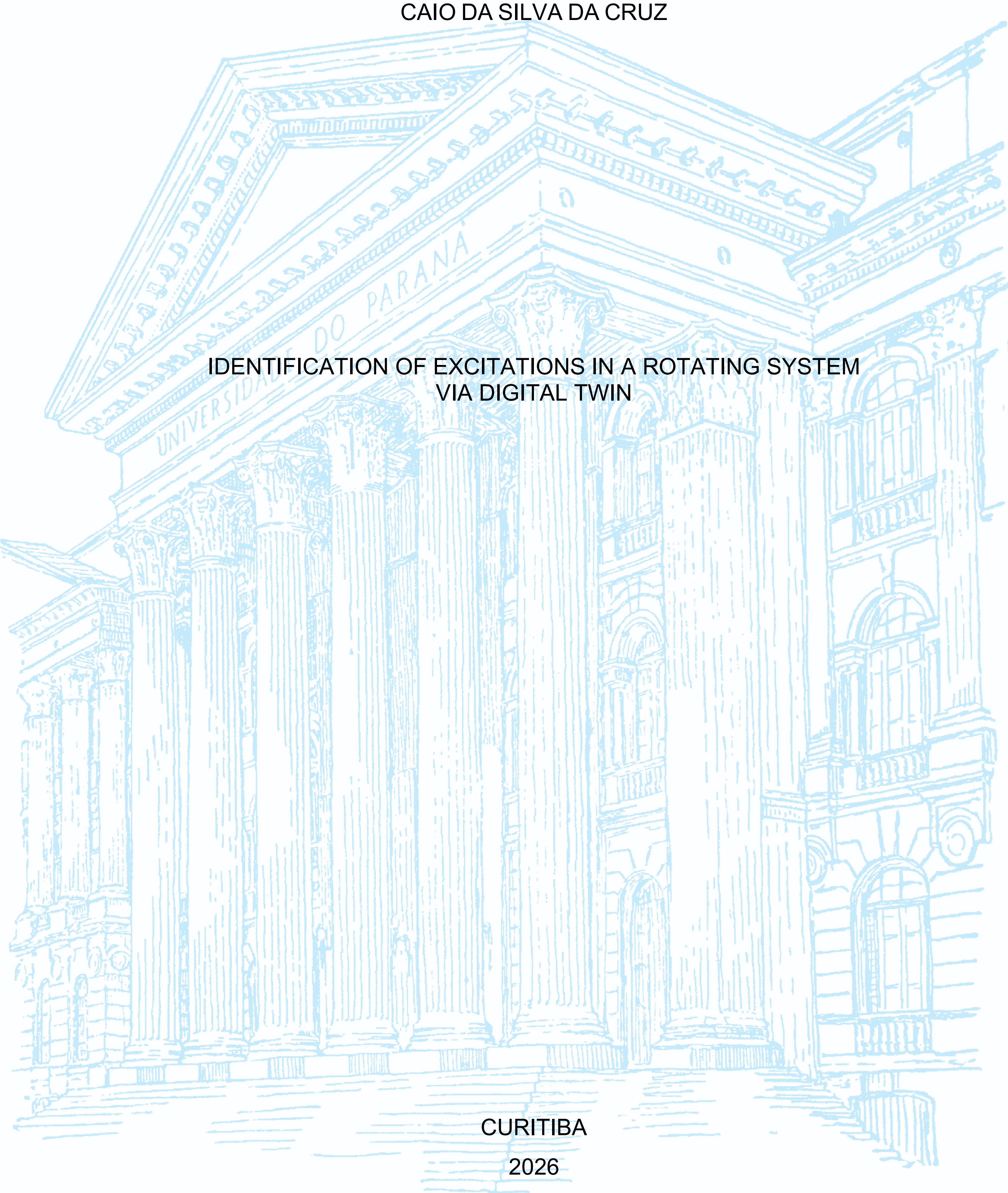
UNIVERSIDADE FEDERAL DO PARANÁ

CAIO DA SILVA DA CRUZ



IDENTIFICATION OF EXCITATIONS IN A ROTATING SYSTEM VIA DIGITAL TWIN

CURITIBA

2026

CAIO DA SILVA DA CRUZ

IDENTIFICATION OF EXCITATIONS IN A ROTATING SYSTEM VIA DIGITAL TWIN

Monograph submitted to PRH12/UFPR – Program in Exploration, Production, Processing, and New Materials in the Oil and Biofuels Industry, Technology Sector, at the Federal University of Paraná, as a partial requirement for the completion of the undergraduate research scholarship offered by the Human Resources Program of the National Agency of Petroleum, Natural Gas and Biofuels – PRH-ANP.

Supervisor: Prof. Dr. Eduardo Márcio de Oliveira Lopes

CURITIBA

2026

ACKNOWLEDGMENTS

I acknowledge the financial support of the Human Resources Program of the National Agency of Petroleum, Natural Gas and Biofuels – PRH-ANP, managed by the Foundation of Support to Research of the State of São Paulo (FAPESP), Brazil. Process no. 2025/17505-5, with special regard to PRH-ANP12/UFPR, Process no. 2025/02686-4.

ABSTRACT

Rotating machines such as motors, pumps, compressors, and turbines are essential to the oil, gas, and biofuels industry, where failures can cause severe economic, environmental, and safety impacts. Condition monitoring through vibration analysis has long been a key strategy to prevent catastrophic events and guide predictive maintenance. In this context, digital twins emerge as powerful tools, providing virtual representations of physical systems which evolve over time in a bidirectional communication. This study aims to develop a strategy to identify dynamic excitations caused by rotating mass imbalance for future use in digital twins. Since such excitations cannot usually be measured by direct means, they are computed by deconvolution from the vibratory responses observed in the physical system of concern, along with a proper system model. The current work employs a simplified rotating system, composed of a cantilever metallic beam clamped at one end and coupled to a DC motor with an unbalanced element at the free end. The force-identification procedure is based on an equivalent single-degree-of-freedom modal representation the parameters of which are obtained from experimental/numerical characterization of the employed system. The computational implementation performs signal filtering, frequency-domain analysis, impulse-response construction, and deconvolution-based force estimation. A finite element model is considered as a complementary numerical representation to support the interpretation of the system dynamics. Results for frequency ranges of concern are presented both numerically and graphically, in connection to the motor angular speed. They supply significant estimations of force amplitudes which would not be available otherwise. It is concluded that the proposed approach can contribute significantly to the predictive maintenance of rotating machinery via digital twins by extending the amount of valuable pieces of information at their disposal.

Keywords: Deconvolution; Force identification; Model updating; Rotating imbalance; Vibratory response.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AIAA – American Institute of Aeronautics and Astronautics
ANP – National Agency of Petroleum, Natural Gas and Biofuels
DC – Direct current
FFT – Fast Fourier Transform
FRF – Frequency Response Function
IFFT – Inverse Fast Fourier Transform
PRH – Human Resources Program
SDOF – Single Degree of Freedom

LIST OF SYMBOLS

M – mass matrix

C – damping matrix

K – stiffness matrix

x(t) – displacement vector in time domain

r(t) – modal coordinate vector in time domain

F(ω) – force vector in frequency domain

H(ω) – receptance matrix

m – number of vibration modes

ω – angular frequency

ζ – damping ratio

SUMMARY

1 INTRODUCTION

1.1 General Context

Rotating machines, such as electric motors, compressors, pumps, and turbines, are widely used in the oil, gas, and biofuels industry. These machines are systematically monitored to ensure that they remain in proper operating condition, especially with respect to safety and the prevention of sudden failures. As in many other types of industrial equipment and machinery, vibration analysis has long been an essential component of condition monitoring, supporting appropriate maintenance interventions.

Four types of maintenance strategies can be pointed out. They are:

- corrective maintenance: maintenance is performed after a failure occurs;
- preventive maintenance: maintenance is carried out after a predefined time interval;
- predictive maintenance: maintenance is based on simulation and condition-monitoring data;
- prescriptive maintenance: maintenance decisions are made based on simulations of multiple possible future scenarios.

Within the scope of PRH 12.1, efforts have been made to develop digital twins for rotating machinery in order to explore new possibilities for structural health monitoring through vibration analysis, with emphasis on predictive and prescriptive maintenance. The aim is to establish two-way communication between the virtual representation, or digital twin, and the physical system, or real twin, so that both can benefit from an interaction that evolves over time. These efforts have involved both computational simulations and experimental tests, with particular emphasis on expanding the capabilities of the digital twin of interest.

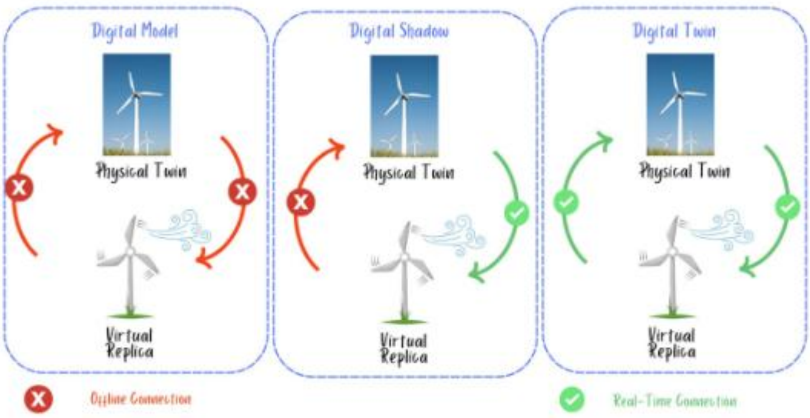
1.2 State of the Art

The use of vibration signals for condition monitoring of equipment and machinery has been extensively applied for a long time (GOLDMAN, 1999; TAYLOR, 2003; BRAUN, 2008; RANDALL, 2011). In industrial applications, real-time vibration information can be compared with historical data associated with a previous healthy condition in order to support the diagnosis of defects that are already present or slowly developing. This is possible because some defects have well-known vibration characteristics, allowing the defect to be categorized from the analysis of these characteristics (LEE, 2016). In the particular case of rotating machinery, this approach is considered one of the most effective ways to predict failures (SCHEFFER; GIRDHAR, 2004).

As a well-established approach for condition monitoring of rotating machinery (RANDALL, 2021), vibration analysis is concerned with the development of new sensors, such as inertial accelerometers, capable of recording vibrational signals in digital fashion, and of digital models, applicable to numerical simulations. Significant advances have been made in data acquisition technologies, including more accurate sensors, more robust databases, and increased computational power. Consequently, improved digitalization has progressed alongside the evolution of vibration-based maintenance analysis, as well as the incorporation of more advanced tools such as digital twins.

A digital twin is a high-fidelity virtual representation of a physical system, capable of capturing real-time data and generating accurate analyses of system behavior (TAO et al., 2022). The development of a digital twin begins with the creation of a digital shadow, through which information from the physical system is received. However, a characteristic that distinguishes a digital twin is the bidirectional real-time flow of information: the physical system provides inputs to the digital model, and the digital model provides outputs back to the physical system, as depicted in Figure 1. Without this bidirectional interaction, there is no true digital twin.

Figure 1 – Representation of digital model, digital shadow and digital twin.



Source: da Silva (2025).

In general, digital twin technology has applications in several areas, including agriculture, architecture and urban planning, engineering, and healthcare (CHAUDHARY et al., 2022; HARTMANN; AUWERAER, 2020). Tao et al. (2019, 2020, 2022) specifically address applications related to manufacturing, design, and maintenance of machines and equipment, respectively. Specific perspectives for the aerospace industry are provided by AIAA (2020), whereas broader state-of-the-art and future perspectives are presented by Kritzinger et al. (2018), Wagg et al. (2020), and Singh et al. (2021). The World Economic Forum recognizes this technology as a tool capable of supporting the oil and gas industry in achieving net-zero goals, given its perceived impact on sustainability (WORLD ECONOMIC FORUM, 2023).

Frequently, emphasis is placed on the acquisition, processing, and use of large amounts of measured and available data. However, beyond this objective, there are also important efforts directed toward obtaining high-quality data, collected through procedures that are well grounded in theory, in order to produce reliable and effective models (COVENEY et al., 2016; CHINESTA et al., 2020).

The development of a digital twin can be based on three types of models: data-driven models, physics-based models, and hybrid models that combine data-driven and physics-based approaches. Kapteyn and Willcox (2020) show that a specific data-informed model can be built from physics-based models representing several states of the real system, with updating performed by a machine learning technique. This approach is illustrated for an aircraft model, and sensing strategies associated with the performance of the machine learning technique are later investigated in greater depth by Kapteyn and Willcox (2022).

The effective use of sensors is also addressed by Moi et al. (2020), in which a digital twin is employed for condition monitoring of a knuckle boom crane. Zhou et al.

(2022) consider the fusion of data of several types in the case of a port crane. The use of machine learning, in turn, is highlighted by Yucesan and Viana (2020).

Of special relevance is the approach presented by Wagg et al. (2020) and Worden et al. (2020), due to their focus on dynamic engineering applications. Wagg et al. (2020) emphasize the synthesis of digital twins, exemplified by the case of a wind turbine. Worden et al. (2020), in turn, highlight the development of a robust mathematical basis for dynamic applications involving digital twins.

Condition monitoring, fault diagnosis, failure prediction, and predictive maintenance of rotating systems through digital twins are already a reality (JOHANSEN; NEJAD, 2019; WANG et al., 2022). However, an important gap remains: the identification of dynamic excitations acting on these systems based on measured vibrations. This identification problem has attracted attention for a long time (EWINS, 2009) and remains open to investigation due to its complexity. Advances in this direction may bring relevant benefits by expanding the representational capability of digital twins.

1.3 Objectives

1.3.1 General objective

The main objective of this work is to develop and evaluate a model-based strategy for identifying dynamic excitations caused by rotating mass imbalance in a simplified rotating system composed of a cantilever metallic beam coupled to a DC motor with an eccentric element. The proposed approach uses measured vibratory responses and numerical simulation to estimate excitation forces that cannot be directly measured, providing a contribution of concern towards future digital twin applications in rotating machinery.

1.3.2 Specific objectives

The specific objectives of this work are:

a) To assemble the experimental setup composed of a cantilever metallic beam, a DC motor, and an eccentric rotating element;

b) To acquire and process vibratory response data from the experimental setup using an impact hammer and an accelerometer;

c) To identify the dominant frequency components associated with the rotating imbalance excitation;

d) To develop a numerical model capable of representing the dynamic behavior of the simplified rotating system within the frequency range of interest;

e) To apply a time-domain deconvolution procedure, based on the impulse response of an equivalent modal model, to estimate the excitation forces from the measured acceleration responses;

f) To analyze the relationship between the estimated force amplitudes and the motor angular speed;

g) To discuss the potential use of the proposed force-identification strategy as a preliminary step towards future digital twin applications in rotating machinery.

1.4 Rationale

The oil and gas industry relies on highly complex rotating machinery, the reliability of which directly affects production efficiency, operational safety, and environmental risk mitigation. Unplanned failures in such systems may result in significant economic losses and operational disruptions, reinforcing the importance of developing even more robust monitoring and analysis approaches, as those associated to vibratory phenomena.

In recent years, the industrial sector has undergone an intense digital transformation, driven by the need for increased efficiency, sustainability, and improved decision-making processes. Technologies capable of integrating physical assets with digital environments have become increasingly relevant, especially in the context of predictive and prescriptive maintenance strategies.

Digital twin technology provides updated virtual representations of real systems in operation and allows maintenance strategies for these systems to be improved. The developed models are updated using data from the real systems, following their current operating condition. This creates new possibilities for system monitoring, with earlier indication of defects that may be developing and of components that may require repair or replacement. It should be emphasized, however, that each system of interest requires its own individual digital twin.

From a scientific perspective, challenges still exist in extracting valuable information from vibration data and in integrating experimental measurements with digital models in a reliable manner. Therefore, research efforts in that direction are essential to advance the application of digital twins in rotating machinery and to support future developments in intelligent maintenance systems.

Thus, it is relevant to investigate whether the dynamic excitations associated with measured vibrations in real physical systems, that is, the forces and/or moments causing the vibrations, can be accurately identified. Considering both linear and, eventually, nonlinear behavior, this identification may represent a significant advance in the diagnosis of the equipment and machinery of interest, making monitoring and maintenance actions more effective.

Finally, the application of digital twin technology may have a significant impact on the oil, gas, and biofuels industry, including contributions to cost reduction in the management of the corresponding industrial plants. It should also be noted that the application of this technology can deepen multidisciplinary cooperation among related fields of knowledge, including Mechanical Engineering, Electrical Engineering, and Computer Science.

2 THEORETICAL FRAMEWORK

Vibration analysis plays a fundamental role in mechanical engineering, enabling the understanding and control of dynamic phenomena that affect the operation and structural integrity of mechanical systems. In industrial applications, such as those in the oil and gas sector, where rotating equipment operates under extreme conditions, vibration analysis is essential for predictive maintenance and the prevention of catastrophic failures.

2.1 Theoretical Modal Analysis

The system of interest in this work is a cantilever beam with a DC motor containing an unbalanced element coupled to its free end. It can be discretely modeled as a linear system with multiple degrees of freedom (DOF) and proportional viscous damping (INMAN, 2013; RAO, 2018). The corresponding matrix equation of motion is

$$M\ddot{\boldsymbol{x}}(t) + C\dot{\boldsymbol{x}}(t) + K\boldsymbol{x}(t) = \boldsymbol{f}(t) \tag{1}$$

where M, C, and K are, respectively, the mass, viscous damping, and stiffness matrices, while the time-dependent acceleration, velocity, displacement, and force vectors are denoted accordingly, by $\ddot{\boldsymbol{x}}(t)$, $\dot{\boldsymbol{x}}(t)$, $\boldsymbol{x}(t)$ and $\mathbf{f}(t)$.

Considering proportional damping, one has

$$C = \alpha M + \beta K \tag{2}$$

where α and β are constants to be assigned based on experience or experiment. The M and K matrices are frequently obtained through the Finite Element Method.

Together with the initial conditions of the system, Eq. (1) can be solved by theoretical modal analysis (INMAN, 2013; RAO, 2018), in which a coupled set of linear ordinary differential equations in physical coordinates is transformed, through an eigenvalue and eigenvector problem, into a decoupled set of equations in modal coordinates. These decoupled equations are solved individually for the relevant excitation. Then, an inverse transformation is applied, allowing the solution in physical coordinates to be obtained.

In theoretical modal analysis, the modal parameters of the system, namely its natural frequencies, modal damping ratios, and mode shapes, are determined and used in the formulation of the equations in modal coordinates. These coordinates are related to the physical coordinates by

$$\boldsymbol{x}(t) = Tr(t) \tag{3}$$

where $\boldsymbol{x}(t)$ is the vector of physical coordinates, $r(t)$ is the vector of modal coordinates, and T is the mode-shape matrix, arranged by columns. Figure 2 illustrates the process for a two-degree-of-freedom system.

In the frequency domain, Eq. (1) can be written as

$$M\bar{X}_A(\omega) + C\bar{X}_V(\omega) + K\bar{X}(\omega) = \bar{F}(\omega) \tag{4}$$

where $\bar{X}_A(\omega)$, $\bar{X}_V(\omega)$, $\bar{X}(\omega)$ and $\bar{F}(\omega)$ are, respectively, the acceleration, velocity, displacement, and force vectors in the frequency domain. The overbar indicates a complex quantity.

From the properties of the Fourier transform, it follows that

$$M[-\omega^2\bar{X}(\omega)] + C[i\omega\bar{X}(\omega)] + K\bar{X}(\omega) = (-\omega^2 M + i\omega C + K)\bar{X}(\omega) = \bar{F}(\omega) \qquad (5)$$

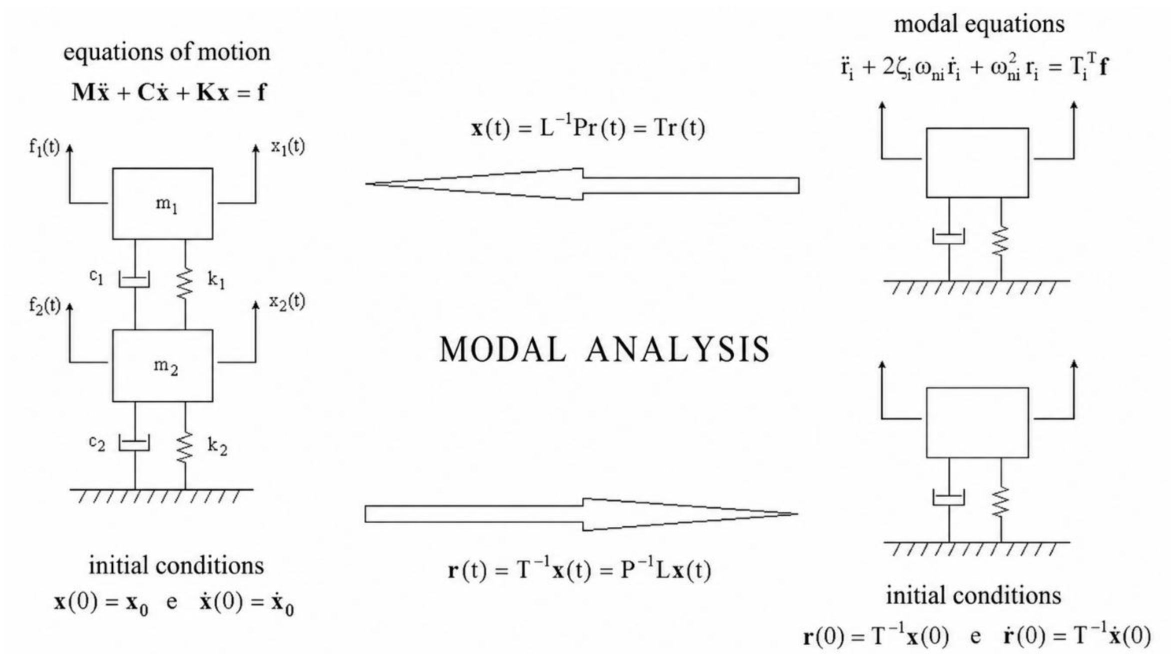
therefore

$$\bar{X}(\omega) = \bar{H}(\omega)\bar{F}(\omega) \qquad (6)$$

where

$$\bar{H}(\omega) = (-\omega^2 M + i\omega C + K)^{-1} \qquad (7)$$

Figure 2 – Representation of theoretical modal analysis in a two-degree-of-freedom system.



Source: adapted from Lopes (2025).

The matrix $\bar{H}(\omega)$ is known as the receptance matrix and relates the displacement vector to the force vector in the frequency domain. Each element of this matrix associates the displacement at a physical coordinate with the force applied at the same or at another physical coordinate. Thus, the element located in the s-th row and r-th column, $\bar{H}_{sr}(\omega)$, expresses the relationship between the displacement, or vibration, at coordinate/point s, $\bar{X}_s(\omega)$, and the force, or excitation, applied at coordinate/point r, $\bar{F}_r(\omega)$. That is,

$$\bar{H}_{sr}(\omega) = \frac{\bar{X}_s(\omega)}{\bar{F}_r}(\omega) \qquad (8)$$

which is a particular receptance (frequency response function or FRF) of the system of interest.

If the modal parameters of the system are known from modal analysis, either theoretical or, as discussed below, experimental, the receptance matrix $\bar{H}(\omega)$ can be expressed as

$$\bar{H}(\omega) = T\, diag \left[\ddots \frac{1}{\left(\omega_{nj}^2 - \omega^2\right) + i\left(2\zeta_j \omega_{nj} \omega\right)} \ddots \right] \mathrm{T}^T \tag{9}$$

while each associated receptance $\bar{H}_{sr}(\omega)$ can be given by

$$\bar{H}_{sr}(\omega) = \sum_{j=1}^{m} \left\{ \frac{\left[u_j u_j^T\right]_{sr}}{\left(\omega_{nj}^2 - \omega^2\right) + i\left(2\zeta_j \omega_{nj} \omega\right)} \right\} \tag{10}$$

where m is the number of vibration modes of the system, while ω_{nj} represents the natural frequencies, ζ_j the damping ratios and u_j the mode shapes in column-vector form. In Eq. (10), each term in the summation in the expression $\bar{H}_{sr}(\omega)$ represents the specific contribution of a given vibration mode to the relationship between the vibration $\bar{X}_s(\omega)$ and the excitation $\bar{F}_r(\omega)$.

The term $\left[\mathrm{u}_j \mathrm{u}_j^T\right]$ is an outer product and, therefore, constitutes a matrix. Thus, since $\left[\mathrm{u}_j \mathrm{u}_j^T\right]_{sr} = u_{sj} u_{rj}$, one can define $u_{sj} u_{rj} = (A_{sr})_j$, where $(A_{sr})_j$ is referred to as a modal constant. Substituting this definition into Eq. (10), resulting in:

$$\bar{H}_{sr}(\omega) = \left[\sum_{j=1}^{m} \frac{(A_{sr})_j}{\left(\omega_{nj}^2 - \omega^2\right) + i\left(2\zeta_j \omega_{nj} \omega\right)} \right] \tag{11}$$

By applying the inverse Fourier transform to Eq. (11), the impulse response function (IRF) that relates to the displacement, or vibration, $x(t)$ at coordinate s to the force, or excitation, at coordinate r can be obtained. This impulse response function is given by (BRENNAN and TANG, 2023)

$$h_{sr}(t) = \sum_{j=1}^{m} \frac{(A_{sr})_j}{m_j \omega_{dj}}\, e^{-\zeta_j \omega_{nj} t} sen\left(\omega_{dj} t\right) \tag{12}$$

where m_j is the modal mass and $\omega_{dj} = \omega_{nj}\sqrt{1 - \zeta_j^2}$ the damped natural frequency, which is approximately equal to the natural frequency for small damping-ratio values. Each term in Eq. (12) corresponds to the impulse response associated with a vibration mode.

For a mechanical system with well-separated vibration modes, the dynamic behavior in the neighborhood of each mode can be approximated as the behavior of a single-degree-of-freedom system. This approximation is adopted in the present work for the first vibration mode of the system of interest.

2.2 Experimental Modal Analysis

If receptances of the system of interest are obtained experimentally (EWINS, 2009), the modal parameters can be estimated through experimental modal analysis. Several algorithms are available for this estimation, ranging from simple to more elaborate methods (MAIA; SILVA, 1997). For example, in a system with well-separated natural frequencies, any receptance in the neighborhood of a natural frequency is dominated by a single term, namely the term associated with that natural frequency.

Thus, for low damping, the natural frequencies can be estimated from the peaks, or maxima, of the receptance magnitude, whereas the damping ratios can be estimated from the -3 dB, or half-power, bandwidths around the peaks (EWINS, 2009). The mode shapes can also be estimated if one row or column of the receptance matrix is available.

Time-domain functions can be numerically estimated from experimental data. From Eq. (8), previously presented, it follows that

$$\bar{X}_s(\omega) = \bar{H}_{sr}(\omega)\bar{F}_r(\omega) \tag{13}$$

$$\bar{F}_r(\omega) = \frac{\bar{X}_s(\omega)}{\bar{H}_{sr}(\omega)} \tag{14}$$

By applying the inverse Fourier transform to Eqs. (13) and (14), the associated displacement, or vibration,$x(t)$, and force, or excitation,$f(t)$, are obtained in the time domain, such that

$$x_s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \bar{X}_s(\omega)\, d\,\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \bar{H}_{sr}(\omega)\bar{F}_r(\omega)\, d\,\omega \tag{15}$$

$$f_r(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \bar{F}_r(\omega)\, d\,\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{\bar{X}_s(\omega)}{\bar{H}_{sr}(\omega)}\, d\,\omega \tag{16}$$

The operations indicated above correspond, respectively, to convolution and deconvolution (HEATON, 2024). Numerically, Eqs. (15) and (16) can be computed using the inverse fast Fourier transform (IFFT). Alternative algorithms can also be found in Heaton (2024).

If the excitation acting on a system results from imbalance, the excitation amplitude, in N, is known to vary with the square of the angular frequency, in rad/s (RAO, 2018). This relationship is considered in the present work, as discussed later.

3 MATERIALS AND METHODS

3.1 Experimental System

The experimental system used in this work consisted of a metallic cantilever beam coupled to a DC motor mounted at the free end. The motor included an eccentric rotating element, which generated a harmonic excitation due to rotating mass imbalance. The clamped end of the beam was attached to a steel block, to approximate a fixed boundary condition.

The choice of a cantilever beam with a motor at the free end was motivated by the need for a simplified and controlled laboratory system capable of reproducing the main dynamic characteristics of a rotating system under imbalance excitation. Although simpler than industrial rotating machinery, this setup allowed the investigation of the relationship between measured vibration responses and the dynamic excitation forces that generated them.

3.2 Beam and Motor Parameters

The numerical model considered a metallic beam with a rectangular cross section. The beam dimensions used in the computational model were 602.0 mm long, 49.8 mm wide, and 6.32 mm thick. The longitudinal modulus of elasticity was taken as 200 GPa, and the density was defined as 7850 kg/m^3, which are consistent with steel properties. The numerical model also included a concentrated mass of 0.245 kg at the free end, representing the motor assembly.

The beam parameters used in the numerical model are summarized in Table 1.

Table 1 – Beam and motor parameters.

Parameter	Symbol	Value
Beam length	L	602.0 mm
Beam width	b	49.8 mm
Beam thickness	h	6.32 mm
Young’s modulus	E	200 GPa
Density	ρ	7850 kg/m^3
Number of finite elements	-	10
Motor mass	m_c	0.245 kg

Source: The author (2026).

3.3 Finite Element Model

A finite element model of the beam-motor system was developed to represent the dynamic behavior of the structure. The beam was discretized into 10 finite elements. Each node contained two degrees of freedom: transverse displacement and rotation. The global mass and stiffness matrices were assembled from the elemental beam matrices.

The second moment of area of the rectangular cross section was calculated from the beam width and thickness, while the mass per unit length was obtained from the material density and cross-sectional area. After assembling the global matrices, boundary conditions were applied to represent the clamped end.

Instead of using a perfectly rigid clamp, the numerical implementation included translational and rotational root springs. This approach allowed the model to account for a non-ideal support condition. A concentrated inertia was also added at the beam tip to represent the motor assembly.

After the assembly of the reduced mass and stiffness matrices, the undamped modal problem was solved. A Guyan reduction procedure was applied to reduce the model size, and the eigenvalue problem was solved to obtain the natural frequencies and mode shapes of the beam-motor system. The resulting modal information was used to support the understanding of the dynamic behavior of the system and to provide a numerical representation of the experimental structure.

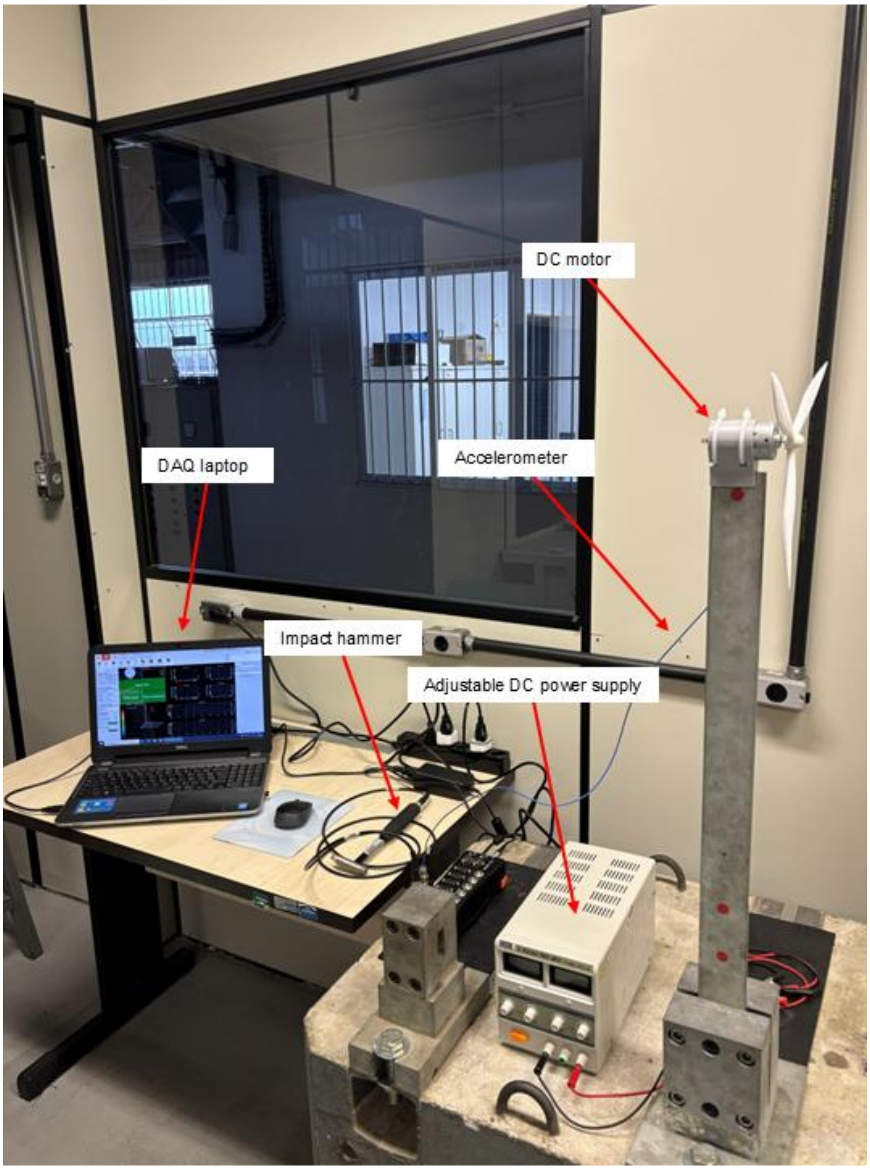
3.4 Modal Characterization

An impact test was used to obtain experimental dynamic information from the beam-motor system. In this test, the structure was excited by an impact hammer, which also acquired the force signal by its associated transducer, and the corresponding acceleration response was measured by an accelerometer. Both signals were processed by a digital signal analyzer interfaced to a notebook. The data from the impact test were used to estimate modal parameters associated with the frequency range of interest.

For the force-identification procedure, the beam-motor system was represented by an equivalent single-degree-of-freedom modal model. The modal parameters adopted in the deconvolution algorithm were a natural frequency of 10.23 Hz, a damping ratio of 0.003, and a modal constant equal to 1.56. These parameters were used to construct the impulse response function employed in the deconvolution process.

It is important to distinguish the role of the two numerical representations used in this work. The finite element model was used to represent the structural dynamic behavior of the beam-motor assembly and to support modal analysis. The deconvolution procedure, however, was performed using a reduced equivalent modal model based on experimentally identified parameters.

Figure 3 – Experimental setup used for the vibration tests.



Source: The author (2026).

3.5 Operational Test under Rotating Imbalance

After the modal characterization, an operational test was performed with the DC motor in operation. The eccentric element attached to the motor generated rotating imbalance forces, which produced vibration in the investigated system.

The measured response consisted of acceleration as a function of time. For acquiring the response, it was used an accelerometer positioned at the free end of beam. Its signal was processed by a digital signal analyzer.

The experimental data used for the operational analysis was obtained from an Excel file produced by the signal-analyzer software. The file contained time and acceleration columns read from the spreadsheet. The sampling frequency adopted in the analysis was 100 Hz, corresponding to a sampling interval of 0.01 s.

Three-time intervals were selected from the measured acceleration signal, corresponding to three different operating conditions of the motor. The selected intervals are specified in Table 2.

Table 2 – Time segments and rotation conditions.

Segment	Data index range	Description
Segment 1	9000 - 12000	First rotation condition
Segment 2	16000 - 19000	Second rotation condition
Segment 3	23000 - 26000	Third rotation condition

Source: The author (2026).

3.6 Preliminary Processing Steps

The measured acceleration signal was submitted to some preliminary processing steps before the actual force-identification procedure. Firstly, as mentioned above, three selected time segments were extracted from the complete acceleration signal. Then, a ninth-order Butterworth band-pass filter was applied to each segment. The adopted frequency band was 10 Hz to 20 Hz, which contains the frequency range of interest for the investigated imbalance excitation.

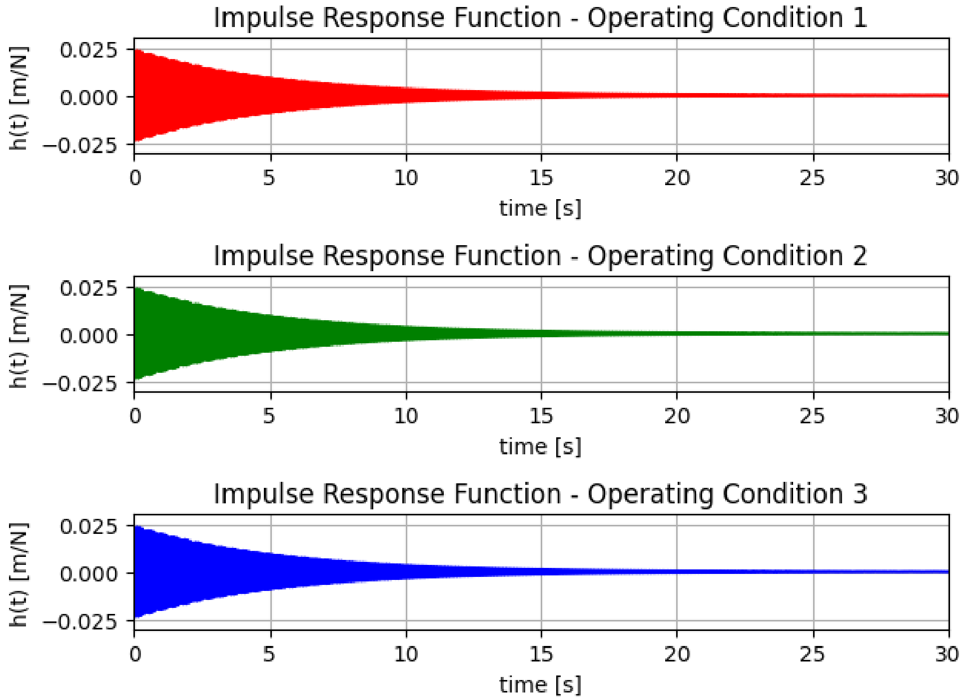
After filtering, the fast Fourier transform (FFT) was applied to each acceleration segment in order to analyze the frequency content of the measured responses. The frequency-domain representation allowed the dominant excitation frequencies to be identified. In the deconvolution code, the frequencies of interest were then taken as approximately 13.93 Hz, 15.67 Hz, and 18.00 Hz for the three selected motor operating conditions.

3.7 Deconvolution-Based Force Identification

The excitation force generated by the rotating imbalance cannot be measured directly. Therefore, a deconvolution-based procedure was applied to estimate the force from the measured acceleration response and the dynamic model of the system.

An equivalent single-degree-of-freedom model was used to generate the system impulse response function. It is given by Eq. (12), applied to a single-degree-of-freedom system. Figure 4 shows that, regardless of the excitation force, the impulse response function is the same because it depends on the system parameters and not on the excitation force.

Figure 4 – Impulse response functions of the investigated system.



Source: The author (2026).

For each selected time segment, a matrix was constructed from circularly shifted versions of the impulse response function. The force vector was then estimated by solving the inverse problem given by

$$[R] \cdot \{f\} = \{a\} \tag{17}$$

where $[R]$ is the matrix associated with the impulse response function, $\{f\}$ is the unknown force vector, and $\{a\}$ is the measured acceleration response. In the implemented approach, the force was obtained by direct matrix inversion. After the force estimation, the same 10–20 Hz band-pass filter was applied to the reconstructed force signal.

The Fourier transform was then applied to the estimated force signals, allowing the force amplitudes to be evaluated in the frequency domain. The force amplitudes used for the final comparison were 29.47 N, 47.77 N, and 81.38 N, associated with angular frequencies corresponding to 13.93 Hz, 15.67 Hz, and 18.00 Hz, respectively.

3.8 Computational Implementation

The numerical procedures were implemented using MATLAB and Python. MATLAB was used in the original deconvolution routine for reading the experimental data, filtering the acceleration signals, computing the Fourier transforms, generating the impulse response functions, constructing the deconvolution matrices, and estimating the excitation forces. Python was used for numerical modeling of the beam-motor system, including finite element assembly, Guyan reduction, modal analysis, and visualization of mode shapes.

The computational workflow can be summarized as follows:

1. acquisition of acceleration data from the experimental system;
2. selection of three-time intervals associated with different motor operating conditions;
3. band-pass filtering between 10 Hz and 20 Hz;
4. frequency-domain analysis of the measured acceleration;
5. definition of modal parameters for the equivalent SDOF model;
6. construction of the impulse response functions;
7. deconvolution-based estimation of the excitation force;
8. frequency-domain analysis of the reconstructed force;
9. comparison between force amplitude and motor angular speed.

No statistical hypothesis tests were applied, since the main objective of the methodology was deterministic signal processing and force reconstruction from measured vibration data.

4 PRESENTATION AND DISCUSSION OF RESULTS

This chapter presents the results obtained from the numerical and experimental procedures described in Chapter 3. First, the natural frequencies of the beam-motor system obtained from the finite element model are related. Then, the experimental modal analysis resulting from the impact test is presented. Finally, the acceleration responses measured during motor operation are analyzed, and the excitation forces caused by rotating imbalance are estimated using the deconvolution-based procedure.

4.1 Finite Element Model Natural Frequencies

The finite element model was used to estimate the natural frequencies and mode shapes of the beam-motor system. The beam was modeled using 10 finite elements, with two degrees of freedom per node: transverse displacement and rotation. The numerical model considered a cantilever metallic beam with rectangular cross section, root stiffness terms to represent the clamped boundary condition, and concentrated mass at the free end to represent the motor assembly. Further details are found in section 3.2. The natural frequencies are shown in Table 3.

Table 3 – Natural frequencies from the finite element model.

Mode	Natural frequency (Hz)
1	16.319
2	69.395
3	206.765
4	418.446
5	708.552

Source: The author (2026).

The first natural frequency obtained from the finite element model was higher than the experimentally identified frequency used in the force-identification stage. This difference indicates that the numerical model captures the general dynamic behavior of the beam-motor system, but still presents deviations with respect to the real experimental setup. These deviations may be associated with uncertainties in the boundary condition, motor attachment, concentrated mass representation and flexibility of the clamping system.

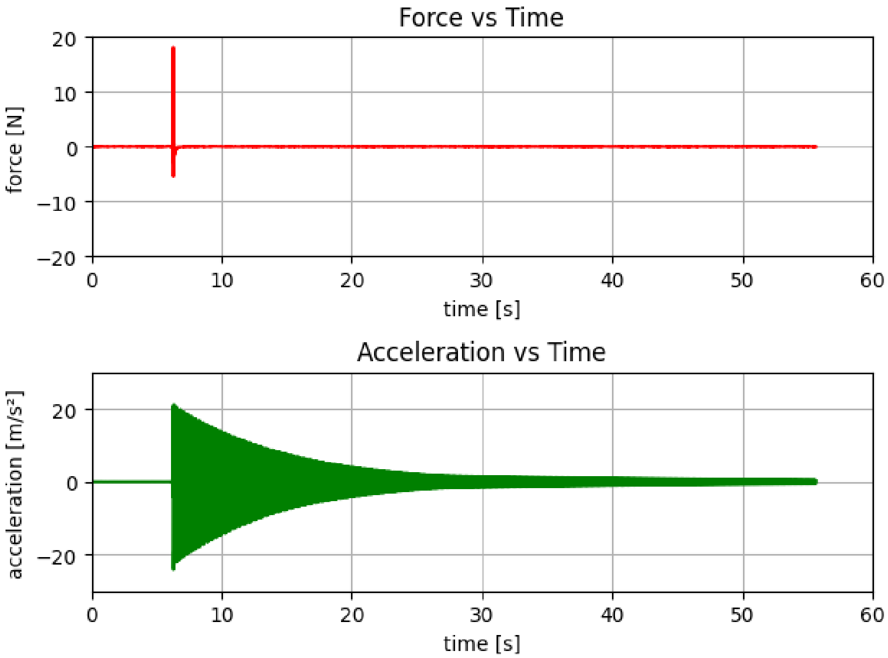
4.2 Experimental Modal Analysis

The experimental modal characterization was performed using impact-test data. The structure was excited by an impact hammer, which contained a force sensor, and the corresponding acceleration response was measured with an accelerometer. The corresponding signals were processed in a digital signal analyzer, as reported in section 3.4.

The resulting frequency response function (FRF), in this case inertance (acceleration over force in the frequency domain), was converted to receptance (displacement over force in the frequency domain). The receptance was then used to identify the relevant modal parameters of the beam-motor system in the frequency range of interest.

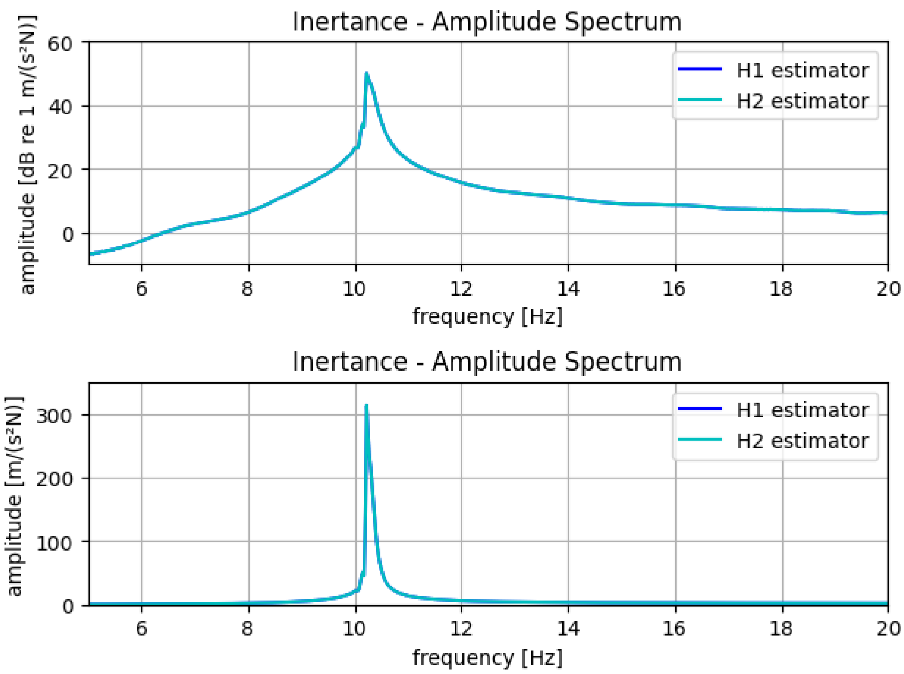
Figure 5 shows the force and acceleration signals from the impact test. Figure 6 presents the corresponding inertance amplitude spectrum, both in dB and linear scales. It is noted in Figure 6 that the H1 and H2 estimators coincide.

Figure 5 – Force and acceleration signals from impact test.



Source: The author (2026).

Figure 6 – Inertance amplitude spectrum.



Source: The author (2026).

For the deconvolution procedure, the system was represented by an equivalent single-degree-of-freedom modal model. The adopted modal parameters are contained in Table 4. These parameters were used to construct the impulse response function adopted in the force-identification algorithm.

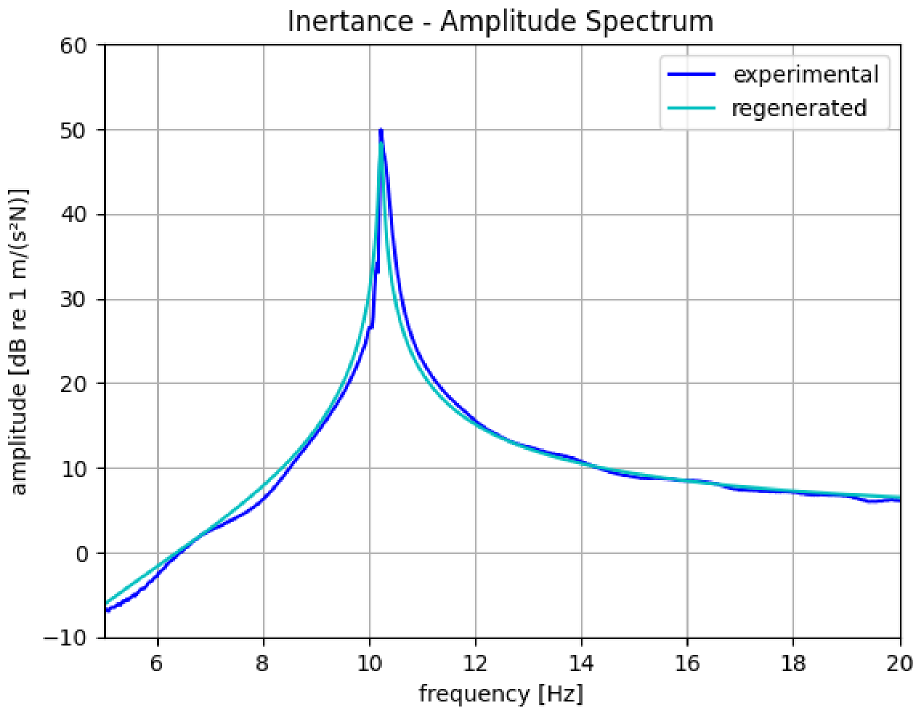
The use of an experimentally adjusted modal model was necessary because the deconvolution procedure depends directly on the dynamic relationship between force and measured response. The comparison between the experimental and regenerated inertance amplitude spectra in the frequency range of interest is shown in Figure 7. It can be visually observed that they are very close.

Table 4 – Modal parameters for the deconvolution procedure.

Parameter	Symbol	Value
Natural frequency	f_n	10.23 Hz
Damping ratio	ζ	0.003
Modal constant	A_1	1.56
Equivalent modal mass	$m = 1/A_1$	0.641 kg

Source: The author (2026).

Figure 7 – Comparison between experimental and regenerated inertance amplitude spectra.



Source: The author (2026).

4.3 Acceleration Responses under Rotating Imbalance

The operational vibration response was acquired with the DC motor in operation. The eccentric rotating element generated imbalance forces, which excited the cantilever beam and produced acceleration responses measured by the accelerometer. The experimental setup and further details are shown in section 3.5.

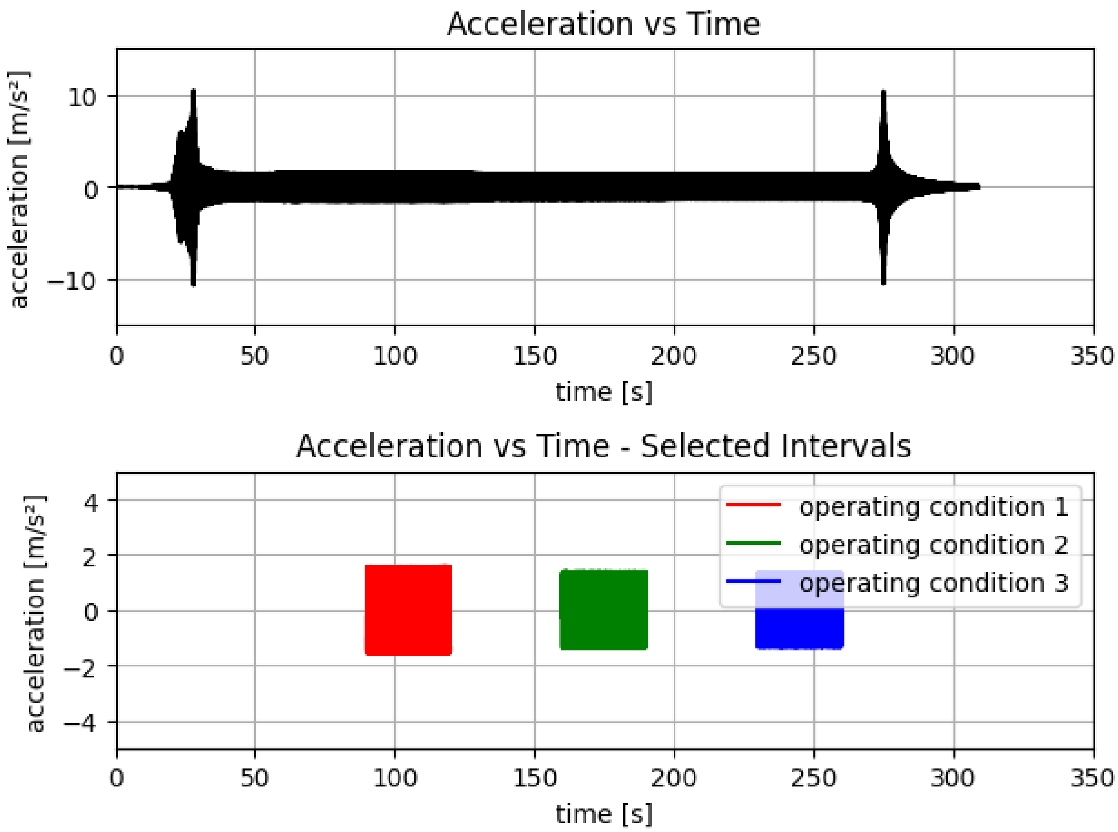
Three-time segments were selected from the complete acceleration signal. These segments corresponded to different motor operating conditions, as detailed in Table 5, and were used separately in the force-identification procedure. Figure 8 shows both the full acceleration signal and the aforementioned time segments.

Table 5 – Selected signal segments for operational analysis.

Segment	Index range	Description
1	9000–12000	First operating condition
2	16000–19000	Second operating condition
3	23000–26000	Third operating condition

Source: The author (2026).

Figure 8 – Full acceleration signal and selected time segments.



Source: The author (2026).

The full acceleration signal in Figure 8 shows variations associated with motor operation over time. Resonance events related to the first vibration mode of the investigated system are associated with high acceleration-amplitude values during both motor start-up and shut-down.

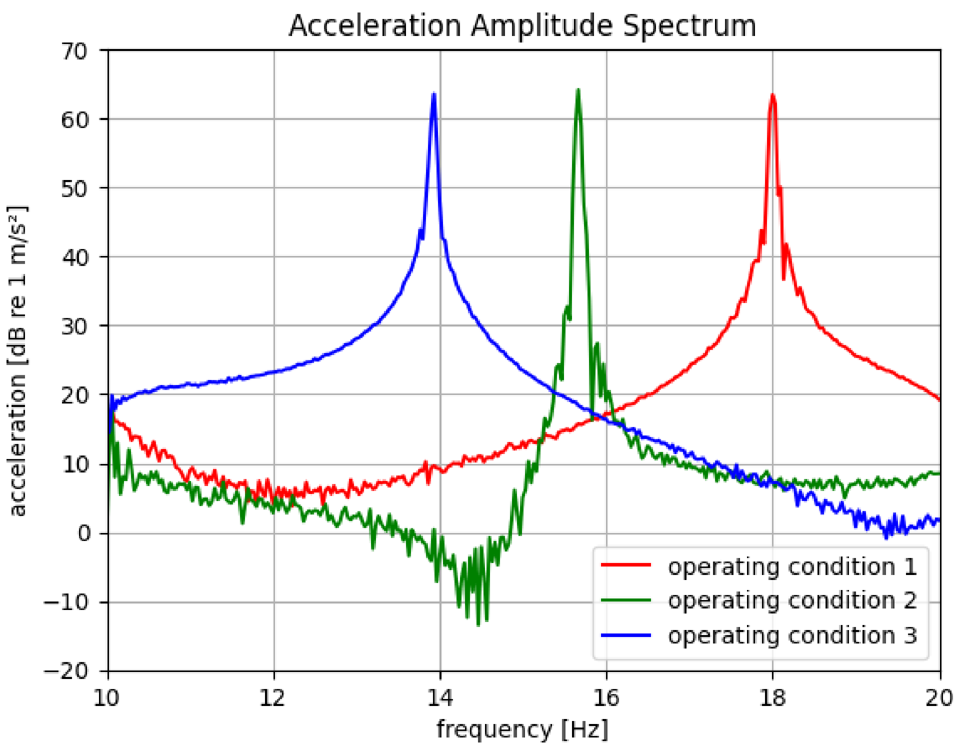
The selected time segments were used because they represented portions of the signal suitable for frequency-domain analysis and deconvolution. It is mentioned that before applying the force-identification procedure, the signals were filtered using a ninth-order Butterworth band-pass filter between 10 Hz and 20 Hz, which is the frequency range of concern.

4.4 Frequency-Domain Analysis of the Measured Responses

The fast Fourier transform (FFT) was applied to the filtered acceleration signals in order to identify the dominant frequency components associated with the rotating imbalance excitation. The frequency range between 10 Hz and 20 Hz was selected because it contained the main response components of interest for the analyzed motor operating conditions.

Figure 9 contains the acceleration amplitude spectra for the three selected operating conditions. The dominant frequencies identified in the corresponding segments are related in Table 6.

Figure 9 – Acceleration amplitude spectra of operating conditions.



Source: The author (2026).

Table 6 – Dominant frequencies in operational acceleration responses.

Segment	Dominant frequency (Hz)	Angular frequency (rad/s)
1	18.00	113.10
2	15.67	98.46
3	13.92	87.52

Source: The author (2026).

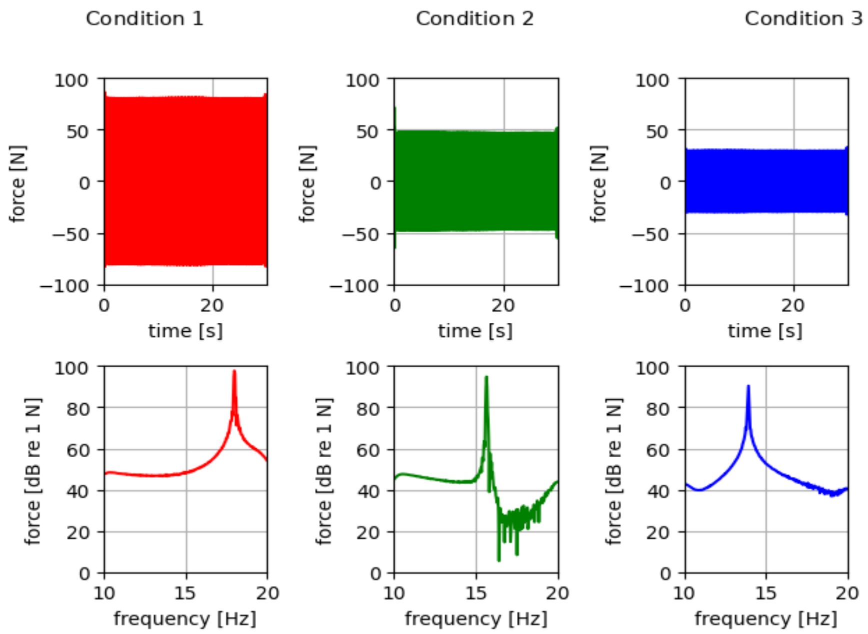
As previously noticed, the selected segments correspond to different motor operating conditions. By experimental convenience, these operating conditions were selected in decreasing order in terms of frequency. Since rotating imbalance generates excitation at the rotational frequency, these frequency components are directly associated with the dynamic force generated by the eccentric mass at the tip of the motor.

4.5 Deconvolution-Based Force Identification

The excitation force cannot be measured directly during the operational test. Therefore, a deconvolution-based procedure was applied to estimate the force from the measured acceleration response and the equivalent modal model.

The reconstructed force signals represent the main result of the proposed deconvolution procedure, since they correspond to the estimated dynamic excitation required to generate the measured acceleration responses. The time-domain reconstructed forces and their associated amplitude spectra are shown in Figure 10. They correspond to the operating conditions already mentioned above.

Figure 10 – Time-domain reconstructed forces and associated amplitude spectra.



Source: The author (2026).

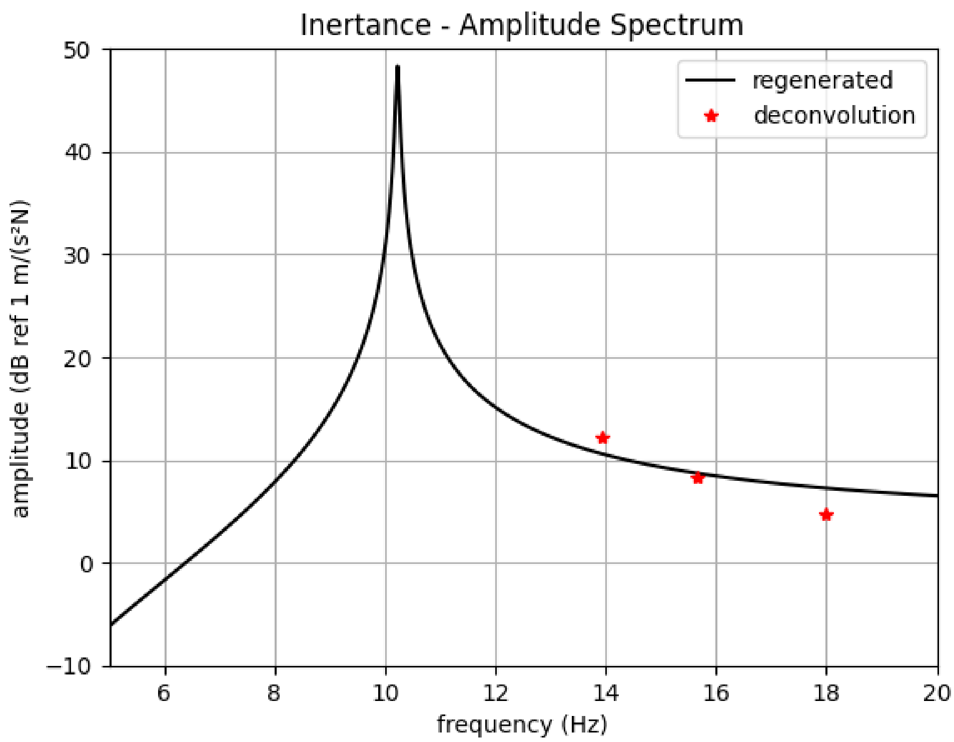
The estimated force amplitudes obtained for the three analyzed operating conditions are shown in Table 7. Figure 11 presents a comparison between the regenerated amplitude spectrum and the point inertance amplitudes calculated from the estimated force amplitudes and the corresponding acceleration amplitudes.

Table 7 – Estimated excitation force amplitudes.

Segment	Frequency (Hz)	Angular frequency (rad/s)	Estimated force amplitude (N)
1	18.00	113.10	81.38
2	15.67	98.46	47.77
3	13.92	87.52	29.47

Source: The author (2026).

Figure 11 – Regenerated inertance amplitude spectrum and point inertance amplitudes.



Source: The author (2026).

It is observed from Table 7 that when the identified operating points are ordered by increasing angular speed, the estimated force amplitudes increase. This trend is consistent with the expected behavior of rotating mass imbalance, for which the force amplitude is proportional to the square of angular speed, as detailed in the next section.

4.6 Relationship Between Force Amplitude and Angular Speed

For a rotating eccentric mass, the theoretical amplitude of the associated imbalance force amplitude is given by

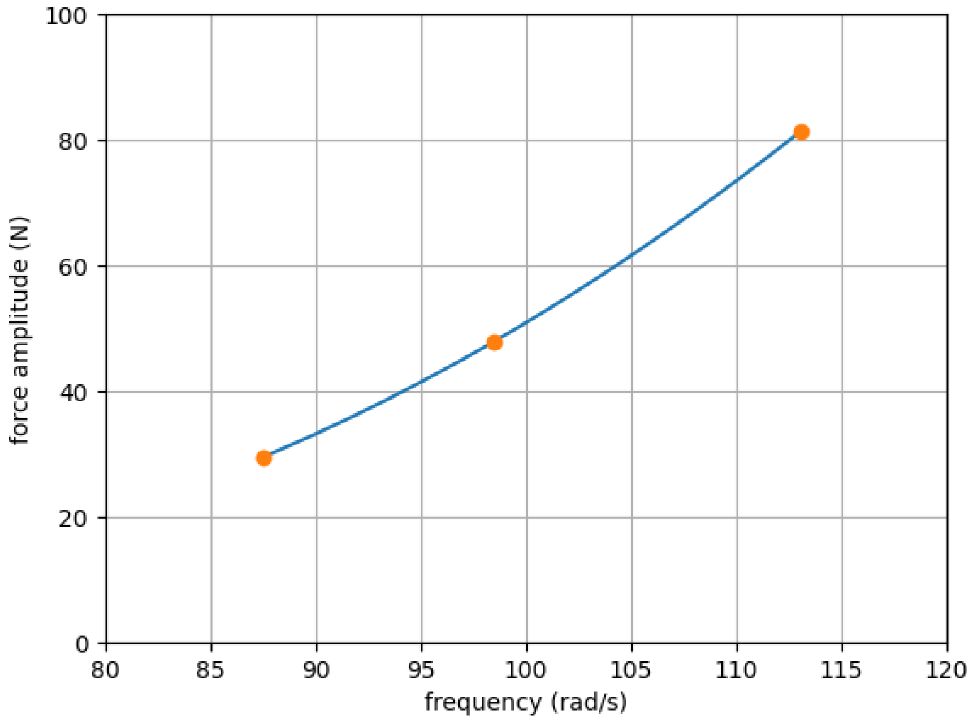
$$F_0 = m_e e \omega^2 \qquad (18)$$

where m_e is the eccentric mass, e is the eccentricity, and ω is the angular speed. Therefore, the expected trend is a quadratic increase in force amplitude with angular speed.

As shown in Figure 12, the identified force amplitudes followed this expected tendency. Between the first and third operating conditions, the angular frequency increased from approximately 87.52 rad/s to 113.10 rad/s, while the estimated force amplitude increased from 29.47 N to 81.38 N. This result indicates that the deconvolution procedure was able to capture a physically coherent relationship between motor speed and excitation force amplitude.

A fitted force-speed curve, also represented in Figure 12, may be used as an approximate representation of the imbalance excitation in the investigated frequency range. However, the interpretation of the absolute force values requires caution, since the estimation depends on the accuracy of the modal model, the filtering procedure, and the conditioning of the deconvolution matrix.

Figure 12 – Identified force amplitude versus angular frequency.



Source: The author (2026).

Since only three operating points were analyzed, the fitted curve should be interpreted as a preliminary indication of the expected quadratic trend rather than as a general calibrated imbalance model. Additional operating speeds and independent validation measurements would be required to improve the robustness of the force-speed relationship.

4.7 Applicable Limitations

Although the obtained results are coherent with the expected behavior of rotating imbalance, some limitations should be considered. First, the excitation force was not directly measured, which prevents direct experimental validation of the reconstructed force amplitudes. The force values should therefore be interpreted as identified quantities obtained from the adopted model and signal-processing procedure.

Second, the deconvolution algorithm used an equivalent single-degree-of-freedom modal model. This simplification is suitable for a preliminary analysis in a limited frequency range, but it does not represent the full dynamic behavior of the beam-motor assembly. Higher modes, nonlinearities, uncertainties in the clamping condition, and motor-structure interaction may affect the measured response.

Third, the deconvolution procedure is sensitive to noise and to the dynamic model used in the inversion process. Errors in the modal parameters may lead to errors in the reconstructed force. For this reason, the selected frequency range, filtering strategy, and modal parameter identification are essential steps in the methodology.

Finally, the results do not represent the implementation of a complete digital twin. The present work provides a force-identification strategy based on experimental vibration data and numerical modeling. This strategy may support future digital twin applications by providing estimated excitation forces that are not directly measurable during operation, but a complete digital twin would still require real-time bidirectional communication between the physical system and its virtual counterpart.

5 FINAL CONSIDERATIONS

5.1 Conclusions

This work aimed to identify dynamic excitation forces caused by rotating mass imbalance in a simplified beam-dc motor system. Since the excitation force could not be directly measured during operation, a deconvolution-based procedure was applied using the measured acceleration response and an equivalent single-degree-of-freedom dynamic model.

The experimental acceleration signals were processed, filtered in the frequency range of interest, and divided into selected operating intervals. The dominant rotational frequencies were 13.93 Hz, 15.67 Hz, and 18.00 Hz. For these operating frequencies, the corresponding force amplitudes were 29.47 N, 47.77 N, and 81.38 N, respectively. When ordered by increasing angular speed, the results showed an associated trend of increasing force, which is consistent with the expected behavior of rotating mass imbalance.

The main contribution of this work is that it is clearly shown that vibration response data, combined with a simplified dynamic model, can be used to estimate excitation forces that are not directly measurable. Such information is relevant for condition monitoring and predictive maintenance of rotating machinery, especially in industrial applications where direct force measurement is difficult or impractical.

However, the results should be interpreted as preliminary estimates. The methodology depends on the quality of the experimental data, the adopted modal parameters, the filtering range, and the numerical conditioning of the deconvolution procedure. In addition, only three operating points were analyzed, and the model used in the deconvolution represents a simplified approximation of the real system.

Therefore, this work should be understood as a foundational step towards future digital twin applications, rather than as the implementation of a complete digital twin. Future developments, as detailed below, may include additional operating conditions, improved dynamic modeling, validation with known imbalance forces, and integration with real-time data acquisition and model updating.

5.2 Suggestions for Future Work

Future work may focus on expanding the experimental database by testing additional motor speeds and different imbalance conditions. This would allow a more robust evaluation of the relationship between angular velocity and estimated excitation force, as well as a better validation of the deconvolution procedure.

Another important improvement would be the refinement of the dynamic model used in the force-identification process. The equivalent single-degree-of-freedom model adopted in this work could be replaced or complemented by a more detailed finite element model, including higher vibration modes, boundary condition flexibility, and the particular influence of the motor mass on the beam dynamics.

The integration of the physical system with the computational model is also recommended as a future development. Real-time data acquisition, automatic signal processing, and continuous model updating could allow the current methodology to evolve from an offline force-identification procedure to a distinct characteristic of a complete digital twin.

Future studies may also investigate the control of the physical system based on the identified excitation forces. Once the excitation is estimated, this information could be used to adjust the motor operation, reduce vibration levels, or support active control strategies.

Finally, artificial neural networks may be explored as complementary tools for force identification and condition monitoring. After being trained with experimental and numerical data, neural networks could assist in estimating excitation forces, identifying operating conditions, or detecting changes in the dynamic behavior of the beam-motor system.

REFERENCES

AIAA. **Digital Twin: definition and value**. An AIAA and AIA Position Paper, 2020. Available at: https://www.aiaa.org/docs/default-source/uploadedfiles/issues-and-advocacy/policy-papers/digital-twin-institute-position-paper-(december-2020).pdf.

BRAUN, S. **Discover Signal Processing: an Interactive Guide for Engineers**. Hoboken: John Wiley & Sons, 2008.

BRENNAN, M. J.; TANG, B. **Virtual Experiments in Mechanical Vibrations: Structural Dynamics and Signal Processing**. Hoboken: John Wiley & Sons, 2023.

CHAUDHARY, G.; KHARI, M.; ELHOSENY, M. (ed.). **Digital Twin Technology**. Boca Raton: CRC Press, 2022.

CHINESTA, F.; CUETO, E.; ABISSET-CHAVANNE, E.; DUVAL, J. L.; EL KHALDI, F. Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. **Archives of Computational Methods in Engineering**, v. 27, n. 1, p. 105-134, 2020.

COVENEY, P. V.; DOUGHERTY, E. R.; HIGHFIELD, R. R. Big data need big theory too. **Philosophical Transactions of the Royal Society A**, v. 374, 20160153, 2016. DOI: https://doi.org/10.1098/rsta.2016.0153.

DA SILVA, S. Not every digital model is a digital twin: an epidemic of misconception. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 47, 594, 2025.

EWINS, D. J. **Modal Testing: Theory, Practice and Application**. 2. ed. Hoboken: John Wiley & Sons, 2009.

GOLDMAN, S. **Vibration Spectrum Analysis**. 2. ed. New York: Industrial Press, 1999.

HARTMANN, D.; VAN DER AUWERAER, H. **Digital twins**. arXiv preprint arXiv:2001.09747, 2020.

HEATON, T. H. **Physics of Earthquakes and Buildings**. 2024. Available at: http://www.its.caltech.edu/~heatont/. Accessed on: May 5, 2025.

INMAN, D. J. **Engineering Vibration**. 4. ed. Boston: Pearson, 2013.

JOHANSEN, S. S.; NEJAD, A. R. On digital twin condition monitoring approach for drivetrains in marine applications. In: **ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering - OMAE2019**, Glasgow, 2019. Paper OMAE2019-95152.

KAPTEYN, M.; WILLCOX, K. **From physics-based models to predictive digital twins via interpretable machine learning**. arXiv preprint arXiv:2004.11356, 2020.

KAPTEYN, M.; WILLCOX, K. Design of digital twin sensing strategies via predictive modeling and interpretable machine learning. **Journal of Mechanical Design**, 2022. Available at: https://kiwi.oden.utexas.edu/papers/Digital-twin-sensing-design-Kapteyn-Willcox.pdf.

KRITZINGER, W.; KARNER, M.; TRAAR, G.; HENJES, J.; SIHN, W. Digital twin in manufacturing: a categorical literature review and classification. **IFAC-PapersOnLine**, v. 51, n. 11, p. 1016-1022, 2018.

LEE, A. W. **Vibration Problems in Machines: Diagnosis and Resolution**. Boca Raton: Taylor & Francis Group, 2016.

LOPES, E. M. O. **Mechanical Vibrations: course notes**. Undergraduate Course in Mechanical Engineering, Federal University of Paraná, 2025. In Portuguese.

MAIA, N. M. M.; SILVA, J. M. M. **Theoretical and Experimental Modal Analysis**. Baldock: Research Studies Press, 1997.

MOI, T.; CIBICIK, A.; ROLVAG, T. Digital twin based condition monitoring of a knuckle boom crane: an experimental study. **Engineering Failure Analysis**, 2020. DOI: https://doi.org/10.1016/j.engfailanal.2020.104517.

RANDALL, R. B. **Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications**. 2. ed. Hoboken: Wiley, 2021.

RANDALL, R. B. **Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications**. Hoboken: John Wiley & Sons, 2011.

RAO, S. S. **Mechanical Vibrations**. 6. ed. London: Pearson, 2018.

SCHEFFER, C.; GIRDHAR, P. **Practical Machinery Vibration Analysis and Predictive Maintenance**. Oxford: Newnes, 2004.

SINGH, M.; FUENMAYOR, E.; HINCHY, E. P.; QIAO, Y.; MURRAY, N.; DEVINE, D. Digital twin: origin to future. **Applied System Innovation**, 2021. DOI: https://doi.org/10.3390/asi4020036.

TAO, F.; ZHANG, M.; NEE, A. Y. C. **Digital Twin Driven Smart Manufacturing**. London: Elsevier/Academic Press, 2019.

TAO, F.; LIU, A.; HU, T.; NEE, A. Y. C. (ed.). **Digital Twin Driven Smart Design**. London: Elsevier/Academic Press, 2020.

TAO, F.; QI, Q.; NEE, A. Y. C. (ed.). **Digital Twin Driven Service**. London: Elsevier/Academic Press, 2022.

TAYLOR, J. I. **The Vibration Analysis Handbook**. Tampa: Vibration Consultants, 2003.

WAGG, D.; WORDEN, K.; BARTHORPE, R.; GARDNER, P. Digital twins: state-of-the-art and future directions for modelling and simulation in engineering dynamics applications. **ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering**, v. 6, n. 3, 030901, 2020.

WANG, J.; LI, Y.; HUANG, Z.; QIAO, Q. Digital twin-driven fault diagnosis service of rotating machinery. In: TAO, F.; QI, Q.; NEE, A. Y. C. (ed.). **Digital Twin Driven Service**. London: Elsevier/Academic Press, 2022. p. 119-138.

WORLD ECONOMIC FORUM. **Digital twins can help the hydrogen economy deliver net zero**. 2023. Available at: https://www.weforum.org/stories/2023/03/digital-twins-can-help-the-hydrogen-economy-deliver-net-zero/. Accessed on: Mar. 5, 2026.

WORDEN, K.; CROSS, E.; BARTHORPE, R.; WAGG, D.; GARDNER, P. On digital twins, mirrors and virtualisations: frameworks for model verification and validation. **ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems**, Part B: Mechanical Engineering, v. 6, n. 3, 030902, 2020.

YUCESAN, Y. A.; VIANA, F. A hybrid model for main bearing fatigue prognosis based on physics and machine learning. In: **AIAA Scitech 2020 Forum**, 2020. DOI: https://doi.org/10.2514/6.2020-1412.

ZHOU, Y.; FU, Z.; ZHANG, J.; LI, W.; GAO, C. A digital twin-based operation status monitoring system for port cranes. **Sensors**, v. 22, n. 9, 3216, 2022. DOI: https://doi.org/10.3390/s22093216.

APPENDIX 1 – PYTHON SCRIPT FOR MODAL PARAMETER IDENTIFICATION FROM IMPACT TEST DATA

This appendix presents the Python script used to process the impact-test data, visualize the force and acceleration signals, calculate the experimental inertance spectra, and regenerate the frequency response function adopted in the modal characterization of the beam–DC motor system.

```
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.optimize import differential_evolution, minimize


BASE_DIR = Path.cwd()


def resolve_file(filename: str) -> Path:
    """
    Locate an input file in the current working directory.

    Accepted examples:
    - impacto_t2.xlsx
    - impacto_t2(1).xlsx
    - impacto_t2(2).xlsx
    """
    candidates = [BASE_DIR / filename]

    stem = Path(filename).stem
    suffix = Path(filename).suffix

    candidates.extend([
        BASE_DIR / f"{stem}(1){suffix}",
        BASE_DIR / f"{stem}(2){suffix}",
    ])

    for candidate in candidates:
        if candidate.exists():
            return candidate

    raise FileNotFoundError(
        f"File not found: {filename}. "
        f"Also searched for {stem}(1){suffix} and {stem}(2){suffix}."
    )


def read_excel_range(path: Path, *, usecols: str, skiprows: int, nrows: int) -> np.ndarray:
    """
    Read a rectangular range from an Excel worksheet and return it as a NumPy array.
    """
    return pd.read_excel(
        path,
        sheet_name=0,
        header=None,
        usecols=usecols,
        skiprows=skiprows,
        nrows=nrows,
        engine="openpyxl",
    ).to_numpy(dtype=float)


def safe_db(x: np.ndarray, eps: float = 1e-30) -> np.ndarray:
    """
    Compute 20*log10(abs(x)) safely, avoiding log10(0).
    """
```

```
    return 20 * np.log10(np.maximum(np.abs(x), eps))


def single_dof_objective(
    normalized_parameters,
    natural_frequency_rad,
    omega_vector,
    experimental_amplitude_db,
    lower_bounds,
    upper_bounds,
):
    """
    Objective function used to adjust the single-degree-of-freedom inertance model.
    """
    parameters = (
        (normalized_parameters - 1) * (upper_bounds - lower_bounds)
        + lower_bounds
    )

    damping_ratio = parameters[0]
    inertance_peak = parameters[1]
    modal_constant = 2 * damping_ratio * inertance_peak

    numerator = modal_constant * (-(omega_vector ** 2))
    denominator = (
        (natural_frequency_rad ** 2 - omega_vector ** 2)
        + (1j * 2 * damping_ratio * natural_frequency_rad * omega_vector)
    )

    regenerated_inertance = numerator / denominator
    regenerated_amplitude_db = safe_db(regenerated_inertance)

    absolute_error = experimental_amplitude_db - regenerated_amplitude_db
    relative_error = absolute_error / experimental_amplitude_db

    objective_value = relative_error @ relative_error.T

    return objective_value


def modal_parameter_identification(show_plots=True, run_minimization=False):
    # Input data files
    time_data_file = resolve_file("impacto_t2.xlsx")
    frequency_data_file = resolve_file("impacto_f2.xlsx")

    # Time-domain data from the impact test
    time_data = read_excel_range(
        time_data_file,
        usecols="A:C",
        skiprows=2,
        nrows=5564,
    )

    # Frequency-domain data from the impact test
    frequency_data = read_excel_range(
        frequency_data_file,
        usecols="C:BZV",
        skiprows=0,
        nrows=10,
    )

    # Time-domain assignment
    time = time_data[:, 0]          # time [s]
    force = time_data[:, 1]         # impact force [N]
    acceleration = time_data[:, 2]  # acceleration [m/s²]

    # Time-domain visualization
    fig1, ax = plt.subplots(2, 1, num=1, clear=True)

    ax[0].plot(time, force, "r-")
    ax[0].axis([0, 60, -20, 20])
    ax[0].grid(True)
    ax[0].set_title("Force vs Time")
    ax[0].set_xlabel("time [s]")
```

```
ax[0].set_ylabel("force [N]")

ax[1].plot(time, acceleration, "g-")
ax[1].axis([0, 60, -30, 30])
ax[1].grid(True)
ax[1].set_title("Acceleration vs Time")
ax[1].set_xlabel("time [s]")
ax[1].set_ylabel("acceleration [m/s²]")

fig1.tight_layout()

# Frequency-domain assignment
frequency = frequency_data[0, :]       # frequency [Hz]
inertance_h1 = frequency_data[4, :]    # H1 estimator amplitude
inertance_h2 = frequency_data[8, :]    # H2 estimator amplitude

# Experimental frequency response function visualization
fig2, ax = plt.subplots(2, 1, num=2, clear=True)

ax[0].plot(frequency, safe_db(inertance_h1), "b-", label="H1 estimator")
ax[0].plot(frequency, safe_db(inertance_h2), "c-", label="H2 estimator")
ax[0].axis([5, 20, -10, 60])
ax[0].grid(True)
ax[0].set_title("Inertance - Amplitude Spectrum")
ax[0].set_xlabel("frequency [Hz]")
ax[0].set_ylabel("amplitude [dB re 1 m/(s²N)]")
ax[0].legend()

ax[1].plot(frequency, inertance_h1, "b-", label="H1 estimator")
ax[1].plot(frequency, inertance_h2, "c-", label="H2 estimator")
ax[1].axis([5, 20, 0, 350])
ax[1].grid(True)
ax[1].set_title("Inertance - Amplitude Spectrum")
ax[1].set_xlabel("frequency [Hz]")
ax[1].set_ylabel("amplitude [m/(s²N)]")
ax[1].legend()

fig2.tight_layout()

# Manual regenerated frequency response function comparison
natural_frequency_hz = 10.23
natural_frequency_rad = 2 * np.pi * natural_frequency_hz
damping_ratio = 0.003

inertance_peak = 260
modal_constant = 2 * damping_ratio * inertance_peak

frequency_fit = np.linspace(5, 20, 500)
omega_fit = 2 * np.pi * frequency_fit

regenerated_inertance = (
    (-(omega_fit ** 2) * modal_constant)
    / (
        (natural_frequency_rad ** 2 - omega_fit ** 2)
        + (1j * 2 * damping_ratio * natural_frequency_rad * omega_fit)
    )
)

plt.figure(3)
plt.clf()
plt.plot(frequency, safe_db(inertance_h1), "b-", label="experimental")
plt.plot(frequency_fit, safe_db(regenerated_inertance), "c-", label="regenerated")
plt.axis([5, 20, -10, 60])
plt.grid(True)
plt.title("Inertance - Amplitude Spectrum")
plt.xlabel("frequency [Hz]")
plt.ylabel("amplitude [dB re 1 m/(s²N)]")
plt.legend()

# Optional numerical minimization block
minimization_result = None

if run_minimization:
    lower_frequency = 5
```

```
upper_frequency = 20

selected_indices = np.where(
    (frequency > lower_frequency) & (frequency < upper_frequency)
)[0]

selected_frequency = frequency[selected_indices]
selected_omega = 2 * np.pi * selected_frequency
number_of_points = len(selected_frequency)

physical_lower_bounds = np.array([5e-3, 2e1], dtype=float)
physical_upper_bounds = np.array([8e-3, 2e2], dtype=float)

normalized_lower_bounds = np.array([1.0, 1.0])
normalized_upper_bounds = np.array([2.0, 2.0])
bounds = list(zip(normalized_lower_bounds, normalized_upper_bounds))

experimental_amplitude_db = safe_db(inertance_h1[selected_indices])

def objective_real(xn):
    return float(
        np.real(
            single_dof_objective(
                xn,
                natural_frequency_rad,
                selected_omega,
                experimental_amplitude_db,
                physical_lower_bounds,
                physical_upper_bounds,
            )
        )
    )

global_result = differential_evolution(objective_real, bounds=bounds)
initial_guess = global_result.x

local_result = minimize(
    objective_real,
    initial_guess,
    method="SLSQP",
    bounds=bounds,
)

optimized_normalized_parameters = local_result.x

optimized_parameters = (
    (optimized_normalized_parameters - 1)
    * (physical_upper_bounds - physical_lower_bounds)
    + physical_lower_bounds
)

damping_ratio = optimized_parameters[0]
inertance_peak = optimized_parameters[1]
modal_constant = 2 * damping_ratio * inertance_peak

regenerated_inertance_optimized = (
    modal_constant * (-(selected_omega ** 2))
    / (
        (natural_frequency_rad ** 2 - selected_omega ** 2)
        + (1j * 2 * damping_ratio * natural_frequency_rad * selected_omega)
    )
)

plt.figure(4)
plt.clf()
plt.plot(frequency, safe_db(inertance_h1), "b-", label="experimental")
plt.plot(
    selected_frequency,
    safe_db(regenerated_inertance_optimized),
    "c-",
    label="regenerated",
)
plt.axis([5, 20, -10, 60])
plt.grid(True)
```

```
        plt.title("Inertance - Amplitude Spectrum")
        plt.xlabel("frequency [Hz]")
        plt.ylabel("amplitude [dB re 1 m/(s²N)]")
        plt.legend()

        minimization_result = {
            "global_result": global_result,
            "local_result": local_result,
            "optimized_normalized_parameters": optimized_normalized_parameters,
            "optimized_parameters": optimized_parameters,
            "damping_ratio": damping_ratio,
            "inertance_peak": inertance_peak,
            "number_of_points": number_of_points,
        }

    # Modal parameters display
    print(f"Natural frequency = {natural_frequency_hz:0.4e} Hz")
    print(f"Damping ratio = {damping_ratio:0.4e}")
    print(f"Modal constant = {modal_constant:0.4e}")
    print(" ")

    if show_plots:
        plt.show()

    return {
        "time": time,
        "force": force,
        "acceleration": acceleration,
        "frequency": frequency,
        "inertance_h1": inertance_h1,
        "inertance_h2": inertance_h2,
        "natural_frequency_hz": natural_frequency_hz,
        "natural_frequency_rad": natural_frequency_rad,
        "damping_ratio": damping_ratio,
        "modal_constant": modal_constant,
        "frequency_fit": frequency_fit,
        "regenerated_inertance": regenerated_inertance,
        "minimization_result": minimization_result,
    }


modal_results = modal_parameter_identification(show_plots=True, run_minimization=False)
```

APPENDIX 2 – PYTHON SCRIPT FOR FORCE IDENTIFICATION BY DECONVOLUTION UNDER ROTATING IMBALANCE

This appendix presents the Python script used to process the operational acceleration data, apply frequency-domain analysis, construct the impulse response functions, estimate the excitation forces by deconvolution, and evaluate the relationship between force amplitude and angular speed.

```
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.linalg import solve
from scipy.signal import butter, filtfilt


BASE_DIR = Path.cwd()

# If True, the deconvolution is performed by explicit matrix inversion.
# If False, a direct linear solver is used instead.
USE_EXPLICIT_INVERSE = True

# Optional qualitative check of the regenerated inertance curve.
PLOT_INERTANCE_DECONVOLUTION_CHECK = False


def resolve_file(filename: str) -> Path:
    """
    Locate an input file in the current working directory.

    Accepted examples:
    - desbanc3.xlsx
    - desbanc3(1).xlsx
    - desbanc3(2).xlsx
    """
    candidates = [BASE_DIR / filename]

    stem = Path(filename).stem
    suffix = Path(filename).suffix

    candidates.extend([
        BASE_DIR / f"{stem}(1){suffix}",
        BASE_DIR / f"{stem}(2){suffix}",
    ])

    for candidate in candidates:
        if candidate.exists():
            return candidate

    raise FileNotFoundError(
        f"File not found: {filename}. "
        f"Also searched for {stem}(1){suffix} and {stem}(2){suffix}."
    )


def read_excel_range(path: Path, *, usecols: str, skiprows: int, nrows: int) -> np.ndarray:
    """
    Read a rectangular range from an Excel worksheet and return it as a NumPy array.
    """
    return pd.read_excel(
        path,
        sheet_name=0,
        header=None,
        usecols=usecols,
        skiprows=skiprows,
        nrows=nrows,
```

```
        engine="openpyxl",
    ).to_numpy(dtype=float)


def safe_db(x: np.ndarray, eps: float = 1e-30) -> np.ndarray:
    """
    Compute 20*log10(abs(x)) safely, avoiding log10(0).
    """
    return 20 * np.log10(np.maximum(np.abs(x), eps))


def half_spectrum_frequency_vector(number_of_samples: int, frequency_resolution: float) -> np.ndarray:
    """
    Build the frequency vector associated with the positive half of a spectrum.
    """
    return np.arange(0, number_of_samples // 2) * frequency_resolution


def build_circshift_matrix(impulse_response: np.ndarray) -> np.ndarray:
    """
    Build the convolution matrix from the impulse response vector using circular shifts.
    """
    h = np.asarray(impulse_response).reshape(-1)
    number_of_samples = len(h)

    matrix_h = np.zeros((number_of_samples, number_of_samples), dtype=float)

    for column_index in range(number_of_samples):
        matrix_h[:, column_index] = np.roll(h, column_index)

    return matrix_h


def deconvolve_by_matrix_inverse(matrix_h: np.ndarray, acceleration: np.ndarray) -> np.ndarray:
    """
    Estimate the force signal from the measured acceleration and the impulse response matrix.
    """
    if USE_EXPLICIT_INVERSE:
        return np.linalg.inv(matrix_h) @ acceleration

    return solve(matrix_h, acceleration, assume_a="gen")


def force_identification_by_deconvolution(show_plots=True):
    # Sampling frequency and sampling interval
    sampling_frequency = 100  # Hz
    sampling_interval = 1 / sampling_frequency

    # Experimental vibration data
    time_data_file = resolve_file("desbanc3.xlsx")

    operational_data = read_excel_range(
        time_data_file,
        usecols="A:B",
        skiprows=2,
        nrows=30909,
    )

    # Full signal assignment
    time = operational_data[:, 0]
    acceleration_full = operational_data[:, 1]
    total_number_of_samples = len(time)

    # Selected operational intervals
    time_1 = time[8999:12000]
    acceleration_1 = acceleration_full[8999:12000]

    time_2 = time[15999:19000]
    acceleration_2 = acceleration_full[15999:19000]

    time_3 = time[22999:26000]
    acceleration_3 = acceleration_full[22999:26000]

    # Initial visualization
```

```
fig1, ax = plt.subplots(2, 1, num=1, clear=True)

ax[0].plot(time, acceleration_full, "k-")
ax[0].axis([0, 350, -15, 15])
ax[0].grid(True)
ax[0].set_title("Acceleration vs Time")
ax[0].set_xlabel("time [s]")
ax[0].set_ylabel("acceleration [m/s²]")

ax[1].plot(time_1, acceleration_1, "r-", label="operating condition 1")
ax[1].plot(time_2, acceleration_2, "g-", label="operating condition 2")
ax[1].plot(time_3, acceleration_3, "b-", label="operating condition 3")
ax[1].axis([0, 350, -5, 5])
ax[1].grid(True)
ax[1].set_title("Acceleration vs Time - Selected Intervals")
ax[1].set_xlabel("time [s]")
ax[1].set_ylabel("acceleration [m/s²]")
ax[1].legend()

fig1.tight_layout()

# Band-pass filtering between 10 Hz and 20 Hz
lower_cutoff_frequency = 10
upper_cutoff_frequency = 20

filter_b, filter_a = butter(
    9,
    [
        lower_cutoff_frequency / (sampling_frequency / 2),
        upper_cutoff_frequency / (sampling_frequency / 2),
    ],
    btype="bandpass",
)

acceleration_1 = filtfilt(filter_b, filter_a, acceleration_1)
acceleration_2 = filtfilt(filter_b, filter_a, acceleration_2)
acceleration_3 = filtfilt(filter_b, filter_a, acceleration_3)

# Frequency-domain transformation: first operating condition
centered_time_1 = time_1 - time_1[0]
acceleration_fft_1 = np.fft.fft(acceleration_1)
number_of_samples_1 = len(acceleration_fft_1)
frequency_resolution_1 = 1 / (centered_time_1[number_of_samples_1 - 1] - centered_time_1[0])
frequency_vector_1 = half_spectrum_frequency_vector(
    number_of_samples_1,
    frequency_resolution_1,
)
acceleration_spectrum_1 = acceleration_fft_1[:len(frequency_vector_1)]

# Frequency-domain transformation: second operating condition
centered_time_2 = time_2 - time_2[0]
acceleration_fft_2 = np.fft.fft(acceleration_2)
number_of_samples_2 = len(acceleration_fft_2)
frequency_resolution_2 = 1 / (centered_time_2[number_of_samples_2 - 1] - centered_time_2[0])
frequency_vector_2 = half_spectrum_frequency_vector(
    number_of_samples_2,
    frequency_resolution_2,
)
acceleration_spectrum_2 = acceleration_fft_2[:len(frequency_vector_2)]

# Frequency-domain transformation: third operating condition
centered_time_3 = time_3 - time_3[0]
acceleration_fft_3 = np.fft.fft(acceleration_3)
number_of_samples_3 = len(acceleration_fft_3)
frequency_resolution_3 = 1 / (centered_time_3[number_of_samples_3 - 1] - centered_time_3[0])
frequency_vector_3 = half_spectrum_frequency_vector(
    number_of_samples_3,
    frequency_resolution_3,
)
acceleration_spectrum_3 = acceleration_fft_3[:len(frequency_vector_3)]

# Acceleration spectra
plt.figure(2)
plt.clf()
```

```
plt.plot(frequency_vector_1, safe_db(acceleration_spectrum_1), "r-", label="operating condition 1")
plt.plot(frequency_vector_2, safe_db(acceleration_spectrum_2), "g-", label="operating condition 2")
plt.plot(frequency_vector_3, safe_db(acceleration_spectrum_3), "b-", label="operating condition 3")
plt.axis([10, 20, -20, 70])
plt.grid(True)
plt.title("Acceleration Amplitude Spectrum")
plt.xlabel("frequency [Hz]")
plt.ylabel("acceleration [dB re 1 m/s²]")
plt.legend()

# System characteristics
natural_frequency_hz = 10.23
natural_frequency_rad = 2 * np.pi * natural_frequency_hz
damping_ratio = 0.003
damped_natural_frequency_rad = natural_frequency_rad * np.sqrt(1 - damping_ratio ** 2)

modal_constant = 1.56
equivalent_modal_mass = 1 / modal_constant

# Impulse response functions
impulse_response_1 = (
    (1 / (equivalent_modal_mass * damped_natural_frequency_rad))
    * np.exp(-damping_ratio * natural_frequency_rad * centered_time_1)
    * np.sin(damped_natural_frequency_rad * centered_time_1)
)

impulse_response_2 = (
    (1 / (equivalent_modal_mass * damped_natural_frequency_rad))
    * np.exp(-damping_ratio * natural_frequency_rad * centered_time_2)
    * np.sin(damped_natural_frequency_rad * centered_time_2)
)

impulse_response_3 = (
    (1 / (equivalent_modal_mass * damped_natural_frequency_rad))
    * np.exp(-damping_ratio * natural_frequency_rad * centered_time_3)
    * np.sin(damped_natural_frequency_rad * centered_time_3)
)

# Impulse response visualization
fig3, ax = plt.subplots(3, 1, num=3, clear=True)

ax[0].plot(centered_time_1, impulse_response_1, "r-")
ax[0].axis([0, 30, -0.03, 0.03])
ax[0].grid(True)
ax[0].set_xlabel("time [s]")
ax[0].set_ylabel("h(t) [m/N]")
ax[0].set_title("Impulse Response Function - Operating Condition 1")

ax[1].plot(centered_time_2, impulse_response_2, "g-")
ax[1].axis([0, 30, -0.03, 0.03])
ax[1].grid(True)
ax[1].set_xlabel("time [s]")
ax[1].set_ylabel("h(t) [m/N]")
ax[1].set_title("Impulse Response Function - Operating Condition 2")

ax[2].plot(centered_time_3, impulse_response_3, "b-")
ax[2].axis([0, 30, -0.03, 0.03])
ax[2].grid(True)
ax[2].set_xlabel("time [s]")
ax[2].set_ylabel("h(t) [m/N]")
ax[2].set_title("Impulse Response Function - Operating Condition 3")

fig3.tight_layout()

# Force determination by deconvolution
matrix_h_1 = build_circshift_matrix(impulse_response_1)
matrix_h_2 = build_circshift_matrix(impulse_response_2)
matrix_h_3 = build_circshift_matrix(impulse_response_3)

reconstructed_force_1 = deconvolve_by_matrix_inverse(matrix_h_1, acceleration_1)
reconstructed_force_2 = deconvolve_by_matrix_inverse(matrix_h_2, acceleration_2)
reconstructed_force_3 = deconvolve_by_matrix_inverse(matrix_h_3, acceleration_3)

# Band-pass filtering of reconstructed forces
```

```
filter_b, filter_a = butter(
    9,
    [
        lower_cutoff_frequency / (sampling_frequency / 2),
        upper_cutoff_frequency / (sampling_frequency / 2),
    ],
    btype="bandpass",
)

reconstructed_force_1 = filtfilt(filter_b, filter_a, reconstructed_force_1)
reconstructed_force_2 = filtfilt(filter_b, filter_a, reconstructed_force_2)
reconstructed_force_3 = filtfilt(filter_b, filter_a, reconstructed_force_3)

# Force spectra
force_fft_1 = np.fft.fft(reconstructed_force_1)
force_spectrum_1 = force_fft_1[:len(frequency_vector_1)]

force_fft_2 = np.fft.fft(reconstructed_force_2)
force_spectrum_2 = force_fft_2[:len(frequency_vector_2)]

force_fft_3 = np.fft.fft(reconstructed_force_3)
force_spectrum_3 = force_fft_3[:len(frequency_vector_3)]

# Force visualization in time and frequency domains
fig4, ax = plt.subplots(2, 3, num=4, clear=True)

ax[0, 0].plot(centered_time_1, reconstructed_force_1, "r-")
ax[0, 0].axis([0, 30, -100, 100])
ax[0, 0].grid(True)
ax[0, 0].set_title("Reconstructed Force - Condition 1")
ax[0, 0].set_xlabel("time [s]")
ax[0, 0].set_ylabel("force [N]")

ax[0, 1].plot(centered_time_2, reconstructed_force_2, "g-")
ax[0, 1].axis([0, 30, -100, 100])
ax[0, 1].grid(True)
ax[0, 1].set_title("Reconstructed Force - Condition 2")
ax[0, 1].set_xlabel("time [s]")
ax[0, 1].set_ylabel("force [N]")

ax[0, 2].plot(centered_time_3, reconstructed_force_3, "b-")
ax[0, 2].axis([0, 30, -100, 100])
ax[0, 2].grid(True)
ax[0, 2].set_title("Reconstructed Force - Condition 3")
ax[0, 2].set_xlabel("time [s]")
ax[0, 2].set_ylabel("force [N]")

ax[1, 0].plot(frequency_vector_1, safe_db(force_spectrum_1), "r-")
ax[1, 0].axis([10, 20, 0, 100])
ax[1, 0].grid(True)
ax[1, 0].set_xlabel("frequency [Hz]")
ax[1, 0].set_ylabel("force [dB re 1 N]")

ax[1, 1].plot(frequency_vector_2, safe_db(force_spectrum_2), "g-")
ax[1, 1].axis([10, 20, 0, 100])
ax[1, 1].grid(True)
ax[1, 1].set_xlabel("frequency [Hz]")
ax[1, 1].set_ylabel("force [dB re 1 N]")

ax[1, 2].plot(frequency_vector_3, safe_db(force_spectrum_3), "b-")
ax[1, 2].axis([10, 20, 0, 100])
ax[1, 2].grid(True)
ax[1, 2].set_xlabel("frequency [Hz]")
ax[1, 2].set_ylabel("force [dB re 1 N]")

fig4.tight_layout()

# Optional qualitative inertance comparison
regenerated_frequency = np.linspace(5, 20, 500)
regenerated_omega = 2 * np.pi * regenerated_frequency

regenerated_inertance = (
    (-(regenerated_omega ** 2) * modal_constant)
    / (
```

```
        (natural_frequency_rad ** 2 - regenerated_omega ** 2)
        + (1j * 2 * damping_ratio * natural_frequency_rad * regenerated_omega)
    )
)

frequency_points = np.array([13.93, 15.67, 18.00])
acceleration_db_points = np.array([63.55, 64.17, 63.45])
force_db_points = np.array([90.40, 94.82, 97.70])

inertance_db_points = acceleration_db_points - force_db_points

if PLOT_INERTANCE_DECONVOLUTION_CHECK:
    plt.figure(6)
    plt.clf()
    plt.plot(
        regenerated_frequency,
        safe_db(regenerated_inertance),
        "k-",
        label="regenerated",
    )
    plt.plot(
        frequency_points,
        inertance_db_points + 39,
        "*r",
        label="deconvolution points",
    )
    plt.axis([5, 20, -10, 50])
    plt.grid(True)
    plt.title("Inertance - Amplitude Spectrum")
    plt.xlabel("frequency [Hz]")
    plt.ylabel("amplitude [dB re 1 m/(s²N)]")
    plt.legend()

# Force amplitude dependence on angular speed
angular_speed = 2 * np.pi * np.array([13.93, 15.67, 18.00])
force_amplitude = np.array([29.47, 47.77, 81.38])

# Second-degree polynomial fit
polynomial_coefficients = np.polyfit(angular_speed, force_amplitude, deg=2)

angular_speed_fit = np.linspace(np.min(angular_speed), np.max(angular_speed), 500)
force_amplitude_fit = np.polyval(polynomial_coefficients, angular_speed_fit)

plt.figure(7)
plt.clf()
plt.plot(angular_speed_fit, force_amplitude_fit, "-", label="second-degree polynomial fit")
plt.plot(angular_speed, force_amplitude, "o", label="estimated force amplitudes")
plt.axis([80, 120, 0, 100])
plt.grid(True)
plt.title("Force Amplitude as a Function of Angular Speed")
plt.xlabel("angular speed [rad/s]")
plt.ylabel("force amplitude [N]")
plt.legend()

if show_plots:
    plt.show()

return {
    "sampling_frequency": sampling_frequency,
    "sampling_interval": sampling_interval,
    "time": time,
    "acceleration_full": acceleration_full,
    "total_number_of_samples": total_number_of_samples,
    "time_1": time_1,
    "time_2": time_2,
    "time_3": time_3,
    "acceleration_1": acceleration_1,
    "acceleration_2": acceleration_2,
    "acceleration_3": acceleration_3,
    "centered_time_1": centered_time_1,
    "centered_time_2": centered_time_2,
    "centered_time_3": centered_time_3,
    "frequency_vector_1": frequency_vector_1,
    "frequency_vector_2": frequency_vector_2,
```

```
        "frequency_vector_3": frequency_vector_3,
        "acceleration_spectrum_1": acceleration_spectrum_1,
        "acceleration_spectrum_2": acceleration_spectrum_2,
        "acceleration_spectrum_3": acceleration_spectrum_3,
        "natural_frequency_hz": natural_frequency_hz,
        "natural_frequency_rad": natural_frequency_rad,
        "damping_ratio": damping_ratio,
        "damped_natural_frequency_rad": damped_natural_frequency_rad,
        "modal_constant": modal_constant,
        "equivalent_modal_mass": equivalent_modal_mass,
        "impulse_response_1": impulse_response_1,
        "impulse_response_2": impulse_response_2,
        "impulse_response_3": impulse_response_3,
        "reconstructed_force_1": reconstructed_force_1,
        "reconstructed_force_2": reconstructed_force_2,
        "reconstructed_force_3": reconstructed_force_3,
        "force_spectrum_1": force_spectrum_1,
        "force_spectrum_2": force_spectrum_2,
        "force_spectrum_3": force_spectrum_3,
        "regenerated_frequency": regenerated_frequency,
        "regenerated_inertance": regenerated_inertance,
        "frequency_points": frequency_points,
        "acceleration_db_points": acceleration_db_points,
        "force_db_points": force_db_points,
        "inertance_db_points": inertance_db_points,
        "polynomial_coefficients": polynomial_coefficients,
        "angular_speed": angular_speed,
        "force_amplitude": force_amplitude,
    }


force_results = force_identification_by_deconvolution(show_plots=True)
```