

# A sensitivity-based one-parameter-at-a-time model updating method

A. Batou<sup>\*a,b</sup>

<sup>a</sup>*Université Paris-Est, Laboratoire Modélisation et Simulation Multi Echelle,  
MSME UMR 8208 CNRS, France*

<sup>b</sup>*School of Engineering, The University of Liverpool, Liverpool, United Kingdom*

---

## Abstract

This paper is interested in model updating problems which consists in identifying optimal values of model parameters by comparing the model outputs with the experimental outputs. Such a problem generally yields a challenging multivariate inverse problem to be solved in high dimension. The high-dimensionality requires the use of a global optimization algorithm in order to explore efficiently the parameters space. In this paper we propose an alternative algorithm which allows each model parameters to be identified separately and sequentially by solving separated univariate inverse problems. For each parameter, a devoted inverse problem is introduced by identifying an output which is sensitive to this parameter only, the sensitivity being quantified using Sobol indices. The proposed method is illustrated through a three-storey structure for which experimental measurements are collected.

*Keywords:* Model Updating, Sobol Indices, Sensitivity Indices, Inverse problems )

---

## 1. Introduction

In the context of model validation, the model updating methods consist in calibrating a computational model in order to reduce the distance between the experimental outputs and the outputs provided by the computational model [7]. The model updating methods of computational models have been

---

\*Corresponding author

*Email address:* [anas.batou@liverpool.ac.uk](mailto:anas.batou@liverpool.ac.uk) (A. Batou\*)

intensively studied during the last four decades. In structural dynamics, efficient methods have been proposed (see for instance [10, 18, 7]) and are now commonly used in industry. There are two main types of methods: the direct methods (see for instance [1, 8, 14, 19]) which consist in directly modifying the stiffness and mass matrices and the indirect methods which consists in updating some physical parameters (see for instance [4, 5, 9]) of the model. The latter method can be described in three steps: (1) The first one consists in constructing a computational model for which the parameters are set to nominal values. These parameters can be related to material properties (Young's modulus, mass density, and so on), geometry (CAD, thickness, area moments of inertia, and so on) and boundary conditions. (2) The second step consists in selecting the parameters to be updated. This step generally relies on engineering expertise combined with a sensitivity analysis of the quantities of interest with respect to the candidate parameters to be updated (see for instance [3, 6, 11]). In [20], for updating stochastic models, the authors have performed the model selection by defining a composite sensitivity index that discriminates explicitly between sets of parameters with correctly-modelled and erroneous statistics. (3) The third step consists in updating the most-sensitive parameters using experimental data. In general, this step is carried out by reducing a "distance" between the experimental outputs and the corresponding outputs computed using the computational model.

The number of most-sensitive parameters is often very large and the updating of those later generally requires the solving of a non-convex inverse problem in high dimension. This high dimension prohibits the use of grid search methods and global optimization methods are required such as genetic algorithm, particle swarm optimization or simulated annealing [12]. These classes of methods require the fitting of solver parameters in order to be efficient and enable the escape from local minima.

The objective of the new approach proposed in the present paper consists in addressing the problem of the high dimension in model updating method by replacing the original inverse problem by a sequence of small dimension (ideally dimension one) inverse problems which can be solved using a grid search approach. To achieve this objective the proposed approach consists in transforming the observed outputs of the model into a set of new outputs. Each new output is sensitive to only one parameter to be updated and is not sensitive to the other parameters. Then by comparing the new outputs of the model to the corresponding experimental ones, it is possible to define a

sequence of separated inverse problems in order to update all the parameters one at a time and thus it is possible to use a grid search algorithm solver for each one-dimensional inverse problem.

In Section 2, the new model updating method is presented. In this section, first the construction of the new outputs is presented, then the model updating method using these outputs is presented. The approach is illustrated through simple examples. In Section 3 the updating method is validated experimentally through a 3-storey structure for which experimental data has been collected.

## 2. Model updating algorithm

### 2.1. Output transformation

In this paper we consider a computational model for which the output vector  $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^m$  is a function of the vector  $\mathbf{x} \in \mathcal{C}_x \subset \mathbb{R}^n$  of the model parameters to be updated. It is assumed that a corresponding experimental output  $\mathbf{y}^{\text{exp}}$  has been collected. At this stage, for readability, it is assumed that the fidelity of model is high and therefore, the experimental response can be predicted by the computational model if the true values of the model parameters are used. The presence of modelling errors will be addressed in Section 2.5. A classical way to identify the model parameter vector  $\mathbf{x}$  consists in finding the value  $\mathbf{x}^{\text{opt}}$  of the vector  $\mathbf{x}$  which makes the model output  $\mathbf{y}(\mathbf{x}^{\text{opt}})$  as close as possible to the experimental output  $\mathbf{y}^{\text{exp}}$ . The distance between the two outputs can be measured using the mean square distance  $\| \mathbf{y}(\mathbf{x}^{\text{opt}}) - \mathbf{y}^{\text{exp}} \|$  for instance, i.e.,

$$\mathbf{x}^{\text{opt}} = \arg \min_{\mathbf{x} \in \mathcal{C}_x} \| \mathbf{y}(\mathbf{x}) - \mathbf{y}^{\text{exp}} \| . \quad (1)$$

The solving of the multivariate inverse problem (1) generally requires the use of a global optimization algorithm such as the simulated annealing algorithm, genetic algorithms or particle swarm optimization, in order to explore the parameter space efficiently [12]. The efficiency of these methods depends on the correct tuning of the algorithm parameters which are problem dependent. A bad tuning of these parameters may yield to a slow convergence of the algorithm and a possible trap in a local minimum. The difficulty of exploring efficiently the parameter space increases with the number of parameter

The method proposed in this paper proceeds differently. For each parameter  $\{x_i, i = 1, \dots, n\}$ , the output  $\mathbf{y}(\mathbf{x})$  is transformed into a new output

$z_i(\mathbf{x})$  with values in  $\mathbb{R}$ :

$$z_i(\mathbf{x}) = g_i(\mathbf{y}(\mathbf{x})), \quad (2)$$

which is sensitive to the parameter  $x_i$  only and thus is not sensitive to the other model parameters. Furthermore it is assumed that this transform is a linear combination of  $n_g$  basis functions  $\{f_j, j = 1, \dots, n_g\}$ , i.e,

$$z_i(\mathbf{x}) = \boldsymbol{\alpha}_i^T \mathbf{f}(\mathbf{y}(\mathbf{x})), \quad (3)$$

where  $\mathbf{f} = (f_1, \dots, f_{n_g})$  is the vector of the basis functions and  $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \dots, \alpha_{i,n_g})$  is the vector of the coordinates in this basis. In the particular case where  $\mathbf{f}(\mathbf{y}(\mathbf{x})) = \mathbf{y}(\mathbf{x})$ , the transform  $g_i$  is just a linear combination of the outputs of the model.

## 2.2. Calculation of the output parameters

For each model parameter  $x_i$ , the transform  $g_i$  and thus the vector  $\boldsymbol{\alpha}_i$  need to be calculated. Since the objective is that each output  $z_i(\mathbf{x})$  is devoted to the identification of the model parameter  $x_i$  only, we calculate the vector  $\boldsymbol{\alpha}_i$  such that the output  $z_i(\mathbf{x})$  is sensitive to the model parameter  $x_i$  only. By replacing the model parameter vector  $\mathbf{x}$  by the random vector  $\mathbf{X}$ , Sobol indices [15, 13] enable the total variance of each random variable  $X_i$  to be quantified. The advantage of these total-effect sensitivity indices is that they are not local and thus do not require to set values for the other parameters. For each output  $Z_i = z_i(\mathbf{X})$  and for each random variable  $X_j$ , the total-effect Sobol index  $\tilde{S}_{i,j}$  is defined by

$$\tilde{S}_{i,j} = \frac{E_{\mathbf{X}_{\sim j}} (\text{Var}_{X_j}(Z_i | \mathbf{X}_{\sim j}))}{\text{Var}(Z_i)}, \quad (4)$$

where  $\mathbf{X}_{\sim j}$  is a random vector for which the components are copies of all the components of  $\mathbf{X}$  except  $X_j$ . Then using Eq.(3) yields

$$\tilde{S}_{i,j} = \frac{\boldsymbol{\alpha}_i^T [E_{\mathbf{X}_{\sim j}} (\text{Cov}_{X_j}(\mathbf{F} | \mathbf{X}_{\sim j}))] \boldsymbol{\alpha}_i}{\boldsymbol{\alpha}_i^T [\text{Cov}(\mathbf{F})] \boldsymbol{\alpha}_i}, \quad (5)$$

with  $\mathbf{F} = \mathbf{f}(\mathbf{y}(\mathbf{X}))$ . Denoting  $[C] = [\text{Cov}(\mathbf{F})]$  and  $[C^j] = [E_{\mathbf{X}_{\sim j}}(\text{Cov}_{X_j}(\mathbf{F} | \mathbf{X}_{\sim j}))]$ , the later equation is then rewritten

$$\tilde{S}_{i,j}(\boldsymbol{\alpha}_i) = \frac{\boldsymbol{\alpha}_i^T [C^j] \boldsymbol{\alpha}_i}{\boldsymbol{\alpha}_i^T [C] \boldsymbol{\alpha}_i}. \quad (6)$$

We can see in Eq.(6) that if matrices  $[C]$  and  $[C^j]$  are estimated (see [13] for the estimation of Sobol indices) then the Sobol index  $\tilde{S}_{i,j}(\boldsymbol{\alpha}_i)$  can easily be calculated for any value of the vector  $\boldsymbol{\alpha}_i$ . This will greatly facilitate the calculation of  $\boldsymbol{\alpha}_i$ .

As explained before, for each parameter  $x_i$ , the vector  $\boldsymbol{\alpha}_i$  defining the corresponding output  $z_i(\mathbf{x})$  has to be calculated such that this output is sensitive to  $x_i$  only. It can easily be shown that the value  $\boldsymbol{\alpha}_i^{\text{opt}}$  of  $\boldsymbol{\alpha}_i$  satisfying this condition is such that

$$\begin{aligned} \tilde{S}_{i,i}(\boldsymbol{\alpha}_i^{\text{opt}}) &= 1, \\ \tilde{S}_{i,j}(\boldsymbol{\alpha}_i^{\text{opt}}) &= 0, \quad \text{for } j \neq i. \end{aligned} \quad (7)$$

Solving Eq.(7) is equivalent to solving the set of equations

$$\begin{aligned} S_{i,i}(\boldsymbol{\alpha}_i^{\text{opt}}) &= 1, \\ S_{i,j}(\boldsymbol{\alpha}_i^{\text{opt}}) &= 0, \quad \text{for } j \neq i. \end{aligned} \quad (8)$$

with,

$$S_{i,j}(\boldsymbol{\alpha}_i) = \boldsymbol{\alpha}_i^T [C^j] \boldsymbol{\alpha}_i. \quad (9)$$

Indeed if Eq.(8) is satisfied then  $Z_i = z_i(\mathbf{X})$  is only sensitive to  $X_i$  and then  $(\boldsymbol{\alpha}_i^{\text{opt}})^T [C^i] \boldsymbol{\alpha}_i^{\text{opt}} = (\boldsymbol{\alpha}_i^{\text{opt}})^T [C] \boldsymbol{\alpha}_i^{\text{opt}}$ , i.e.  $\tilde{S}_{i,i}(\boldsymbol{\alpha}_i^{\text{opt}}) = 1$ . The solution of Eq.(8) can be calculated by solving the following mean square problem,

$$\boldsymbol{\alpha}_i^{\text{opt}} = \arg \min_{\boldsymbol{\alpha}_i \in \mathbb{R}^{n_g}} \Gamma(\boldsymbol{\alpha}_i). \quad (10)$$

where

$$\Gamma(\boldsymbol{\alpha}_i) = (S_{i,i}(\boldsymbol{\alpha}_i) - 1)^2 + \sum_{i \neq j} (S_{i,j}(\boldsymbol{\alpha}_i))^2 + \lambda \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i. \quad (11)$$

The last term in the right-hand side of Eq.(11) is a Tikhonov regularization

term ([17]). The gradient vector  $\nabla\Gamma(\boldsymbol{\alpha}_i)$  and the Hessian matrix  $[H(\boldsymbol{\alpha}_i)]$  of function  $\Gamma(\boldsymbol{\alpha}_i)$  are written as

$$\nabla\Gamma(\boldsymbol{\alpha}_i) = 4(S_{i,i}(\boldsymbol{\alpha}_i) - 1)[C^i]\boldsymbol{\alpha}_i + 4 \sum_{i \neq j} \{S_{i,j}(\boldsymbol{\alpha}_i)[C^j]\boldsymbol{\alpha}_i\} + 2\lambda\boldsymbol{\alpha}_i, \quad (12)$$

$$\begin{aligned} [H(\boldsymbol{\alpha}_i)] &= 4(S_{i,i}(\boldsymbol{\alpha}_i) - 1)[C^i] + 8[C^i]\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T[C^i]^T \\ &+ 4 \sum_{i \neq j} \{S_{i,j}(\boldsymbol{\alpha}_i)[C^j] + 2([C^j]\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T[C^j]^T)\} + 2\lambda[I_n]. \end{aligned} \quad (13)$$

Analysing the Hessian matrix  $[H(\boldsymbol{\alpha}_i)]$ , it can be deduced that since  $S_{i,i}(\boldsymbol{\alpha}_i) \leq 1$ , the Hessian matrix is not always positive definite unless  $\lambda$  is sufficiently large. Nevertheless, in the neighbourhood of the solution,  $S_{i,i}$  is close to one and therefore, the Hessian matrix is locally positive definite and thus function  $\Gamma(\boldsymbol{\alpha}_i)$  becomes convex when approaching the solution. The regularization parameter  $\lambda$  in Eq.(11) has to be chosen sufficiently large to guaranty the convexity of  $\Gamma(\boldsymbol{\alpha}_i)$  and sufficiently small to limit the impact of the regularization term. The optimization problem defined by Eqs. (10) and (11) can be solved using any minimization algorithm. Here we suggest to solve it using the Newton iterative method written as

$$\boldsymbol{\alpha}_i^{i+1} = \boldsymbol{\alpha}_i^i - [H(\boldsymbol{\alpha}_i^i)]^{-1} \nabla\Gamma(\boldsymbol{\alpha}_i^i). \quad (14)$$

In practice, the parameter  $\lambda$  is first set to zero. Then, if the minimization algorithm (14) does not converge, this value is increased until the algorithm converges. Let  $[S]$  with entries  $S_{i,j}$  be the sensitivity matrix between the model parameters  $x_j$  and the outputs  $z_i$ . In the ideal case, after calculation of  $\boldsymbol{\alpha}_1^{\text{opt}}, \dots, \boldsymbol{\alpha}_n^{\text{opt}}$ , matrix  $[S]$  should be an identity matrix. If not, it means that the inputs are not identifiable separately and another function basis should be used. For instance, if a polynomial basis is chosen. Then the basis can be enriched with higher degree polynomials until  $[S]$  is an identity matrix. But it should be noted that increasing the size of the basis increases the size of matrices  $[C], [C^1], \dots, [C^m]$ . Then if the size of the basis is too large, the calculation of  $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n$  becomes computationally expensive.

### 2.3. Examples

Before introducing the model updating method in the next section, some examples are presented here in order to illustrate the construction of the

transformed outputs. In the first example,  $\mathbf{x} = (x_1, x_2, x_3) \in [1, 2]^3$  and  $\mathbf{y}(\mathbf{x}) = (x_1, x_2 + x_3, 4x_1, x_3, 3)$ . The basis functions are such that  $\mathbf{f}(\mathbf{y}(\mathbf{x})) = \mathbf{y}(\mathbf{x})$ , i.e.  $z_i(\mathbf{x}) = \boldsymbol{\alpha}_i^T \mathbf{y}(\mathbf{x})$  for  $i = 1, 2, 3$ . Therefore for each parameter  $x_i$ , the transformed output will be a linear combination of the components of the output vector  $\mathbf{y}(\mathbf{x})$ . The previously described algorithm is run with  $\lambda = 5 \times 10^{-10}$  and 40,000 Monte Carlo samples for the computation of the Sobol index matrices. The random variables  $X_i$  have uniform distribution in  $[1, 2]^3$ . The initial values of  $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$  are generated randomly. The following solutions are obtained:

$$\begin{aligned}\boldsymbol{\alpha}_1 &= (0.2, 0.0, 0.8, 0.0, 0.0), \\ \boldsymbol{\alpha}_2 &= (0.0, 3.4, 0.0, -3.4, 0.0), \\ \boldsymbol{\alpha}_3 &= (0.0, 0.0, 0.0, 3.4, 0.0),\end{aligned}\tag{15}$$

which correspond to the exact inverse reconstructions of  $x_1, x_2, x_3$  respectively. For this case, the sensitivity matrix  $[S]$  is exactly identity.

In the second example,  $\mathbf{x} = (x_1, x_2, x_3) \in [-0.4, 0.4]^3$  and  $\mathbf{y}(\mathbf{x}) = (\cos(x_1 + x_2), \sin(x_2), x_1 x_2, x_3, 3)$ . In a first case, the basis functions are, as the previous example, such that  $\mathbf{f}(\mathbf{y}(\mathbf{x})) = \mathbf{y}(\mathbf{x})$ , i.e.  $z_i(\mathbf{x}) = \boldsymbol{\alpha}_i^T \mathbf{y}(\mathbf{x})$  for  $i = 1 \dots, 3$ . Again we use  $\lambda = 5 \times 10^{-10}$  and 40,000 Monte Carlo samples for the computation of the Sobol index and the initial values of  $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$  are generated randomly. The following solutions are obtained:

$$\begin{aligned}\boldsymbol{\alpha}_1 &= (33.0, 0.0, 84.2, 0.0, 0.0), \\ \boldsymbol{\alpha}_2 &= (0.0, 8.7, 0.0, 0.0, 0.0), \\ \boldsymbol{\alpha}_3 &= (0.0, 0.0, 0.0, 8.7, 0.0).\end{aligned}\tag{16}$$

For this case, the sensitivity matrix  $[S]$  is

$$[S] = \begin{pmatrix} 0.5 & 0.5 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}.\tag{17}$$

As we can see in the first row of  $[S]$ , the algorithm has failed to construct an output sensitive to  $x_1$  only, which can be easily understood regarding the function  $\mathbf{y}(\mathbf{x})$ . For the second case, the function basis is enriched using the squares of the model parameters in the combination, i.e.  $\mathbf{f}(\mathbf{y}(\mathbf{x})) =$

$(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x})^2)$ . In this case, the new solutions are

$$\begin{aligned}\boldsymbol{\alpha}_1 &= (36.6, 0.0, 163.3, 0.0, 0.0, 68.1, 82.2, -5.8, 0.0, 0.0), \\ \boldsymbol{\alpha}_2 &= (0.0, 8.7, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0), \\ \boldsymbol{\alpha}_3 &= (0.0, 0.0, 0.0, 8.7, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0).\end{aligned}\tag{18}$$

For this case, the matrix  $[S]$  is

$$[S] = \begin{pmatrix} 1.0 & 0.006 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix},\tag{19}$$

therefore  $[S]$  is almost identity. The obtained result can be explained using quadratic approximations of cosine and sine functions which are present in the function  $\mathbf{y}(\mathbf{x})$ . For more complex functions more enrichments may be necessary.

#### 2.4. Model updating

Once the vectors  $\boldsymbol{\alpha}_1^{\text{opt}}, \dots, \boldsymbol{\alpha}_n^{\text{opt}}$  are calculated, then the matrix  $[S]$  with entries  $S_{i,j}$  can be estimated. Each entry  $S_{i,j}$  represents how much the output  $z_i$  (defined by vector  $\boldsymbol{\alpha}_i$ ) is sensitive to the model parameter  $x_j$ . Ideally, this matrix should be the identity matrix, meaning that for each model parameter  $x_i$ , the function  $z_i$  (constructed using the optimal value  $\boldsymbol{\alpha}_i^{\text{opt}}$ ) is exclusively sensitive to  $x_i$ . In practice, this rarely happens (even if the basis is enriched) and, for each input  $x_i$ , the discrepancy with respect to this ideal case can be quantified by the value  $\Gamma(\boldsymbol{\alpha}_i^{\text{opt}})$ . If the value  $\Gamma(\boldsymbol{\alpha}_i^{\text{opt}})$  is sufficiently small then the model parameter  $x_i$  can be identified independently of the other model parameters as the solution of the mean square problem,

$$x_i^{\text{opt}} = \arg \min_{x_i \in \mathcal{C}_{x_i}} \| \boldsymbol{\alpha}_i^T \mathbf{f}(\mathbf{y}(x_1, \dots, x_i, \dots, x_n)) - \boldsymbol{\alpha}_i^T \mathbf{f}(\mathbf{y}^{\text{exp}}) \|^2,\tag{20}$$

in which  $\boldsymbol{\alpha}_i^T \mathbf{f}(\mathbf{y}^{\text{exp}})$  is the transformed experimental output and where the parameters  $\{x_j, j \neq i\}$  can have arbitrary values. We can then define a sequential algorithm for the identification of the model parameters:

- 1 - Rank the model parameters with respect to the values  $\Gamma(\boldsymbol{\alpha}_i^{\text{opt}})$ . The indices of the ranked model parameters are denoted  $r_1, \dots, r_n$ .
- 2 - Keep only the  $n_\epsilon$  model parameters  $x_{r_1}, \dots, x_{r_{n_\epsilon}}$  for which  $\Gamma(\boldsymbol{\alpha}_{r_i}^{\text{opt}}) < \epsilon$ ,  $\epsilon$  being a threshold to be defined.



- 3 - Initialize the model parameters vector  $\mathbf{x} = \mathbf{x}_0$ .
- 4 - Identify  $x_{r_1}^{\text{opt}}$  solving Eq.(20) (using any univariate inverse problem solver) and replace  $x_{0,r_1}$  by  $x_{r_1}^{\text{opt}}$ .
- 5 - Identify iteratively  $x_{r_2}, \dots, x_{r_{n_\epsilon}}$  with the same method.

Once all the model parameters  $x_{r_1}, \dots, x_{r_{n_\epsilon}}$  are identified, we can recalculate the vectors  $\boldsymbol{\alpha}_{r_{n_\epsilon+1}}, \dots, \boldsymbol{\alpha}_{r_n}$  of the remaining parameters (if any) and proceed with the same algorithm, the other model parameters being prescribed to their identified values. The algorithm has to be repeated until all the model parameters are identified or until all the  $\Gamma(\boldsymbol{\alpha}_{r_i}^{\text{opt}}) > \epsilon$  which would mean that the remaining model parameters can't be identified separately.

The complete model updating method is described in Algorithm 1.

---

**Algorithm 1** Model updating algorithm

---

```

while  $n > 0$  do
    Estimate  $[C], [C^1], \dots, [C^n]$ 
    Calculate  $\boldsymbol{\alpha}_1^{\text{opt}}, \dots, \boldsymbol{\alpha}_n^{\text{opt}}$  using Eq.(14)
    Rank the model parameters such that  $\Gamma(\boldsymbol{\alpha}_{r_1}^{\text{opt}}) < \dots < \Gamma(\boldsymbol{\alpha}_{r_n}^{\text{opt}})$ 
    Keep only the  $n_\epsilon$  model parameters for which  $\Gamma(\boldsymbol{\alpha}_{r_i}^{\text{opt}}) < \epsilon$ 
    Initialize the model parameters vector  $\mathbf{x}^{\text{opt}}$ 
    for  $\ell = 1, \dots, n_\epsilon$  do
        Identify  $x_{r_\ell}^{\text{opt}}$  solving Eq.(20) (using any univariate inverse problem solver)
        Update the  $r_\ell^{\text{th}}$  component of  $\mathbf{x}^{\text{opt}}$ 
    end
     $\mathbf{x} \leftarrow (x_{n_\epsilon+1}, \dots, x_{n_\epsilon+1})$ 
     $n \leftarrow n - n_\epsilon$ 
end

```

---

### 2.5. Remarks

#### *i - Other measures of dependence*

In the present paper, to construct outputs which depend only on one parameter, sensitivity indices have been used. Instead we could have used the mutual information (which is a measure of dependence between two random variables) between model parameters and the transformed output. Unfortunately this approach would require the estimation of probability distributions in high dimension for the calculation of the parameters  $\boldsymbol{\alpha}_i^{\text{opt}}$  and would not

enable to separate  $\alpha_i$  as it was the case using sensitivity indices (see Eq.(6)). Another possibility would consist in weakening the independence using correlation between the system parameters and the output instead of the mutual information. This method is numerically very efficient but provides, as expected, good results only when the systems parameters and the outputs have approximatively Gaussian probability distributions (in which case correlation and dependence are equivalent).

*ii - Computational cost*

In the method presented here, the computational cost is mainly devoted to the estimation of matrices  $[C], [C^1], \dots, [C^n]$  using the Monte Carlo simulation method. This estimation is performed before the calculation of  $\alpha_1^{\text{opt}}, \dots, \alpha_n^{\text{opt}}$ . The other steps (calculation of  $\alpha_1^{\text{opt}}, \dots, \alpha_n^{\text{opt}}$  and identification of  $\mathbf{x}^{\text{opt}}$ ) are negligible.

*iii - Modelling errors*

If modelling errors are present due, for instance, to the simplicity of the model or measurement noise, then they can be taken into account using a probabilistic approach by introducing a random vector  $\mathbf{N}$  at an advanced stage of the calculation. There are several ways to do this. For instance, an additive random vector  $\mathbf{N}$  can be included directly in the output of the model:

$$\bar{\mathbf{y}}(\mathbf{x}) = \mathbf{y}(\mathbf{x}) + \mathbf{N}. \quad (21)$$

There are other ways to take into account uncertainties induced by modelling error. For instance, in structural dynamic, the nonparametric probabilistic approach [16] uses a random matrix approach to take into account these uncertainties. In this case, random vector  $\mathbf{N}$  contains the entries of the random matrices. For all the cases, the matrix  $[C^j] = [E_{\mathbf{X}_{\sim j}}(\text{Cov}_{X_j}(\mathbf{F} | \mathbf{X}_{\sim j}))]$  is replaced by

$$[\bar{C}^j] = [E_{\mathbf{X}_{\sim j}}(\text{Cov}_{X_j, \mathbf{N}}(\mathbf{F} | \mathbf{X}_{\sim j}))]. \quad (22)$$

Then the rest of the method is kept unchanged. When introducing modelling errors, the variance of the output increases, making the search of a sensitivity matrix  $[S]$  equal to identity difficult. Therefore, the algorithm for calculating the vectors  $\alpha_i$  will automatically discard the components of the outputs that



Figure 1: 3-storey structure.

are highly impacted by modelling errors, making the updating method more robust to modelling errors.

### 3. Experimental application

In this section, the methodology presented in this paper is applied to the model updating of a 3-storey structure equipped with accelerometers.

#### 3.1. Test rig

The real structure is made up with 4 aluminium blocks connected each with the others with thin aluminium plates (see Fig. 1). The blocks have length 0.2 m, width 0.15 m and thickness 0.02 m. The thin plates have length 0.15 m, thickness 0.0012 m and width 0.015 m. The lowest block is clamped to the ground. The three other ones are equipped with accelerometers. The top block is excited with a hammer shock along  $x$ -direction. The measured Frequency Response Functions FRFs of the acceleration across  $x$ -direction are plotted in Fig. 2. As expected, we can see three peaks corresponding to the three global modes of the structure.

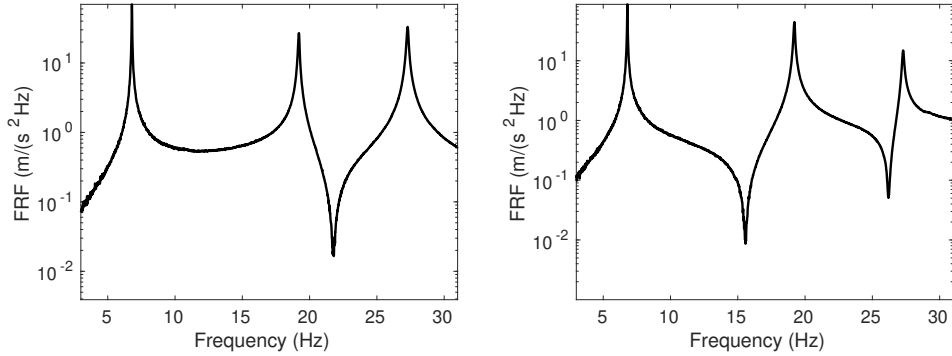


Figure 2: Experimental FRF in acceleration for storey 1 (left) and storey 2 (right).

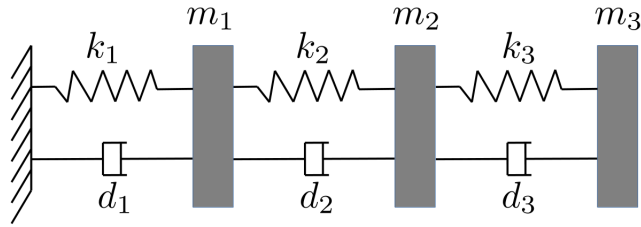


Figure 3: Three masses model.

### 3.2. Model

The torsional and axial vibrations of the structure and the local vibration of the blocks and the plates are neglected. Therefore the structure is modelled as three masses connected by springs and dampers as shown in Fig. 3. The masses of the plates are  $m_1 = m_2 = m_3 = 1.63$  kg. The equivalent stiffnesses of the springs (representing the bending stiffness) are  $k_1 = k_2 = k_3 = 1.35 \times 10^4$  N/m. The damping ratio associated with the viscous dampers ( $\xi_i = d_i / (2\sqrt{m_i k_i})$ ,  $i = 1, 2, 3$ ) are  $\xi_1 = \xi_2 = \xi_3 = 1.4 \times 10^{-3}$ . The FRFs of the model (before updating) is compared to the experimental ones in Fig. 4. It can be seen in this figure large discrepancies in all the frequency band of analysis. To reduce this discrepancies, all the parameters will be updated in the next section.

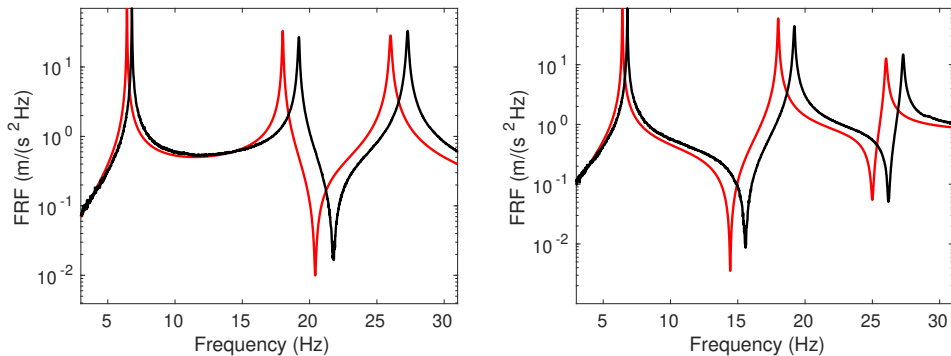


Figure 4: Comparison between the experimental and numerical FRF in acceleration for storey 1 (left) and storey 2 (right).

### 3.3. Model updating

The model parameters that are used to update the model are: the three masses, the three stiffnesses and the three damping ratios. For each of these 9 parameters, a multiplicative updating coefficient is introduced. The prior output corresponds to the concatenation of the logarithm of the modulus of the FRFs shown in Fig. 4. A random noise is included in the model corresponding to independent and identically distributed centred Gaussian random variables with standard deviation 0.19. To construct the transformed outputs for the updating, we use  $\lambda = 1 \times 10^{-5}$  and 2,000 Monte Carlo samples for the computation of the Sobol index matrices. The common logarithm ( $\log_{10}$ ) of the multiplicative updating parameters have uniform distributions in  $[-0.1, 0.1]$ . This parametrization enables the multiplicative parameters to vary between 0.1 and 10. The initial values of  $\alpha_i$ ,  $i = 1, \dots, 9$  are generated randomly. The so-obtained sensitivity matrix  $[S]$  is represented in Fig. 5. We can see in this figure that only the first six parameters (the three masses and the three stiffnesses) can be identified separately from the other parameters. We have tried to improve the sensitivity matrix by enriching the functional basis with quadratic and cubic functions. Nevertheless no noticeable improvement has been observed. Therefore the identification of the three other parameters (damping ratios) will require another iteration for which the results will be presented later. As described in Algorithm 1, the three masses and the three stiffnesses are then identified separately using the new outputs and a grid search for each parameter. The new values for these parameters are  $m_1 = 1.47$  kg,  $m_2 = 1.52$  kg,  $m_3 = 1.54$  kg,  $k_1 =$

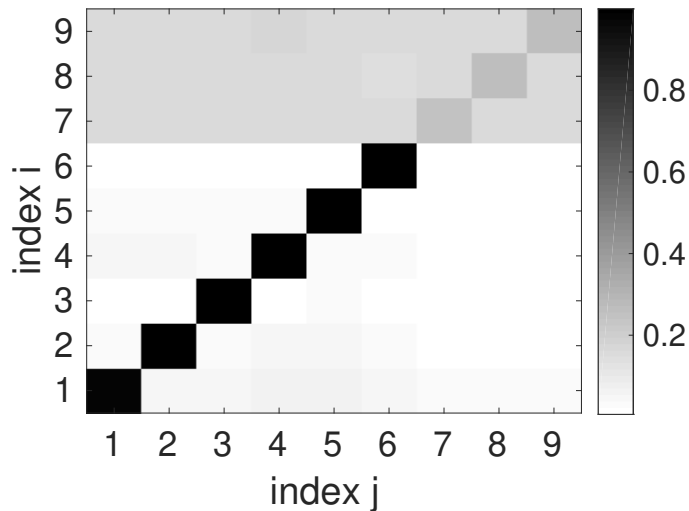


Figure 5: Sensitivity matrix  $[S]$ .

$1.41 \times 10^4$  N/m,  $k_2 = 1.30 \times 10^4$  N/m,  $k_3 = 1.41 \times 10^4$  N/m. The corresponding new FRFs are plotted in Fig. 6. Compared to the initial results plotted in Fig. 2, we can see that the updating of the masses and stiffnesses has drastically reduced the discrepancies between the experimental FRFs and the numerical one.

As explained before the damping ratios could not be identified in the previous iteration. Therefore, these parameters are calibrated in a new iteration by recalculating the vectors  $\alpha_i$ , with these parameters only. The value for the other parameters are fixed to the ones identified in the previous iteration. The sensitivity matrix  $[S]$  is represented in Fig. 7. The obtained matrix is not perfectly diagonal. But the error is sufficiently low to identify the damping ratios separately. The new values of the damping ratios are  $\xi_1 = 4.5 \times 10^{-3}$ ,  $\xi_2 = 5.1 \times 10^{-4}$  and  $\xi_3 = 2.1 \times 10^{-3}$ . The corresponding new FRFs are plotted in Fig. 8. Compared to Fig. 6, we can see a slight improvement concerning the amplitude of the three resonance peaks.

#### 4. Conclusions

In this paper, a new model updating method has been presented. This method enables the updating of each model parameter separately by constructing, for each parameter, a transform of the measured output which is

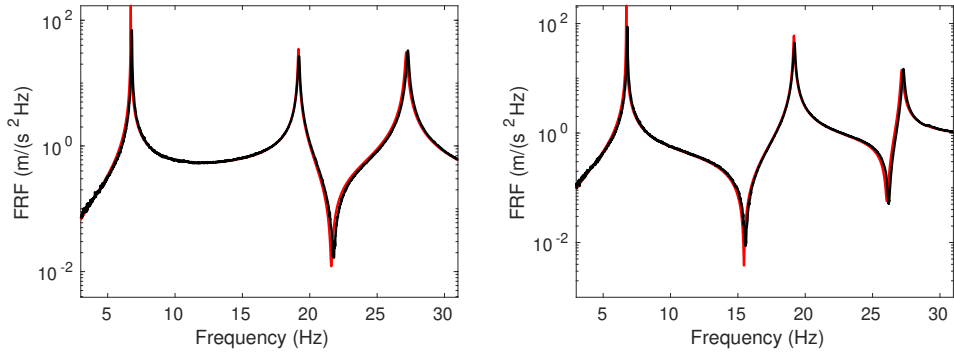


Figure 6: Comparison between the experimental and numerical FRF in acceleration for storey 1 (left) and storey 2 (right), after the first updating iteration (6 parameters only).

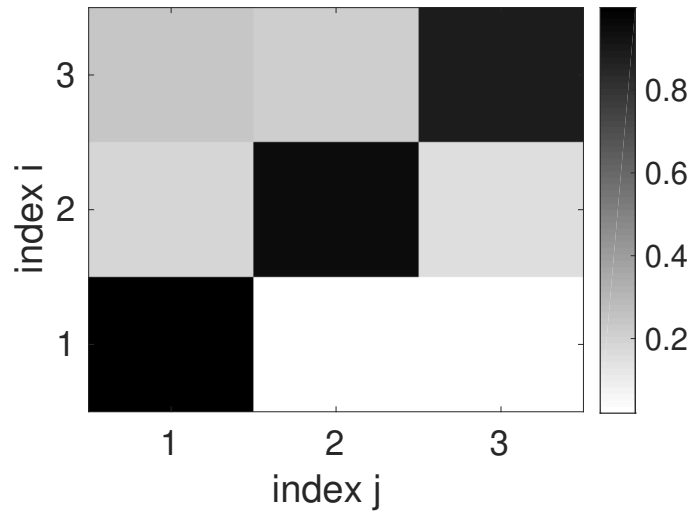


Figure 7: Sensitivity matrix  $[S]$ .

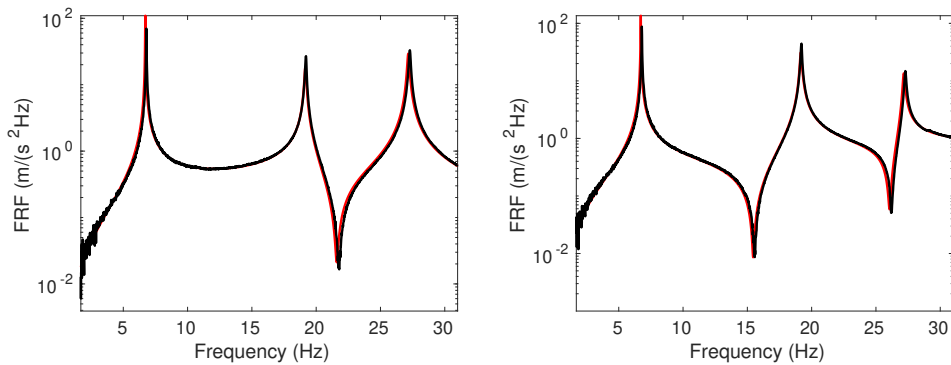


Figure 8: Comparison between the experimental and numerical FRF in acceleration for storey 1 (left) and storey 2 (right), after the second updating iteration (3 damping ratios only).

sensitive to this parameter only. The sensitivity is measured by Sobol indices. The separability of each parameter can be directly quantified using the obtained sensitivity matrix. Then each separable parameter can be updated using a univariate inverse problem solver. Then the procedure is restarted with the remaining non-separable parameters.

The method has been illustrated with a 3-storey structures for which experimental FRFs are measured. All the parameters of the model have been updated using two iterations of the algorithm. The response of the updated model is then very close to the experimental response. Such results would have been difficult to obtain using a manual updating.

In this work modelling errors are taken into account. Nevertheless other sources of uncertainties/variability may exist in the model. In this case a stochastic model should be constructed and should be updated using realizations of experimental outputs. The extension of the present work to the updating of such stochastic models will be addressed in future works.

- [1] M. Baruch, Optimisation procedure to correct stiffness and flexibility matrices using vibration data, *AIAA Journal*. 16(11) (1978) 1208–1210.
- [2] M. Baruch, Modal data are insufficient for identification of both mass and stiffness matrices, *AIAA journal*. 35(11) (1997) 1797–1798.
- [3] Y. Ben-Haim, U. Prells, Selective sensitivity in the frequency domain: Theory, *Mechanical Systems and Signal Processing*. 7 (1993) 461–475.



- [4] A. T. Bollinger, Finite element model updating for FEA/EMA modal correlation via constrained optimization theory, IMAC Proc., 882–888, 1994.
- [5] S. Cogan, D. Lenoir, G. Lallement, An improved frequency response residual for model correction, IMAC Proc., 568–573, 1995.
- [6] S. Cogan, G. Lallement, F. Ayer, Y. Ben-Haim, Model order reduction by selective sensitivity, AIAA Journal. 35(3) (1997) 557–562.
- [7] M. I. Friswell, J. E. Mottershead, Finite Element Model Updating in Structural Dynamics, Kluwer Academic Publishers, 1995.
- [8] M. I. Friswell, D. J. Inman, D. F. Pilkey, Direct updating of damping and stiffness matrices, AIAA journal. 36(3) (1998), 491–493.
- [9] B. Jaishi, W.-X. Ren, Structural Finite Element Model Updating Using Ambient Vibration Test Results, Journal of Structural Engineering, 131(4), (2005), 617-628.
- [10] L. Ljung, System Identification: Theory for the User, Prentice-Hall, New York, 1987.
- [11] J.E. Mottershead, M. Link, M.I. Friswell, The sensitivity method in finite element model updating: A tutorial, Mechanical Systems and Signal Processing, 25(7) (2011), 2275–2296,
- [12] P. Pardalos, E. Romeijn, Handbook of global optimization, volume 2, Kluwer: Dodrecht, 2002.
- [13] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index, Comput. Phys. Commun. 181 (2) (2010) 259270.
- [14] M. J. Schultz, D. J. Inman, Model updating using constrained eigenstructure assignment, Journal of Sound and Vibration. 178(1) (1994) 113-130.
- [15] I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Math. Comput. Simul. 55 (1) (2001) 271–280.

- [16] C. Soize, Random matrix theory for modeling uncertainties in computational mechanics, *Computer Methods in Applied Mechanics and Engineering*. 194 (2005) 1333-1366.
- [17] A.N. Tikhonov, V.Y. Arsenin. *Solutions of Ill-Posed Problems*. Winston and Sons, Washington DC, 1977.
- [18] E. Walter, L. Pronzato, *Identification of Parametric Models from Experimental Data*, Springer-Verlag, Berlin, 1997.
- [19] D. C. Zimmerman, T. Simmermacher, Model correlation using multiple static load and vibration tests, *AIAA journal*. 33(11) (1995) 2182-2188.
- [20] Z. Yuan, P. Liang, T. Silva, K. Yu, J.E. Mottershead, Parameter selection for model updating with global sensitivity analysis, *Mechanical Systems and Signal Processing*, 115 (2019), 483–496,