Topology optimization of a mechanical component subject to dynamic constraints*

Sonia Calvel[1], Marcel Mongeau[2†]

1: Technocentre Renault, 78288 Guyancourt cedex, France. sonia.calvel@voila.fr

2: LAAS-CNRS, 31077 Toulouse cedex 4, France. mongeau@cict.fr

Technical report LAAS N° 05374 (July 2005) available from http://mip.ups-tlse.fr/perso/mongeau

Abstract

This paper is concerned with the optimization of continuum structures under dynamic loading using methods from topology design. The constraint functions are non-linear and implicit, their evaluation requires the resolution of a computation-intensive finite-element analysis performed by a black-box commercial structural mechanics software such as MSC/Nastran. We first present a brief overview of topology optimization methods. Then, we propose a practical topology design methodology for handling problems with dynamic constraints over a range of forcing frequencies. We also introduce a new interpolation scheme for the SIMP method that can remove spurious modes in low-density areas and that allows fill-in. We present encouraging 3-D computational experiments on the design of a specific car component for Renault under acoustic and vibratory constraints, using a public-domain sequential quadratic programming software to drive the overall optimization process.

Key words: topology design, topology optimization, structural design, structural optimization, fictive material method, SIMP method, dynamic constraints, automotive industry, mathematical programming, sequential quadratic programming.

1 Introduction

As a consequence of the increasing demand for cars that are more comfortable and safer, the mass of cars steadily increased since 1980. Most car manufacturers are now interested in producing lighter cars in order to save raw materials and in order to produce cars that consume less fuel, and therefore pollute less, while maintaining a high level of comfort and security. This fuel saving concern is critical in Europe, as the European Automobile Manufacturers Association proposed in 1997 to reduce in 2008 the CO_2 emission of its cars by 25% with respect to the emission level of 1995.

*This research was prompted by Renault which also provided the data

†Corresponding author

Topology optimization is concerned with determining the main features and the general shape of a component or of a mechanical structure at the beginning of engineering projects. Current topology optimization software cannot explicitly deal with dynamic constraints such as those involving vibro-acoustics. This is a serious limitation which prevents car manufacturers from using topology optimization for designing car components.

In this paper we consider the problem of finding the design of a car component that is of minimal mass subject not only to static constraints (static stiffness) but also to dynamic (acoustic and vibratory) constraints. The use of mathematical programming algorithms to solve very specific topology optimization problems has been widely described in the literature (see for instance [6]). However, the features of the very general nature of the constraints of the topology optimization problem we consider here make it especially difficult to solve. First, the constraint functions are non-linear and implicit. Their evaluation requires the resolution of a finite-element analysis performed by a black-box commercial structural mechanics software such as MSC/Nastran: one single evaluation can require up to a few hours of computational time. Second, the problem is not convex and the quality of the local minimum obtained depends upon the choice of both the initial solution and the parameters that define the specific topology optimization method implemented. Third, practical instances of this problem involve from a few hundreds up to one thousand optimization variables.

An approach that is often used in the automotive industry, attempts at approximating the dynamic constraints with "equivalent" static constraints, so that classical topology optimization methods, such as those implemented in the commercial software Altair/OptiStruct, can be applied on the resulting mathematical model. However, practitioners in the industry are not content with the results obtained thereby. A similar strategy is considered in [6, Section 2.1.2] where a criterion called *dynamic compliance* is defined in the context of forced vibrations. Otherwise, to our knowledge the only published work related to our problem can be found in [24] and in references therein, although the problem considered there is not the same: they search for solutions of low mass among those whose dynamic behavior fits well with that of a component from an older, well-proven vehicle (a target). More precisely, their objective is to minimize the deviation (in the least-square sense) of the magnitudes of the vibratory displacements at several frequencies from an ideal target for the vibratory spectrum. The obvious drawback of such an approach is that it gives little degree of freedom for designing new, original components.

In this paper we propose a practical topology optimization methodology to address structural optimization problems involving costly black-box *dynamic* constraints. In the specific automotive application that we shall consider, our objective will be to minimize the mass subject to dynamic constraints requiring that the magnitudes of the vibratory (forced frequency) displacements are below a threshold value for a given range of frequencies. Our approach combines tools from topology optimization with a state-of-the-art mathematical programming method. After a brief overview of topology optimization methods, we explain in the next section why the SIMP (Solid Isotropic Material with Penalization) topology optimization method emerges as only alternative to deal with our problem. Section 3 describes the various implementation strategies we chose to address our problem. This includes a new interpolation scheme for vibration problems that can

remove spurious modes in low-density areas, and it includes a method for handling problems with dynamic constraints over a range of forcing frequencies. We implement our methodology within a working framework of MSC/Nastran for the finite-element analyses, with the public-domain mathematical programming software FSQP [10] (Feasible Sequential Quadratic Programming) to drive the overall optimization process. We present in Section 4 some 3-D computational experiments on the design of a specific car component: an engine accessory support with both static and dynamic constraints. We draw conclusions in Section 5.

2 Topology Optimization

In this section, we give an an overview of various approaches to topology design, including an outline of the computational issues one has to consider for the different methods. We finally discuss why we had chosen a specific approach for the final implementation.

In common practice, structural optimization problems used to be solved by trial and error. First, a proposition of design of the component is evaluated by a structural mechanics software, and if it does not satisfy the constraints, the engineer proposes a modified design and so on. This way of doing is costly in time. It is nowadays often replaced by the use of iterative shape optimization software. In the automotive industry, shape optimization has been used since the 80's and became more popular in the 90's (see [8] for more details). However, as explained in [6], the main drawback of such methods is that the resulting design is highly dependent upon the initial geometry (the starting point). Indeed, through the iterations the successive shapes can only change in their frontiers, and their topology (number of components, number of holes, number of sides) must remain the same. Since the starting point is often a known solution (for instance, from a previous vehicle), shape optimization commonly yields expected, classical designs. Moreover, shape optimization methods can be very costly in computer time, since when the shape goes too far away from the initial shape, the structure requires a new meshing.

Topology optimization addresses the more general issue of finding a design without *a priori* on the number of components, the number of holes, or the number of sides of the part to be designed. This gives a potential for more diversified solutions. The automotive industry got interested in topology optimization at the very end of the 80's. However until the mid 90's, the examples found in the literature remained mostly theoretical [8]. The topology optimization problem is concerned with laying out material in an optimal manner. More formally, given a maximal structural domain $\Omega \subset I\!R^3$ and a material with density ρ^0 and properties A^0, the aim is to find a sub-domain $\omega \subseteq \Omega$ that will contain the material, solution of the optimization problem:

$$\begin{aligned} \min_{\omega \subseteq \Omega} \quad & f(\omega) \\ \text{subject to} \quad & g_i(\omega) \leq 0, \qquad i \in I, \\ & h_i(\omega) = 0, \qquad i \in E, \end{aligned}$$

where f is the objective function, g_i and h_i are (real-valued) constraint functions, and I and E are finite index sets. In typical instances, these functions are not explicit in ω and require calling a finite-element method in order to evaluate them.

Available topology optimization software include Altair/OptiStruct, which is used namely at Renault on various design problems. However, such software packages suffer from limitations of applicability. For instance, the current version of Altair/OptiStruct can only address optimization problems whose objective functions and constraint functions represent :

- the mass,
- the volume,
- the compliance,
- one of the eigenvalues,
- the displacement at a node, or
- a weighted combination of the previous functions.

As a consequence, it is not possible to deal with, for instance, the stress tensor or the dynamic constraints (forced response). These are essential in our context of vibratory and acoustic requirements.

We next describe very briefly the main topology optimization methods in order to see why we chose the SIMP method for the class of problems we consider in the current paper.

We divide the topology optimization methods into three types: evolutionary methods, shape-gradient based methods and material distribution methods.

2.1 Evolutionary Methods

Apart from their name, these methods have no link with the genetic or evolutionary methods that are well known in optimization (and to which she shall refer in the sequel). Evolutionary methods for topology optimization, which rely on *fully stressed design* [14], remove a small quantity of material in the parts of the domain where the stress tensor has low value (in practice, grid elements are removed). We refer the interested reader to [26] for more detail. These methods can be applied for general objective/constraint functions (see for instance [13] for mechanical components), but they are heuristics, which require a delicate tuning of parameters. Moreover, removing elements, which cannot be reintegrated to the structure, often yields wrong decisions.

2.2 Shape-Gradient Based Methods

Classical shape optimization methods are based on the notion of shape gradient, which monitors the sensitivity of the objective function upon the displacement of the frontier of the domain [23]. Two categories of topology optimization methods are based on the concept of shape derivative: level-set methods and topological-derivative based methods.

Level-set methods were introduced in [21]. A shape ω in the maximal structural domain $\Omega \subset I\!R^3$ is captured as the level set of a function ψ:

$$\begin{cases} \psi(x) = 0 \Longleftrightarrow x \in \partial\omega \bigcap \Omega, \\ \psi(x) < 0 \Longleftrightarrow x \in \omega, \\ \psi(x) > 0 \Longleftrightarrow x \in (\Omega \setminus \omega), \end{cases}$$

where $\partial\omega$ denotes the frontier of ω. An interesting application of level-set methods can be found in [2] where general objective functions can be taken into account when shape gradients are available. A drawback of level-set methods is that no new hole will be introduced during the optimization process. More importantly for our application context, the adjoint state must be available. Thus, level-set methods can only be applied on very specific types of objective and constraint functions.

Topological-derivative based methods were introduced in [9]. They use an extension of the shape gradient which states, for every point of the current structure, the appropriateness of introducing a hole of infinitesimal size. A drawback is that eliminated grid elements cannot be reintegrated during the optimization process. Again, here, the adjoint state must be available. As a consequence, topological-derivative based methods, like level-set methods, can only be applied on very specific types of objective and constraint functions.

2.3 Material distribution Methods

The methods that lay out material on a design grid subdivide further into three categories: discrete optimization methods, homogenization methods and the SIMP method.

Instead of using a sub-domain variable ω, **Discrete optimization methods** use a binary variable

$$\rho \in \{0, \rho_0\} \tag{1}$$

for each design-grid element (value ρ_0 if we put material, and zero otherwise), where ρ_0 is the density of the material we want to lay out. Typically, this formulation of the topology optimization problem is then solved using branch and bound, simulated annealing, or genetic algorithms (see [11, 12, 20] for applications in the automotive industry). However, due to the excessive computing time required for each objective/constraint function evaluation (finite-element analysis), the use of such approaches is limited to very coarse grids. It is clear that an optimal solution obtained on too coarse a grid will not contain any information usable for a design engineer.

Alternative methods relax each discrete constraint (1) into $0 \leq \rho \leq \rho_0$, which corresponds to considering a composite porous material. The two remaining categories of material distribution methods use such a relaxation. They differ in the way they then define the properties A of the material for densities between 0 and ρ^0.

Homogenization methods consider that a material with a density $0 < \rho < \rho^0$ is a composite material having a specific micro-structure. Two types of micro-structure are the perforated cells [4] and the sequential laminated materials [1]. The macroscopic properties of the material are then deduced from this specific predefined micro-structure. In

the case of sequential laminated materials, the properties A of the material for intermediate densities (ρ different from 0 or ρ_0) can be explicitly computed. A post-processing penalization step will then transform the continuous solution obtained after optimization into a solution without intermediate densities. For now, such methods are limited to linear elasticity and most theoretical results involve very specific objective/constraint functions such as the compliance and the mass.

We next describe the **SIMP method** (Simplified Isotropic Material with Penalization), which is due to Bendsøe [3] (see also further work in [19, 27]). It is simple to implement and it does not rely on restrictive assumptions on the form of the problem (contrary to the homogenization methods). It allows for instance to deal with multiple-objective and multidisciplinary problems in [6] (or in [28] for examples in the automotive industry), as well as multi-material problems in [5, 22]. The SIMP method avoids the delicate choice of a specific micro-structure. It rather uses a property matrix A that does not necessarily correspond to a realistic material: to any intermediate density $0 < \rho < \rho^0$, corresponds an arbitrary matrix A. More precisely, the following simple interpolation (heuristic) formulae are assumed to relate structural properties A_i and density ρ_i at the i^{th} grid element:

$$\rho_i = \mu_i \rho^0, \tag{2}$$
$$A_i = \mu_i^p A^0, \tag{3}$$

where $p \geq 1$ is a predefined constant, $0 \leq \mu_i \leq 1$ is the optimization variable, ρ^0 and A^0 are the density and the properties of the (original) material. The optimization variable μ_i takes value 1 if we put material at grid element i, and zero otherwise. A mathematical programming method (such as sequential quadratic programming, which we shall describe in Section 3) is then used to solve the topology optimization problem written under the following form:

$$\begin{array}{lll} \min\limits_{\mu} & f(\mu) & \\ \text{subject to} & g_i(\mu) \leq 0, & i \in I, \\ & h_i(\mu) = 0, & i \in E, \\ & 0 \leq \mu_i \leq 1, & i = 1, 2, \ldots N, \end{array}$$

where f , g_i and h_i are, again, some implicit objective/constraint functions requiring the use of a finite-element code, I and E are finite index sets, N is the number of design grid elements, and μ stands for the vector $(\mu_1, \mu_2, \ldots, \mu_N) \in I\!R^N$. Contrary to the homogenization methods, a post-processing penalization step is not required in the SIMP approach. Indeed, for the large class of topology optimization problems that are concerned with minimizing mass while maximizing stiffness, there will be fewer intermediate densities as the iterations of the mathematical programming method will proceed. This can be seen by considering Figure 1, which plots both mass and stiffness with respect to the density μ_i (for some $p > 1$). From equations (2) and (3), mass is proportional to μ_i, while stiffness is proportional to μ_i^p. Hence, any local mathematical programming method applied to the SIMP formulation of the topology optimization problem will attempt at pushing towards 1 the value of the μ_is that are close to 1 at the current iterate, since a

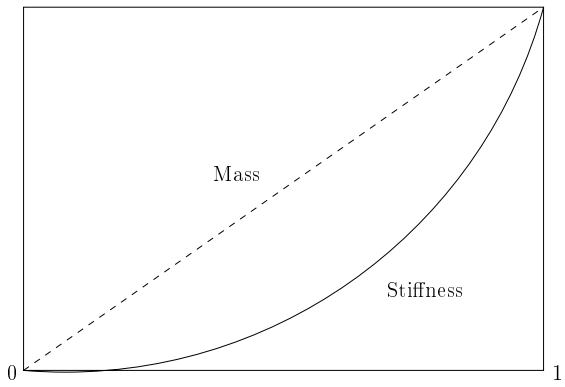

Figure 1: Mass and stiffness with respect to the density μ_i.

great increase of stiffness will then be obtained at the cost of a small increase of mass. Simultaneously, the grid elements with a density value μ_i close to 0 at the current iterate will be decreased towards 0, because a significant decrease of mass can be obtained at the cost of a negligible decrease of stiffness. Naturally, one can then expect that the solution obtained will be highly dependent upon the initial solution chosen. We shall discuss this matter in Section 4 as well as how the above-described implicit penalization is affected by the way the value of the parameter p is managed.

It is this latter approach, the SIMP method, that we chose to address our topology optimization problem. We gave indications as to why the other approaches are not viable in our context of application. The most critical reason in favor of the SIMP method is the very general nature of the constraints it can address. Recall that we consider in this paper a practical application involving dynamic (acoustic and vibratory) constraints, whose evaluation requires the resolution of a computation-intensive finite-element analysis performed by a black-box commercial software.

3 Implementation Strategies

In this section we detail our practical topology optimization methodology to address structural optimization problems involving costly black-box dynamic constraints. More precisely, we first discuss in Section 3.1 the choice of a mathematical programming method to drive the optimization process. We then give in Section 3.2 indications as to how the penalty parameter p must be managed when using the SIMP approach. We introduce in Section 3.3 an original way to deal with dynamic constraints. Finally, we propose a new type of fictive material for the SIMP method, and we describe its advantages in Section 3.4.

3.1 Mathematical Programming Driver: SQP

We need a mathematical programming method to drive the optimization process.

Note first that we must reject *a priori* derivative-free methods (see [18] for a complete survey) due to the (relatively) high dimension of our problem. More importantly, we *do*

obtain, at no extra cost, derivatives each time we call MSC/Nastran in order to evaluate our objective and constraint functions. It would be a waste to ignore this sensitivity information when trying to improve the current structure.

Typically in structural optimization, the type of mathematical programming methods used are of the type *sequential convex programming* (e.g. sequential linear programming, or the method of moving asymptotes [25]). At each iteration, sequential convex programming methods solve a separable convex subproblem. This resolution can take place in a low-dimensional search space using duality, provided that few constraints are involved in the structural optimization problem under consideration. However, in our costly-evaluation context, this computational saving is not worth using such a poor approximation at each iteration (based on first-order information and on simplistic separate functions). We prefer relying on the richer quadratic approximation of sequential programming programming (SQP) type methods, which are the current state of the art in mathematical programming (together with interior-point algorithms) [16]. Note also that methods such as the method of moving asymptotes require an intensive work of parameter tuning, which is not the case for typical SQP algorithms.

We choose a specific class of SQP algorithms, called *feasible SQP* methods. All iterates generated by such methods satisfy the constraints of the optimization problem. This is an appropriate feature in practical situations because:

- it is not desirable to evaluate costly functions outside the feasible set,
- the objective function may well not be defined at all outside the feasible set,
- should the algorithm stop prematurely, the current iterate will be a feasible solution for the problem.

The specific feasible SQP method we use in order to perform our numerical experiments is the public-domain software FSQP [15] developed by Tits *et al.* in [17].

3.2 Tuning the value of the Penalty Parameter p

In this section, we study the impact of the choice of the value of the parameter p in the SIMP method. Due to the excessive CPU time required to evaluate the constraint functions in our industrial problem, we first performed numerous computational experiments on a toy (academic) problem: a simple cantilever. We report only our conclusions here, the detail of these experiments (performed both in the static and in the dynamic case) can be found in [7].

As expected with the SIMP method, the number of undesirable intermediate densities (i.e. different from zero or one) decreases as we increase the value of the penalty parameter p. On the other hand, the quality of the solution obtained decreases as we increase p: with too high a value of p (e.g. $p > 4$), the optimization process converges too rapidly towards an uninteresting nearby local minimum. Our aim is to reach some compromise between two conflicting objectives: minimizing mass and minimizing the number of intermediate densities.

Our computational experiments showed that for $p > 1$, we obtain different (locally optimal) solutions, depending on the initial solution chosen. Implementing a continuation

method, a schedule of slowly increasing values of p, as advised in [6], did not prove to yield significative improvement in our context of dynamic constraints.

Following our preliminary numerical experiments, we choose the compromise value $p = 3.5$, and kept it constant throughout the optimization process.

An interesting observation from the above preliminary experiments is that there were always more intermediate densities near $\mu_i = 1$ than near $\mu_i = 0$. We shall see in Section 3.4 that this is due to the specific type of fictive material involved in the SIMP approach, and we shall propose an alternative fictive material model (a new interpolation scheme).

3.3 Modeling Dynamic Constraints

We now describe how we handle the dynamic constraints. We are required to obtain a topology design which has its vibratory response (the measured displacement following certain excitation frequencies) below a given reference value, say γ, over a range of forcing frequencies. In practice, we want these dynamic constraints to be satisfied at every discretized step of the frequency range. For each proposition of topology design, we must run a finite-element analysis, performed typically by a black-box structural mechanics software such as MSC/Nastran. This latter gives punctual values at every discretized step of the frequency range. Formally, the dynamic constraints stipulate that

$$u_{d,k}(\mu_k, f) \leq \gamma_{d,k}, \qquad d = 1, 2, 3;\ k \in K;\ \text{ for all } f \in [f_{min}, f_{max}], \tag{4}$$

where:

$u_{d,k}$	magnitude of the displacement at node k in the d^{th} direction of space,
μ_k	material density at node k,
$\gamma_{d,k}$	maximal value imposed at node k in the d^{th} direction of space,
K	index set of nodes,
$[f_{min}, f_{max}]$	frequency range of interest.

If for a certain frequency f, the computed magnitude of displacement is above the threshold $\gamma_{d,k}$, then the proposed design is not acceptable.

How can we implement such a constraint so that it can be taken into account by a mathematical programming method? Part of the difficulty is that we want to impose that the displacements, over the *whole frequency range*, be below γ. However, in practice when we call MSC/Nastran we only obtain values at the *discretized* steps of the frequency range. We must therefore rewrite the dynamic constraints under the discretized form:

$$u_{d,k}(\mu_k, f) \leq \gamma_{d,k}, \qquad d = 1, 2, 3;\ k \in K;\ f = f_1, f_2, \ldots, f_{N_f}, \tag{5}$$

where N_f is the number of discretized frequencies used to cover the frequency range $[f_{min}, f_{max}]$. Such a large number of constraints has proved to be a handicap for mathematical programming methods such as active-set based methods (which include SQP-type methods). One way to get round this problem is to replace the multiple constraints (5) with a single constraint:

$$\max_{f \in \{f_1, f_2, \ldots, f_{N_f}\}} u_{d,k}(\mu_k, f) \leq \gamma_{d,k}, \qquad d = 1, 2, 3;\ k \in K. \tag{6}$$

However, an important drawback of this equivalent way of writing (5) is that the mathematical programming method would have to deal with a non-differentiable constraint. As a consequence, we rather consider the alternative aggregated l_q-norm constraint:

$$\Big(\sum_{i=1}^{N_f} u_k(\mu_k, f_i)^q\Big)^{1/q} \leq \gamma_{d,k}, \tag{7}$$

where q is an integer such that $q \geq 2$. Constraint (7) is differentiable. It approximates well constraint (6) if q is set to a sufficiently large integer. Based on computational experiments on our academic test problem (cantilever), we set empirically this value to the compromise value $q = 6$ (smaller values of q did not allow constraint (7) to model correctly our dynamic constraints (5), whereas we want to avoid approaching non-differentiability numerically which would result from too large a value of q).

Further computational experiments performed on the simple cantilever test problem revealed another difficulty related to the discretization of the dynamic constraints. Clearly, the finer the discretization step is, the better will be the discrete approximation of the dynamic constraints (4) (at a higher computational cost nevertheless). However, even if we use a very fine discretization step, there is always a phenomenon of hidden constraint. For instance, the displacement measurements at the design found by our optimization procedure are everywhere below the prescribed threshold $\gamma_{d,k}$ when using a given frequency discretization step, say of 25 Hz, but when we refine the discretization step to, say 2 Hz and call Nastran to evaluate the same design, we observe that constraint (4) is violated for some frequency f (hidden peaks in between the coarse discretized steps), which means that the design is in fact not acceptable. Here is how we overcome this numerical difficulty: we impose a stricter constraint, using an empirical threshold value γ smaller than the required one. We shall use this strategy in the numerical results reported in Section 4, avoiding thereby relying on excessively fine discretizations which would yield prohibitive CPU time spent in MSC/Nastran calls.

3.4 A New Type of Fictive Material

In this section we introduce an alternative penalty function for the SIMP method. In other words, we propose to replace the traditional heuristic formulae (2) and (3) used in SIMP, with a new interpolation scheme. Function

$$F_p(x) := x^p \tag{8}$$

is at the basis of the standard SIMP interpolation scheme. Some difficulties encountered while solving our topological optimization problem with dynamical constraints seem to be caused by particular features of F_p. One of these difficulties is related to the spurious modes which are observed in low-density areas. Such local modes cause undue increase in computation time and memory space requirements for Nastran and makes it difficult to identify relevant modes. As explained in [6], local eigenfrequencies are found in areas where the mass/stiffness ratio is very large, that is to say for the SIMP method at grid elements i such that the density μ_i is near zero. Indeed, for such grid elements, the

mass/stiffness ratio (which is equal to $\frac{\mu_i}{\mu_i^p}$) tends towards infinity as μ_i tends towards zero. These undesirable local frequencies are therefore a consequence of the fact that holes in the structure are modeled with soft (i.e. low-density) material, in order to avoid removal of the corresponding grid elements as the optimization iterations proceed. A solution to this problem involves keeping the mass/stiffness ratio finite when the density decreases.

A second problem that we observed with function F_p from our preliminary numerical experiments on our academic problem, is the fact that during the optimization process, null or very low densities are never increased. Close examination of Figure 1 shows why. Let us consider a grid element where the density is low (μ_i close to zero). Then, a small increase of the density μ_i will have a negligible impact on stiffness (μ_i^p remains close to zero) although a significant contribution to the mass (the curve $y = x$ has a positive slope). Thus, such an increase of density is unfavorable and unlikely to be undertaken by the optimization algorithm. The problem is related to the null derivative of the function F_p at $x = 0$. Note that the analogue does not occur for densities near one, where the derivative of F_p is finite. This explains why, among the intermediate densities, the majority of densities are near zero rather than near one in the various tests we performed.

In order to overcome both problems mentioned above (spurious modes and no filling of holes), we propose replacing the standard penalization function F_p with

$$F_\alpha(x) := \frac{1}{\alpha}\Big(x + (\alpha - 1)x^{\alpha+1}\Big)$$

in the interpolation scheme (3). In other words, equation (3) is replaced with:

$$A_i = \frac{1}{\alpha}\Big(\mu_i + (\alpha - 1)\mu_i^{\alpha+1}\Big)A^0. \tag{9}$$

The new penalization function F_α has a behavior at $x = 0$ which is "symmetrical" to that at $x = 1$: $F'_\alpha(0) = \frac{1}{\alpha}$ and $F'_\alpha(1) = \alpha$. Figure 2 displays plots of F_α and F_p. Let us consider again our academic test problem in order to illustrate computationally the advantages of the new function F_α over F_p. The dimension of the cantilever is 100 mm x 100 mm x 10 mm, the density of the material is $\rho^0 = 7800$ kg/m^3. We restrict our tests to the simple static case. We apply a force F = 1000 N at nodes 1 and 2, as displayed on Figure 3, and we require the corresponding displacements to be below 0.0128 mm. We use as design grid: 20 x 20 x 1 and an identical computing grid. We choose as starting point for the optimization: all 400 densities equal to 0.9, and we keep $p = 3.5$ for the standard SIMP interpolation scheme based on the function F_p. On the frequency range 0–500 Hz, the solution obtained reveals 798 local modes (between 0 and 2 Hz) and 12 global modes. Figure 4 displays the first local mode and the last global mode. Our computational experiments with the alternative F_α-based interpolation scheme give comparable results on the point of view of the design obtained. However, this time there is no local mode even for solutions with intermediate densities. In this case the ratio mass/stiffness does not tend towards infinity (but rather towards $\frac{1}{\alpha}$) as the density tends towards zero. Another positive effect of using F_α revealed by our numerical tests, is that the densities which are initialized at zero can effectively be increased as the optimization

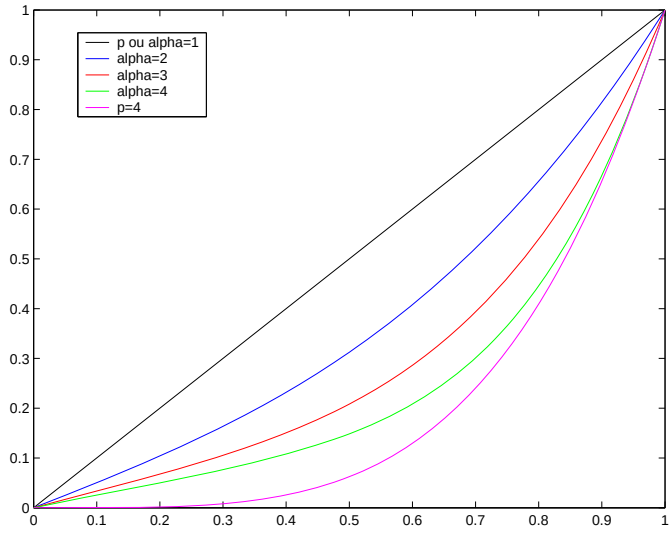

Figure 2: From left to right: F_α for $\alpha = 1, 2, 3, 4$ and F_p for $p = 4$.

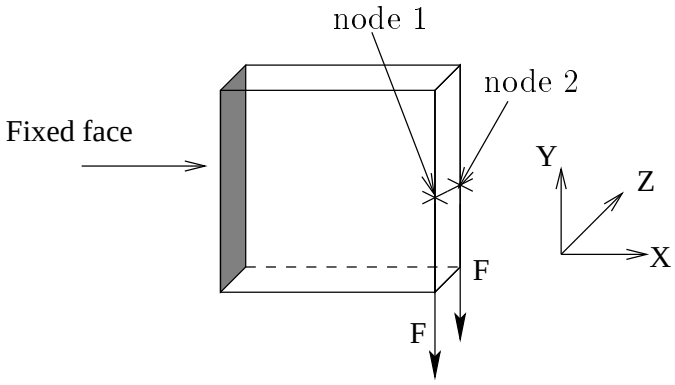

Figure 3: Cantilever toy problem.

proceeds (and even for larger values of the penalty parameter α), contrary to the case where one uses the standard F_p function. In the latter case, the zero-density elements (or the elements with density nearly zero) remain constant throughout the optimization iterations when the penalty parameter p is set to $p > 1$.

To conclude this section, let us simply mention that we first consider the following penalty function as an alternative to F_p :

$$F_\lambda(x) := \frac{f_\lambda(x) + g_\lambda(x) + |f_\lambda(x) - g_\lambda(x)|}{2},$$

where

$$\begin{aligned} f_\lambda(x) &:= 1 + \frac{2}{(3+\lambda)(1-\lambda)}\Big(\lambda - \sqrt{3 - 2\lambda - (3+\lambda)(1-\lambda)x}\,\Big), \\ \text{and} \quad g_\lambda(x) &:= \frac{(3+\lambda)(1-\lambda)x^2}{4} + x\lambda. \end{aligned}$$

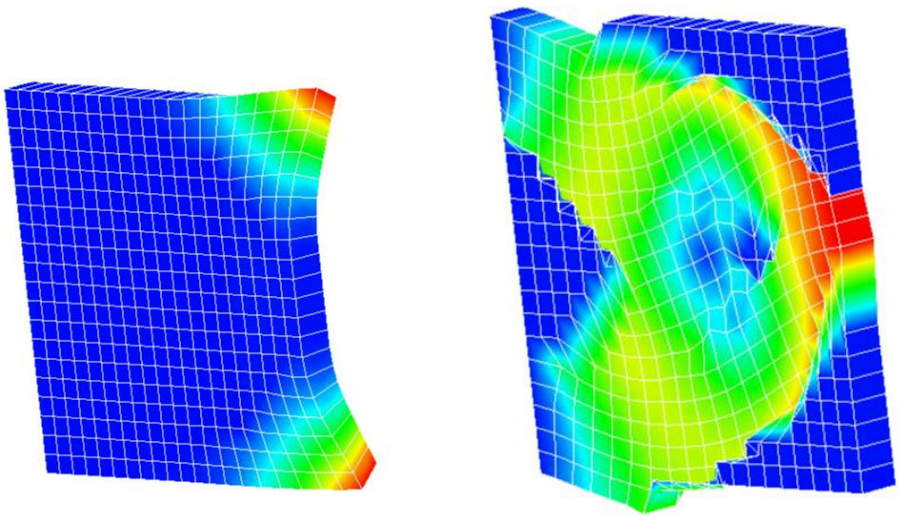

Figure 4: Local mode (left) and global mode (right).

This function not only displays symmetry at $x = 0$ and at $x = 1$, it is also completely symmetric with respect to the line of equation $y = 1 - x$, as shown on Figure 5. As a consequence, it considers material and absence of material in a completely symmetrical

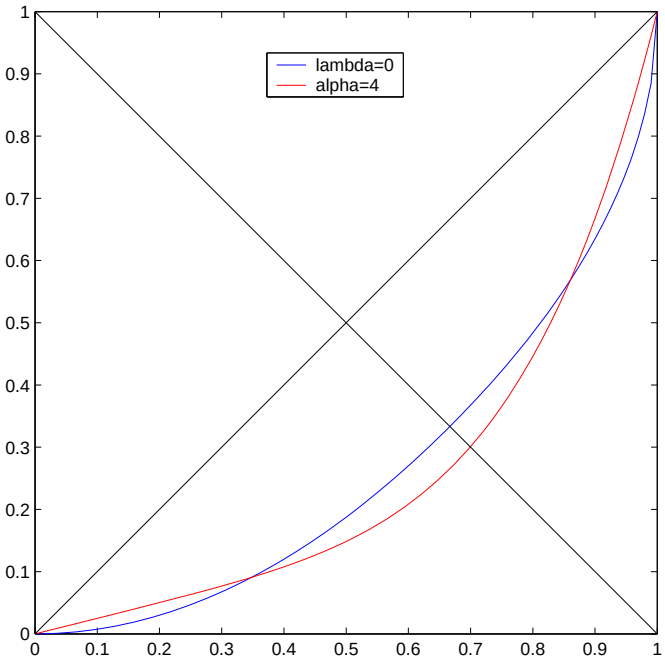

Figure 5: F_λ (with $\lambda = 0$) and F_α (with $\alpha = 4$).

manner (permuting the convention "0 for void" and "1 for material" does not affect

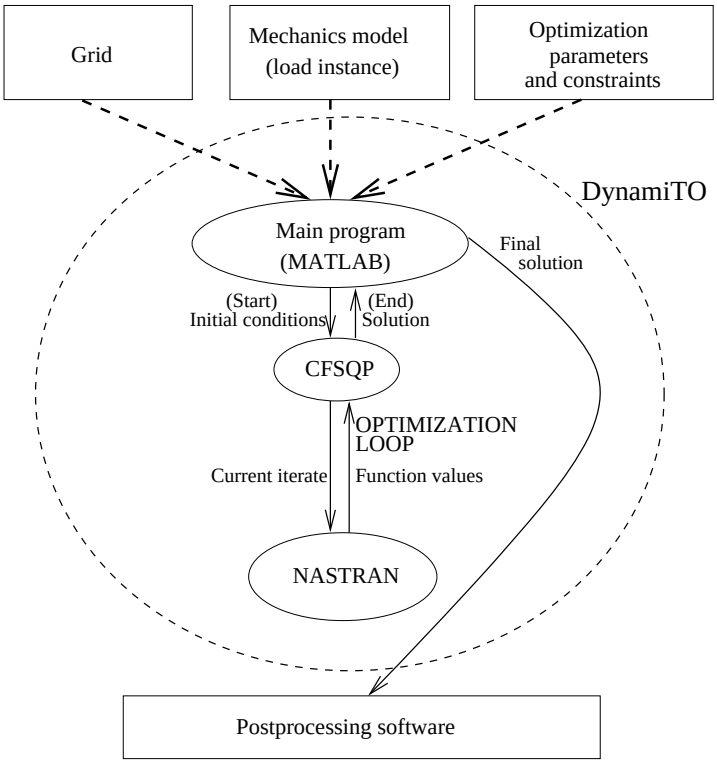

Figure 6: Overall architecture of DynamiTO

results). However, programming language restrictions within Nastran prevented us from testing the F_λ-based interpolation scheme.

4 Computational Experiments

We present in this section some 3-D computational experiments on an industrial problem.

Let us first review our overall computational methodology, which we named DynamiTO. Figure 6 displayed the coupling of the various elements involved. DynamiTO is a MATLAB program which couples Nastran (finite-element analyses: evaluation of the objective functions and the constraints) and the public-domain mathematical programming software CFSQP (written in C) to drive the overall optimization process. The SIMP method is only involved in the calls to Nastran in order to evaluate functions (objective and constraints). Recall that the SIMP method determines the material properties (Young modulus) in each design grid element from the corresponding density, following equation (3) (or equation (9), when using our new interpolating scheme). The user-provided input data include:

- a .dat Nastran file specifying the load instance, the type of analysis, the definition of the optimization functions (objective and constraints) and the grid (elements and nodes);

- the parameter values required in the optimization, i.e. a starting point, constraint bounds, the value of the SIMP penalty parameter p (or α, when using our new interpolating scheme);
- the mechanics constraints, i.e. the nodes on which displacements are measured, the corresponding directions, and when dynamic constraints are involved: the frequency range and discretization step.

Based on various tests, we choose the Armijo (monotone) line search option of FSQP. Also, in order to avoid extremely slow progress in the line searches of FSQP, we need to scale the various (objective and constraint) function and gradient values at every iterate $\mu^{(k)}$ as follows :

$$\begin{aligned} f(\mu^{(k)}) &:= f(\mu^{(k)}) \times \frac{1}{||\nabla f(\mu^{(0)})||_\infty}, \\ \nabla f(\mu^{(k)}) &:= \nabla f(\mu^{(k)}) \times \frac{1}{||\nabla f(\mu^{(0)})||_\infty}, \\ g_i(\mu^{(k)}) &:= g_i(\mu^{(k)}) \times \frac{1}{||\nabla g_i(\mu^{(0)})||_\infty}, \\ \text{and } \nabla g_i(\mu^{(k)}) &:= \nabla g_i(\mu^{(k)}) \times \frac{1}{||\nabla g_i(\mu^{(0)})||_\infty}, \end{aligned}$$

where f is the objective function, g is the i^{th} constraint function, and $\mu^{(0)}$ denotes the initial value for the density (at iteration 0). Finally, in order to avoid the numerous and costly iterations yielding negligible objective-function progress, we tighten the stopping criterion of FSQP. The algorithm now stops as soon as

$$||f(\mu^{(k+1)}) - f(\mu^{(k)})|| < 10^{-15},$$

i.e. when two consecutive objective-function values differ by less than 10^{-15}, even if the norm of the gradient of the Lagrangian is not as small as required by the original FSQP.

We now present 3-D computational experiments on the design of a specific car component: an engine accessory support. It is a light car component (around 2 kg) designed to support the compressor and the alternator, each of which weighs around 6 kg. Stiffness and resistance of the engine accessory support are therefore crucial. An example of engine accessory support is showed on Figure 7. Accessories are modeled with masses. Consequently, every mass value given in the sequel includes the mass of the accessories. For mechanical computation purposes, we aggregate into two master nodes (which we shall call Point 1 and Point 2) the different grid points where the accessories are fastened. The grid design and the analysis grid are the same, except for some "non-design" elements (shown in grey on Figure 7) whose densities remains constantly at 1 throughout the optimization process. The number of design elements is 1354. The starting point is

$$\mu_i^{(0)} = 0.9, \quad i = 1, \ldots, 1354.$$

We use the standard SIMP penalty function F_p, and we keep the penalty parameter constant at the value $p = 3.5$.

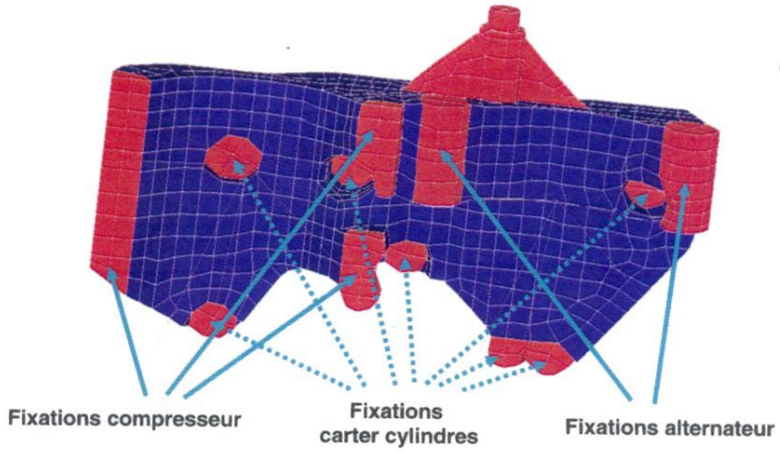

Figure 7: An engine accessory support and the design grid.

First, we perform a preliminary test by considering only static constraints. We shall come back afterwards to the real problem, which does involve dynamic constraints. The only interest of this test is to confirm that our methodology is validated by comparison with the results obtained using the commercial software Altair/OptiStruct. A downwards force of 1 Newton is applied at Points 1 and 2 and we impose that the vertical displacement be less than 1.25×10^{-5} mm. The solution obtained by DynamiTO (Design N°1) and OptiStruct (Design N°2) are displayed on Figure 8. The designs are almost iden-

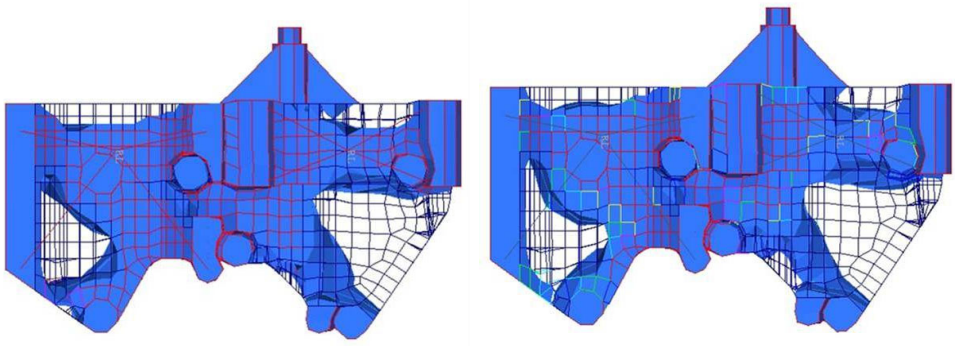

Figure 8: Design N°1 obtained by DynamiTO (left) and Design N°2 by OptiStruct (right), both under static constraints only.

tical and the respective masses are similar (13.01 kg for DynamiTO and 12.99 kg for Altair/OptiStruct). Both solutions satisfy the imposed static constraints.

However, the real problem of interest in our vibro-acoustic application context involves dynamic constraints. Software such as Altair/OptiStruct cannot deal with such problems. Let us now consider the design problem with dynamic constraints imposed on the master nodes (Points 1 and 2) and resulting from forcing frequencies in the three directions of

space:

$$\max_{f\in[f_{min},f_{max}]} u_{3,1}(\mu, f) \leq \gamma, \tag{10}$$

$$\max_{f\in[f_{min},f_{max}]} u_{3,2}(\mu, f) \leq \gamma. \tag{11}$$

For this test we use a force of 1 Newton and a tolerance $\gamma = 1.2 \times 10^{-3}$ mm. Since the initial solution (all $\mu_i = 0.9$) reveals a fundamental mode at 353 Hz, we restrict the frequency range to the interval [0 Hz , 500 Hz] with a discretization step of 10 Hz. Remark that Designs N°1 and N°2 (Figure 8) are not satisfactory because they violate the dynamic constraints (10) and (11). As explained in Section 3.3, we model these constraints using the l_6-norm of the sampled collection of displacements:

$$\Big(\sum_i u_{3,P}(\mu, f_i)^6\Big)^{1/6} \leq \gamma, \quad P = 1, 2, \tag{12}$$

where the summation runs over all discretized frequencies f_i considered (here every 10 Hz). For this problem, each single Nastran analysis (one evaluation for the optimization process) requires around two hours of CPU time on a Sun Ultra 10 station.

The main features of the optimal design found by our DynamiTO methodology (Design N°3) are displayed on the first line of Table 1. This solution does satisfy the dynamic

Design N°	Penalty function	γ $(10^{-3}mm)$	Mass (kg)	Eval.	Iter.	Intermediate densities	Constraints satisfied?
3	F_p	1.2	12.833	379	46	1330	no
4	F_p	1.1	12.789	171	33	450	no
5	F_p	1	13.973	128	37	22	yes
6	F_α	1	13.987	187	51	54	yes

Table 1: Tests using different right-hand side tolerances γ for the right-hand side of the dynamic constraints for the optimization

constraint (12) with a discretization step of 10 Hz. However, when analyzing it with a finer discretization step for the frequencies, namely 2 Hz instead of 10 Hz, we observe that Design N°3 does not satisfy the dynamic constraints (10) and (11). As shown on Figure 9, where we evaluate the displacements at Points 1 and 2 at every 2 Hz, there is a peak that goes beyond the tolerance $\gamma = 1.2\times10^{-3}$ mm. This violation was hidden during the optimization process, which relied on a 10 Hz discretization step for constraint (12). As mentioned in Section 3.3, using a finer discretization during the optimization process only (greatly) increase the computational cost, without preventing such violating peaks to appear at even finer discretizations. We therefore apply the simple strategy we described in Section 3.3. We successively run tests using stricter tolerances values for γ. We obtain a satisfactorily design when using $\gamma = 10^{-3}$ mm (Design N°5). Its features are displayed on Table 1. We observe on Figure 10 that Design N°5 violates the artificially stricter constraint ($\gamma = 10^{-3}$ mm) but satisfies the required dynamic constraint ($\gamma = 1.2 \times 10^{-3}$ mm). Moreover, Design N°5 has the advantage of having fewer intermediate densities

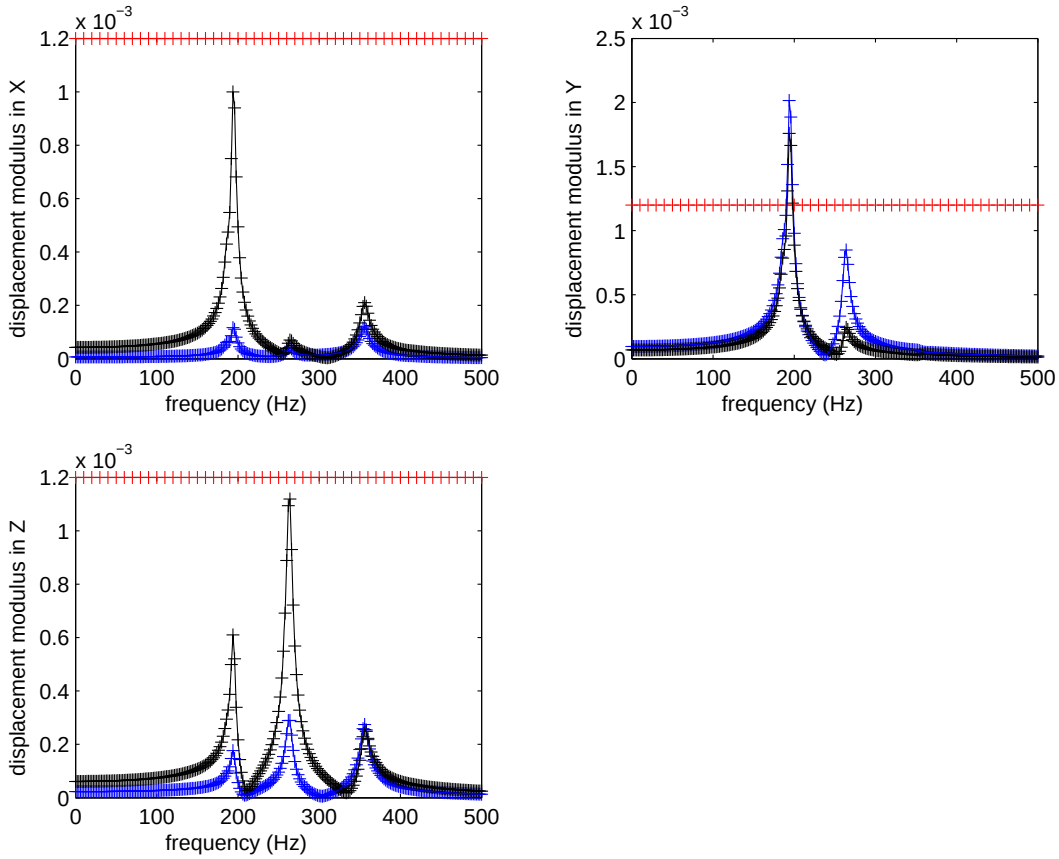

Figure 9: Dynamic constraints for Design N°3 (optimization using $\gamma = 1.2 \times 10^{-3}$).

than previous designs. Figure 11 displays Design N°5 and Design N°6. The latter was also obtained with our DynamiTO methodology, with the same settings except for the fact that we replaced the standard SIMP interpolating scheme F_p with the penalty function F_α introduced in Section 3.2. Design N°6 is very similar to Design N°5. In the case of the specific engine accessory support application, the use of this new interpolating scheme yields no benefit. The difficulties mentioned in Section 3.4 and linked to spurious modes are not relevant here. Indeed, in the present application, displacements are only monitored at elements where the density is not near zero, since the force is put at points where the accessories are fastened (where the density remains equal to 1).

5 Conclusion

We surveyed in this paper the computational methods for topology optimization in view of solving a difficult structural optimization problem involving computation-intensive dynamic constraints. We explained why the SIMP topology optimization method was the only viable approach, and we introduced a practical methodology for addressing our problem. Our approach relies on a state-of-the-art method from mathematical pro-

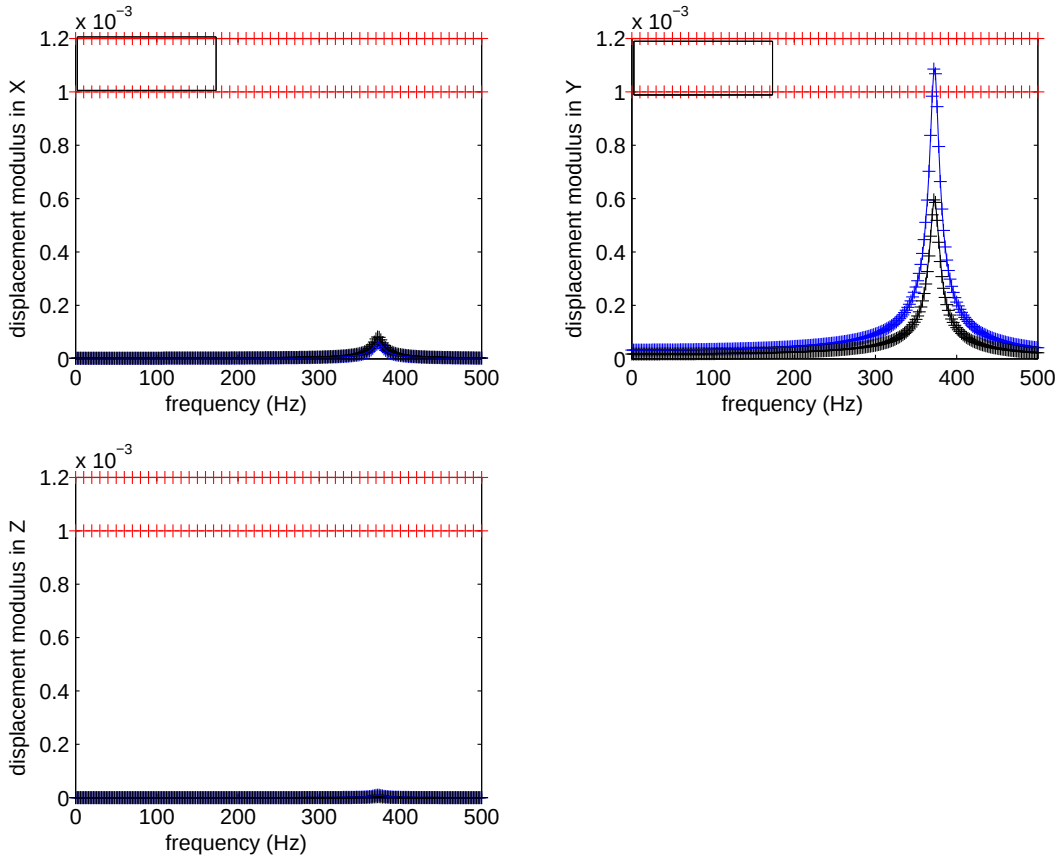

Figure 10: Dynamic constraints for Design N°5 (optimization using the stricter constraint $\gamma = 10^{-3}$).

gramming (sequential quadratic programming) and on the SIMP topology optimization method. We described the implementation of our methodology using the commercial software MSC/Nastran for the finite-element analysis and the public-domain mathematical programming code FSQP to drive the overall optimization process. We presented a simple strategy consisting in using stricter tolerances for the dynamic constraints allowed to avoid designs violating the dynamic constraints between discretized frequencies. We demonstrated the viability of our approach with encouraging 3-D results obtained on a real-world mechanical component at Renault where it is intended to be used as a decision-analysis tool for designing new, original components.

We also introduced in this paper a new material density interpolation scheme for removing non-physical modes in low-density regions and which is simpler than other methods proposed in the literature. Although the superiority of the interpolation scheme introduced could not be demonstrated on the specific automotive application we presented, numerical tests performed on an academic test problem suggest that the new penalty function does remove spurious modes in low-density areas. Moreover, these preliminary tests suggest that the penalty function we introduced allows for more degree of freedom in the search space, as it allows fill in. Future work should attempt at exploiting this feature

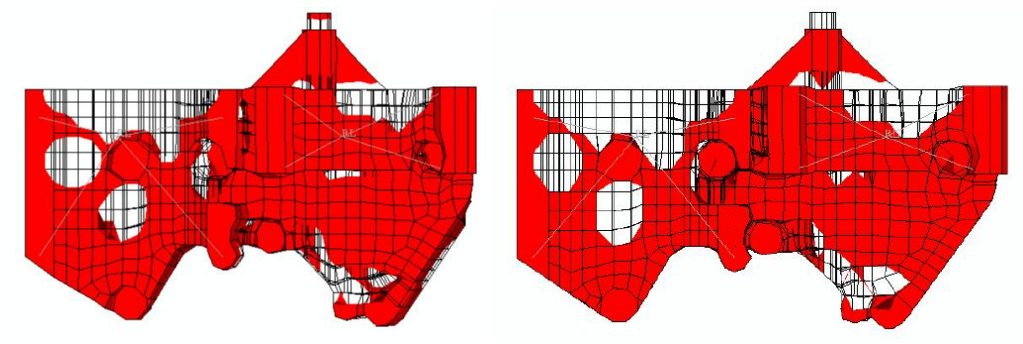

Figure 11: Design N°5 obtained using F_p (left), and Design N°6 obtained using F_α (right).

as a diversification strategy for topology optimization problems for which algorithms tend to get trapped into uninteresting local optima.

References

[1] G. Allaire, Z. Belhachmi, and F. Jouve. The homogenisation method for topology and shape optimization. Single and multiple loads case. *Revue européenne des éléments finis*, 5:649–672, 1996.

[2] G. Allaire, F. Jouve, and A. M. Toader. A level-set method for shape optimization. *Compte rendu de l'Acadmie des Sciences de Paris*, 334:1–6, 2002.

[3] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.

[4] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, 71:197–224, 1988.

[5] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, 69:635–654, 1999.

[6] M. P. Bendsøe and O. Sigmund. *Topology optimization. Theory, methods and applications*. Springer Verlag, 2003.

[7] S. Calvel. *Conception d'organes automobiles par optimisation topologique*. PhD thesis, Institut de Mathématiques, Université Paul Sabatier, Toulouse, France, October 2004. Available from: `http://tel.ccsd.cnrs.fr/documents/archives0/00/00/71/96/`.

[8] C. J. Chen and M. Usman. Design optimization for automotive applications. *International journal of vehicule design*, 25:126–141, 2001.

[9] H. A. Eschenauer, V. V. Kobolev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8:42–51, 1994.

[10] FSQP. `http://www.aemdesign.com/downloadfsqp.htm`.

[11] H. Hamda, F. Jouve, E. Lutton, M. Schoenauer, and M. Sebag. Compact unstructured representations for evolutionary design. *Applied intelligence*, 16:139–155, 2002.

[12] H. Hamda, I. C. Parmee, O. Roudenko, and M. Schoenauer. Multi-objective evolutionary topological optimum design. In *Fifth international conference on adaptative computing design and manufacture*, volume 5, pages 121–132, 2002.

[13] L. Harzheim and G. Graf. Optimization of engineering components with the SKO method. *SAE technical paper series*, 951104, 1995.

[14] E. Hinton and J. Sienz. Fully stressed topological design of structures using an evolutionary procedure. *Engineering computations*, 12:229–244, 1995.

[15] C. T. Lawrence, J. L. Zhou, and A. L. Tits. User's guide for CFSQP version 2.5. Technical report, Electrical Engineering Department and Institute for Systems Research, University of Maryland, 1997.

[16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[17] E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical programming*, 59:261–276, 1993.

[18] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 1998.

[19] G. I. N. Rozvany, M. Zhou, and O. Sigmund. Topology optimization in structural design. In H. Adeli, editor, *Advances in design optimization*, chapter 10, pages 340–399. Chapman and Hall, 1994.

[20] E. Sandgren and E. Jensen. Automotive structural design employing a genetic optimization algorithm. *SAE technical paper series*, 920772, 1992.

[21] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of computational physics*, 163(2):489–528, 2000.

[22] O. Sigmund. Topology optimization of multi-material, multi-physics structures. In *Third World Congress of Structural and Multidisciplinary Optimization (WCSMO), Niagara Falls, USA, May 1999*, 2000.

[23] J. Sokolowski and J. Zolesio. Introduction to shape optimization: Shape sensitivity analysis. In *Series in computational mathematics*, volume 10. Springer, 1992.

[24] C. A. Soto. Structural topology optimization for tactile response improvement in the automotive industry. In R.-J. Yand and D. E. Smith, editors, *AMD-Vol. 227, Design optimization with applications in industry*, pages 37–48. ASME, 1997.

[25] K. Svanberg. The method of moving asymptotes. A new method for structural optimization. *International journal for numerical method in engineering*, 24:359–373, 1987.

[26] Y. M. Xie and G. P. Steven. *Evolutionary structural optimisation*. Springer Verlag, 1997.

[27] R. J. Yang and C. H. Chuang. Optimal topology design using linear programming. *Computers and structures*, 52(2):265–276, 1994.

[28] R. J. Yang, T. J. Walsh, and P. A. Schilke. A general formulation for topology optimization. *SAE technical paper series*, 942256, 1994.