XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING

Finite Element Model Calibration Using Genetic Algorithm and Bayesian Optimization

Thiago A. M. Souza[1], Giedre A. Sirilo[1], João V. F. Dias[1], Túlio N. Bittencourt[2], Hermes Carvalho[2]

[1]*Dept. of Civil Engineering, University of Espírito Santo*
Room 103 CT 1 – Av. Fernando Ferrari,29075-910, Espírito Santo, Brazil
thiago.arthursouza@gmail.com, giedresirilo@hotmail.com, joao.v.dias@ufes.br
[2]*Graduate Program of Civil Engineering, University of São Paulo*
Av. Prof. Luciano Gualberto 158, 05508-900, São Paulo/São Paulo, Brazil
tbitten@usp.br, *hermesc2000@gmail.com*

Abstract. Calibration techniques are essential for applying finite element models (FEM) in structural assessment and monitoring, given the uncertainties inherent to modeling. Among indirect calibration approaches, metaheuristic algorithms stand out for their ability to explore large search spaces and avoid local minima, but they require many costly model evaluations. To address these limitations, surrogate-based approaches have emerged, approximating the behavior of the objective function and substantially reducing the number of physical model evaluations needed to obtain suitable solutions. Bayesian Optimization (BO), for example, employs a probabilistic model to estimate the target function, enabling not only efficient optimization but also providing quantitative insights into parameter sensitivity and prediction uncertainty. In this work, the performance of a widely used metaheuristic, the Genetic Algorithm (GA), is compared with that of BO in the calibration of a synthetic bridge model, using modal data with added noise. The results demonstrate BO's versatility, competitive accuracy under noisy conditions, and markedly lower computational cost, while also indicating that hybrid strategies can further improve precision.

Keywords: optimization algorithm, calibration, modal analysis, Bayesian optimization, finite elements model.

1 Introduction

Finite element models are widely used to simulate the behavior of systems across many fields of science and engineering, as highlighted by Sanayei et al. [1]. In structural engineering, finite element models (FEM) serve as a computational tool for monitoring and assessment of structures, providing an alternative to traditional methods such as visual inspections, which are highly subjective and often fail to detect non-visible degradation mechanisms. As noted by Chen and Ni [2], these limitations and the deterioration process make it difficult to determine a structure's condition from service life, age, or visual aspects alone. The use of modern computational technologies therefore helps to support evaluations and to mitigate such problems.

The development of numerical models involves multiple levels of uncertainty — from incorrect assumptions and inappropriate simplifications to parametric uncertainty, particularly in material properties — which, according to Simoen et al. [3], can produce significant deviations in the predicted response. These challenges motivated the development of finite element model updating (FEMU), a set of procedures aimed at calibrating numerical models using responses extracted from a reference structure through static and/or dynamic testing. Friswell and Mottershead [4] define FEMU as an iterative procedure for correcting numerical models based on experimental data. Broadly speaking, updating approaches fall into two categories: direct methods, in which the stiffness and mass matrices are adjusted directly, and indirect methods, in which model parameters are tuned to improve agreement between modelled and measured responses.

Indirect methods are widely adopted in structural engineering problems, as physically meaningful adjustments enhance both the relevance of the model and, as highlighted by Girardi et al. [5], its simulation

capability. In the case of iterative indirect methods, algorithms incorporating various optimization strategies are employed, with stochastic, nature-inspired approaches being particularly common. Examples include traditional algorithms such as the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), proposed respectively by Holland [6] and Eberhart and Kennedy [7]. However, due to their iterative nature and the inherent complexity of engineering problems, these methods often entail high computational costs.

As an alternative, other approaches employ statistical models to predict the analysis response and strategically select the evaluation points, thereby reducing the computational burden. This is the case of Bayesian Optimization (BO), which combines a probabilistic surrogate model with an acquisition function to guide the calibration process, as discussed by Raviolo et al. [8]. In this work, BO will be compared with the widely adopted GA in terms of both computational cost and the accuracy of the solutions obtained.

2 Methodology

To evaluate the calibration methodologies, a simplified bridge model (Fig. 1) was proposed. The model consists of a 25-m-long, 8-m-wide and 0.30-m-thick reinforced concrete deck supported by two longitudinal girders located 1.5 m from each deck edge. Each girder has a cross-section 0.60 m wide and 2.00 m high, and is supported at its ends by elastic bearings modeled as linear springs. The concrete material properties and the support stiffness values are specified in Fig. 1. This reference model was used to generate synthetic data for the FEMU procedure.

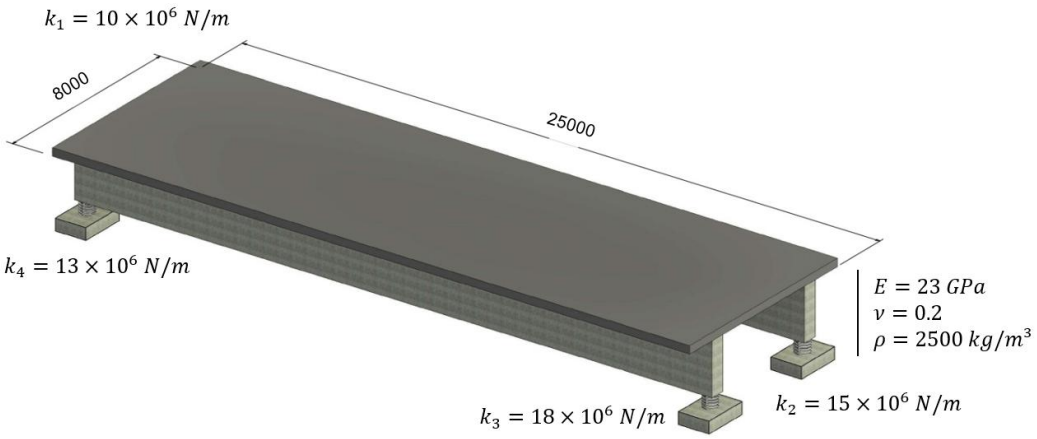


Figure 1. Schematic representation of the structure, dimensions in millimeters

The workflow employed two software packages: ANSYS Mechanical APDL 2025 R1 for finite element modelling and extraction of modal parameters, and Python 3.13 to receive model data, manage the optimization algorithm, and automate ANSYS executions with updated parameters at each iteration, as illustrated in the flowchart in Fig. 2.

2.1 Structural Modelling

The bridge deck was modeled in ANSYS using the SHELL181 element. The longitudinal girders were represented by BEAM188 elements, and the supports were modeled with COMBIN14 spring elements having individually assigned stiffnesses. Translations in the global X and Z directions were constrained at the slab edges, while translation in the Y direction was provided by the springs. Seven parameters were used as design variables for calibration, encompassing material properties and spring stiffnesses (see Tab. 1). The corresponding search ranges were chosen from the literature and were kept broad for the spring stiffnesses due to higher uncertainty.

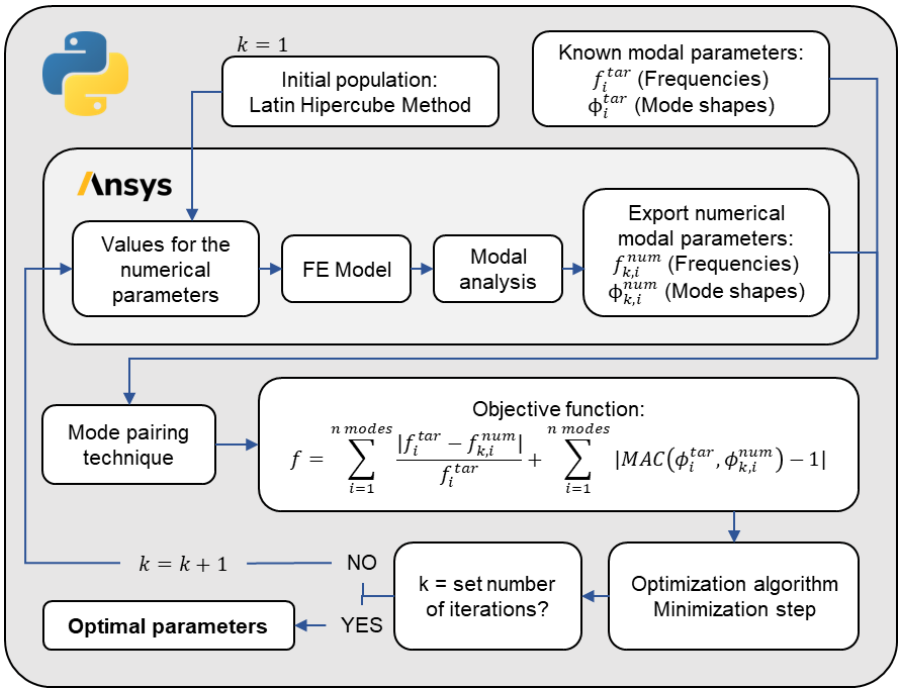
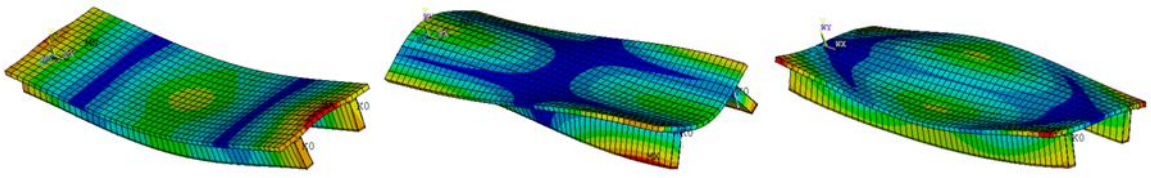


Figure 2. Flowchart of the calibration methodology based on optimization algorithms

Table 1. Design variables and search range

Parameter	Target Values	Lower Bound	Upper Bound	Unit
E	23.00	20.00	30.00	GPa
ν	0.20	0.20	0.50	-
ρ	2500.00	2400.00	2600.00	kg/m³
k_1	1.00E+07	1.00E+06	1.00E+08	N/m
k_2	1.50E+07			
k_3	1.80E+07			
k_4	1.30E+07			

A modal analysis was carried out using the Lanczos method to obtain twenty natural frequencies and their associated mode shapes. Displacement data were extracted at twelve points, and three mode shapes of the reference bridge model are shown in Fig. 3.



Vibration mode 7 (f = 13.18 Hz) Vibration mode 8 (f = 14.45 Hz) Vibration mode 9 (f = 17.56 Hz)

Figure 3. Vibration modes of the reference model

2.2 Objective Function and mode pairing

The calibration process involves an inverse optimization problem in which a function is defined to be minimized. In this study, the objective function is based on modal data, comprising one term associated with the

natural frequencies and another with the Modal Assurance Criterion (MAC) values, as expressed in Eq. 1.

$$f = \sum_{i=1}^{n\,modes} \frac{|f_i^{tar} - f_i^{num}|}{f_i^{tar}} + \sum_{i=1}^{n\,modes} |MAC(\phi_i^{tar}, \phi_i^{num}) - 1| \quad (1)$$

The terms f_i^{tar} and ϕ_i^{tar} denote pseudo-experimental vibration data — specifically, the natural frequency and the displacement vector corresponding to vibration mode i — both extracted from the initially established model. In contrast, f_i^{num} and ϕ_i^{num} correspond to the values obtained from the numerical model at each optimization step.

The MAC, proposed by Allemang and Brown [9] and detailed in Eq. 2, is used to evaluate the consistency between the mode shapes of the reference and numerical models. Values close to 1 indicate strong similarity between modes, while values near 0 suggest low correlation. This function is also applied prior to the fitness calculation to reorder the numerical frequencies and their associated mode shapes based on the correlation with the reference modes, in a step called mode pairing. This procedure ensures that closely spaced frequencies are not mistakenly swapped during the comparisons.

$$MAC(\phi_i^{tar}, \phi_i^{num}) = \frac{\left|\phi_i^{tar^T} \phi_i^{num}\right|^2}{\left(\phi_i^{tar^T} \phi_i^{tar}\right)\left(\phi_i^{num^T} \phi_i^{num}\right)} \#(2)$$

2.3 Optimization Process

The optimization process was developed in Python within an object-oriented package that manages both the optimization components and the interface with the modeling software, controlling the analyses and importing the results. The Genetic Algorithm (GA), implemented in-house by the authors, optimizes the model by emulating principles of natural selection and genetic evolution through its various operators. For the present problem, a mutation rate of 10% was employed, parent selection was performed using the tournament method, uniform crossover was applied with a rate of 60%, and an elitism rate of 10% was adopted to preserve the best solutions.

For the Bayesian Optimization (BO), the implementation was carried out in Python using the scikit-optimize library [10], with adjustments specific to the problem at hand. BO employs a probabilistic model — in this case, a Gaussian Process — that provides not only an estimate of the objective function value at unexplored points but also a measure of uncertainty associated with each prediction. Its behavior is governed by the acquisition function, which defines the strategy for selecting new candidate points based on the probabilistic model's predictions. In this study, three acquisition functions were evaluated: Expected Improvement (EI), Probability of Improvement (PI), and Upper Confidence Bound (UCB).

For both algorithms, the initial population (or initial sampling in the case of BO) was generated using the Latin Hypercube Sampling (LHS) method to ensure a more uniformly distributed and representative coverage of the search space. For the GA, a population of 70 individuals (ten times the number of decision variables) was adopted, with 30 generations, totaling 1,960 evaluated individuals, including the initial population and excluding the 10% preserved in each generation without revaluation. Considering the more selective sampling strategy of BO and the additional computational cost of recalibrating the probabilistic model after each evaluation, the total number of evaluations in this case was approximately 25% of that of the GA, amounting to 490, including 70 initial points evaluated prior to building the surrogate model.

Both algorithms were tested with noise levels of 0%, 5%, and 15% applied to the reference data, and their performance was evaluated in terms of accuracy and computational time. Each case was executed at least four times.

3 Results and Discussion

Starting with the GA, four runs were performed as described in Section 2.3, and the average results are compiled in Tab. 2.

Table 2. Average results for Genetic Algorithm

Parameter	Reference Value	0% noise		5% noise		15% noise	
		Updated Value	Error (%)	Updated Value	Error (%)	Updated Value	Error (%)
E (GPa)	2.30E+10	2.41E+10	5%	2.35E+10	2%	2.33E+10	1%
ν	0.20	0.12	-40%	0.12	-40%	0.12	-40%
ρ (kg/m³)	2500	2.58E+03	3%	2509.56	0%	2.58E+03	3%
k_1 (N/m)	1.00E+07	8.86E+06	-11%	5.66E+06	-43%	8.64E+06	-14%
k_2 (N/m)	1.50E+07	1.66E+07	11%	3.15E+07	110%	2.19E+07	46%
k_3 (N/m)	1.80E+07	1.94E+07	8%	3.24E+07	80%	2.70E+07	50%
k_4 (N/m)	1.30E+07	1.56E+07	20%	2.12E+07	63%	2.71E+07	108%
Average Fitness		0.242		2.200		2.672	
Best Fitness		0.110		0.719		1.384	

Considerable discrepancies can be observed for certain parameters, particularly the Poisson's ratio across all scenarios, which will be discussed later, based on the length-scales returned by the BO's kernel. The results for the remaining variables are satisfactory for the noise-free case; however, the application of 5-15% noise affects the support stiffness values. Figure 4 presents the GA convergence across generations for the different noise levels, showing mean values and variation ranges. It can be observed that, although the algorithm improves fitness at a similar rate during the initial generations and converges to approximately around the same generation, the noisy scenarios struggle to further refine the solution thereafter.

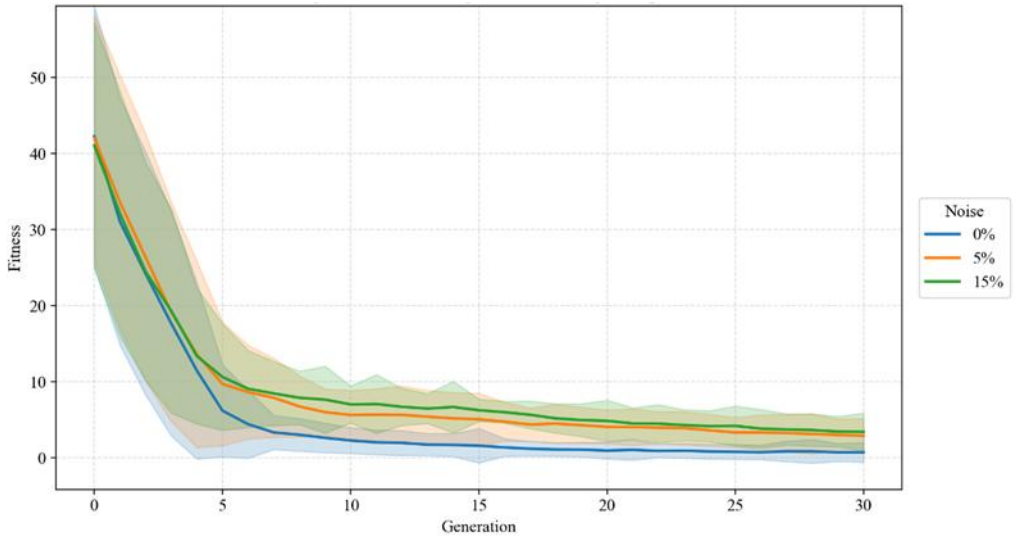


Figure 4. Convergence line for Genetic Algorithm

Following the steps described in Section 2.3, the performance of different acquisition functions was assessed for BO. The PI function (Fig. 5) exhibited the most exploitation-oriented behavior for the problem at hand, achieving both the best fitness values for the given evaluations and the fastest convergence, and was therefore selected for further analyses. Variations of the hyperparameter ξ, which governs the exploration-exploitation trade-off, were also tested; however, the final value was kept at the library default of 0.01. The average results for the BO runs using the PI acquisition function, ξ = 0.01, 70 seed points, and 490 total evaluations for different noise levels are presented in Tab. 3. Figure 6 shows the convergence of the algorithm for the different scenarios.

In the noise-free case, BO performed slightly worse than GA, accumulating larger errors in the support stiffness estimates. Despite the rapid convergence observed, the algorithm did not reach the precision of the stochastic solution. However, when noise was introduced, BO produced more accurate results for both modal data and model parameters. Combined with its faster processing — each BO run required between 1.5 h and 2 h,

whereas a GA run took more than 3 h on the same machine (Intel® Core™ i5-7200U CPU with a 2.50 GHz base frequency) — BO proved to be an appropriate and versatile tool.

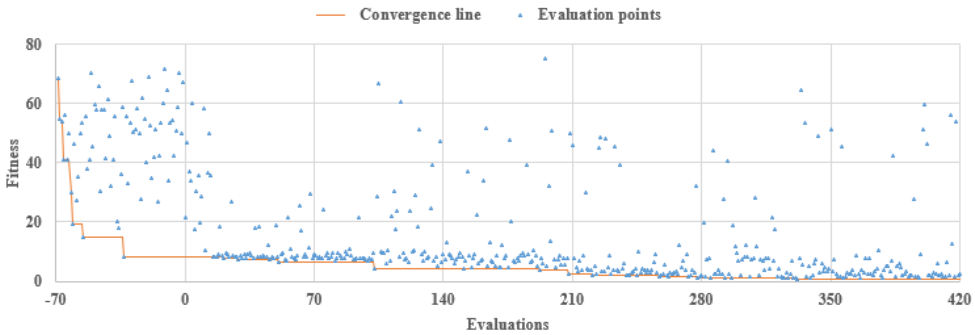


Figure 5. BO behavior using PI acquisition function

Table 3. Average results for Bayesian Optimization

Parameter	Reference Value	0% noise		5% noise		15% noise	
		Updated Value	Error (%)	Updated Value	Error (%)	Updated Value	Error (%)
E (GPa)	2.30E+10	2.40E+10	4%	2.27E+10	-2%	2.36E+10	3%
ν	0.20	0.1503803	-25%	0.1526368	-24%	0.1692053	-15%
ρ (kg/m³)	2500	2.55E+03	2%	2457.7883	-2%	2.48E+03	-1%
k_1 (N/m)	1.00E+07	7.50E+06	-25%	5.53E+06	-45%	4.67E+06	-53%
k_2 (N/m)	1.50E+07	1.62E+07	8%	1.79E+07	19%	1.19E+07	-21%
k_3 (N/m)	1.80E+07	2.47E+07	37%	2.26E+07	25%	2.88E+07	60%
k_4 (N/m)	1.30E+07	1.63E+07	26%	1.46E+07	12%	1.49E+07	15%
Average Fitness		0.444		0.883		1.992	
Best Fitness		0.388		0.642		1.601	

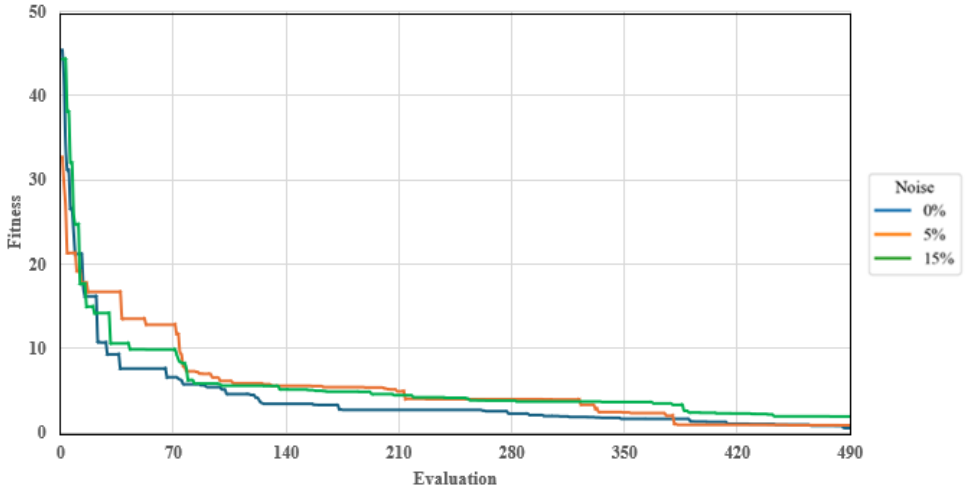


Figure 6. Convergence line for Bayesian Optimization

Additionally, BO's kernel length-scales can be recovered to assess variable sensitivity: smaller values indicate greater influence on the objective function, while larger values suggest lower impact, allowing for

quantitative comparison. For a representative run of the first case, the smallest length-scale corresponded to Young's modulus, followed by slightly larger values for support stiffness and density, and much larger values for Poisson's ratio — explaining the inability of the algorithms to improve this parameter, which could be disregarded in a sensitivity analysis.

Despite its versatility, BO did not reach the same level of accuracy achieved by GA in the first case, indicating the need for alternative strategies or hybrid methodologies. For instance, initializing GA with the best individuals found by BO allowed the optimization to reach a fitness value of 0.06 and small parameter errors within 11 generations — better than the results obtained after 30 full GA generations.

4 Conclusions

Both methodologies proved effective for application in FEMU, but BO demonstrated greater versatility in handling noisy scenarios, while also being less costly and providing insights into the input variables. However, the difficulty in refining accuracy beyond a certain threshold highlights the need for new strategies, either through the hybrid application of algorithms or, as discussed by Torzoni et al. [11], the use of multi-fidelity models — starting with a low-fidelity model to quickly capture the structural behavior and then employing a high-fidelity model to enrich or correct the prediction. Future studies may explore different scenarios to improve the performance of these algorithms, as well as the application to more complex structures and data from experimental campaigns.

Acknowledgements. The authors acknowledge the financial support provided by the Brazilian Federal Government Agency CAPES, the National Council for Scientific and Technological Development (CNPq), the State Agency of Espírito Santo (FAPES), and the Vale S.A. "Cátedra Under Rail" program.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] M. Sanayei, A. Khaloo, M. Gul, and F. Necati Catbas, "Automated finite element model updating of a scale bridge model using measured static and modal test data," *Eng. Struct.*, vol. 102, pp. 66–79, Nov. 2015, doi: 10.1016/j.engstruct.2015.07.029.

[2] H. Chen and Y. Ni, *Structural Health Monitoring of Large Civil Engineering Structures*, 1st ed. Wiley, 2018. doi: 10.1002/9781119166641.

[3] E. Simoen, G. D. Roeck, and G. Lombaert, "Dealing with uncertainty in model updating for damage assessment: A review," *Mech. Syst. Signal Process.*, vol. 56–57, pp. 123–149, May 2015, doi: 10.1016/j.ymssp.2014.11.001.

[4] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*, vol. 38. in Solid Mechanics and its Applications, vol. 38. Dordrecht: Springer Netherlands, 1995. doi: 10.1007/978-94-015-8508-8.

[5] M. Girardi, C. Padovani, D. Pellegrini, M. Porcelli, and L. Robol, "Finite element model updating for structural applications," *J. Comput. Appl. Math.*, vol. 370, p. 112675, May 2020, doi: 10.1016/j.cam.2019.112675.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 1992. doi: 10.7551/mitpress/1090.001.0001.

[7] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan: IEEE, 1995, pp. 39–43. doi: 10.1109/MHS.1995.494215.

[8] D. Raviolo, M. Civera, and L. Z. Fragonara, "A Comparative Analysis of Optimization Algorithms for Finite Element Model Updating on Numerical and Experimental Benchmarks," *Buildings*, vol. 13, no. 12, p. 3010, Dec. 2023, doi: 10.3390/buildings13123010.

[9] R. J. Allemang and D. L. Brown, "A Correlation Coefficient for Modal Vector Analysis," in *Proceedings of the 1st International Modal Analysis Conference & Exhibit*, 1982, pp. 110–116.

[10] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi, *scikit-optimize/scikit-optimize*. (2021).

[11] M. Torzoni, A. Manzoni, and S. Mariani, "A Deep Neural Network, Multi-fidelity Surrogate Model Approach for Bayesian Model Updating in SHM," in *European Workshop on Structural Health Monitoring*, vol. 254, P. Rizzo and A. Milazzo, Eds., in Lecture Notes in Civil Engineering, vol. 254. , Cham: Springer International Publishing, 2023, pp. 1076–1086. doi: 10.1007/978-3-031-07258-1_108.