (standard deviation \( \sigma_3 = 11.11\% \) for the 3rd variable, and \( \sigma_5 = 4.2\% \) for the 5th variable). From these distributions, a Monte Carlo simulation of 10,000 points is launched for each optimal solution \( x_{\text{opt}}^{(k)} \); the corresponding results are drawn in the objective space, each solution being depicted with its dispersion due to the tolerances (cf. Figure 7). While the 4 solutions \( x_{\text{opt}}^{(4)}, x_{\text{opt}}^{(13)}, x_{\text{opt}}^{(80)} \) and \( x_{\text{opt}}^{(140)} \) are mutually non-dominated and present a low dispersion rate, the solution \( x_{\text{opt}}^{(40)} \) is clearly dominated (by \( x_{\text{opt}}^{(80)} \) and \( x_{\text{opt}}^{(140)} \)) and must be discarded.

It is important to emphasize that these results are obtained for the same value of the weights (\( w_1 = w_2 = 0.5 \)); the difference between the results is explained only by the fact that the cost function is multi-modal (i.e. presents several local minima). Each optimal solution depends directly on the starting point used for the optimization.

For this case, to get a better idea of the performances that can be reached in the whole design space, an \textit{a posteriori} method is more adapted, as demonstrated in the next section.
5.4 Multiobjective optimization by the Non-dominated Sorting Genetic Algorithm II

In *a posteriori* methods, the main step consists in drawing up the shape of the Pareto front, in order to get information about the ranges of performances that can be reached for each objective. As the genetic algorithms do not work with a single point at each iteration, but with a population of individuals evolving with respect to a specific criterion (called the *fitness* function [4]), they are ideally suited for multiobjective optimization. After the random generation of the initial population, the individuals are selected following the value of their fitness function: the individuals with higher fitness values are more likely to be chosen to take part of the process of recombination (cf. Figure 8 [left]). There are various methods to find out the nondominated solutions using evolutionary algorithms, as VEGA (Vector Evaluated Genetic Algorithm, proposed by Schaffer [5]), MOGA (Multi-Objective Genetic Algorithm [6]) and NSGA (Non-dominated Sorting Genetic Algorithm [7]).

Recent advances in *a posteriori* techniques include CMEA (Constraint Method-based Evolutionary Algorithm [8]), NPGA2 (Niched-Pareto Genetic Algorithm 2 [9]), NSGA-II (Nondominated Sorting Genetic Algorithm-II [10]), PAES (Pareto Archived Evolution Strategy [11]) and SPEA2 (Strength Pareto Evolutionary Algorithm 2 [12]).

In this work, the NSGA-II method is applied for the optimization of the intake port; the values of the objectives being estimated thanks to the surrogate models based on kriging.

---

**Figure 7:** Results for the weighted sum method for different starting points.

**Figure 8:** General flow-chart of a genetic algorithm [left]; description of NSGA (in a minimization problem): ranking of the population with respect to non-domination [right].
The results at various generations are depicted in Figures 9 to 11 (for a population of 200 individuals), showing a rapid convergence towards the Pareto front. The total number of calls to the simulations amounts to 10,000 (i.e. 50 generations x 200 individuals), demonstrating the absolute need for surrogate models to accomplish this multiobjective optimization task.

Once an overview of the Pareto front has been found, a multicriteria decision aid method can be used to select a unique solution following the user’s preferences [3]. Beside the reduction of the post-processed responses by means of a general meta-model, another option is possible, consisting in the
reduction of the 3D velocity field inside the combustion chamber, as described below.

6. Proper Orthogonal Decomposition (POD)

6.1 POD – Method of snapshots

From a series of snapshots of the velocities computed by the CFD simulation, the POD (also called Principal Component Analysis or Karhunen-Loève Expansion) allows for expanding a field as a sum of eigenmodes constituting a vector base; any field can be expressed as a linear combination of its corresponding modes. The main difficulty of this approach in shape optimization consists in the variation of the boundaries of the domain between different snapshots and in the incompatibility of meshes automatically generated by CAD programs.

In this work, a parameter-dependent POD is proposed to decompose the coupling variables, by using the method of snapshots [13]. For instance, for a variable $v$, starting from the values of $v$ (called snapshots) obtained by accurate numerical simulations for a representative set of designs, the idea is to build a linear basis to represent any vector $v$, this basis being guaranteed by construction to be optimal to describe a given sample set of observations:

$$v = \sum_{i=1}^{n_s} \phi_i \alpha_i + \bar{v} \quad (4)$$

where:
- $\bar{v}$ is the mean vector (averaged over all snapshots);
- $n_s$ is the number of snapshots;
- the $\{\phi_i; i=1, \ldots, n_s\}$ are the POD basis vectors;
- the $\{\alpha_i; i=1, \ldots, n_s\}$ are the POD linear expansion coefficients.

The construction of the POD basis is divided in 3 parts. First, the “deviation” matrix $V$ is built by storing the pressure vectors of all snapshots, and subtracting the mean vector $\bar{v}$ to each row of $V$:

$$V = \begin{bmatrix} v_1^1 - \bar{v}_1 & \cdots & v_1^{n_s} - \bar{v}_1 \\ v_2^1 - \bar{v}_2 & \cdots & v_2^{n_s} - \bar{v}_2 \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (5)$$

Then, the covariance matrix $C$ is calculated:

$$C = V \cdot V^T \quad (6)$$

Finally, the POD modes $\phi_i$ are obtained by extracting the eigenvectors of $C$.

As the velocity field is defined on different meshes for the intake port problem (due to the automatic re-meshing of the whole domain for any new design), a preliminary interpolation (on a reference grid) is performed.

The reconstruction of the $z$-velocity field on a section of the cylinder is depicted in Figure 12 (with 6 modes kept for the approximation), and the results show a very good adequacy between the accurate simulation (left) and its POD approximation (right).

![Figure 12: z-component of the velocity field in a section of the cylinder: computed by the CFD simulation [left] and reconstructed by POD with 6 modes [right].](image)

The average reconstruction error (over the 146 designs of the snapshot database) obtained by POD is depicted in Figure 13, with respect to the number of successive modes used for the approximation.
6.1 POD – Adaptation for incompatible meshes

Nevertheless, when the meshes and/or geometries are incompatible, a deeper insight on the POD definition care should be considered. Indeed, for the intake port, the direct CAD-CFD simulation process generates incompatible meshes (defined on the same geometrical domain). To address this issue with proper care, a sound POD procedure for the response (here: the velocity field) must be defined on the same geometrical domain, but accounting for the different finite element meshes.

In the general case, if we consider a continuous function $u(x,t)$, where $x$ denotes the position and $t$ the “time” (in the discrete case, for a parameter-based POD, $t$ will be an integer referring to the number of the snapshot), the covariance between two points $x$ and $y$ can be defined by (7):

$$k(x,y) = \int_{t_0}^{t} u(x,t) u(y,t)' \, dt.$$  

Then, the POD problem can be written as follows:

$$\int_{\Omega} k(x,y) v(y) \, dy = \lambda v(x) \quad \forall x \in \Omega,$$

where $\Omega$ is the space domain.

For incompatible meshes, the functions $u$ can be interpolated, using a classical finite element approximation:

$$u(x) = \sum_{i=1}^{N} N_i(x) u_i,$$

Alternatively, in order to guarantee a smoother interpolation of the data, a diffuse interpolation (also called Moving Least Squares interpolation [14,15]), is proposed in this work:

$$u(x) = p(x)' a_x,$$

where the coefficients $a_x$ are determined by minimizing for each $x$ the functional $J_x$:

$$\min J_x(a_x) = \frac{1}{2} \sum_{i} w_i(x,x) (p(x)' a_x - u_i)^2,$$

where the weights $w_i$ depends on the distance between $x$ and the initial data set points.

Then, for $m$ snapshots (with $N$ components for each snapshot), the snapshot matrix is given by (12):

$$U(N,m) = \begin{bmatrix} u_1(t_1) - \bar{u}_1 & u_1(t_2) - \bar{u}_1 & \cdots & u_1(t_m) - \bar{u}_1 \\ u_2(t_1) - \bar{u}_2 & u_2(t_2) - \bar{u}_2 & \cdots & u_2(t_m) - \bar{u}_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_N(t_1) - \bar{u}_m & u_N(t_2) - \bar{u}_m & \cdots & u_N(t_m) - \bar{u}_m \end{bmatrix},$$

and the covariance matrix can be computed as introduced in §6.1:

$$C_{ij} = \sum_k U_{ik} U_{kj}' = \sum_k u_i(t_k) u_j(t_k).$$

A different operator can be defined for building the covariance matrix; by lack of space, its definition will be left for the oral presentation, along with the comparison of the POD techniques used in this work for the intake port application.

7. Conclusions

In this work, the multiobjective optimization of an intake port is discussed. In order to perform this task at a reasonable cost, two approaches for model reduction are investigated.

First, a general meta-model based on kriging is devised to build response surfaces of the macroscopic variables (the tumble and mass flow rate). The surrogate surfaces allow for performing several tasks, as the multiobjective optimization (by the weighted sum
technique and a Pareto method called NSGA-II). A sensitivity analysis is also possible to account for the tolerances on two variables (related to the manufacturing process of the intake port). The high number of function evaluations required by these methods demonstrate the necessity of inexpensive surrogate models.

Another model reduction approach consists in re-creating the whole 3D velocity field (from which the objectives are computed), by means of the Proper Orthogonal Decomposition method. However, in order to account for incompatible meshes (automatically generated by the CAD and meshing process), a specific definition of the POD operation is mandatory, in particular for the computation of the covariance matrix. The results obtained with different POD techniques will be exposed in detail in the oral presentation.

8. Acknowledgments
This work has been supported by the National Agency of French Research/National Network of Software Technologies (ANR/RNTL), in the framework of the OMD project.

9. References