Finite Element Model Updating using Bayesian Optimization Algorithm

Cássio B. Mainardes[1], Sergio H. S. Carneiro[1]

[1]*Faculdade do Gama, Universidade de Brasília*
Área Especial de Indústria Projeção A, 72.444-240, DF, Brasil
cassiobus@gmail.com, shscarneiro@unb.br

Abstract. Structural engineering projects require reliable information about the parameters of the designed systems that convey precise predictions about the behavior of the system under different load cases. These predictions are derived from numerical engineering models using the well-established Finite Element Method (FEM). However, due to the presence of uncertainties and assumptions made during the construction of the model, the resulting response may not accurately represent the real structural behavior. Thus, the established approach involves supplying the numerical model with experimental data to reduce numerical-experimental error in a process called Finite Element Model Updating (FEMU). The focus of this paper is on the implementation of the Bayesian Optimization Algorithm (BOA) to solve FEMU problems. To assess the behavior of the algorithm, many cases are analyzed from lower evaluation time to higher complexity cases and then compared with other methods, such as Bayesian Inference, and Particle Swarm Optimization. Results show that BOA can attain good results in a short amount of time when applied to FEMU for low dimensionality problems compared with well-established methods. Overall, this paper reviews FEMU methods and proposes the implementation of a novel approach, demonstrating its effectiveness in solving model updating problems.

Keywords: finite element model updating; bayesian optimization algorithm; structural dynamics

1 Introduction

Structural engineering projects are often subject to severe ambient conditions, being commonly exposed to many occurrences of dynamical loads. Such loads may result in unwanted or dangerous vibrations if not correctly addressed, potentially leading to catastrophic failures. Therefore, correct evaluation of the system's dynamic properties and response is very much needed. This is often done by constructing a numerical Finite Element (FE) model, which allows experimentation and study of changes in the construction of the model by easy modification of its parameters.

However, FE models are not usually entirely accurate. These models exhibit many errors that arise due to uncertainty from many sources. Ereiz [1] and Sehgal [2] list several factors that contribute to the modeling error of FE models, such as incorrect assumption of material and section properties, idealization of the model, faulty boundary conditions, joint modeling, rounding errors, and many others. On the other hand, the real structure is not ideal as well. Marwala [3] addresses as uncertainties fabrication processes, damage, and errors and noise in experimental measurements.

On behalf of all these sources of uncertainty, FE models need to be corrected for proper use in the prediction of dynamical properties of the system. Finite Element Model Updating (FEMU) is an inverse engineering problem in which the input and output of the system are known, but the system has unknown or inaccurate parameters. The objective of this problem is to correlate numerical and experimental measures of dynamical properties, adjusting system parameters. Several techniques for FEMU exist and have been reviewed by Ereiz [1], Sehgal [2] and Gomes

[4].

These techniques may be classified as optimization and sampling techniques. Algorithms used for optimization minimize a function, commonly the error between natural frequencies of experimental and numerical measures. Examples of used algorithms for this end are Particle Swarm Optimization (PSO), used by Li [5], Marwala [6], Qin [7], and many others, and Genetic Algorithm (GA), also studied by Marwala [6], Gomes [4] to name a few. These established algorithms are fast to implement and attain good results.

Sampling techniques, on the other hand, demand repeated sampling of the system to attain parameter distribution curves or likelihood distribution. The most used sampling technique is Bayesian Inference, which has been studied by authors such as Marwala [3] [8], Carlon [9] and Lu [10]. This method not only gives the most probable parameters of the system but also accounts for some uncertainty, however, it's quite slow given that a high number of samples is needed for good resolution of the likelihood distribution.

In this work, the use of the Bayesian Optimization Algorithm (BOA) in FEMU studied by Mainardes [11] is continued, and its application is compared with other FEMU techniques. BOA is a derivative-free optimization algorithm that operates by constructing a statistical model with measured points to predict the best point to evaluate next. This process is done by another optimization of an acquisition function built with the statistical model, adding more time to a single interaction with the intent of reducing total iterations of the objective function. This works well when the evaluation of the objective function is computationally expensive, reducing overall runtime. However, BOA is a low-accuracy algorithm, meaning that results may not be as precise as PSO, and works better in low-dimension problems. This algorithm has seen lots of uses in many areas, as shown by Malu [12], and even applications in aerospace engineering by Lam [13] and Morita [14].

Overall, this work reviews Bayesian Inference, PSO and BOA, comparing those methods in three different evaluation time cases. The first is to optimize a simple test function, using PSO and BOA. Then, for the application of FEMU, a honeycomb panel is used with two different models of different complexity, using the natural frequencies as the objective function. In the following sections, a quick review of the used methods is done. Then, the honeycomb panel and its numerical models are described before showing results and discussion about the three used methods.

2 Mathematical background

2.1 Dynamic systems

Gomes [4] states that eq. (1) describes the free vibration of undamped multiple-degree-of-freedom systems:

$$(\boldsymbol{K} - \omega^2 \boldsymbol{M})\{\phi\} = \boldsymbol{0}. \tag{1}$$

which **K** and **M** are stiffness and mass matrices, respectively. Solution of eq. (1) through an eigenvalue problem returns the eigenvalues ω^2 associated with natural frequencies and eigenvectors $\{\phi\}$ that represent mode shapes. FE models allow the construction of matrices **K** and **M**, thus obtaining the natural frequencies that will be used to correlate numerical and experimental data.

2.2 Bayesian inference

The Bayesian Inference method is derived directly from Bayes Theorem, functioning by updating the belief upon the distribution of a set of parameters with insertion of observed data. Bayes theorem is given by eq. (2):

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{P(\boldsymbol{D})} \tag{2}$$

where $\boldsymbol{\theta}$ is the vector that represents the set of parameters to be updated and **D** is the matrix of observed data. Thus, eq. (2) correlates the probability distribution of the observed data $P(\boldsymbol{D})$, the initial probability distribution of the set of parameters, known as the prior distribution $P(\boldsymbol{\theta})$, and the likelihood distribution $P(\boldsymbol{D}|\boldsymbol{\theta})$ to the distribution of the parameters given the observed data or the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{D})$.

Throughout this work, the likelihood and the priori functions are built to be normal, guaranteeing a normal posterior distribution through eq. (2), as stated by Marwala [3]. Thus, the likelihood distribution is represented by eq. (3):

$$P(\boldsymbol{D}|\boldsymbol{\theta}) = \frac{1}{z(D|\theta)} exp\left(-\sum_i^{N_m} \frac{\beta_i}{2}\left(\frac{f_i^m - f_i(\boldsymbol{\theta})}{f_i^m}\right)^2\right) \quad (3)$$

where β_i are weighting constants associated with each natural frequency, N_m is the number of measured modes, and f_i^m and $f_i(\boldsymbol{\theta})$ are measured and numerical experimental frequencies, respectively. $z(\boldsymbol{D}|\boldsymbol{\theta})$ is a normalization factor. Analogously, the prior distribution is given by eq. (4):

$$P(\boldsymbol{\theta}) = \frac{1}{z(\boldsymbol{\theta})} exp\left(-\sum_i^{D} \frac{\alpha_i}{2}\left(\frac{\theta_i - \theta_i^0}{\theta_i^0}\right)^2\right) \quad (4)$$

where θ_i is an updating parameter, θ_i^0 is the prior mean of the updating parameter, D is the number of updating parameters, and $z(\boldsymbol{\theta})$ is the normalization factor. Combining both equations using Bayes Theorem, the posterior distribution is obtained and given by eq. (5):

$$P(\boldsymbol{\theta}|\boldsymbol{D}) \propto exp\left(-\sum_i^{N_m} \frac{\beta_i}{2}\left(\frac{f_i^m - f_i(\boldsymbol{\theta})}{f_i^m}\right)^2 - \sum_i^{D} \frac{\alpha_i}{2}\left(\frac{\theta_i - \theta_i^0}{\theta_i^0}\right)^2\right) \quad (5)$$

which is also normal. This is the distribution of probabilities from which samples will be obtained using a Markov Chain Monte Carlo (MCMC) sampled with Metropolis acceptance criteria. This algorithm will generate a set of samples and obtain $P_i(\boldsymbol{\theta}|\boldsymbol{D})$ using eq. (5). Then, the next sample will be obtained based on a normal distribution centered on the current set of parameters, calculating probability $P_{i+1}(\boldsymbol{\theta}|\boldsymbol{D})$. This new sample is accepted or rejected by the Metropolis criteria, which Marwala [6] and Sadegh [15] describe in eq. (6):

$$P_{accept}(i \rightarrow i+1) = min\,(1, \frac{P_{i+1}(\boldsymbol{\theta}|\boldsymbol{D})}{P_i(\boldsymbol{\theta}|\boldsymbol{D})}) \quad (6)$$

if accepted, the set of parameters $\boldsymbol{\theta_i}$ will be updated to $\boldsymbol{\theta_{i+1}}$, resulting in a random walk that, given enough samples, will represent the posterior distribution which may be used to obtain data about the parameters of the system.

2.3 Particle swarm optimization

PSO is an optimization algorithm initially proposed by Kennedy and Ebehart [16] that aims to copy the observed behavior of flocks of birds and fishes. It rapidly gained popularity being of simple implementation and great results, gaining several adaptations over the years. Authors such as Li [5] and Marwala [6] and many others have already implemented such method for FEMU, being a very popular algorithm.

This algorithm operates on so-called particles. Those particles are attracted by the region of the best result they have crossed and by the best spot any particle ever crossed. Thus, each particle i may be represented by two variables at each iteration k, position $\boldsymbol{p_i}(k)$ and velocity $\boldsymbol{v_i}(k)$. Initially, each particle is given a random position and velocity which are updated every iteration by eqs. (7) and (8):

$$\boldsymbol{v_i}(k+1) = w(k) \cdot \boldsymbol{v}_i(k) + c_1 \cdot rand(0{,}1) \cdot \left(l_{best,i} - \boldsymbol{p}_i(k)\right) + c_2 \cdot rand(0{,}1) \cdot \left(g_{best} - \boldsymbol{p}_i(k)\right) \quad (7)$$

$$\boldsymbol{p}_i(k+1) = \boldsymbol{p}_i(k) + \boldsymbol{v}_i(k+1) \quad (8)$$

where $w(k)$ is a factor between 0 and 1 that reduces the speed at later iterations for better convergence, rand(0,1) is a uniform random number between 0 and 1,$l_{best,i}$ is the best position found by the particle i, g_{best} is the best position found overall and c_1 and c_2 are acceleration constants.

2.4 Bayesian optimization algorithm

BOA operates by constructing a statistical model of the objective function. Here this is done by Gaussian

Processes (GP), which is then used to construct an acquisition function in terms of mean and standard deviation of the model. This function is then maxed to obtain the best next point of evaluation. This essentially increases the time taken to run an iteration, however, reduces overall time by reducing the number of total iterations needed, which is advantageous when the evaluation of an objective function is expensive. The details of the functioning of the BOA are given by Snoek [17], while a review of GPs can be seen by Wang [18].

The statistical model is constructed using data from all points available, creating a GP whose mean and standard deviation at every point in the domain are obtainable by eqs. (9) and (10), respectively:

$$\mu(\boldsymbol{x}^*) = k(\boldsymbol{x}, \boldsymbol{x}^*)^T (k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_r^2 \boldsymbol{I})^{-1} \boldsymbol{y} \tag{9}$$

$$\sigma(\boldsymbol{x}^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) + k(\boldsymbol{x}, \boldsymbol{x}^*)^T (k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_r^2 \boldsymbol{I})^{-1} k(\boldsymbol{x}, \boldsymbol{x}^*) \tag{10}$$

where $\boldsymbol{x}$ is the vector of observed points, $\boldsymbol{x}^*$ is an unobserved point in the domain and $\boldsymbol{y}$ is the vector of observations so far. σ_r is a hyperparameter of the algorithm that represents expected noise in observed samples. $k(\boldsymbol{x}, \boldsymbol{x})$ is the matrix of covariance between points, which is obtained through a kernel function. Snoek [16] states that for optimization problems, a Matern 52 function works well, however other options exist. The Matern is given by eq. (11):

$$k_{M52}(\boldsymbol{x}, \boldsymbol{x}^*) = \sigma_v \left(1 + \frac{\sqrt{5r^2(x,x^*)}}{\sigma_l} + \frac{5}{3\sigma_l^2} r^2(\boldsymbol{x}, \boldsymbol{x}^*)\right) exp\left(-\frac{\sqrt{5r^2(x,x^*)}}{\sigma_l}\right) \tag{11}$$

where σ_v and σ_l are hyperparameters of the kernel function of verticality and horizontality, respectively, and $r^2(\boldsymbol{x}, \boldsymbol{x}^*)$ is the squared Euclidian distance between two points in the space. Finally, the acquisition function is defined in terms of the calculated mean and deviation. For this work, the Expected Improvement function is defined and used as in eq. (12):

$$EI(\boldsymbol{x}) = (\mu - f_{min} - \lambda) \cdot \Phi\left(\frac{\mu - f_{min} - \lambda}{\sigma}\right) + \sigma \cdot \varphi\left(\frac{\mu - f_{min} - \lambda}{\sigma}\right) \tag{12}$$

where f_{min} is the minimum value observed and λ is a hyperparameter that encourages exploration of unseen areas of the domain when assigned higher values. Finally, φ and Φ are the probability and cumulative density functions, respectively.

3 Materials and methods

3.1 Materials and experimental setup

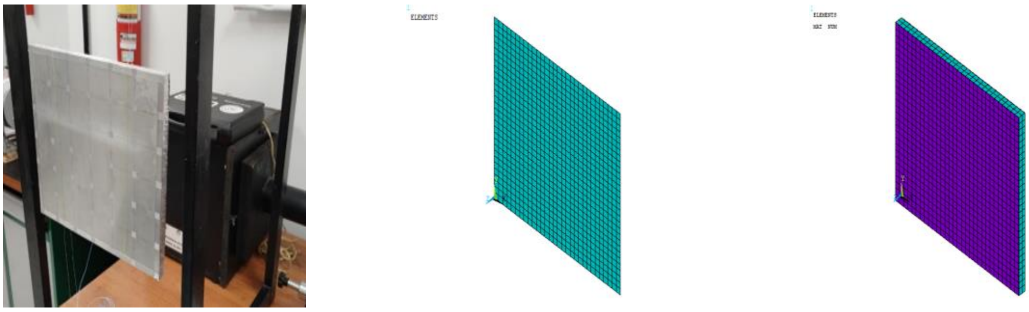
For this work, experimental measurements of a honeycomb Al-Al panel, illustrated on Fig. 1a, were taken with impact testing performed with a PCB 086c02 model impact hammer. The measures were obtained by a 352a21 model accelerometer and recorded with 01 db dB4 acquisition hardware and software for impact on a total of 49 points over the panel. Recorded data was then processed using SimCenter Testlab 2306 software, obtaining mode shapes and natural frequencies for the first five modes.

Full specifications for the honeycomb panel are given in the previous work by Mainardes [11], and the most relevant geometrical and mechanical properties of the core and aluminum external layer are summarized in Tab 1.

3.2 Methodology and modeling

Before applying the FEMU methodology to the honeycomb panel, both optimization functions are applied to minimize a simple test function. This is made to observe how BOA behaves when given an otherwise very simple function with very quick evaluation time. Thus, the chosen test function was the Sphere Function, given by eq. (13), which has a global minimum at the origin:

$$f(\boldsymbol{x}) = \sum_{i=1}^{3} x_1^2 \tag{13}$$



(a) (b) (c)

Figure 1. Panel and modeling (a) Honeycomb panel (b) Laminated Model (c) Solid-Plate Model.

Table 1. Geometric and mechanical parameters of the honeycomb panel

Material	Property	Value
Honeycomb HexWeb CRIII – Al 5056 – 1/4” – 0,001P (10P) core	Shear Modulus G_L	$221\ MPa$
	Shear Modulus G_W	$103\ MPa$
	Equivalent density ρ_{core}	$82\ kg/m^2$
	Thickness t_{core}	$14.4\ mm$
Aluminum 2024-T3	Elastic Modulus E_{al}	$73.9\ GPa$
	Density ρ_{al}	$2780\ kg/m^2$
	Thickness t_{al}	$0.3\ mm$
Honeycomb panel	Overall Thickness t	$15\ mm$
	Length L	$280\ mm$
	Width W	$300\ mm$

The choice of three parameters is made as further uses of both algorithms will also be three-dimensional for this work. Then, FEMU is applied to the honeycomb panel. There are two modelings of the panel, both described by Mainardes [11] in previous work, which are named as Laminated Model and Solid-Shell Model, that are shown in Figs. 1b and 1c. The function to be optimized for PSO and BOA is given by eq. (14):

$$f(\boldsymbol{\theta}) = \sum_{i=1}^{5}\left(c_i \cdot \left(\frac{f_i^m - f_i(\boldsymbol{\theta})}{f_i^m} \cdot 100\right)^2\right) \tag{14}$$

where c_i is a weighting factor. The vector $\boldsymbol{\theta}$ is comprised of three parameters, namely Shear Moduli G_L and G_W of Honeycomb core and external layer Elastic Modulus E_{al}. The range at which each parameter is searched is given in Tab. 2.

Table 2. Search range of updated parameters

Parameter	Minimum search range	Maximum search range
G_L	$100\ MPa$	$300\ MPa$
G_W	$100\ MPa$	$300\ MPa$
E_{al}	$60\ GPa$	$80\ GPa$

As for algorithm parameters, Bayesian inference takes a total of 5000 samples, while PSO searches with 10 particles over 100 generations. Finally, BOA performs 120 evaluations of objective function before stopping.

4 Results and discussion

Starting with the optimization of the sphere function, Table 3 shows the results obtained by both algorithms when run over 500 times in this case. Analyzing Table 3, PSO performs better than BOA for low-complexity functions. Not only it is way faster to evaluate, due to BOA having to construct the statistical model and acquisition function, but it also shows the low accuracy aspect of BOA. While results of BOA over 500 runs tend to stay around 0, they have relatively high deviation, especially when compared to PSO results. This shows that for simple cases, BOA is not adequate, being of hard implementation and poor results.

Table 3. Coefficients in constitutive relations

Results	x_1	x_2	x_3	Elapsed time (s)
PSO mean	$-7.7 \cdot 10^{-8}$	$-7.3 \cdot 10^{-8}$	$-6.9 \cdot 10^{-8}$	0.0013
PSO deviation	0.000001964	0.000002451	0,000003331	0.0035
BOA mean	0.0148	−0.0088	−0.0056	65.753
BOA deviation	0.4698	0.4643	0.4522	1.1724

Tables 4, 5, and 6 show results when applying these algorithms as FEMU methods to both the honeycomb numerical models. It is noted that evaluation of the Laminated Model takes around 4.4 seconds while evaluating the Solid-Plate model takes about 10.3 seconds.

Table 4. Updated parameters for all FEMU methods

Algorithm	Laminated Model				Solid Plate Model			
	$G_L\ (MPa)$	$G_W\ (MPa)$	$E_{al}\ (GPa)$	$Time\ (s)$	$G_L\ (MPa)$	$G_W\ (MPa)$	$E_{al}\ (GPa)$	$Time\ (s)$
Bayesian inference (mean)	222.0	128.8	72.9	25050	266.5	136.3	72.9	47201
PSO	212.7	122.7	73.3	4575	267.3	134.4	72.7	10209
BOA	214.0	122.6	73.3	657	265.5	134.7	72.7	1246

Table 5. Coefficients in constitutive relations

Laminated Model	Experimental		Bayesian Inference		PSO		BOA	
	Freq.	c_i	Freq.	Error (%)	Freq.	Error (%)	Freq.	Error (%)
Mode 1	664.7	5	676.7	−1.805	676.1	−1.715	676.1	−1.715
Mode 2	1031	3	1015	1.552	1014	1.610	1014	1.610
Mode 3	1334	3	1321	0.960	1320	1.027	1321	0.997
Mode 4	1600	1	1598	0.106	1592	0.494	1592	0.475
Mode 5	1699	1	1699	0.041	1693	0.353	1694	0.324

The results show that not only does BOA perform slightly better than PSO, but it is also about seven to eight times faster than PSO, which proves especially advantageous when using it for FEMU on the higher evaluation time of the Solid-Plate model. The low accuracy of BOA presented in the test function case also shows not to be a problem when applying to FEMU, as BOA was able to attain good, if not better, results than Bayesian Inference and the highly accurate PSO. These results also highlight the difference between sampling and optimization algorithms. Bayesian inference is much more time-consuming than both other methods, however, due to its nature, uncertainty of the parameters is also acquired through the probability distribution function, which may be desired when working with highly uncertain systems.

Table 6. Coefficients in constitutive relations

Solid-Plate Model	Experimental		Bayesian Inference		PSO		BOA	
	Freq.	c_i	Freq.	Error (%)	Freq.	Error (%)	Freq.	Error (%)
Mode 1	664.7	5	669.9	−0.782	668.7	−0.602	668.7	−0.602
Mode 2	1031	3	1026	0.456	1024	0.650	1024	0.640
Mode 3	1334	3	1330	0.315	1328	0.442	1328	0.472
Mode 4	1600	1	1599	0.063	1595	0.275	1595	0.281
Mode 5	1699	1	1707	−0.465	1704	−0.294	1704	−0.271

On the other hand, the optimization algorithms shown are fast and precise, offering good, updated models that accurately represent the dynamic characteristics of the studied system, as evidenced by the maximum frequency error of less than 2% on the Laminated Model and less than 0.7% on the more precise Solid-Plate Model.

Overall, BOA attained great results on the FEMU applications presented. However, many cases demand higher dimensionality problems, in which further testing of BOA is needed, as Malu [12] presents some challenges faced when extending the application of this algorithm for higher dimension cases.

5 Conclusions

BOA and other methods were used in FEMU problem application using a honeycomb Al-Al panel, attaining great results, especially in the Solid-Plate model, achieving good correlation between numerical and experimental frequencies. Testing shows that BOA underperforms when applied to simple test functions when compared to PSO, however, excels when compared with the same methods as it is applied to model updating, being faster and achieving comparable results.

Some further investigations are still ongoing when applying BOA to other cases, exploring higher dimensional problems for further inspection of scenarios where the algorithm under or overperforms when compared with other already well-stablished algorithms for FEMU.

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] Suzana Ereiz, Ivan Duvnjak, Javier Fernando Jiménez-Alonso. “Review of finite element model updating methods for structural applications”. Structures, Volume 41, 2022, Pages 684-723, ISSN 2352-0124.
[2] Sehgal, S., Kumar, H. “Structural Dynamic Model Updating Techniques: A State of the Art Review”. Arch Computat Methods Eng 23, 515–533 (2016).
[3] Marwala, T., Boulkaibet, I., and Adhikari, S., 2017. “Probabilistic Finite Element Model Updating Using Bayesian Statistics: Applications to Aeronautical and Mechanical Engineering”. Publicado por Wiley, 1 st edition, 248p.
[4] Gomes, Guilherme F., et al. (2022). “On the use of Modal Test Data in Inverse Problems: Fundamentals and Applications”. In Jorge, Ariosto B., et al. (Eds.) Fundamental Concepts and Models for the Direct Problem, Vol. II, UnB, Brasilia, DF, Brazil, pp. 311–348. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.
[5] Li, K., et al., 2022. “Structural Dynamic Model Updating with Automatic Mode Identification Using Particle Swarm Optimization”. Appl. Sci., 12, 8958.
[6] Marwala, T. 2004. “Dynamic Model Updating Using Particle Swarm Optimization Method”. arXiv.
[7] Qin, S., et al., 2018. “Dynamic Model Updating for Bridge Structures Using the Kriging Model and PSO Algorithm Ensemble with Higher Vibration Modes”. Sensors 2018, 18, 1879.
[8] Marwala T. and Sibisi S. “Finite element model updating using bayesian framework and modal properties”. Journal of Aircraft, vol. 42, n. 1, pp 275-278, 2005.
[9] Carlon, Andre G., et al. (2022). “Bayesian Optimal Experimental Design”. In Jorge, Ariosto B., et al. (Eds.) Uncertainty Modeling: Fundamental Concepts and Models, Vol. III, UnB, Brasilia, DF, Brazil, pp. 479–501. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.
[10] Lu, P., et al., 2023. “Static Behavior Prediction of Concrete Truss Arch Bridge Based on Dynamic Test Data and Bayesian Inference”. International Journal of Structural Stability and Dynamics.
[11] Mainardes C., Carneiro S. “Dynamic Model Updating of Al-Al Honeycomb Sandwich Panels for Aerospace Applications”. XLIV CILAMCE, v. 5 n. 05, 2023, M7 Health Monitoring and Numerical Modeling of Structures.
[12] Malu, M., Dasarathy, G. and Spanias, A., 2021. "Bayesian Optimization in High-Dimensional Spaces: A Brief Survey," 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, pp. 1-8.
[13] Lam, R., et al., 2018. “Advances in Bayesian Optimization with Applications in Aerospace Engineering”.
[14] Morita, Y. et al., 2021. “Applying Bayesian optimization with Gaussian processo regression to computational fluid dynamics problems”. Journal of Computational Physics, Vol. 449.
[15] Sadegh, M., and Vrugt, J.A., 2014, “Approximate Bayesian Computation using Markov Chain Monte Carlo simulation: DREAM(ABC)”. Water Resour. Res., 50, 6767–6787.
[16] Kennedy J. and Eberhart R., 1995. "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, pp. 1942-1948 vol.4.
[17] J. Snoek, H. Larochelle and R. Adams. “Practical Bayesian Optimization of Machile Learning Algorithms”. ArXiv, 2012.
[18] J. Wang. “An intuitive tutorial to gaussian processes regression”. Ingenuity Labs Research Institute, 2022.