POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA CIVILE,
AMBIENTALE E TERRITORIALE

Finite Element Analysis of Vehicle-Bridge Interaction in Abaqus

Tesi di Laurea Magistrale in
Civil Engineering - Ingegneria Civile

Author: **Andres Felipe Ordonez Ruiz**

Student ID: 248433
Advisor: Prof. Luca Martinelli
Co-advisors: Eng. Tommaso Panigati
Academic Year: 2025-26

Abstract

Structural Health Monitoring (SHM) techniques are increasingly utilized to facilitate the assessment and management of existing bridge infrastructure. Among vibration-based methods, Operational Modal Analysis (OMA) serves as an effective tool for identifying the dynamic characteristics of structures under service conditions. This thesis develops a vibration-based model-updating framework for an in-service bridge, employing experimentally identified modal information.

Structural response data are analyzed using OMA techniques implemented within the pyOMA framework to extract the dynamic properties of the bridge. Given the characteristics of the available data and operational monitoring conditions, the analysis primarily targets the identification of natural frequencies.

A finite element model of the bridge is constructed in Abaqus, using shell elements to model the deck and the primary structural components. The modeling strategy intends to balance computational efficiency with physical realism and addresses practical issues related to element connectivity and parameter sensitivity.

Model calibration is performed using the Douglas–Reid method, considering only natural frequencies and omitting mode shape information. This approach is justified by the limited sensitivity of the identified mode shapes to the selected updating parameters, as reported in the literature and corroborated by the model development and updating process in this study.

The topic of vehicle–bridge interaction analysis is discussed through a review of relevant literature that outlines established modeling procedures and potential extensions to the proposed framework, and a modeling of a sprung-mass vehicle is performed to analyze the VBI in the bridge. The results show that frequency-based model updating is a practical and reliable strategy for calibrating numerical models of in-service bridges.

Keywords: Structural Health Monitoring, Operational Modal Analysis, Model Updating, Finite Element Modeling, Bridge Structures, Moving Load Analysis

Abstract in lingua italiana

Le tecniche di Structural Health Monitoring (SHM) sono sempre più utilizzate per supportare la valutazione e la gestione delle infrastrutture di ponte esistenti. Tra i metodi basati sulle vibrazioni, la Operational Modal Analysis (OMA) si conferma uno strumento efficace per l'identificazione delle caratteristiche dinamiche delle strutture in condizioni di esercizio. In questo contesto, viene sviluppato un framework di model updating basato su misure vibrazionali per un ponte in esercizio, utilizzando informazioni modali identificate sperimentalmente.

I dati di risposta strutturale sono analizzati mediante tecniche OMA implementate nell'ambiente pyOMA, con l'obiettivo di estrarre le proprietà dinamiche del ponte. Considerando le caratteristiche del dataset disponibile e le condizioni di monitoraggio operativo, l'analisi si concentra principalmente sull'identificazione delle frequenze naturali.

Un modello agli elementi finiti del ponte è sviluppato in Abaqus, utilizzando elementi shell per la modellazione dell'impalcato e dei principali componenti strutturali. La strategia di modellazione mira a garantire un equilibrio tra efficienza computazionale e rappresentazione fisicamente realistica, affrontando aspetti pratici relativi alla connettività tra elementi e alla sensibilità dei parametri.

La calibrazione del modello viene eseguita mediante il metodo di Douglas–Reid, considerando esclusivamente le frequenze naturali ed escludendo le informazioni sulle forme modali. Questa scelta è motivata dalla limitata sensibilità delle forme modali identificate rispetto ai parametri selezionati per l'aggiornamento, come riportato in letteratura e ulteriormente confermato nel processo di sviluppo e aggiornamento del modello condotto in questo studio.

Il tema dell'analisi dell'interazione veicolo–ponte è affrontato attraverso una revisione della letteratura pertinente, che illustra le procedure di modellazione consolidate e le possibili estensioni del quadro metodologico proposto; inoltre, viene sviluppato un modello di veicolo a massa sospesa per analizzare la VBI sul ponte. I risultati mostrano che l'aggiornamento del modello basato sulle frequenze è una strategia pratica e affidabile per calibrare i modelli numerici di ponti in esercizio.

Parole chiave: Monitoraggio della Salute Strutturale, Analisi Modale Operativa, Aggiornamento di Modelo, Modellazione agli elementi finiti, Strutture da ponte, Analisi dei carichi mobili

Contents

1 | Introduction

The growing need for safe, reliable, and cost-effective management of existing bridge infrastructure has increased interest in Structural Health Monitoring (SHM) techniques. Vibration-based approaches are especially useful because they allow for the assessment of structural performance using measured dynamic responses under operational conditions. Within this context, Operational Modal Analysis (OMA) has become a key methodology for extracting modal parameters, such as natural frequencies and mode shapes, without the need for controlled excitation. This makes OMA especially suitable for in-service bridges, as discussed in Operational Modal Analysis of Civil Engineering Structures [20].

Although SHM technologies are increasingly available, the validation and standardization of data-driven and model-based methods remain significant challenges. A principal limitation when advancing SHM practices is the lack of high-quality, long-term datasets from full-scale bridges operating under realistic environmental and traffic conditions. As highlighted in the SCSHM benchmark study on bridge in-service structural monitoring [13], many available methods are still verified using numerical simulations or laboratory-scale experiments, which do not capture the complexity and uncertainty of real structures. Therefore, benchmark datasets based on real bridges are essential for enabling objective comparison, validation, and advancement of SHM methodologies.

This thesis is conducted within the framework of the SCSHM benchmark study, which provides an extensive dataset from a densely instrumented in-service bridge subjected to operational traffic loads. The availability of this benchmark data offers a unique opportunity to investigate vibration-based model updating techniques using experimentally derived information while addressing the practical challenges of real-world monitoring data.

The initial phase of this research focuses on analyzing measured data using Operational Modal Analysis techniques implemented through the pyOMA framework [18], with the objective of extracting reliable estimates of the bridge's natural frequencies. These frequencies serve as reference quantities for calibrating a numerical finite element (FE) model of the structure.

A detailed finite element model of the bridge is developed in Abaqus, utilizing shell elements to represent the deck and main structural components. The modeling assumptions are defined to balance computational efficiency and physical realism, in accordance with standard guidelines for bridge structural modeling and analysis, as outlined in Chapter 4 of Bridge Design Practice [4]. Several modeling problems occur during this phase, including mesh definition, element connectivity, and sensitivity to material and geometric parameters. These aspects are discussed to point out their influence on the predicted dynamic response.

Model calibration is performed using the Douglas–Reid (DR) model updating method, focusing exclusively on natural frequencies and excluding mode shape information. This approach is justified by the limited sensitivity of the experimentally identified mode shapes to the selected updating parameters, as discussed in the literature [21] and observed during the model development and updating process in this thesis. As a result, the updating strategy relies solely on frequency information, which indicates greater responsiveness and reliability within the adopted modeling framework.

This thesis models a vehicle–bridge interaction (VBI) analysis in Abaqus where a simplified type of vehicle is taken, namely the so called sprung-mass vehicle model. The sprung-mass is integrated with the updated bridge finite element (FE) model using a node-to-surface contact formulation. The dynamic simulation is carried out with a representative crossing truck at a given speed and known characteristics. Moreover, the main theoretical background and relevant literature are reviewed to represent the modeling choices and to identify possible limitations regarding the initial modelind taken and to recognize feasible directions for future research.

In summary, this thesis seeks to advance vibration-based model updating of in-service bridges by means of integrating benchmark monitoring data with numerical modeling, whilst explicitly addressing the practical limitations, assumptions, and challenges experienced in applied structural engineering research.

1.1. Objectives of the Thesis

The main objective of this thesis is to investigate a vibration-based model updating framework for an in-service bridge by integrating experimentally identified dynamic characteristics with Abaqus finite element modeling.

To achieve the objective, the specific objectives of the thesis are:

- To analyze measured structural response data using Operational Modal Analysis

techniques with the aim of validating the dynamic properties of an in- service bridge under operational conditions, focusing in particular on the extraction of natural frequencies.

- To develop a finite element model of the bridge using shell elements in Abaqus, adopting modeling assumptions that balance computational efficiency as well as physical realism.
- To assess the sensitivity of numerical model parameters on the dynamic response, highlighting the influence of material properties, geometric characteristics, and modeling choices on the predicted natural frequencies.
- To perform model calibration using the Douglas–Reid model updating method, employing a strategy based on frequencies and evaluating its effectiveness in reducing differences between numerical predictions and identified experimental frequencies.
- To review the theoretical background and available methodologies for vehicle–bridge interaction analysis and to implement an Abaqus-based VBI model for the BM span, establishing a workflow that can be refined in future extensions of the present work.

2 | Literature Review for the SCSHM In-Service Monitoring Benchmark Bridge

This thesis applies a monitoring-driven workflow that progresses from in-service vibration measurements to a calibrated finite element (FE) model. Accordingly, the literature review is structured around the methods required for each stage of this process. Operational Modal Analysis (OMA) is first introduced to demonstrate how modal properties can be identified from output-only data collected under operating conditions, thereby providing the experimental frequencies for calibration. Subsequently, shell- and plate-based FE modeling of bridge decks and I-girder components is examined, as this scheme captures the bridge's structural behavior and supports the use of strain measurements for model validation. The review then addresses frequency-based model updating, with particular emphasis on the Douglas–Reid approach, which is adopted in this thesis to efficiently reduce variations between experimental and numerical natural frequencies while preserving physically meaningful parameter values. Finally, vehicle–bridge interaction concepts and Abaqus-oriented contact modeling are briefly considered to contextualize how the calibrated model may be extended to traffic-related dynamic analyses and moving-load applications.

2.1. Operational Modal Analysis (OMA) for bridge structures

Operational Modal Analysis (OMA) comprises modal identification procedures that estimate modal parameters—such as natural frequencies, damping ratios, and mode shapes—using only response measurements, without requiring measurement of excitation forces. In civil engineering, especially for bridges, this approach is advantageous because applying controlled input forces is often impractical due to the large size of the structures, their low natural frequencies, and operational constraints [18, 20].

OMA uses the continuous excitation of bridges by ambient and operational actions, including wind snd traffic. Instead of considering these actions as disturbances, OMA regards them as convenient, naturally available excitation sources that enable dynamic identification under real operating conditions. As a result, OMA is fast, cost-effective, and non-intrusive, allowing tests to be conducted without service interruption and ensuring that the identified modal properties reflect the structure's actual operational state [20].

Methodologically, most OMA approaches are derived from classical Experimental Modal Analysis (EMA) but are reformulated within a stochastic framework, as the input is unmeasured and modeled as random. Consequently, OMA is closely linked to random signal processing concepts, such as correlations and spectral densities, and is typically addressed as a multiple-input identification problem, given that ambient excitation is distributed rather than localized [20].

Despite its advantages, OMA presents practical challenges for bridge applications. Because vibration amplitudes are often small under operational conditions, successful testing requires high-sensitivity, low-noise sensors and a reliable measurement chain. Additionally, the validity of OMA results depends on assumptions about the excitation characteristics and system behavior, such as approximate linearity within the identification range. These assumptions must be verified or at least considered during test planning and result interpretation [7, 20].

Overall, OMA is now widely accepted for modal identification in civil engineering, with numerous successful applications to bridges and other large-scale infrastructure [20]. In bridge engineering, OMA is particularly well-suited to Structural Health Monitoring (SHM) requirements, as it enables repeated or continuous identification without artificial excitation, thus supporting automation and long-term monitoring of dynamic properties [6].

2.2. Modal identification techniques based on output-only data

Output-only modal identification techniques are commonly classified by (i) analysis domain (frequency or time), (ii) approach type (nonparametric, based on spectral or correlation estimates, or parametric, involving model fitting), and (iii) the effective modal complexity, distinguishing between single-degree-of-freedom (SDOF) and multiple-degree-of-freedom (MDOF) cases. This classification system points out the primary trade-offs encountered in practice, such as balancing simplicity and robustness in contrast to reso-

lution and modeling effort [20].

2.2.1. Classification of output-only identification methods

Frequency-domain approaches extract modal information from the spectral density matrix of measured outputs. These methods are regularly employed for initial assessments and for identifying frequency bands containing modal activity. In contrast, time-domain approaches derive modal information from correlation functions or time-series models, enabling modal parameter assessment without measured input forces [20].

Nonparametric methods, based on estimated spectra or correlations, are valued for their ease of use and transparency. In contrast, parametric models employ explicit mathematical models, such as ARMA or state-space representations, and tend to yield improved performance in cases with closely spaced modes or significant noise. However, parametric techniques require additional decisions regarding model order and validation criteria [20].

2.2.2. Frequency-domain nonparametric methods

Initial output-only identification efforts in civil structures frequently used power spectral density (PSD) analysis and Operational Deflection Shapes (ODSs). ODSs characterize the vibration pattern at a specific frequency during operational excitation and can approximate mode shapes under certain conditions, although they may also combine multiple modal contributions [20].

An important advance was the introduction of Frequency Domain Decomposition (FDD) approaches, which utilize spectral density matrix decomposition to isolate dominant modal contributions and enable the practical extraction of modal information from operational data. Nonparametric frequency-domain tools are frequently employed as rapid field checks to obtain preliminary frequency estimates and to inform the design of more advanced identification procedures [12, 20].

2.2.3. Time-domain correlation-based methods: NExT family

Time-domain output-only methods have evolved through procedures based on correlation functions, commonly categorized within the Natural Excitation Technique (NExT) framework. The central concept is that, under appropriate assumptions, output correlation functions derived from ambient or operational responses contain modal information, enabling the application of classical modal extraction techniques without measured input forces. These methods are particularly useful for direct analysis of time histories and for

integrating operational testing with traditional time-domain modal identification concepts [20].

2.2.4. Parametric techniques: ARMA/PEM and state-space (SSI)

Parametric output-only identification methods involve fitting explicit models to measured responses, which can yield improved modal estimates, particularly in cases with closely spaced modes or significant measurement noise. Two principal parametric families are:

ARMA-type representations and Prediction Error Methods (PEM): these provide an "external" description of the system dynamics utilizing fitted time-series relationships.

State-space methods, such as Stochastic Subspace Identification (SSI), offer an internal representation of system dynamics and are now central to contemporary operational modal analysis (OMA) practice. In SSI, modal parameters are extracted from a state-space model and are typically interpreted and selected using tools like the stabilization diagram, which tracks poles across increasing model orders and retains them based on stability criteria [12, 20].

2.2.5. Blind source separation methods

Output-only identification may also utilize blind source separation techniques, such as Second-Order Blind Identification (SOBI). These methods use second-order statistics to separate latent components from measured responses, allowing modal information, including mode shape patterns, to be inferred from the estimated mixing relationships. This approach is particularly attractive for automated or large-scale processing workflows [20].

2.2.6. Practical aspects: validation and data reduction

Operational monitoring often involves large datasets, requiring the identification process to balance resolution, robustness, and computational capacity. Practical measures such as filtering, decimation, and the use of reference channels are commonly implemented to reduce redundancy and increase efficiency. However, inadequate reference selection may result in missing or poorly observed modes, such as local modes or repeated roots not captured by the selected references. Consequently, it is standard practice to combine multiple methods, for instance, employing frequency-domain tools for preliminary analysis and subsequently applying parametric techniques like SSI for final estimates, while utilizing stability and consistency checks to validate the identified modes [9, 20].

2.3. Finite Element modeling of bridges using plate elements

The bridge superstructure under investigation comprises a concrete deck, longitudinal and transverse steel I-girders, and truss elements. The deck, I-girders, and trusses are modeled using a plate and shell discretization, with the girder webs and flanges explicitly represented as shell regions. This modeling approach is informed by established bridge modeling guidance and the objective of achieving direct consistency among finite element analysis outputs and available measured data.

Bridge modeling guidance indicates that plate elements are appropriate for representing plate bending behavior and the associated internal actions, such as moments and shear forces. In contrast, shell elements capture plate bending and shear in addition to membrane actions. Shell elements are primarily employed when local stress levels are of interest, and their results can be interpreted at the mid-surface as well as at the top and bottom surfaces [4]. This approach is consistent with the behavior of bridge superstructures, in which the deck functions as a plate-like component distributing loads, and the steel I-girders may undergo both bending and in-plane effects depending on boundary conditions, bracing, and overall geometry.

A primary motivation for this study is the measurement-driven approach: strain gauges were installed on the webs and flanges of the main longitudinal girders. Modeling I-sections with explicit web and flange shell surfaces enables direct extraction of numerical strain and stress values at sensor locations, thereby eliminating the need for additional assumptions inherent in line-element (beam) idealizations and section-force-based stress recovery. Bridge analysis guidance also highlights the importance of capturing the integrated behavior of girder systems, noting that load transfer and equilibrium in I-girder systems involve interactions among girders and associated components; thus, analyses should consider the integrated behavior of the structural system [4]. For bridges with curved alignments and skewed supports, advanced software employing full three-dimensional analysis models may be necessary to accurately capture load and structural responses [4].

From an implementation standpoint, Abaqus shell modeling is well-suited to this approach. Abaqus performs numerical integration at section points through the shell thickness, where shell element variables, including stress and strain measures, are evaluated and output [5]. The software distinguishes between thin-only shell formulations that enforce Kirchhoff-type constraints and more general shell formulations that account for transverse shear effects, offering flexibility in selecting appropriate shell behavior for thin plate-like

components such as decks, webs, and flanges [5]. Furthermore, Abaqus provides guidance and examples demonstrating the suitability of standard shell elements for thin plate-like structural components where the thickness is small relative to other dimensions.

In summary, the plate and shell modeling strategy adopted in this study is justified because it accounts for the deck and girder components based on their dominant structural actions, including bending, shear, and membrane effects. This approach also supports refined modeling in cases where geometric and structural features render simplified spine models less reliable, and it enables direct and physically meaningful comparisons between measured strains on the girder web and flange and finite element-predicted strains at shell section points and surfaces [4, 5].

2.4. Model updating techniques in structural dynamics

Finite Element (FE) model updating seeks to minimize variations between experimentally identified dynamic characteristics of a structure and corresponding numerical predictions by adjusting selected model parameters. The resulting updated model operates as a calibrated baseline for interpreting monitoring results and conducting further analyses. In civil engineering practice, model updating is frequently carried out in conjunction with Operational Modal Analysis (OMA), since ambient vibrations are readily available during normal service and can be exploited to identify the dynamic response of large structures. Although forced-vibration testing with electrodynamic shakers (vibrodynes) can also be performed, it usually entails traffic restrictions and substantial field logistics, and is therefore seldom adopted for bridges. [20, 21].

2.4.1. General classification of updating approaches

Model updating techniques are commonly classified into direct and indirect (iterative) approaches [15]. Direct methods alter the mass and stiffness matrices to achieve agreement with measured modal properties. Although these methods can yield an exact match for selected modes, the resulting matrices may lack physical meaning. In contrast, indirect methods update physically meaningful parameters such as elastic properties, density, boundary stiffness, and connection stiffness. This approach is generally preferred in civil engineering applications, where interpretability is essential [15].

Indirect methods include both manual and automated strategies. Manual updating depends on engineering judgment and localized adjustments. Automated updating, by con-

trast, is formulated as an optimization problem that minimizes the discrepancy among experimental and numerical modal features [15, 20].

2.4.2. Selection of target modal quantities

The update process requires selecting experimental targets to guide calibration. Common choices are natural frequencies, mode shapes, and damping ratios. In many structural health monitoring (SHM) applications, natural frequencies are preferred because they are more robust under operating conditions and less sensitive to sensor layout and scaling. Mode shapes offer additional spatial information, but their experimental estimation is often less reliable in field monitoring, especially with limited instrumentation. Damping ratios are typically the most uncertain and are therefore employed cautiously [20].

2.4.3. Optimization problem and objective functions

Automated updating is typically posed as a constrained minimization problem, where the decision variables are the updating parameters α:

$$\min_{\alpha} J(\alpha) \tag{2.1}$$

Here, α denotes the vector of updating parameters, and **J** quantifies the mismatch between experimental and numerical modal properties. In practice, **J** may be defined in various ways, such as absolute or relative frequency errors, weighted sums, or norms of residual vectors. The selection of the objective function is critical, as it can substantially affect the stability, robustness, and physical plausibility of the calibrated parameters.

The minimization can be carried out using:

- Local (gradient-based) optimization, typically efficient but potentially sensitive to the initial point and to local minima;
- Global optimization methods (e.g., genetic algorithms, particle swarm), generally more robust to local minima but computationally demanding.

In SHM-oriented applications, computational efficiency is a primary constraint. This consideration motivates the development of approaches that minimize the number of full FE re-analyses required during each update cycle.

The objective function presented in the following equation is the most commonly employed in model updating [21].

$$J = \frac{1}{2}\sum_{i=1}^{M}\left(\frac{f_i^{num} - f_i^{exp}}{f_i^{exp}}\right)^2 \tag{2.2}$$

Here, f_i^{exp} and f_i^{num} represent the experimental and numerical natural frequencies of the i-th mode, respectively, and M denotes the number of identified modes used in the calibration. Alternative formulations may incorporate mode-shape correlation terms or MAC residuals when reliable experimental mode shapes are available [15, 20].

Optimization can be carried out using local gradient-based algorithms or global search procedures. This choice involves a trade-off between the computational cost and robustness to local minima [15].

This work adopts the Douglas–Reid (DR) method for the model updating procedure. The objective function in equation (2.2) represents the formulation most frequently used in DR-based model updating [3, 10, 21, 25]. Its specialization for the DR approach is provided in equation (2.3).

$$J = \frac{1}{2}\sum_{i=1}^{M}\left(\frac{f_i^{DR} - f_i^{exp}}{f_i^{exp}}\right)^2 \tag{2.3}$$

In this context, the superscript $^{\text{DR}}$ indicates that the parameter is estimated using the DR approach, while $^{\text{exp}}$ denotes the parameter identified by OMA.

2.4.4. Parameter choice and identifiability

The selection of updating parameters is critical for solution stability. In field applications, the number of reliably identified modes is usually limited, and over-parameterization can lead to ill-conditioned or non-unique solutions. For this reason, parameters are commonly screened using sensitivity analyses, and smoothing/weighting approaches may be introduced to reduce non-physical parameter drift, see [15, 20].

2.4.5. Douglas–Reid (DR) method

The DR method for model updating [8, 21] is a simplified procedure, first introduced in the early 1980s, that minimizes differences between two datasets by exploring combinations of a prescribed set of updating parameters. In structural applications, it is widely used to reduce mismatches between experimental and numerical natural frequencies [10, 19, 21]. Some studies have also extended its application to address discrepancies in mode shapes

[3, 21, 25]. In the present application, updating is performed using natural frequencies only, as the identified mode shapes were found to be weakly sensitive to the selected parameters [19, 21]. Therefore, the method is described below with reference to frequencies only. The extension to include modal displacement information follows the same rationale and is detailed in [3, 21].

Let $f_i^{\text{exp}}(i = 1, ..., M)$ denote the experimental estimates of the natural frequencies of M modes obtained from an OMA test. For model updating, these quantities are not compared directly with the FE modal frequencies. Instead, a functional dependence of the numerical natural frequencies on the updating parameters is introduced first, as reported in Equation (2.4):

$$f_i^{\text{FEM}} = f_i^{\text{FEM}}(x_1, x_2, x_3, \ldots, x_k, \ldots, x_N) \quad \text{with } i = 1, \ldots, M \text{ and } k = 1, \ldots, N \tag{2.4}$$

Here, N is the total number of updating parameters. To minimize the error between the numerical frequencies predicted by Equation (2.4) and the corresponding experimental values, it is necessary to define the admissible variation ranges for the updating parameters. Within these ranges, the DR method represents the FE response using a quadratic interpolating function that approximates the variation of f_i^{FEM} with respect to the updating parameters:

$$f_i^{DR} = C_i + \sum_{k=1}^{N} \left(A_{i,k} x_k + B_{i,k} x_k^2 \right) \quad \text{with } i = 1, \ldots, M \text{ and } k = 1, \ldots, N \tag{2.5}$$

where f_i^{DR} is the natural frequency of the i-th mode predicted by the quadratic approximation equation (2.5), and C_i, $A_{i,k}$, and $B_{i,k}$ are the 2N+1 unknown coefficients of the interpolating function.

These coefficients are identified through a limited set of preliminary numerical modal analyses, according to the following steps:

- Nominal values x_k^B and lower/upper bounds (x_k^L, x_k^U) are defined for each updating parameter. Each variable is then normalized with respect to the nominal value:

$$\frac{x_k^L}{x_k^B} \leq \frac{x_k^B}{x_k^B} \leq \frac{x_k^U}{x_k^B} \tag{2.6}$$

- The $2N + 1$ unknown coefficients in Equation (2.5) are obtained by solving the

system of equations obtained from evaluating the FE model at a set of sampling points. In particular, one parameter at a time is varied between its lower and upper bounds, while all remaining parameters are kept at their nominal values, and the DR approximation is enforced to match the FE frequencies for each corresponding parameter set:

$$\begin{cases} f_i^{DR}(x_1^B,\ldots,x_k^B,\ldots,x_N^B) = f_i^{FEM}(x_1^B,\ldots,x_k^B,\ldots,x_N^B) \\ f_i^{DR}(x_1^U,\ldots,x_k^B,\ldots,x_N^B) = f_i^{FEM}(x_1^U,\ldots,x_k^B,\ldots,x_N^B) \\ f_i^{DR}(x_1^L,\ldots,x_k^B,\ldots,x_N^B) = f_i^{FEM}(x_1^L,\ldots,x_k^B,\ldots,x_N^B) \\ f_i^{DR}(x_1^B,\ldots,x_k^U,\ldots,x_N^B) = f_i^{FEM}(x_1^B,\ldots,x_k^U,\ldots,x_N^B) \\ f_i^{DR}(x_1^B,\ldots,x_k^L,\ldots,x_N^B) = f_i^{FEM}(x_1^B,\ldots,x_k^L,\ldots,x_N^B) \\ f_i^{DR}(x_1^B,\ldots,x_k^B,\ldots,x_N^U) = f_i^{FEM}(x_1^B,\ldots,x_k^B,\ldots,x_N^U) \\ f_i^{DR}(x_1^B,\ldots,x_k^B,\ldots,x_N^L) = f_i^{FEM}(x_1^B,\ldots,x_k^B,\ldots,x_N^L) \end{cases} \qquad i = 1,\ldots,M \quad (2.7)$$

the previous set of equations in (2.7) can be formulated in matrix form as follows:

$$\{f_i^{FEM}\} = [C]\{K_i\} \qquad i = 1,\ldots,M \quad (2.8)$$

with:

$$[C] = \begin{bmatrix} 1 & x_1^B & x_1^{B2} & \cdots & x_N^B & x_N^{B2} \\ 1 & x_1^U & x_1^{U2} & \cdots & x_N^B & x_N^{B2} \\ 1 & x_1^L & x_1^{L2} & \cdots & x_N^B & x_N^{B2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_1^B & x_1^{B2} & \cdots & x_N^U & x_N^{U2} \\ 1 & x_1^B & x_1^{B2} & \cdots & x_N^L & x_N^{L2} \end{bmatrix} \quad (2.9)$$

$$\{K_i\} = \begin{Bmatrix} C_i \\ A_{i,1} \\ B_{i,1} \\ \vdots \\ A_{i,N} \\ B_{i,N} \end{Bmatrix} \qquad i = 1,\ldots,M \quad (2.10)$$

SO, the unknown coefficients of the DR model can be obtained as follows:

$$\{K_i\} = [C]^{-1}\{f_i^{FEM}\} \qquad i = 1, \ldots, M \tag{2.11}$$

Once the coefficients of the DR approximation are determined, Equation (2.5) can serve as a surrogate for the FE model in evaluating the objective function to be minimized, thus addressing the model updating problem. This formulation is adapted from [21].

2.5. Vehicle–bridge interaction

Vehicle–bridge interaction (VBI) describes the coupled dynamic problem in which a moving vehicle subsystem and a bridge subsystem mutually influence each other through wheel–deck contact forces. Unlike simplified moving-load approaches that neglect vehicle inertia and suspension, VBI formulations capture the feedback loop between vehicle dynamics, mass–spring–damper systems, and bridge response. This feedback becomes significant when vehicle and bridge frequencies, speed, and surface irregularities amplify variations in contact force. A primary modeling challenge is that interaction forces are not spatially fixed; the load location changes continuously, and contact conditions may evolve, including contact, separation, and reattachment. These difficulties demand robust numerical strategies for contact detection and enforcement [2, 14, 22].

Classical VBI studies frequently utilize analytical or semi-analytical formulations, such as modal superposition, moving coordinates, and Dirac delta forcing. These approaches often require handling convective terms, time-varying coupling operators, and potential loss of symmetry in the governing matrices. Saleeb and Kumar [22] propose an alternative approach that facilitates the use of standard commercial finite element (FE) solvers. They model both the primary structure and the moving subsystem as deformable bodies undergoing large relative sliding, coupled through contact and interaction capabilities available in Abaqus and similar FE packages. This method bypasses "non-standard" terms in classical formulations by employing a purely Lagrangian FE treatment with contact, enabling separation and reattachment and providing direct access to response quantities, such as stresses and interaction forces, in the FE output database.

To reduce custom coding effort, as demonstrated in [14], a unified Abaqus-based framework is introduced for the analysis of complex, realistic systems, including high-speed trains and multi-block bridges. This approach demonstrates that a single FE environment eliminates redundant data exchange between solvers and enables the incorporation of nonlinear materials and nonlinear dynamics if necessary.

2.5.1. Contact modeling in Abaqus-based VBI

In Abaqus VBI implementations, wheel–deck interaction is typically simulated using either node-to-surface contact, in which the wheel or tire node acts as the slave contacting the bridge surface, or surface-to-surface contact, where the wheel surface contacts the bridge surface. Both methods depend on contact detection and constraint enforcement during large sliding [2, 14]. Abuodeh and Redmond present a framework that couples quarter-car and half-car vehicle models to the bridge via node-to-surface contact. This configuration grants the wheel node to traverse the bridge surface while contact pressures are enforced using a selected pressure overclosure law. They employ "hard" pressure overclosure behavior as a baseline to minimize penetration and improve numerical robustness in dynamic contact [2].

Lu et al. [14] describe a surface-to-surface contact setting for train–bridge interaction in Abaqus, stressing that contact nodes are not fully known in advance because the active contact region evolves with motion. In practice, the solver forms effective node-to-surface contact pairs during the solution by projecting slave nodes onto the master surface. Within this framework, frictionless tangential behavior is often adopted in baseline validations to focus on normal contact enforcement, although finite sliding is essential to accurately represent the continuous movement of wheel contact along the bridge or track surface.

Penalty-based enforcement is commonly used in these settings, introducing contact stiffness contributions that function as "virtual springs" activated during overclosure. This approach allows for separation when a gap opens and reactivation upon re-contact [14]. Such behavior is particularly relevant in VBI, as fluctuations in contact force due to dynamics or surface irregularities can lead to intermittent loss of contact in idealized cases. Moving-force models cannot represent this phenomenon without additional assumptions [14, 22].

2.5.2. Dynamic integration for VBI contact analyses

Because contact status can change within a time step, dynamic VBI analyses are sensitive to time step size, numerical damping, and convergence tolerances. Abuodeh and Redmond [2] propose a practical multi-step procedure in Abaqus, which includes an initial static settling configuration followed by implicit dynamic traversal. They discuss steps to stabilize contact and mitigate contact chatter inside dynamic simulations. Additionally, they highlight that appropriate solver settings and analysis staging can reduce redundant computational cost while maintaining stable interaction forces during traversal.

Lu et al. [14] adopt the Hilber-Hughes-Taylor (HHT) method in Abaqus for time integration and outline a convergence criterion that checks contact conditions and residual measures, with a time step reduction if convergence is not achieved. These choices are consistent with the general requirement in contact dynamics for introducing controlled numerical dissipation and adaptive time stepping, which help avoid spurious oscillations in contact forces while preserving physically meaningful bridge and vehicle responses.

2.5.3. Surface irregularities and their influence on VBI response

Road surface irregularities are a primary source of excitation in VBI. They directly affect the relative wheel–deck displacement and modulate the normal contact force. In practice, surface roughness can increase dynamic amplification, broaden the response bandwidth, and generate higher acceleration levels in both the vehicle and the bridge. This is especially true when vehicle and bridge frequencies are similar, and interaction is more pronounced [2, 14]. Therefore, a realistic representation of surface roughness is essential for reproducing operational response levels or conducting sensitivity studies on vehicle speed and structural parameters [14].

In FE simulations, surface irregularities are typically introduced either by specifically modeling the rough road profile within the bridge surface geometry or mesh, or by applying equivalent excitation terms. However, explicit geometric roughness can be computationally costly, as achieving adequate resolution often requires fine meshes, particularly when short-wavelength components are present [24]. To resolve these challenges and maintain compatibility with commercial FE software, Zhang et al. [24] propose an effective alternative. Their method conveys road roughness using two external forces: one applied to the vehicle and one to the bridge, rather than discretizing the profile within the FE mesh. This approach is based on the observation that, in the coupled equations of motion, roughness effects can be represented as external excitations acting on each subsystem. The method allows implementation in Abaqus without the need for in-house coding or extremely small elements dedicated solely to capturing the surface profile.

A key advantage of this approach is that it keeps a fully Abaqus-based workflow while accounting for roughness effects. Zhang et al. [24] show that the method accurately reproduces bridge and vehicle responses and reduces computational time, both of which are important for repeated simulations. They generate roughness profiles with a power spectral density (PSD) process, following standard road roughness definitions in [1]. This provides a systematic way to specify surface inputs in VBI analyses without resorting to ad hoc profiles.

3 | Experimental Data and Operational Modal Analysis of the SCSHM In-Service Monitoring Benchmark Bridge

3.1. Description of the monitored bridge

The case study analyzed in this thesis utilizes the SCSHM benchmark [13] for bridge in-service structural monitoring, which was developed to offer a comprehensive, long-term dataset for the validation of both data-driven and model-based structural health monitoring (SHM) methods. The benchmark structure consists of a nine-span bridge with a total length of 291 meters, with individual span lengths ranging from 22.71 meters to 47.54 meters (see figure3.1) [13].

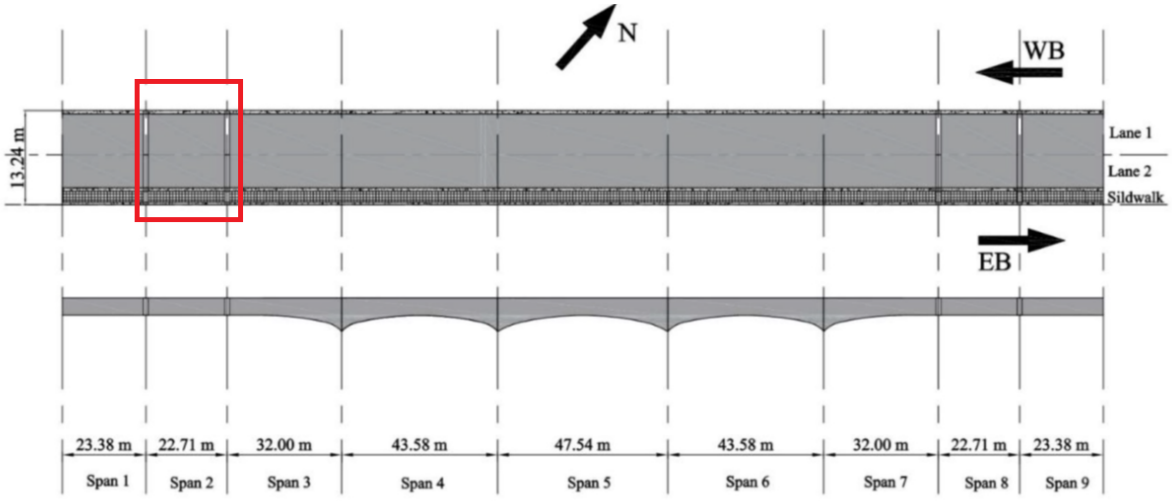


Figure 3.1: Dimensions of the bridge and the location of the monitored section (red square). Image taken from [13].

This benchmark study examines Span 2, indicated by the red square in figure 3.1 and hereafter referred to as the Benchmark span (BM span). The BM span is a simply supported structure accommodating two lanes of traffic: one eastbound lane (direction NE) and one westbound lane (direction SW). The west side of the BM span features a pinned support, while the east side incorporates a movable support to accommodate thermal expansion [13].

The cross-section of the bridge superstructure consists of four haunched steel I-girders that support a fully composite reinforced concrete (RC) deck, as shown in figure 3.2. The deck measures 13.27 m in out-to-out width, with girders spaced uniformly at 3.28 m. Cantilevered sections extend 1.4 m on the left side and 2.03 m on the right side.Figure 3.2 also presents the overall dimensions of the bridge superstructure and the geometry of the barrier walls [13].

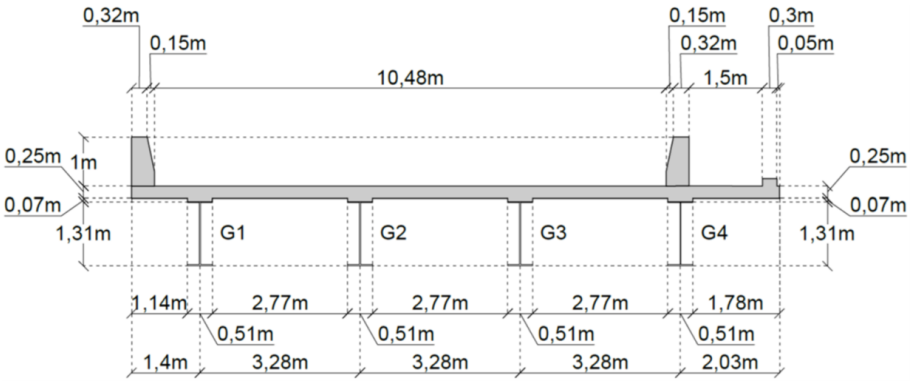


Figure 3.2: BM span superstructure cross-section. Image taken from [13].

3.2. Sensor layout and measured data of the monitored bridge

Instrumentation comprised 32 electric resistive strain gages for monitoring strains and 6 thermocouples for measuring air and structural temperatures beneath the deck. The bridge was instrumented at six cross-sections along its length, designated AA through FF, see Figure 3.3.

As shown in Figure 3.4, instrumentation in Sections AA and EE includes electrical resistance strain gages (ESGs) installed on the support diaphragms midway between adjacent girders to measure strains along the diaphragm axes. These gages provide data on axle weights, spacings, and vehicle speeds, which are subsequently used to determine the weights and spacings of individual axles and closely spaced two- and three-axle groups.

The ESGs installed in Sections FF (Figure 3.5) and BB (Figure 3.6) were mounted on the web of each girder to measure longitudinal strains at the top and bottom flanges and at mid-height between the flanges. These measurements facilitate the determination of gross vehicle weight (GVW), estimation of the transverse load distribution pattern in positive moment regions, and assessment of the degree of composite action between the deck and girders in these regions. Additionally, four thermocouples were installed on the deck at Section BB, see figure 3.6.

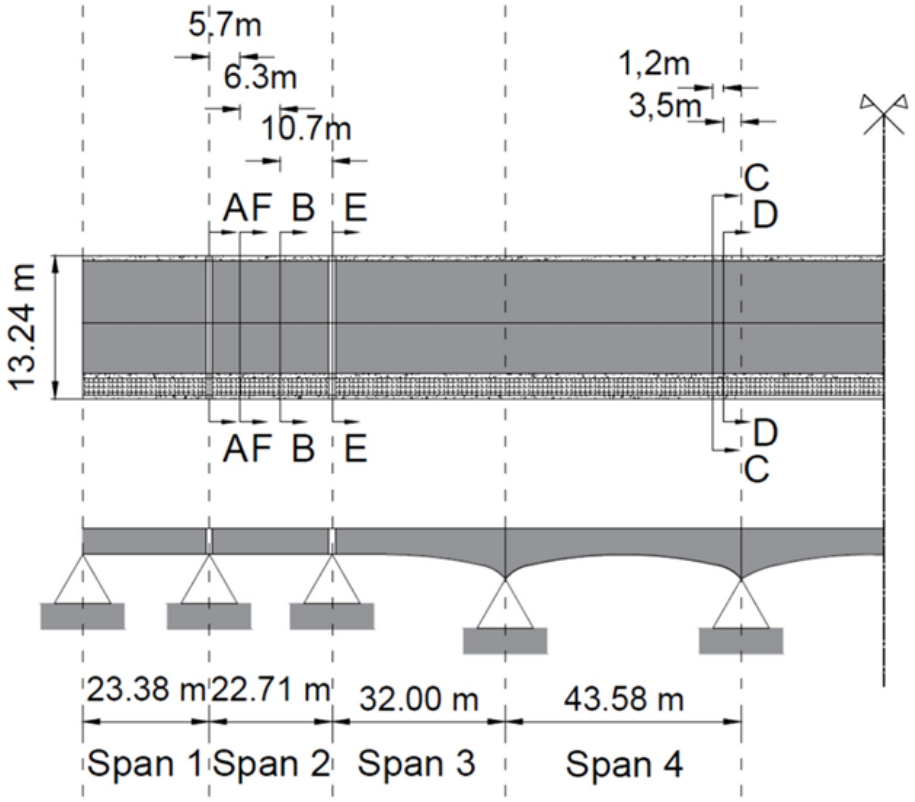


Figure 3.3: Locations of the monitored cross-sections (A-E). Image taken from [13].

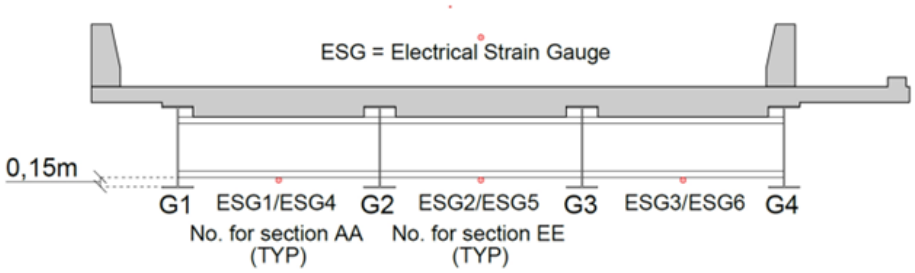


Figure 3.4: Cross-section at the monitored sections AA (Gauges ESG1, ESG2 and ESG3) and EE (Gauges ESG4, ESG5, and ESG6). Image taken from [13].

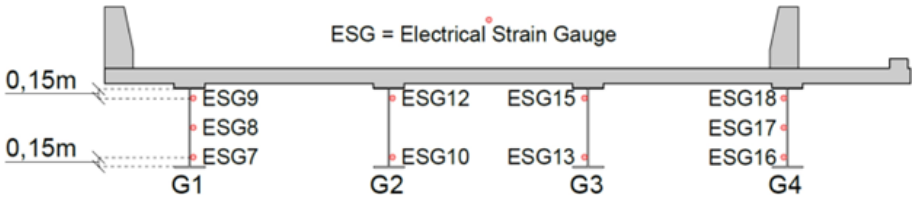


Figure 3.5: Cross-section at the monitored section FF. Image taken from [13].

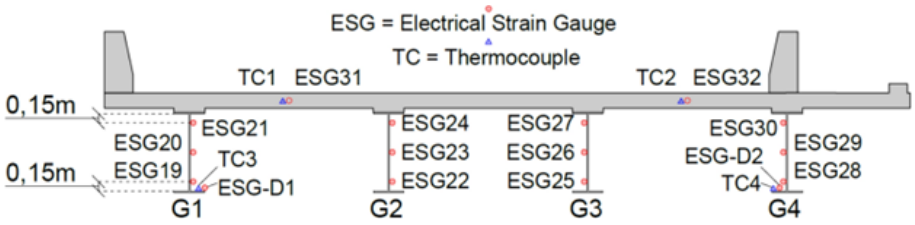


Figure 3.6: Cross-section at the monitored section BB. Image taken from [13].

The data acquisition (DAQ) system provides continuous strain sampling at 200 Hz, image capture during important events, data processing, and transmission to a server. Strain data were filtered using a 7-point moving-average window to remove electrical noise above 60 Hz. For events exceeding 80 microstrain, the sum of mid-span strains from all girders was calculated.

3.3. Overview of Operational Modal Analysis techniques

Operational Modal Analysis (OMA) consists of modal identification procedures that estimate parameters such as natural frequencies, damping ratios, and mode shapes using only response measurements, without the need for excitation force measurements. In the context of bridges, OMA is especially advantageous because controlled excitation is often impractical given the large structural scale, low natural frequencies, and operational limitations [18, 20].

Within the present framework, ambient and operational actions, such as wind and traffic, are treated as naturally available excitation sources, enabling identification under actual operating conditions. Consequently, OMA is generally non-intrusive and well-suited for repeated or continuous monitoring, in accordance with Structural Health Monitoring (SHM) requirements [6, 20]. However, OMA results depend on assumptions regarding excitation characteristics and system behavior, such as approximate linearity within the

identification range. These assumptions must therefore be considered during test planning and interpretation [18, 20].

3.3.1. Classification of output-only identification methods

Output-only modal identification methods are typically classified according to analysis domain (frequency or time), approach type (nonparametric versus parametric or model-based), and effective modal complexity (single-degree-of-freedom (SDOF) versus multiple-degree-of-freedom (MDOF)). This classification stresses the inherent trade-offs between simplicity and robustness versus resolution and modeling effort [20].

Frequency-domain approaches extract modal information from the spectral density matrix of measured outputs and are frequently employed for preliminary screening of frequency bands that contain modal activity. In contrast, time-domain approaches derive modal information from correlation functions or time-series and state-space models, enabling identification directly from response time histories [20].

3.3.2. Frequency-domain nonparametric methods

Initial output-only applications frequently relied on power spectral density (PSD) inspection and Operational Deflection Shapes (ODSs). ODSs characterize vibration patterns at specific frequencies during operational excitation and can approximate mode shapes under favorable conditions, although they may include contributions from multiple modes [20].

Frequency Domain Decomposition (FDD) methods constitute a significant practical advancement. By decomposing the spectral density matrix, dominant modal contributions can be isolated, which enables efficient extraction of modal information from operational data. In practice, these frequency-domain tools are widely used for rapid field checks to obtain preliminary frequency estimates and to guide more advanced identification procedures [12, 20].

3.3.3. Time-domain correlation-based methods (NExT family)

Time-domain output-only methods also encompass correlation-based procedures within the Natural Excitation Technique (NExT) family. When appropriate assumptions are met, output correlation functions computed from ambient or operational responses contain modal information, which allows classical time-domain modal extraction techniques to be applied without measured input forces. This approach is notably useful when working

directly with time histories [20].

3.3.4. Parametric techniques: ARMA/PEM and state-space (SSI)

Parametric output-only identification includes fitting explicit models to measured responses, which often increases performance in the presence of noise or closely spaced modes. Two principal families are commonly referenced: autoregressive moving average (ARMA) representations with Prediction Error Methods (PEM), and state-space approaches such as Stochastic Subspace Identification (SSI). SSI yields an internal representation of system dynamics, and modal parameters are typically selected using stabilization diagrams that track poles across increasing model orders and retain them based on stability criteria [12, 20].

3.3.5. Blind source separation methods

Output-only identification can also utilize blind source separation techniques, such as Second Order Blind Identification (SOBI), which employ second-order statistics to separate latent components from measured responses and infer modal patterns through the estimated mixing relationships. These methods are especially suitable for automated or large-scale automated processing workflows [20].

3.3.6. Practical aspects for SHM-oriented processing

Operational monitoring typically generates large datasets, requiring practical identification strategies that balance resolution, robustness, and computational effort. Common measures include filtering, decimation, and careful selection of reference channels. Inadequate reference selection can result in missing or poorly observed modes. Therefore, it is standard practice to combine methods, using frequency-domain tools for preliminary screening and parametric methods such as SSI for final estimates, supported by stability and consistency checks [9, 20].

3.4. Modal identification of the monitored bridge using pyOMA

PyOMA, including its graphical user interface (GUI), offers an output-only Operational Modal Analysis workflow specifically designed for civil structures. It implements both frequency-domain and time-domain methods. The objective is to estimate the exper-

imental modal parameters of the bridge, with a focus on natural frequencies. When required, damping ratios and mode shapes can also be estimated from the measured vibration response within operational excitation [18].

3.4.1. Data preparation and formatting (input to PyOMA)

Prior to performing modal identification, the measured response signals must be organized in a matrix format containing only numerical columns, with each column representing a measurement channel. The PyOMA GUI accepts file formats such as .txt, .csv, .xls, and .xlsx. These files should not include textual headers, as the GUI requires exclusively numerical data.

In this study, eight representative measurement sets were selected to identify the natural frequencies using the PyOMA software [18]. The strain gauges utilized for identification were installed in Sections BB and FF, corresponding to the mid-span and quarter-span of the bridge, respectively (see Figure 3.3). For Section BB, the gauges ESG19, ESG22, ESG25, and ESG28 were used, while for Section FF, ESG7, ESG10, ESG13, and ESG16 were selected, as illustrated in Figure 3.6 and Figure 3.5.

Table 3.1 presents a partial dataset from the eight strain gauges used for Operational Modal Analysis. This example illustrates the required data format for uploading to the software. Additionally, the software requires two supplementary files for geometrical representation: one containing the coordinates (X, Y, Z) of the selected degrees of freedom and another specifying their nodal connectivity. The user must also assign the measurement channels to the corresponding nodes. For detailed instructions, refer to [18].

Table 3.1: Example of Raw data from the 8 strain gages.

-0.1	0.1	-0.3	-0.1	-0.4	0.1	-0.2	-0.2
0.2	0.2	-0.3	0.1	-0.3	0.2	-0.1	0.1
0.1	0.4	-0.0	-0.0	-0.0	0.2	-0.0	0.3
0.2	0.1	-0.2	0.2	0.1	0.0	-0.2	0.4
-0.1	0.0	-0.2	0.3	-0.2	-0.2	-0.3	0.1
-0.1	-0.1	-0.4	0.0	-0.4	0.1	-0.5	0.3
0.1	0.3	-0.2	-0.1	-0.1	0.1	-0.4	0.1
0.2	0.1	0.1	-0.2	-0.1	0.3	-0.1	0.1
0.1	0.1	0.1	-0.2	-0.1	0.4	0.1	0.1
0.2	0.2	-0.1	-0.2	-0.1	0.2	0.1	-0.0
0.1	0.1	-0.2	-0.3	-0.0	0.3	0.2	-0.0

3.4.2. Pre-processing and peak selection (SV diagram)

PyOMA supports basic signal pre-processing operations such as detrending and decimation. Filtering can also be applied in scripted workflows. Following pre-processing, modal identification typically begins by computing the singular values of the output power spectral density (PSD) matrix, resulting in a singular value (SV) diagram. Peaks in the SV diagram, which correspond to candidate natural frequencies, are identified as shown in Figure 3.7. This peak selection step provides an initial set of frequencies that are afterwards refined using FDD, EFDD, FSDD, or SSI methods [18].

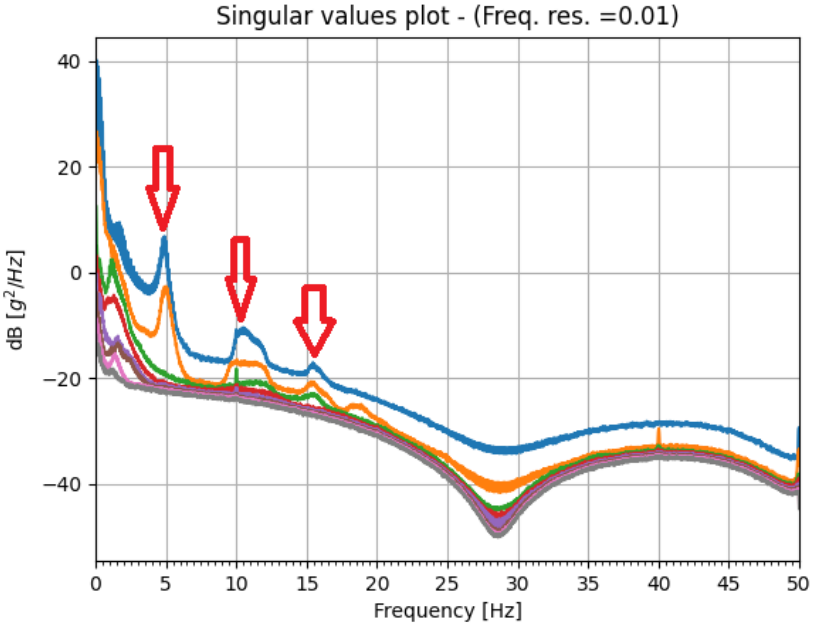


Figure 3.7: Singular Values plot

3.4.3. Frequency-domain identification (FDD, EFDD, FSDD)

After peak selection, PyOMA performs frequency-domain extraction of modal parameters:

- **FDD** extracts (primarily) natural frequencies and mode shapes from the decomposition at the selected peaks.
- **EFDD** extends FDD by estimating damping using the SDOF "bell" extraction and correlation-based decay fitting around each peak.
- **FSDD** is an alternative enhancement that isolates modal coordinates by modal filtering and can yield improved autocorrelation functions and damping estimates.

In the PyOMA GUI, these analyses generate plots, such as those from SDOF bell extraction, and tabulated outputs. All results are automatically saved in the designated results directory.

For the FDD method, the software produces the results presented in Table 3.2.

Table 3.2: FDD results of the first 3 natural frequencies of the monitored bridge

```
Frequencies: 4.680000  5.270000  10.660000
Mode Shapes:
0.826701    0.821522    -0.390080
0.599610    0.389788    0.756156
0.533854    0.116663    0.875404
0.459252    -0.168162   -0.330291
1.000000    1.000000    -0.543498
0.782329    0.526162    0.945500
0.651249    0.131276    1.000000
0.744046    -0.287974   -0.625196
```

For the FSDD and EFDD methods, the software generates tables similar to that shown in Table 3.2. Additionally, these methods produce the following figures.

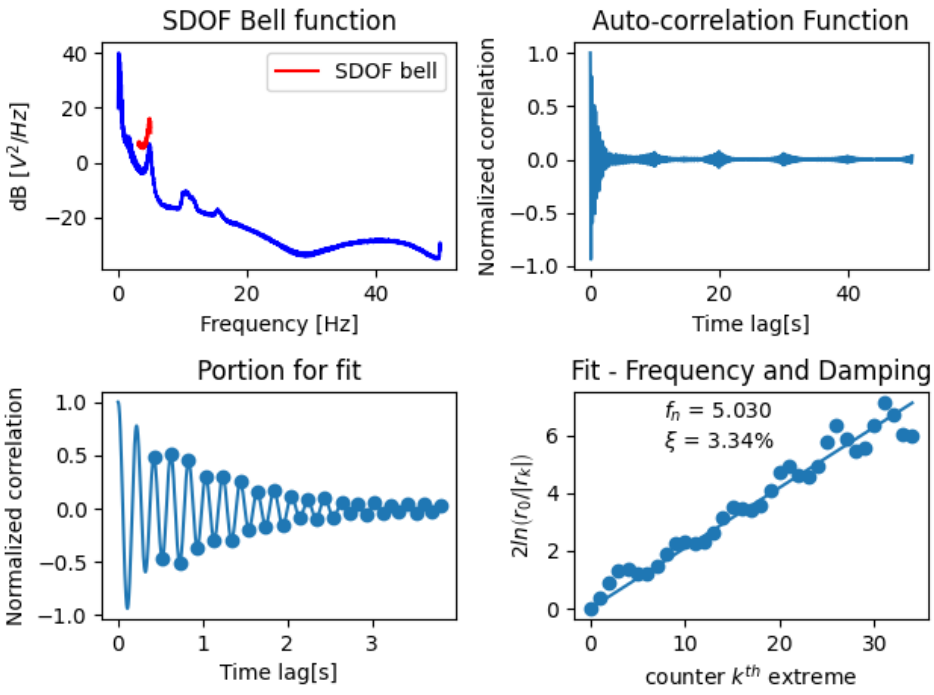


Figure 3.8: FSDD results of the first natural frequencies of the monitored bridge.

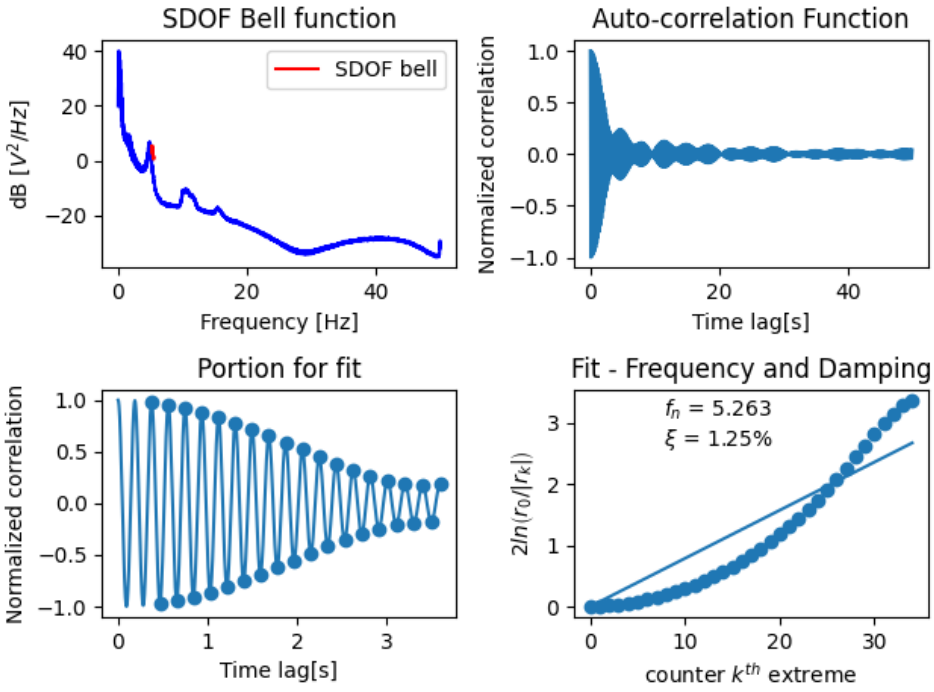


Figure 3.9: FSDD results of the second natural frequencies of the monitored bridge.

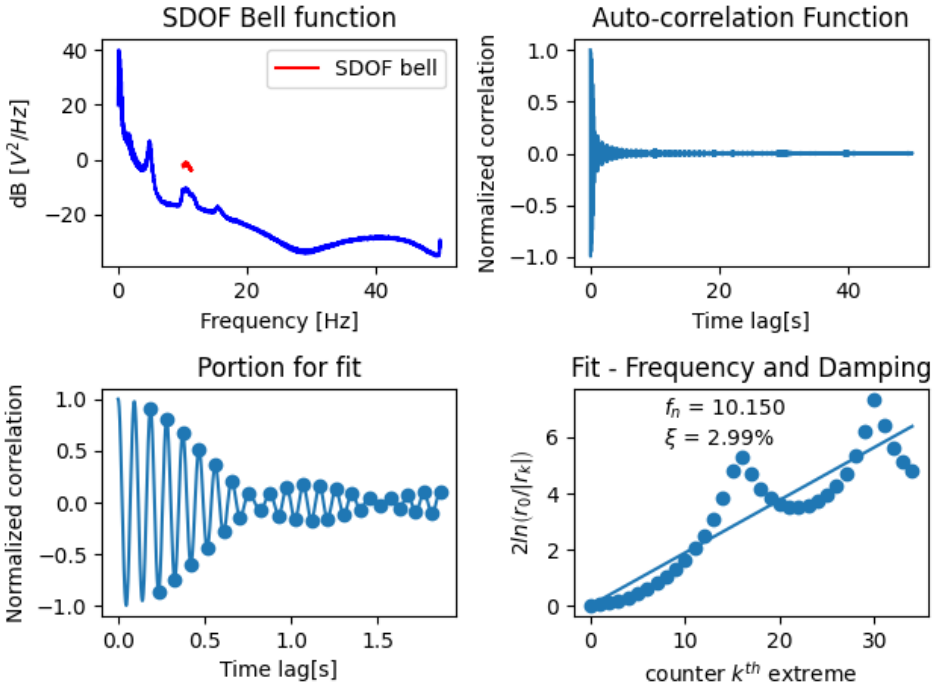


Figure 3.10: FSDD results of the third natural frequencies of the monitored bridge.

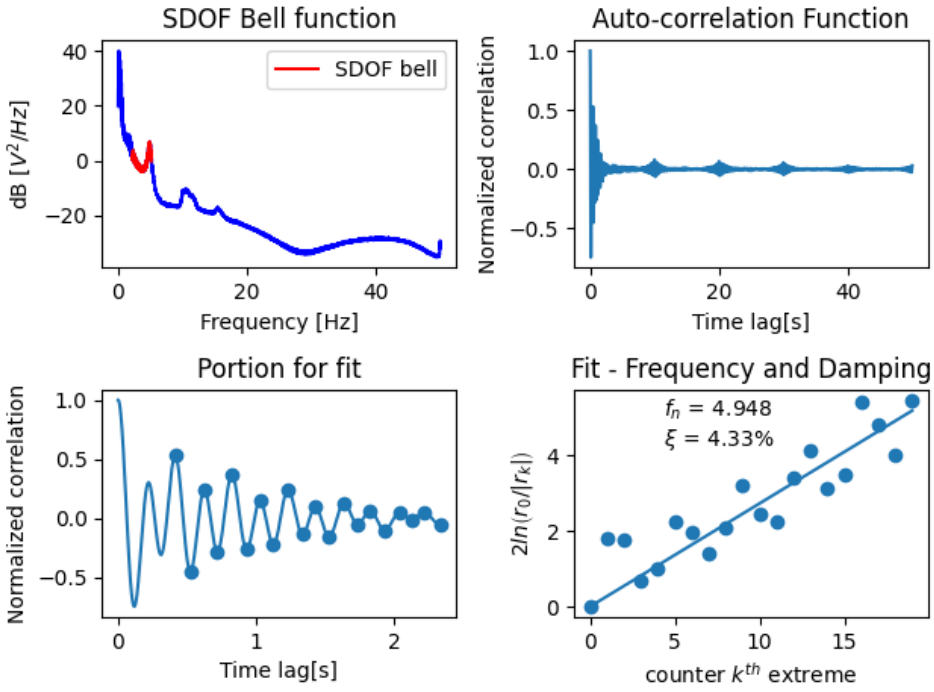


Figure 3.11: EFDD results of the first natural frequencies of the monitored bridge.

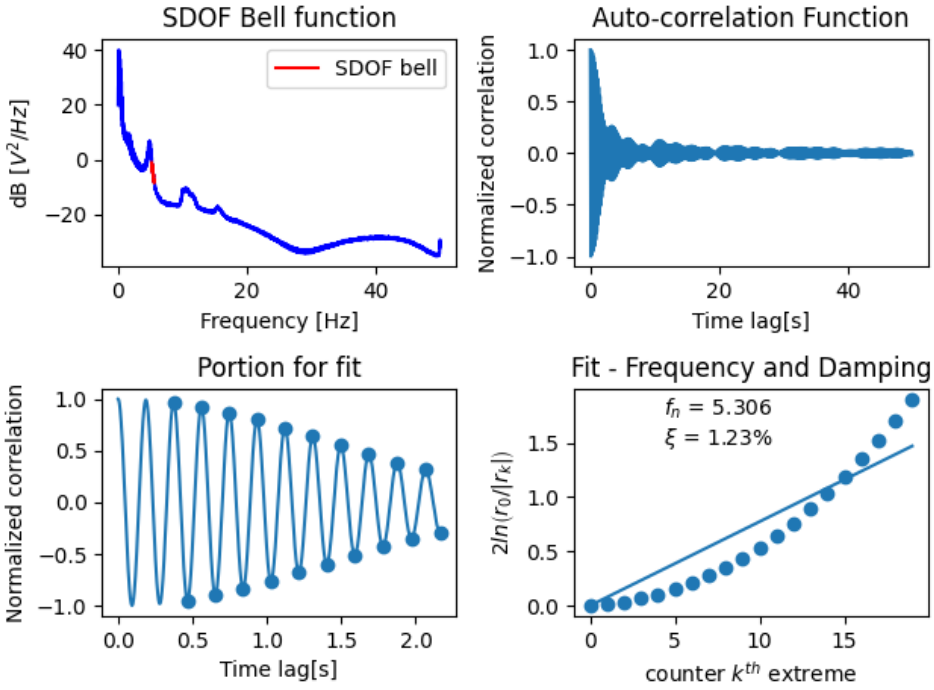


Figure 3.12: EFDD results of the second natural frequencies of the monitored bridge.

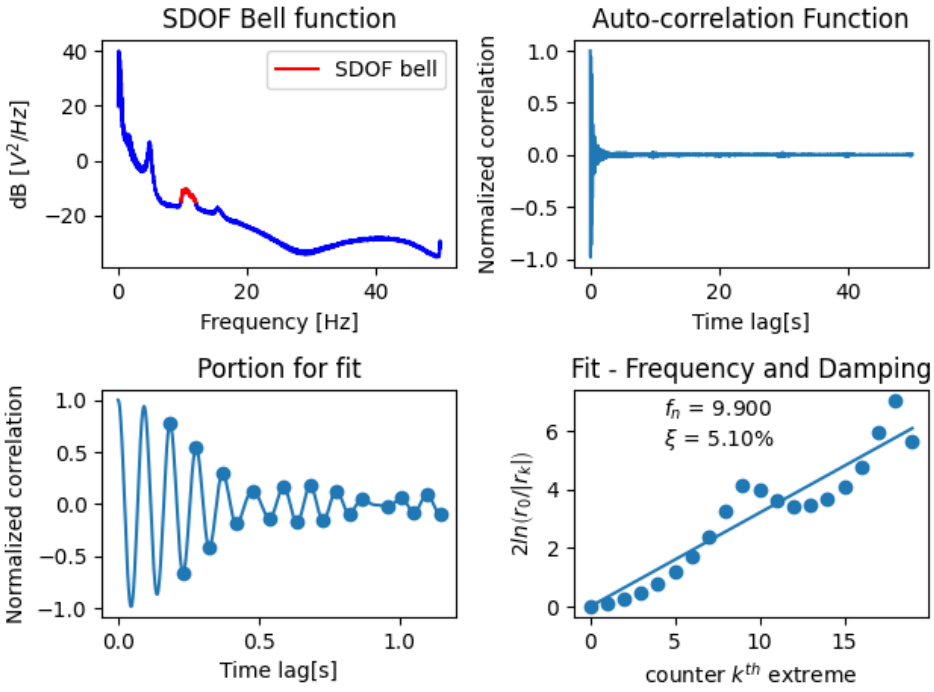


Figure 3.13: EFDD results of the third natural frequencies of the monitored bridge.

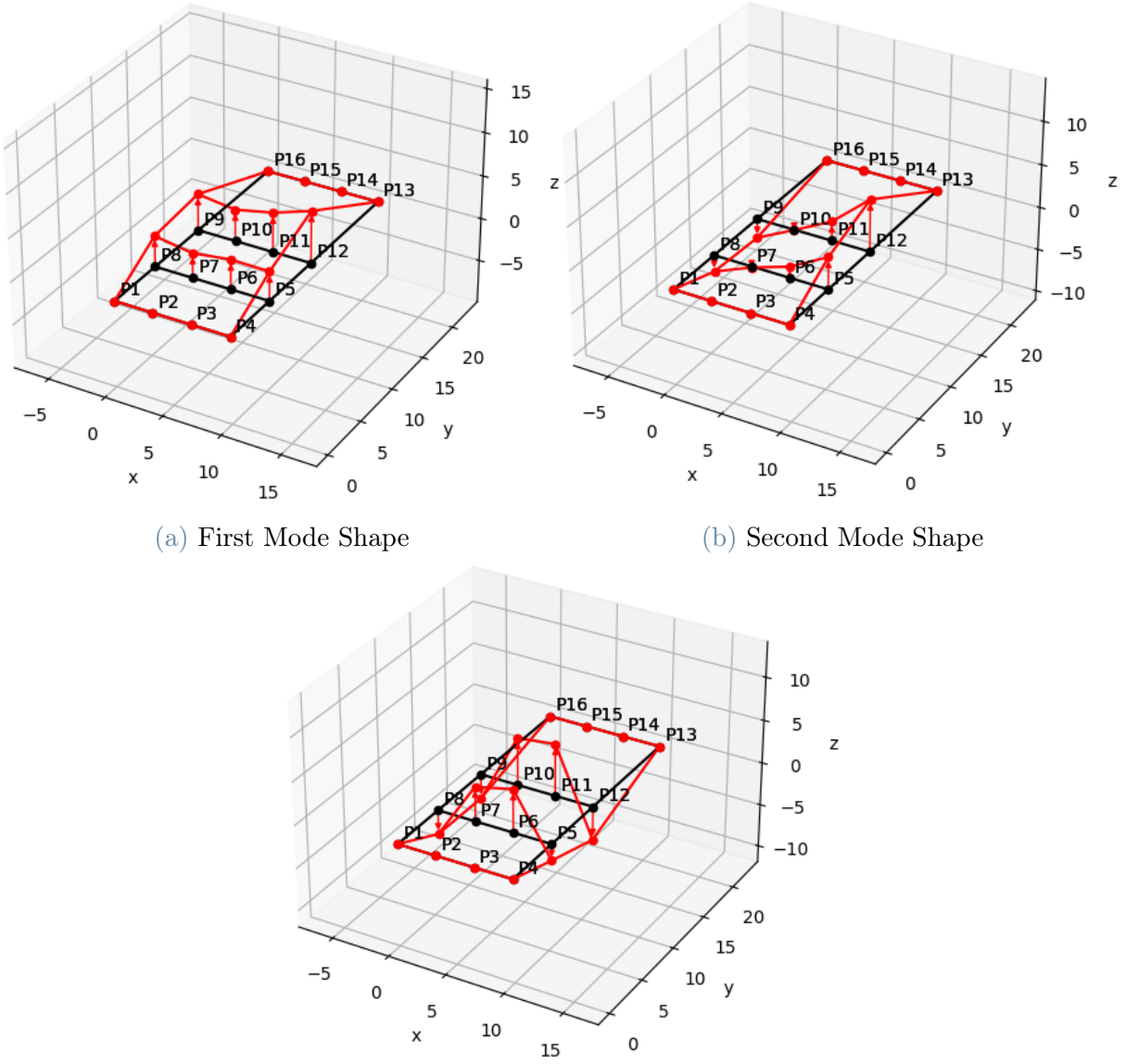
3.4.4. Time-domain identification, SSI-dat and SSI-cov, and stabilization diagram

To increase confidence in the identified frequencies, particularly when peaks are closely spaced or affected by noise, PyOMA implements Stochastic Subspace Identification (SSI) in two forms:

- SSI-dat (data-driven) and SSI-cov (covariance-driven) both producing a stabilization diagram where poles are classified by stability in frequency, damping, and mode shape. The user must select key SSI settings such as the number of time lags (block rows) and the maximum model order, and then interpret stable pole alignments near the previously selected peak frequencies.

The PyOMA GUI reports SSI results by identifying stable-pole alignments near the frequencies selected during the peak-picking step. This approach maintains consistency between FDD-based screening and SSI refinement.

These methods enable the software to compute both the geometrical representation of the mode shapes and the stabilization diagram, as illustrated in Figures 3.14 and 3.15, respectively.



(a) First Mode Shape

(b) Second Mode Shape

(c) Third Mode Shape

Figure 3.14: First 3 mode shapes of the Monitored Bridge.

In Figure 3.15, each identified pole is classified using five labels (0.0–4.0), represented by different colors. These labels indicate the degree of stability of the pole when the model order is increased, according to predefined tolerance thresholds. In particular, the stability checks are performed by comparing the pole properties between consecutive model orders (or within a neighborhood in the stabilization diagram). A pole is considered "stable" when its estimated modal parameters remain sufficiently consistent, and therefore it is more likely to correspond to a physical structural mode rather than numerical noise. The labels have the following interpretations:

- **0.0** / **red:** unstable;
- **1.0** / **dark orange:** stable in frequency;
- **2.0** / **light orange:** stable in frequency and mode shape;
- **3.0** / **yellow:** stable in frequency and damping;
- **4.0** / **green:** stable in frequency, damping, and mode shape.

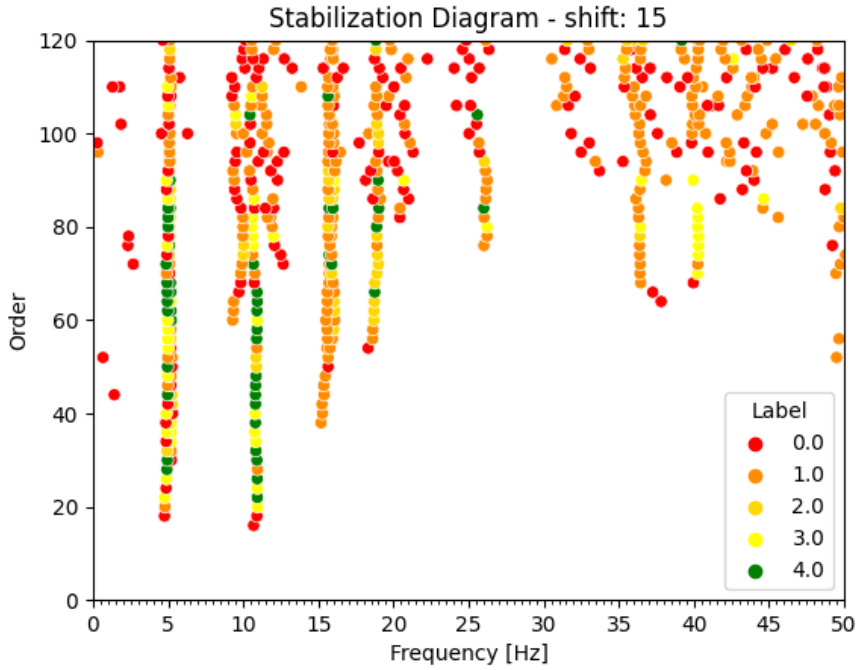


Figure 3.15: Stabilization Diagram.

3.5. Extracted natural frequencies and mode shapes

Based on the OMA processing described in Section 3.4, three natural frequencies were identified for the benchmark span. These results expand upon the benchmark reference study, where the bridge vibration properties were initially characterized from free-vibration accelerations recorded after the passage of a reference truck. In that study, the first experimental natural frequency was reported as 4.65 Hz [13].

To support the interpretation of the monitored response, the benchmark study also developed a finite element (FE) model of the BM span in SAP2000 (v24.2). The model provides a numerical reference for comparison with the identified modal parameters and was formulated as an efficient 3D representation of the composite superstructure. The

reinforced concrete deck was modeled using shell (area) elements, while the steel main girders, transverse trusses, end beams, and concrete barriers were represented with beam (line) elements.

Composite action was introduced by rigidly linking the deck shell nodes to the beam nodes of the girders and barriers. For a consistent structural assembly, most components were first defined in a common global plane and then positioned at their actual elevations through joint offsets for the shell nodes, together with displaced joints and beam cardinal points for the line elements. Where member axes did not intersect directly, rigid-body constraints were used to enforce connectivity. Secondary components that were not explicitly represented in the stiffness model (e.g., stiffeners of the girders) were instead accounted for through added mass in the model mass source [13].

Linear-elastic material properties for steel and concrete were adopted following previous work on the bridge. The mesh was refined in regions where load transfer details were more critical (e.g., near transverse beams and concrete fill), using element sizes of approximately 150 mm locally and 300 mm elsewhere, and the deck-to-transverse-beam load transfer was checked through a point-load verification. With this setup, the SAP2000 model predicted a first-mode frequency of 4.67 Hz, which is in close agreement with the experimental value of 4.65 Hz. The same model reported 5.26 Hz and 10.7 Hz for the second and third numerical modes, respectively [13].

Within the scope of this thesis, the first mode was confirmed by the current OMA analysis, and two additional natural frequencies were identified from the operational strain measurements. This expanded modal target set facilitates subsequent calibration and model updating.

Table 4.2 presents the natural frequencies computed using the PyOMA software. These results are consistent with the first frequency reported in the previous study [13] and also align with the numerical modeling performed in SAP2000.

Table 3.3: Natural Frequencies from Benchmark Paper, SAP2000 Model, and OMA

Mode	**Benchmark Paper [Hz]**	**SAP2000 Model [Hz]**	**OMA [Hz]**
f_1	4.65	4.67	4.64
f_2	–	5.26	5.23
f_3	–	10.7	10.64

The resulting mode shapes, identified using the PyOMA software and shown in Figure 3.14, can be interpreted in terms of their physical significance and compared with the numerical mode shapes reported in the benchmark study [13]. Specifically, the three

extracted modes exhibit the following characteristics:

- **Mode at 4.64 Hz (first vertical bending).** The identified mode shape is dominated by *global vertical bending* of the benchmark span, with maximum vertical deformation occurring in the span region and minimal motion at the supports. The deformation pattern is consistent with a *first bending mode* of a simply supported span, where the response is primarily symmetric and governed by flexural behavior of the composite girder–deck system. This interpretation is aligned with the first mode reported in the benchmark study [13].
- **Mode at 5.23 Hz (first torsional mode).** The second extracted mode exhibits a clear *torsional* character, with opposite-sign vertical displacements across the deck width (i.e., one side moving upward while the other moves downward), indicating rotation of the deck–girder system about the longitudinal axis. Such a shape is typical of the *first torsional mode* of multi-girder composite decks and is expected to appear close to the first bending frequency. The presence of this mode is consistent with the benchmark numerical model, in which the second mode was reported in the same frequency range and interpreted as torsion-dominated [13].
- **Mode at 10.64 Hz (plate-dominated mode).** The mode identified around 10.64 Hz shows a more *localized deformation pattern* than the first two modes, with higher spatial curvature and less resemblance to a single global bending or torsional deformation. This behavior is representative of a *plate-dominated mode*, where the response may be influenced by local bending of the deck plate and the relative deformation between the deck and the girder lines. Such modes are generally more sensitive to modeling assumptions and may therefore show greater variability than simplified beam-type interpretations. Nevertheless, the identified frequency is close to the third numerical mode reported in the benchmark SAP2000 model [13].

4 | Finite Element Modeling of the Bridge

Reliable simulation of bridge response during operational and traffic-related loading requires a calibrated finite element (FE) model. In monitoring-oriented studies, the FE model functions not only as a numerical representation of geometrical and material properties but also as a baseline that must reproduce measured dynamic behavior before it can be confidently used for prediction, interpretation of monitoring results, or scenario analyses. This thesis adopts a monitoring-driven workflow that progresses from output-only field measurements to a calibrated FE model, using experimentally identified modal properties as reference targets for calibration, as discussed in section 2. The reliability of subsequent analyses, such as moving-load or vehicle–bridge interaction simulations, depends directly on the FE model's ability to correctly capture the structural stiffness distribution, mass representation, support conditions, and connectivity details that govern modal response.

Bridge modeling guidance defines a structural model by its structural components, joints, and boundary conditions, as well as material and load modeling; the appropriate level of model discretization is determined by the analysis purpose and required accuracy [4]. These principles are especially relevant for the benchmark bridge examined in this study. The present work adopts a shell-based representation for the deck, steel I-girder components, and truss elements, which allows direct comparison with available strain measurements on girder webs and flanges and decreases reliance on stress-recovery assumptions inherent to beam-only models, as previously discussed in section 2. This modeling approach corresponds to the established practice in the benchmark study, where shell elements are used for deck representation and composite action is enforced through connectivity constraints between deck and girder elements [13].

The modeling process is structured in two phases. The initial phase defines geometry, element types, material properties, discretization level, and boundary conditions. The subsequent calibration phase refines selected parameters to minimize variations between

numerical and experimental modal properties, with a focus on natural frequencies as reliable calibration targets, particularly when mode shapes are weakly sensitive to the chosen parameters, as supported in section 2. The FE implementation uses Abaqus software, which supports shell modeling with through-thickness section-point evaluation of stress and strain outputs and provides suitable formulations for both thin and moderately thick shell behavior. This makes Abaqus appropriate for modal calibration and for subsequent extensions to contact-based moving subsystem simulations [5].

4.1. FE Choices and assumptions

This thesis uses a monitoring-driven workflow in which in-service measurements are processed to identify modal properties and subsequently used to calibrate a numerical model that serves as a predictive baseline for further simulations, as discussed in section 2 and [20]. The FE model development is organized into two phases: the initial modeling phase defines geometry, element types, connectivity, and material properties to represent the bridge system as consistently as possible, while the calibration phase refines selected physically meaningful parameters to ensure that the numerical modal characteristics reproduce the experimentally identified targets.

This approach complies with bridge modeling guidance, which defines structural modeling as the specification of structural components, joints, and boundary conditions, along with material and load representations, and points out that the required modeling fidelity depends on the analysis objective and the response quantities of interest [4, 5]. Within this, the central modeling decision in this thesis is the adoption of a shell-based representation for the concrete deck, steel I-girder components, and truss elements. The bridge superstructure functions as an integrated deck–girder system, and the shell discretization enables the model to represent stiffness distribution and load transfer mechanisms while supporting strain-consistent post-processing at sensor locations, as illustrated in figures 3.5 and 3.6.

For calibration targets, model updating is performed using natural frequencies only. This choice is motivated by the higher robustness of frequency estimates obtained under operational conditions, which are also less sensitive to sensor layout, and by the reduced reliability of experimentally identified mode shapes in typical field configurations or with limited instrumentation as discussed in section 2, [20].

Additionally, for the updating parameters used in this study, mode shapes were observed to show weak sensitivity. Further information on the adoption of a frequency-focused calibration strategy is discussed in section 2 [19, 21].

4.2. Geometry definition and discretization

The structure examined in this thesis is the benchmark span (BM span) of the SCSHM in-service monitoring study, which is a simply supported span of a multi-span bridge carrying two traffic lanes. The BM span features a reinforced concrete deck that is fully composite with four lines of haunched steel I-girders, as well as transverse truss members and barrier walls, see figure 3.2. According to the benchmark documentation, the west end functions as a pinned support and the east end is movable to accommodate thermal expansion in figure 3.1. The documentation also provides the principal cross-sectional dimensions and girder spacing used to reconstruct the section properties and inertia distribution [13].

In line with bridge modeling guidance, discretization is understood as the formulation of the structural system using discrete elements and their interactions so as to reproduce prototype behavior, including stiffness distribution, mass/inertia representation, and load-transfer mechanisms [4]. For the present shell-based model, geometry definition was therefore carried out by locating the mid-surface of each structural component (deck, longitudinal and transverse I-girders, and truss members) directly in its true geometric position. This was done to preserve the correct offsets between adjacent components (e.g., deck mid-surface relative to the girder web/flange mid-surfaces and to transverse members), which is important for obtaining both the static load path and the inertia/stiffness distribution that govern the dynamic response.

This modeling choice directly illustrates a key aspect of the benchmark FE implementation: in the SAP2000 model, elements were first defined in a common plane and then offset to their actual cross-section locations so that composite action and load transfer could be represented correctly [13]. In the benchmark model, this offset strategy was critical because the longitudinal axes of main girders and the axes of transverse members do not necessarily intersect; therefore, rigid-body assignments were required to enforce correct connectivity between members and avoid artificial discontinuities [13]. In the Abaqus model developed here, the same structural requirement applies: the connection strategy is defined to guarantee geometric compatibility and continuous force transfer between the complete set of bridge components while observing the true relative positioning of their mid-surfaces and connectivity is validated by checking that connected components deform compatibly.

A refined geometric representation is particularly important in this thesis because the bridge is modeled primarily with shell elements. The literature review in section 2 supports this approach by noting that shell modeling is appropriate when local stress and strain output is required and when direct agreement with recorded strain data is necessary [4, 5].

As strain gauges in the benchmark are installed on the webs and flanges of the main longitudinal girders, explicitly representing these components with shell regions enables numerical strain extraction at locations and through-thickness section points consistent with the instrumentation, thereby avoiding the stress-recovery assumptions inherent in beam-only idealizations.

Compared with the benchmark SAP2000 model, where web stiffeners were not modeled as stiffness-providing elements and were included only as added masses [13], the present Abaqus shell model adopts a more geometry-faithful description of the longitudinal I-girders. In particular, the stiffening plates attached along the girder web and connecting the web to the upper and lower flanges are included in the geometric representation to better reproduce the real section's stiffness and mass distribution and to reduce the risk of artificial local flexibility in the shell discretization, see figure 4.1. This adoption is consistent with the structural role of stiffeners in steel I-girders (improving web stability as well as guaranteeing the section response is dominated by global bending behavior rather than local plate instability), and it is also consistent with the intent of the benchmark modeling discussion, which explicitly acknowledges stiffeners as a non-negligible mass contribution and treats them carefully even in a simplified model. The barrier walls are also included because the benchmark documentation identifies them as part of the superstructure configuration [13]. From a modeling standpoint, barriers contribute certainly to mass and may add to global stiffness, especially in torsional response and edge restraint effects. Including barriers therefore supports the overall objective of achieving a well-approximated volumetric representation of the bridge system—capturing not only geometry but also the mass and stiffness properties that control both static and dynamic behavior—consistent with general bridge modeling principles [4].

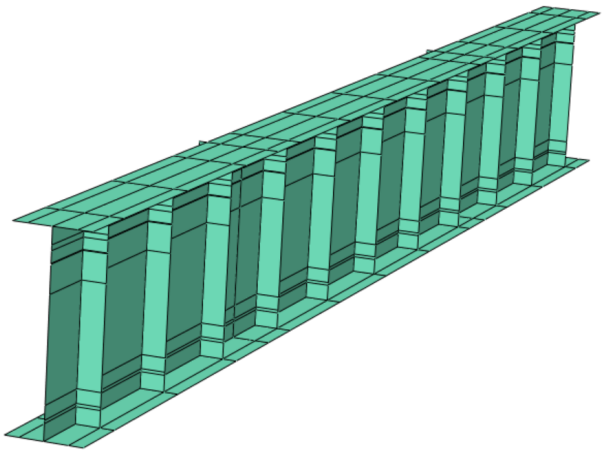


Figure 4.1: Longitudinal I girder and its stiffeners representation

4.3. Plate element formulation and material properties

Bridge analysis practice distinguishes plate and shell elements according to the structural actions they represent. Plate elements primarily capture bending behavior and associated moments and shear forces, whereas shell elements represent combined bending, shear, and membrane actions, making them appropriate when in-plane effects and local stress/strain fields may be relevant [4].Shell discretization is employed in this thesis to model the elements that constitute the BM span. This supports both global dynamic behavior representation and direct strain output extraction at locations comparable to installed sensors [4]. For each section of the bridge, the chosen elements are presented below. Figure 4.2 is presented to better understand the modeling decision and material properties of the bridge are presented in different colors, namely element types.

- **Deck:** The deck was modeled using shell elements, depicted in red in Figure 4.2, representing the concrete deck. Section properties were obtained from technical drawings and the previous study report.
- **Barriers:** The barriers were modeled using shell elements, illustrated in gray in Figure 4.2, corresponding to the concrete barriers. Section properties were obtained from technical drawings and the previous study report.
- **Base Course:** The base course was modeled using shell elements, shown in green in Figure 4.2, representing the asphalt layer. Section properties were obtained from technical drawings and the previous study report.
- **Longitudinal and Transversal I-sections:** These elements were modeled using shell elements, depicted in green in Figure 4.3, representing the steel I-sections. Section properties were obtained from technical drawings and the previous study report.
- **Trusses:** The trusses were modeled using shell elements, illustrated in blue in Figure 4.3, corresponding to the steel truss members. Section properties were obtained from technical drawings and the previous study report.

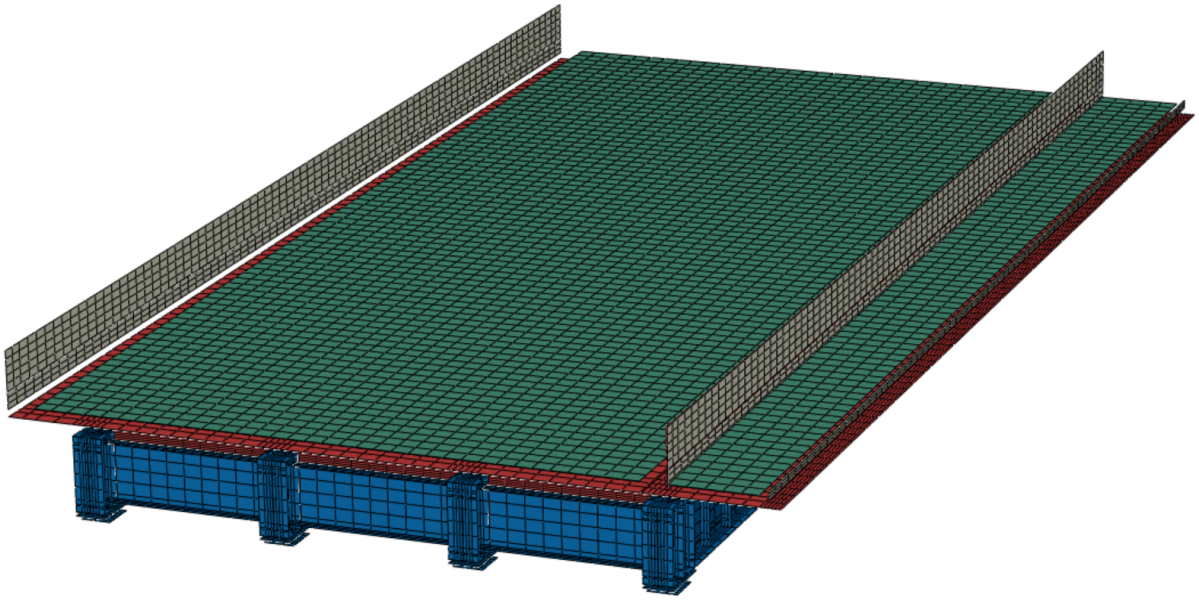


Figure 4.2: BM Span in ABaqus: Upper view

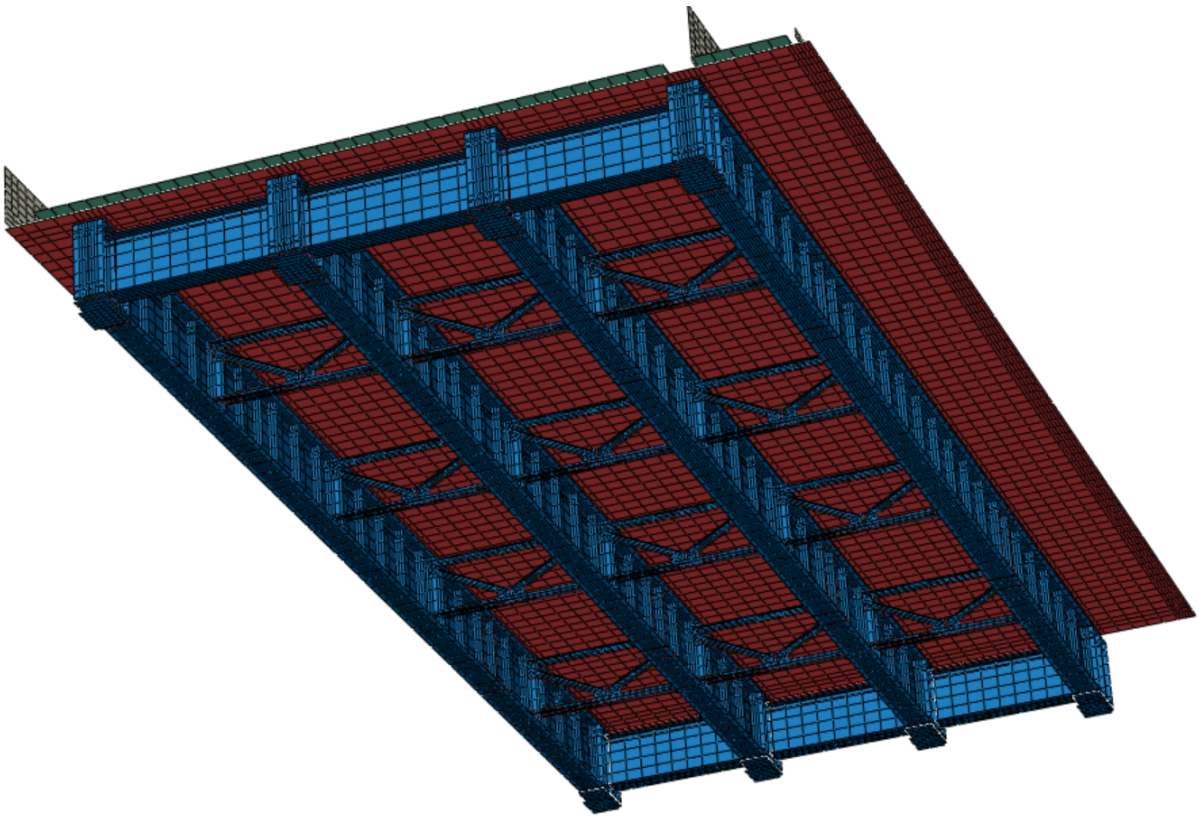


Figure 4.3: BM Span in ABaqus: Lower view

From an implementation perspective, Abaqus evaluates stress and strain in shell elements at section points through the thickness, enabling response extraction at the top, mid-

surface, and bottom integration points as required for comparison in conjunction with recorded strains. Additionally, Abaqus offers general-purpose shell formulations applicable to both thin and moderately thick shell problems, supporting practical modeling of bridge decks and steel plate components [5].

Material properties are assigned to represent the main stiffness and mass contributions of concrete and steel components. The benchmark study reports representative elastic moduli and mass densities selected to match documented bridge characteristics and prior related studies [13]. In this thesis, materials are initially treated as linear elastic for modal analysis and frequency-based updating, consistent with the small-amplitude vibration assumptions underlying operational modal identification, see section 2 and [20]. When secondary components are not explicitly represented as stiffness-providing elements, their influence is accounted for through equivalent mass contributions however in the abaqus model, all of the elements were treated as structural ellements and they are contributing in stiffnes and mass.

Table 4.1: Material Properties

Material	**Young's Modulus [MPa]**	**Poisson's Ratio [-]**	**Density [kg/m^3]**
Concrete – Deck	32627	0.2	2548.5
Concrete – Barriers	32627	0.2	2548.5
Steel – I girder and Truss	200000	0.3	7848.8
Asphalt – Base course	4000	0.3	2300

4.4. Boundary conditions (BCs) and constraints

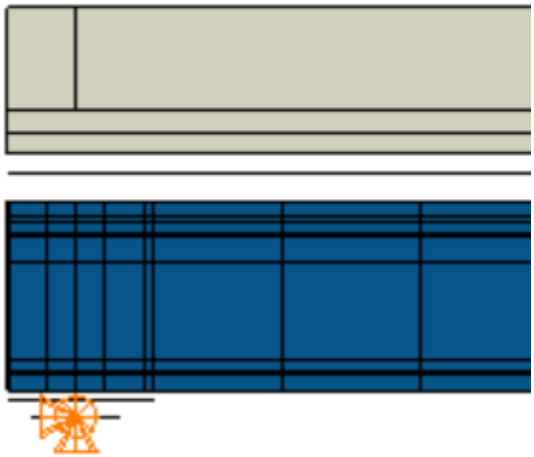
Boundary conditions and connectivity constraints considerably alter modal characteristics and must therefore accurately reflect the as-built structural support behavior. Bridge modeling guidance addresses standard end-support idealizations, such as pinned, roller, and fixed supports, and notes that partially restrained behavior can be simulated with springs when flexibility is relevant [4]. In this thesis, boundary conditions are defined to represent support behavior in a manner consistent with the modal targets identified from monitoring data, as discussed in section 2 and [20]. The BM Span is a simply supported span, the west side of the BM span has a pinned support and the east side has a movable support that allows for thermal expansion, see figure 4.4.

Figure 4.4: BM span supports. Image taken from [13].

In Abaqus, benchmark support conditions are simulated through modeling bearing regions as thick plates positioned above the support devices. Boundary conditions are applied along the contact line between each support device and its corresponding thick plate. This approach delivers stable and realistic support behavior in the shell-based model and prevents unintended rotation about the longitudinal axis of the span.

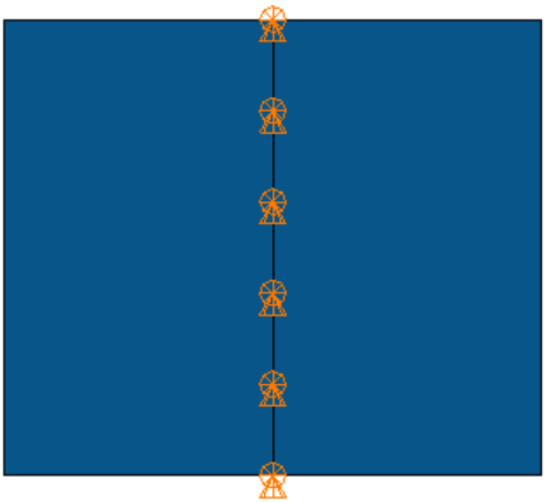
According to the benchmark definition of BM-span supports, the pinned support restrains all three translational degrees of freedom: longitudinal, transverse, and vertical. On the other hand, the movable support restrains only vertical and transverse translations, which allows longitudinal translation to accommodate thermal expansion without introducing axial restraint.

Figure 4.5 illustrates the assignment of the specified boundary conditions.

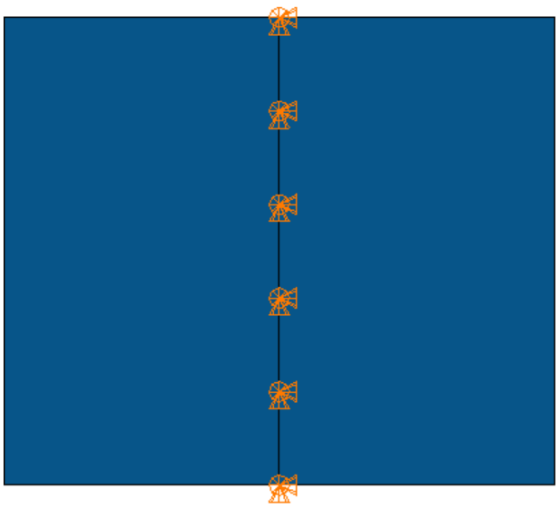


(a) West Support: Lateral view

(b) East Support: Lateral view



(c) West Support: Supporting Line



(d) East Support: Supporting Line

Figure 4.5: Representation of the Boundary Conditions in Abaqus.

Connectivity constraints are equally critical. The benchmark FE model enforces composite behavior by rigidly connecting deck shell nodes to the supporting members, including offsets and rigid-body assignments where member axes do not intersect. Although the Abaqus model developed here uses a more explicit shell representation for girder components to support strain-consistent output, the same functional requirement applies: compatibility between deck and girders must be ensured so that load transfer is realistic and modal response is not distorted by unintended releases or artificial slip [13].

Abaqus provides coupling and constraint tools that enable enforcement of displacement compatibility between reference nodes and node sets, and can represent rigid or semi-

rigid connections depending on modeling requirements [5]. These constraints are applied selectively to guarantee compatibility and to avoid artificial stiffness increases that could bias modal frequencies.

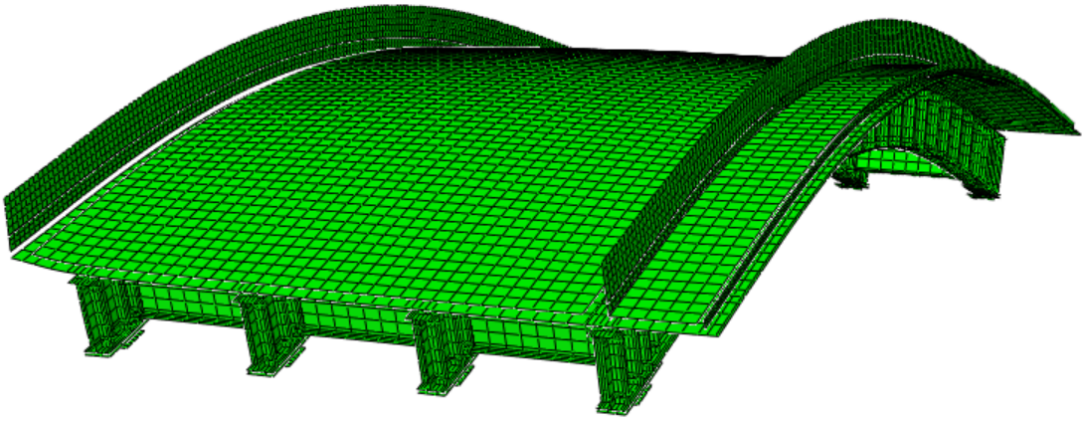
4.5. Initial modal analysis results

The initial finite element (FE) model is subjected to eigenvalue analysis to determine baseline natural frequencies and mode shapes before calibration. This process establishes the initial discrepancy with the experimentally identified modal properties obtained from operational modal analysis (OMA) (Section 2, [20]) and serves as the starting point for the damage recognition (DR) updating procedure. Bridge modeling literature identifies eigenvalue analysis as a standard method for characterizing structural dynamic behavior and informing subsequent simulations [4]. The benchmark study indicates that the first mode corresponds to global bending, while the second mode is characterized by torsional behavior; numerical results for these modes show good agreement with experimental observations [13].

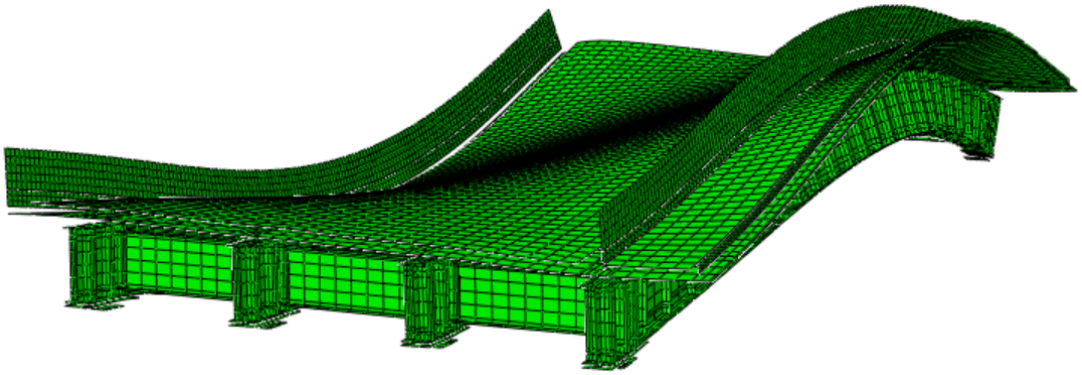
Table 4.2 presents the natural frequencies of the bridge FE model constructed with the initial assumptions, while Figure 4.6 displays the corresponding mode shapes.

Table 4.2: Modal frequencies obtained from the first FE model of the bridge

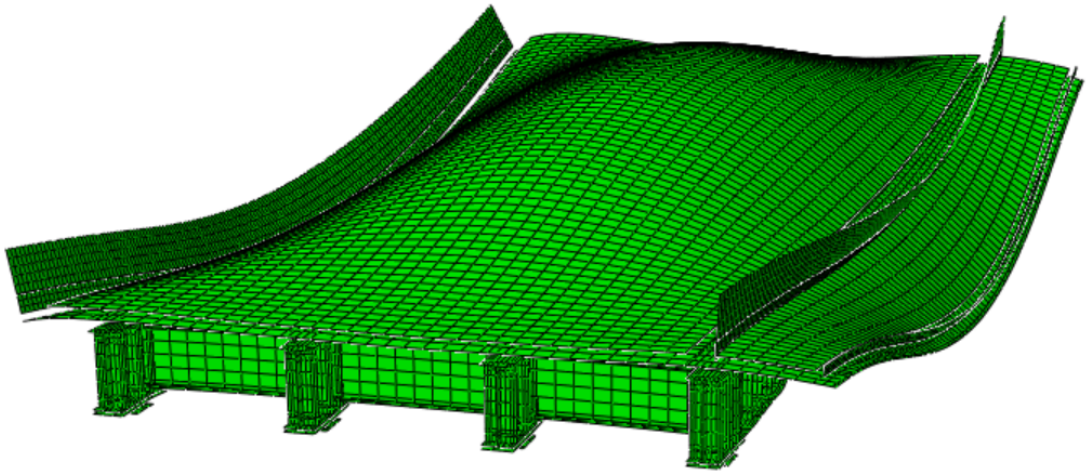
No. Mode	Modal Mode Type	Numerical frequencies [Hz]	OMA - Experimental Frequencies [Hz]
1st	1st Vertical	4.5824	4.64
2nd	1st Torsional	6.8288	5.23
3rd	Plate Mode	11.785	10.64



(a) First Mode Shape



(b) Second Mode Shape



(c) Third Mode Shape

Figure 4.6: First 3 mode shapes of the 1st FE model.

The previous model did not explicitly include bearing devices. Due to discrepancies between the numerical modal properties and the experimental natural frequencies identified by OMA, the model is refined through incorporating the bearing devices. The objective

is to examine the influence of the bearings on the FE model's natural frequencies and mode shapes, and to determine whether their inclusion reduces the mismatch with the experimentally identified frequencies.

Figure 4.4 shows the bearing supports. In Abaqus, these supports are idealized as rigid body elements because their local deformation is negligible compared to the bridge's global response. Each support is represented by a reference point, to which the corresponding support surface is kinematically coupled. Boundary conditions are applied exclusively at these reference points. Figure 4.7 illustrates the numerical representation of the bearing devices and the boundary conditions enforced on their reference points.

(a) West Support: Lateral view (b) East Support: Lateral view

Figure 4.7: Boundary Conditions with Bearing Devices in Abaqus.

Figure 4.8 presents the second FE model, which includes the bearing devices.

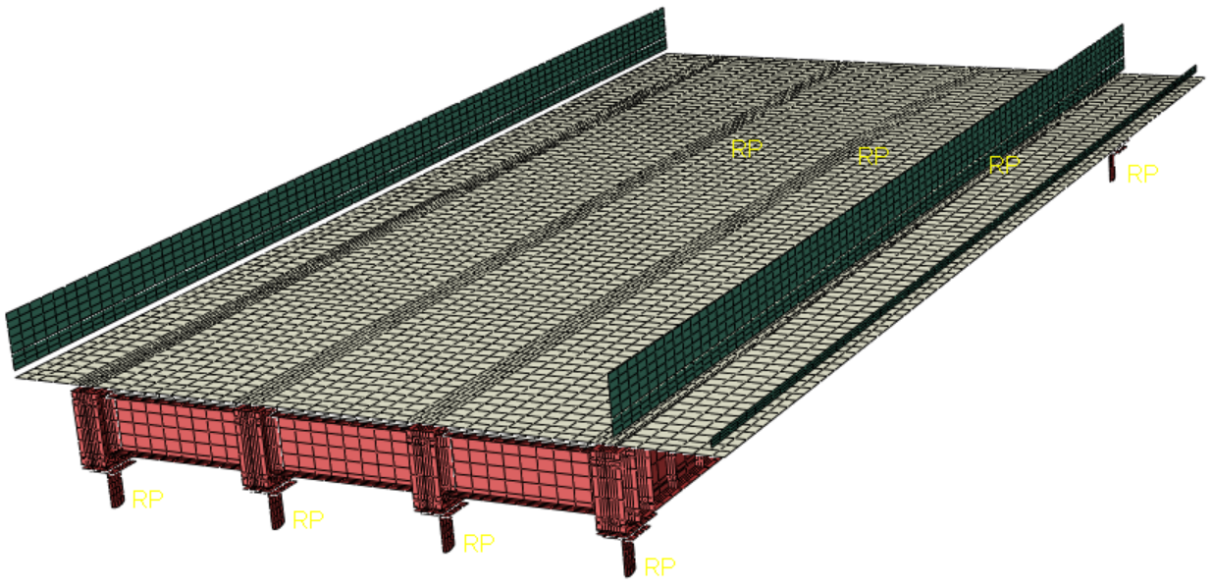
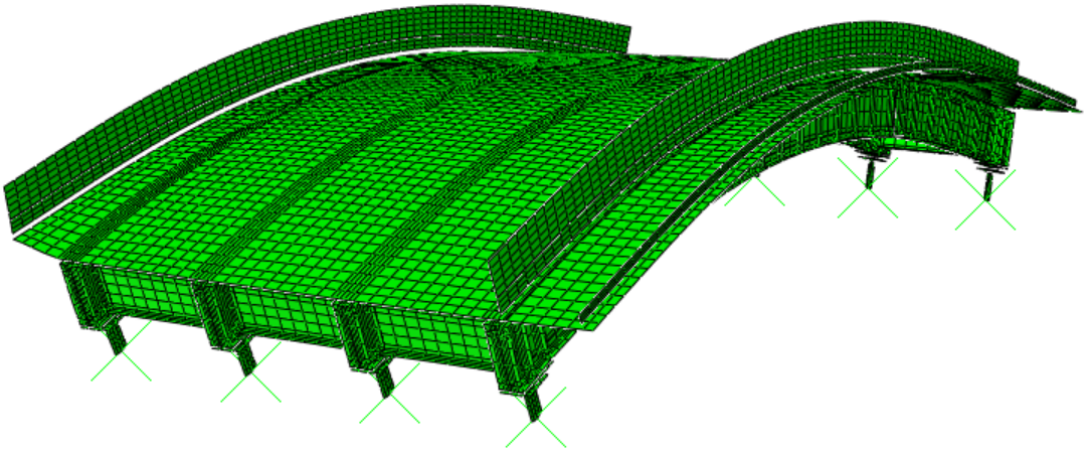


Figure 4.8: FE model with bering devices

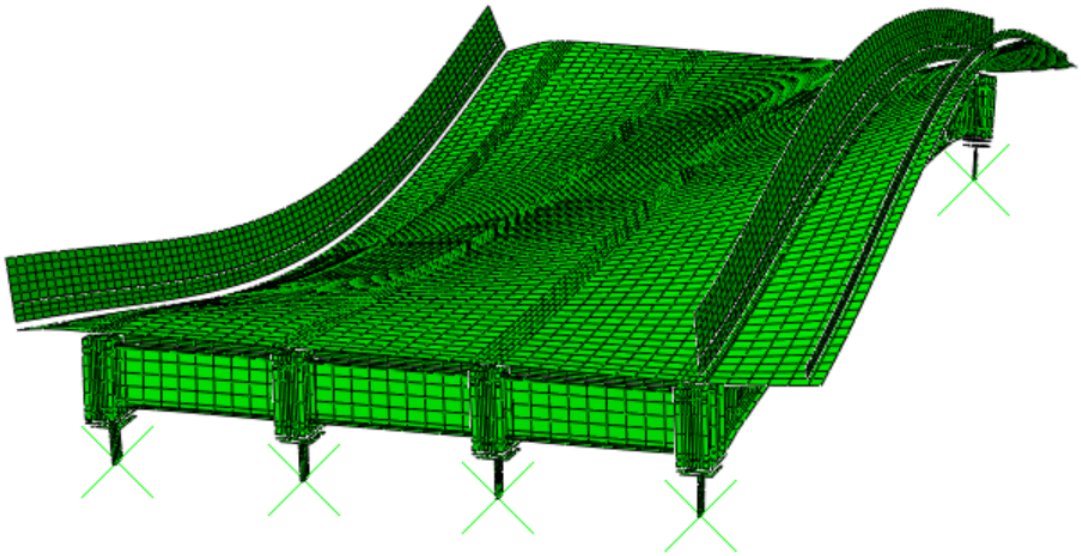
Table 4.3 lists the natural frequencies of the bridge FE model with bearing devices, and Figure 4.9 presents the corresponding mode shapes.

Table 4.3: Modal frequencies obtained from the FE model of the bridge with bearing devices

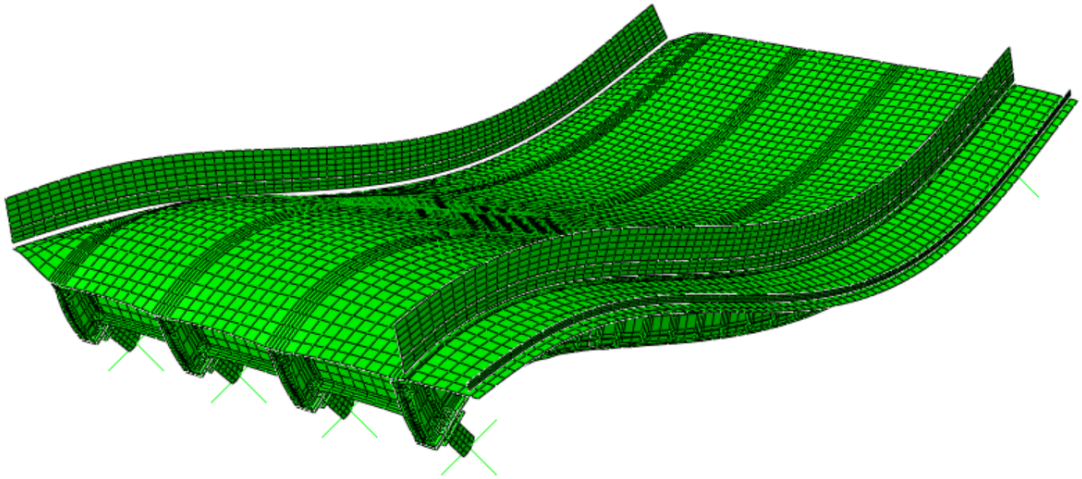
No. Mode	Modal Mode Type	Numerical frequencies [Hz]	OMA - Experimental Frequencies [Hz]
1st	1st Vertical	4.0039	4.64
2nd	1st Torsional	6.0998	5.23
3rd	2nd Vertical	9.0009	10.64



(a) First Mode Shape



(b) Second Mode Shape



(c) Third Mode Shape

Figure 4.9: First 3 mode shapes of the FE model of the bridge with bearing devices.

Both FE variants reproduce the expected low-order global behaviors, specifically the first vertical bending and first torsion modes. Therefore, the comparison focuses on (i) frequency shifts resulting from support idealization and (ii) the consistency of the identified mode shapes with the experimental (OMA) reference. As shown in Tables 4.2 and 4.3, the inclusion of bearing devices leads to a reduction in the first two natural frequencies (from 4.5824 Hz to 4.0039 Hz for the first vertical mode, and from 6.8288 Hz to 6.0998 Hz for the first torsional mode), indicating a softer global response due to increased support compliance. Additionally, the third mode in the bearing model shifts from a higher-frequency local or plate-type mode (11.785 Hz) to a second global vertical mode (9.0009 Hz), suggesting that the bearing representation influences both the frequency values and the ordering and relative contribution of global and local deformation responses.

The experimentally identified mode shapes and those from the BM span SAP2000 model agree more closely with the results from the first FE model. Although the bearing model offers an alternative and potentially more detailed support representation, its inclusion does not enhance the overall agreement with the experimental modal behavior when both frequency values and mode-shape characteristics are considered. Consequently, the first FE model is selected as the baseline configuration for subsequent calibration.

5 | Model Updating Based on Experimental Modal Data

The BM span benchmark bridge is a simply supported structure equipped with strain and temperature instrumentation. The dataset was released to facilitate strain-based model updating and system identification under operational and environmental variation [13].

Despite the development of a detailed finite element (FE) model based on drawings and refined geometric hypotheses, uncertainties remain regarding material properties, dead-load representation, and geometric idealizations. In the benchmark study, the SAP2000 FE model predicts a first-mode frequency of 4.67 Hz, which closely agrees with the experimentally obtained value of 4.64 Hz. The second and third modes are also reported as 5.26 Hz and 10.66 Hz, respectively [13].

Table 5.1: Comparison of Modal frequencies: SAP2000 model and OMA analysis

No.	**OMA - Experimental Frequencies [Hz]**	**SAP2000 - Numerical frequencies [Hz]**	**Error [%]**
1st	4.64	4.67	0.7
2nd	5.23	5.26	0.6
3rd	10.64	10.66	0.2

Table 5.1 presents a comparison between the natural frequencies identified from the operational modal analysis (OMA) campaign and the numerical frequencies predicted by the SAP2000 model, including the corresponding relative errors. The differences are minor, below 1%. These results demonstrate that the baseline SAP2000 model accurately reproduces the global dynamic characteristics of the BM span. Consequently, no further calibration of the SAP2000 model is necessary based on modal frequencies, and it can be used as a reliable reference for later investigations of the monitored structure.

The next stage requires conducting a coupled Vehicle–Bridge Interaction (VBI) analysis, extracting strain time histories at sensor locations, and comparing these through measured strains from the benchmark dataset. This approach requires modeling the vehicle and

bridge as two interacting dynamic subsystems. Since SAP2000 does not support a fully coupled vehicle–bridge formulation, the BM span was implemented in Abaqus, which enables such simulations.

In vibration-based monitoring, frequency estimates obtained from OMA are generally the most robust quantities, whereas experimentally identified mode shapes are less reliable due to limited spatial observability and scaling challenges. This observation motivates the adoption of a frequency-based updating formulation, consistent with the approach commonly used in Douglas–Reid (DR) model updating, where natural frequencies are preferred, particularly when mode shapes have weak sensitivity to the selected updating parameters [19, 21].

5.1. Overview of the Douglas–Reid (DR) method

The Douglas–Reid (DR) method is an indirect, automated model updating procedure originally introduced in the early 1980s to minimize the discrepancy between two datasets by exploring combinations of a prescribed set of physically meaningful updating parameters. In structural dynamic applications, it is widely used to reduce the scatter among experimental and numerical natural frequencies [8, 19, 21].

Model updating is formulated by introducing a functional dependence of the numerical natural frequencies on a set of(N) updating parameters $x_k (k = 1, ..., N)$. Let $f_i^{\exp} (i = 1, ..., M)$ denote the experimental estimates of the natural frequencies. A complete discussion of the Douglas–Reid method is provided in Section 2. Within admissible bounds for each updating parameter, the DR method approximates the FE response by a quadratic interpolating function, as reported in Equation 2.5.

The main advantage of DR is that, once the coefficients in Equation 2.5 are identified from a limited set of FE analyses, the surrogate model can replace repeated FE eigenvalue extractions during the optimization stage. In other words, after constructing the surrogate, the minimization of the objective function can be carried out without further FE analyses, which greatly decreases computational cost while retaining physically interpretable updating parameters (see Section 2 and [8, 10, 19, 21]).

5.2. Selection of updating parameters

In indirect model updating, parameters are chosen to preserve physical meaning (e.g., elastic properties, density, boundary or connection stiffness), which is generally preferable in civil engineering applications [15]. Based on the modeling uncertainties encountered

in the FE development and on the physical characteristics of the BM span (composite steel–concrete superstructure with barrier walls and asphalt surfacing), six parameters were selected:

- Young's modulus of deck concrete
- Mass density of deck concrete
- Young's modulus of barriers concrete
- Mass density of concrete barriers
- Mass density of asphalt
- Span length of the BM span

The benchmark description stresses the composite deck–girder system and the presence of barrier walls. The dataset and FE model details underline the necessity of correctly representing geometry and mass distribution for dynamic analyses [13].

Stiffness-related parameters, such as Young's modulus, were selected to capture uncertainty in the effective concrete stiffness contribution. Density parameters, including mass density, were introduced to address uncertainty in the mass representation of structural and non-structural components. Span length (L) was included as a geometric parameter to account for probable discrepancies resulting from model idealization of end regions and the effective modeled span. This selection is consistent with indirect updating principles and maintains a moderate parameter space relative to the number of target modal quantities [15].

5.2.1. Sensitivity Analysis

Sensitivity analysis assesses the influence of each parameter on the dynamic characteristics of the structure. The objective is to identify which parameters most significantly affect the response of the monitored bridge, as determined by dynamic analysis of the numerical model. The procedure begins by examining the individual effect of each updating parameter, followed by simultaneous variation of multiple variables to determine the optimal solution.

The sensitivity coefficient for each model updating parameter $S_{n,p}$ is computed for n frequencies and p updating parameters, as shown in Equation 5.1. This coefficient quantifies the influence of each updating parameter on the dynamic behavior of the structure. Each parameter is varied by 10%, and the corresponding sensitivity coefficient is calculated.

$$S_{i,j} = 100\frac{X_i}{f_i}\frac{\Delta f_i}{\Delta X j} \quad \text{with } i = 1,\ldots,n \text{ and } j = 1,\ldots,p \tag{5.1}$$

where

$S_{i,j}$: sensitivity coefficient

X_i : value of initial updating parameter i

f_i : Initial modal frequency i

Δf_i : variation of frequency after having change the initial parameter i

ΔX_j : variation of updating parameter j

$i = 1,\ldots,n, \quad j = 1,\ldots,P$

Figure 5.1 presents the results of the sensitivity analysis. The mass density of the deck exhibits the greatest influence, while the other parameters also have a significant impact to the structural response. The results confirm the appropriateness of the selected parameters as updating variables in the modal updating procedure.

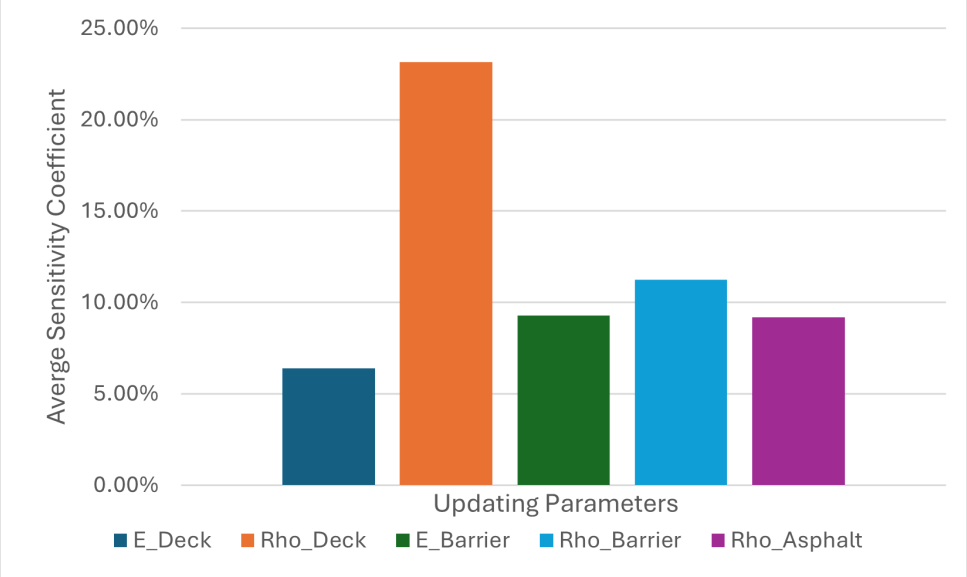


Figure 5.1: Sensitivity Coefficient regarding the updating parameters

5.2.2. Objective function definition

Automated updating is formulated as a constrained minimization problem, where the decision variables are the updating parameters and are determined by means of minimizing an objective function [15]; see Equation 2.1.

Consistent with Section 2 and established practice in DR-based model updating, the objective function adopted is the relative frequency least-squares formulation (Equation 2.5), which is the most frequently used objective function in DR updating [21]. Mode shapes were not included in the objective function, as they are typically weakly sensitive to the updating variables in DR applications. Additionally, frequency-based residuals provide more stable targets under working conditions [21].

5.3. Updating using the first three natural frequencies

The updating procedure employed the first three experimentally identified natural frequencies. Focusing on the lowest modes is consistent with structural health monitoring (SHM) practice, as these modes are typically identified more robustly under service conditions and are less affected by limited observability than higher-order modes. Limiting the number of calibration targets also helps mitigate ill-conditioning as the number of unknown updating parameters increases, stressing the importance of careful parameter choice and sensitivity screening [15, 20].

This choice is further supported by the characteristics of the available monitoring system. The BM span is instrumented primarily with electrical resistance strain gauges and thermocouples, and the sensing layout was designed for weigh-in-motion, transverse load distribution assessment, and the study of composite action, rather than for modal testing with dedicated vibration sensors [13]. As a result, the available measurements provide reliable information for identifying global frequency content, whereas experimental estimation of mode shapes is less robust due to the use of strain responses instead of direct velocity or acceleration measurements and poor sensor placement for spatial observability of modal shape patterns. Therefore, the updating strategy relies exclusively on the first three natural frequencies [15, 20].

5.3.1. DR coefficient construction and FE runs

Section 2 introduces the equation required to construct the surrogate for the FE model (Equation 5.2). The coefficients of the vector K_i, as defined in Equation 5.3 for (i=1,. . . ,3), must be determined.

$$\{f_i^{FEM}\} = [C]\{K_i\} \qquad i = 1, \ldots, 3 \tag{5.2}$$

with:

$$\{K_i\} = \begin{Bmatrix} C_i \\ A_{i,1} \\ B_{i,1} \\ \vdots \\ A_{i,N} \\ B_{i,N} \end{Bmatrix} \qquad i = 1, \ldots, 3 \tag{5.3}$$

$$[C] = \begin{bmatrix} 1 & x_1^B & x_1^{B2} & \cdots & x_N^B & x_N^{B2} \\ 1 & x_1^U & x_1^{U2} & \cdots & x_N^B & x_N^{B2} \\ 1 & x_1^L & x_1^{L2} & \cdots & x_N^B & x_N^{B2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_1^B & x_1^{B2} & \cdots & x_N^U & x_N^{U2} \\ 1 & x_1^B & x_1^{B2} & \cdots & x_N^L & x_N^{L2} \end{bmatrix} \qquad N = 1, \ldots, 6 \tag{5.4}$$

Since the matrix C is a constant matrix that depends only on the vector of the updating parameters X represented below in the equation 5.5

$$\{X\} = \begin{bmatrix} E_{Deck} & \rho_{deck} & E_{Barriers} & \rho_{Barriers} & \rho_{Asphalt} & L_{Span} \end{bmatrix} \qquad N = 6 \tag{5.5}$$

where N represents the number of updating nominal parameters.

Nominal values of the vector X and their corresponding lower and upper bounds were defined in Section 2. The variables are normalized with respect to the nominal value, which is the procedure used to construct the matrix C. The resulting matrix has dimensions $2N+1$ by $2N+1$. Table 5.2 presents the initial/nominal values of the updating parameters along with their lower and upper limits and Table 5.4 shows the normalized values of the initial parameters alongside their respective limits.

Table 5.2: Normalized values of the nominal, lower and upper updating values

Parameter	Initial/Nominal Values	Normalized Lower Value	Normalized Upper Values
E_{Deck} [MPa]	32627	0.5	1.5
ρ_{deck} [kg/m^3]	2.549E-09	0.6	1.4
$E_{Barriers}$ [MPa]	32627	0.5	1.5
$\rho_{Barriers}$ [kg/m^3]	2.549E-09	0.6	1.4
$\rho_{Asphalt}$ [kg/m^3]	2.3E-09	0.6	1.4
L_{Span} [m]	22.71	0.98	0.95

Table 5.3: Normalized values of the nominal, lower and upper updating values

Run	X_1 (E_{Deck})	X_2 (ρ_{deck})	X_3 ($E_{Barriers}$)	X_4 ($\rho_{Barriers}$)	X_5 ($\rho_{Asphalt}$)	X_6 (L_{Span})
1	1	1	1	1	1	1
2	0.5	1	1	1	1	1
3	1.5	1	1	1	1	1
4	1	0.6	1	1	1	1
5	1	1.4	1	1	1	1
6	1	1	0.5	1	1	1
7	1	1	1.5	1	1	1
8	1	1	1	0.6	1	1
9	1	1	1	1.4	1	1
10	1	1	1	1	0.6	1
11	1	1	1	1	1.4	1
12	1	1	1	1	1	0.98
13	1	1	1	1	1	0.95

Therefore, the number of FE modal analyses required to solve the DR method is also $2N+1$, as the DR method enforces Equation 5.6 at each sampling point:

$$\{f_i^{DR}(X)\} = \{f_i^{Exp}(X)\} \tag{5.6}$$

After obtaining the $2N+1$ FE modal analyses by varying each updating parameter at its lower and upper limits, the vector $\{f_i^{Exp}(X)\}$ can also be constructed, as shown in Table 5.4.

Table 5.4: Frequencies values of the $2N+1$ FE modal analysis by varying the updating parameters with respect to the limits to build the $\{f_i^{Exp}(X)\}$ vectors

Run	f_1 **[Hz]**	f_2 **[Hz]**	f_3 **[Hz]**
1	4.5824	6.8288	11.785
2	4.4915	6.5449	11.007
3	4.6315	7.0057	12.38
4	5.1511	7.5503	13.056
5	4.1676	6.2781	10.83
6	4.2538	6.2264	11.536
7	4.8243	7.2451	11.947
8	4.7057	7.2376	12.468
9	4.4653	6.4729	11.207
10	4.7997	7.0665	12.223
11	4.3917	6.6126	11.393
12	4.7531	7.061	11.899
13	5.0255	7.4419	12.128

After having defined the system of equation, the unknown coefficients of the DR model can be obtained as follows:

$$\{K_i\} = [C]^{-1}\{f_i^{FEM}\} \qquad i = 1, \ldots, M \tag{5.7}$$

After computing the 2N+1 unknown coefficients C_i, $A_{i,k}$, and $B_{i,k}$, the surrogate model can be used to minimize the objective function without repeatedly solving the full FE eigenvalue problem. The optimization was implemented in Python (see Appendix B.1) with the minimization of Equation 5.8:

$$J = \frac{1}{2}\sum_{i=1}^{M}\left(\frac{f_i^{num} - f_i^{exp}}{f_i^{exp}}\right)^2 \tag{5.8}$$

After obtaining the optimal parameter vector from the DR-based minimization, a final FE modal analysis is carried out using the updated parameters to verify the frequency match in the full FE model. This validation step is necessary because the DR formulation is a quadratic surrogate based on a finite set of FE evaluations. The optimum identified on the surrogate must be confirmed compared to the original eigenvalue problem to ensure

that the predicted improvement is not influenced by approximation inaccuracies and that the updated FE model is appropriate for further analyses [13, 15, 21].

5.4. Updated model results

In Table 5.5, the optimal updating values are reported in terms of their normalized values, their corresponding non-normalized (physical) values, and the percentage variation with respect to the initial (baseline) parameter values.

Table 5.5: Normalized values of the nominal, lower and upper updating values

Parameter	Initial Values	Normalized Optimal Values	Optimal Values	Variation
E_{Deck} [MPa]	32627	0.5	16313.5	50%
ρ_{deck} [kg/m^3]	2.549E-09	0.893	2.277E-09	10.7%
$E_{Barriers}$ [MPa]	32627	0.2	6525.4	80%
$\rho_{Barriers}$ [kg/m^3]	2.549E-09	1.7	4.333E-09	70%
$\rho_{Asphalt}$ [kg/m^3]	2.3E-09	0.35	8.05E-10	65%
L_{Span} [m]	22.71	0.993	22.552	0.7%

To quantify the agreement between the numerical and experimental natural frequencies, the relative error of the i_{th} Mode is computed as 5.9

$$e_i[\%] = 100\frac{\left|f_i^{FEM} - f_i^{exp}\right|}{f_i^{exp}} \tag{5.9}$$

Since comparing the modes one by one does not yield a single overall accuracy measure, the root-mean-square (RMS) percentage error (Eq.5.10) is also evaluated. Because it penalizes larger mismatches more strongly, it provides a compact indicator of the agreement between numerical and experimental frequencies over the n modes considered.

$$\text{RMS} = \sqrt{\frac{1}{n}\sum e_i^2} \tag{5.10}$$

The results reported in Table 5.6 are obtained by running a FE modal analysis using the optimal updating values identified through the DR method (Table 5.5). The percentage errors are also reported together with the RMS value defined in Eq. 5.10

Since the FE modal analysis performed with the optimal updating values still shows a

Table 5.6: Comparison of Modal frequencies: Abaqus FE1 model and OMA analysis

No.	**OMA [Hz]**	**Initial FE [Hz]**	**Error [%]**	**Optimal FE [Hz]**	**Error [%]**
1^{st}	4.64	4.58	1.3	4.09	11.09
2^{nd}	5.23	6.83	30.6	5.24	0.2
3^{rd}	10.64	11.79	10.8	10.5	1.3
		RMS=	18.8	**RMS=**	6.4

residual mismatch with respect to the experimental frequencies, an additional modeling consideration was introduced during the development of this thesis. Specifically, the span length was slightly reduced in order to improve the agreement of the *first* natural frequency, without attending to the matching of the second and third frequencies at this stage. This update was then followed by a new model updating run to assess whether the modified geometry leads to an overall improved representation of the bridge dynamic characteristics.

The BM span length was shortened by $15cm$, i.e., from a clear span of $22.71m$ to $22.56m$. As a result, the first numerical frequency increased from $4.58Hz$ to $4.638Hz$.

For the subsequent DR-based updating, the coefficient matrix C of the DR formulation was kept unchanged. Therefore, the coefficient matrix is still the one reported in Table 5.4. However, since the DR coefficients are identified using the FE frequencies evaluated at the $2N + 1$ sampling points, the corresponding frequency vectors obtained from the 13 FE runs were recomputed for the shortened-span configuration. These updated FE frequency values are reported in Table 5.7.

Table 5.7: Frequencies values of the $2N+1$ FE modal analysis by varying the updating parameters with respect to the limits to build the $\{f_i^{Exp}(X)\}$ vectors of the second model

Run	f_1 **[Hz]**	f_2 **[Hz]**	f_3 **[Hz]**
1	4.6381	6.9081	11.8230
2	4.4915	6.5449	11.007
3	4.6315	7.0057	12.38
4	5.1511	7.5503	13.056
5	4.1676	6.2781	10.83
6	4.2538	6.2264	11.536
7	4.8243	7.2451	11.947
8	4.7057	7.2376	12.468
9	4.4653	6.4729	11.207
10	4.7997	7.0665	12.223
11	4.3917	6.6126	11.393
12	4.752	7.0596	11.898
13	5.0225	7.439	12.126

After computing all parameters required for the DR method, the optimal updating values were determined following the procedure described in previous sections. Table 5.8 presents the optimal updating values obtained after introducing the shortened-span configuration for the BM span. In this second updating run, the parameter limits were revised by widening the admissible ranges to allow greater flexibility and avoid excessively restrictive bounds in the optimization process.

Table 5.8: Normalized values of the nominal, lower and upper updating values

Parameter	**Initial Values**	**Normalized Optimal Values**	**Optimal Values**	**Variation**
E_{Deck} [MPa]	32627	0.5	16313.5	50%
ρ_{deck} [kg/m^3]	2.549E-09	0.885	2.258E-09	11.5%
$E_{Barriers}$ [MPa]	32627	0.2	6525.4	80%
$\rho_{Barriers}$ [kg/m^3]	2.549E-09	1.7	4.333E-09	70%
$\rho_{Asphalt}$ [kg/m^3]	2.3E-09	0.35	8.05E-10	65%
L_{Span} [m]	22.71	0.9398	21.202	6%

The results reported in Table 5.9 are obtained by running a FE modal analysis using the

optimal updating values identified through the DR method (Table 5.8).

Table 5.9: Comparison of Modal frequencies: Abaqus FE2 model and OMA analysis

No.	OMA [Hz]	Initial FE [Hz]	Error [%]	Optimal FE [Hz]	Error [%]
1st	4.64	4.638	0.04	4.587	1.1
2nd	5.23	6.908	32.1	5.848	11.8
3rd	10.64	11.823	11.1	10.904	2.5
		RMS=	19.6	**RMS=**	7

A direct comparison is required to determine which modeling option best represents the dynamic characteristics of the actual BM span bridge. Table 5.10 summarizes the two modeling options and their respective errors.

Table 5.10: Comparison of Modal frequencies: OMA analysis with respect to the two FE Abaqus Modal Analysis

No.	OMA [Hz]	Abaqus FE1 [Hz]	Error FE1 [%]	Abaqus FE2 [Hz]	Error FE2 [%]
1st	4.64	4.09	11.09	4.587	1.1
2nd	5.23	5.24	0.2	5.848	11.8
3rd	10.64	10.5	1.3	10.904	2.5

Table 5.10 demonstrates that the two updated Abaqus models show a different level of agreement with respect to the experimentally identified modal frequencies. Abaqus FE1 closely reproduces the second and third frequencies, with errors of 0.2% and 1.3%, respectively, but underestimates the first frequency by approximately 11.1%. In contrast, Abaqus FE2 reliably predicts the first frequency (1.1% error) and maintains a moderate error for the third frequency (2.5%), although it overestimates the second frequency by 11.8%.

Because the subsequent analyses address coupled Vehicle–Bridge Interaction and the reproduction of strain time histories under moving loads, an accurate representation of the fundamental (first) mode is given priority, as it is expected to govern the global dynamic response. Accordingly, Abaqus FE2 is selected as the reference updated model for the VBI simulations. The suitability of this selection will be further evaluated by comparing numerically predicted strain responses in conjunction with measured strain records from the benchmark dataset.

6 | Moving Load and Vehicle-Bridge Interaction

6.1. Introduction to moving load problems in bridges

Traffic loading constitutes a primary source of service-level demand in highway bridges and is a major contributor to operational vibration. In the simplest modeling approach, traffic is represented as a sequence of moving forces applied at specified locations along the deck. This approximation is often adequate for quasi-static effects; however, it neglects the inertia and suspension dynamics of the vehicle. Moreover, that approximation does not allow capturing interaction phenomena such as contact-force modulation, dynamic amplification near resonance, or intermittent loss of contact [14, 22].

A more general formulation yields the vehicle–bridge interaction (VBI) problem, in which the bridge and the vehicle are modeled as two dynamically coupled subsystems. They exchange forces through the wheel–deck interface. Unlike spatially fixed loads, the excitation in a vehicle–bridge interaction (VBI) problem travels along the span with the vehicle. As the wheels move, the wheel–deck interface may also evolve in time, with phases of contact, possible loss of contact, and reattachment. From a numerical standpoint, this leads to an inherently nonlinear problem because the analysis must account for large sliding contact and changing constraints, even when the bridge itself is modeled with linear elastic material behavior [2, 14, 22].

For the SCSHM benchmark BM span, the monitoring dataset acquired during in-service truck passages supports extending the calibrated bridge model to traffic-induced dynamic analyses. The benchmark study points out the importance of bridge geometry, composite deck–girder behavior, and realistic traffic loading for accurate response interpretation [13]. Building on the updated finite element (FE) model developed in Section 5, this chapter introduces the adopted vehicle–bridge interaction (VBI) framework in Abaqus and documents the assumptions used to generate an initial interaction analysis model.

6.2. Review of vehicle–bridge interaction modeling procedures

VBI modeling strategies can be broadly grouped into three categories [2, 14, 22]:

- Analytical and semi-analytical formulations.
- Mixed approaches (co-simulation / matrix export).
- Fully FE-based coupled modeling within a commercial package.

In classical VBI studies, the bridge is typically modeled adopting beam and plate idealizations. The moving vehicle by means of time-variant coupling terms (for instance: Dirac-delta forcing and moving-coordinate descriptions) that rely on analytical or semi-analytical formulations. While computationally effective, such formulations usually require careful treatment of convective terms, time-dependent operators and potential symmetry breaking in the governing matrices. These features make implementation more difficult for complex bridges and contact cases [22].

A common alternative is the use of mixed approaches (co-simulation or matrix export), in which the bridge finite element (FE) model is constructed in a commercial solver and system matrices are exported to an external code where vehicle equations are integrated and contact forces are assembled. This strategy can improve modeling efficiency for complex bridges, but becomes cumbersome when nonlinearities are present or when repeated data exchange is required [11, 14, 16, 17, 23].

However, recent work has shown that building both subsystems in one finite element (FE) space, "fully FE based coupled modeling" within a commercial package, can be significantly simpler to implement for complex structures while taking full advantage of powerful contact algorithms and built-in integration procedures. In [22], Saleeb and Kumar present a framework in which the main structure and sliding subsystem are represented as deformable bodies undergoing large relative sliding coupled through contact capabilities provided by Abaqus. This process removes a large number of non-standard analytical terms by preserving a purely Lagrangian treatment, allowing for separation and reattachment, as well as direct access to stresses and interaction forces in the output database.

Along the same line, Lu et al. present a unified Abaqus-based framework for dynamic VBI analysis, demonstrating that a single FE package can decrease exchange of data that can be redundant and can incorporate nonlinear materials and nonlinear dynamics if needed [14]. Abuodeh and Redmond in [2] focus on Abaqus-only VBI models and show

that quarter-car and half-car idealizations can be coupled to a bridge through node-to-surface contact with stable enforcement of contact pressures that yield in a method that is computational efficiency.

Given the objective of extending the updated BM span model to vehicle–bridge interaction (VBI) simulations in Abaqus, see the step by step modeling procedure in Appendix C, this work adopts the fully FE-based strategy, utilizing Abaqus contact and implicit dynamic analysis to represent the coupled system [2, 14, 22].

6.3. Numerical Modeling in Abaqus

This section outlines the Abaqus modeling framework developed to simulate vehicle–bridge interaction for the BM span. The following subsections introduce the simplified sprung-mass vehicle model, summarize the updated bridge model selected from Chapter 5, and detail the node-to-surface contact formulation used to enforce wheel–deck interaction.

6.3.1. Vehicle Model Description

The vehicle subsystem is modeled as a simplified sprung-mass system (see Figure 6.1), representing the dominant vertical interaction effects and enabling computational efficiency for repeated simulations. Consistent with standard VBI practice, the vehicle is represented by lumped masses connected by spring and dashpot elements, resulting in a reduced number of degrees of freedom compared to a full multibody vehicle model [2].

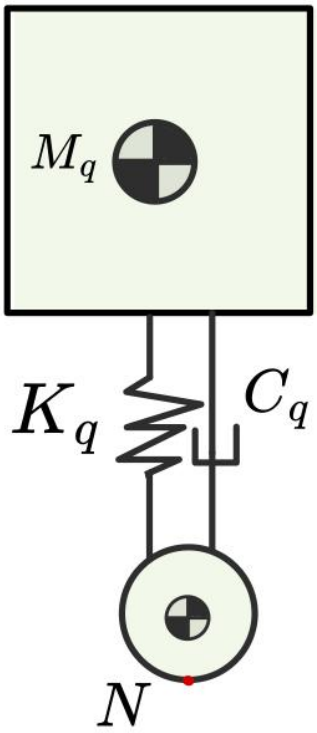


Figure 6.1: A schematic of a sprung-mass system

where M_q, K_q, C_q and C are the mass, stiffness, damping, and contact node of the vehicle.

In Abaqus implementations, this class of model is usually constructed by defining rigid or lumped-mass reference points (RPs) and connecting them with discrete spring and dashpot elements. Abuodeh and Redmond [2] show that such a model can be coupled to a bridge via node-to-surface contact, while keeping a stable procedure for first bringing the vehicle to static equilibrium due to gravity, followed by traversal in an implicit dynamic step. This staged approach is adopted to avoid convergence issues that often arise when gravity, damping, and contact are activated simultaneously.

For this study, the vehicle speed and primary parameters associated with the truck crossing the BM span are summarized below. Figure 6.2 presents the available test cases that include vehicle position and truck configuration. Test No. 5 (highlighted in Figure 6.2) was adopted for the analysis. The corresponding vehicle properties, such as the crossing speed, used in the numerical model are listed in Table 6.1. The truck corresponding to test 5 and its total weight are shown in Figure 6.3

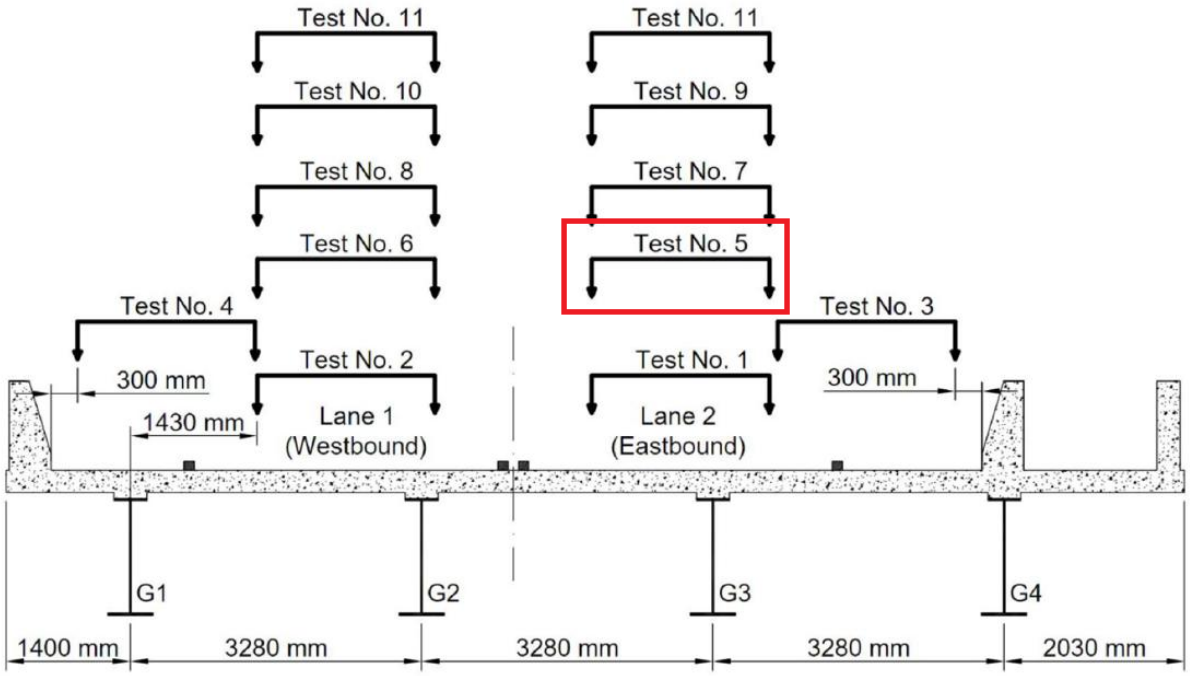


Figure 6.2: Location and Direction of the modeled truck. Image taken from [13]

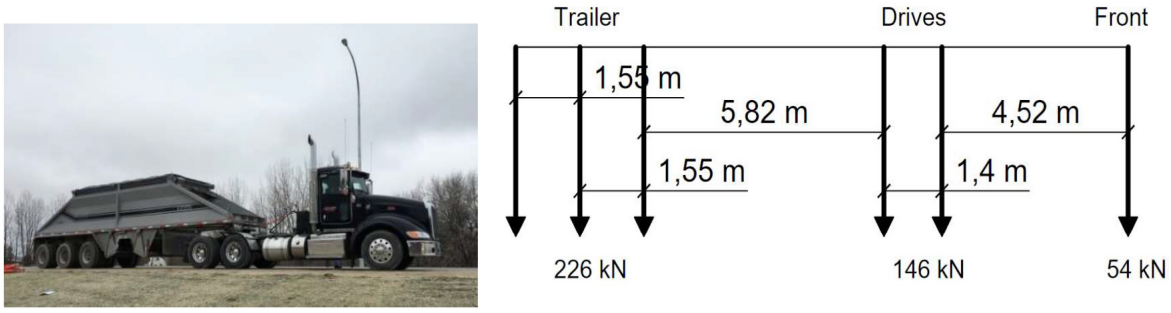


Figure 6.3: Truck, Axle weights, and Spacing of the axles. Image taken from [13]

Table 6.1: Characteristics of the vehicle modeled in Abaqus. Data taken from [13]

Test No.	Direction	Approximate Speed [km/h]	Total Weight [kN]	Spring Stiffness [N/mm]	Damping Coeff. [N.s/mm]
5	Eastbound	45	426	3300	175

6.3.2. Bridge model used for interaction analysis

The bridge subsystem adopted for VBI corresponds to the updated Abaqus FE2 model developed in section 5. Table 5.10 shows the modal properties of the chosen model. The model geometry and component layout follow the benchmark description of the BM span, which is a simply supported composite steel–concrete superstructure carrying two lanes of traffic and consisting of four longitudinal steel girders supporting a reinforced concrete deck and barrier walls [13].

For the interaction analyses in this chapter, the updated bridge model is used without further modification of material properties. The modal properties of the Abaqus FE2 are shown in Table 5.10 and the Model with the Sprung vehicle is presented in Figure 6.4.

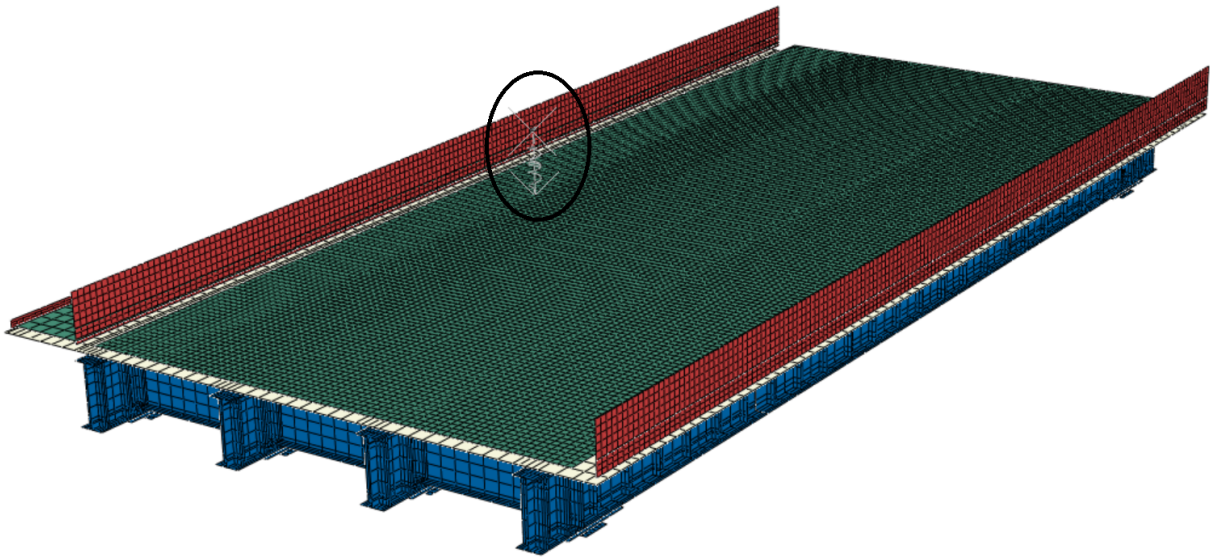


Figure 6.4: Abaques FE2 model and sprung-mass vehicle of the BM span

6.3.3. Node-to-surface contact formulation

In Abaqus-based VBI, wheel–deck interaction is commonly implemented using either node-to-surface or surface-to-surface contact definitions. Both require automatic contact detection during large sliding and robust constraint enforcement to maintain physically meaningful normal contact forces [2, 14]. In the present study, a node-to-surface formulation is adopted, where the wheel/tire node acts as the slave and the bridge running surface acts as the master surface. This configuration enables the wheel node to travel over the bridge surface while Abaqus enforces the normal contact constraint through a selected pressure–overclosure relationship [2].

In the standard node-to-surface contact formulation, the contact constraints are defined

so that each slave node on one side of the interface interacts with a corresponding projection point on the master surface on the opposite side (see Figure 6.5). Each constraint therefore links a single slave node to a set of nearby master nodes whose interpolated values determine the response at the projection point [5].

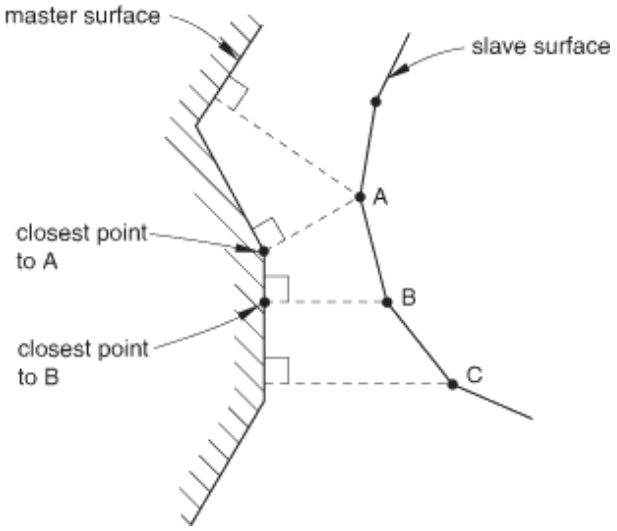


Figure 6.5: Node-to-surface contact discretization. Image taken from [5]

The bridge deck surface is defined as the master surface because it provides a continuous contact target along the travel path. The moving wheel node is treated as the slave entity. This choice is consistent with standard Abaqus contact practice for problems dominated by a moving discrete contact point [2, 22].

in [2], Abuodeh and Redmond report that using a "hard" pressure–overclosure relationship is effective in minimizing penetration and reducing numerical chatter throughout dynamic contact. The same baseline choice is adopted here. In hard contact, compressive pressure is transmitted when contact closes, while tensile transfer across the interface is not allowed, which is consistent with the unilateral nature of wheel–deck normal interaction [22].

To focus on vertical interaction and to avoid additional nonlinearities, tangential friction is neglected in the baseline model. This assumption is consistent with typical validation-focused VBI studies where the objective is to test normal contact enforcement and coupled vertical response [14]. The use of finite sliding is essential, since the wheel contact location undergoes large relative motion along the deck surface [22].

6.3.4. Moving load implementation strategy

The vehicle motion is imposed through a prescribed horizontal displacement (or velocity) of the vehicle reference point(s), such that the wheel node traverses the bridge surface over a time window consistent with the chosen speed. This approach corresponds to the Abaqus-based strategy in which the vehicle and bridge are modeled independently and then coupled through contact, while the vehicle is "pushed" across the bridge within a time period corresponding to the target velocity [2]. Since contact activation and traversal can introduce transient effects and convergence sensitivity, the simulation was performed using a staged procedure to establish a stable initial condition before the crossing analysis.

The analysis was therefore carried out in three consecutive steps:

- **Static (bridge self-weight):** a static analysis was first performed to apply gravity to the bridge model only, accounting for the self-weight of the structure.
- **Implicit dynamic (vehicle gravitational settlement):** an implicit dynamic step was then introduced to apply the vehicle mass under gravity and allow the system to reach equilibrium. The step duration was selected long enough for the transient effects associated with the sudden activation of the vehicle weight to decay, so that the response is not affected by an abrupt application of the self weight.
- **Implicit dynamic (vehicle crossing):** finally, a second implicit dynamic step was performed to simulate the vehicle crossing the BM span at a constant speed of 45 km/h, with the wheel–deck interaction enforced through the adopted node-to-surface contact formulation.

Because the contact condition may evolve within a time increment, the traversal analysis is sensitive to time increment size, numerical damping, and convergence tolerances. For this reason, conservative time-increment limits were adopted to ensure stable contact enforcement and smooth response histories during the vehicle passage [14].

6.4. Dynamic Interactions Results

This subsection presents the main results obtained from the Abaqus vehicle–bridge interaction simulations described in Section 6.3. The aim is to test the dynamic response of the BM span model subjected to a vehicle that is crossing at the selected speed, and to verify that the coupled implementation produces stable and physically consistent response histories with the assumptions made (vehicle model, contact formulation, and traversal strategy).

The discussion focuses on time history quantities extracted from the analysis, including the vertical displacement histories of:

- the wheel reference point (contact node)
- the sprung-mass reference point (Body of the truck)
- a selected bridge response location within the monitored span.

The results are interpreted mainly in terms of numerical consistency and response trends, rather than as a full coupled validation including vehicle-side dynamics.

To illustrate the coupled response during the vehicle passage, the vertical displacement time histories extracted from the interaction analysis are reported in Figures 6.6, 6.7, 6.8, 6.9, and 6.10. The selected outputs correspond to:

- the sprung-mass reference point: which captures the dominant vertical response of the vehicle body, see Figure 6.6.
- the wheel/contact reference point: which represents the motion of the contact node enforcing wheel–deck interaction, see Figure 6.7.
- the suspension deflection: This tells how much the spring is compressing/expanding while the truck moves, see Figure 6.9.
- a representative bridge point located at the mid-span section: used to characterize the structural response under the moving vehicle, see Figure 6.10.

In the following plots, downward displacements are reported as negative values. The time axis spans the vehicle traversal interval at 45 km/h, from the instant the wheel enters the bridge to the instant it exits the span.

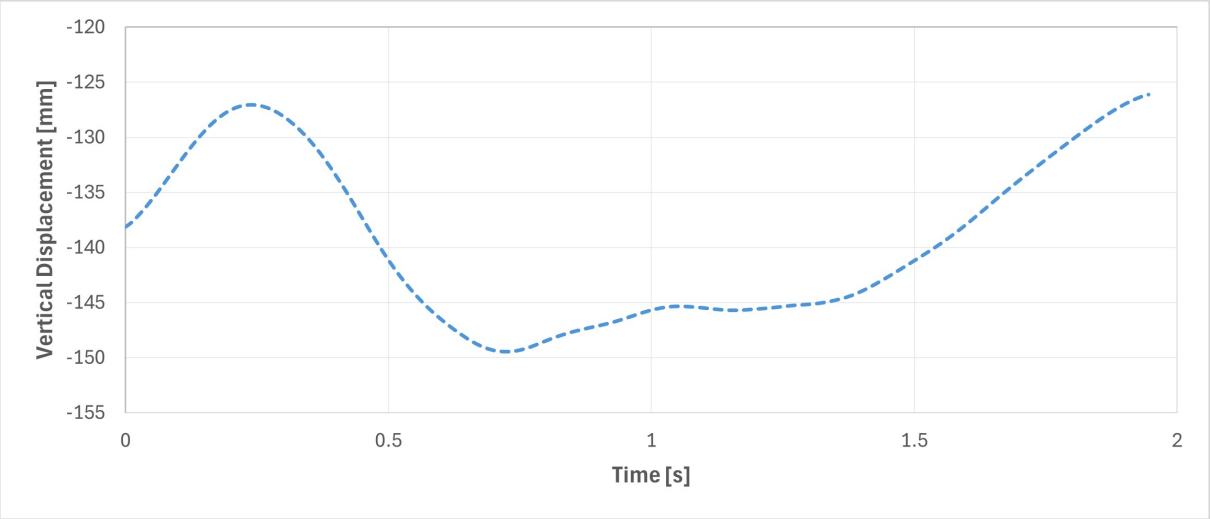


Figure 6.6: Time-history of the displacement of the sprung-mass reference point

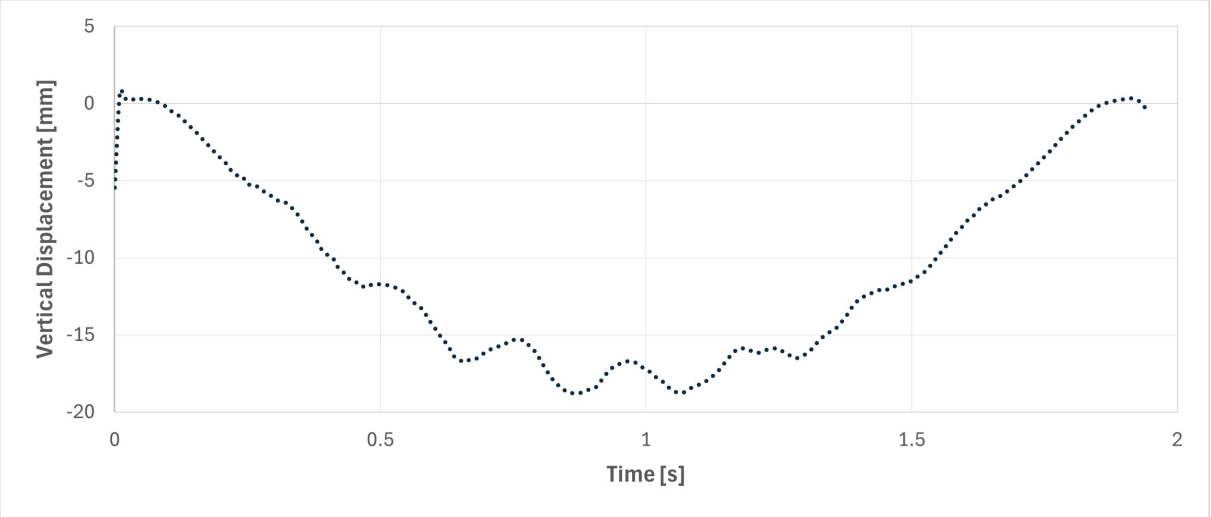


Figure 6.7: Time-history of the displacement of the wheel reference point

Here, "suspension deflection" refers to the relative vertical displacement between the sprung mass and the wheel ($(U_{Sprung}$ -(U_{Wheel}).

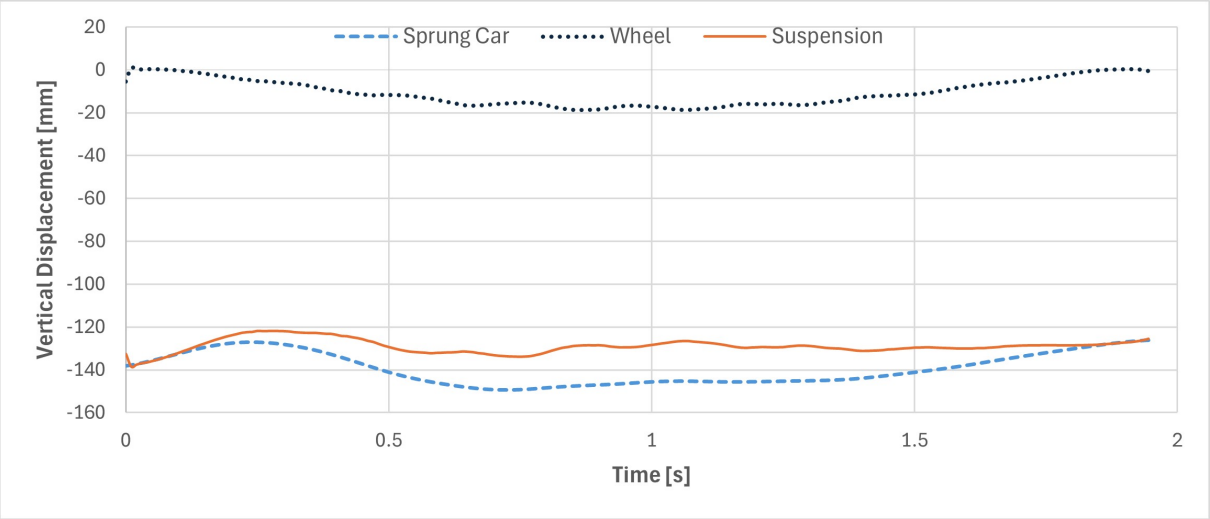


Figure 6.8: Time-history of the vertical displacement of the sprung mass, wheel mass, and corresponding suspension deflection

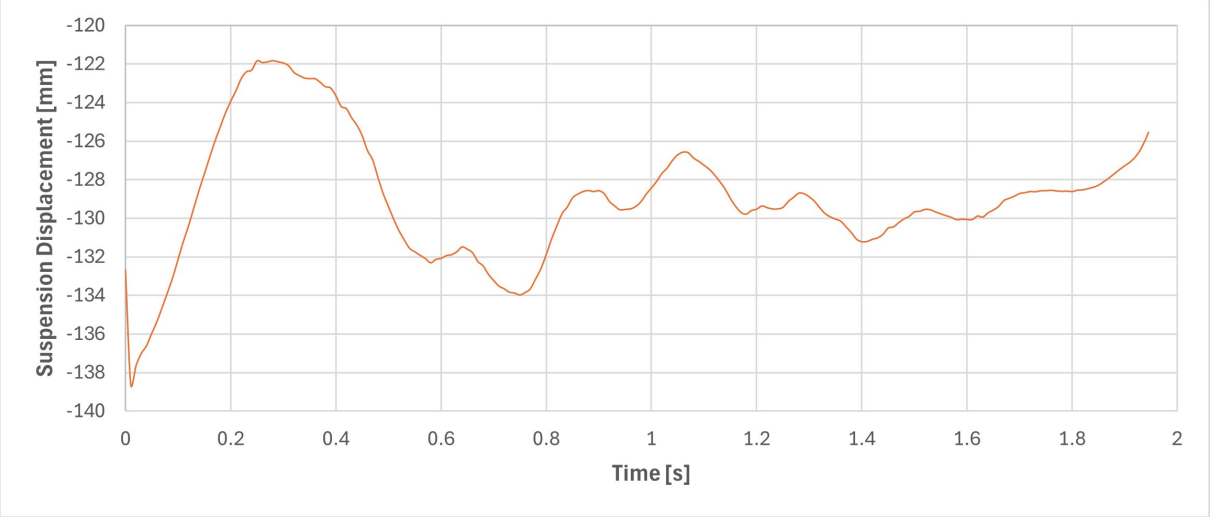


Figure 6.9: Time-history of the displacement of the suspension deflection

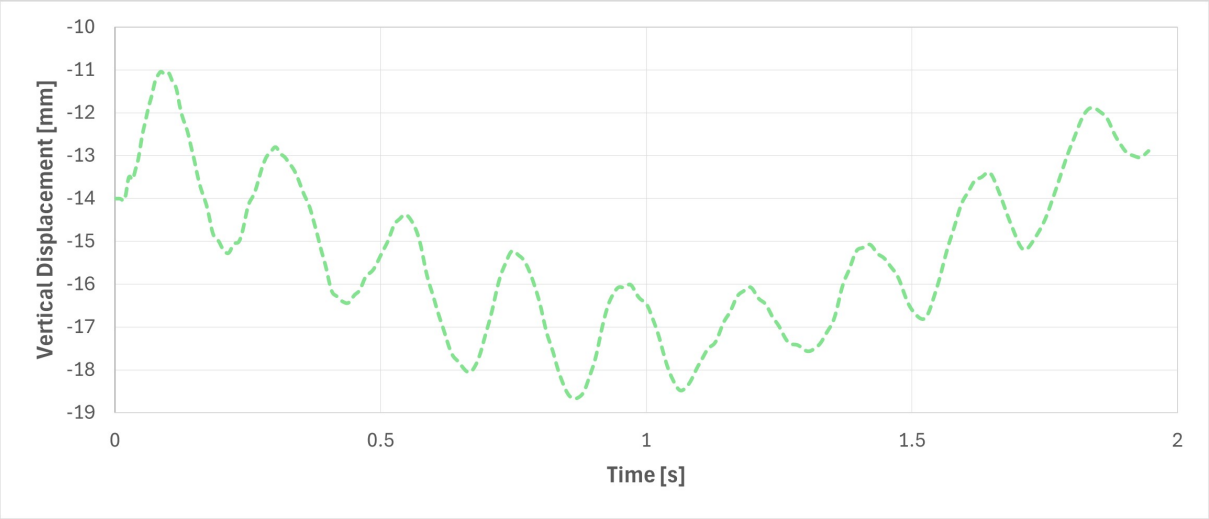


Figure 6.10: Time-history of the displacement at mid-span of the outer I-girder section

Overall, the displacement histories indicate a reliable interaction response during the vehicle crossing. The wheel and bridge responses show consistent trends, the maximum downward response occurring near mid-crossing, followed by recovery as the vehicle exits the span. The sprung-mass response is smoother and dominated by low-frequency motion, which is expected for the vehicle body. The bridge mid-span displacement exhibits additional oscillatory content compared to the reference-point outputs, exhibiting the ability to capture local structural vibrations and numerical sensitivity. Moreover, no abrupt jumps are observed in the histories, showing stable contact enforcement throughout the traversal.

6.5. Limitations of the current study

The present VBI model is intended as a first coupled implementation based on the updated bridge model. Several limitations have to therefore be stated clearly.

The vehicle is modeled as a reduced-DOF sprung-mass system and does not capture full multibody dynamics, suspension asymmetry, or detailed axle modeling. Even though the sprung-mass is consistent with efficient Abaqus-only VBI frameworks, it limits realism for heavy multi-axle trucks and may affect predicted interaction forces and bridge response [2].

During the development of the ongoing thesis, simulations were performed taking into account that the surface is smooth and has no irregularities; however, surface irregularities in practice can increase dynamic amplifications and broaden response bandwidth. Espe-

cially when vehicle and bridge frequencies are close [2, 14]. Roughness was not considered with the aim to reduce modeling complexity and/or computational time. To perform the implementation of surface irregularities see [24].

The bridge is treated as linear elastic and does not account for material nonlinear behavior, localized damage, or nonlinear connection behavior. While Abaqus-based frameworks are able to include nonlinearities when required, the present work focuses on developing a stable coupled workflow [14].

Although the bridge model was calibrated to modal information (section 5 and is consistent with the benchmark structural description [13], the present VBI implementation has not been validated against coupled measurements of vehicle-side dynamics (wheel contact forces or suspension response). The benchmark dataset primarily provides bridge-side strain time histories that do not directly enable verification of the simulated interaction forces for the vehicle-bridge interaction.

7 | Conclusions and Future Work

7.1. Conclusions

This thesis covers the operational strain-measurement analysis, the development of a calibrated finite element (FE) model of an in-service bridge span, and the implementation of vehicle–bridge interaction in a commercial FE package Abaqus. The study was carried out in the BM span of the SCSHM in-service monitoring benchmark [13] and emphasizes practical considerations influencing the reliability of vibration-based calibration and subsequent traffic simulations.

The Operational Modal Analysis (OMA) that was performed within the `pyOMA` environment enabled the identification of three natural frequencies of the BM span using output-only strain data. The frequencies obtained in the section 3 (4.64 Hz, 5.23 Hz, and 10.64 Hz) allowed to first confirm the previously reported benchmark frequency of 4.65 Hz and second to introduce two additional values, which permit expanding the modal reference set for calibration. The corresponding mode shapes were interpreted as first vertical bending, first torsion, and a plate mode.

A FE model of the bridge was developed in Abaqus using shell elements to represent the deck, transverse trusses, and longitudinal/transversal steel sections. The development of the model points out the significant influence of boundary conditions, connectivity assumptions, and mass representation on the predicted modal properties. To better understand the FE modeling, two Abaqus variants were examined to assess the impact of alternative support idealizations and modeling details on the dynamic response. The first model did not include the bearing devices of the BM span bridge, whereas the second model included them using rigid body discretization.

Model calibration was performed using the Douglas-Reid (DR) model updating method, which employed a frequency-based objective function and excluded mode shapes as explained in section 5. Six physically meaningful updating parameters were considered, including stiffness and mass quantities relations (deck and barrier properties, asphalt mass) and the span length parameter. The sensitivity analysis showed that the span length and

the mass density of the deck have a dominant influence on the modal response; however, the remaining parameters (the young modulus of the deck/barriers and mass density of the barriers/asphalt) also play a role in the model updating, see 5.1.

The DR method enabled the improvement of the overall agreement between experimental and numerical frequencies with respect to the initial FE models in Abaqus without running the updating process. For the Abaqus FE1 model, the update reduced the root mean square (RMS) frequency mismatch but revealed that improving higher frequencies could worsen the matching with the first natural frequency. To counteract the mismatch in the first frequency, a second Abaqus FE2 model incorporated a length reduction of the span to capture the first natural frequency before the DR method. The latter model configuration led to a better agreement for the first frequency and reduced differences over the first three target frequencies. Therefore, small geometric idealizations at end regions and span-length definition can significantly influence low-order dynamics and should be explicitly considered during model updating.

The updated bridge model was subsequently integrated into a vehicle-bridge interaction (VBI) simulation in Abaqus, where the vehicle model is a simplified sprung-mass vehicle coupled via node-to-surface contact. A stepped analysis procedure, comprising bridge gravity, vehicle gravitational settlement, and implicit dynamic interaction, was implemented to stabilize contact and prevent numerical instabilities. The time histories outputs of vertical displacement for the wheel contact node, sprung mass (vehicle body) point, and bridge mid-span node showed consistent results during the passage of the vehicle through the span, with smooth responses and no abrupt jumps that indicated stable contact enforcement under the adopted assumptions. Even though this VBI analysis does not constitute full validation against coupled measurements, it establishes a comprehensive Abaqus procedure that can be extended to benchmark simulations.

To sum up, this thesis shows that frequency-based model updating using OMA and DR methods is an effective approach for calibrating an Abaqus FE model of an in-service bridge when the available monitoring data primarily enable identification of natural frequencies. The results allows to draw the conclusion that the calibrated model can be extended to moving-vehicle simulations using fully FE-based modeling within a commercial package, such as Abaqus, providing a baseline for future developments in validation and time-domain calibration using measured events of traffic.

7.2. Recommendations for Future Work

Advancing the current Abaqus-based vehicle-bridge interaction (VBI) model toward a validated and benchmark-ready simulation tool requires the following developments.

Incorporate realistic road roughness into the simulation framework that is a practical improvement adopting standardized roughness definitions [1] and efficient equivalent-force formulations that avoid excessive mesh enhancement [24]. This would allow sensitivity studies on speed, roughness level, and vehicle properties while keeping the workflow fully in Abaqus.

Given that the benchmark dataset primarily features truck traffic and axle-group effects are important for interpreting responses [13], future models should include a multi-axle sprung-mass procedure, namely half-car or bicycle models, using Abaqus-based strategies that preserve computational efficiency [2].

The benchmark dataset includes strain measurements at multiple sections and lanes, facilitating direct validations of simulated strain time histories and measured events. Representative truck passages and estimated speeds or axle spacings should be selected based on the benchmark processing chain [13]. This validation is essential prior to employing the VBI model for further inference or scenario analyses.

Once a validated coupled model is established, VBI simulations should be used to compare measured and simulated strains for known or identified truck passages. This enables a second-stage calibration that targets time-domain response quantities, in addition to modal parameters, thereby expanding the approach to include model updating based on moving-load response.

Bibliography

[1] Mechanical vibration — road surface profiles — reporting of measured data, 2016.

[2] O. R. Abuodeh and L. Redmond. A framework for developing efficient vehicle-bridge interaction models within a commercial finite element software. In H. Y. Noh, M. Whelan, and P. S. Harvey, editors, *Dynamics of Civil Structures, Volume 2*, pages 67–73, Cham, 2023. Springer International Publishing.

[3] C. Baggio, V. Sabbatini, S. Santini, and C. Sebastiani. Comparison of different finite element model updates based on experimental onsite testing: the case study of san giovanni in macerata. *Journal of Civil Structural Health Monitoring*, 11:767–790, 2021. doi: 10.1007/s13349-021-00480-1.

[4] California Department of Transportation. *Bridge Design Practice*. Caltrans, Sacramento, CA, USA, 2022. URL `https://dot.ca.gov/programs/engineering-services/manuals/bridge-design-practice`. Accessed 2026-02-06.

[5] Dassault Systèmes. *Abaqus User's Manual*, 2006. URL `https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm`. Version 6.6.

[6] Z. Deng, M. Huang, N. Wan, and J. Zhang. The current development of structural health monitoring for bridges: A review. *Buildings*, 13(6), 2023. ISSN 2075-5309. doi: 10.3390/buildings13061360.

[7] Devriendt, D. Sitter, Vanlanduit, and Guillaume. Operational modal analysis in the presence of harmonic excitations by the use of transmissibility measurements. *Mechanical Systems and Signal Processing*, 23(3):621–635, 2009. ISSN 0888-3270. doi: 10.1016/j.ymssp.2008.07.009.

[8] B. Douglas and W. Reid. Dynamic tests and system identification of bridges. *Journal of the Structural Division*, 108:2295 – 2312, 1982. doi: 10.1061/JSDEAG.0006057.

[9] D. Ewins. *Modal Testing: Theory, Practice and Application*. Mechanical Engineering

Research Studies: Engineering Dynamics Series. Wiley, 2009. ISBN 9780863802188. URL https://books.google.it/books?id=09_zDwAAQBAJ.

[10] C. Gentile, Saisi, and A. Cabboi. Structural identification of a masonry tower based on operational modal analysis. *International Journal of Architectural Heritage*, 9, 02 2015. doi: 10.1080/15583058.2014.951792.

[11] X. H and N. Z. Dynamic interaction of vehicles and structures. *Science Compress, Beijing*, 2005.

[12] H. Herlufsen, P. Andersen, S. Gade, and N. Møller. Identification techniques for operational modal analysis – an overview and practical experiences. *Proceedings of the 1st International Operational Modal Analysis Conference, IOMAC 2005*, 2005.

[13] M. P. Limongelli, D. Thomson, S. Alampalli, and et al. SCSHM benchmark study on bridge in-service structural monitoring. *J Civil Struct Health Monit 15*, pages 849–863, 2025. doi: 10.1007/s13349-024-00846-1.

[14] X. Lu, C.-W. Kim, and K.-C. Chang. Finite element analysis framework for dynamic vehicle-bridge interaction system based on abaqus. *International Journal of Structural Stability and Dynamics*, 20(03):2050034, 2020. doi: 10.1142/S0219455420500340.

[15] J. Mottershead and M. Friswell. Model updating in structural dynamics: A survey. *Journal of Sound and Vibration - J SOUND VIB*, 167:347–375, 10 1993. doi: 10.1006/jsvi.1993.1340.

[16] S. Neves, A. Azevedo, and R. Calçada. A direct method for analyzing the vertical vehicle–structure interaction. *Engineering Structures*, 34:414–420, 2012. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2011.10.010. URL https://www.sciencedirect.com/science/article/pii/S014102961100407X.

[17] T. S. Paraskeva, E. G. Dimitrakopoulos, and Q. Zeng. Dynamic vehicle–bridge interaction under simultaneous vertical earthquake excitation. *Bulletin of Earthquake Engineering*, 15:71–95, 2017. doi: 10.1007/s10518-016-9954-z.

[18] D. P. Pasca, A. Aloisio, and M. M. R. et al. PyOMA and PyOMA_GUI: A python module and software for operational modal analysis. *SoftwareX*, 20, 2022. doi: 10.1016/j.softx.2022.101216.

[19] F. Peña, P. B. Lourenço, N. Mendes, and D. V. Oliveira. Numerical models for the seismic assessment of an old masonry tower. *Engineering Structures*, 32(5):1466–1478, 2010. ISSN 0141-0296. doi: 10.1016/j.engstruct.2010.01.027.

[20] C. Rainieri and G. Fabbrocino. *Operational Modal Analysis of Civil Engineering Structures*. Springer, New York, 1 edition, 2014. doi: 10.1007/978-1-4939-0767-0. URL `https://link.springer.com/book/10.1007/978-1-4939-0767-0`. Accessed 2026-02-06.

[21] I. Rosati, G. Fabbrocino, and C. Rainieri. A discussion about the Douglas-Reid model updating method and its prospective application to continuous vibration-based SHM of a historical building. *Engineering Structures*, 273, 2022. ISSN 0141-0296. doi: 10.1016/j.engstruct.2022.115058.

[22] A. F. Saleeb and A. Kumar. Automated finite element analysis of complex dynamics of primary system traversed by oscillatory subsystem. *International Journal for Computational Methods in Engineering Science and Mechanics*, 12(4):184–202, 2011. doi: 10.1080/15502287.2011.580830.

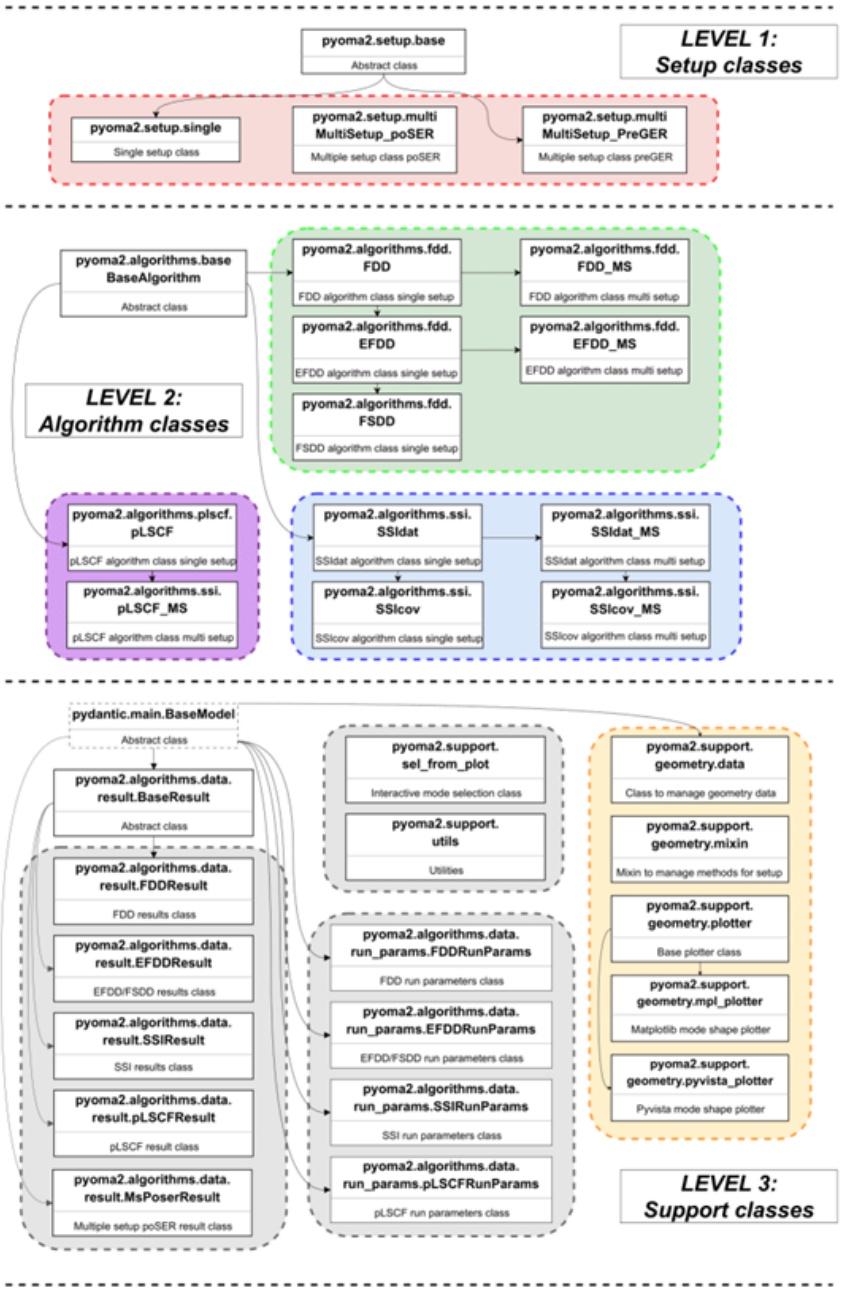
[23] D. Su, Y. Fujino, T. Nagayama, Y. Jaime Hernandez, and M. Seki. Vibration of reinforced concrete viaducts under high-speed train passage: Measurement and prediction including train-viaduct interaction. *Structure and Infrastructure Engineering*, 6:621–633, 10 2010. doi: 10.1080/15732470903068888.

[24] Y. Zhang, H. Zhao, and S. Lie. A simple approach for simulating the road surface roughness involved in vehicle-bridge interaction systems. *International Journal of Structural Stability and Dynamics*, 18:1871009, 01 2018. doi: 10.1142/S0219455418710098.

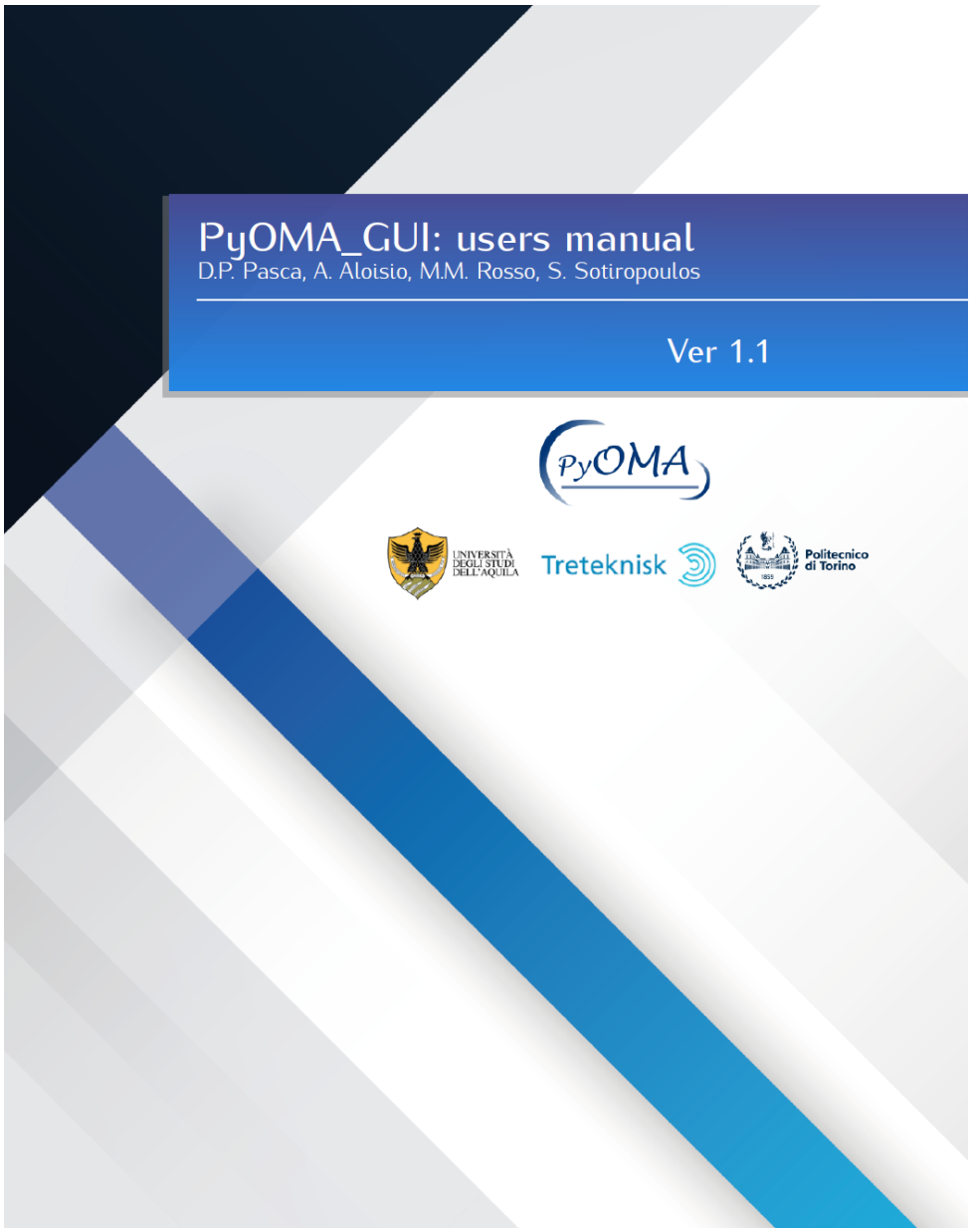
[25] Álvaro Bautista-De Castro, L. J. Sánchez-Aparicio, L. F. Ramos, J. Sena-Cruz, and D. González-Aguilera. Integrating geomatic approaches, operational modal analysis, advanced numerical and updating methods to evaluate the current safety conditions of the historical bôco bridge. *Construction and Building Materials*, 158:961–984, 2018. ISSN 0950-0618. doi: 10.1016/j.conbuildmat.2017.10.084.

A | Appendix A

A.1. Schematic organization of the module showing inheritance between classes of the PyOMA GUI



A.2. PyOMA interface guide



Contents

CONTENTS v

1. Updates of the newer releases

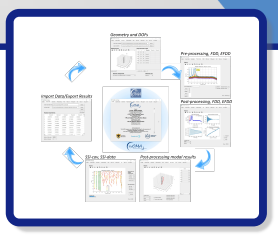
1.1 Last update of PyOMA_GUI users manual

October 29, 2022

1.2 Update version 1.1

- Minor bugs fixed in selecting identified peak frequencies;
- Minor bugs fixed in scale factor for visualizing deformed shape;
- Minor bugs fixed in general;
- Since GitHub does not allow pushing single files exceeding 100MB (total PyOMA_GUI.exe file is 104 MB) it is necessary to download the two archives and then extract it on local machine.

2. Introduction



In the structural health monitoring (SHM) paradigm, operational modal analysis (OMA) comprises several techniques and algorithms for estimating the dynamic characteristics of a structure in operational conditions from its vibration response. The OMA method has been spreading in the last years due to multiple advantages compared to input-output identification methods. In the current work, the authors present the implementation of a Python module named PyOMA and its Graphical User Interface (GUI) PyOMA_GUI. This software provides a user-friendly framework for the first time in the Python environment for estimating the experimental modal parameters (natural frequencies, mode shapes, damping ratios) of a structure from output-only vibration measurements in operational conditions.
PyOMA module is an open-source Python module that implements a complete output-only OMA framework for researchers, engineers, and practitioners. Through the implemented functions, it is possible to estimate the modal parameters of a civil structure using dynamic identification techniques derived from stochastic subspace identification (SSI) and frequency domain decomposition (FDD) [1], as illustrated in Fig.2.1. In addition, the authors provided a graphical user interface software version of the current module, called PyOMA_GUI, see Fig.2.2. The PyOMA_GUI aims to improve the appeal of the existing open-source Python OMA module, which has already been used in several applications. Not secondarily, the graphical user interface does not require any Python expertise or Python coding knowledge pre-requisite.
The present users manual is based on the research article [2] from the same authors.

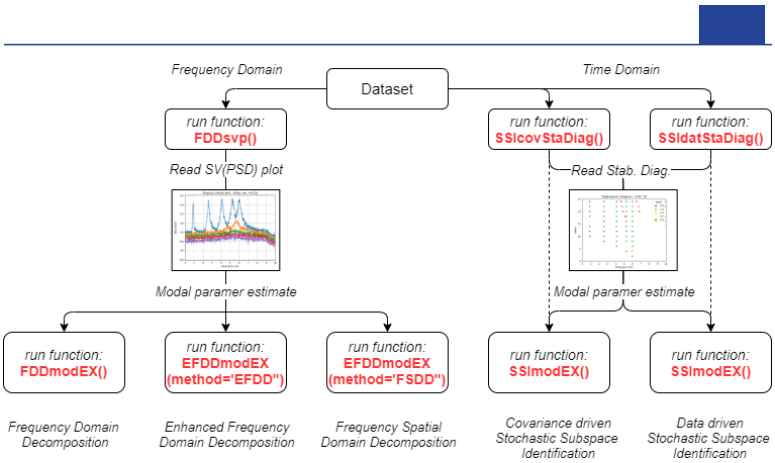


Figure 2.1: Functions implemented in PyOMA python module.

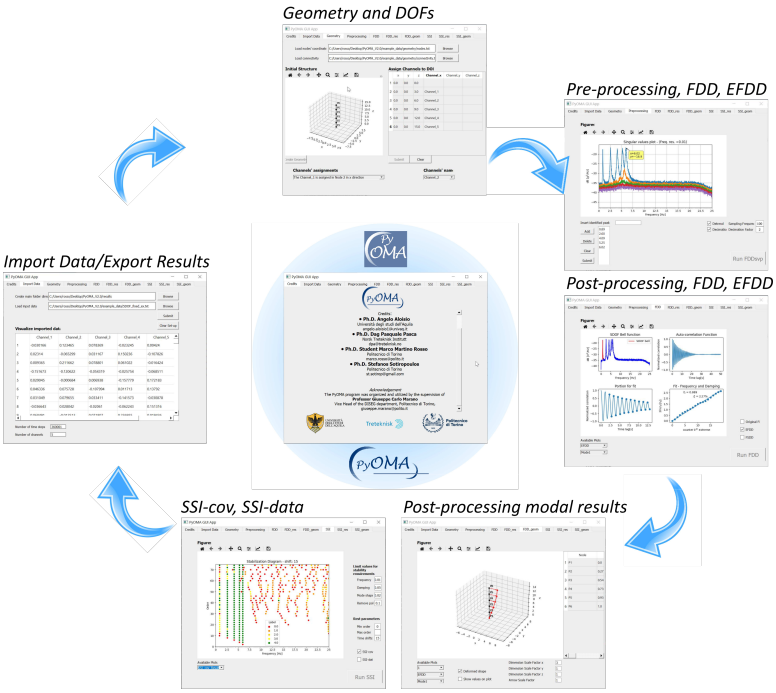


Figure 2.2: Functions implemented in PyOMA python module.

2.1 Example considered in the following chapters

In the following chapters, the users manual presents the main functionalities of the PyOMA_GUI software, based on a specific example. The output-only vibration data needed to carry on the proposed example have been generated with the function `oma.Exdata()` of the PyOMA python module.
Let us consider a 5 Degrees of Freedom (DOF) shear-type frame with lumped mass m = 25.91 [Ns^2/mm] at each floor, and the same story stiffness k = 10000 [N/mm] to all stories. Solving the eigenvalue problem gives the natural frequencies of the system:

$$f_n = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{Bmatrix} = \begin{Bmatrix} 0.88995 \\ 2.59776 \\ 4.09511 \\ 5.2607 \\ 6.0001 \end{Bmatrix} [Hz]$$

And the (unity normalized) mode shapes:

$$\Phi = \begin{bmatrix} \{\phi_1\} & \{\phi_2\} & \{\phi_3\} & \{\phi_4\} & \{\phi_5\} \end{bmatrix} = \begin{bmatrix} 0.28463 & -0.763521 & 1 & 0.918986 & -0.5462 \\ 0.5462 & -1 & 0.28463 & -0.763521 & 0.918986 \\ 0.763521 & -0.5462 & -0.918986 & -0.28463 & -1 \\ 0.918986 & 0.28463 & -0.5462 & 1 & 0.763521 \\ 1 & 0.918986 & 0.763521 & -0.5462 & -0.28463 \end{bmatrix}$$

The damping matrix is calculated assuming a constant damping of 2% to all modes. Synthetic signals, corresponding to the acceleration time history at each floor, are generated by the function using scipy's `signal.StateSpace` class. All the 5 DOF are excited by a Gaussian white noise input, then the results from each channel are polluted with a noise source with SN=10%.

3. Import data

Figure 3.1: PyOMA_GUI.exe icon.

When the user double click on the *PyOMA_GUI.exe* (Fig.3.1), the first *Credits*'s tab appears (Fig.3.2)
The first tab of the software is the *Import data* tab (Fig.3.3). Firstly, the user has to select an **empty folder** which would be the folder in which the software will automatically export the results, in terms of graphs images, and text files. This folder can be in any location of the O/S. It can be an existing folder or a new one that the user may create on the dialog box (Fig.3.4).
Thereafter, the user must upload the input data file with the Browse button. The current admissible file formats are **.txt, .xls, .xlsx, .csv**. The files have to be pre-processed externally, e.g. removing any textual headers, eventually providing to the software numerical columns data only (Fig.3.5). The first five lines of the uploaded data will be displayed in the table below (Fig.3.6). The user has the possibility to change the names of the channels, automatically set to *Channel_x*, for *x* as an integer value varying from 1 to the total number of columns of the imported file. With a double click on the headers' names, the user may manually modify the name referring to each specific column.
At the bottom of the Import data tab, the software will show the number of time steps and the number of channels retrieved from the imported data (Fig.3.7). In order to proceed, the user must push the submit button (Fig.3.8). Lastly, there is the Clear Set-up button that clears a whole set-up, to start from the beginning of a new analysis.

Figure 3.2: First view at the software opening: *Credits*' tab.

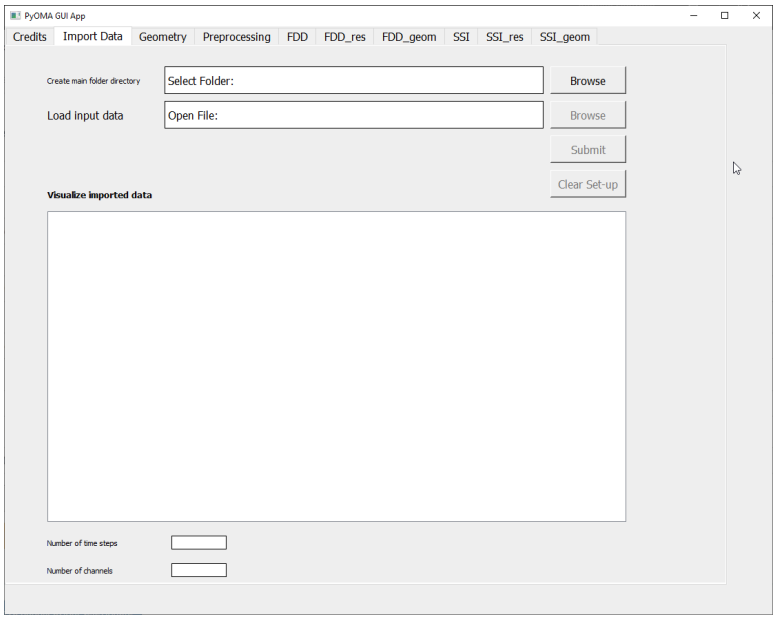


Figure 3.3: Second tab of the software: *Import data* tab.

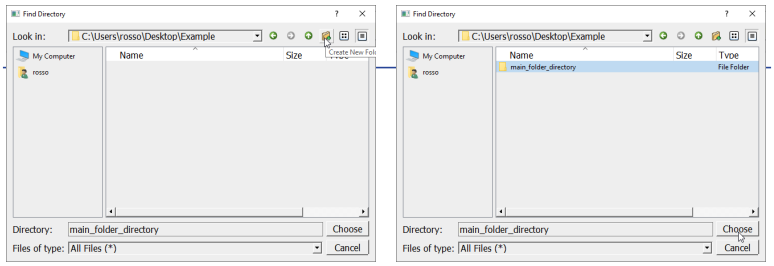


Figure 3.4: Create a new empty working directory.

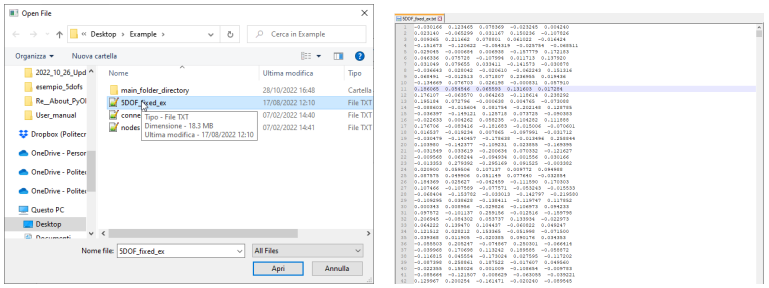


Figure 3.5: Create a new empty working directory.

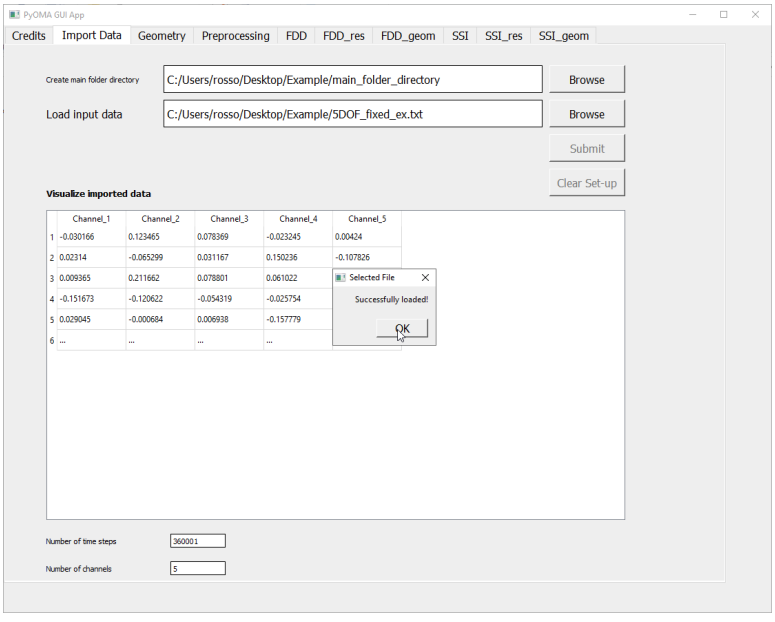


Figure 3.6: Create a new empty working directory.

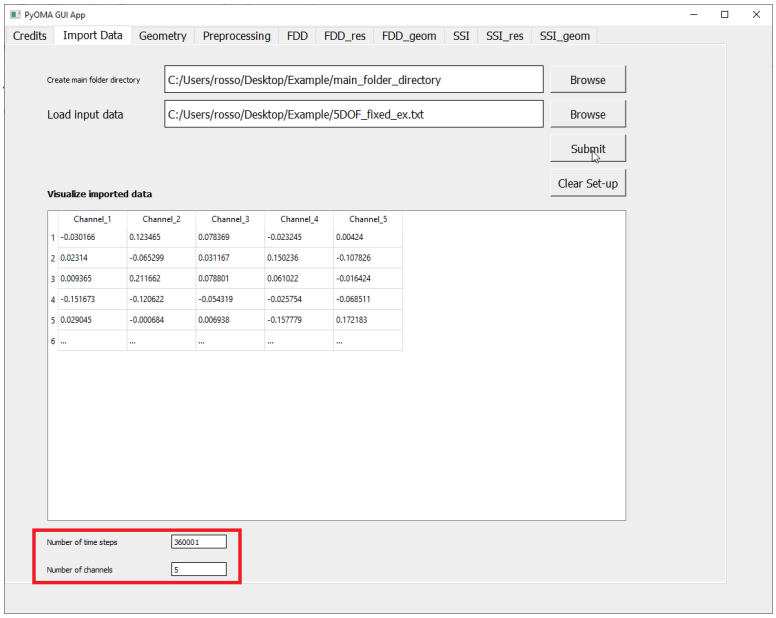


Figure 3.7: Create a new empty working directory.

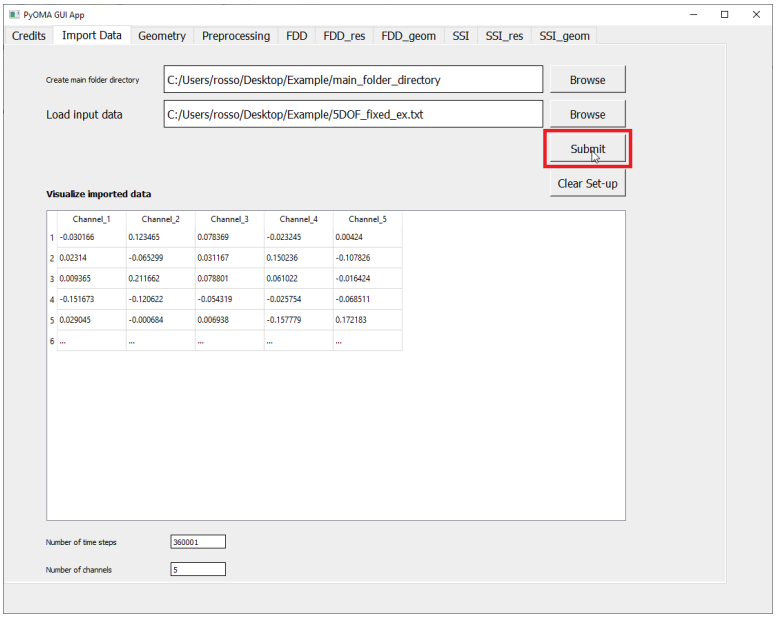


Figure 3.8: Create a new empty working directory.

4. Geometry

The second tab of the PyOMA_GUI software is the *Geometry* tab, see Fig.4.1. In this tab, the user has to browse and upload the text files (**txt** format) with the nodes' coordinates (Fig.4.2 and Fig.4.3) and connectivity (Fig.4.4) to visualize a schematic visualization of the structures and the monitored degree of freedoms (DOFs) and sensors' locations. Pushing the button Create Geometry the user can see a wire-frame model of the imported structure (Fig.4.5).
As shown in Fig.4.6, in the table **Assign Channels to DOF**, the user must manually connect the proper DOF of each node of the table (Channel_x, Channel_y, Channel_z) the name of the imported channel, referred to the specific column of the imported data from the previous tab. The user can check the correct name of the channel in the combo-box Channels' name (Fig.4.7). After pressing the submit button, the user may check the correct assignments of each DOF to the specific channel of the imported data file in the combo-box below (Fig.4.8).

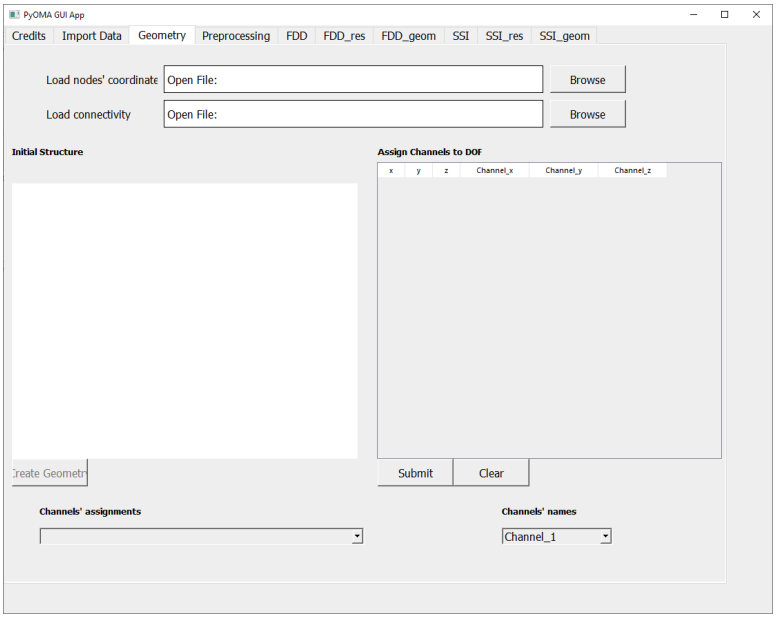


Figure 4.1: *Geometry* tab.

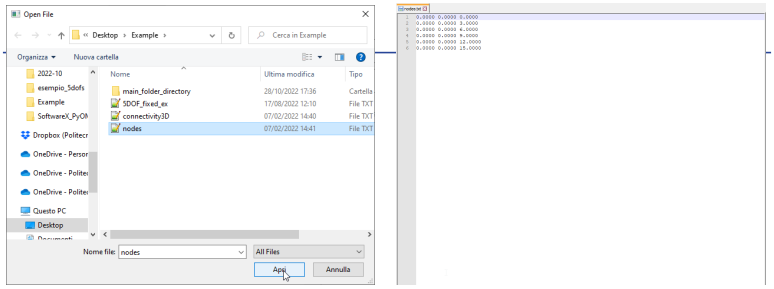


Figure 4.2: Upload nodes file.

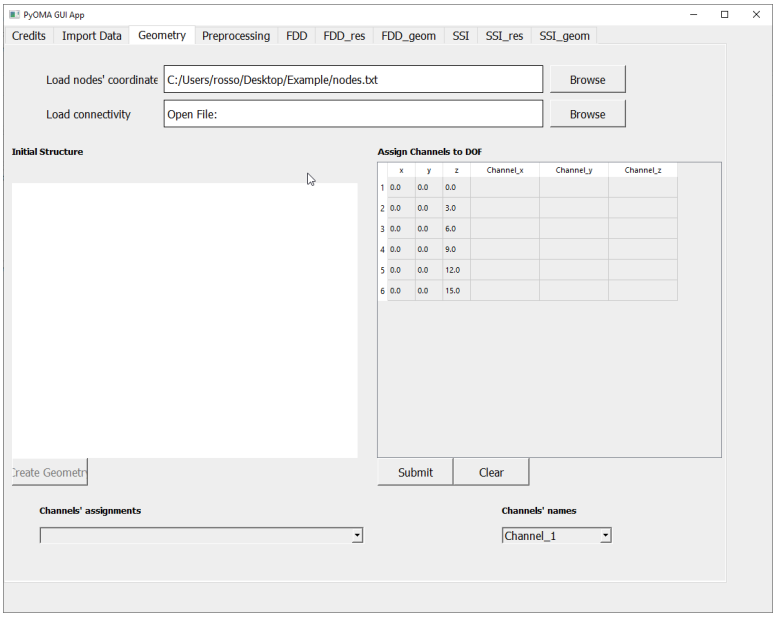


Figure 4.3: *Geometry* tab after uploading the nodes file.

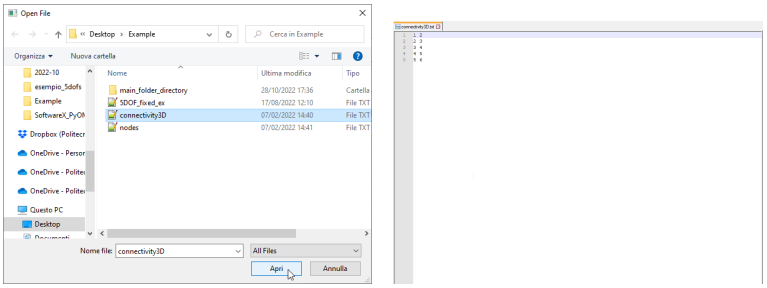


Figure 4.4: Upload nodes' connectivity file.

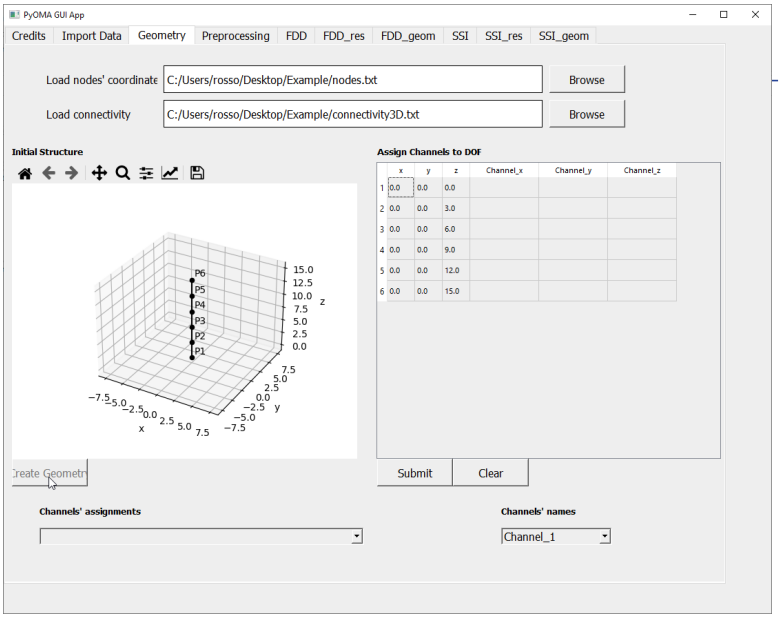


Figure 4.5: Graphical check of the imported nodes and connectivity.

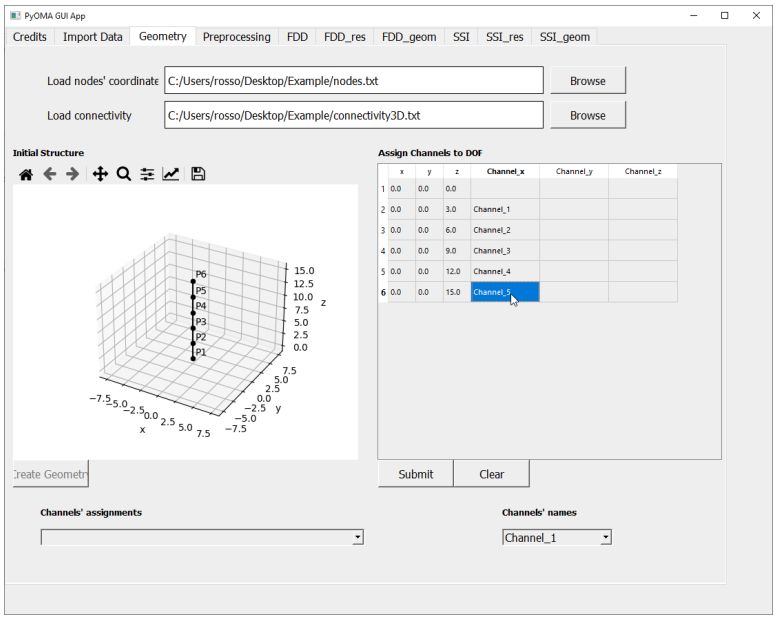


Figure 4.6: Manual assignment of the imported data channels to each node DOF.

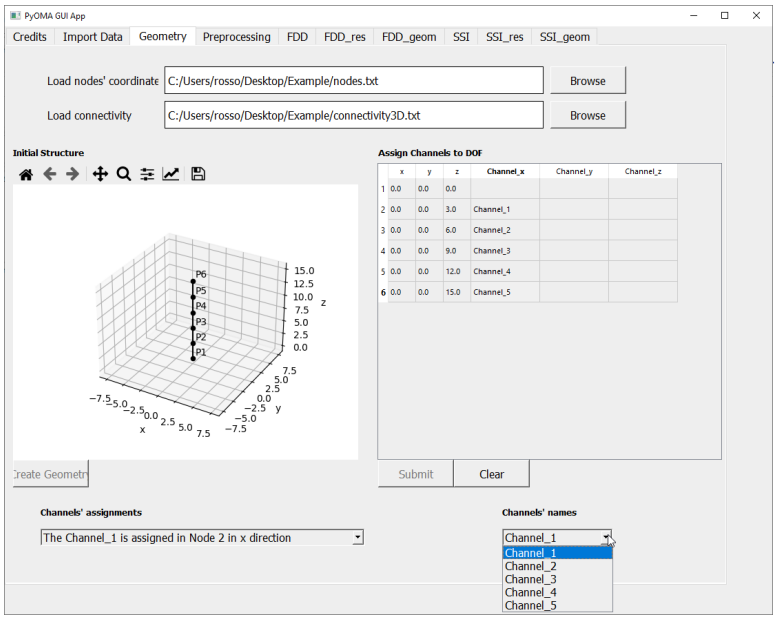


Figure 4.7: How to check the names of the imported data channels.

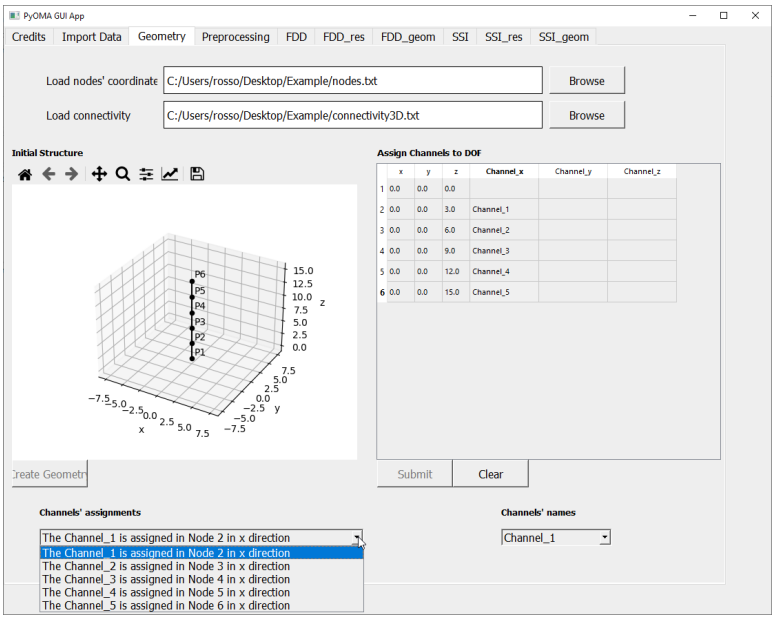


Figure 4.8: How to check if the nodes have been correctly assigned.

5. PreProcessing

In the tab *PreProcessing* (Fig.5.1), the user has the possibility to perform a prior signal processing in terms of detrending of the imported data files and a subsequent decimation procedure. At first, the user has to provide the **sampling frequency** of the signals and sign the corresponding checkboxes to perform detrening and/or decimation on the imported signals (Fig.5.2). The preprocessing is executed when the user clicks on the **Run FDD_svp** button (Fig.5.3). The command will run the function `oma.FDDsvp` of the PyOMA python module to perform the Frequency Domain Decomposition (FDD) algorithm, which returns a plot of the Singular Values (SV) of the Power Spectral Density (PSD) matrix, and a dictionary that contains the results that will be processed later to extract the modal properties. In the figure, the user can select the peak of interest and with the button Add, it is added to the list of the identified peaks. The rest buttons are to help us with the list (Delete a single item or Clear the whole list). With the submit button we are locking the selection and we are ready to proceed with the analysis.
At this point, the user has the possibility to perform the **peak peaking** procedure in order to select directly by clicking on the singular values graph (Fig.5.4), the peak corresponding to the natural frequencies of the structural system. After the click on a certain peak, the user can save the selected peak in the chosen peaks by clicking on the add button (Fig.5.5). After the peak selection, the user has to click submit button to proceed with the OMA analysis (Fig.5.6).

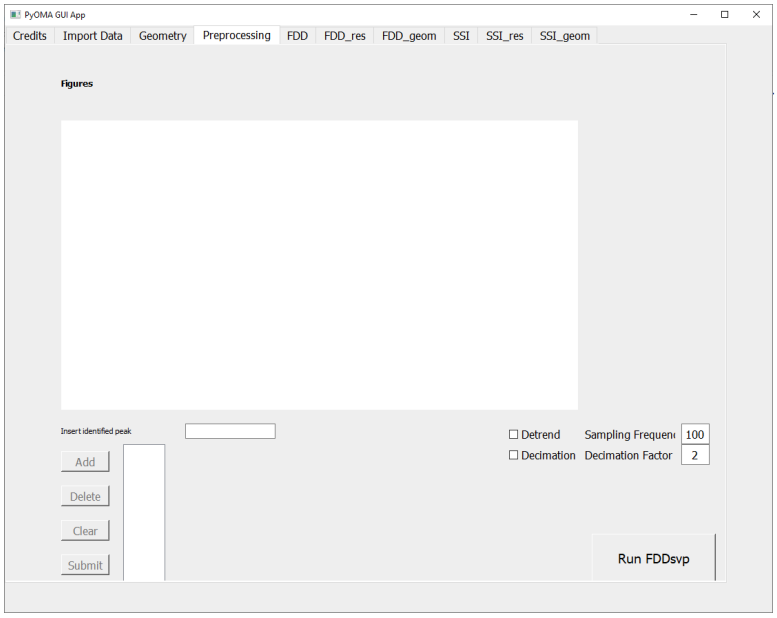


Figure 5.1: *Preprocessing* tab.

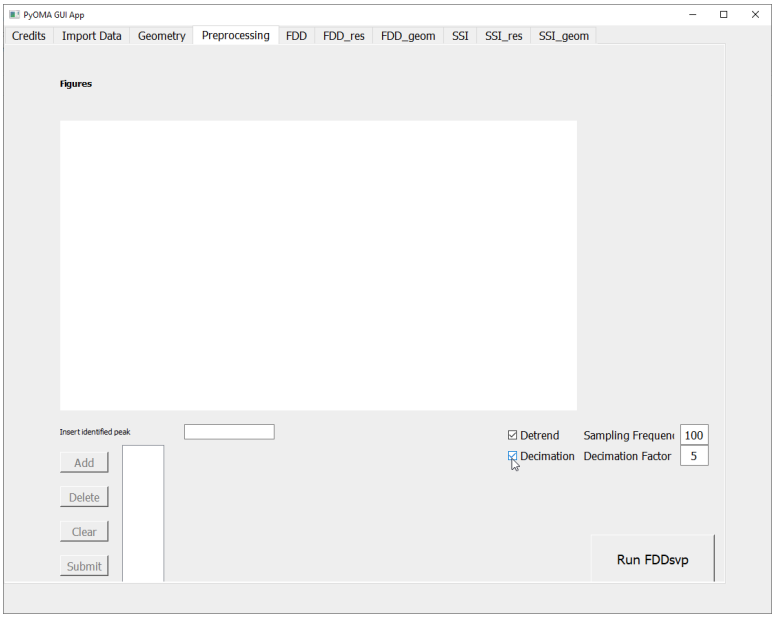


Figure 5.2: Sign the checkboxes to perform detrening and/or decimation on the imported signals.

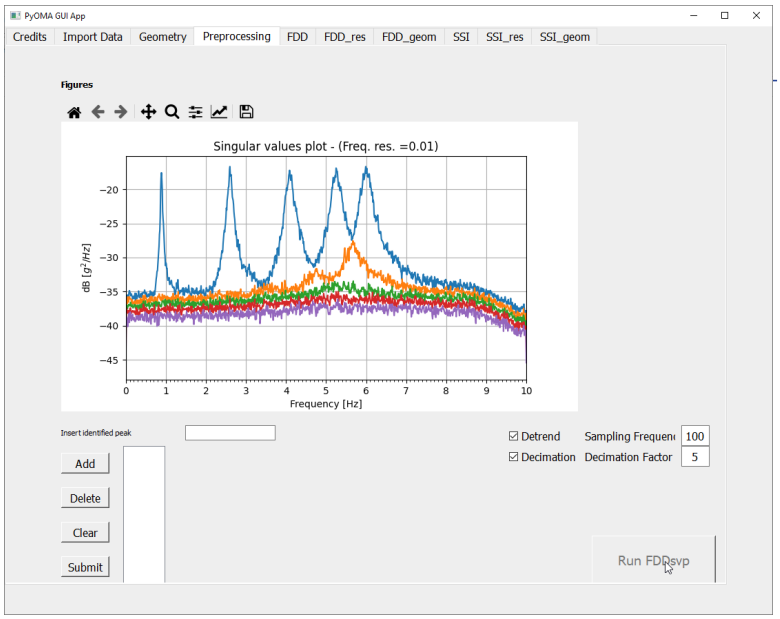


Figure 5.3: The preprocessing is executed when the user click on the **Run FDD_svp** button.

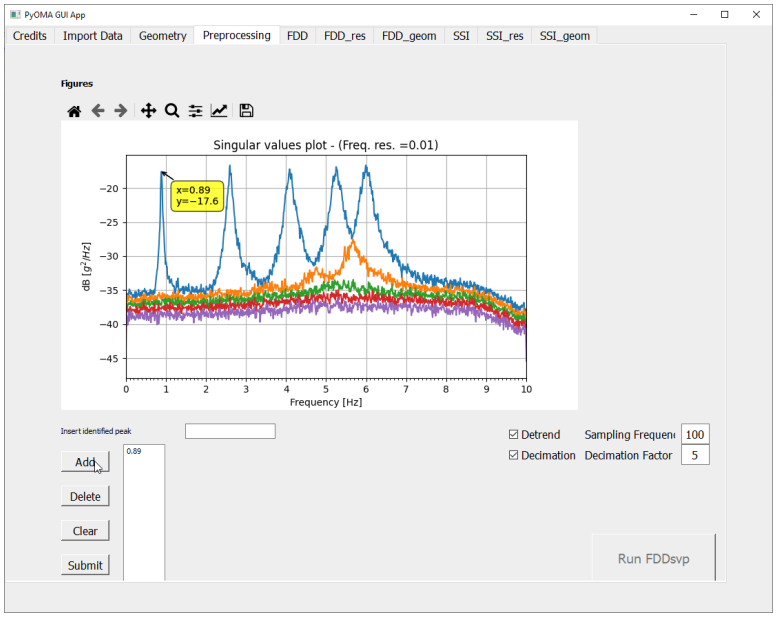


Figure 5.5: Add the peak of interest to the list of the saved peaks.

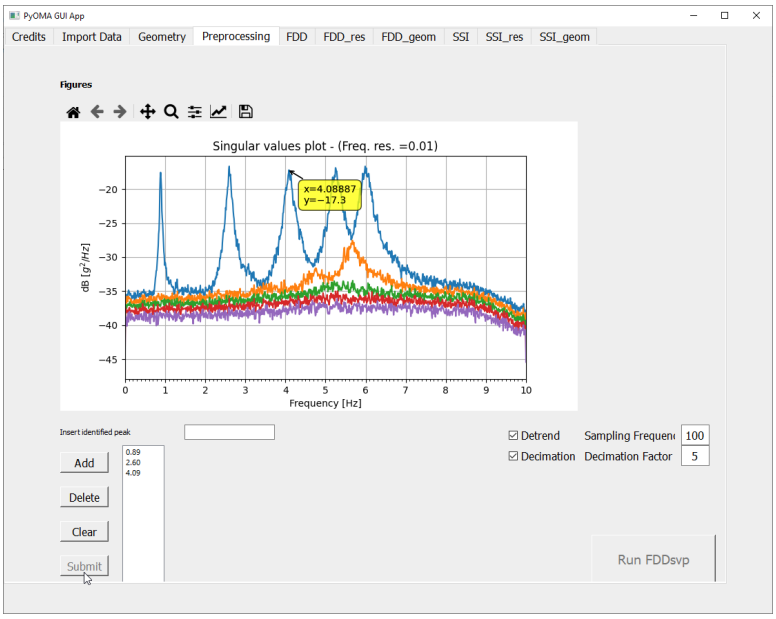


Figure 5.6: Submit the peaks of interest list to proceed with the OMA.

6. FDD

The current tab is the *FDD* tab, see Fig.6.1. This tab provides the user the ability to perform the frequency domain dynamic identification methods on the imported data based on the frequency domain decomposition (FDD) algorithm.
The user has to sign the checkboxes corresponding to the methods of interest and then push the button *Run FDD*. In the dropdown menu, there are the available graphs produced from the enhanced frequency domain decomposition (EFDD) and the Frequency-Spatial Domain Decomposition (FSDD) method [3]. These checkboxes allow the user to customize the return, permitting the extraction of the single DOF (SDOF) bells extraction from the PSD peaks [1], one for each mode, as depicted for instance in Figs. 6.2-6.3-6.4-6.5. The visualized figure is **refreshed** every time the user selects the desired mode from the dropdown menu is performed (Fig.6.5). The results of the analyses are automatically stored in the results directory previously selected in the *Import data* tab.
In detail, the first two checkboxes will run the function `oma.FDDmodEX` of the PyOMA python module and/or the `oma.EFDDmodEX` function to extract the modal information according to the FDD method and/or the EFDD method respectively. These functions return a dictionary that contains the results of the identification in terms of modal properties, which are shown in the next tab *FDD_res*. Specifically, the `oma.FDDmodEX` function will only extract the natural frequency and the mode shape, according to the original FDD algorithm as presented in [4]. On the other hand, the `oma.EFDDmodEX()` function extracts the modal properties (frequencies, mode shapes, damping) according to the EFDD algorithm as presented in [5]. The last checkbox will run the FSDD method which extracts the modal properties (frequencies, mode shapes, damping) according to the Frequency-Spatial Domain Decomposition method argued in [3]. The latter method isolates the modal coordinates by modal filtering and provides enhanced output PSD estimates, which yield better auto-correlation functions. These functions take as input the list of peaks previously identified in the singular values diagram provided by the function `oma.FDDsvp`.

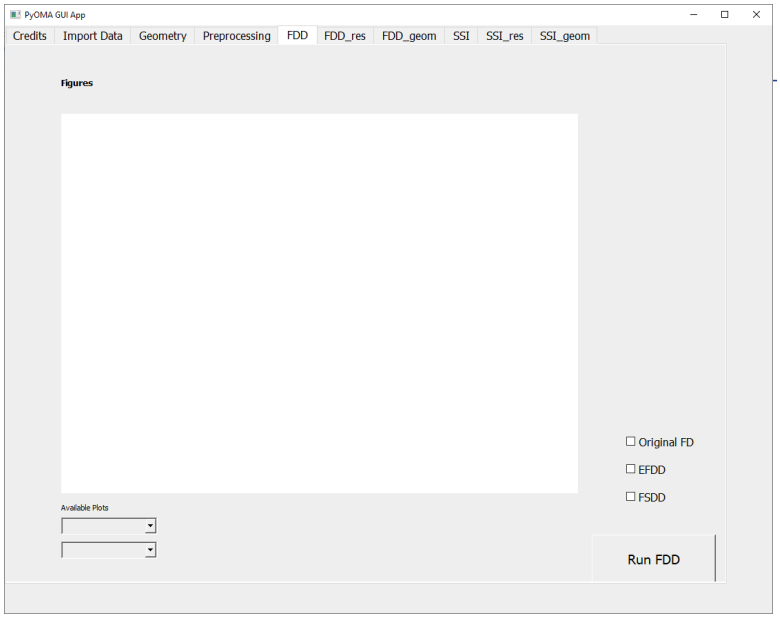


Figure 6.1: *FDD* tab.

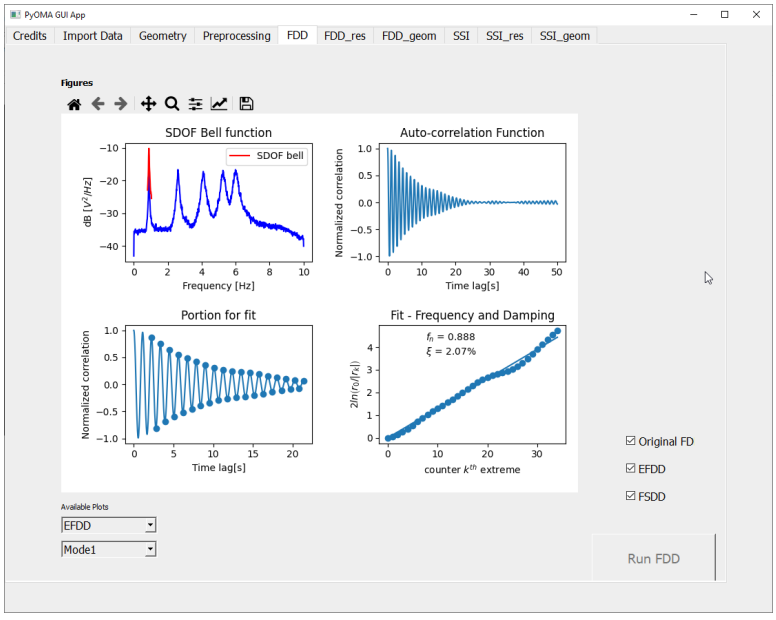


Figure 6.2: *FDD* tab.

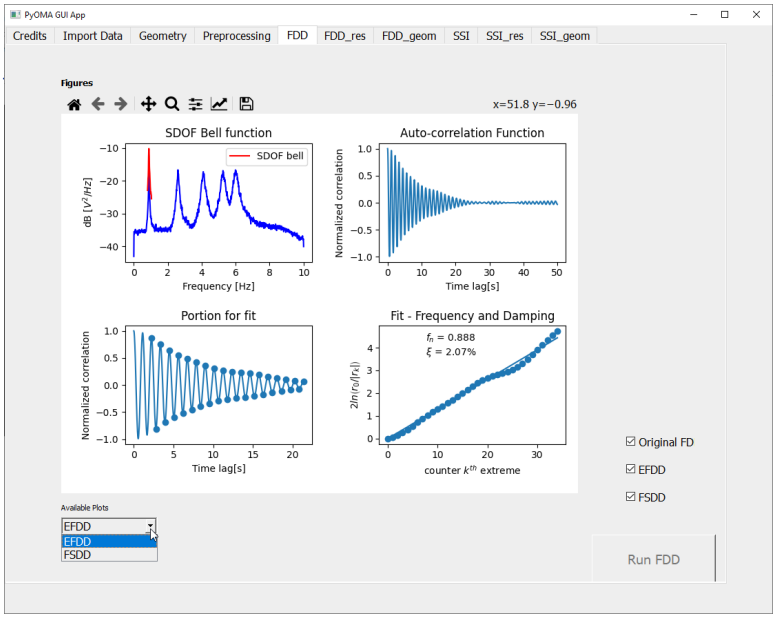


Figure 6.3: *FDD* tab.

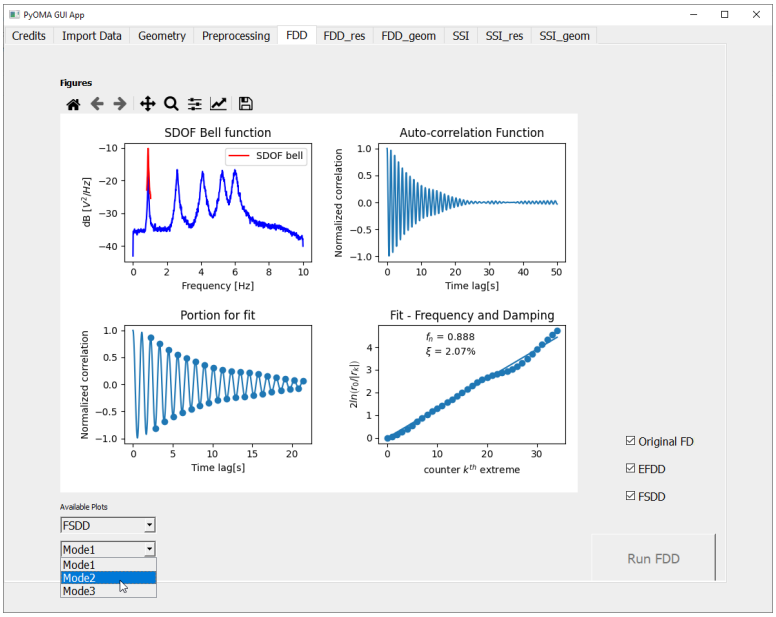


Figure 6.4: *FDD* tab.

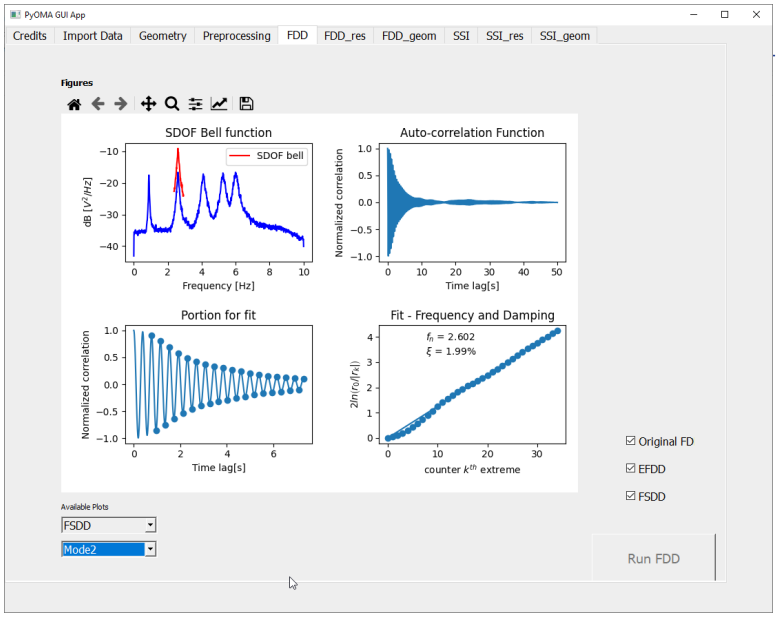


Figure 6.5: *FDD* tab.

7. FDD_res

In the current tab named *FDD_res* the user can inspect all the results derived from the FDD, EFDD, and/or FSDD analyses as depicted in Fig.7.1.
The results of the analyses and the SDOF bell figures have been automatically stored in the previously selected results directory in the *Import data* tab.

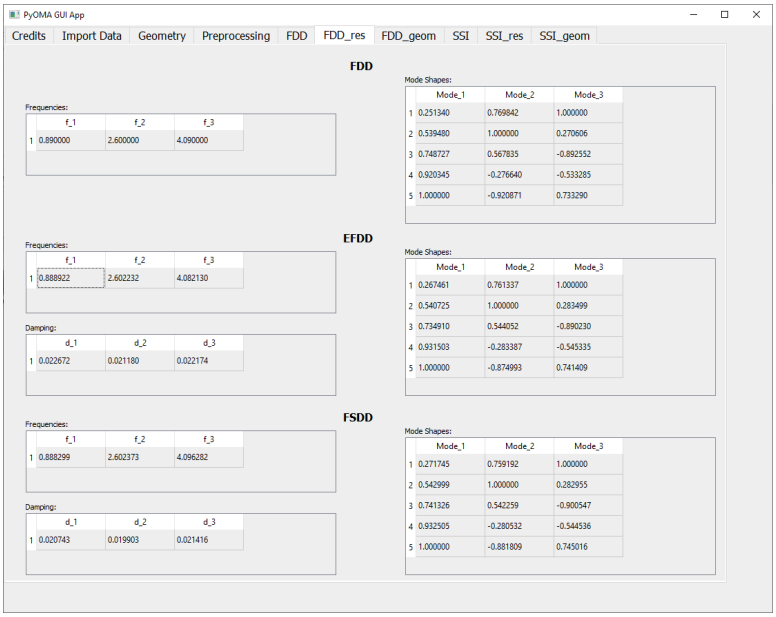


Figure 7.1: *FDD_res* tab.

8. FDD_geom

In the current tab named *FDD_geom* (Fig.8.1), the user can inspect the resulting mode shapes derived from the FDD, EFDD, and/or FSDD analyses as depicted in Figs.8.2-8.3. The resulting mode shapes can be visualized as static deformations, with the possibility to show them in several forms. As already mentioned, the figure is **refreshed** only when the mode of interest is selected from the side dropdown menu. In the side table, the values of the selected mode shape are reported for each node and each DOF, whereas the selected mode shape is represented in a vectorial way depicting red arrows (Fig.8.4). Extra abilities for the graph are provided, such as the *deformed shape* checkbox which permits to visualize of the mode shape as a piecewise line (Fig.8.6), or the *show values plot* checkbox which print the values of the vectorial components directly on the graph (Fig.8.7). Four scaling factors are also provided: the first three scale factors provide the ability to scale the deformed shape components in the global x,y, and z directions respectively (Fig.8.5), whereas the last one permits to increase or decrease the thickness of the depicted mode shape arrows (Figs.8.8-Fig.8.9).

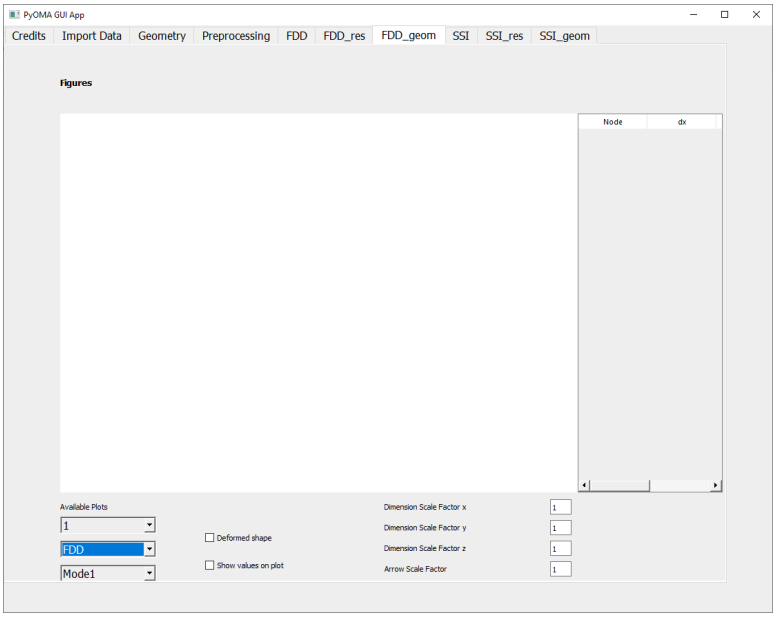


Figure 8.1: *FDD_geom* tab.

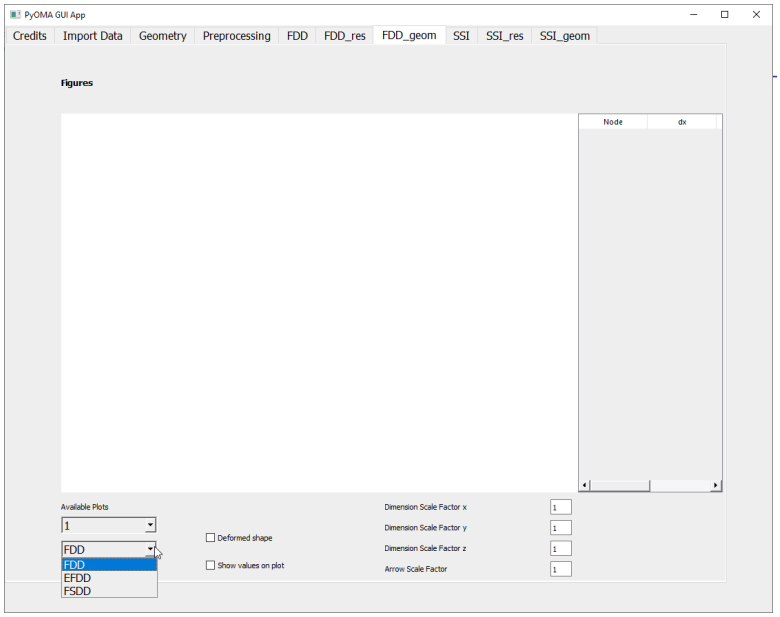


Figure 8.2: List of run methods. The user have to select the method of which is interested in.

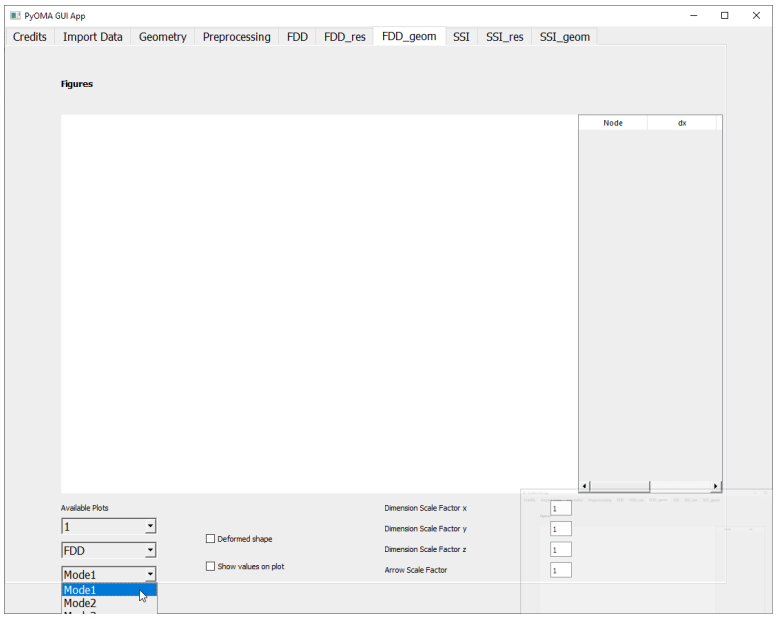


Figure 8.3: List of available mode shapes graphs for the selected method.

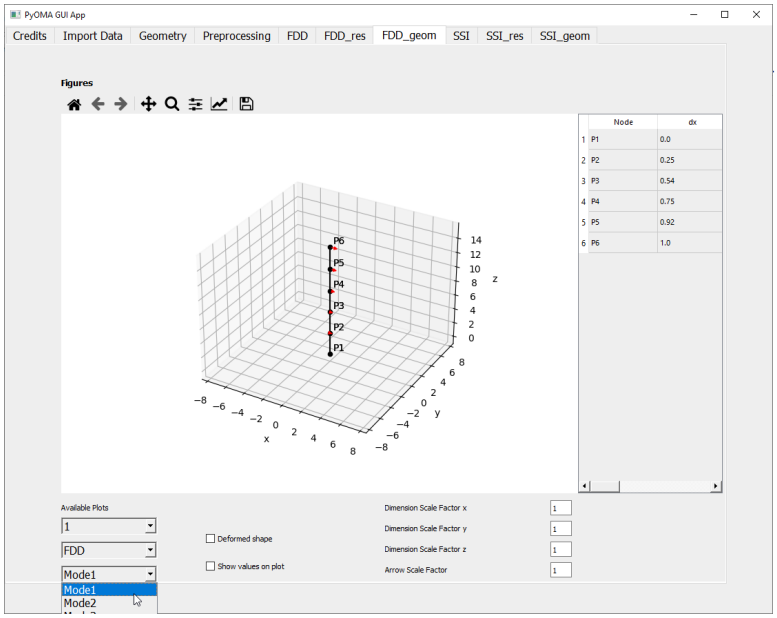


Figure 8.4: Mode shapes visualized as vectorial components only.

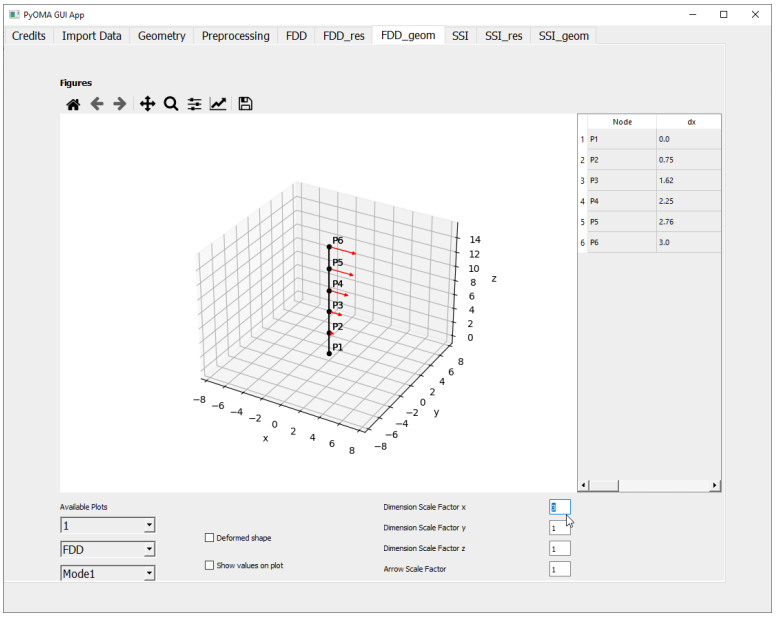


Figure 8.5: Mode shape vectorial components scaling for a better visualization.

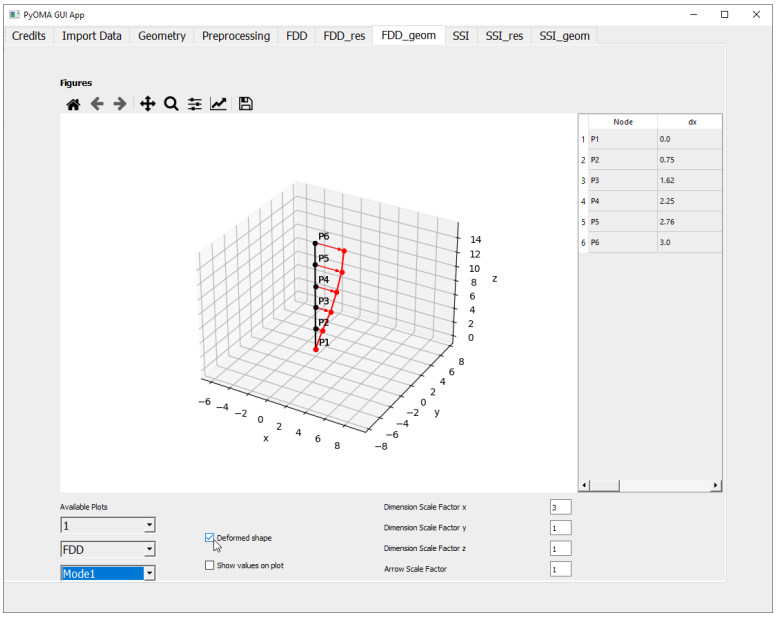


Figure 8.6: Mode shape representation with lines.

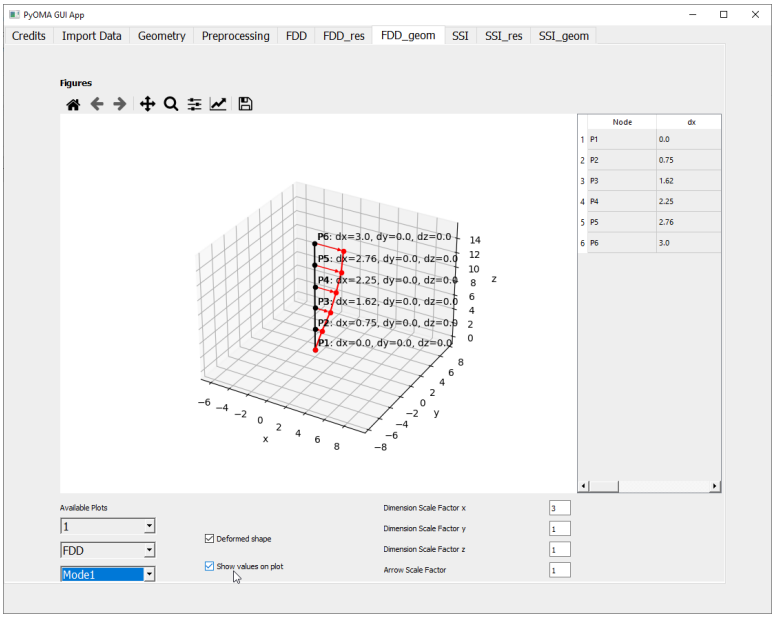


Figure 8.7: Adding annotations referred to the vectorial components of the mode shape 1.

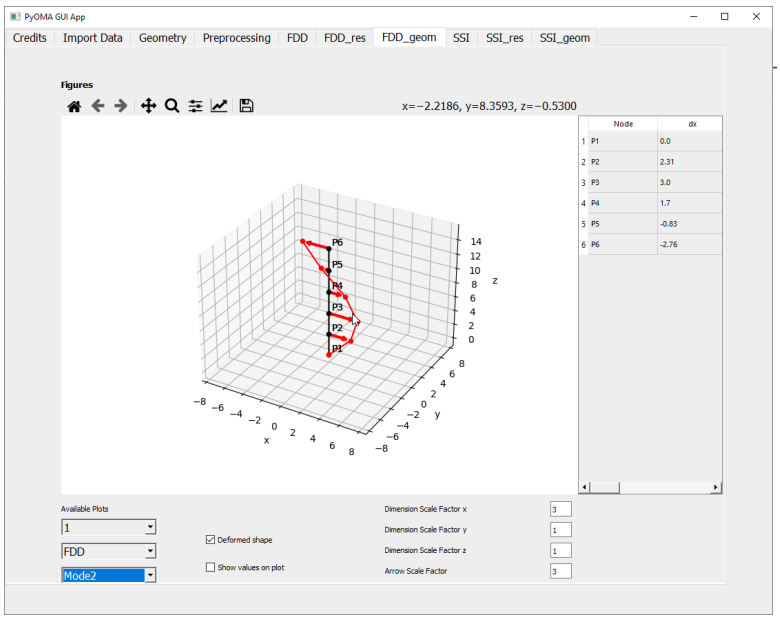


Figure 8.8: *FDD_geom* tab: mode shape 2.

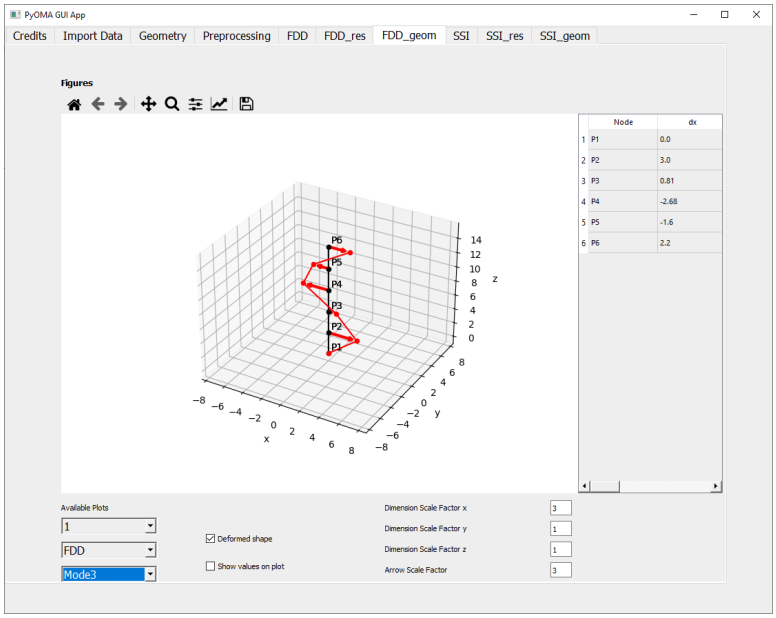


Figure 8.9: *FDD_geom* tab: mode shape 3.

9. SSI

The current tab is the *SSI* tab, see Fig.9.1. This tab provides the user the ability to perform the time domain dynamic identification methods on the imported data based on the stochastic subspace identification (SSI) algorithm.
The user has just to select the checkboxes corresponding to the desired type of SSI algorithm. The checkbox *SSI-dat* [6] performs the data-driven SSI algorithm version (Fig.9.3) which recalls the command `oma.SSIdatStaDiag` of the PyOMA python module. On the other hand, the checkbox *SSI-cov* executes the covariance-based SSI approach [7] (Fig.9.3) which recalls the command `oma.SSIcovStaDiag` of the PyOMA python module.
For these two SSI methodologies, the user is required to provide the number of block rows, better acknowledged as time lags or *time shift* parameter. The optional parameters allow the user to define the maximum model order, minimum order, and limit values to be used for the stability criteria of the poles in the stabilization diagram. The colors of the poles in Figs. 9.2 and 9.3, identified by the numbers 0.0 to 4.0 in the legend, indicate respectively: unstable, stable in frequency, stable in frequency and mode shape, stable in frequency and damping, stable in frequency damping and mode shape.
The results of the analyses and the generated stabilization diagrams are automatically stored in the results directory previously selected in the *Import data* tab.

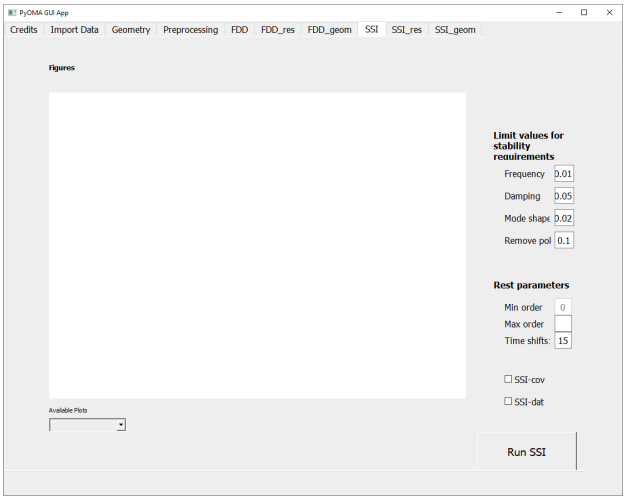


Figure 9.1: *SSI* tab.

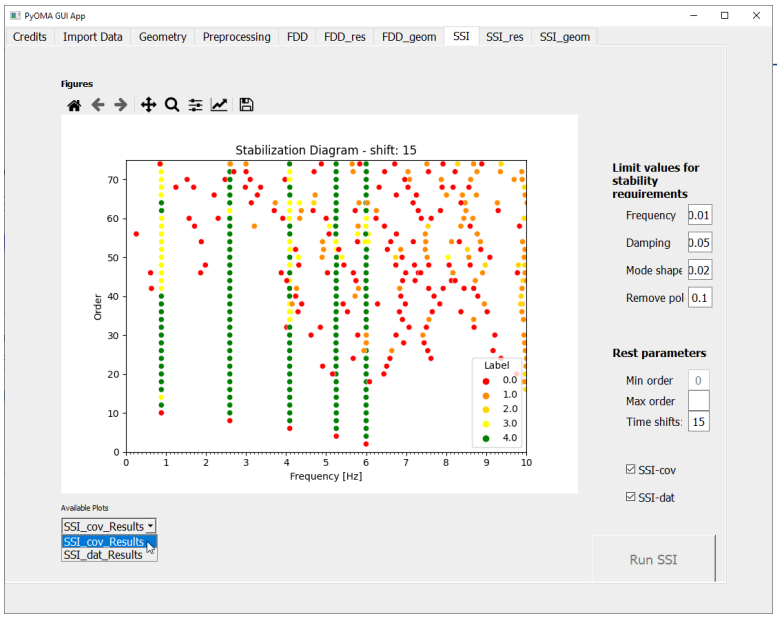


Figure 9.2: SSI-cov stabilization diagram.

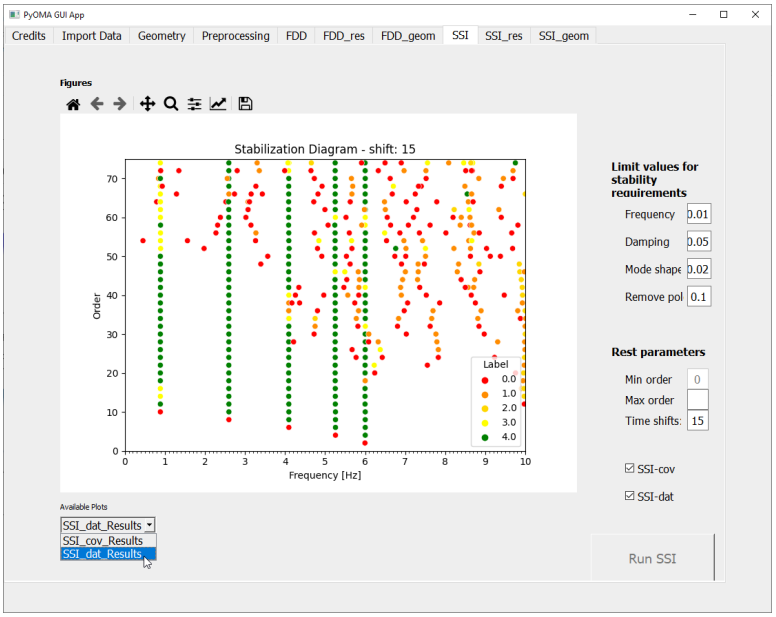


Figure 9.3: SSI-dat stabilization diagram.

10. SSI_res

In the current tab named *SSI_res* the user can inspect all the results derived from the *SSI* tab as depicted in Fig.7.1. The SSI results in terms of the natural frequencies of the system are referred to the identified alignments of the stable poles. These alignments are searched in the nearby of the natural frequencies values provided in the previous peak-peaking procedure.

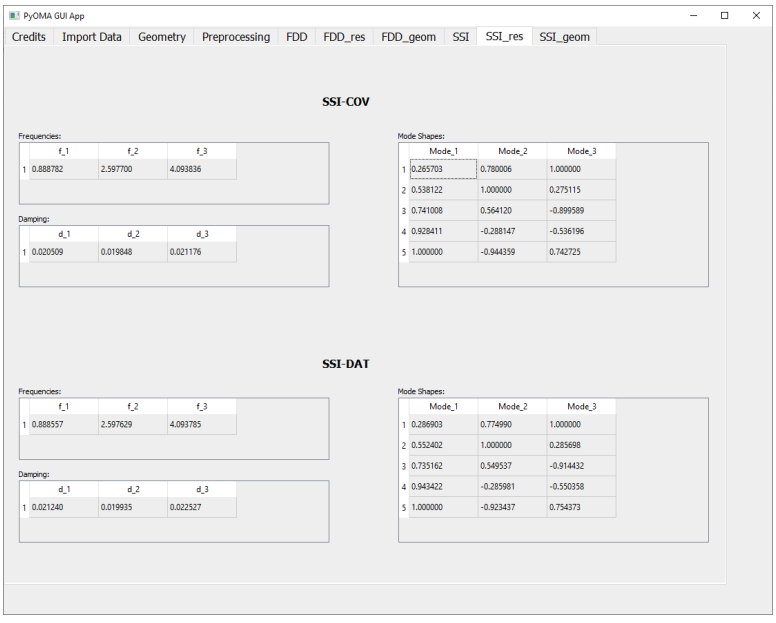


Figure 10.1: *SSI_res* tab.

11. SSI_geom

In the current tab named *SSI_geom* (Fig.11.1), the user can inspect the resulting mode shapes derived from the SSI-cov and/or SSI-dat analyses (Fig.11.2). The resulting mode shapes can be visualized as static deformations, with the possibility to show them in several forms. As already mentioned, the figure is **refreshed** only when the mode of interest is selected from the side dropdown menu. In the side table, the values of the selected mode shape are reported for each node and each DOF, whereas the selected mode shape is represented in a vectorial way depicting red arrows (Fig.11.3). Extra abilities for the graph are provided, such as the *deformed shape* checkbox which permits to visualize of the mode shape as a piecewise line (Fig.11.4), or the *show values plot* checkbox which print the values of the vectorial components directly on the graph (Fig.11.7). Four scaling factors are also provided: the first three scale factors provide the ability to scale the deformed shape components in the global x,y, and z directions respectively (Fig.11.5), whereas the last one permits to increase or decrease the thickness of the depicted mode shape arrows (Fig.11.6).

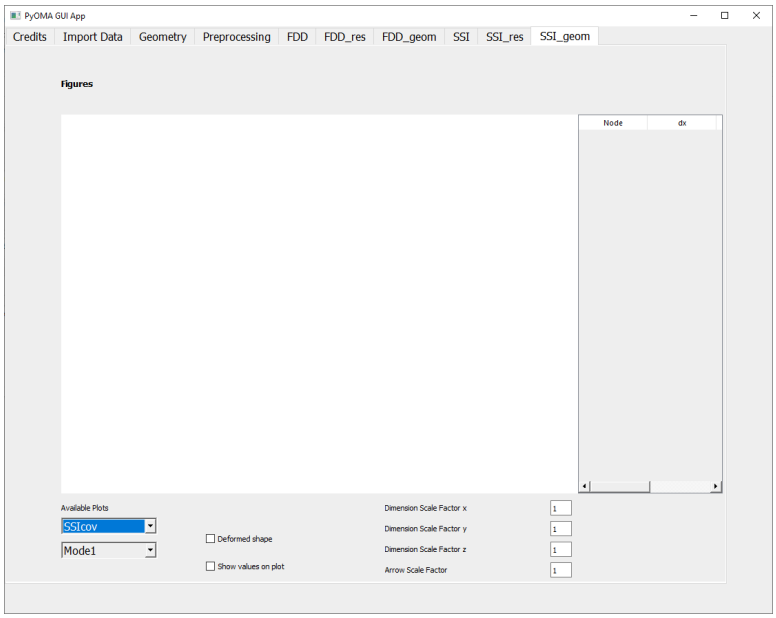


Figure 11.1: *SSI_geom* tab.

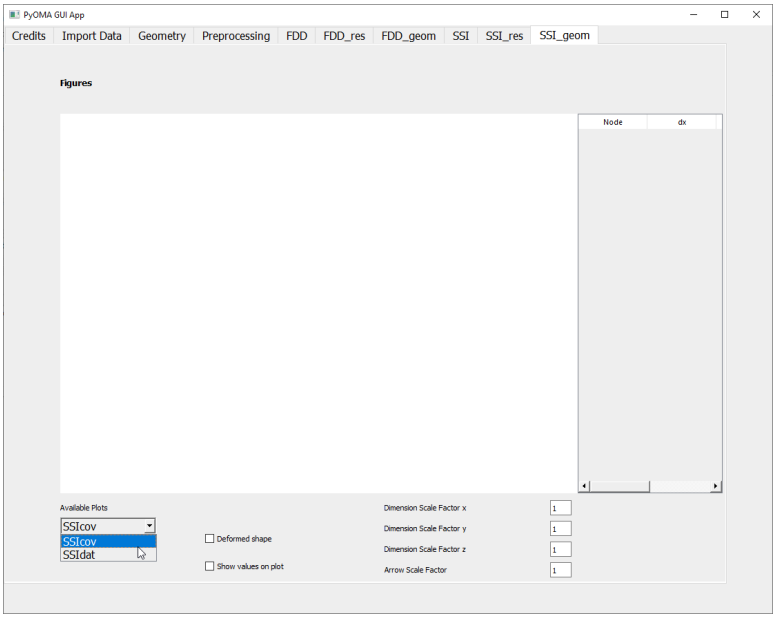


Figure 11.2: *SSI_geom* tab.

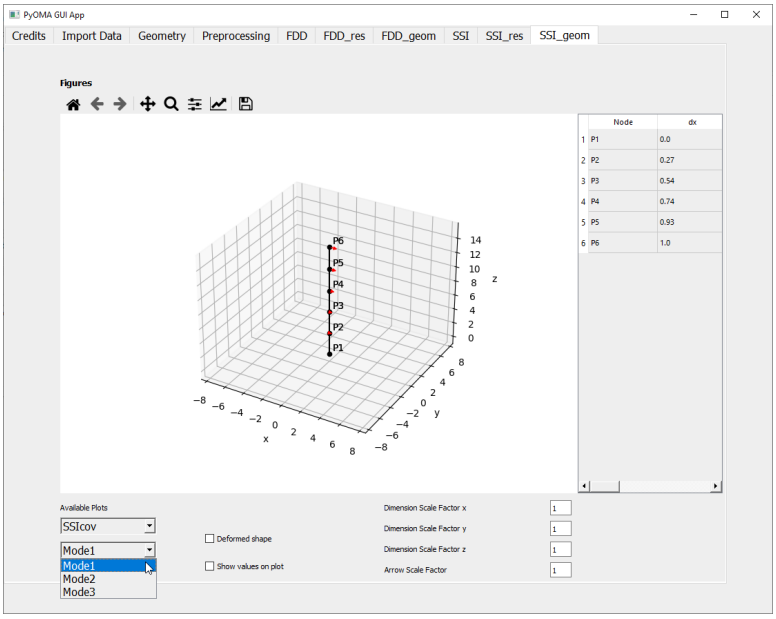


Figure 11.3: *SSI_geom* tab.

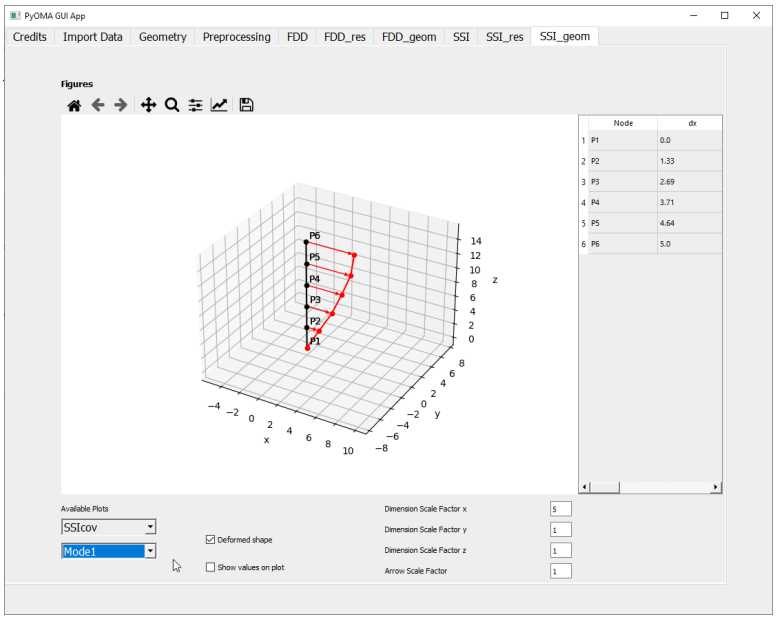


Figure 11.4: *SSI_geom* tab.

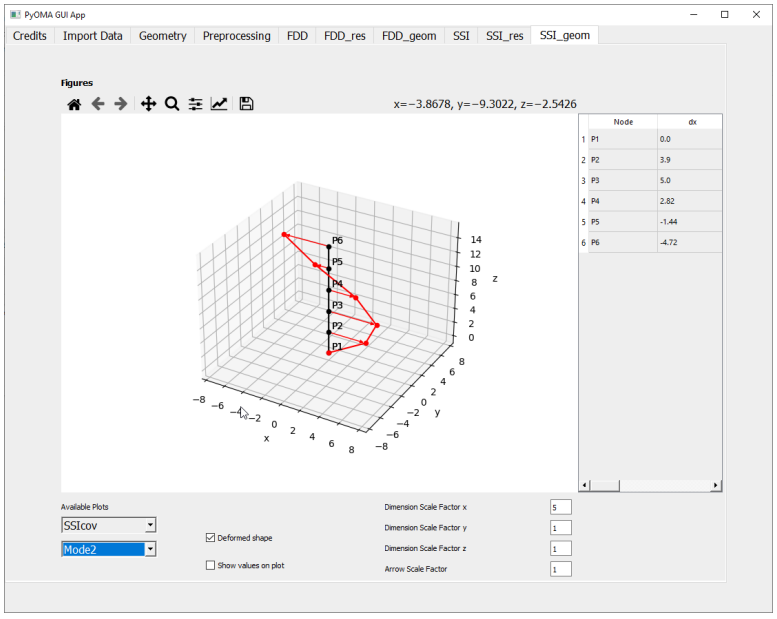


Figure 11.5: *SSI_geom* tab.

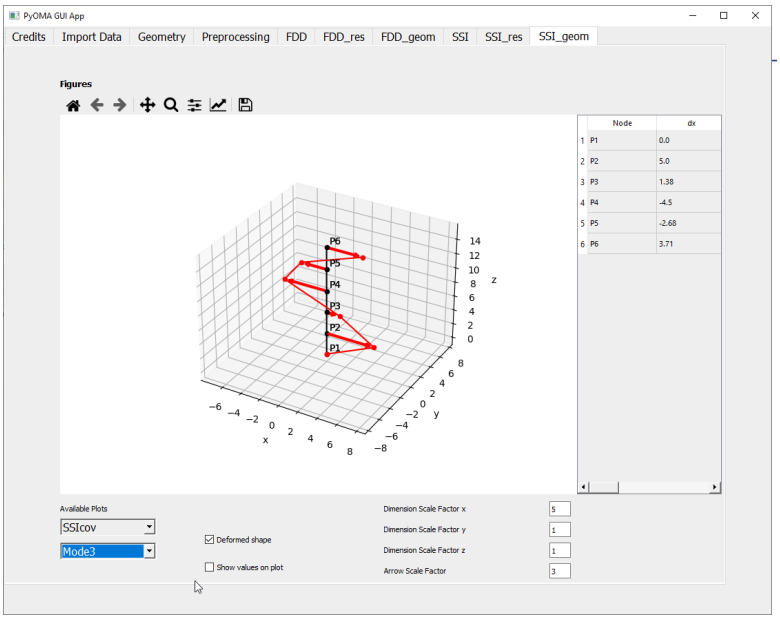


Figure 11.6: *SSI_geom* tab.

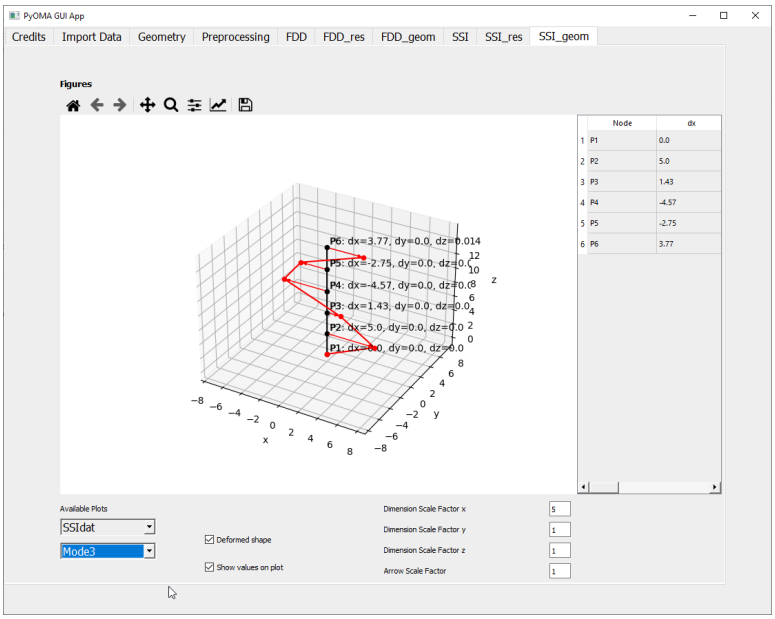


Figure 11.7: *SSI_geom* tab.

12. Example

The PyOMA_GUI has been implemented with PyQt5 library, adopting a notebook-style with tabs that noticeably help every type of user follow the right steps to perform OMA in a semi-assisted approach. In the following, the main steps in Fig. 2.2 are briefly described and discussed. However, with the software, the authors provided the files with the same example of the 5 Degrees of Freedom (DOFs) shear-type frame discussed in the previous sections.

12.1 Import Data

After running the executable PyOMA_GUI file, the user must select the first tab to import vibration data, as illustrated in Fig. 12.1. In the first place, the user must select the current working directory, i.e. a folder in which the output files will be stored. It can be in any location on the user's machine, and it can be an existing folder or a new one that may be created on the dialogue box. After that, the user must browse the input data file, which will also be displayed in the table to check the success of the data uploading. At the user's will, it is possible to customize the imported data table headers' names with a simple double click. The data visualization permits the user to check the number of time steps and the number of channels. To proceed with the following stages of the OMA procedure, the user must push the `Submit` button. Furthermore, to start a new analysis, the user may press the `Clear Set-up` button to completely reset the PyOMA_GUI software memory and cleanse the uploaded data.

12.2 Geometry

The geometry definition is fundamental to provide the user with a simplified visualization of the identified mode shapes according to the monitored DOFs and the available measurement channels, as depicted in Fig. 12.2 (a). The user must browse a text file which contains the nodes' coordinates and connectivity to visualize a

starting undeformed scheme of the structure under study according to the monitored DOFs only. Pushing the button `Create Geometry` a wire-frame graph of the structure appears. In the table `Assign Channels to DOF`, the user must insert the exact index of the table (Channel_x, Channel_y, Channel_z) and the precise name of the monitored channel/DOF. Those identical names are shown in the combo-box Channels' name to remind the user, preventing him from returning to the previous imported data tab tediously. In the combo box, the exact assignments of the channels are provided.

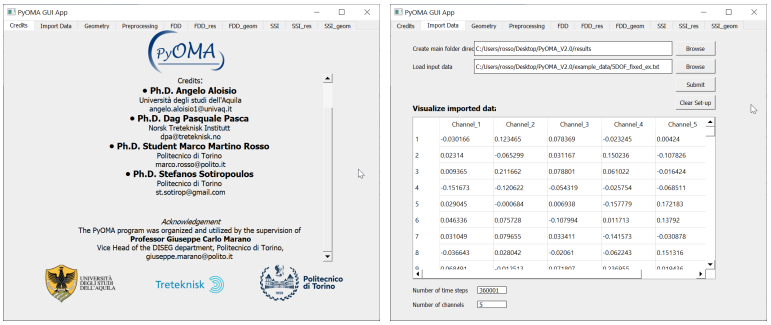
12.3 Pre-processing and peak-peaking approach

The PyOMA_GUI allows the user to perform signal basic pre-processing procedures in the same manner as the PyOMA module, illustrated in Fig. 12.2 (b). The user must select the parameters of the problem, e.g. the sampling frequency and decimation factor and some default values are provided if the user does not set them explicitly. Thereafter, it is possible to run the `FDD_svp` function. In the SV graph figure, it is possible to perform the peak-peaking approach to select the peak of interest, and with the button `Add`, they are added to the list of the identified peaks. The other buttons close to the list permits customizing the identified peaks list, e.g. deleting a single item or clearing the whole list. With the `Submit` button, the user confirms the selection of the identified peaks, and it is possible to proceed with analyzing the results of the FDD approaches.

12.4 FDD, SSI and their implemented variants with PyOMA_GUI

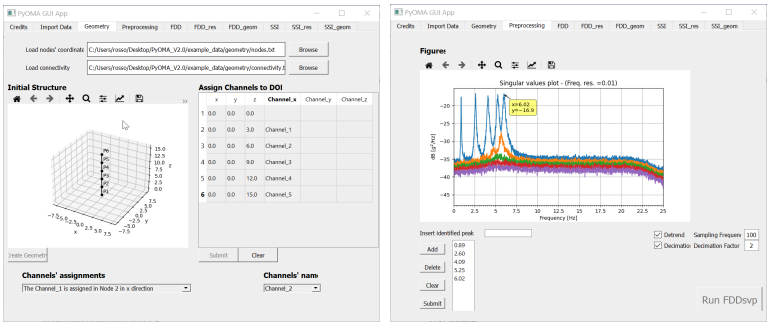
Passing to the next tab of the notebook, it is possible to run the FDD, SSI and their implemented variants in the PyOMA module with the graphical user interface software. In the FDD tab, the user can choose in the checkboxes the methods of interest among FDD, EFDD and FSDD. Pushing the button Run FDD, the results are illustrated in terms of SDOF bell extraction for each mode. The available diagrams are listed in the dropdown menu in the bottom-left part of the window, and the above figure refreshes the user anytime to modify its selection. In the `FDD_res` tab, the user may explore all the frequency domain algorithms' results in tabular form. All the results (figures included) are automatically stored in the working directory previously selected. In the `FDD_geom` tab, the final mode shape deformations are presented in several forms, as depicted in Fig. 12.3. By selecting the desired mode, the figure is updated. In the table, the deformation on each node is presented, while extra functionalities for the plot are provided. The checkbox `Deformed shape` illustrates in the same figure the deformed structure, and the checkbox `Show values` activates the functionality to show the information of the table directly on the figure. At the

same time, four scaling factors are also provided to adjust and customize the mode shape appearance. The last remaining tabs are related to the SSI method, which working mechanisms are similar and equivalent to the so-far illustrated frequency-domain methods. The stabilization diagram obtained with the PyOMA_GUI software is illustrated in Fig. 12.4.



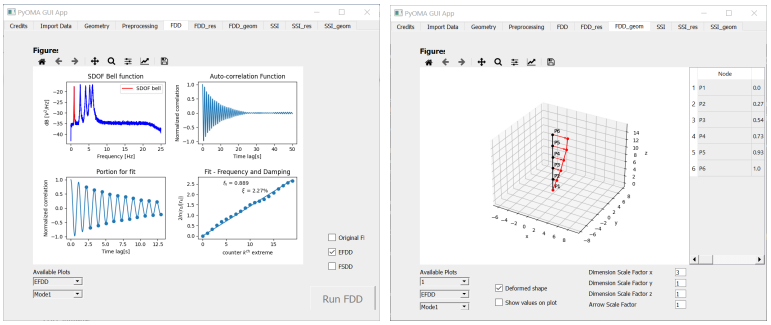
(a) (b)

Figure 12.1: PyOMA_GUI 5 DOFs shear type example overview. (a) Initial tab after starting the software execution; (b) Import data tab.



(a) (b)

Figure 12.2: PyOMA_GUI 5 DOFs shear type example overview. (a) geometry definition; (b) Preprocessing and FDD algorithm execution with SV decomposition diagram for peak-peaking identification.



(a) (b)

Figure 12.3: PyOMA_GUI 5 DOFs shear type example overview. (a) geometry definition; (b) Preprocessing and FDD algorithm execution with SV decomposition diagram for peak-peaking identification.

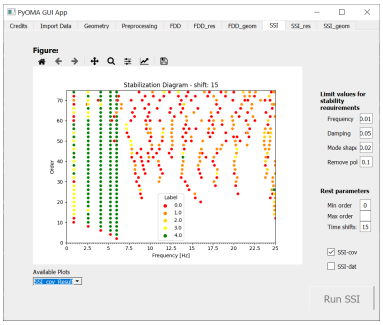


Figure 12.4: PyOMA_GUI 5 DOFs shear type example overview. Stabilization diagram retrieved with SSI algorithm.

Bibliography

[1] Carlo Rainieri and Giovanni Fabbrocino. Operational modal analysis of civil engineering structures. *Springer, New York*, 142:143, 2014.

[2] Dag Pasquale Pasca, Angelo Aloisio, Marco Martino Rosso, and Stefanos Sotiropoulos. Pyoma and pyoma_gui: A python module and software for operational modal analysis. *SoftwareX*, 20:101216, 2022.

[3] Lingmi Zhang, Tong Wang, and Yukio Tamura. A frequency–spatial domain decomposition (fsdd) method for operational modal analysis. *Mechanical systems and signal processing*, 24(5):1227–1239, 2010.

[4] Rune Brincker, Lingmi Zhang, and Palle Andersen. Modal identification of output-only systems using frequency domain decomposition. *Smart materials and structures*, 10(3):441, 2001.

[5] Rune Brincker, Carlos E Ventura, and Palle Andersen. Damping estimation by frequency domain decomposition. In *Proceedings of IMAC 19: A Conference on Structural Dynamics: februar 5-8, 2001, Hyatt Orlando, Kissimmee, Florida, 2001*, pages 698–703. Society for Experimental Mechanics, 2001.

[6] Peter Van Overschee and BL De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[7] Bart Peeters and Guido De Roeck. Reference-based stochastic subspace identification for output-only modal analysis. *Mechanical systems and signal processing*, 13(6):855–878, 1999.

B | Appendix B

B.1. DR Coefficients and Optimization code to find the optimal parameters

```
import numpy as np
import pandas as pd
from scipy.optimize import minimize   # <-- added

# ---------- USER SETTINGS ----------
csv_file = r"D:\Thesis\03. Model Updating\08. DR L_24310 Matching F1\Coeff1.csv"   # <-- your file
num_params = 6              # x1..x5
num_modes = 3               # f1..f3
num_FE_Runs=2*num_params+1
print(f"\nThe number of FE runs needed for the updating parameters {num_params} are {num_FE_Runs}")
# -----------------------------------

# 1. Read data (no header in file)
col_names = [
    "run_id",
    "x1_Edeck",
    "x2_Rhodeck",
    "x3_Ebarrier",
    "x4_Rhobarrier",
    "x5_Rhoasphalt",
    "x6_SpanLength",
    "f1", "f2", "f3"
]
df = pd.read_csv(csv_file, header=None, names=col_names)
```

```
25
26  # 2. Parameter matrix (R x N)
27  X_params = df[["x1_Edeck",
28                 "x2_Rhodeck",
29                 "x3_Ebarrier",
30                 "x4_Rhobarrier",
31                 "x5_Rhoasphalt",
32                 "x6_SpanLength"]].to_numpy()
33  R = X_params.shape[0]
34
35  # 3. Build DR design matrix: [1, x1, x1^2, ..., xN, xN^2]
36  #    size: (R x (1 + 2*num_params))
37  X_design = np.ones((R, 1 + 2*num_params))
38
39  for k in range(num_params):
40      xk = X_params[:, k]
41      X_design[:, 1 + 2*k]   = xk        # linear term in x_k
42      X_design[:, 1 + 2*k+1] = xk**2     # quadratic term in x_k
43
44  print(X_design)
45
46  # 4. Solve for DR coefficients mode by mode
47  coeffs = {}  # mode_index -> coefficient vector theta_i
48
49  for mode_idx in range(1, num_modes+1):
50      f = df[f"f{mode_idx}"].to_numpy()   # (R,)
51      # 11 equations, 11 unknowns: solve exactly (or use lstsq for
        ↪ robustness)
52      theta = np.linalg.lstsq(X_design, f, rcond=None)[0]
53      coeffs[mode_idx] = theta
54
55  # 5. Print coefficients
56  param_names = ["x1_Edeck",
57                 "x2_Rhodeck",
58                 "x3_Ebarrier",
59                 "x4_Rhobarrier",
60                 "x5_Rhoasphalt",
```

```
                "x6_SpanLength"]

for mode_idx in range(1, num_modes+1):
    theta = coeffs[mode_idx]
    print(f"\n=== Mode {mode_idx} DR coefficients ===")
    print(f"C_{mode_idx} = {theta[0]:.6g}")
    for k in range(num_params):
        Aik = theta[1 + 2*k]
        Bik = theta[1 + 2*k + 1]
        print(f"  A_{mode_idx},{k+1} (linear in {param_names[k]}):
        ↪  {Aik:.6g}")
        print(f"  B_{mode_idx},{k+1} (quad   in {param_names[k]}):
        ↪  {Bik:.6g}")

# 6. Helper: evaluate DR-predicted frequencies for a given [x1..x6] to
↪  see if the system works and found if the
# estimates by DR match the FE frequencies for each run.
def evaluate_dr(x_factors, coeffs_dict=coeffs):
    """
    x_factors: iterable of length 5 [x1, x2, x3, x4, x5]
    coeffs_dict: output 'coeffs' from above
    returns: np.array of length num_modes with predicted frequencies
    ↪  [f1..f4]
    """
    x_factors = np.asarray(x_factors, dtype=float)
    freqs = []
    for mode_idx in range(1, num_modes+1):
        theta = coeffs_dict[mode_idx]
        val = theta[0]  # C_i
        for k in range(num_params):
            xk = x_factors[k]
            val += theta[1 + 2*k] * xk + theta[1 + 2*k + 1] * xk**2
        freqs.append(val)
    return np.array(freqs)

# 7. Quick check: DR prediction at each run should match input
↪  frequencies fairly well
```

```
print("\nCheck: DR prediction vs FE frequencies for each run:")
for idx, row in df.iterrows():
    x_vec = row[["x1_Edeck",
                 "x2_Rhodeck",
                 "x3_Ebarrier",
                 "x4_Rhobarrier",
                 "x5_Rhoasphalt",
                 "x6_SpanLength"]].to_numpy()
    f_dr = evaluate_dr(x_vec)
    f_fe = row[["f1","f2","f3"]].to_numpy()
    print(f"Run {int(row['run_id'])}: FE = {f_fe},  DR =
    ↪  {np.round(f_dr, 4)}")

# ---------------------------------------------------------------------
# 8. OPTIMIZATION: minimize J = 1/2 * sum_i ((f_i^DR -
↪  f_i^exp)/f_i^exp)^2
# ---------------------------------------------------------------------

# Experimental frequencies (Hz)
f_exp = np.array([4.64, 5.23, 10.64])   # modes 1..3

def objective(x):
    """
    Objective function J(x) for scipy.optimize.minimize.
    x: array-like length 5 [x1, x2, x3, x4, x5]
    """
    f_dr = evaluate_dr(x, coeffs)        # DR-predicted frequencies
    r = (f_dr - f_exp) / f_exp           # relative errors
    return 0.5 * np.dot(r, r)            # 1/2 * sum(r_i^2)

# Initial guess (nominal factors)
x0 = np.ones(num_params)

# Bounds for the N parameters (factors)
bounds = [
    (0.5, 1.4),  # x1 = E_Deck factor
    (0.6, 1.5),  # x2 = rho_Deck factor
```

```
    (0.2, 1.4),   # x3 = E_Barrier factor
    (0.5, 1.7),   # x4 = rho_Barrier factor
    (0.35, 1.40),  # x5 = rho_Asphalt factor
    (0.7, 1.4),    # x6 = Span_Length factor
]

# Run the optimizer
res = minimize(
    objective,
    x0,
    method="L-BFGS-B",
    bounds=bounds
)

print("\n===== OPTIMIZATION RESULT =====")
print("Optimal x* (factors) =", res.x)
print("Minimum J(x*) =", res.fun)

f_opt = evaluate_dr(res.x, coeffs)
print("DR frequencies at optimum:", f_opt)
print("Relative errors at optimum [%]:", (f_opt - f_exp) / f_exp *
↪  100.0)
```

C | Appendix C

C.1. VBI modeling procedure in Abaqus

Once the base bridge FE model has been obtained , the vehicle-bridge interaction (VBI) model can be implemented in Abaqus following a step-by-step procedure. The sequence below is intended to ensure numerical robustness and to reduce common convergence issues associated with contact and transient dynamics as discussed in sections 2, 6, and [2].

- **Bridge baseline model.** Complete the bridge model definition (geometry, mesh, materials/sections, connections, and boundary conditions) and verify that the bridge analysis runs correctly on its own.
- **Analysis steps definition.** Define the analysis sequence needed for a stable moving-load simulation:
    1. **Static step.** A preliminary static step might be used to apply the bridge model only, accounting for the self weight of the structure.
    2. **Implicit Dynamic (vehicle gravitational settlement).** an implicit dynamic step was then introduced to apply the vehicle mass under gravity and allow the system to reach equilibrium. The step duration was selected long enough for the transient effects associated with the sudden activation of the vehicle weight to decay, so that the response is not affected by an abrupt application of the self weight.
    3. **Implicit Dynamic (vehicle crossing).** finally, a second mplicit dynamic step is performed to simulate the vehicle crossing the bridge under consideration at a constant speed, with the wheel–deck interaction enforced through the adopted node-to-surface contact formulation.

Step Manager

	Name	Procedure	Nlgeom	Time
✔	Initial	(Initial)	N/A	N/A
✔	Static	Static, General	OFF	1
✔	FirstGravitational	Dynamic, Implicit	OFF	1.5
✔	Moving_Load	Dynamic, Implicit	OFF	1.945

Figure C.1: Analysis Steps

- **Vehicle definition (single sprung-mass model).** Create the vehicle as:
  1. a **rigid body** representing the vehicle body (sprung mass) controlled by a reference point (RP_S),
  2. a **wheel point** defined by a second reference point or node set close to the deck (RP_W).

Figure C.2: Body car and wheel definition

- **Suspension elements.** Connect RP_S and RP_W using a **spring stiffness** k and a **dashpot damping** c, acting along the vertical direction.

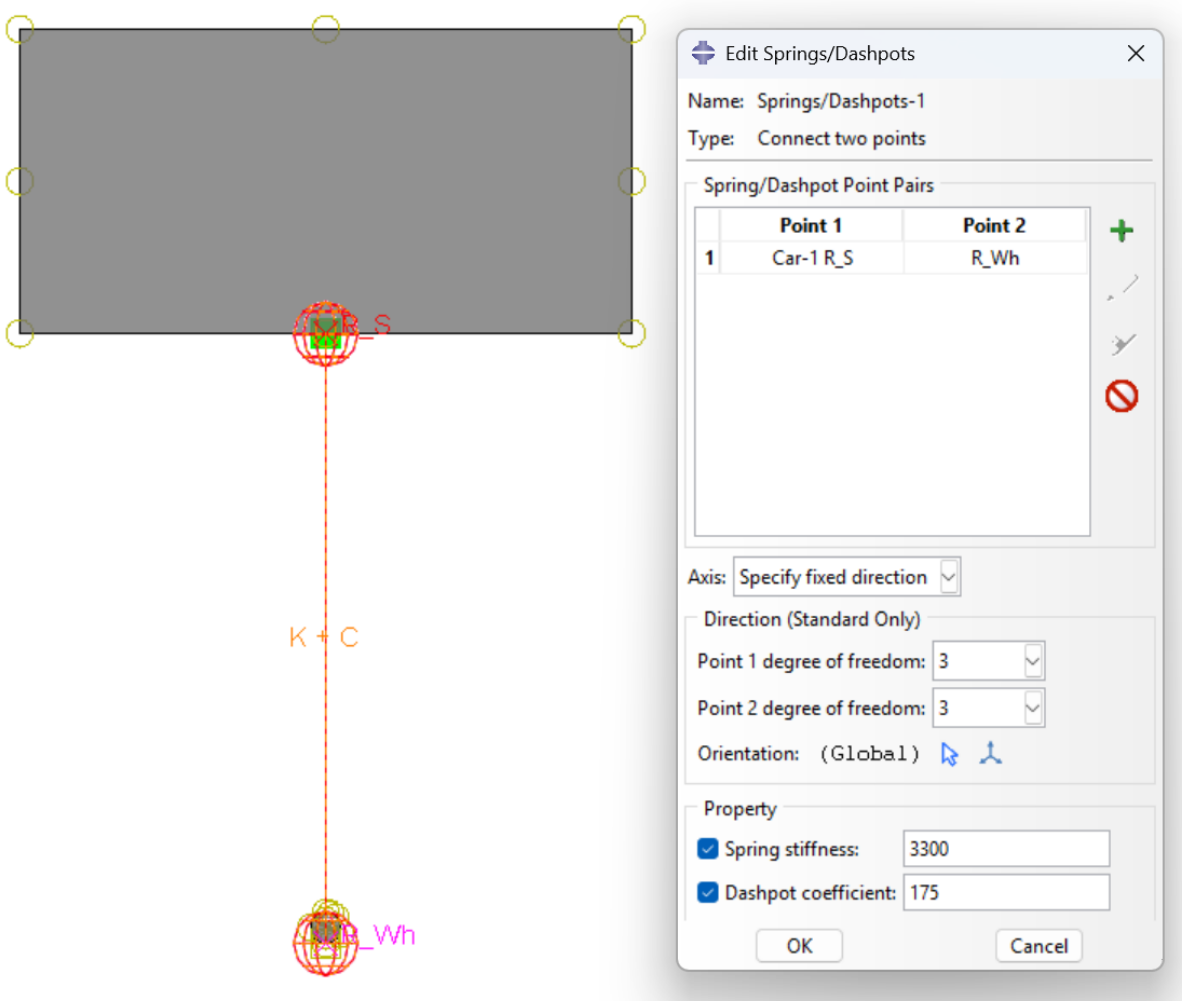
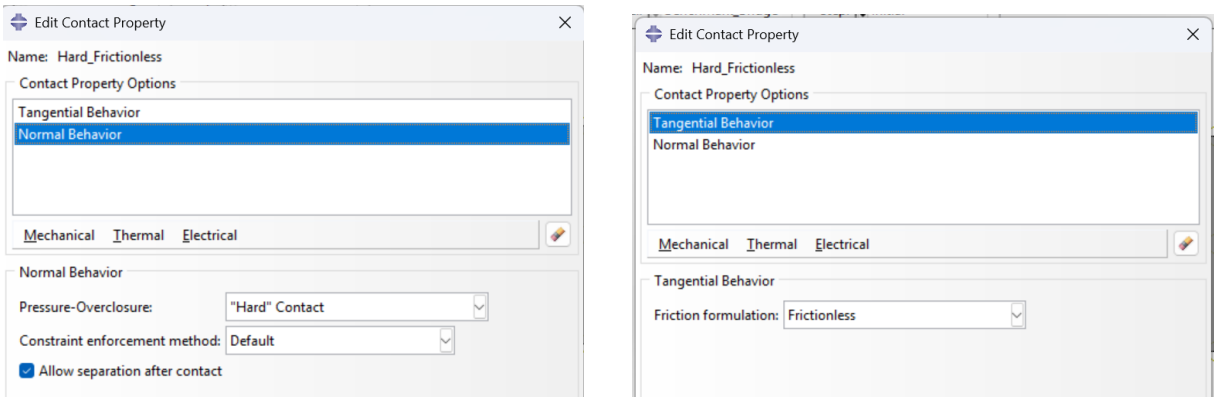


Figure C.3: Spring stiffness and Dashpot Coefficient

- **Contact interaction properties.** Define a contact property with:
  1. **Normal behaviour:** hard contact,
  2. **Tangential behaviour:** frictionless



(a) Normal Behavior Definition (b) Tangential Behavior Definition

Figure C.4: Contact properties

- **Contact geometry definition.** Create:
  1. the **deck top surface** as the master contact surface,
  2. the **wheel node/RP set** as the slave entity for node-to-surface contact.
- **Contact assignment.** In the Interaction Manager, assign the node-to-surface contact. Being the wheel node as the slave and the deck surface as the master surface.

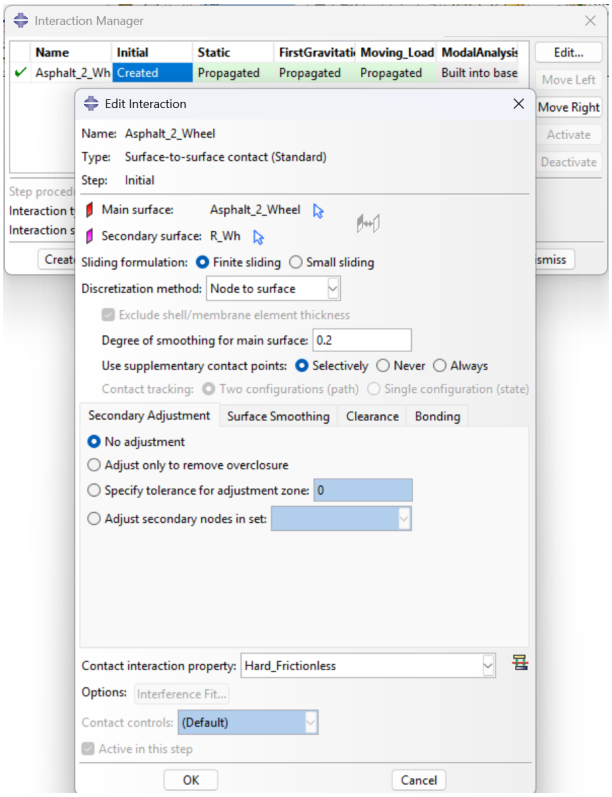
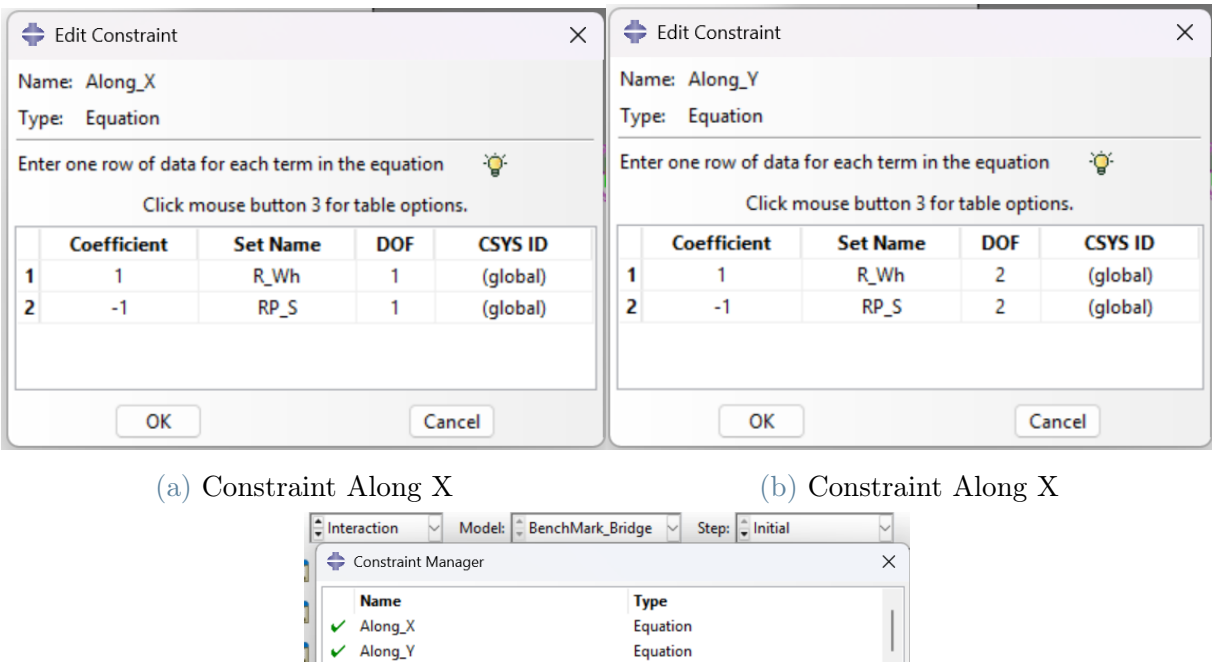


Figure C.5: Surface-to-Node Assignment

- **Kinematic constraints for stable tracking.** enforce in the horizontal plane between the wheel node and the Body Car:

  1. **Longitudinal tracking:** $U_1(\mathrm{RP}_W) = U_1(\mathrm{RP}_S)$,
  2. **Transverse tracking:** $U_2(\mathrm{RP}_W) = U_2(\mathrm{RP}_S)$,



(a) Constraint Along X (b) Constraint Along X

(c) Applied Constraints

Figure C.6: Kinematic Constraints.

while leaving the vertical direction free to allow suspension deflection and contact enforcement.

- **Self-weight application via concentrated forces.** Since body loads (gravity) cannot be applied to reference points carrying lumped mass, apply the vehicle weight through **concentrated forces** at RP_S.:

$$CF_3 = -m\,g$$

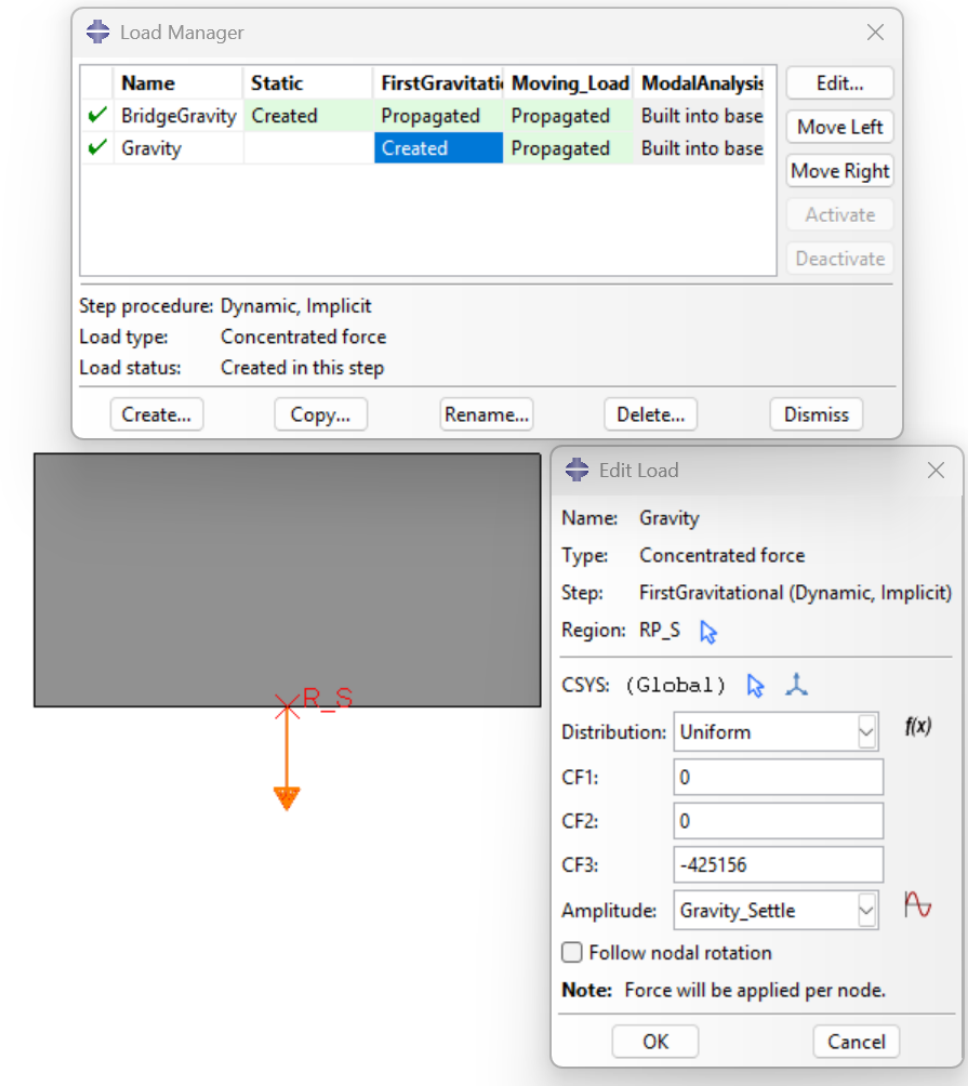


Figure C.7: Vehicle self weight application

- **Boundary conditions by step.**

  1. **Static step:** Fix vertical displacement of the wheel $U_3 = 0$ only in the static step since the contact is active when the dynamic analysis is performed. Then, Fix the lateral displacement $U_2 = 0$ and the rotations $U_{Ri} = 0$ of the reference sprung point from the static step to the final dynamic analysis keeping U_1 without constraint to then applied the velocity and U_3 free to allow vertical displacement.

  2. **Settlement step:** The boundary conditions are propagated from the static analysis

  3. **Crossing step:** prescribe the longitudinal motion in U_1.

- **Vehicle motion.** Implement the vehicle traversal through a prescribed longitudinal displacement at a constant speed:

$$U_1(t) = L\,\frac{t}{T}, \qquad T = \frac{L}{V}$$

where L is the travel distance and V the vehicle speed. In Abaqus, this is applied as a displacement boundary condition in U_1 with a tabular amplitude from $(0, 0)$ to $(T, 1)$.

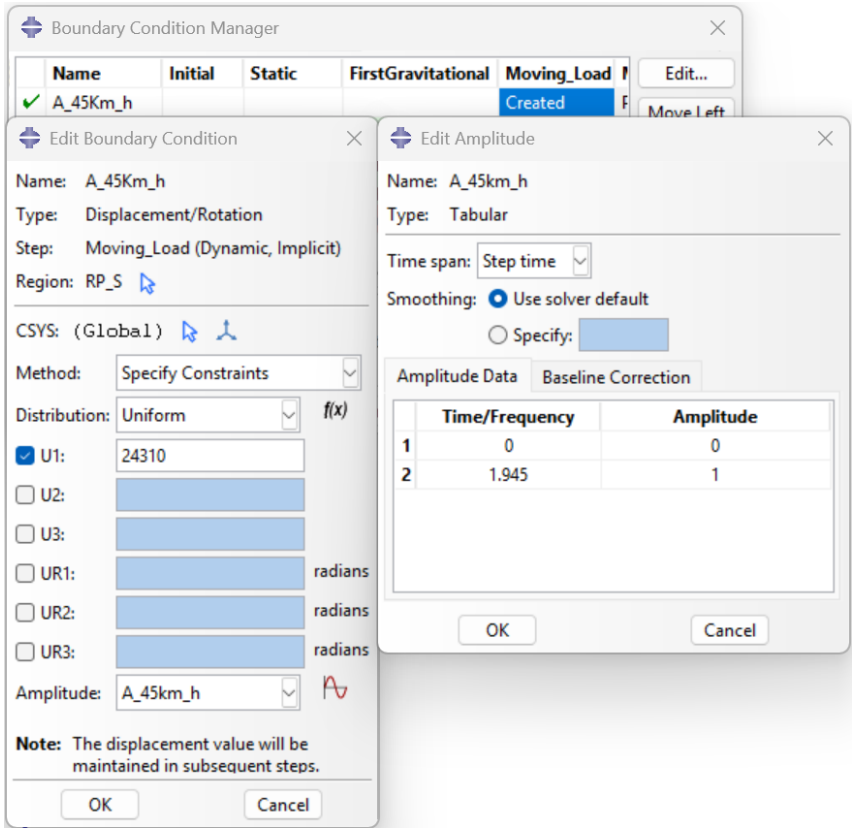


Figure C.8: Boundary Conditions to the Wheel and Sprung-Mass

List of Figures

List of Tables

Acknowledgements

This thesis represents the culmination of a long journey marked by study, dedication, love, gratitude, and inspiration. Behind these words stand the people who have consistently supported, guided, and encouraged me to persevere. It is to them that I owe my deepest thanks.

First and foremost, I would like to express my sincere gratitude to my advisor and co-advisors, Luca Martinelli and Tommaso Panigati, for giving me the opportunity to carry out this meaningful and enriching project. Through their guidance, I was able not only to learn new concepts, but also to deepen and internalize the knowledge acquired during my master's studies. Their constant availability, patience, and willingness to teach and clarify were fundamental throughout this process.

Secondly, I wish to thank Valentyna, a truly loyal, committed, trustworthy, and supportive woman, whose outstanding qualities are far too many to list here. From the day we met, she has taught me invaluable lessons and helped me become a better person. Thank you for sharing this beautiful and challenging journey with me that we consciously chose to undertake together. I would also like to express my sincere gratitude to her family, who played an important role in our formation, and especially to Doña Paolita, who welcomed me and treated me as a son.

Thirdly, I am deeply grateful to my family, who have supported me in countless ways throughout my life and have never let me down. My heartfelt thanks go to my mother, who guided and advised me night after night, often during our evenings watching soap operas together. To my brothers, who constantly push me to give my best — even though, in truth, they are my greatest inspiration. To my stepfather, who has always been present and supportive in our home. And, above all, to my grandparents, who played a fundamental role in raising me. Every time I think of them, I am moved to tears of happiness and admiration, and I find the strength to go even further.

Fourthly, I would like to thank my friends and colleagues, who have always taken care of me, not only during this academic journey but throughout my life. And For sure, I want to thank that guy, me, who dreamed to get a Master abroad and work to get it done.

Esta tesis representa la culminación de un largo camino marcado por el estudio, la dedicación, el amor, la gratitud y la inspiración. Detrás de estas palabras están las personas que, de manera constante, me han apoyado, guiado y animado a perseverar. A ellas les debo mi más profundo agradecimiento.

En primer lugar, quisiera expresar mi sincero agradecimiento a mi director y codirectores, Luca Martinelli y Tommaso Panigati, por brindarme la oportunidad de llevar a cabo este proyecto significativo y enriquecedor. Gracias a su orientación, pude no solo aprender nuevos conceptos, sino también profundizar e interiorizar los conocimientos adquiridos durante mis estudios de maestría. Su disponibilidad constante, paciencia y disposición para enseñar y aclarar dudas fueron fundamentales a lo largo de todo este proceso.

En segundo lugar, deseo agradecer a Valentyna, una mujer verdaderamente leal, comprometida, digna de confianza y solidaria, cuyas cualidades extraordinarias son demasiadas para enumerarlas aquí. Desde el día en que nos conocimos, me ha enseñado lecciones invaluables y me ha ayudado a convertirme en una mejor persona. Gracias por compartir conmigo este camino hermoso y desafiante que conscientemente decidimos emprender juntos. También quisiera expresar mi más sincero agradecimiento a su familia, que desempeñó un papel importante en nuestra formación no solo como profesionales, sino también como seres humanos, y especialmente a Doña Paolita, quien me recibió y me trató como a un hijo.

En tercer lugar, estoy profundamente agradecido con mi familia, que me ha apoyado de innumerables maneras a lo largo de mi vida y nunca me ha fallado. Mi más sentido agradecimiento es para mi madre, quien me guió y aconsejó noche tras noche, a menudo durante nuestras veladas viendo telenovelas juntos. A mis hermanos, que constantemente me impulsan a dar lo mejor de mí, aunque, en verdad, ellos son mi mayor inspiración. A mi padrastro, que siempre ha estado presente y brindando apoyo en nuestro hogar. Y, por encima de todo, a mis abuelos, quienes desempeñaron un papel fundamental en mi crianza. Cada vez que pienso en ellos, se me llenan los ojos de lágrimas de felicidad y admiración, y encuentro la fuerza para llegar aún más lejos.

En cuarto lugar, quisiera agradecer a mis amigos y colegas, quienes siempre han cuidado de mí, no solo durante este recorrido académico, sino a lo largo de toda mi vida. Y, por supuesto, quiero agradecerle a ese estudiante, a mí mismo, que soñó con obtener una maestría en el extranjero y trabajó para lograrlo.