Formulation and Optimization Algorithm Comparison for the FE Model Updating of Large-Scale Structures

[1] Xinjun Dong, [1, 2] Yang Wang
[1] School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA, USA
[2] School of Electrical and Computing Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract:

This research studies finite element (FE) model updating formulations utilizing the measured frequency-domain modal properties, i.e. resonance frequencies and mode shapes. The modal properties provided by an FE model are usually different from these collected through experimentally testing an as-built structure. To update the FE model parameters, optimization problems are formulated to minimize the difference between experimental and simulated modal properties. Various FE model parameters can be selected as optimization variables for model updating, such as the elastic moduli of structural members, as well as stiffness values of support springs. Two modal property difference formulations are studied in this research, one using MAC values and the other using direct differences between eigenvectors. For each updating formulation, Jacobian derivative of the objective function is derived in detail. To find the optimal solution of the formulated optimization problem, two optimization algorithms are studied for comparison, namely the Levenberg-Marquardt and the trust-region-reflective algorithms. Randomly generated starting values of the optimization variables are adopted to increase the chance of finding global minimum of the optimization problem. Finally, the presented model updating approaches with different optimization algorithms are validated with small- and large-scale numerical examples. Interested readers can find the MATLAB code and data at https://github.com/ywang-structures/Structural-Model-Updating.git.

Keywords: finite element model updating, modal property difference formulations, nonconvex optimization, Jacobian derivative of objective function, optimization algorithms

1 Introduction

In modern structural analysis, a great amount of effort has been devoted to finite element (FE) modeling towards simulating the behavior of an as-built structure. In general, predictions by FE models often differ from in-situ experimental measurements. Various inaccuracies in the models can contribute to the discrepancy. For example, idealized connections and support conditions are commonly used in structural analysis and design, while these conditions with infinite or zero stiffness do not exist in reality. In addition, material properties of an as-built structure are usually different from the nominal values, particularly for concrete. To obtain a more accurate FE model that truly reflects behaviors of an as-built structure, data collected from the in-situ experiments can be used to update the values of selected model parameters (e.g. support stiffness or mass parameters) in the FE model. This process is known as FE model updating. An updated FE model can more accurately predict structural behavior under various loading conditions, and also serve as the baseline for identifying structural property change over time, potentially due to deterioration. In the meantime, benefiting from the development of low-cost wireless sensing systems [1-3], more and more structural sensors are available for measuring structural responses. As a result, large amount of sensor data collected from as-built structures are becoming available for FE model updating, which on the other hand, poses computational challenges for model updating.

Numerous FE model updating algorithms have been developed and practically applied in the past few decades [4]. Most algorithms can be categorized into two groups, i.e. time-domain and frequency-domain approaches. Time-domain approaches deal with time history data collected from the actual structure directly, without the requirement for extracting modal properties [5-8]. In addition, some time-domain approaches are capable of updating nonlinear structural hysteresis parameters. For example, early research

started with extended Kalman filter (EKF) [9]. The EKF approach forms a state vector containing the displacement and velocity of all degrees of freedom (DOFs), as well as the selected structural parameters to be updated. The state vector is estimated through dynamic state propagation and corrected by minimizing the covariance of the estimation error of the formulated state vector. Overall, the time-domain approaches suffer convergence difficulties and high computational cost when applied to large-scale FE models.

Different from the time-domain approaches, the frequency-domain approaches can update an FE model using frequency-domain modal properties extracted from experimental measurements. These include resonance frequencies, vibration mode shapes and damping ratios. In particular, early researchers started by minimizing an objective function consisting of the differences between measured and simulated resonance frequencies. Naturally, this is formulated as a mathematical optimization problem with the selected structural parameters as optimization variables. This category of model updating approaches is named as modal property difference approach. For example, Zhang *et al.* proposed an eigenvalue sensitivity-based model updating approach that was applied on a scaled suspension bridge model, and the updated FE model shows resonance frequencies closer to the experimental measurements [10]. Salawu reviewed various model updating algorithms using resonance frequency difference, and concluded that differences in frequencies may not be sufficient enough for accurately identifying structural parameter values [11]. As a result, other modal properties, e.g. mode shapes or modal flexibility, were investigated for model updating. For example, Moller and Friberg adopted the modal assurance criterion (MAC)-related function for updating the model of an industrial structure, in attempt to make the updated FE model generate mode shapes that are closer to those extracted from experimental measurements [12]. FE model updating using differences in simulated and experimental mode shapes and frequencies was also applied to damage assessment of a reinforced concrete beam [13]. For damage assessment, the updated FE model for the undamaged structure is first taken as "healthy" baseline. If later model updating of the structure shows stiffness reduction of some structural components, the reduction can potentially be caused by damage (e.g. corrosion-induced section loss). Aiming at practical applications, Yuen developed an efficient model updating algorithm using frequencies and mode shapes at only a few selected degrees-of-freedom (DOFs) for the first few modes [14].

This research focuses on frequency-domain model updating approaches, in particular the modal property difference approaches. With selected updating parameters as optimization variables, the frequency-domain approaches can be formulated as optimization problems. Accordingly, the problem can potentially be solved using commercial optimization software, such as the MATLAB optimization toolbox [15]. This research studies two modal property difference formulations, one using MAC values and the other using direct differences between eigenvectors. Two local optimization algorithms, i.e. Levenberg-Marquardt [16] and trust-region-reflective [17], are adopted to solve the optimization problems. In general, such optimization problems are nonconvex. Thus, the optimal structural parameter values found by the local optimization algorithms usually cannot guarantee the global optimality of the solution [18-20]. To increase the chance of finding the global optimum of the optimization problem, randomly generated starting values of the updating variables are selected to initiate the optimization search. Finally, the presented model updating approaches with different optimization algorithms are validated with a few numerical examples.

The remainder of the paper is organized as follows. Section 2 presents the optimization formulation of the two modal property difference approaches, one using MAC values and the other using eigenvector differences. For each formulation, the Jacobian derivative of the objective function with respect to updating variables is derived. In this research, a novel way of obtaining the Jacobian derivative of eigenvector with respect to the updating variables is presented. Section 3 introduces the two local optimization algorithms used to solve the optimization problems, i.e. the Levenberg-Marquardt and trust-region-reflective. Section 4 shows the numerical examples for investigating the performance of the different optimization algorithms when applied to the two model updating formulations. Finally, a summary and discussion are provided.

2 Finite element model updating formulations

This section first introduces two modal property difference formulations for finite element (FE) model updating. In addition, to facilitate the optimization process, the Jacobian derivative of the objective function for each formulation with respect to optimization variables is also derived.

2.1 Modal property difference formulations

In order to update selected stiffness parameters of a linear structure, a vector variable, $\boldsymbol{\alpha} \in \mathbb{R}^{n_\alpha}$, is formed to contain the corresponding updating variables. The j-th ($j = 1 \ldots n_\alpha$) entry of $\boldsymbol{\alpha}$, α_j, corresponds to a parameter such as Young's modulus, or support spring stiffness. Each α_j is to be treated as an optimization variable in the optimization problem that attempts to minimize the difference between experimental and simulated modal properties. In this study, $\boldsymbol{\alpha}$ is scaled to represent the relative change percentage from nominal value of each stiffness parameter. As a result, a value of $\alpha_j = 0$ means the parameter takes the same value as the nominal one; a value of $\alpha_j = -0.5$ means a 50% reduction from the nominal value. The stiffness matrix can be formulated as an affine matrix function of the vector variable $\boldsymbol{\alpha}$:

$$\mathbf{K}(\boldsymbol{\alpha}) = \mathbf{K_0} + \sum_{j=1}^{n_\alpha} \alpha_j \mathbf{K}_j \qquad 1$$

Here $\mathbf{K}(\boldsymbol{\alpha}): \mathbb{R}^{n_\alpha} \to \mathbb{R}^{N \times N}$ represents an affine matrix function of $\boldsymbol{\alpha}$; N denotes the number of degrees of freedom (DOFs) of the structure; $\mathbf{K}_0$ is the nominal stiffness matrix prior to model updating, usually generated based on design drawings and nominal material properties; $\mathbf{K}_j$ is a constant influence matrix of the j-th stiffness parameter being updated, which corresponds to the updating variable α_j. In this study, it is assumed that the structural mass matrix ($\mathbf{M}$) is accurate enough and does not require updating. When needed, a similar formulation can be constructed to update mass parameters such as density values.

The modal property difference approach is usually formulated as an optimization problem that minimizes the difference between experimental and simulated modal properties. Using dynamic testing data collected from an as-built structure, usually the first few resonance frequencies, $\omega_i^{\mathrm{EXP}}, i = 1 \ldots n_{\mathrm{modes}}$, and the corresponding mode shapes can be extracted. Here n_{modes} denotes the number of experimentally measured modes. For each ω_i^{EXP}, an "experimental eigenvalue" is easily calculated as $\lambda_i^{\mathrm{EXP}} = \left(\omega_i^{\mathrm{EXP}}\right)^2$. For each mode shape eigenvector, the experimental data only provides entries for the DOFs with sensor instrumentation, i.e. $\boldsymbol{\psi}_i^{\mathrm{EXP,m}} \in \mathbb{R}^{n_{\mathrm{m}}}$ for the measured DOFs. On the other hand, the simulated modal properties (λ_i and $\boldsymbol{\psi}_i \in \mathbb{R}^N$) can be generated by the FE model. For example, for certain updating variable $\boldsymbol{\alpha}$, the stiffness matrix $\mathbf{K}(\boldsymbol{\alpha})$ is first assembled using Eq. 1. The simulated eigenvalues and eigenvectors, λ_i and $\boldsymbol{\psi}_i$, are the solution of the generalized eigenvalue problem:

$$[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i \mathbf{M}]\{\boldsymbol{\psi}_i\} = \mathbf{0} \qquad 2$$

In the sense that λ_i and $\boldsymbol{\psi}_i$ implicitly depend on $\boldsymbol{\alpha}$, they are written as functions of $\boldsymbol{\alpha}$ hereinafter, as $\lambda_i(\boldsymbol{\alpha}): \mathbb{R}^{n_\alpha} \to \mathbb{R}$ and $\boldsymbol{\psi}_i(\boldsymbol{\alpha}): \mathbb{R}^{n_\alpha} \to \mathbb{R}^N$. To reflect the measured DOFs in the formulation, define $\boldsymbol{\psi}_i(\boldsymbol{\alpha}) = [\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha}) \quad \boldsymbol{\psi}_i^{\mathrm{u}}(\boldsymbol{\alpha})]^{\mathrm{T}}$, where $\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha}) \in \mathbb{R}^{n_{\mathrm{m}}}$ corresponds to DOFs that are measured/instrumented, and $\boldsymbol{\psi}_i^{\mathrm{u}}(\boldsymbol{\alpha}) \in \mathbb{R}^{n_{\mathrm{u}}}$ corresponds to the unmeasured DOFs. Note that $n_{\mathrm{m}} + n_{\mathrm{u}} = N$, the total number of DOFs. The entry in $\boldsymbol{\psi}_i^{\mathrm{EXP,m}}$ with the largest magnitude is denoted the q_i-th entry ($1 \le q_i \le n_{\mathrm{m}}$), and $\boldsymbol{\psi}_i^{\mathrm{EXP,m}}$ is normalized so that the q_i-th entry equals 1. Correspondingly, the simulated eigenvector, $\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})$, is also normalized so that the q_i-th entry equals 1.

2.1.1 Modal property difference formulation with MAC values

The first modal property difference approach studied in this research is proposed by Moller and Friberg [12], where the (vector) optimization variable $\boldsymbol{\alpha}$ corresponds to stiffness parameters to be updated.

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \sum_{i=1}^{n_{\text{modes}}} \left\{ \left(\frac{\lambda_i^{\text{EXP}} - \lambda_i(\boldsymbol{\alpha})}{\lambda_i^{\text{EXP}}} \cdot w_{\lambda_i} \right)^2 + \left(\frac{1 - \sqrt{\text{MAC}_i(\boldsymbol{\alpha})}}{\sqrt{\text{MAC}_i(\boldsymbol{\alpha})}} \cdot w_{\boldsymbol{\psi}_i} \right)^2 \right\} \tag{3(a)}$$

$$\text{subject to} \quad \boldsymbol{L}_{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \boldsymbol{U}_{\boldsymbol{\alpha}} \tag{3(b)}$$

where w_{λ_i} represents the weighting factor of the eigenvalue difference; $w_{\boldsymbol{\psi}_i}$ represents the weighting factor of the eigenvector difference; $\boldsymbol{L}_{\boldsymbol{\alpha}}$ and $\boldsymbol{U}_{\boldsymbol{\alpha}} \in \mathbb{R}^{n_{\boldsymbol{\alpha}}}$ denote the lower and upper bounds for the optimization variable vector $\boldsymbol{\alpha}$, respectively. The sign "$\leq$" is overloaded to represent element-wise inequality; $\text{MAC}_i(\boldsymbol{\alpha})$ represents the modal assurance criterion between the *i*-th experimental and simulated mode shapes/eigenvectors at measured DOFs, i.e. $\boldsymbol{\psi}_i^{\text{EXP,m}}$ and $\boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha})$ [21].

$$\text{MAC}_i(\boldsymbol{\alpha}) = \frac{\left(\left(\boldsymbol{\psi}_i^{\text{EXP,m}}\right)^{\text{T}} \boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha}) \right)^2}{\left\| \boldsymbol{\psi}_i^{\text{EXP,m}} \right\|_2^2 \left\| \boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha}) \right\|_2^2}, i = 1 \dots n_{\text{modes}} \tag{4}$$

Here $\|\cdot\|_2$ denotes the $\mathcal{L}_2$-norm of a vector. Ranging from 0 to 1, the MAC value represents the similarity between two vectors. When two vectors are collinear, the MAC value is close to 1. When two vectors are orthogonal, the MAC value is close to 0. Normally, the numerical optimization algorithms are iterative. At every iteration step, the algorithm recalculates the value of the objective function, using the updated value of $\boldsymbol{\alpha}$ at current step. Specifically, the stiffness matrix $\mathbf{K}(\boldsymbol{\alpha})$ is first assembled using the current value of $\boldsymbol{\alpha}$ (Eq. 1). Then, the simulated modal properties, i.e. $\lambda_i(\boldsymbol{\alpha})$ and $\boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha})$, are obtained by solving the generalized eigenvalue problem using the new matrix $\mathbf{K}(\boldsymbol{\alpha})$, as shown in Eq. 2. Finally, an updated set of simulated eigenvalues and eigenvectors from FE model are used to evaluate the objective function in Eq. 3(a). Using nomenclature in optimization, the objective function is an oracle form of updating variable $\boldsymbol{\alpha}$, where the function can be evaluated for any feasible $\boldsymbol{\alpha}$, but has no explicit form. This objective function is generally nonconvex [18]. As a result, off-the-shelf local optimization algorithms usually cannot guarantee the global optimality. To increase the chance of finding the global minimum, optimization process will be initiated from multiple starting values randomized within the bounds.

2.1.2 Modal property difference formulation with eigenvector differences

Still using $\boldsymbol{\alpha}$ as the optimization variables, the second modal property difference formulation is directly based upon the differences between the mode shape/eigenvector entries at the measured DOFs.

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \sum_{i=1}^{n_{\text{modes}}} \left\{ \left(\frac{\lambda_i^{\text{EXP}} - \lambda_i(\boldsymbol{\alpha})}{\lambda_i^{\text{EXP}}} \cdot w_{\lambda_i} \right)^2 + \left\| \mathbf{Q}_i \left\{ \boldsymbol{\psi}_i^{\text{EXP,m}} - \boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha}) \right\} \cdot w_{\boldsymbol{\psi}_i} \right\|_2^2 \right\} \tag{5(a)}$$

$$\text{subject to} \quad \boldsymbol{L}_{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \boldsymbol{U}_{\boldsymbol{\alpha}} \tag{5(b)}$$

The selection matrix $\mathbf{Q}_i \in \mathbb{R}^{(n_{\text{m}}-1) \times n_{\text{m}}}$ is define as

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I}_{q_i - 1} & \mathbf{0}_{(q_i-1)\times 1} & \mathbf{0}_{(q_i-1)\times(n_{\text{m}}-q_i)} \\ \mathbf{0}_{(n_{\text{m}}-q_i)\times(q_i-1)} & \mathbf{0}_{(n_{\text{m}}-q_i)\times 1} & \mathbf{I}_{n_{\text{m}}-q_i} \end{bmatrix} \tag{6}$$

where $\mathbf{I}_{q_i-1}$ and $\mathbf{I}_{n_{\text{m}}-q_i}$ denote identity matrices with size of $q_i - 1$ and $n_{\text{m}} - q_i$, respectively. It can be seen that upon the aforementioned normalization of $\boldsymbol{\psi}_i^{\text{EXP,m}}$ and $\boldsymbol{\psi}_i^{\text{m}}(\boldsymbol{\alpha})$, Eq. 5 directly minimizes the differences between the entries in the experimental and simulated eigenvectors at measured DOFs (except for the q_i-th entry). This is different from using MAC values to quantify the vector difference as shown in Eq. 3(a). Similar to Eq. 3(a), the objective function in Eq. 5(a) is also in oracle formulation of updating

variable $\boldsymbol{\alpha}$. The formulation is generally nonconvex as well. As a result, local optimization algorithm usually cannot guarantee the global optimality.

2.2 Jacobian derivative of modal property difference formulations

As mentioned in Section 2.1, the numerical optimization algorithms solving Eq. 3 and 5 are iterative. At every iteration step, the Jacobian derivative (short-named as Jacobian) of the objective function is often used to determine the search direction. For an objective function $f(\mathbf{x})$: $\mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}$, the Jacobian is defined as $\mathrm{D}_\mathbf{x} f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_{n_\mathbf{x}}}\right] \in \mathbb{R}^{1\times n_\mathbf{x}}$. In this subsection, the Jacobian of the objective function shown in Eq. 3 and 5 will the derived.

2.2.1 Jacobian of MAC value formulation

To facilitate the Jacobian derivation, the objective function in Eq. 3 needs some rewriting. A residual vector function, $\mathbf{r}(\boldsymbol{\alpha})$: $\mathbb{R}^{n_\boldsymbol{\alpha}} \to \mathbb{R}^{2\cdot n_{\text{modes}}}$ is first defined as follows,

$$\mathbf{r}(\boldsymbol{\alpha}) = [\mathbf{r}_1(\boldsymbol{\alpha}) \quad \mathbf{r}_2(\boldsymbol{\alpha}) \quad \ldots \quad \mathbf{r}_{n_{\text{modes}}}(\boldsymbol{\alpha})]^{\mathrm{T}} \tag{7}$$

where $\mathbf{r}_i(\boldsymbol{\alpha})$: $\mathbb{R}^{n_\boldsymbol{\alpha}} \to \mathbb{R}^2$ equals $\left[\frac{\lambda_i^{\mathrm{EXP}} - \lambda_i(\boldsymbol{\alpha})}{\lambda_i^{\mathrm{EXP}}} \cdot w_{\lambda_i} \quad \frac{1-\sqrt{\mathrm{MAC}_i(\boldsymbol{\alpha})}}{\sqrt{\mathrm{MAC}_i(\boldsymbol{\alpha})}} \cdot w_{\boldsymbol{\psi}_i}\right]^{\mathrm{T}}, i = 1 \ldots n_{\text{modes}}$. Using $\mathbf{r}(\boldsymbol{\alpha})$, the optimization problem in Eq. 3 is equivalent to

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad f(\boldsymbol{\alpha}) = \mathbf{r}(\boldsymbol{\alpha})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\alpha}) \tag{8(a)}$$

$$\text{subject to} \quad \boldsymbol{L}_\boldsymbol{\alpha} \le \boldsymbol{\alpha} \le \boldsymbol{U}_\boldsymbol{\alpha} \tag{8(b)}$$

The Jacobian for $f(\boldsymbol{\alpha})$ in Eq. 8(a), $\mathrm{D}_\boldsymbol{\alpha} f \in \mathbb{R}^{1\times n_\boldsymbol{\alpha}}$, equals $\mathrm{D}_\mathbf{r} f \cdot \mathrm{D}_\boldsymbol{\alpha}\mathbf{r}$ by using the chain rule. The first entry is $\mathrm{D}_\mathbf{r} f = 2\mathbf{r}^{\mathrm{T}} \in \mathbb{R}^{1\times(2\cdot n_{\text{modes}})}$. The second entry is $\mathrm{D}_\boldsymbol{\alpha}\mathbf{r} = [\mathrm{D}_\boldsymbol{\alpha}\mathbf{r}_1 \quad \mathrm{D}_\boldsymbol{\alpha}\mathbf{r}_2 \quad \cdots \quad \mathrm{D}_\boldsymbol{\alpha}\mathbf{r}_{n_{\text{modes}}}]^{\mathrm{T}} \in \mathbb{R}^{(2\cdot n_{\text{modes}})\times n_\boldsymbol{\alpha}}$. Recall the definition of MAC value in Eq. 4, each $\mathrm{D}_\boldsymbol{\alpha}\mathbf{r}_i \in \mathbb{R}^{2\times n_\boldsymbol{\alpha}}$ can be formed as:

$$\mathrm{D}_\boldsymbol{\alpha}\mathbf{r}_i = \begin{bmatrix} -\frac{w_{\lambda_i}}{\lambda_i^{\mathrm{EXP}}} \cdot \mathrm{D}_\boldsymbol{\alpha}\left(\lambda_i(\boldsymbol{\alpha})\right) \\ \left(\frac{-w_{\boldsymbol{\psi}_i}}{\sqrt{\mathrm{MAC}_i(\boldsymbol{\alpha})}}\right)\left(\frac{\left(\boldsymbol{\psi}_i^{\mathrm{EXP,m}}\right)^{\mathrm{T}}}{\left(\boldsymbol{\psi}_i^{\mathrm{EXP,m}}\right)^{\mathrm{T}}\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})} - \frac{\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right)^{\mathrm{T}}}{\left\|\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right\|_2^2}\right)\mathrm{D}_\boldsymbol{\alpha}\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right) \end{bmatrix}, i = 1 \ldots n_{\text{modes}} \tag{9}$$

The formulation for $\mathrm{D}_\boldsymbol{\alpha}\left(\lambda_i(\boldsymbol{\alpha})\right) \in \mathbb{R}^{1\times n_\boldsymbol{\alpha}}$ and $\mathrm{D}_\boldsymbol{\alpha}\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right) \in \mathbb{R}^{n_{\mathrm{m}}\times n_\boldsymbol{\alpha}}$ have been well studied by researchers [22, 23]. Nevertheless, a simplified way of obtaining $\mathrm{D}_\boldsymbol{\alpha}\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right)$ based on the normalization of $\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})$ is presented, without expressing the derivative as a linear combination of all the eigenvectors (as in [23]). Recall the generalized eigenvalue equation for the *i*-th mode:

$$[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i\mathbf{M}]\{\boldsymbol{\psi}_i\} = \mathbf{0} \tag{10}$$

By differentiating Eq. 10 with respect to the *j*-th updating variable, α_j, the following equation can be obtained.

$$[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i\mathbf{M}]\frac{\partial\boldsymbol{\psi}_i}{\partial\alpha_j} = \frac{\partial\lambda_i}{\partial\alpha_j}\mathbf{M}\boldsymbol{\psi}_i - \mathbf{K}_j\boldsymbol{\psi}_i \tag{11}$$

Assume that the eigenvalues are distinct, and define the modal mass of the *i*-th mode as $m_i = (\boldsymbol{\psi}_i)^{\mathrm{T}}\mathbf{M}\boldsymbol{\psi}_i$. Then, pre-multiply Eq. 11 by $(\boldsymbol{\psi}_i)^{\mathrm{T}}$, and note $(\boldsymbol{\psi}_i)^{\mathrm{T}}[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i\mathbf{M}] = \mathbf{0}$. Eq. 11 can be simplified as follows, and $\partial\lambda_i/\partial\alpha_j \in \mathbb{R}$ can be obtained.

$$(\boldsymbol{\psi}_i)^{\mathrm{T}}\mathbf{K}_j\boldsymbol{\psi}_i = \frac{\partial\lambda_i}{\partial\alpha_j}(\boldsymbol{\psi}_i)^{\mathrm{T}}\mathbf{M}\boldsymbol{\psi}_i \tag{12(a)}$$

$$\frac{\partial \lambda_i}{\partial \alpha_j} = \frac{(\boldsymbol{\psi}_i)^{\mathrm{T}} \mathbf{K}_j \boldsymbol{\psi}_i}{m_i} \tag{12(b)}$$

As a result, Jacobian of the i-th simulated eigenvalue, $\mathrm{D}_{\boldsymbol{\alpha}}(\lambda_i) \in \mathbb{R}^{1\times n_{\boldsymbol{\alpha}}}$, with respect to updating vector, $\boldsymbol{\alpha}$, can be found as follows:

$$\mathrm{D}_{\boldsymbol{\alpha}}(\lambda_i) = \begin{bmatrix} \frac{\partial \lambda_i}{\partial \alpha_1} & \frac{\partial \lambda_i}{\partial \alpha_2} & \cdots & \frac{\partial \lambda_i}{\partial \alpha_{n_{\boldsymbol{\alpha}}}} \end{bmatrix} = \begin{bmatrix} \frac{(\boldsymbol{\psi}_i)^{\mathrm{T}} \mathbf{K}_1 \boldsymbol{\psi}_i}{m_i} & \frac{(\boldsymbol{\psi}_i)^{\mathrm{T}} \mathbf{K}_2 \boldsymbol{\psi}_i}{m_i} & \cdots & \frac{(\boldsymbol{\psi}_i)^{\mathrm{T}} \mathbf{K}_{n_{\boldsymbol{\alpha}}} \boldsymbol{\psi}_i}{m_i} \end{bmatrix} \tag{13}$$

After obtaining $\partial \lambda_i / \partial \alpha_j$, Eq. 11 is reused to find the only remaining unknown term, $\partial \boldsymbol{\psi}_i / \partial \alpha_j \in \mathbb{R}^N$. However, $\partial \boldsymbol{\psi}_i / \partial \alpha_j$ cannot be directly obtained from Eq. 11, because $[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i \mathbf{M}]$ is rank deficient by one assuming that the eigenvalue λ_i is distinct. Nevertheless, as previously mentioned, $\boldsymbol{\psi}_i^{\mathrm{m}}$ is normalized so that the q_i-th entry always equals a constant 1. As a result, the q_i-th entry in vector $\partial \boldsymbol{\psi}_i^{\mathrm{m}} / \partial \alpha_j$ is zero, i.e. $\partial \psi_{q_i,i}^{\mathrm{m}} / \partial \alpha_j = 0$. Because of the separation by measured and unmeasured DOFs, $\boldsymbol{\psi}_i(\boldsymbol{\alpha}) = [\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha}) \quad \boldsymbol{\psi}_i^{\mathrm{u}}(\boldsymbol{\alpha})]^{\mathrm{T}}$, the q_i-th entry in $\partial \boldsymbol{\psi}_i / \partial \alpha_j$ is also zero, i.e. $\partial \psi_{q_i,i} / \partial \alpha_j = 0$. This is utilized to resolve the rank deficiency issue of $[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i \mathbf{M}]$. Specifically, define $\mathbf{P}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{0}_{(n_{\mathrm{m}}-1)\times n_{\mathrm{u}}} \\ \mathbf{0}_{n_{\mathrm{u}}\times n_{\mathrm{m}}} & \mathbf{I}_{n_{\mathrm{u}}} \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}$, which extends $\mathbf{Q}_i$ in Eq. 6 from measured DOFs to all DOFs. Then, pre-multiplying and post-multiplying $[\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i \mathbf{M}]$ in Eq. 11 by $\mathbf{P}_i$ and $\mathbf{P}_i^{\mathrm{T}}$ to cross out the q_i-th row and q_i-th column, $\mathbf{B}_i \in \mathbb{R}^{(N-1)\times(N-1)}$ is generated.

$$\mathbf{B}_i = \mathbf{P}_i [\mathbf{K}(\boldsymbol{\alpha}) - \lambda_i \mathbf{M}] \mathbf{P}_i^{\mathrm{T}} \tag{14}$$

Next, pre-multiply $\left(\frac{\partial \lambda_i}{\partial \alpha_j} \mathbf{M}\boldsymbol{\psi}_i - \mathbf{K}_j \boldsymbol{\psi}_i\right)$ in Eq. 11 by $\mathbf{P}_i$ to eliminate the q_i-th row and obtain $\mathbf{b}_{ij} \in \mathbb{R}^{N-1}$:

$$\mathbf{b}_{ij} = \mathbf{P}_i \cdot \left(\frac{\partial \lambda_i}{\partial \alpha_j} \mathbf{M}\boldsymbol{\psi}_i - \mathbf{K}_j \boldsymbol{\psi}_i\right) \tag{15}$$

Finally, recalling $\partial \psi_{q_i,i} / \partial \alpha_j = 0$, the elimination of the q_i-th row in Eq. 11 is equivalent to the following.

$$\mathbf{B}_i \begin{Bmatrix} \frac{\partial (\mathbf{Q}_i \boldsymbol{\psi}_i^{\mathrm{m}})}{\partial \alpha_j} \\ \frac{\partial (\boldsymbol{\psi}_i^{\mathrm{u}})}{\partial \alpha_j} \end{Bmatrix} = \mathbf{b}_{ij} \tag{16}$$

Thus, the Jacobian of the i-th simulated eigenvector with respect to the updating variables can be shown as:

$$\begin{Bmatrix} \frac{\partial (\mathbf{Q}_i \boldsymbol{\psi}_i^{\mathrm{m}})}{\partial \alpha_j} \\ \frac{\partial (\boldsymbol{\psi}_i^{\mathrm{u}})}{\partial \alpha_j} \end{Bmatrix} = \mathbf{B}_i^{-1} \mathbf{b}_{ij} \tag{17}$$

In summary, $\partial \boldsymbol{\psi}_i^{\mathrm{m}} / \partial \alpha_j$ is 0 at q_i-th entry and other entries are provided by the equation above. The Jacobian of the simulated eigenvector at measured DOFs, $\mathrm{D}_{\boldsymbol{\alpha}}(\boldsymbol{\psi}_i^{\mathrm{m}}) \in \mathbb{R}^{n_{\mathrm{m}}\times n_{\boldsymbol{\alpha}}}$ in Eq. 9, can be obtained as follows:

$$\mathbf{D}_{\boldsymbol{\alpha}}(\boldsymbol{\psi}_i^{\mathrm{m}}) = \begin{bmatrix} \dfrac{\partial \boldsymbol{\psi}_i^{\mathrm{m}}}{\partial \alpha_1} & \dfrac{\partial \boldsymbol{\psi}_i^{\mathrm{m}}}{\partial \alpha_2} & \cdots & \dfrac{\partial \boldsymbol{\psi}_i^{\mathrm{m}}}{\partial \alpha_{n_{\boldsymbol{\alpha}}}} \end{bmatrix} = \begin{bmatrix} \partial \psi_{1,i}^{\mathrm{m}}/\partial \alpha_1 & \partial \psi_{1,i}^{\mathrm{m}}/\partial \alpha_2 & \cdots & \partial \psi_{1,i}^{\mathrm{m}}/\partial \alpha_{n_{\boldsymbol{\alpha}}} \\ \vdots & \vdots & \cdots & \vdots \\ \partial \psi_{q_i-1,i}^{\mathrm{m}}/\partial \alpha_1 & \partial \psi_{q_i-1,i}^{\mathrm{m}}/\partial \alpha_2 & \cdots & \partial \psi_{q_i-1,i}^{\mathrm{m}}/\partial \alpha_{n_{\boldsymbol{\alpha}}} \\ 0 & 0 & \cdots & 0 \\ \psi_{q_i+1,i}^{\mathrm{m}}/\partial \alpha_1 & \partial \psi_{q_i+1,i}^{\mathrm{m}}/\partial \alpha_2 & \cdots & \partial \psi_{q_i+1,i}^{\mathrm{m}}/\partial \alpha_{n_{\boldsymbol{\alpha}}} \\ \vdots & \vdots & \cdots & \vdots \\ \psi_{n_{\mathrm{m}},i}^{\mathrm{m}}/\partial \alpha_1 & \psi_{n_{\mathrm{m}},i}^{\mathrm{m}}/\partial \alpha_2 & \cdots & \psi_{n_{\mathrm{m}},i}^{\mathrm{m}}/\partial \alpha_{n_{\boldsymbol{\alpha}}} \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ q_i - 1 \\ q_i \\ q_i + 1 \\ \vdots \\ n_{\mathrm{m}} \end{matrix} \tag{18}$$

After obtaining the Jacobian of simulated eigenvalue and eigenvector at measured DOFs, $\mathrm{D}_{\boldsymbol{\alpha}}(\lambda_i)$ and $\mathrm{D}_{\boldsymbol{\alpha}}(\boldsymbol{\psi}_i^{\mathrm{m}})$, the analytical Jacobian in Eq. 9 can be calculated.

2.2.2 Jacobian of eigenvector difference formulation

Similar to the method introduced in Section 2.2.1, in order to derive the Jacobian of the eigenvector difference formulation in Eq. 5, a residual vector, $\mathbf{r}(\boldsymbol{\alpha}) : \mathbb{R}^{n_{\boldsymbol{\alpha}}} \to \mathbb{R}^{n_{\mathrm{m}} \cdot n_{\mathrm{modes}}}$ is defined as follows

$$\mathbf{r}(\boldsymbol{\alpha}) = [\mathbf{r}_1(\boldsymbol{\alpha}) \quad \mathbf{r}_2(\boldsymbol{\alpha}) \quad \ldots \quad \mathbf{r}_{n_{\mathrm{modes}}}(\boldsymbol{\alpha})]^{\mathrm{T}} \tag{19}$$

where $\mathbf{r}_i(\boldsymbol{\alpha}) : \mathbb{R}^{n_{\boldsymbol{\alpha}}} \to \mathbb{R}^{n_{\mathrm{m}}}$ equals $\begin{bmatrix} \left(\lambda_i^{\mathrm{EXP}} - \lambda_i(\boldsymbol{\alpha})\right)/\lambda_i^{\mathrm{EXP}} \cdot w_{\lambda_i} \\ \mathbf{Q}_i\{\boldsymbol{\psi}_i^{\mathrm{EXP,m}} - \boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\} \cdot w_{\boldsymbol{\psi}_i} \end{bmatrix}, i = 1 \ldots n_{\mathrm{modes}}$. Using $\mathbf{r}(\boldsymbol{\alpha})$, the optimization problem in Eq. 5 for the eigenvector difference formulation can also be rewritten the same as Eq. 8 (for the MAC value formulation), with the objective function $f(\boldsymbol{\alpha}) = \mathbf{r}(\boldsymbol{\alpha})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\alpha})$. Again, the Jacobian for $f(\boldsymbol{\alpha})$, $\mathrm{D}_{\boldsymbol{\alpha}} f \in \mathbb{R}^{1 \times n_{\boldsymbol{\alpha}}}$ equals $\mathrm{D}_{\mathbf{r}} f \cdot \mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}$ from the chain rule. However, the residual vector $\mathbf{r}$ has a different dimension from Eq. 8 for the MAC value formulation. For the eigenvector difference formulation, the first Jacobian entry is $\mathrm{D}_{\mathbf{r}} f = 2\mathbf{r}^{\mathrm{T}} \in \mathbb{R}^{1 \times (n_{\mathrm{m}} \cdot n_{\mathrm{modes}})}$. Meanwhile, the second entry is $\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r} = [\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}_1 \quad \mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}_2 \quad \cdots \quad \mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}_{n_{\mathrm{modes}}}]^{\mathrm{T}} \in \mathbb{R}^{(n_{\mathrm{m}} \cdot n_{\mathrm{modes}}) \times n_{\boldsymbol{\alpha}}}$, where each $\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}_i \in \mathbb{R}^{n_{\mathrm{m}} \times n_{\boldsymbol{\alpha}}}$ can be formed as follows:

$$\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}_i = \begin{bmatrix} -\dfrac{\mathrm{D}_{\boldsymbol{\alpha}}\left(\lambda_i(\boldsymbol{\alpha})\right)}{\lambda_i^{\mathrm{EXP}}} \cdot w_{\lambda_i} \\ -\mathbf{Q}_i \mathrm{D}_{\boldsymbol{\alpha}}\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right) \cdot w_{\boldsymbol{\psi}_i} \end{bmatrix}, i = 1 \ldots n_{\mathrm{modes}} \tag{20}$$

The Jacobian of the i-th simulated eigenvalue and eigenvector at measured DOF, $\mathrm{D}_{\boldsymbol{\alpha}}\left(\lambda_i(\boldsymbol{\alpha})\right)$ and $\mathrm{D}_{\boldsymbol{\alpha}}\left(\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right)$, have been introduced in Eqs. 13 and 18.

3 Optimization algorithms

Thus far, both modal property difference formulations are presented as an optimization problem. A number of optimization algorithms can be attempted towards solving the optimization problems. For example, MATLAB optimization toolbox supports various algorithms. However, because both of the optimization problems (Eqs. 3 and 5) are nonconvex, these off-the-shelf algorithms can only find local minima. While some can better the chance, none can guarantee the global optimality of the solution [18-20].

In this research, the `lsqnonlin` solver in MATLAB optimization toolbox [15] is adopted to numerically solve the optimization problems. The solver specializes on nonlinear least squares problems where the objective is to minimize the square of 2-norm of a residual vector $\mathbf{r} \in \mathbb{R}^{\mathrm{m}}$:

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \sum_{i=1}^{m} r_i^2(\mathbf{x}) = \|\mathbf{r}(\mathbf{x})\|_2^2 = \mathbf{r}(\mathbf{x})^{\mathrm{T}}\,\mathbf{r}(\mathbf{x}) \tag{21}$$

where $r_i(\mathbf{x})$: $\mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}$ as a residual function is usually nonlinear. Assembling the residuals in a vector form, $\mathbf{r}(\mathbf{x})$: $\mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}^m$ is defined as $[r_1(\mathbf{x}) \quad r_2(\mathbf{x}) \quad \cdots \quad r_m(\mathbf{x})]^\mathrm{T}$. The gradient, $\nabla f(\mathbf{x}) \in \mathbb{R}^{n_\mathbf{x}}$ and Hessian, $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{n_\mathbf{x} \times n_\mathbf{x}}$ of $f(\mathbf{x})$: $\mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}$ can be expressed as follows:

$$\nabla f(\mathbf{x}) = 2[\mathrm{D}_\mathbf{x}\mathbf{r}]^\mathrm{T} \cdot \mathbf{r}(\mathbf{x}) \tag{22(a)}$$

$$\nabla^2 f(\mathbf{x}) = 2[\mathrm{D}_\mathbf{x}\mathbf{r}]^\mathrm{T} \cdot \mathrm{D}_\mathbf{x}\mathbf{r} + 2\sum_{i=1}^{m} r_i(\mathbf{x})\, \nabla^2 r_i \tag{22(b)}$$

where $\mathrm{D}_\mathbf{x}\mathbf{r} \in \mathbb{R}^{m \times n_\mathbf{x}}$ is defined as the Jacobian matrix of the scalar residuals ($r_i, i = 1 \cdots m$) with respect to the optimization variables ($x_j, j = 1 \cdots n_\mathbf{x}$). Neglecting the higher-order second term in $\nabla^2 f(\mathbf{x})$, the optimization algorithms adopted by `lsqnonlin` in MATLAB uses $2[\mathrm{D}_\mathbf{x}\mathbf{r}]^\mathrm{T} \cdot \mathrm{D}_\mathbf{x}\mathbf{r}$ to approximate the Hessian matrix.

From certain starting points, the `lsqnonlin` solver can find a local minimum of the objective function through the Levenberg-Marquardt algorithm, which is a combination of the steepest descent and the Gauss-Newton algorithm [16]. At every iteration, the algorithm first linearizes the objective function (Eq. 21) with respective to the corresponding optimization variables. When the current solution is far from a local optimum, the Levenberg-Marquardt algorithm approaches the steepest descent algorithm. On the other hand, when the current solution is close to a local optimum, the Levenberg-Marquardt algorithm approaches the Gauss-Newton algorithm. The Levenberg-Marquardt algorithm can be used to solve both optimization problems in Eqs. 3 and 5, i.e. both the MAC value and eigenvector difference formulations. The drawback of the Levenberg-Marquardt implementation in MATLAB is that it does not allow setting the upper and lower bounds of the optimization variables.

In addition to the Levenberg-Marquardt algorithm, `lsqnonlin` solver also provides the trust-region-reflective algorithm to solve an optimization problem [17]. The trust-region-reflective algorithm approximates the original problem with a quadratic subproblem within a small region around the current solution point, i.e. a trusted region. The quadratic subproblem is formulated using the same gradient and approximated Hessian of the original problem. By solving the quadratic subproblem using the two-dimensional subspace approach, a solution of current subproblem can be obtained [24, 25]. If the decrease of the objective function evaluated at current step is within the prescribed upper and lower bounds, the solution will be accepted, and the algorithm will continue with the next iteration. Otherwise, the trusted region at the current iteration will be adjusted, and the quadratic subproblem is solved again with the new region. Iteratively, the optimization converges to a local minimum of the objective function. The advantage of the trust-region-reflective implementation in MATLAB is that it allows user to define the upper and lower bounds of the optimization variables. However, the trust-region-reflective algorithm implemented in MATLAB cannot solve underdetermined problems. For the algorithm to work, the length of residual vector, *m*, should be at least as large as the number of variables $n_\mathbf{x}$ (Eq. 21).

From Eqs. 3 and 5, the optimization problem of both modal property difference formulations can be equivalently rewritten to satisfy the least-squares format required for the `lsqnonlin` solver. When using MAC value formulation in Eq. 3, the optimization variable $\mathbf{x}$ is the updating vector variable $\boldsymbol{\alpha}$. So that the MAC value formulation in Eq. 3(a) is rewritten in least squares form as $f(\boldsymbol{\alpha}) = \|\mathbf{r}(\boldsymbol{\alpha})\|_2^2$, the residual vector shown in Eq. 21, $\mathbf{r}(\boldsymbol{\alpha})$: $\mathbb{R}^{n_\alpha} \to \mathbb{R}^{2 \cdot n_\mathrm{modes}}$, is formulated as a function of variable $\boldsymbol{\alpha}$. The length of the residual vector is $m = 2 \cdot n_\mathrm{modes}$. The formulation of $\mathbf{r}(\boldsymbol{\alpha})$, which is previously given in Eq. 7, is repeated as follows in an explicit form.

$$\mathbf{r}(\boldsymbol{\alpha}) = \begin{bmatrix} \left(\lambda_1^{\mathrm{EXP}} - \lambda_1(\boldsymbol{\alpha})\right)/\lambda_1^{\mathrm{EXP}} \cdot w_{\lambda_1} \\ \left(1 - \sqrt{\mathrm{MAC}_1(\boldsymbol{\alpha})}\right)/\sqrt{\mathrm{MAC}_1(\boldsymbol{\alpha})} \cdot w_{\boldsymbol{\psi}_1} \\ \vdots \\ \left(\lambda_{n_{\mathrm{modes}}}^{\mathrm{EXP}} - \lambda_{n_{\mathrm{modes}}}(\boldsymbol{\alpha})\right)/\lambda_{n_{\mathrm{modes}}}^{\mathrm{EXP}} \cdot w_{\lambda_{n_{\mathrm{modes}}}} \\ \left(1 - \sqrt{\mathrm{MAC}_{n_{\mathrm{modes}}}(\boldsymbol{\alpha})}\right)\Big/\sqrt{\mathrm{MAC}_{n_{\mathrm{modes}}}(\boldsymbol{\alpha})} \cdot w_{\boldsymbol{\psi}_{n_{\mathrm{modes}}}} \end{bmatrix} \quad 23$$

When using the eigenvector difference formulation in Eq. 5, the optimization variable $\mathbf{x}$ is the updating vector variable $\boldsymbol{\alpha}$. For the eigenvector difference formulation in Eq. 5(a) to be rewritten in the form as $f(\boldsymbol{\alpha}) = \|\mathbf{r}(\boldsymbol{\alpha})\|_2^2$, the residual vector shown in Eq. 21, $\mathbf{r}(\boldsymbol{\alpha})$: $\mathbb{R}^{n_{\boldsymbol{\alpha}}} \to \mathbb{R}^{n_m \cdot n_{\mathrm{modes}}}$, is formulated as a function of $\boldsymbol{\alpha}$. The formulation is previously given in Eq. 19 and repeated as follows.

$$\mathbf{r}(\boldsymbol{\alpha}) = \begin{bmatrix} \left(\lambda_1^{\mathrm{EXP}} - \lambda_1(\boldsymbol{\alpha})\right)/\lambda_1^{\mathrm{EXP}} \cdot w_{\lambda_1} \\ \mathbf{Q}_1\left\{\boldsymbol{\psi}_1^{\mathrm{EXP,m}} - \boldsymbol{\psi}_1^{\mathrm{m}}(\boldsymbol{\alpha})\right\} \cdot w_{\boldsymbol{\psi}_1} \\ \vdots \\ \left(\lambda_{n_{\mathrm{modes}}}^{\mathrm{EXP}} - \lambda_{n_{\mathrm{modes}}}(\boldsymbol{\alpha})\right)/\lambda_{n_{\mathrm{modes}}}^{\mathrm{EXP}} \cdot w_{\lambda_{n_{\mathrm{modes}}}} \\ \mathbf{Q}_{n_{\mathrm{modes}}}\left\{\boldsymbol{\psi}_{n_{\mathrm{modes}}}^{\mathrm{EXP,m}} - \boldsymbol{\psi}_{n_{\mathrm{modes}}}^{\mathrm{m}}(\boldsymbol{\alpha})\right\} \cdot w_{\boldsymbol{\psi}_{n_{\mathrm{modes}}}} \end{bmatrix} \quad 24$$

Here each term $\mathbf{Q}_i\left\{\boldsymbol{\psi}_i^{\mathrm{EXP,m}} - \boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})\right\} \cdot w_{\boldsymbol{\psi}_i}$ is a $(n_m - 1) \times 1$ vector. As a result, the length of the residual vector is $m = n_m \cdot n_{\mathrm{modes}}$ (recall that n_m is the number of measured/instrumented DOFs).

Finally, at each step of optimization process, by default, `lsqnonlin` calculates the search gradient, $\nabla f(\mathbf{x})$, of the objective function numerically using the finite difference method [26]. The numerically calculated gradient results are affected by the difference $\Delta\mathbf{x}$, i.e. step size of $\mathbf{x}$, and more prone to inaccuracies. Meanwhile, instead of using the numerically calculated gradient, `lsqnonlin` also accepts user-provided analytical formulation of the gradient. Given that gradient simply equals the transpose of Jacobian, i.e. $\nabla f(\mathbf{x}) = (\mathrm{D}_{\mathbf{x}} f)^{\mathrm{T}}$, the definition of $\mathrm{D}_{\boldsymbol{\alpha}} f$ in Section 2.1.1 as well as $\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}$ in Eq. 9 can be used to calculate the analytical gradient for the MAC value formulation. Similarly, definition of $\mathrm{D}_{\boldsymbol{\alpha}} f$ in Section 2.2.2 as well as $\mathrm{D}_{\boldsymbol{\alpha}}\mathbf{r}$ in Eq. 20 can be used to calculate the analytical gradient for the eigenvector difference formulation.

4 Numerical Studies

To investigate the performance of modal property difference formulations for model updating, numerical studies are conducted. The first example is a steel pedestrian bridge, and the second one is a concrete building frame. Table 1 summarizes the applicable algorithms for the two optimization formulations (Eqs. 3 and 5). When solving the optimization problems of each formulations, both the Levenberg-Marquardt and the trust-region-reflective algorithms are adopted for comparison. For example, Case 2(b) means applying the trust-region-reflective algorithm onto the eigenvector difference formulation. For both algorithms, the effect of using numerical or analytical gradient during the optimization process will be compared through the concrete building frame. Furthermore, as mentioned in Section 2, both objective functions in Eqs. 3 and 5 are nonconvex, which means that global optimality of the solution cannot be guaranteed. To increase the chance of finding the global minimum, optimization process will be initiated

from many feasible and randomized starting values of the updating variables. Modal properties of the structure with actual/correct stiffness values of $\boldsymbol{\alpha}$ are used as the "experimental" properties, i.e. λ_i^{EXP} and $\boldsymbol{\psi}_i^{\mathrm{EXP,m}}$ in Eqs. 3 and 5. For practicality, only some of the DOFs are instrumented and only modes associated with the lowest few resonance frequencies are made available for model updating. In these numerical studies, the weighting factors for the eigenvector and eigenvalue are set as 1 for simplicity, i.e. $w_{\lambda_i} = w_{\boldsymbol{\psi}_i} = 1, i = 1 \ldots n_{\mathrm{modes}}$.

Table 1 Model updating algorithms for objective function in Eqs. 3 and 5

Optimization problem	Optimization algorithms	Case #
MAC value formulation (Eq. 3)	Levenberg-Marquardt trust-region-reflective	1(a) 1(b)
Eigenvector difference formulation (Eq. 5)	Levenberg-Marquardt trust-region-reflective	2(a) 2(b)

4.1 Steel Pedestrian Bridge

Figure 1 shows the FE model of a steel pedestrian bridge, which is based on a pedestrian bridge on Georgia Tech campus. Constructed in SAP2000, a commercial structural software package, the bridge model contains 46 nodes. For both of the two left-end notes, the longitudinal (x-direction) DOF are constrained, while a vertical spring (k_{z1}) is allocated to represent non-ideal boundary conditions. For the front one between the two nodes (Figure 1), a transverse spring (k_{y1}) is also allocated. Similarly, at the right-end side, a vertical spring (k_{z2}) is allocated at both nodes, and a transverse spring (k_{y2}) is allocated at the front one between the two nodes; both nodes are free in the longitudinal direction. In total, the FE model has 274 DOFs. Although mainly a frame structure, the segmental diagonal bracings in top plane and two side planes are truss members. The modeling software SAP2000 assigns non-zero concentrated mass only to the translational DOFs. As a result, the mass matrix (**M**) is a diagonal matrix whose diagonal entries associated with rotational DOFs equal zero. As shown in Figure 1, it is assumed that 7 uniaxial and 7 biaxial accelerometers are instrumented for model updating. The uniaxial accelerometer measures vertical vibration of the structure, and the biaxial accelerometer measures vertical and transverse vibration. In total, 21 out of the 274 DOFs are measured, i.e. the length of $\boldsymbol{\psi}_i^{\mathrm{EXP,m}}$ and $\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})$ is $n_{\mathrm{m}} = 21$.

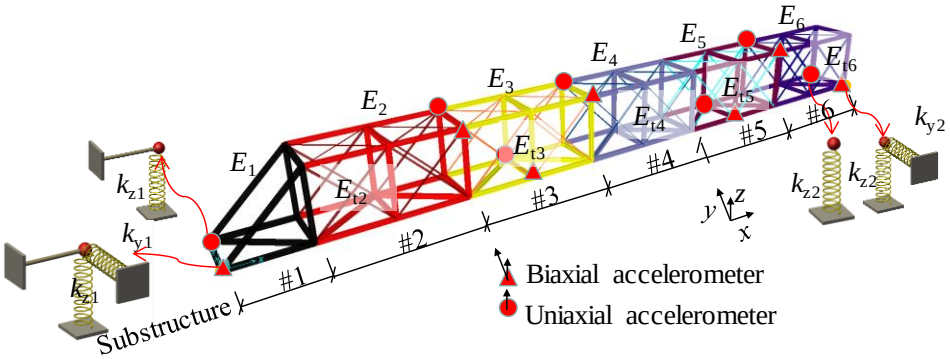


Figure 1. Steel pedestrian bridge and sensor instrumentation

It is assumed that the mass matrix (**M**) is accurate enough and does not require updating. Figure 1 shows how the entire structure is divided into six substructures for FE model updating. Substructure #1 contains only one segment from the left end of the bridge span. Other five substructures each contains two segments. Table 2 lists the stiffness parameters to be updated. The first substructure only contains frame members, and E_1 represents the elastic modulus of those frame members. From substructure 2 to substructure 6, each contains both frame and truss members. E_2 ~ E_6 represent the elastic moduli of frame members in each

substructure, and $E_{t2} \sim E_{t6}$ represent the elastic moduli of truss members in each substructure. The updated parameters also include the stiffness values of support springs (k_{y1}, k_{z1}, k_{y2} and k_{z2}). Table 2 lists the nominal and actual values of the stiffness parameters. In total, this model updating problem has 15 updating variables, i.e. $n_{\boldsymbol{\alpha}} = 15$. The column α_i^{act} in Table 2 lists the actual/correct values of $\boldsymbol{\alpha}$ (that are to be identified through model updating). For example, the correct solution of α_1 is calculated as the relative change of the actual E_1 value from its nominal value: $\alpha_1^{\text{act}} = \frac{E_1^{\text{act}} - E_1^{\text{nom}}}{E_1^{\text{nom}}} = \frac{30{,}450 - 29{,}000}{29{,}000} = 0.05$, i.e. a 5% increase from the nominal value.

Table 2 Structural properties of the steel pedestrian bridge

Stiffness parameters		Nominal value	Actual value	Updating variables	α_i^{act}
Elastic moduli of frame members (kips/in^2)	E_1	29,000	30,450	α_1	0.05
	E_2	29,000	30,450	α_2	0.05
	E_3	29,000	27,550	α_3	-0.05
	E_4	29,000	26,100	α_4	-0.10
	E_5	29,000	31,900	α_5	0.10
	E_6	29,000	24,650	α_6	-0.15
Elastic moduli of truss members (kips/in^2)	E_{t2}	29,000	33,350	α_7	0.15
	E_{t3}	29,000	27,550	α_8	-0.05
	E_{t4}	29,000	26,100	α_9	-0.10
	E_{t5}	29,000	31,900	α_{10}	0.10
	E_{t6}	29,000	23,200	α_{11}	0.20
Support spring stiffness (kips/in)	k_{y1}	400	280	α_{12}	-0.30
	k_{z1}	500	320	α_{13}	0.60
	k_{y2}	400	280	α_{14}	-0.30
	k_{z2}	500	320	α_{15}	0.60

For this steel pedestrian bridge, it is assumed that the first three vibration modes ($n_{\text{modes}} = 3$) are available for model updating. As described in Section 3, the Levenberg-Marquardt algorithm can be applied to underestimated problems. The algorithm is always applicable to both MAC value and eigenvector difference formulations, i.e. Case 1(a) and 2(a) in Table 1. On the other hand, the trust-region-reflective algorithm cannot solve underdetermined problems; the algorithm only works if the length of residual vector is no less than the number of optimization variables, i.e. $m \geq n_{\mathbf{x}}$ in Eq. 21. Consider first applying the algorithm on MAC value formulation, i.e. Case 1(b) in Table 1. As presented in Eq. 23, the length of the residual vector (m in Eq. 21) equals $2 \cdot n_{\text{modes}} = 6$. Meanwhile, the number of optimization variables $n_{\mathbf{x}}$ in Eq. 21 equals $n_{\boldsymbol{\alpha}} = 15$. As a result, $m < n_{\mathbf{x}}$ and the problem is underdetermined. Now consider Case 2(b), applying the trust-region-reflective algorithm on the eigenvector difference formulation, m equals $n_{\text{m}} \cdot n_{\text{modes}} = 21 \cdot 3 = 63$, as presented in Eq. 24. The number of optimization variables $n_{\mathbf{x}}$ still equals $n_{\boldsymbol{\alpha}} = 15$. As a result, $m > n_{\mathbf{x}}$, making the problem not underdetermined; the trust-region-reflective algorithm can be applied to Case 2(b).

When using MATLAB `lsqnonlin` with the trust-region-reflective algorithm, the upper and lower bounds of $\boldsymbol{\alpha}$ are simply set to be 1 and -1, respectively. This means that the actual stiffness parameters are assumed to be within ±100% of the nominal values. (For applications where different bound values may be deemed appropriate, these can be easily set in the computer code.) In addition, for each case shown in Table 1, the optimization process is initiated from 100 random starting points, which are uniformly randomly generated between the upper and lower bounds of $\boldsymbol{\alpha}$. On the other hand, when using the Levenberg-Marquardt algorithm with MATLAB `lsqnonlin`, upper and lower bounds cannot be handled by the toolbox, as described in Section 3. Consequently, optimal result sets obtained from Levenberg-Marquardt algorithm that are out of the bounds are discarded and not included in the final result sets. Instead, the starting point

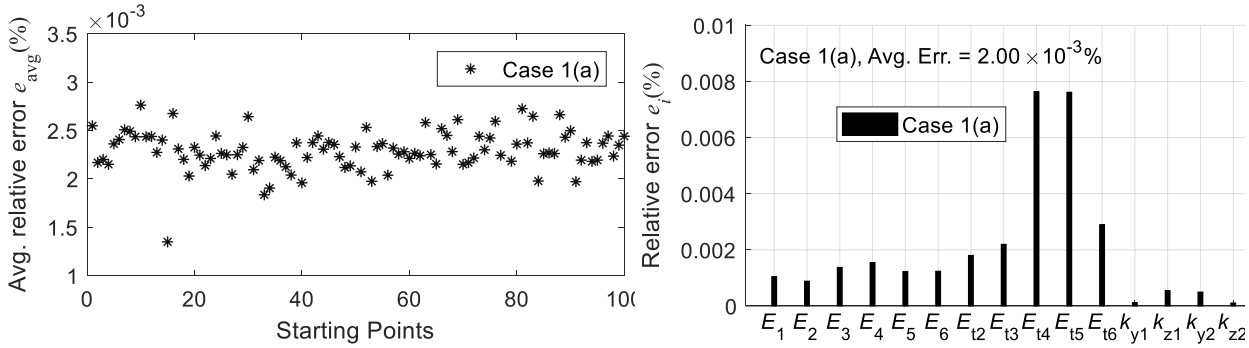
is replaced with the next randomly generated point that can conclude the search within the desired bounds of [-1, 1]. As a result, for Case 1(a) and 2(a) that use Levenberg-Marquardt algorithm, in order to obtain 100 sets of optimal results that are within the bounds, the Levenberg-Marquardt algorithm may have to solve the optimization problems from more than 100 starting points.

4.1.1 Updating results of MAC value formulation

Figure 2 plots the model updating results for MAC value formulation when using analytical gradient shown in Eq. 9 (instead of the default numerical gradient calculated by MATLAB). As shown in Table 1, Case 1(a) represents solving the MAC value formulation with Levenberg-Marquardt algorithm. Using α_i^* to represent the optimal solution of each search, the relative error of every stiffness parameter can be calculated as the relative difference of the updated stiffness parameter value from the actual stiffness parameter value. For example, if $E_1^* = E_1^{\text{nom}} \cdot (1 + \alpha_i^*)$ is the optimal stiffness value of E_1, the corresponding relative error is calculated based on the actual/correct stiffness value, $E_1^{\text{act}} = E_1^{\text{nom}} \cdot (1 + \alpha_i^{\text{act}})$. As a result, the relative error $e_1 = \frac{|E_1^* - E_1^{\text{act}}|}{E_1^{\text{act}}} \times 100\% = \frac{|\alpha_1^* - \alpha_1^{\text{act}}|}{1+\alpha_1^{\text{act}}} \times 100\%$. In general, the relative error of the *i*-th stiffness parameter is calculated as:

$$e_i = \frac{|\alpha_i^* - \alpha_i^{\text{act}}|}{1 + \alpha_i^{\text{act}}} \times 100\%, \qquad i = 1 \cdots n_{\boldsymbol{\alpha}} \tag{25}$$

where α_i^{act} is the actual value of updating variable α_i, i.e. the value listed in the last column of Table 2. Figure 2(a) shows the average relative error $e_{\text{avg}} = \frac{1}{n_{\boldsymbol{\alpha}}} \sum_{i=1}^{n_{\boldsymbol{\alpha}}} e_i$ among all $n_{\boldsymbol{\alpha}} = 15$ updated stiffness parameters, for each of the 100 sets of optimal results that are within the bounds. Horizontal axis is the sequence number of 100 randomized starting points. Recall that Case 1(a) uses the Levenberg-Marquardt algorithm, for which the implementation in MATLAB does not allow setting the bounds of the optimization variables. Whereas for this problem, 27 sets of the optimal results from Case 1(a) are out of the [-1,1] bounds, which indicates the nonconvexity of the MAC value formulation. The figure shows that, after discarding 32 sets of invalid results, the remaining optimal result sets are close to the correct stiffness values, i.e. $e_{\text{avg}} < 3.1\times10^{-3}\%$. Among the 100 sets of optimal results for each case, the best solution is chosen as the solution set with the minimum objective function value. For the best solution, Figure 2(b) plots the relative error of each stiffness parameter (e_i). The figure demonstrates that the best solution is almost identical to the actual stiffness value by observing that the largest and average error equal to $7.6\times10^{-3}\%$ and $2.0\times10^{-3}\%$, respectively. On a PC with an Intel i7-7700 CPU and 16 GB RAM, it takes 7 hours and 14 minutes to obtain the 100 optimal result sets that are within the bounds.

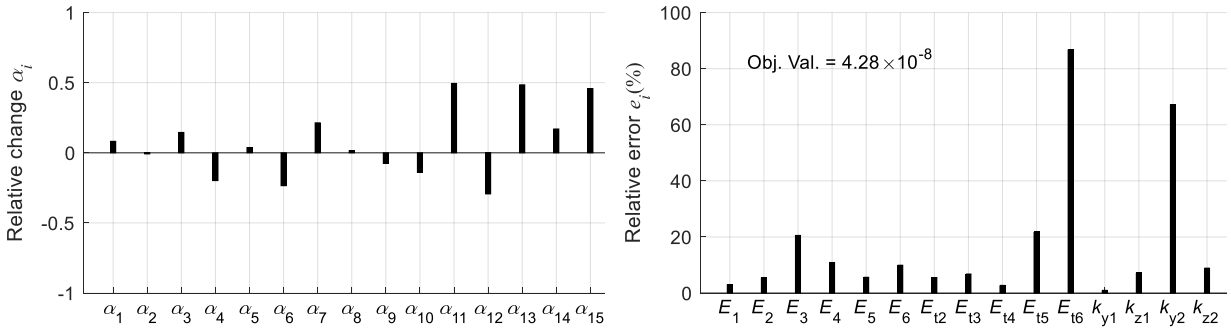


(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

Figure 2. Updating results of MAC value formulation using analytical gradient

4.1.2 Nonconvexity of MAC value formulation

Further study on nonconvexity is conducted in this section, using a local optimum found within the [-1, 1] bounds of updating variable $\boldsymbol{\alpha}^*$. The updating variable values of this local optimum, α_i, are displayed in Figure 3(a), while the corresponding relative stiffness errors, e_i, are shown in Figure 3(b). The maximum error is 86.84% and the objective function equals 4.28×10^{-8}.



(a) Optimal value of updating variable $\boldsymbol{\alpha}^*$ (b) Relative error of stiffness parameters

Figure 3. Updating results of the optimal set

The gradient, $\nabla f(\boldsymbol{\alpha}^*) \in \mathbb{R}^{15}$ and Hessian, $\nabla^2 f(\boldsymbol{\alpha}^*) \in \mathbb{R}^{15\times15}$ at this local optimum are calculated. The $\mathcal{L}_2$-norm of the gradient, i.e. $\|\nabla f(\boldsymbol{\alpha}^*)\|_2 = 2.10 \times 10^{-6}$, is close to zero; the Hessian matrix is positive definite, i.e. $\nabla^2 f(\boldsymbol{\alpha}^*) \succ 0$, by verifying the sign of all eigenvalues. These confirm that the results shown in Figure 3(a) represent a feasible local minimum of MAC value formulation.

While it is impossible to visualize 15-dimensional space $f(\boldsymbol{\alpha})$, Figure 4 displays a walk from the local optimal value of $\boldsymbol{\alpha}^*$, to the actual value of $\boldsymbol{\alpha}$, i.e. $\boldsymbol{\alpha}^{\mathrm{act}}$ shown in Table 2, along a hyperline in $\mathbb{R}^{15}$ space. The hyperline is defined with a scalar θ; along the hyperline, the values of updating variables are $\boldsymbol{\alpha}(\theta) = \boldsymbol{\alpha}^* + \theta \cdot (\boldsymbol{\alpha}^{\mathrm{act}} - \boldsymbol{\alpha}^*)$. When $\theta = 0$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$; when $\theta = 1$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{\mathbf{act}}$. Accordingly, the *y*-axis represents the MAC value objective function (Eq. 3 or 8) evaluated at different value of $\boldsymbol{\alpha}$ along the hyperline:

$$f\big(\boldsymbol{\alpha}(\theta)\big) = f\big(\boldsymbol{\alpha}^* + \theta \cdot (\boldsymbol{\alpha}^{\mathrm{act}} - \boldsymbol{\alpha}^*)\big)$$

Had $f(\boldsymbol{\alpha})$ been convex on $\boldsymbol{\alpha}$, $f\big(\boldsymbol{\alpha}(\theta)\big)$ should also be convex on θ (see page 68 in the 7th version of reference [18]). The Figure 4 clearly shows that there are two valleys along the hyperline, locating at $\theta = 0$ and 1, respectively. Therefore, the optimization problem of MAC value formulation is confirmed to be noncovnex.

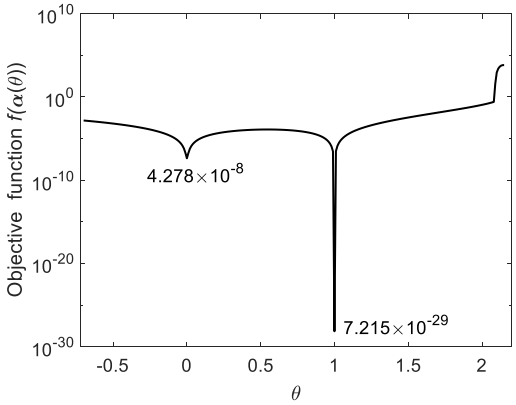
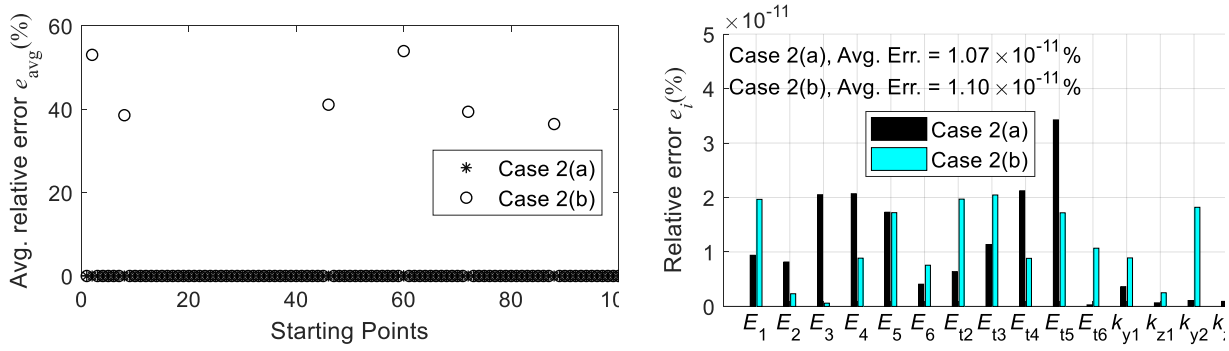


Figure 4. Hyperline walk from local minimum to global minimum of updating variable

4.1.3 Updating results of eigenvector difference formulation

Recall that Case 2(a) and 2(b) (Table 1) stand for applying Levenberg-Marquardt and trust-region-reflective algorithms, respectively, to the eigenvector difference formulation. For each optimization case, Figure 5(a) shows the average relative error (e_{avg}) of the optimal result sets from 100 randomized starting points. Analytical gradient shown in Eq. 20 is used during the optimization process. For Case 2(a) with the Levenberg-Marquardt algorithm, 61 sets of optimal results are out of the bounds and hence discarded. Figure 5(a) demonstrates that the final 100 in-bound result sets from Case 2(a) end up at the correct values of the updating variables ($e_{avg} < 2.80\times10^{-11}$%). For Case 2(b) with the trust-region-reflective algorithm, although all the optimal result sets are guaranteed to be within the [-1,1] bound, six optimization searches fail to converge at the correct value. The average error of these seven result sets are from 36% to 54%. After inspecting the seven incorrect result sets, it is found that each result set has at least one updating variable α_i hitting bound (which also implies the nonconvexity of the eigenvector difference formulation). For each optimization case, a best solution among the 100 optimal result sets is selected by the smallest objective function value, and Figure 5(b) plots the relative errors of each stiffness parameter for the two best solutions. Both best solution sets provide correct stiffness values, with the average error equal to 1.07×10^{-11} % for Case 2(a) and 1.10×10^{-11} % for 2(b). Using the PC with an Intel i7-7700 CPU and 16 GB RAM, it takes only 32 minutes for Case 2(a) to obtain the 100 optimal solutions within the bounds and 14 hours and 8 minutes for Case 2(b). For this problem, using Levenberg-Marquardt algorithm takes much shorter computational time, while providing accurate results.



(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

Figure 5. Updating results of eigenvector difference formulation using analytical gradient

To summarize the model updating of the steel pedestrian bridge, we first discuss the application of Levenberg-Marquardt algorithm on both the MAC value and the eigenvector difference formulations, i.e. Cases 1(a) and 2(a). Although for each formulation some out-of-bound optimal result sets are rejected, all the in-bound results can converge around the correct values of updating parameters. Comparing the two formulations, the accuracy of optimal result sets from the eigenvector difference formulation is higher than the MAC value formulation, and meanwhile the computational time of the eigenvector difference formulation to obtain the 100 in-bound optimal result sets is shorter.

Recall that for this example the trust-region-reflective algorithm cannot be applied to the MAC value formulation, i.e. Case 2(a) is not applicable. When using the trust-region-reflective algorithm on the eigenvector difference formulation – Case 2(b), some of the optimal search results converge to the bounds, which implies the nonconvexity of the objective function. Finally, we compare Cases 2(a) and 2(b), i.e. applying the two optimization algorithms on the eigenvector difference formulation. The Levenberg-Marquardt algorithm appears more efficient for this example. However, it should be noted that the MATLAB implementation of Levenberg-Marquardt algorithm cannot handle variable bounds. This can present challenges when applying to field testing data from as-built structures, where most of the

optimization result sets may end up outside the required bounds. One the other hand, the trust-region-reflective algorithm allows setting bounds and ensures the optimization results are within bounds.

4.2 Concrete Building Frame

The second structure studied in this research is a concrete building frame (Figure 6), which simulates a full-scale test structure in the Structural Engineering and Materials Laboratory on Georgia Tech campus. The test frame structure consists of two bays and two stories, and it was meant to be representative of low-rise reinforced concrete office buildings in the central and eastern United States built in the 1950s-1970s [27]. The columns and beams are modeled with frame elements. Corresponding to dense sensor instrumentation, seven segments are allocated per column on each story, and twelve segments per beam in each bay. In SAP2000 to ensure stiffness contribution from both concrete and rebars, along every column or beam segment, one frame element is assigned for the concrete material and another frame element is assigned for the steel reinforcement. Each floor slab is meshed into 175 shell elements. In total, the FE model of the concrete building frame has 2,302 DOFs. Similar to the FE model of the steel pedestrian bridge, the mass matrix (**M**) is a diagonal matrix whose diagonal entries associated with rotational DOFs equal zero. Figure 6 also shows the accelerometer instrumentation for this simulation study, and the corresponding measurement directions. A total of 43 DOFs are measured, i.e. the length of $\boldsymbol{\psi}_i^{\mathrm{EXP,m}}$ and $\boldsymbol{\psi}_i^{\mathrm{m}}(\boldsymbol{\alpha})$ is $n_{\mathrm{m}} = 43$.

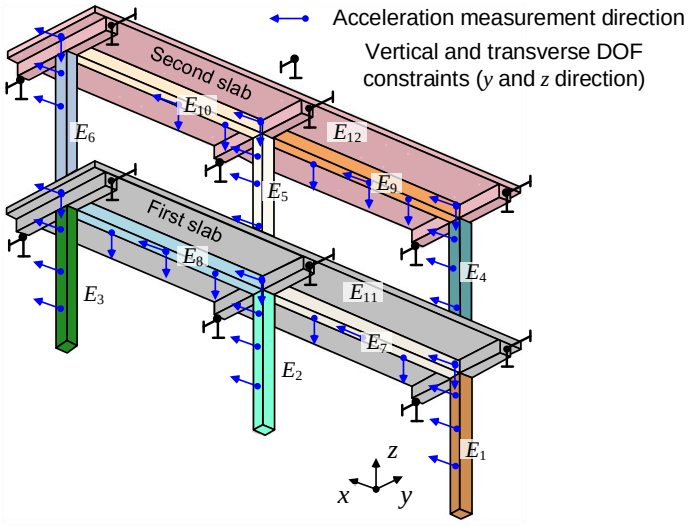


Figure 6. Model of a 2-story 2-bay concrete building frame [27] (height in *z*: 2 × 12 ft.; length in *x*: 2 ×18 ft.; width in *y*: 9 ft.) and sensor instrumentation

As shown in Figure 6, the accelerometers measure longitudinal and vertical vibration, i.e. *x* and *z* directions. Thus only in-plane vibration mode shapes, i.e. in *x*-*z* plane, can be extracted from measurement data. To avoid the side effect of out-of-plane mode shapes (in *y*-*z* plane) on FE model updating, the vertical and transverse DOFs (*y* and *z* direction) at both ends of the three transverse beams (along *y* direction) on each slab are constrained. Lastly, at the bottom of three columns on the first slab, all six DOFs are constrained to represent an ideal fixed support.

Table 3 lists the stiffness parameters to be updated. As shown in Figure 6, in the first story, $E_1 \sim E_3$ represent the concrete elastic moduli of members in the three columns; E_7 and E_8 represent the concrete elastic moduli of longitudinal beam members (along *x* direction); E_{11} represents the concrete elastic moduli of the first slab and the associated transverse beam members (along *y* direction). Similarly, other moduli for the second story can be found in the figure. While this study only involves simulation, the selection of moduli corresponds to different concrete pours during the construction, and thus is in preparation for future model

updating of the as-built structure with experimental data. Compared to concrete, the elastic modulus of steel reinforcement is considered to be accurate enough, and thus not being updated in this study.

For all the concrete moduli being updated, Table 3 lists the nominal and actual values. In total, there are 12 updating variables for this model updating, i.e. $n_{\boldsymbol{\alpha}} = 12$. The column α_i^{act} in Table 3 lists the actual values of $\boldsymbol{\alpha}$, i.e. the ideal solutions to be identified through FE model updating.

Table 3 Structural properties of the concrete building frame

Stiffness parameters		Nominal value	Actual value	Updating variables	α_i^{act}
Elastic moduli of concrete members (kips/in^2)	E_1	3,900	3,510	α_1	-0.10
	E_2	3,700	4,440	α_2	0.20
	E_3	3,700	4,440	α_3	0.20
	E_4	3,200	3,040	α_4	-0.05
	E_5	3,200	3,840	α_5	0.20
	E_6	3,200	3,680	α_6	0.15
	E_7	3,200	3,680	α_7	0.15
	E_8	3,200	3,520	α_8	0.10
	E_9	3,400	3,060	α_9	-0.10
	E_{10}	3,400	2,890	α_{10}	-0.15
	E_{11}	3,200	3,840	α_{11}	0.20
	E_{12}	3,400	3,910	α_{12}	0.15

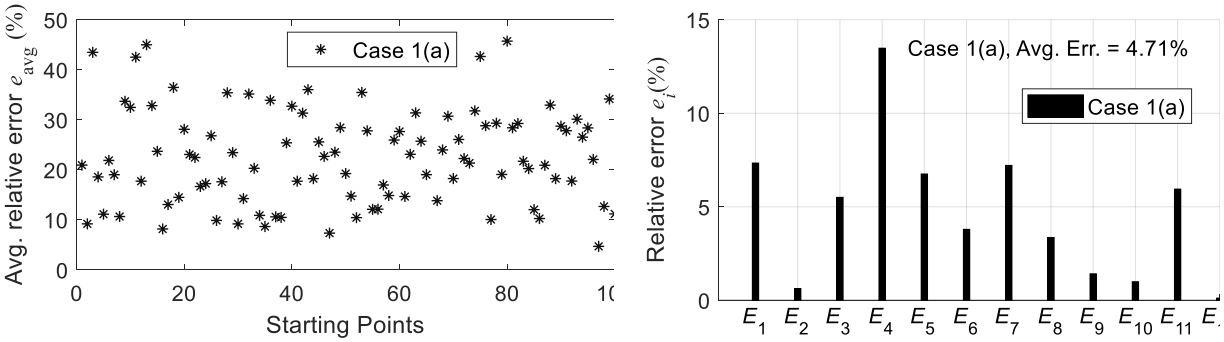
For updating the FE model, it is assumed that the first three vibration modes ($n_{\text{modes}} = 3$) are available. As shown in Eq. 23, when using MAC value formulation (Eq. 3) to perform FE model updating, the length of the residual vector (m in Eq. 21) equals $2 \cdot n_{\text{modes}} = 6$. Meanwhile, $n_{\mathbf{x}}$ in Eq. 21 equals $n_{\boldsymbol{\alpha}} = 12$. As a result, $m < n_{\mathbf{x}}$ and the problem is underdetermined; the trust-region-reflective is not applicable for the MAC value formulation (Case 1(b) in Table 1). For the eigenvector difference formulation in Eq. 5, as presented in Eq. 24, the residual vector length m equals $n_{\text{m}} \cdot n_{\text{modes}} = 43 \cdot 3 = 129$. As a result, with the number of optimization variables $n_{\mathbf{x}}$ still equal to $n_{\boldsymbol{\alpha}} = 12$, the problem is not underdetermined; the trust-region-reflective algorithm can be applied to the eigenvector difference formulation (Case 2(b) in Table 1). Finally, as described in Section 3, the Levenberg-Marquardt algorithm can be applied to both optimization problems in Eqs. 3 and 5, i.e. both Case 1(a) and 2(a) in Table 1 are applicable.

When using MATLAB `lsqnonlin` with trust-region-reflective, the upper and lower bounds of $\boldsymbol{\alpha}$ are set to be 1 and -1, respectively. Similar to the steel pedestrian bridge example, for each applicable case shown in Table 1, the optimization process is initiated from 100 random starting points within the bounds of $\boldsymbol{\alpha}$. Finally, when using MATLAB `lsqnonlin` with Levenberg-Marquardt, the optimal result sets that are out of the assigned bounds are rejected, and the corresponding starting point will be replaced with the next randomly generated point that can achieve valid optimal results.

4.2.1 Updating results of MAC value formulation

Case 1(a) is first studied, applying the Levenberg-Marquardt algorithm on the MAC value formulation. Instead of using the analytical gradient calculated by Eq. 9, the optimization search is first performed using gradient calculated numerically by MATLAB through finite difference method. For each of the 100 successful runs, Figure 7(a) displays the average relative error (e_{avg}) for all $n_{\boldsymbol{\alpha}} = 12$ stiffness parameters after discarding ten optimal result sets that are out of the [-1,1] bounds during the optimization process. With average error $e_{\text{avg}} > 4.71\%$, the figure shows that none of the optimization processes converge close to the correct value of updating variables. For the solution set that achieves the minimum objective function value among the 100 starting points (point #97), Figure 7(b) plots the relative error (e_i) of each the stiffness parameter. The figure shows the obtained stiffness parameter values are not reasonable, with the maximum

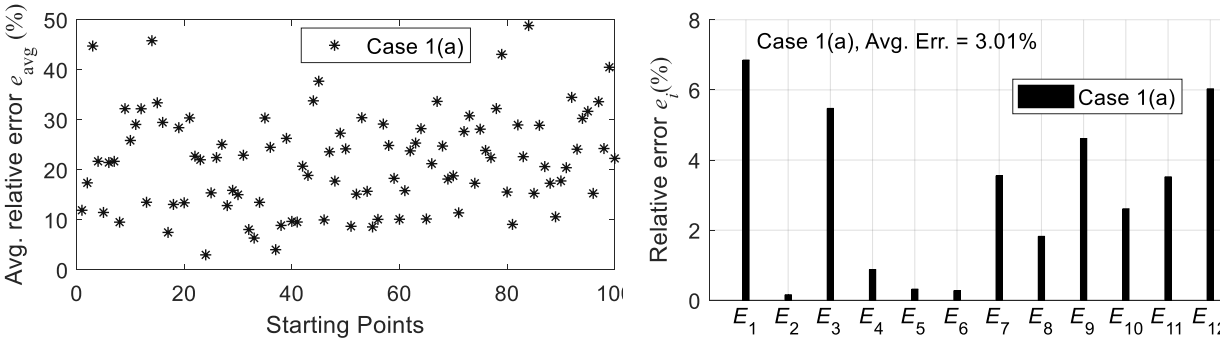
relative error larger than 13%. On a PC with an Intel i7-7700 CPU and 16 GB RAM, it takes 18 hours and 58 minutes to obtain the 100 optimal result sets that are within the bounds.



(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

Figure 7. Updating results of MAC value formulation using numerical gradient

The Levenberg-Marquardt optimization for the MAC value formulation is then repeated with analytical gradient calculated by Eq. 9 instead. During the optimization process, five result sets from the Levenberg-Marquardt algorithm are out of the [-1, 1] bounds, and thus discarded. Figure 8(a) plots the e_{avg} for each of the 100 in-bound optimal result sets. Since all e_{avg} is greater than 3.01%, it can be concluded that again none of the optimal result sets converge close to the correct values. For the result set from starting point #24 with the smallest objective function value, Figure 8(b) shows the relative errors of the stiffness parameters (e_i). Although the results of using analytical gradient is better than using numerical gradient, the results are still not reasonable given that the maximum error is larger than 6.5%. On a PC with an Intel i7-7700 CPU and 16 GB RAM, it takes 22 hours and 58 minutes to obtain the 100 optimal result sets that are within the bounds.



(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

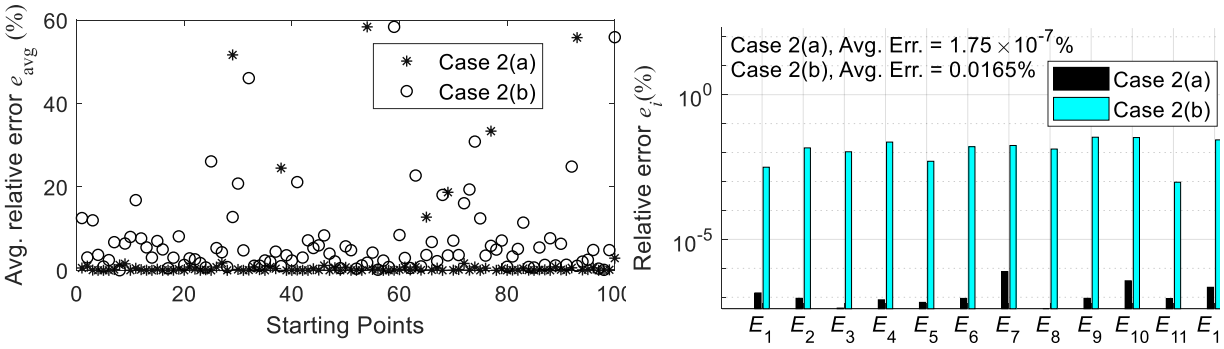
Figure 8. Updating results of MAC value formulation using analytical gradient

In conclusion, for this concrete frame structure, the MAC value formulation cannot provide reasonable FE model updating results either using numerical gradient or analytical gradient during the optimization process.

4.2.2 Updating results of eigenvector difference formulation

Recall that Case 2(a) and 2(b) refers to applying Levenberg-Marquardt and trust-region-reflective algorithms, respectively, on the eigenvector difference formulation. For each of the two cases, Figure 9(a) plots the average relative error of $n_{\alpha} = 12$ stiffness parameters for 100 in-bound optimal result sets. The figure is obtained when numerical gradient is used during the optimization process. For Case 2(a), after discarding nine optimal result sets that are out of the [-1,1] bounds, 86 out of the 100 inbound result sets

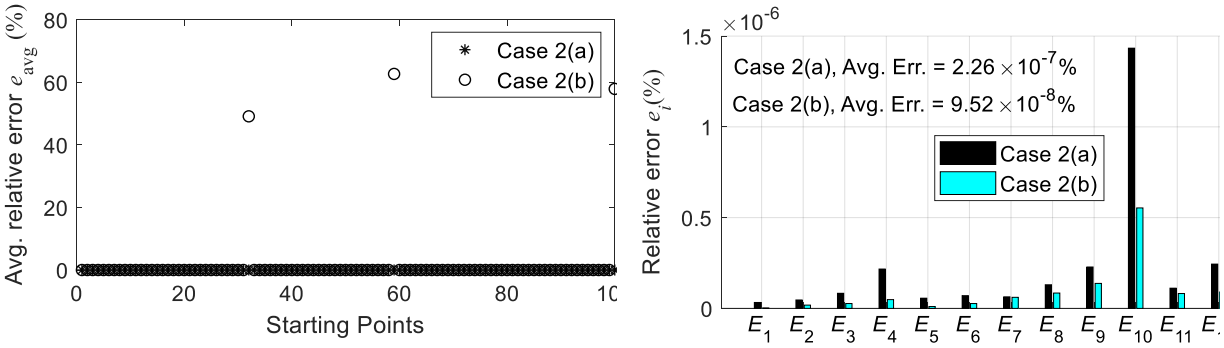
can update the stiffness parameter with an acceptable accuracy, i.e. $e_{avg} < 1\%$. On the other hand, for Case 2(b), only 16 optimal result sets converge around the actual updating parameter values with $e_{avg} < 1\%$. For each case, the best solution is again selected as the one with the minimum objective function value among 100 result sets. The relative errors of optimal stiffness parameter values are plotted in Figure 9(b). It can be seen that the accuracy of optimal result sets from Case 2(a) is much higher than Case 2(b), with e_{avg} equal to $1.75\times10^{-7}\%$ and 0.0165% for the two cases, respectively. Using the same PC with an Intel i7-7700 CPU and 16 GB RAM, it takes 23 hours and 16 minutes for Case 2(a) to obtain the 100 optimal solutions within the bounds. On the other hand, Case 2(b) took 22 hours 38 minutes.



(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

Figure 9. Updating results of eigenvector difference formulation using numerical gradient

For comparison, both optimization algorithms are repeated on the eigenvector difference formulation, but using analytical gradient in both optimization Case 2(a) and 2(b). Figure 10(a) plots the average relative error of all stiffness parameters for 100 optimal result sets within the [-1, 1] bounds. After discarding 14 optimal result sets that are out of the [-1, 1] bounds, all the optimal result sets from Case 2(a) converge close to the actual value with an average error smaller than $2.16\times10^{-6}\%$. As for Case 2(b), the figure shows that three out of 100 optimal result sets end up with large relative error, ranging from 49.14% to 62.70%. Upon inspection of the four result sets with large errors, it is found that similar to the steel pedestrian bridge, each result set has at least one updating variable α_i hitting bound. For each optimization case, the best result with the minimum objective function value among 100 result sets is identified; the relative stiffness errors of each best result set are shown in Figure 10(b). The best solution of the two optimization cases have similar accuracy. Using the same PC with an Intel i7-7700 CPU and 16 GB RAM, it takes 1 hour and 13 minutes for Case 2(a) to obtain the 100 optimal solutions within the bounds and 52 minutes for Case 2(b).



(a) Average relative error of stiffness parameters (b) Relative error of the stiffness parameters

Figure 10. Updating results of eigenvector difference formulation using analytical gradient

In summary, for model updating of the concrete building frame, again only the Levenberg-Marquardt algorithm can be applied to the MAC value formulation (Case 1(a)). However, the updating cannot provide a set of optimal results with acceptable accuracy, either using numerical or analytical gradient during the optimization process. As for the eigenvector difference formulation, when using numerical gradient, both Levenberg-Marquardt (Case 2(a)) and trust-region-reflective (Case 2(b)) algorithms can find the correct updating parameter values, but the relative error of best optimal result set from the Levenberg-Marquardt algorithm is much smaller than the trust-region-reflective algorithm. On the other hand, when using analytical gradient, all the in-bound optimal result sets from Levenberg-Marquardt algorithm converge around the correct values. While a few optimal result sets from the trust-region-reflective algorithm converge to the assigned bounds, all other results converge to the correct values. Overall, it can be concluded that when applying both optimization algorithms on the eigenvector difference formulation, using analytical gradient not only provides more accurate model updating results in general, but also can find the correct updating parameter values more efficiently. The study demonstrates the advantage of using analytical gradient versus numerical gradient.

5 Conclusions and Future Work

This paper presents two finite element (FE) model updating formulations using frequency-domain properties. The objective function of both model updating formulations aims to minimize the difference in resonance frequencies and mode shapes obtained from the experimental and the simulated FE model. One formulation uses MAC values and the other uses direct differences between eigenvectors. To facilitate the optimization process, the analytical gradient of both modal property difference formulations is provided, including a new approach of obtaining the eigenvector gradient with respect to the updating variables.

In order to solve the optimization problem of both model updating formulations, two local optimization algorithms implemented in MATLAB, i.e. the Levenberg-Marquardt and the trust-region-reflective algorithms are adopted. Considering that the objective functions of both model updating formulations are nonconvex, in general the global optimality of the optimization results cannot be guaranteed. Therefore, in order to increase the chance of finding the global minimum, randomly generated starting points are adopted to start the optimization search.

The simulation results of steel pedestrian bridge and concrete building frame examples can lead to the following conclusions:

i. The MAC value formulation can correctly identify stiffness parameter values for a relatively simpler structural model (the steel pedestrian bridge), but may fail to provide reasonable results when the complexity of structure increases (as in the concrete building frame);
ii. The eigenvector difference formulation is able to correctly update the structural stiffness values for both simpler and more complex structures. The formulation performs better than the MAC value formulation throughout the numerical examples studied in this research;
iii. The Levenberg-Marquardt algorithm implemented in MATLAB performs more efficiently and can be applied when the optimization problem is underdetermined. However, the implementation does not allow setting bounds of updating variables. On the other hand, the trust-region-reflective algorithm cannot be applied to underdetermined problems, but can ensure the optimization search results are within bounds.
iv. Using analytical gradient during the optimization process in general not only provide more accurate model updating results, but also save computing time.

In future studies, field experimental data will be used to perform the FE model updating of the as-built structures introduced in Section 2. In addition, for solving the optimization problems in model updating, other optimization algorithms will be studied to improve the chance of finding the globally optimal results.

6 Acknowledgment

This research is partially sponsored by the National Science Foundation (#CMMI-1150700 and #CMMI-1634483). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the sponsors.

7 References

[1] J. P. Lynch and K. J. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *The Shock and Vibration Digest,* vol. 38, pp. 91-128, 2006.

[2] M. Kane, D. Zhu, M. Hirose, X. Dong, B. Winter, M. Häckell, J. P. Lynch, Y. Wang, and A. Swartz, "Development of an extensible dual-core wireless sensing node for cyber-physical systems," in *Proceedings of SPIE, Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security*, p. 90611U, San Diego, California, USA, 2014.

[3] R. A. Swartz, D. Jung, J. P. Lynch, Y. Wang, D. Shi, and M. P. Flynn, "Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system," in *Proceedings of the 5th International Workshop on Structural Health Monitoring*, Stanford, CA, 2005.

[4] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*, Dordrecht; Boston: Kluwer Academic Publishers, 1995.

[5] J. N. Yang, S. L. Lin, H. W. Huangl, and L. Zhou, "An adaptive extended Kalman filter for structural damage identification," *Structural Control & Health Monitoring,* vol. 13, pp. 849-867, 2006.

[6] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Annals of the Institute of Statistical Mathematics,* vol. 55, pp. 409-422, 2003.

[7] E. N. Chatzi and A. W. Smyth, "The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non‐collocated heterogeneous sensing," *Structural Control and Health Monitoring,* vol. 16, pp. 99-123, 2009.

[8] T. Sato and K. Qi, "Adaptive H∞ filter: its application to structural identification," *Journal of Engineering Mechanics,* vol. 124, pp. 1233-1240, 1998.

[9] M. Hoshiya and E. Saito, "Structural identification by extended Kalman filter," *Journal of Engineering Mechanics-ASCE,* vol. 110, pp. 1757-1770, 1984.

[10] Q. W. Zhang, C. C. Chang, and T. Y. P. Chang, "Finite element model updating for structures with parametric constraints," *Earthquake Engineering & Structural Dynamics,* vol. 29, pp. 927-944, 2000.

[11] O. S. Salawu, "Detection of structural damage through changes in frequency: A review," *Engineering Structures,* vol. 19, pp. 718-723, 1997.

[12] P. W. Moller and O. Friberg, "Updating large finite element models in structural dynamics," *AIAA Journal,* vol. 36, pp. 1861-1868, 1998.

[13] A. Teughels, J. Maeck, and G. De Roeck, "Damage assessment by FE model updating using damage functions," *Computers & Structures,* vol. 80, pp. 1869-1879, 2002.

[14] K. V. Yuen, "Updating large models for mechanical systems using incomplete modal measurement," *Mechanical Systems and Signal Processing,* vol. 28, pp. 297-308, 2012.

[15] MathWorks Inc., *Optimization Toolbox™ User's Guide*, R2015b ed., Natick, MA, 2015.

[16] J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*. vol. 630, G. A. Watson, Ed.: Springer Berlin Heidelberg, 1978, pp. 105-116.

[17] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization,* vol. 6, pp. 418-445, 1996.

[18] S. P. Boyd and L. Vandenberghe, *Convex Optimization*: Cambridge University Press, 2004.

[19] A. Barvinok, *A Course in Convexity* vol. 54, Providence, RI: American Mathematical Society, 2002.

[20] M. Berger, "Convexity," *American Mathematical Monthly,* vol. 97, pp. 650-78, 1990.

[21] R. J. Allemang and D. L. Brown, "A correlation coefficient for modal vector analysis," in *Proceedings of the 1st international modal analysis conference*, pp. 110-116, 1982.

[22] R. Fox and M. Kapoor, "Rates of change of eigenvalues and eigenvectors," *AIAA Journal,* vol. 6, pp. 2426-2429, 1968.

[23] R. B. Nelson, "Simplified calculation of eigenvector derivatives," *AIAA Journal,* vol. 14, pp. 1201-1205, 1976.

[24] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, "Approximate solution of the trust region problem by minimization over two-dimensional subspaces," *Mathematical Programming,* vol. 40, pp. 247-263, 1988.

[25] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM Journal on Scientific Computing,* vol. 21, pp. 1-23, 1999.

[26] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*: SIAM, 2007.

[27] X. Dong, X. Liu, T. Wright, Y. Wang, and R. DesRoches, "Validation of wireless sensing technology densely instrumented on a full-scale concrete frame structure," in *Proceedings of International Conference on Smart Infrastructure and Construction (ICSIC)*, Cambridge. United Kingdom, 2016.