



DESENVOLVIMENTO DE PROGRAMA PARA IDENTIFICAÇÃO DOS PARÂMETROS MODAIS DE UMA ESTRUTURA UNIDIMENSIONAL

MIGUEL FERNANDO PINTO QUEIRÓS

outubro de 2024

**DESENVOLVIMENTO DE PROGRAMA PARA
IDENTIFICAÇÃO DOS PARÂMETROS MODAIS DE
UMA ESTRUTURA UNIDIMENSIONAL**

Miguel Fernando Pinto Queirós

**Dissertação para obtenção do Grau de Mestre em
Engenharia Mecânica, Área de Especialização em
Construções Mecânicas**

Orientador: Jorge Manuel da Costa Fonseca Justo

Co-orientador: Hernâni Miguel Reis Lopes

Júri:

Presidente:

Armando José Vilaça de Campos, Professor Adjunto, ISEP

Vogais:

Pedro José da Silva Carvalho Pereira de Sousa, Professor Auxiliar, FEUP

Jorge Manuel da Costa Fonseca Justo, Professor Adjunto, ISEP

Agradecimentos

Em primeiro lugar gostaria de agradecer aos meus orientadores, Jorge Manuel da Costa Fonseca Justo e Hernâni Miguel Reis Lopes, pela disponibilidade, ajuda e toda a paciência demonstrada ao longo deste processo. Apesar de mais longo do que esperávamos, nunca baixaram os braços e mostraram-se sempre positivos e confiantes no meu trabalho. Estarei sempre grato por isso.

Agradecer também ao ISEP, que nos últimos anos tem sido praticamente como uma casa. Não foi a primeira opção, mas rapidamente se tornou a opção certa.

De seguida, gostaria de agradecer à minha família por acreditarem sempre em mim e me darem sempre o alento necessário para eu continuar a querer ser melhor. Aos meus pais, que nunca me deixaram sem um porto de abrigo. Ao David, por todas as conversas simplesmente para desanuviar, e claro, todo o apoio demonstrado. Tenho orgulho imenso em ser teu irmão.

Agradecer também aos meus amigos, por todos os momentos ao longo desta jornada e pela ajuda. Em especial ao Henrique, por me ter emprestado o portátil para concluir o trabalho, ao Gonçalo pelas idas até ao H para avançar no mesmo e ao Carlos pela ajuda indeterminável. E claro, a todos os outros que marcaram esta caminhada. Não foi fácil, mas lá fomos conseguindo

E por último, agradecer à Filipa, pela amizade, compreensão, carinho e apoio incondicional. Obrigado pela ajuda diária e por me fazeres uma pessoa melhor todos os dias. Sem ti nada disto era possível.

Resumo

O presente trabalho consiste no desenvolvimento de um programa em *Python* para a medição das funções de resposta em frequência, e posterior identificação das frequências e modos naturais de vibração de uma estrutura unidimensional. Este teve por base uma revisão bibliográfica sobre os conceitos essenciais ao desenvolvimento do programa. O programa, realizado em código de fonte aberta, ou seja, não requer licenciamento, subdivide-se na aquisição e processamento dos sinais, e na posterior análise para a identificação das frequências e modos. Na rotina de aquisição e processamento é realizada aquisição dos sinais, os quais são processados para se obterem as funções de resposta em frequência. Na rotina de análise são identificadas as frequências e os respectivos a partir dos valores de pico das funções de resposta em frequência medidas. Como forma de validação do programa implementado, foram identificadas as três primeiras frequências e modos naturais de vibração de uma viga em aço para a condição livre-livre. O ensaio foi realizado com um martelo impacto e um acelerómetro fixo numa extremidade da viga, e o sinal adquirido com uma placa de aquisição da *National Instruments*®. Os resultados obtidos foram comparados com as soluções numérica, realizada no *software ANSYS*®, e analítica. Os resultados obtidos pelos três métodos encontram-se bastante próximos, pelo que permite validar o programa desenvolvido e o procedimento de medição experimental. Perspetivaram-se ainda melhorias ao mesmo, nomeadamente a implementação de um *trigger level*.

Palavras-chave: *Python*, Frequências naturais, Modos naturais, Ensaio experimental, Função de resposta em frequência.

Abstract

This work consists of developing a Python program for measuring frequency response functions and subsequently identifying the natural vibration frequencies and modes of a one-dimensional structure. It was based on a bibliographical review of the concepts essential to the development of the program. The program, made in open-source code, i.e. it does not require licensing, is subdivided into signal acquisition and processing, and subsequent analysis to identify frequencies and modes. In the acquisition and processing routine, signals are acquired and processed to obtain the frequency response functions. In the analysis routine, the frequencies and modes are identified from the peak values of the measured frequency response functions. As a way of validating the program implemented, the first three frequencies and natural modes of vibration of a steel beam were identified for the free-floating condition. The test was carried out with an impact hammer and an accelerometer fixed at one end of the beam, and the signal acquired with a National Instruments® acquisition board. The results obtained were compared with numerical solutions, carried out using ANSYS® software, and analytical solutions. The results obtained by the three methods are very close, which makes it possible to validate the program developed and the experimental measurement procedure. Improvements to the program are also planned, namely the implementation of a trigger level.

KEYWORDS: Python, Natural frequencies, Natural modes, Experimental test, Frequency response function.

Índice

Lista de Figuras.....	ix
Lista de Tabelas.....	xi
Acrónimos e Símbolos.....	xiii
1. Introdução	1
1.1. Contextualização	1
1.2. Objetivos	1
1.3. Metodologia de desenvolvimento.....	2
1.4. Estrutura do relatório.....	2
2. Revisão bibliográfica	5
2.1. Vibrações e a as suas aplicações.....	5
2.2. Análise modal	5
2.2.1. Graus de liberdade	7
2.2.2. Modos de vibração.....	8
2.2.3. Método de sobreposição modal.....	10
2.2.4. Transformada de Fourier.....	10
2.2.5. Função resposta em frequência (FRF)	11
2.2.6. Análise modal experimental (EMA)	12
2.3. Equipamentos e funcionamento	13
2.3.1. Instrumentos de Vibração	13
2.3.2. Acelerómetros.....	15
2.3.3. Placa de aquisição de sinal	16
2.4. Programas	17
2.4.1. LabVIEW	17
2.4.2. MATLAB.....	18
2.4.3. Python	18
3. Desenvolvimento de programa para determinação dos parâmetros modais	21
3.1. Introdução	21
3.2. Programa de aquisição e processamento	23
3.2.1. Aquisição	23
3.2.2. Pré-processamento	27
3.2.3. Processamento.....	32
3.3. Programa de análise	34
3.3.1. Leitura de dados.....	34
3.3.2. Pós-processamento.....	35
3.3.3. Obtenção dos modos de vibração	35

4. Resultados e discussão.....	37
4.1. Ensaio experimental	37
4.2. Solução numérica	46
4.3. Solução analítica	50
4.4. Comparação e discussão de resultados	53
5. Conclusão e proposta de trabalhos futuros	57
5.1. Conclusões.....	57
5.2. Proposta de trabalhos futuros.....	58
Referências	59
Declaração de Integridade	63
Apêndice A	65
Apêndice B	67
Apêndice C	79

Lista de Figuras

Figura 1 - Sistema massa mola [4]	5
Figura 2 -Modelo com um Grau de Liberdade (Adaptado [8])	7
Figura 3 – Estrutura com dois Graus de Liberdade (Adaptado [8])	8
Figura 4 - Modos de Vibração (Adaptado [5])	9
Figura 5 - Pontos Nodais de uma viga encastrada numa extremidade (Adaptado [5])	10
Figura 6 - Diagrama FRF (Adaptado [9])	11
Figura 7 - Procedimento EMA (Adaptado [16])	13
Figura 8 - Típico Martelo de Impacto (Adaptado [23])	15
Figura 9 - Relação entre Sensibilidade e Frequência de um Acelerómetro (Adaptado [23])	15
Figura 10 - Colocação de Acelerómetros [25]	16
Figura 11 - NI USB-4432 [26]	17
Figura 12 - Rotina principal	21
Figura 13 - Instalação das bibliotecas.....	22
Figura 14 – Diagrama de Blocos do Programa de Aquisição e Processamento	22
Figura 15 - Diagrama de Blocos do Programa de Análise	23
Figura 16 - Função de aquisição	25
Figura 17 – Criação do ficheiro de gravação.....	26
Figura 18 - Sinal completo da Força	26
Figura 19 - Sinal da Força truncado no tempo	27
Figura 20 - Sinal da Resposta.....	27
Figura 21 - Função de deteção do impacto	28
Figura 22 - Função de ponderação da força	29
Figura 23 - Sinal da Força Ponderado	29
Figura 24 – Sinal no tempo da força não ponderado, força ponderada e a função de ponderação	30
Figura 25 - Função de ponderação da aceleração	30
Figura 26 - Sinal da aceleração ponderada.....	31
Figura 27 – Sinal no tempo da aceleração não ponderado, da aceleração ponderada e a função de ponderação	31
Figura 28 – Representação gráfica da função ponderação da aceleração.....	32
Figura 29 - Função FFT	32
Figura 30 – Autoespectro da força	33
Figura 31 – Autoespectro da aceleração	33
Figura 32 –FRF medida através do programa desenvolvido.....	34
Figura 33 - Função de gravação de dados	34
Figura 34 - Função de leitura de dados	35
Figura 35 - Função de interpolação	36
Figura 36 - Viga.....	37
Figura 37 - Ensaio Experimental Viga	38
Figura 38 - Martelo de impacto PCB Piezotronics modelo 086C01	38

Figura 39 - Acelerómetro da PCB Piezotronics modelo 352A24	39
Figura 40 - NI USB 4431.....	39
Figura 41 - Leitura de dados do canal 0 no ponto 2	40
Figura 42 - Leitura de dados do canal 1 no ponto 2	40
Figura 43 - Sinal da força ponderado no ponto 2	41
Figura 44 - Sinal da força ponderado no ponto 2	41
Figura 45 – FRF medida no ponto 1.....	42
Figura 46 - FRF medida no ponto 2	42
Figura 47 - FRF medida no ponto 3	42
Figura 48 - FRF medida no ponto 4	43
Figura 49 - FRF medida no ponto 5	43
Figura 50 – FRF medida no ponto 6.....	43
Figura 51 - FRF medida no ponto 7	44
Figura 52 – FRF medida no ponto 8.....	44
Figura 53 – FRF medida no ponto 9.....	44
Figura 54 - Primeiro modo de natural de vibração - experimental.....	45
Figura 55 - Segundo modo de natural de vibração - experimental	46
Figura 56 - Terceiro modo de natural de vibração - experimental	46
Figura 57 - Modelo geométrico da viga.....	47
Figura 58 – Modelo numérico da viga	47
Figura 59 - Vibração Corpo Rígido	48
Figura 60 - Primeiro modo natural de vibração- numérico	48
Figura 61 - Segundo modo natural de vibração- numérico	49
Figura 62 - Terceiro modo natural de vibração- numérico	49
Figura 63 - Quarto Modo de Vibração – Análise Numérica	49
Figura 64 - Quinto Modo de Vibração - Análise Numérica	50
Figura 65 - Primeiro modo de natural de vibração – analítico	52
Figura 66 - Segundo modo de natural de vibração – analítico	52
Figura 67 - Terceiro modo de natural de vibração – analítico	53
Figura 68 – Primeiro modo natural de vibração obtido pelos três métodos	54
Figura 69 - Segundo modo natural de vibração obtido pelos três métodos.....	55
Figura 70 - Terceiro modo natural de vibração obtido pelos três métodos	55

Lista de Tabelas

Tabela 1 – Propriedades da viga.....	37
Tabela 2 - Frequências naturais obtidas experimentalmente	45
Tabela 3 - Frequências naturais obtidas numericamente	48
Tabela 4 - Números de onda	50
Tabela 5 - Frequências naturais obtidas analiticamente	51
Tabela 6 - Resumo de frequências naturais da viga	53
Tabela 7 - Erro relativo entre frequências naturais	54

Acrónimos e Símbolos

Lista de Acrónimos

API	Interface de Programação de Aplicações
DC	Diagrama Cinético
DCL	Diagrama de Corpo Livre
DEM	Departamento de Engenharia Mecânica
DFT	Transformada Discreta de Fourier
DOF	Grau de Liberdade
EMA	Análise Modal Experimental
FFT	Transformada Rápida de Fourier
FRF	Função de Resposta em Frequência
FEM	Método de Elementos Finitos
GUI	Interface Gráfica de Utilizador
ISEP	Instituto Superior de Engenharia do Porto
LMA	Laboratório de Mecânica Aplicada do Departamento de Engenharia Mecânica
MEF	Método de Elementos Finitos
NI	<i>National Instruments</i> ®
P.Porto	Instituto Politécnico do Porto

Lista de Símbolos

m	massa	kg
k	rigidez	N/m
c	coeficiente de amortecimento	N. s/m
c_c	coeficiente crítico de amortecimento	N.s/m
ξ	fator de amortecimento	[–]
ω_n	frequência natural	rad/s
F	força externa	N
k_n	número de onda	[–]

1. Introdução

Neste capítulo é realizado o enquadramento do trabalho em causa, é definida, também, a sua importância e ainda os objetivos que se pretendem atingir com o mesmo.

Posteriormente, apresenta-se a metodologia de desenvolvimento do trabalho em estudo e, por fim, a estrutura do relatório.

1.1. Contextualização

No projeto e desenvolvimento de componentes sujeitos a solicitações dinâmicas, nomeadamente na indústria automóvel [1], de forma a poder prevenir avarias e falhas em serviço, é necessário determinar a resposta dinâmica, a qual depende das frequências naturais, modos e razões de amortecimento modal. A determinação correta destes parâmetros pode ser concretizada via experimental, recorrendo ao Método de Análise Modal Experimental, EMA. Este método permite obter as características fundamentais de uma estrutura ou componentes, ou seja a massa, rigidez e amortecimento associados, e estudar esta mesma estrutura relativamente ao seu comportamento dinâmico. Este método envolve, ainda, a excitação da estrutura ou componente e a medição da sua resposta, em condições controladas.

Em suma, uma EMA permite a criação de um modelo numérico que tem como objetivo a reprodução do comportamento dinâmico da estrutura. Este modelo numérico permite que lhe sejam realizadas modificações, garantido que estas não afetaram as suas características fundamentais, tais como a massa, rigidez e amortecimento.

1.2. Objetivos

Os principais objetivos deste trabalho são o desenvolvimento de um programa para realizar a aquisição dinâmica de sinal, o seu processamento e análise, em fonte aberta (*opensource*), que permita a identificação de parâmetros modais, nomeadamente frequências naturais e modos de vibração de uma estrutura unidimensional

Este programa, sendo desenvolvido numa linguagem de fonte aberta, garante deste modo, a possibilidade de introduzir ajustes e atualizações. Esta é uma vantagem muito interessante, pois, devido à normal evolução dos sistemas operativos de um computador, os *softwares* comerciais utilizados para a aquisição e tratamento de dados acabam por ficar desatualizados, limitando assim a sua utilização.

1.3. Metodologia de desenvolvimento

Para a concretização dos objetivos propostos será necessário a construção de raiz de um programa para realizar a aquisição dinâmica de sinal, o seu processamento e análise. Deste modo, este programa terá de ser capaz de configurar os parâmetros de aquisição do equipamento *National Instruments*® existente no Laboratório de Mecânica Aplicada situado nas Instalações do ISEP. O programa terá, ainda, de ser capaz de adquirir os sinais e realizar o processamento dos mesmo de modo a obter funções de resposta em frequência. Finalmente, a partir da análise das funções de resposta em frequência, o programa terá de ser capaz de extrair os parâmetros modais.

O programa em estudo deverá ser desenvolvido na linguagem *Python*.

Relativamente aos equipamentos necessário para o desenvolvimento deste trabalho, presentes no Laboratório de Mecânica Aplicada, estes são a placa de aquisição de sinal, martelo de impacto, acelerómetros e computador.

1.4. Estrutura do relatório

Este relatório divide-se em cinco capítulos, sendo eles a Introdução, Revisão bibliográfica, Desenvolvimento de programa para determinação de parâmetros modais, Resultados e discussão e por fim, Conclusão e proposta de trabalhos futuros.

No que diz respeito à Introdução, nesta encontra-se a contextualização do trabalho em estudo. São, também, descritos neste capítulo os objetivos que se pretendem com a realização deste trabalho. Posteriormente apresentam-se as metodologias de desenvolvimento que envolvem todos os equipamentos e programas que estão envolvidos para atingir os objetivos do trabalho. Por fim é apresentada a estrutura do relatório, com uma breve descrição dos capítulos e subcapítulos presentes neste trabalho.

De seguida, encontra-se o segundo capítulo que consiste na apresentação dos diversos aspetos abordados para uma melhor compreensão do trabalho em estudo. Deste modo, realiza-se uma breve introdução ao conceito de vibrações e ainda às suas aplicações. Posto isto, passou-se para uma explicação de análise modal e das diversidades que esta apresenta, nomeadamente no que diz respeito ao número de eixos em que a estrutura se pode deslocar (Graus de Liberdade) e modos de vibração. De seguida, abordaram-se os assuntos mais práticos do trabalho, sendo estes a Transformada de Fourier, a Função Resposta em Frequência e a Análise Modal Experimental. De seguida, referiram-se os equipamentos associados ao trabalho em estudo, equipamentos que se encontram no LMA, sendo eles o Martelo de Impacto, os Acelerómetros e Placa de Aquisição, e realizou-se a explicação do seu funcionamento. Por fim, abordaram-se os programas existentes para aquisição e tratamento de sinal existentes, *LabVIEW* e *MATLAB*, e ainda a linguagem *Python*. Através desta última linguagem deve ser desenvolvido o programa para realizar a aquisição e processamento de sinal que permitirá a identificação de parâmetros modais de uma estrutura unidimensional

Posteriormente, encontra-se o terceiro capítulo relativo ao desenvolvimento de programa para a determinação de parâmetros modais, abordando os métodos e aplicação associados. Assim, e tendo em conta a apresentação dos aspetos abordados, aborda-se como se realizou a divisão em duas partes do programa em desenvolvimento. Numa primeira parte, ou seja, na aquisição e processamento de sinal, abordam-se todas as configurações e cuidados a ter para o bom funcionamento entre a placa e o programa. Efetuou-se, inicialmente, uma verificação desse funcionamento da placa de aquisição de sinal sendo posteriormente realizada a programação e constante validação mesma. Depois, realizou -se a programação do pré-processamento de sinal, de modo a truncar o sinal apenas para as amostras necessárias. De seguida, programou-se o processamento do sinal, de modo a trabalhar o sinal truncado tendo como objetivo obter a transformada de Fourier do sinal da força e do sinal da resposta para, no fim, obter a função de resposta em frequência do caso de estudo em análise. Por último, e obtidas as funções de resposta em frequência de diversos pontos da estrutura unidimensional em estudo, surge a necessidade de determinar as frequências naturais de vibração e mostrar a representação gráfica dos modos de vibração da mesma, através do programa de análise, sendo então este o principal objetivo desta última parte.

De seguida, encontra-se o quarto capítulo. Este capítulo tem como objetivo, depois do enquadramento teórico e prático realizado nos dois capítulos anteriores, a explicação do ensaio experimental propriamente dito. Inicialmente é realizado uma contextualização e apontados os parâmetros necessários para a realização do mesmo, sendo neste caso a realização de um ensaio experimental numa viga de 596x40x6 (mm) dividida em 9 pontos onde seria aplicado o impacto através do martelo de impacto. De seguida, encontram-se os resultados obtidos do ensaio descrito, com a apresentação das funções de resposta em frequência dos pontos em análise de modo que seja possível uma melhor compreensão do comportamento da estrutura e, também, as frequências e modos naturais de vibração da viga em estudo. Posteriormente realiza-se uma análise numérica no ANSYS® com o objetivo de obter as frequências e modos de vibração nas mesmas condições que o ensaio experimental e efetua-se também, uma análise analítica com o mesmo propósito. Em último lugar, é realizada uma comparação dos resultados obtidos no programa desenvolvido e os restantes métodos aplicados de modo a verificar a proximidade dos resultados obtidos.

No que diz respeito ao quinto e último capítulo, neste apresentam-se as considerações finais do trabalho onde se faz um resumo dos resultados obtidos e dificuldades apresentadas ao longo deste processo e, também, propostas para a constante melhoria deste programa realizado em fonte aberta.

Introdução

2. Revisão bibliográfica

No presente capítulo são descritos os conceitos fundamentais para a realização do trabalho. Deste modo, conceitos como análise modal, equipamentos e programas são aprofundados ao longo do capítulo.

2.1. Vibrações e as suas aplicações

No ramo de engenharia, uma análise de vibrações permite a resolução de problemas de estruturas ou máquinas e melhorar, assim, o seu funcionamento. Por exemplo, na indústria automóvel, de modo a identificar problemas relacionados com veículos [1] e ainda na indústria aeroespacial uma vez que este permite identificar problemas associados com aeronaves [2].

Em suma, o estudo de vibrações desempenha um papel crucial na constante melhoria de segurança de uma estrutura e no melhor desempenho da mesma uma vez que permite o estudo dinâmico da estrutura, sem alteração da mesma. Isto é, através da construção de um modelo numérico, é possível prever o comportamento da estrutura em função da alteração introduzida.

2.2. Análise modal

Análise modal é o estudo das propriedades dinâmicas de um sistema [3]. Esta tem como objetivo a resolução de problemas relacionados com vibrações em máquinas ou estruturas e para tal é necessário recorrer à obtenção das características dinâmicas dessa mesma máquina ou estrutura em análise.

O sistema mais simples enquadrado num estudo dinâmico é o sistema massa mola que se encontra representado na Figura 1. Este é constituído por uma massa m em movimento de translação e uma mola de rigidez k .

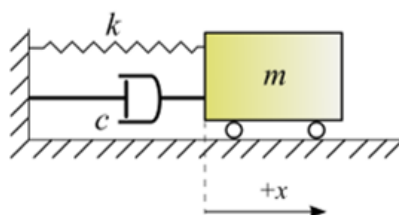


Figura 1 - Sistema massa mola [4]

As características dinâmicas definem-se por três parâmetros [5].

- Frequências Naturais – Conjunto de frequências intrínsecas a qualquer objeto, formado por um material elástico, que vibrará quando perturbado, isto é, caso exista uma excitação. No caso de a estrutura ser amortecida, a descrição matemática desta frequência será complexa, tendo, então, uma parte imaginária que descreve a taxa de oscilação e uma parte real que descreve a taxa de amortecimento. No caso de um sistema com apenas um grau de liberdade, a frequência natural, ω_n , pode ser calculada através da Equação 1.

$$\omega_n = \sqrt{\frac{k}{m}} \quad (1)$$

- Amortecimento – trata-se de qualquer dos vários mecanismos físicos em que a energia mecânica é dissipada e, deste modo, removida do sistema vibratório. Na falta de uma entrada constante de energia, a vibração tende para o repouso como consequência direta dos mecanismos de amortecimento presentes. O fator de amortecimento, ζ , pode ser calculado através da Equação 2.

$$\zeta = \frac{c}{c_c} \quad (2)$$

Sendo que o coeficiente crítico de amortecimento, c_c , é obtido através da Equação 3.

$$c_c = 2\sqrt{k \times m} \quad (3)$$

- Modos de Naturais de Vibração – padrão de vibração exibido por uma estrutura ou sistema descrito, geralmente como um vetor de valores, definindo as amplitudes relativas de deslocamento de cada grau de liberdade que descreve o movimento de um sistema complexo.

A equação diferencial, Equação 4, permite descrever a posição de um sistema com um grau de liberdade, em regime forçado com amortecimento, para um instante no tempo.

$$F(t) = m\ddot{x}(t) + c\dot{x}(t) + kx(t) \quad (4)$$

Sendo que, $F(t)$ é uma solicitação externa sofrida pela estrutura, m a massa equivalente da estrutura, c o amortecimento equivalente da estrutura e k a rigidez equivalente da estrutura. Deste modo, é necessário conhecer as condições iniciais, isto é, posição e velocidade inicial, do sistema de modo a ser possível a obtenção da posição do sistema [4].

Uma vez identificados os parâmetros modais, estão reunidas todas as condições para determinar o comportamento dinâmico de uma estrutura ou máquina.

Todos os sistemas podem ter diversas frequências naturais, que estão relacionadas com o material, densidade e amortecimento do sistema [3]. Assim, em casos onde exista uma frequência de uma solicitação externa igual a uma das frequências naturais do sistema, ocorre uma ressonância que leva o sistema a vibrar com amplitudes extremamente elevadas e potencialmente perigosas.

Posto isto, os parâmetros modais de todos os modos de vibração, inseridos no intervalo de frequências de interesse, apresentam-se como uma descrição completa de uma estrutura sujeita a esta análise [6].

2.2.1. Graus de liberdade

Um grau de Liberdade (DOF) trata-se de um ponto de medição e direção definido de uma estrutura. Relativamente aos índices associados, um índice i é utilizado, habitualmente, para indicar a resposta, um índice j é usado para indicar uma excitação. Para além destes, são ainda utilizados os índices x, y e z com o objetivo de indicar as direções [7].

- **Sistemas com um grau de liberdade**

Um sistema com um Grau de Liberdade caracteriza-se por apenas ser necessário existir uma coordenada para se conseguir definir, completamente, a posição da estrutura em qualquer instante do tempo [5]. Através da Figura 2 pode-se observar um sistema com apenas um grau de liberdade e analisado tendo em conta o seu Diagrama de Corpo Livre (DCL).

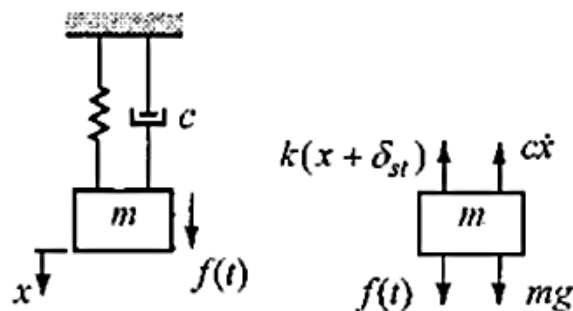


Figura 2 -Modelo com um Grau de Liberdade (Adaptado [8])

- **Sistema com dois ou mais graus de liberdade**

Um sistema com dois ou mais Graus de Liberdade caracteriza-se por ser necessário existir duas ou mais coordenadas para se conseguir definir, completamente, a posição da estrutura em qualquer instante do tempo [5]. Na Figura 3, encontra-se representado um sistema que contém dois graus de liberdade sendo representados o Diagrama de corpo livre, DCL associado à estrutura.

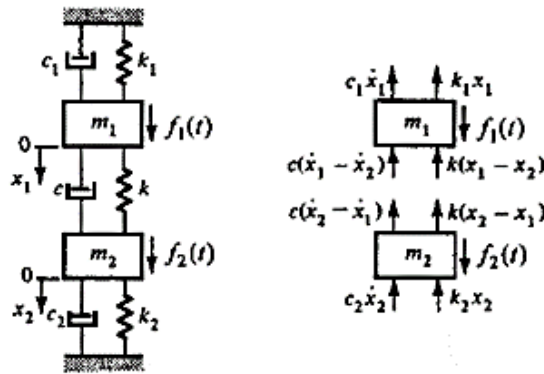


Figura 3 – Estrutura com dois Graus de Liberdade (Adaptado [8])

2.2.2. Modos de vibração

Modos de vibração estão diretamente associados às propriedades de um sistema. Modos de vibração (Ressonâncias) são determinados pelas propriedades da estrutura, entre elas, a sua massa, rigidez e comportamento ao amortecimento [9]. Estes, os modos de vibração, dependem ainda de condições fronteira associada ao sistema. Cada modo de vibração é definido pelo padrão natural de vibração, amortecimento modal e condição de solitação. Se em algum momento, existir uma alteração nas propriedades do material ou até mesmo nas condições fronteira, haverá uma alteração nos modos de vibração do sistema.

Deste modo, e tendo em conta a Figura 4, é possível observar os primeiros 12 modos de vibração de uma estrutura plana fixa numa das extremidades.

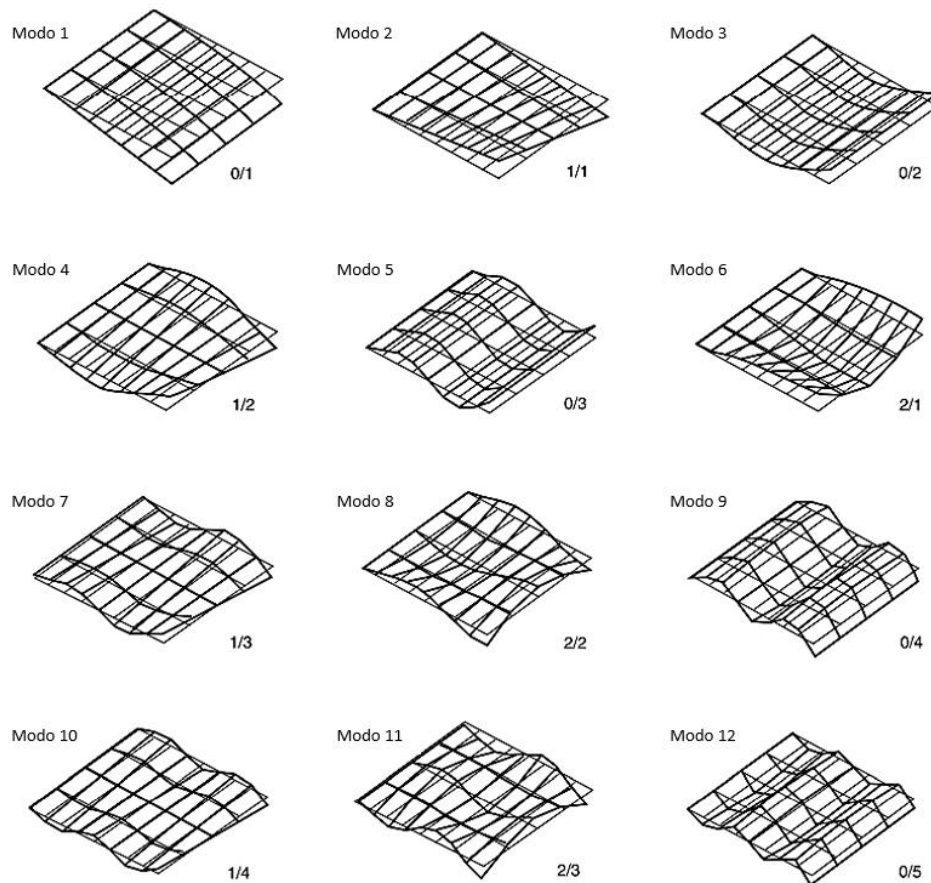


Figura 4 - Modos de Vibração (Adaptado [5])

Associado à oscilação de uma estrutura, é de reparar que se esta não estiver em fase em toda a sua plenitude, existirá um ponto onde a vibração é nula, isto é, ponto nodal. Um ponto nodal distingue-se por ser um ponto de uma estrutura que não oscila para qualquer instante no tempo [4].

Deste modo, através da identificação dos modos naturais de vibração é possível, ainda, identificar os pontos nodais associados à estrutura em análise. Na Figura 5, são notórios os diferentes pontos nodais de uma viga encastrada numa extremidade.

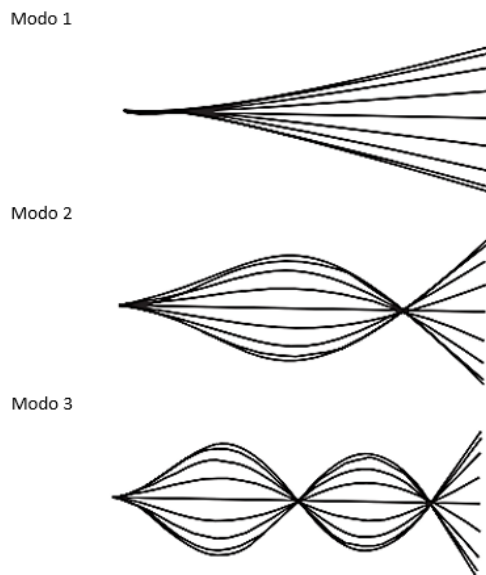


Figura 5 - Pontos Nodais de uma viga encastrada numa extremidade (Adaptado [5])

No que se refere à resposta vibratória de uma estrutura para frequências distintas das de ressonância, esta pode ser obtida através da técnica de sobreposição modal, a qual resulta da contribuição das diferentes formas modais. O grau de contribuição de cada forma modal será função da proximidade da sua frequência natural e das características de excitação da estrutura.

2.2.3. Método de sobreposição modal

Na maior parte dos *softwares* de elementos finitos encontra-se um método destinado à análise e obtenção das equações de movimentos associadas a cargas dinâmicas, método de sobreposição modal. Este método caracteriza-se como sendo uma técnica que consiste no desacoplamento das equações do sistema de movimento e resolução das mesmas. Posteriormente, através do acoplamento das equações resolvidas é possível obter a resposta global do sistema [10].

Deste modo, este método é aconselhado para análise de vibrações de estruturas complexas, isto é, com diversos graus de liberdade.

A boa utilização deste método consiste em, primeiramente, obter as frequências naturais e modos de vibração da estrutura em estudo.

2.2.4. Transformada de Fourier

Métodos como a transformada rápida de Fourier (FFT) podem resolver eficientemente problemas que contêm alguns graus de liberdade [11].

Relativamente à FFT, esta apresenta-se como um algoritmo computacional extremamente eficiente para a Transformada Discreta de Fourier (DFT). No que toca a sinais de vibração usa-se a FFT de uma sequência real (domínio do tempo) para uma sequência complexa (domínio da frequência) [5]. Deste modo, a transformada rápida de Fourier tem sido desenvolvida com o

objetivo de reduzir o tempo computacional e garantir a evolução dos coeficientes DFT. O algoritmo FFT foi desenvolvido pelo matemático Carl Friedrich Gauss e independente, a meio do século 20, pelos físicos James Cooley e John Tukey [12].

No que diz respeito à Transformada Discreta de Fourier (DFT), esta é uma representação de Fourier de uma sequência finita e a transformada em si, também, é uma sequência em vez de uma função contínua de frequência. Deste modo, a DFT é fundamental para o processamento de sinal digital [13].

2.2.5. Função resposta em frequência (FRF)

Para além dos parâmetros modais, obter a FRF de uma estrutura permite uma análise dessa mesma estrutura [14]. Este é um modelo muito eficiente de um sistema linear, é um modelo de domínio de frequência, onde o espectro de saída é definido como o espectro de entrada ponderado por um sistema descrito pela Equação 5.

$$X(\omega) = H(\omega) \cdot F(\omega) \quad (5)$$

Sendo que $X(\omega)$ é a função de entrada, $F(\omega)$ a função de saída e $H(\omega)$ a função transferência que representa a relação entre a entrada e a saída.

Uma FRF é uma medida de quanto deslocamento, velocidade e aceleração de uma estrutura que tem em um grau de liberdade, por unidade de força de excitação [9]. Esta apresenta-se como uma medida fundamental pois isola as propriedades dinâmicas inerentes de uma estrutura e permite assim descrever o comportamento dinâmico de uma estrutura [15].

Um conjunto de medições de FRF é capaz de obter os parâmetros modais experimentais. Esta descreve a relação entrada saída entre dois pontos de uma estrutura de acordo com a sua função frequência, como é visível na Figura 6.

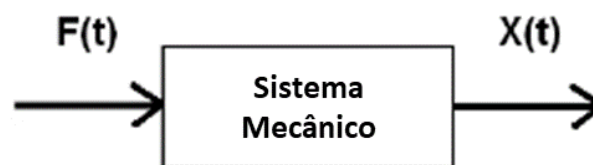


Figura 6 - Diagrama FRF (Adaptado [9])

Assim, é notório que a FRF é obtida a partir da razão entre a Transformada de Fourier da resposta do sistema e a Transformada de Fourier da força aplicada. Esta relação permite avaliar como o sistema responde a diferentes frequências de excitação, independentemente da magnitude da força aplicada. A FRF fornece uma visão clara de como a força de entrada gera uma resposta no sistema. Ao analisar esta razão, é possível identificar as frequências naturais, os modos naturais, e a razão de amortecimento do sistema, permitindo uma caracterização completa e precisa do comportamento dinâmico da estrutura.

2.2.6. Análise modal experimental (EMA)

Os parâmetros de vibração, tais como frequências naturais, modos de vibração e amortecimento são possíveis de obter através de uma técnica intitulada de Análise Modal Experimental. Esta permite obter as propriedades mecânicas de uma estrutura através de uma estimativa dos parâmetros modais [16]. A análise experimental apresenta-se como essencial para a análise dinâmica de uma estrutura uma vez que assim é possível obter as propriedades da estrutura, que através de uma análise analítica seria de uma obtenção mais complexa [17].

Esta análise consiste na excitação de uma estrutura com, por exemplo um martelo de impacto, sendo que, posteriormente, existe a medição da sua função de resposta através do uso de, por exemplo, acelerómetros [18]. Esta técnica caracteriza-se ainda por ser uma técnica não destrutiva [19].

Deste modo, a Análise Modal Experimental consiste num conjunto de procedimentos experimentais que tem como objetivo a construção de um modelo matemático de modo a conseguir descrever o comportamento dinâmico do objeto ou estrutura em estudo. Assim, a utilidade deste método pode ser aplicada de modo a obter os seguintes aspetos [5]:

- Visualização dos modos de vibração da estrutura em análise com o objetivo de obter uma maior compreensão relativamente às propriedades dinâmicas reais da estrutura e, também, uma melhor perceção da estrutura devido à existência de uma aproximação “física” da mesma;
- Comparação da vibração anteriormente medida, e correspondentemente a vibração real, com os parâmetros previstos através de um modelo teórico;
- Detetar e, possivelmente retificar, erros relativos ao modelo teórico;
- Prever o comportamento de estruturas compostas pelo acoplamento de dois ou mais componentes;
- Detecção de danos ou eventuais alterações da integridade de uma estrutura durante a sua vida útil;
- Identificação de parâmetros difíceis de obter ou de prever tais como o amortecimento, os efeitos de atrito dinâmico e excitações realizadas através de forças das quais as fontes são desconhecidas.

Pode-se, ainda, descrever este processo, Análise Modal Experimental, de acordo com os seguintes passos:

- Medição da resposta da estrutura em análise tendo em consideração uma excitação controlada e conhecida;
- Estudo das funções de resposta resultantes, de modo que seja possível identificar as propriedades modais da estrutura em análise;
- Construção de um modelo matemático de acordo com as propriedades anteriormente adquiridas.

Resumidamente, a estrutura é excitada através de um martelo de impacto, sendo que as medições realizadas através de acelerómetros. Este tipo de medição exige, então, um programa para aquisição e tratamento de sinal apropriado. Realizado este tratamento de dados, é obtida, então, a FRF. Por fim, e obtidos os parâmetros modais, é possível obter as características da estrutura, EMA [16]. Através da Figura 7 é perceptível o procedimento da Análise Modal Experimental

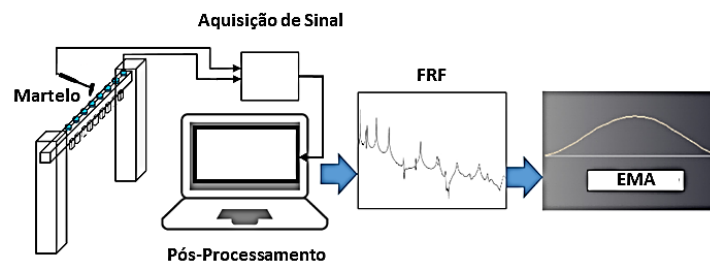


Figura 7 - Procedimento EMA (Adaptado [16])

Em suma, EMA permite a obtenção das características fundamentais de uma estrutura, o que permite uma melhor percepção do comportamento dinâmico da estrutura.

2.3. Equipamentos e funcionamento

De modo que seja possível a obtenção dos parâmetros modais, é necessário existir inicialmente uma excitação da estrutura através de uma determinada solicitação. Posteriormente é necessário a aquisição e tratamento de sinal de modo a obter as características de vibração e, de seguida, as características da própria estrutura [20].

2.3.1. Instrumentos de Vibração

A força de excitação aplicada a uma estrutura pode ser gerada por diferentes aparelhos [6].

Deste modo, podem ser utilizados excitadores conectados, que permanecem acoplados à estrutura durante toda a análise, como por exemplo:

- “Shakers” eletromagnéticos;
- “Shakers” eletrohidráulicos;
- Massas rotativas excêntricas.

Por outro lado, podem, ainda, ser excitadores não-conectados, que após a excitação, separam-se da estrutura. Exemplos destes excitadores incluem:

- Martelo de Impacto;
- Pêndulo;
- Cabo suspenso capaz de produzir “snap-back”.

A título de curiosidade, e tendo em conta uma análise modal de uma estrutura, uma excitação acústica não pode ser utilizada, uma vez que, não é possível obter controlo sobre a direção e ponto de excitação provocado por esta mesma excitação [6].

Relativamente ao martelo de impacto, este apresenta-se como uma das técnicas de excitação mais utilizadas para o método de análise modal experimental [21], [22]. Os martelos de impacto contêm um transdutor de força e um elemento de controlo de rigidez. Este apresenta vantagens e desvantagens relativamente à sua utilização [6].

Assim, as vantagens de um teste realizado por um martelo de impacto são as seguintes:

- Velocidade de teste;
- Não tem necessidade de acessórios;
- Não altera a massa da estrutura;
- Barato e portátil.

No que toca às desvantagens deste tipo de excitador, estas são as seguintes:

- Baixa precisão na obtenção das propriedades não lineares;
- Necessário aplicar uma maior energia quando se pretende analisar uma estrutura grande, sendo que pode existir, assim, uma danificação local da estrutura;
- Permite controlar apenas o limite superior da faixa de frequências que pode ser excitada, limitando a análise detalhada de frequências mais baixas, o que impossibilita uma análise pormenorizada da estrutura.

De destacar ainda que os resultados associados a um ensaio realizado com um martelo de impacto dependem de diversos fatores, como por exemplo a rapidez com que o impacto é aplicado e se deste resultam impactos múltiplos, da firmeza com que este mesmo impacto é realizado e, também, da robustez do corpo do martelo [23]. Assim, o martelo de impacto apresenta-se como uma técnica de excitação totalmente enquadrada ao trabalho. Na Figura 8, encontra-se um exemplo de um martelo de impacto e os seus componentes.

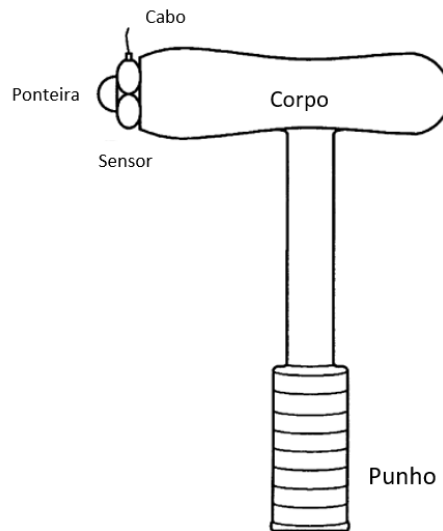


Figura 8 - Típico Martelo de Impacto (Adaptado [23])

2.3.2. Acelerómetros

De modo a obter a informação dinâmica da estrutura em estudo, em muitos estudos são utilizados acelerómetros, uma vez que estes permitem a obtenção da informação dinâmica num determinado ponto [17]. Estes podem apresentar diferentes características consoante o tipo de acelerómetro que esteja a ser utilizado. Como por exemplo os acelerómetros MEMS e acelerómetros piezoelétricos [23], [24].

Estes aparelhos apresentam uma vasta gama de frequências de utilização (0,5 Hz a 20kHz) e para frequências mais baixas, até 50 Hz, podem ser utilizados sensores lineares especiais de modo a garantir uma elevada precisão. Devido à elevada gama de frequências lineares e de amplitudes dinâmicas, os acelerómetros caracterizam-se por serem bastante versáteis [23]. Na Figura 9 é possível observar a relação entre a sensibilidade e a frequência de um acelerómetro.

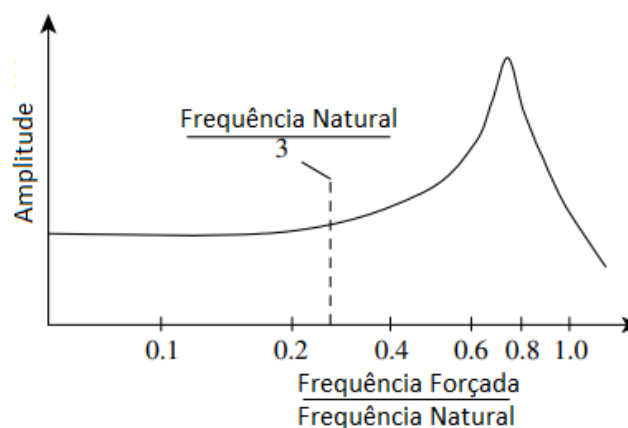


Figura 9 - Relação entre Sensibilidade e Frequência de um Acelerómetro (Adaptado [23])

De destacar ainda que os acelerómetros têm como principal função a conversão de uma quantidade física, associada ao impacto realizado na estrutura, numa quantidade elétrica, sendo depois esse sinal transmitido ao *hardware* e posteriormente analisado [20].

No que diz respeito à medição da resposta, qualquer um dos parâmetros de movimento (deslocamento, velocidade ou aceleração) pode ser obtido. Deste modo, características essenciais de modo a escolher a melhor opção para o transdutor são, por exemplo, boa linearidade, massa desprezável (de modo que este não influencie o peso associado à estrutura), intervalo dinâmico elevado, baixa sensibilidade transversal, intervalo de frequências elevado e *design* simples [6].

Assim, a colocação de acelerómetros na estrutura em estudo permite, para o método EMA, uma aquisição do sinal de vibração [25]. Esta colocação é a primeira tarefa a realizar-se numa análise dinâmica, sendo que esta escolha pode variar com diversos fatores. Nomeadamente a experiência do operador, simulação prévia de elementos finitos ou previsão da resposta dinâmica da estrutura [20]. Na Figura 10 é notório um exemplo da colocação de 12 acelerómetros para a realização desse mesmo efeito.

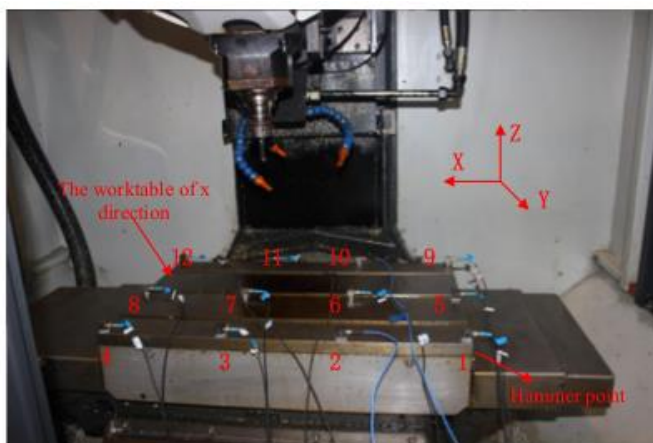


Figura 10 - Colocação de Acelerómetros [25]

2.3.3. Placa de aquisição de sinal

A placa de aquisição de sinal a utilizar no estudo da estrutura proposta será o NI USB-4432, existente no LMA, do DEM, situado no ISEP. Este é um dispositivo desenvolvido pela *National Instruments*®, capaz de fazer o levantamento do sinal de vibração numa estrutura. Este contém 5 canais de entrada e, apresenta uma gama dinâmica elevada, sendo o rácio entre o sinal e o ruído de 100 dB. Caracteriza-se, ainda, por apresentar uma taxa de aquisição máxima, ajustável aos necessários em laboratório, de 102.4 kS/s [26]. Na Figura 11 encontra-se representado o modelo referido anteriormente e fornecido pela *National Instruments*®



Figura 11 - NI USB-4432 [26]

De frisar que, o canal de saída analógica apresenta-se como ideal para ensaios de impacto e consequente resposta por parte da estrutura.[27].

2.4. Programas

Tendo em conta o propósito de desenvolver um programa de aquisição e tratamento de sinal em fonte aberta (*opensource*) descrevem-se abaixo os diferentes programas inseridos neste mesmo propósito. Ou seja, tanto relativamente à recolha e análise de dados tendo como base programas como *LabVIEW* e *MATLAB* e posteriormente a programação em código aberto e com menores limitações na linguagem *Python*.

2.4.1. LabVIEW

O programa *LabVIEW* (*National Instruments*®), acrónimo para *Laboratory Virtual Instrument Engineering Workbench*, é um software para aquisição de dados e controlo dos mesmos. Este tem como base um programa que envia, recebe e integra dados de e para laboratórios equipados com um GPIB (*General Purpose Interface Bus*) tendo posteriormente passado para um sistema capaz de aceitar dados em formato digital ou analógico e realizar análise de dados e controlo de instrumentos. De destacar, ainda, que foi o primeiro programa a incluir técnicas de programação gráfica, que tornam a programação mais perceptível e transparente para o usuário e uma sequência de processamento visível para o mesmo [28].

O *software* em causa apresenta-se como tendo diversas vantagens quando comparado com outras linguagens de programação. A maior vantagem deste programa é a sua rica interface gráfica do utilizador (GUI) e, ainda, *drivers* de *hardware* [29].

Este programa inclui uma vasta panóplia de funções matemáticas, estatísticas e de processamento de sinal digital. Na execução do mesmo, esta não é controlada pela ordem em que as instruções foram introduzidas, mas sim pelos dados que são gerados. A título de exemplo, sub-rotinas que usam dados provenientes de uma placa de conversão como entrada não são executados até que os dados tenham sido totalmente adquiridos. De destacar, ainda, que todos os elementos básicos de linguagem de programação estão implementados no programa. A criação de um programa *LabVIEW* inicia-se com a determinação das variáveis de entrada, sendo estas colocadas no painel frontal. No mesmo sentido, as saídas do programa

são, também, colocadas no painel frontal, sendo que a saída se pode encontrar na forma numérica, textual ou gráfica [28].

2.4.2. MATLAB

O programa *MATLAB* (*MathWorks*®) é uma plataforma de programação desenvolvida para atender as diversas necessidades nos ramos de ciência e engenharia, principalmente. Este tem como objetivo possibilitar a análise e *design* de sistemas. Através deste programa, é possível a análise de dados, desenvolver algoritmos e a criação de modelos e aplicações. Este software permite uma autonomia por parte do utilizador, sendo possível a implementação das suas ideias em aplicativos corporativos e dispositivos incorporados ao programa [30].

A utilidade deste programa é de uma enorme variedade, tanto a nível académico como nível industrial. Deste modo, este programa utiliza-se em inúmeras áreas científicas, nomeadamente para o estudo do intervalo de batimentos cardíaco [31] e, ainda, para o estudo de sinais sísmicos [32].

Por fim, tanto o *MATLAB* como o *LabVIEW* são capazes da realização da aquisição e tratamento de sinal, contudo a identificação dos parâmetros modais da estrutura requer uma análise muito mais complexa com a intervenção dos utilizadores. Pois, o programa a ser desenvolvido deve possuir essas mesmas características, para que a análise a realizar se possa ajustar a cada de tipo de problema.

2.4.3. Python

A linguagem *Python* é uma linguagem de programação interpretada, uma vez que um código realizado nesta linguagem é executado diretamente pelo utilizador, e de alto nível. Devido às suas estruturas de dados incorporadas, combinadas com digitação e ligação dinâmica, fazem com que esta linguagem se destine ao desenvolvimento rápido de aplicativos ou apenas como conector para componentes existentes. Esta linguagem caracteriza-se como sendo simples e de fácil aprendizagem. O interpretador *Python* e a sua extensa biblioteca estão disponíveis gratuitamente, na forma código fonte ou binário para todas as principais plataformas [33]. Deste modo, esta linguagem, e respetivas bibliotecas associadas, capazes de serem utilizadas para problemas de ordem numérica e de previsão [34]

Esta linguagem apresenta-se como uma alternativa ao *MATLAB* apresentando diversas vantagens como ser de fonte aberta, licença gratuita, rápida apresentar uma gestão de memória eficiente e acessível a diversas plataformas. As bibliotecas *NumPy*, *SciPy* e *Matplotlib* conferem-lhe a possibilidade de trabalhar em aplicações científicas e numéricas [35].

A biblioteca *NumPy*, *Numeric Python* ou *Numerical Python*, tem como principal objetivo lidar com matrizes permitindo, assim, a utilização das funções predefinidas para a realização de cálculos numéricos e matemáticos. Esta biblioteca destina-se ao cálculo matricial, sendo deste modo possível a realização de operações algébricas lineares e operações algébricas mais complexas, o que permite assim ser utilizada como base de cálculo e ser a biblioteca responsável pela obtenção da Transformada de Fourier [36], [37].

No que diz respeito a SciPy ou ScientificPython, destina-se ao processamento de sinal, processamento de imagens, aplicação de um programa de integração e operações de interpolação.[35].

A biblioteca Matplotlib permite criar gráficos a partir de dados, sendo relevante para este programa a representação gráfica dos dados adquiridos no tempo, dos sinais em frequência e dos modos naturais de vibração.[35].

Assim, a linguagem *Python* apresenta-se como uma mais-valia pois possibilita uma permanente atualização, e uma capacidade de se ajustar a diferentes cenários.

Revisão bibliográfica

3. Desenvolvimento de programa para determinação dos parâmetros modais

3.1. Introdução

Neste capítulo descreve-se o código desenvolvido em linguagem *Python* para a aquisição e processamento de sinais, tendo como primeiro objetivo medir as FRF de uma estrutura. Posteriormente, descreve-se o programa de análise das FRF, responsável pela identificação das frequências e modos naturais de vibração e sua representação gráfica.

Este encontra-se dividido em duas sub-rotinas, programa de aquisição e processamento e programa de análise, sendo que a escolha da sub-rotina é realizada a partir de um programa principal. A lista do programa principal pode ser consultada no apêndice A, a do programa de aquisição e processamento no Apêndice B e por último o programa de análise no Apêndice C, estando também acessível através do link https://github.com/mqueiros018/modal_analysis. Na Figura 12 pode-se observar o código da rotina principal.

```
import aquisicao
import analise

# definir as sub-rotinas
def final():
    print("Executar o Programa de Aquisição e Processamento")

def analise():
    print("Executar o Programa de Análise")

# função principal para as sub-rotinas
def main():
    aquisicao() # executa o programa de aquisição e processamento
    analise() # executa o programa de análise

# função principal
if __name__ == "__main__":
    main()
```

Figura 12 - Rotina principal

Desenvolvimento de programa para determinação dos parâmetros modais

Primeiramente, teve de se instalar as bibliotecas necessárias para o desenvolvimento do programa, nomeadamente *nidaqmx*, *numpy*, *matplotlib* e *scipy*. Assim, através da Figura 13, é possível observar o comando utilizado para instalar as bibliotecas usadas pelo programa.

```
pip install nidaqmx
pip install numpy
pip install matplotlib
pip install scipy
```

Figura 13 - Instalação das bibliotecas

Depois de instaladas todas as bibliotecas necessárias passou-se ao desenvolvimento dos programas descritos anteriormente.

Relativamente ao programa de aquisição e processamento, este tem como objetivo a obtenção das FRF de modo que seja possível a sua posterior análise. Na Figura 14 encontra-se um diagrama de blocos com as diversas secções presentes neste programa. Inicialmente é realizada a aquisição de sinal através da placa de aquisição da NI, sendo depois necessário o pré-processamento dos sinais adquiridos. Este envolve a truncagem do sinal e posterior ponderação do sinal no tempo. De seguida, os sinais são processados com a determinação do correspondente espectro em frequência, por aplicação da transformada de Fourier. A combinação dos sinais espectrais da excitação e da resposta permite a obtenção da função de transferência do sistema ou da FRF. Por fim, os dados são armazenados no formato txt.

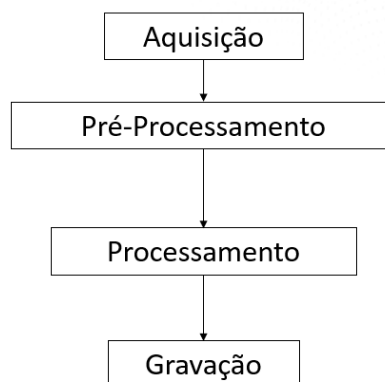


Figura 14 – Diagrama de Blocos do Programa de Aquisição e Processamento

Para além do programa anteriormente descrito, desenvolveu-se uma outra sub-rotina, designada de programa de análise. Na Figura 15 apresenta-se o diagrama de blocos, que descreve as principais etapas deste programa. Este começa por ler as FRF medidas pelo programa de aquisição e processamento. No pós-processamento é realizada a análise dos valores de pico da amplitude das FRF, a partir dos quais são extraídas as frequências naturais de vibração. A partir das frequências naturais são extraídas as amplitudes da componente imaginária das FRF e combinado com a distribuição espacial dos pontos de medição são contruídos os modos naturais de vibração, os quais são representados na forma gráfica. Por último, a informação com as frequências e os modos naturais de vibração são gravados no formato de formato txt.

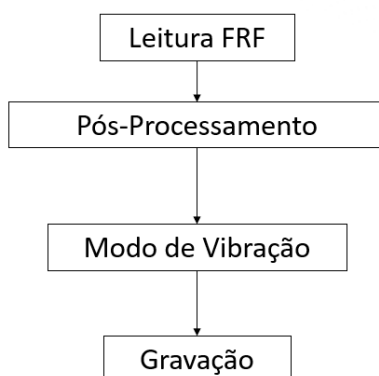


Figura 15 - Diagrama de Blocos do Programa de Análise

3.2. Programa de aquisição e processamento

3.2.1. Aquisição

A primeira etapa da aquisição envolve o registo de sinais produzidos pelo martelo de impacto e acelerómetro. Para esse efeito, posicionou-se um acelerómetro numa das extremidades, responsável pela leitura da resposta associada ao impacto. A sua validação foi realizada por comparação com os mesmos sinais registados num osciloscópio.

Para a aquisição de sinal foi utilizada a placa NI USB4432. Assim, foi necessário usar a biblioteca *nidaqmx*, que permite a configuração dos canais utilizados e do tempo do ensaio. Deste modo, criou-se uma função, *readdaq*. Esta tem como objetivo o interface entre o aparelho NI e o *script* desenvolvido de modo que o sinal gerado seja então adquirido. Para que este funcione na sua plenitude é necessária a definição de alguns parâmetros de aquisição, nomeadamente o tempo de leitura, frequência de leitura e por consequência o número de leituras a realizar, resultado da multiplicação do tempo de leitura pela frequência de leitura. É ainda necessário definir a sensibilidade associada ao sensor e a corrente de excitação. Assim, é possível a obtenção de dados de leitura, uma vez definidos estes parâmetros.

Para além da definição dos parâmetros anteriores, é necessário também, a configuração dos canais responsáveis pela aquisição do sinal de impacto. Sendo que neste caso, foram utilizados 2 canais da placa de aquisição de sinal. No que diz respeito ao canal 0, este foi configurado para o martelo de impacto. Já quanto ao canal 1, este configurou-se para efetuar a ligação do acelerómetro.

No que diz respeito ao canal correspondente à ligação da placa ao martelo de impacto, canal 0 da placa, este configurou-se de modo a usar um sensor de força IEPE para medir a carga aplicada pelo martelo. Tendo em conta esta nuance, teve de se atribuir parâmetros à configuração do canal, tais como as unidades da carga, intervalo dessa mesma carga e ainda a sensibilidade, e respetivas unidades, do sensor.

Desenvolvimento de programa para determinação dos parâmetros modais

O canal 0 configurou-se para medir a força em Newton, sendo, no entanto, a entrada de sinal realizada em tensão, e com uma gama de carga entre -10 e +10 V. A sensibilidade do martelo foi configurada para 1000 mV/N sendo este um valor de referência de forma a facilitar a aquisição e posterior processamento dos dados. Por fim, o valor de corrente de excitação foi estabelecido como 2,1 mA, sendo este valor o indicado para o correto funcionamento do sensor IEPE, segundo a NI.

Já quanto ao canal 1, canal configurado tendo em conta o uso do acelerómetro, consideram-se diversos aspetos importantes que garantissem o bom funcionamento do acelerómetro e consequentemente do canal correspondente. A sensibilidade definiu-se como 1000 mV/g, de modo que a conversão dos sinais da aceleração fosse precisa. Relativamente às unidades de medição, esta encontra-se definida em g, pratica comum na utilização de sensores de aceleração sendo posteriormente realizado o processamento destes dados para m/s^2 , unidade de aceleração mais comum e mais perceptível numa futura análise.

Para além disso, este canal, o canal 1, foi configurado de modo a trabalhar com uma referência em dB de 1, de forma a garantir uma elevada precisão na leitura de sinais com baixa amplitude. A configuração do acoplamento fora definida como AC, permitindo a filtragem dos componentes DC presentes no sinal. A excitação do sensor foi também estabelecida para uma corrente interna de 2,1 mA e o intervalo de medição encontra-se ajustado para um intervalo de -10 a +10 V, de forma a assegurar uma ampla gama de valores de aceleração. Por último, a configuração do terminal definiu-se como pseudodiferencial mantendo um equilíbrio entre a rejeição do ruído e a simplicidade do circuito.

Por fim, e terminadas as configurações dos canais associados ao martelo e ao acelerómetro foi necessário programar a taxa de aquisição e respetivos parâmetros para esta função de modo a garantir a correta amostragem dos sinais. No que diz respeito à taxa de aquisição, esta configurou-se tendo em conta a frequência de aquisição definida, de modo a garantir que os sinais fossem armazenados com a resolução temporal adequada à análise em estudo. A configuração da amostragem encontra-se definida como contínua permitindo a leitura contínua dos dados sem interrupções, sendo que o número de amostras por canal posteriormente calculado.

Para além dos parâmetros até agora definidos, o sinal de *clock* de amostragem foi ajustado para a ativação no flanco ascendente, garantindo a sincronização precisa dos dados obtidos da aquisição. Após a configuração da taxa de aquisição, a tarefa de aquisição inicia-se, os dados da leitura são armazenados sendo posteriormente a tarefa encerrada, finalizado o tempo de aquisição. Este procedimento garante que os dados são adquiridos adequadamente e de forma contínua, durante o intervalo de tempo que a função de aquisição está em funcionamento.

Toda a programação efetuada para a realização da função acima descrita encontra-se na Figura 16.

```

def readdaq(N_pedido, freq_aquisicao_ped):
    Analog_read_task = nidaqmx.task.Task('Analog_read_task') # Iniciar o DAQ device
    (Tarefa)
    # configuração canal 0
    ai_channel = Analog_read_task.ai_channels.add_ai_force_iepe_chan(
        'Dev1/ai0', name_to_assign_to_channel='force',
        terminal_config=TerminalConfiguration.DEFAULT,
        units=constants.ForceUnits.NEWTONS,
        min_val=-10,
        max_val=+10,
        sensitivity=1000,

    sensitivity_units=constants.ForceIEPESensorSensitivityUnits.MILLIVOLTS_PER_NEWTON,
    current_excit_val=2.100000e-3,
    custom_scale_name='')

    # configuração canal 1
    ai_channel_1 = Analog_read_task.ai_channels.add_ai_accel_chan(
        'Dev1/ai1', name_to_assign_to_channel='acel',
        units=constants.AccelUnits.G, sensitivity= 1000,
        sensitivity_units=constants.AccelSensitivityUnits.MILLIVOLTS_PER_G)
    ai_channel_1.ai_accel_db_ref=1
    ai_channel_1.ai_coupling=constants.Coupling.AC

    ai_channel_1.ai_excit_voltage_or_current=constants.ExcitationVoltageOrCurrent.USE_CU
    RRENT
    ai_channel_1.ai_excit_src=constants.ExcitationSource.INTERNAL
    ai_channel_1.ai_excit_val=0.0021
    ai_channel_1.ai_max=+10
    ai_channel_1.ai_min=-10
    ai_channel_1.ai_term_cfg=constants.TerminalConfiguration.PSEUDO_DIFF
    ai_channel_1.ai_accel_units.METERS_PER_SECOND_SQUARED

    # definir taxa
    Analog_read_task.timing.cfg_samp_clk_timing(rate = freq_aquisicao_ped,
    sample_mode=constants.AcquisitionType.CONTINUOUS,
        samps_per_chan = N_pedido)
    Analog_read_task.sample_clk_active_edge=constants.Edge.RISING
    Analog_read_task.samp_timing_type=constants.SampleTimingType.SAMPLE_CLOCK

    Analog_read_task.start()

    valores_lidos_aq =
    Analog_read_task.read(number_of_samples_per_channel=int(N_pedido))
    Analog_read_task.stop()

    Analog_read_task.close() # Encerrar a tarefa

    return valores_lidos_aq # devolve os valores obtidas através da placa de aquisição
    
```

Figura 16 - Função de aquisição

Desenvolvimento de programa para determinação dos parâmetros modais

De destacar que, os sinais adquiridos são representados por vetores, sendo que cada vetor possui uma dimensão resultante do produto entre o período e a taxa de aquisição. Assim, é gerada uma matriz de valores temporais que indica a duração total do tempo de aquisição.

Seguidamente, é necessário realizar o armazenamento dos dados previamente adquiridos. Este é efetuado através de uma função denominada *writefiledata* que guarda num ficheiro de texto a informação como os valores temporais, frequência, número de amostras associados à aquisição em questão. De frisar que, na criação do ficheiro de texto, e com o auxílio da biblioteca *time*, o nome do ficheiro é gravado com a data do momento em que a função foi executada. Assim, na Figura 17 pode-se observar a criação do ficheiro de gravação.

```
named_tuple = time.localtime() # get struct_time
time_string = time.strftime("%Y_%m_%d_%H_%M_%S", named_tuple)
nome_ficheiro_resultados = 'Resultados_p_NI_USB_4432_' + time_string + '.txt' # Título
criado e indexado ao tempo (nome unico)
```

Figura 17 – Criação do ficheiro de gravação

Os sinais adquiridos são também representados na forma gráfica. Numa fase de validação, estes foram registados em paralelo num osciloscópio com o objetivo de servir de base de comparação com os obtidos através da placa de aquisição e, assim, poder validar o código implementado.

Com o intuito de continuar esta validação, realizou-se um teste experimental de modo a validar o código desenvolvido por comparação com um programa já existente em *MATLAB*. Assim, foi fornecido um sinal de impacto conhecido ao programa de forma a ser possível a validação das etapas seguintes, com um tempo de amostragem de 2 s e uma frequência análise de 400 Hz. Na Figura 18 encontra-se um registo no tempo da força de impacto.

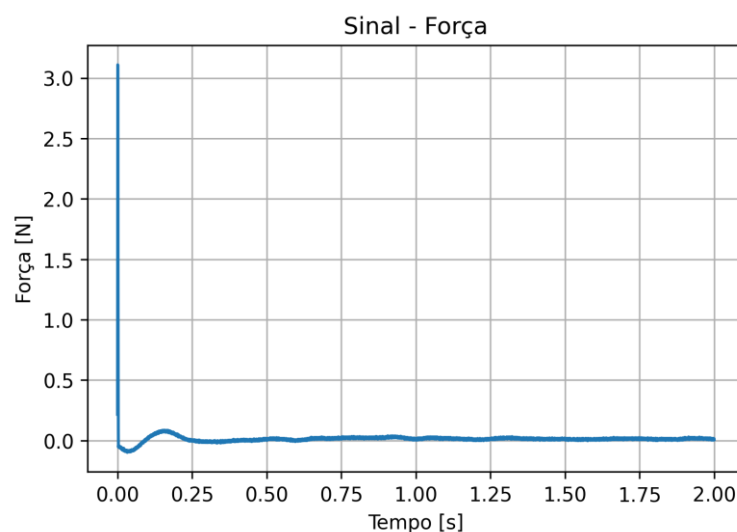


Figura 18 - Sinal completo da Força

Na Figura 19 encontra-se o detalhe da evolução da força nos primeiros instantes. Com este detalhe é mais saliente o instante do impacto.

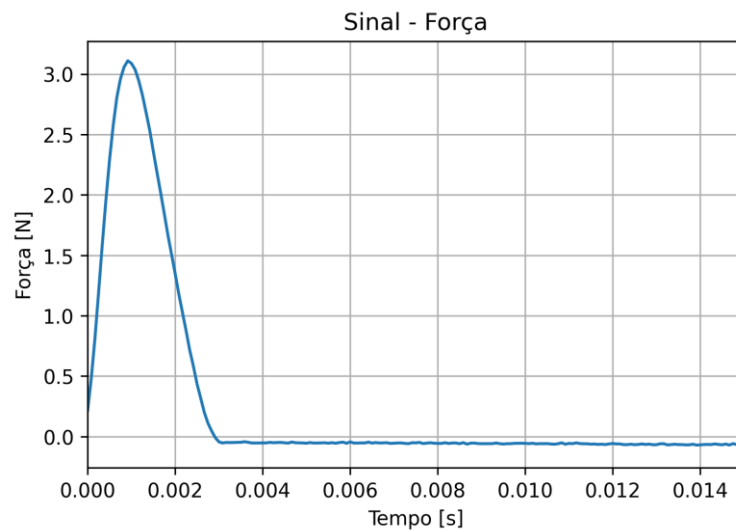


Figura 19 - Sinal da Força truncado no tempo

Para o mesmo intervalo de tempo e com início no mesmo instante, apresenta-se na Figura 20 a evolução da resposta da aceleração em vibração livre.

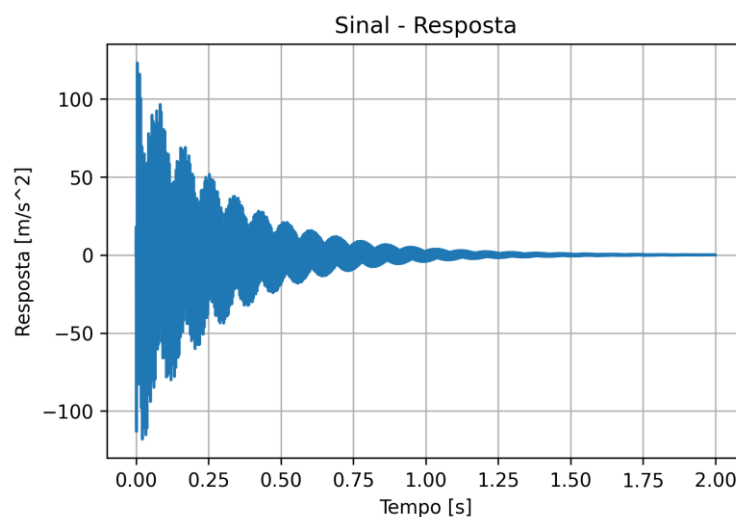


Figura 20 - Sinal da Resposta

3.2.2. Pré-processamento

Uma vez que, a placa de aquisição, na função definida, desde o instante 0 até ao instante final definido como *time out*, guarda todos os dados independentemente se existiu impacto ou em que momento este aconteceu. Deste modo, surgiu a necessidade de truncar o sinal desde o momento do impacto até um determinado instante de tempo a definir.

Relativamente a este aspeto, e tendo em conta que não foi possível a utilização da função *trigger* disponível na biblioteca *nidaqmx*, este realizou-se de forma manual, através da definição de um *trigger level* que permitiria a identificação do pico resultante do impacto do martelo na geometria em análise. De destacar que os dados obtidos através do impacto realizado pelo

Desenvolvimento de programa para determinação dos parâmetros modais

martelo, se correto, no canal correspondente, apenas teria um pico. Na existência de 2 ou mais revelaria o insucesso do impacto realizado.

Tendo em conta todos estes aspetos, definiu-se então um *trigger level* em tensão (uma vez que o sinal obtido pelos canais continua, até então, em tensão) e após identificação do pico resultante do impacto, o programa guarda o número de pré amostras definido anteriores ao índice do pico até ao total de amostras. Em caso de erro ou de não acontecimento do impacto, o programa lança um aviso. Na Figura 21 encontra-se a programação da função de deteção de impacto.

```
def detetar_impacto(data, threshold):
    # usar numpy.where para encontrar os índices onde o valor excede o threshold
    indices = np.where(data > threshold)[0]
    # print(indices)

    # verificar se algum índice foi encontrado e retornar o primeiro, caso contrário
    retornar -1
    if indices.size > 0:
        return indices[0] # devolve o primeiro índice encontrado
    else:
        return -1 # devolve o valor -1 se nenhum valor exceder o threshold
def processar_dados(valores_lidos, threshold_impacto, num_pre_samples):
    i_impacto = detetar_impacto(valores_lidos[0], threshold_impacto)
    if i_impacto != -1:
        start_index = max(i_impacto - num_pre_samples, 0)
        valores_lidos = valores_lidos[:, start_index:]
        # print(f"Start index: {start_index}")
    else:
        print("Impacto não detetado")
    return valores_lidos, i_impacto
```

Figura 21 - Função de deteção do impacto

Programado na totalidade os parâmetros de aquisição e configurações do canal foi apenas necessário definir os parâmetros para a leitura dos parâmetros anteriormente descritos. Assim foi estabelecido um *time out*, isto é, durante o intervalo de tempo definido o programa irá guardar os dados independente do instante em que é realizado o impacto do martelo. Para isto utilizou-se uma frequência de análise de e uma respetiva frequência de aquisição, obtida através da multiplicação da frequência de aquisição por 1,5 e por 10. Este cálculo garante que a frequência de análise é 15 vezes maior que a frequência de aquisição, que garante uma análise de dados realizada com uma resolução temporal melhorada, capturando o comportamento do sinal com precisão.

Definidos estes parâmetros calculou-se, também o número de amostras que se iria obter da leitura. Este é obtido através da multiplicação do tempo de amostragem pela frequência de aquisição.

De seguida, os dois sinais são ponderados. A função ponderação no tempo do tipo retangular de amplitude unitária é aplicado ao sinal da força para o intervalo correspondente ao impacto do martelo com a estrutura, sendo no restante intervalo imposta amplitude nula. Pretende-se através desta função de ponderação eliminar da análise os artefactos que surgem no sinal adquirido, os quais têm como principal origem o movimento oscilatório do próprio martelo e o ruído eletrónico. Deste modo, na Figura 22 pode-se observar a programação da função de ponderação da força.

```
threshold_estabilizacao = 0.1

# encontrar o índice de estabilização
indice = None
for i in range(amostras):
    i = i + num_pre_samples
    if abs(sinal_forca[i]) < threshold_estabilizacao:
        indice = i+1
        break

# verificar se encontramos um índice de estabilização
if indice is not None:

    # criar ponderacao_forca e definir valores após o índice de estabilização como zero
    ponderacao_forca = np.ones(amostras)
    ponderacao_forca[0] = 0
    ponderacao_forca[indice:] = 0
else:
    print("Nenhum índice de estabilização encontrado.")
```

Figura 22 - Função de ponderação da força

Na Figura 23 encontra-se representado o sinal da força ponderado.

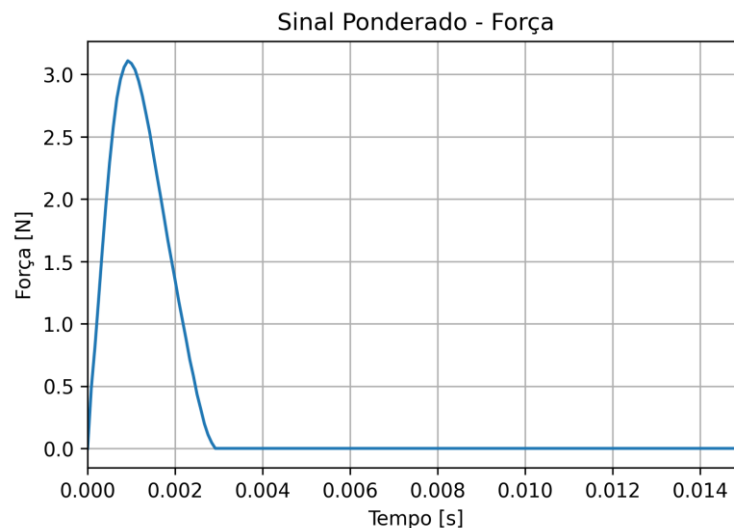


Figura 23 - Sinal da Força Ponderado

Desenvolvimento de programa para determinação dos parâmetros modais

A título de comparação, encontra-se na Figura 24 representado o sinal da força antes da ponderação, o sinal da força após efetuada a ponderação e a função de ponderação do sinal.

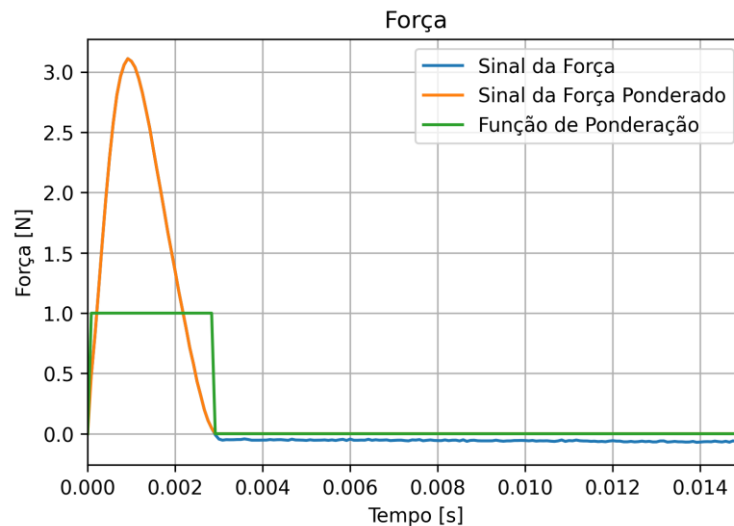


Figura 24 – Sinal no tempo da força não ponderado, força ponderada e a função de ponderação

No que se refere à resposta em regime livre, esta caracteriza-se por apresentar uma amplitude no tempo que descreve segundo uma função logarítmica. Como tal, deve ser uma função de ponderação como o mesmo decaimento e apresentar no final uma amplitude nula. Através desta função de ponderação pretende-se garantir a repetibilidade do sinal amostrado, evitando o efeito de *Leakage*, que causa a distribuição de energia de uma frequência específica para frequências vizinhas, dando origem a uma representação espectral imprecisa. Na representação do sinal no domínio da frequência, este traduz-se pela atenuação e suavização das amplitudes nas ressonâncias.

```
# ponderação resposta
d = 30 # frequencia de declive Hz
ponderacao = np.ones(amostras)
ponderacao[int(0.9 * amostras):] = 1 * np.exp(-d * valores_de_tempo_frf[:int(0.1 *
amostras)])
ponderacao[-10:] = 0
```

Figura 25 - Função de ponderação da aceleração

Na Figura 26 está representado o sinal da aceleração ponderado.

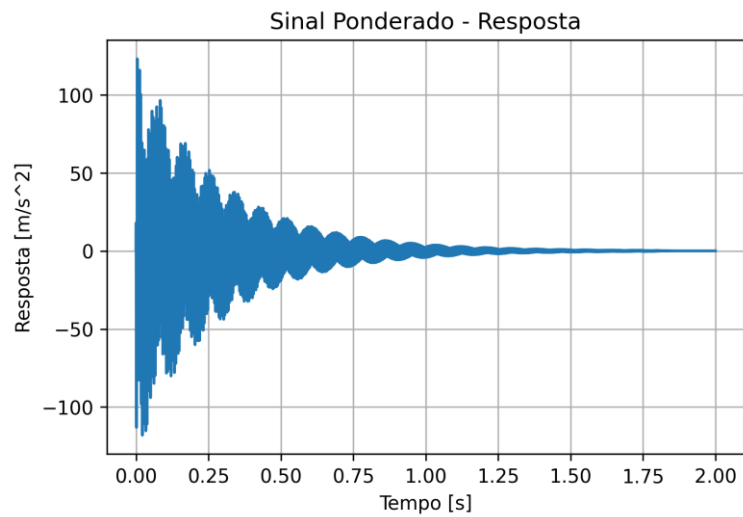


Figura 26 - Sinal da aceleração ponderada

A título de comparação, encontra-se na Figura 27 representado o sinal no tempo da aceleração antes da ponderação, a aceleração ponderada e a função de ponderação.

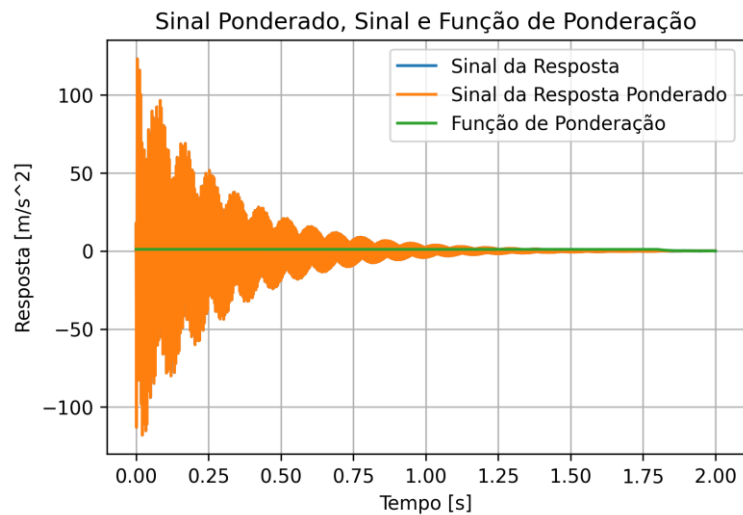


Figura 27 – Sinal no tempo da aceleração não ponderado, da aceleração ponderada e a função de ponderação

Uma vez que, tendo em conta a gama de valores associado à resposta não é possível analisar, através da Figura anterior, a função ponderação na sua plenitude, apresenta-se na Figura 28 a representação gráfica da função ponderação da aceleração.

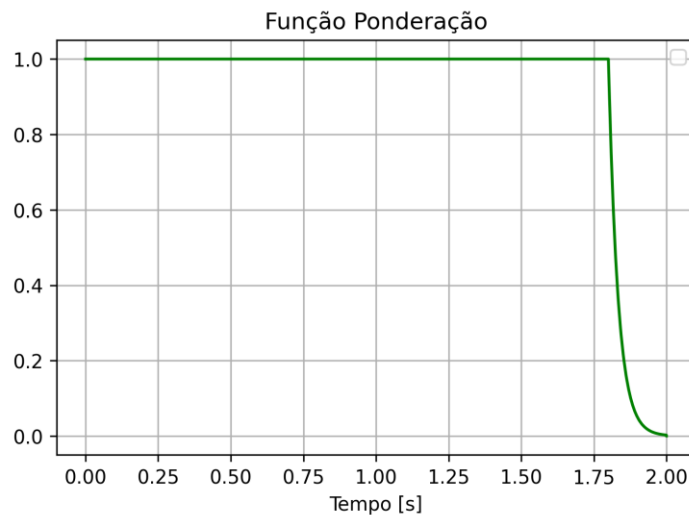


Figura 28 – Representação gráfica da função ponderação da aceleração

3.2.3. Processamento

Após a aquisição, truncagem e ponderação do sinal, este foi processado com o objetivo de se obter a respectiva FRF. Primeiramente, e uma vez que os dados de entrada se encontram em tensão, dividiu-se a amplitude do sinal em tensão correspondente à força e à aceleração pela respectiva sensibilidade associada a cada transdutor. A partir dos sinais ponderados são determinados a respectiva resposta espectral por aplicação da FFT. Através da Figura 29, encontra-se a função responsável pela obtenção da FFT.

```
def calcular_fft(sinal_ponderado, amostras, freq_max):  
    fourier = np.fft.fft(sinal_ponderado)  
    amplitude = np.abs(fourier)[:indice_freq + 1] ** 2  
    fase = np.angle(fourier)[:indice_freq + 1]  
    freq = np.arange(0, freq_max/2 + 1, 1/amostragem)  
  
    return fourier, freq, amplitude, fase
```

Figura 29 - Função FFT

Os autoespectros são determinados multiplicando o espectro pelo seu complexo conjugado, sendo que o da força e da aceleração são representados, respectivamente, nas Figura 30 e Figura 31.

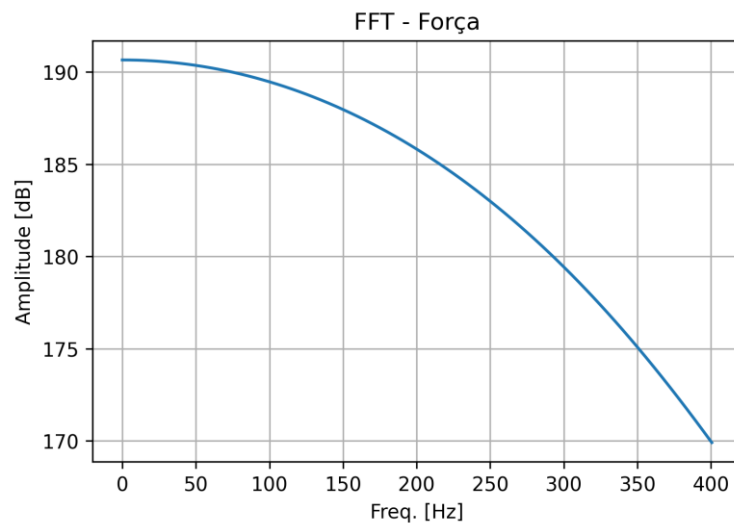


Figura 30 – Autoespectro da força

No que diz respeito ao autoespectro da aceleração representado na Figura 31, já é possível verificar o comportamento da estrutura ao longo das frequências e deste modo identificar as frequências naturais, dentro da gama de frequências de análise, entre 0 Hz a 400 Hz.

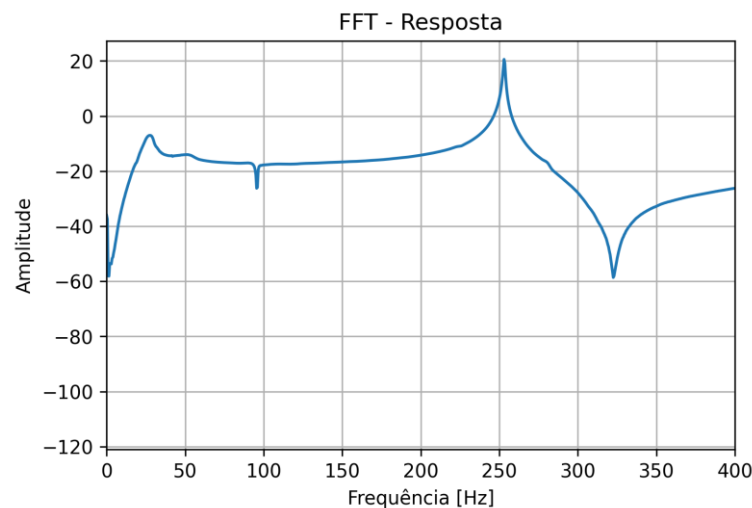


Figura 31 – Autoespectro da aceleração

Analisando as transformadas de Fourier obtidas, é possível observar que, no autoespectro da força, verifica-se que a amplitude varia com a frequência, característica típica de uma excitação do tipo transiente. Quanto à amplitude do autoespectro da aceleração apresenta um valor de pico próximo dos 260Hz e uma antirressonância nos 320Hz. Contudo, esta amplitude da resposta terá de ser corrigida, uma vez que a força de excitação não foi uniforme em toda a gama de frequências, conforme se observa na Figura 30 Para a determinação da FRF entre o ponto de excitação e o ponto de medição da resposta foi usado o estimador do tipo 1, definida,

$$H_1(\omega) = \frac{G_{xy}(\omega)}{G_{xx}(\omega)} \quad (6)$$

Desenvolvimento de programa para determinação dos parâmetros modais

Este estimador foi usado por permitir atenuar o ruído no sinal da resposta, o qual se considera ser mais preponderante entre os dois sinais medidos. Assim, na Figura 32 encontra-se a FRF obtida através do processamento anteriormente descrito.

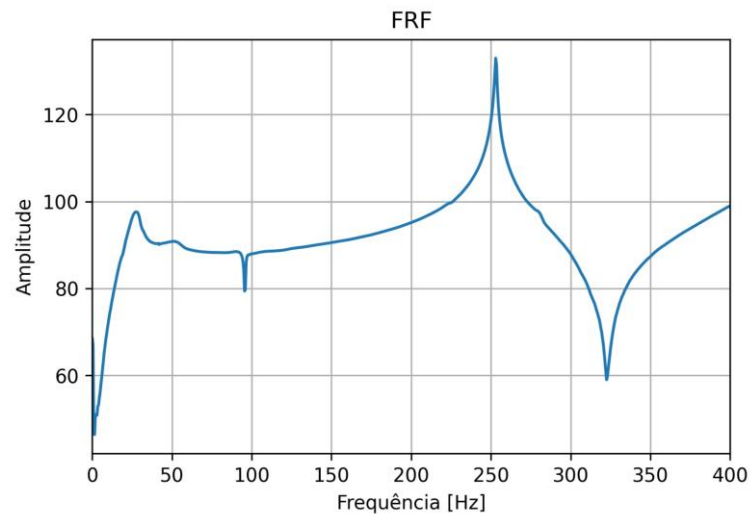


Figura 32 –FRF medida através do programa desenvolvido

Por fim e verificado todo este processo, criou-se um sistema que permite ao utilizador definir o número de medições pretendidas. Após cada medição, o utilizador deve confirmar se pretende ou não guardar os dados, tendo em conta a qualidade dos mesmos. Depois de confirmadas todas as medições introduzidas anteriormente pelo utilizador, apresenta-se a FRF média obtida do ensaio e os dados são armazenados. Deste modo, na Figura 33 encontra-se a função de gravação dos dados obtidos experimentalmente.

```
# guardar os dados  
dados = np.column_stack((np.abs(freq_resposta), np.real(frf_media[:indice_freq +  
1]), np.imag(frf_media[:indice_freq + 1])))  
np.savetxt(nome_ficheiro_resultados, dados, delimiter=',', header='Frequência, FRF - parte  
real, FRF - parte imaginária', comments='')
```

Figura 33 - Função de gravação de dados

3.3. Programa de análise

3.3.1. Leitura de dados

Uma vez que as FRF são obtidas através do programa anteriormente explicado, é necessário realizar a leitura dos dados obtidos através do programa de aquisição e processamento. Deste modo, o programa de análise lê os resultados, armazenados em formato txt, relativos às FRF do caso de estudo.

```

def ler_frf(arquivo):
    with open(arquivo, 'r') as f:
        linhas = f.readlines()

    freq_resposta = []
    valores_reais = []
    valores_imaginarios = []
    for linha in linhas[1:]: # saltar a primeira linha (cabeçalho)
        partes = linha.strip().split(',')
        freq = float(partes[0])
        real = float(partes[1])
        imag = float(partes[2])
        freq_resposta.append(freq)
        valores_reais.append(real)
        valores_imaginarios.append(imag)

    return np.array(freq_resposta), np.array(valores_reais), np.array(valores_imaginarios)

```

Figura 34 - Função de leitura de dados

3.3.2. Pós-processamento

O pós-processamento das FRF envolve a identificação das frequências e respectivos modos naturais de vibração a partir da análise das amplitudes das FRF medidas. Existem no mercado diversos programas dedicados à análise modal experimental. Todavia estes parâmetros podem ser obtidos de uma forma aproximada utilizando uma metodologia mais simples, uma vez que as estruturas apresentam, em geral, baixo amortecimento estrutural. A metodologia implementada assenta na identificação das frequências cujas amplitudes apresentam valores de pico e a partir destas são extraídas as componentes real e imaginária da FRF. Estas frequências são assumidas como correspondentes às frequências naturais de vibração da estrutura, sendo o seu valor obtido tomando a médias das frequências correspondentes ao valor de pico. Os respectivos modos são construídos a partir das amplitudes da componente imaginária da FRF à frequência natural.

3.3.3. Obtenção dos modos de vibração

De seguida, e uma vez identificadas as frequências naturais, obtém-se os modos naturais de vibração. Estes são construídos tomando a amplitude da parte imaginária à frequência natural e considerando o ponto de excitação da estrutura. Nesta construção é necessário indicar ao programa o número e a distância entre pontos de medição.

Para minimizar a incerteza associada a este método de identificação dos modos naturais de vibração, na sua representação é usada a função interpolação cúbica.

Tendo como objetivo a análise dos modos de vibração do ensaio em estudo, e depois da aquisição e processamento dos dados vindos dessa mesma aquisição, realizou-se um programa

Desenvolvimento de programa para determinação dos parâmetros modais

com o intuito de ler os resultados obtidos do processamento e posterior representação gráfica dos modos de vibração obtidos.

Para este efeito, o programa precisa do comprimento da estrutura unidimensional assim como o número de pontos em que este se encontra dividido de modo a saber quantos pontos precisará para a representação gráfica. Necessita, também, da frequência desejada, sendo esta obtida através da FRF dos diversos pontos apresentados anteriormente.

Assim, sabendo a posição dos pontos divisórios da estrutura, da frequência desejada, o programa obtém a informação previamente processada e, para os valores de frequência desejados, armazena o valor imaginário da amplitude correspondente para a representação gráfica do modo de vibração escolhido.

Por fim, para uma representação gráfica mais próxima da realidade utilizou-se a função *interpolate* com o objetivo de realizar uma interpolação cúbica dos dados anteriormente armazenados. Esta função pretende estimar os valores não definidos e através da interpolação cúbica, é definida uma curva tendo em conta os valores de modo a obter o modo de vibração mais próximo da realidade. Na Figura 35 pode-se observar a programação da função de interpolação para a representação gráfica.

```
# criação da função de interpolação cúbica
interp_func = interp1d(posicao, modulo_vibacao, kind='cubic')

# criação de um array de posições para a interpolação
posicao_fina = np.linspace(posicao.min(), posicao.max(), num=500)
modulo_vibacao_interpolado = interp_func(posicao_fina)
```

Figura 35 - Função de interpolação

4. Resultados e discussão

Neste capítulo é descrito o procedimento seguido para a medição das FRF numa viga em aço na condição livre-livre. São descritas as etapas necessárias à identificação das suas frequências naturais e modos de vibração.

Segue-se a apresentação das frequências e modos naturais de vibração da viga determinados a partir da solução desenvolvida em fonte aberta. É realizada, também uma solução numérica no software ANSYS® e uma solução analítica com objetivo de comparar os resultados determinados. Por último, efetua-se essa análise comparativa e discussão dos resultados obtidos pelos diferentes métodos.

4.1. Ensaio experimental

O ensaio experimental de uma viga, na condição livre-livre, teve como objetivo identificar as suas frequências e modos naturais de vibração. Na Tabela 1 podem-se examinar as propriedades da viga em estudo.

Tabela 1 – Propriedades da viga

$\rho(\text{kg/m}^3)$	$E(\text{MPa})$	$L(\text{m})$	$b(\text{m})$	$h(\text{m})$
7850	210	0,596	0,04	0,006

A condição livre-livre foi criada através da montagem da viga sobre uma esponja de elevada flexibilidade. Posteriormente, foram marcados 9 pontos, visíveis na Figura 36, igualmente espaçados, ao longo do comprimento da viga para a sua seção média.

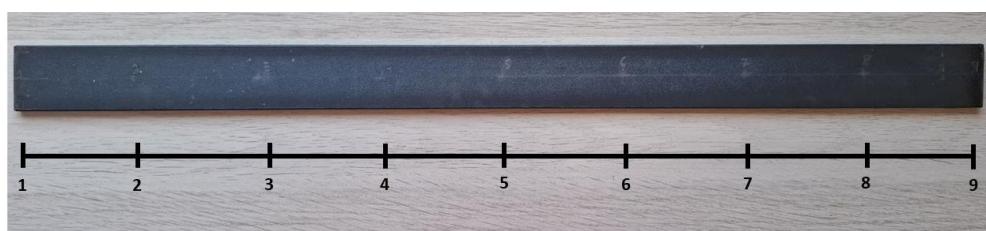


Figura 36 - Viga

O acelerómetro foi montado numa das suas extremidades, no ponto 1, sendo a excitação aplicada nos diferentes pontos em que foi dividida a viga.

Resultados e discussão

As FRF do tipo aceleração entre pares de pontos foram medidas tomando a média aritmética de três medições. Na Figura 37 apresenta-se a montagem experimental, com a viga apoiada sobre a esponja, o acelerômetro montado no ponto 1, o martelo de impacto, a placa de aquisição e o computador necessário para o controle da placa NI e gravação dos sinais.

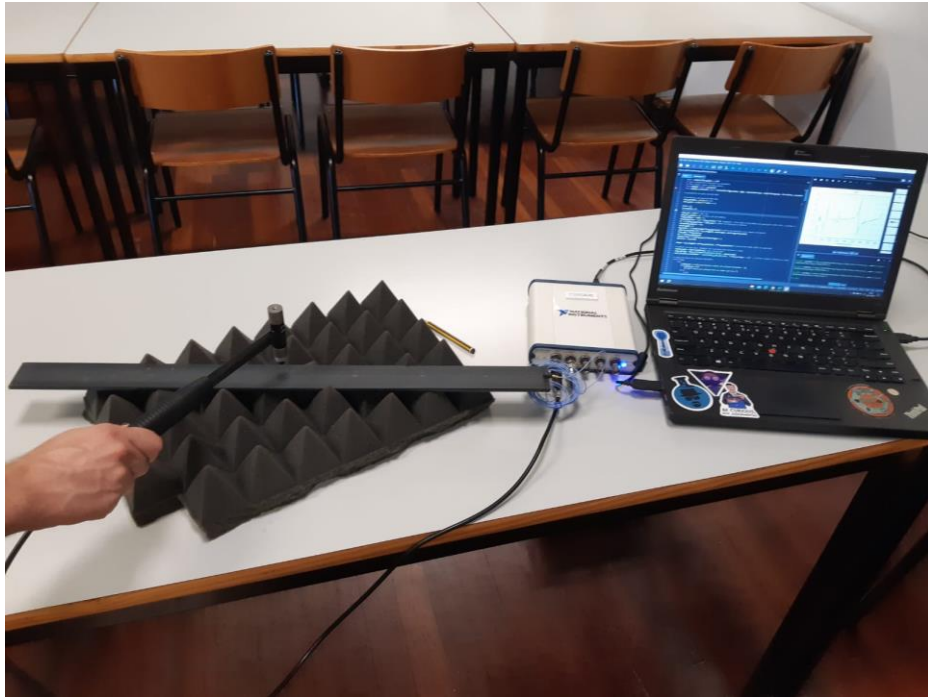


Figura 37 - Ensaio Experimental Viga

No ensaio foi usado um martelo de impacto PCB Piezotronics modelo 086C01 com a sensibilidade de 11,2 mV/N [38]. Na Figura 38 encontra-se o martelo de impacto utilizado.



Figura 38 - Martelo de impacto PCB Piezotronics modelo 086C01

Foi usado, também um acelerômetro da PCB Piezotronics modelo 352A24 com a sensibilidade de 9,92 mV/ms² [39]. Este pode-se observar na Figura 39.



Figura 39 - Acelerómetro da PCB Piezotronics modelo 352A24

Por fim, foi utilizada a placa de aquisição NI USB-4432. Através da Figura 40 pode-se observar a placa utilizada, assim como os seus canais de entrada.



Figura 40 - NI USB 4431

De seguida, especificam-se as condições de aquisição e de processamento. O *time out* para o ensaio foi estabelecido como 10 s, para permitir realizar de forma conveniente o impacto com o martelo e conseqüente truncamento do sinal. A frequência de análise foi estabelecida como 800 Hz, de modo a garantir a análise de frequências naturais até 800 Hz, sendo que, segundo o critério de Nyquist, a frequência mínima de amostragem deverá ser o dobro da frequência de análise, ou seja 1600 Hz, para evitar a ocorrência do fenómeno de *aliasing*, isto é a distorção do sinal devido a uma amostragem inadequada. Assim, para melhorar a qualidade do sinal medido, nomeadamente a nível de amplitude, foi decidido utilizar uma frequência de amostragem de 24 kHz, uma vez que esta tem uma relação de 30 vezes a frequência do sinal de análise. Na Figura 41 e Figura 42, encontra-se, a título de exemplo, uma das leituras efetuados ao longo do ensaio experimental antes da truncagem, ou seja, obtida em continua ao longo do *time out* definido do sinal da força e da aceleração, respetivamente.

Resultados e discussão

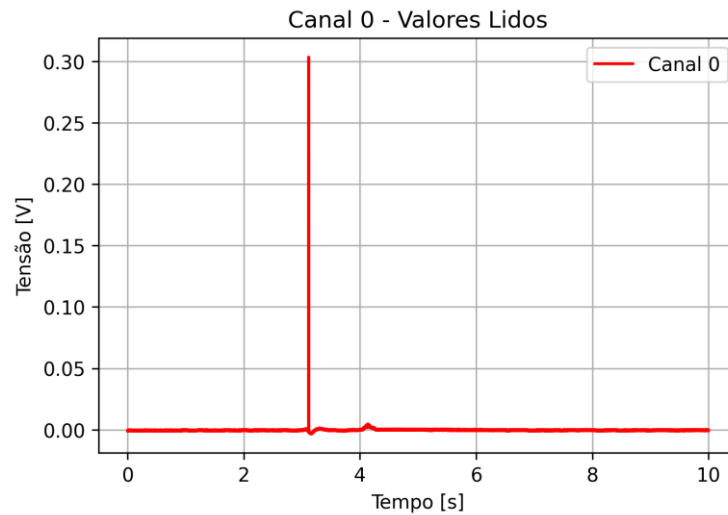


Figura 41 - Leitura de dados do canal 0 no ponto 2

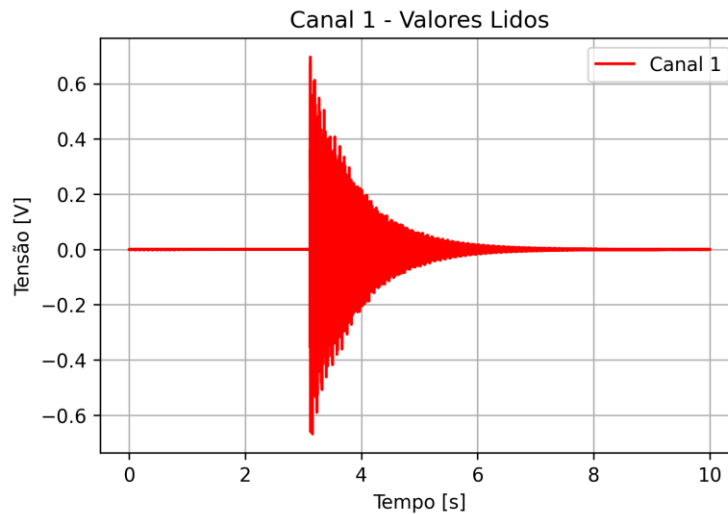


Figura 42 - Leitura de dados do canal 1 no ponto 2

No que diz respeito ao pós-impacto, o número de pré amostras, isto é, a quantidade de amostras a armazenar antes do índice do impacto, foi fixado em 6 e o valor em tensão definido para identificação do pico no canal 0, isto é o impacto realizado pelo martelo, estabeleceu-se como 0,1 V. O tempo de amostragem fixou-se em 2 s, dando origem a um número de amostras de 48000. Assim, apresenta-se, no seguimento do exemplo mostrado anteriormente, o sinal tanto da força como da aceleração, ponderado e truncado.

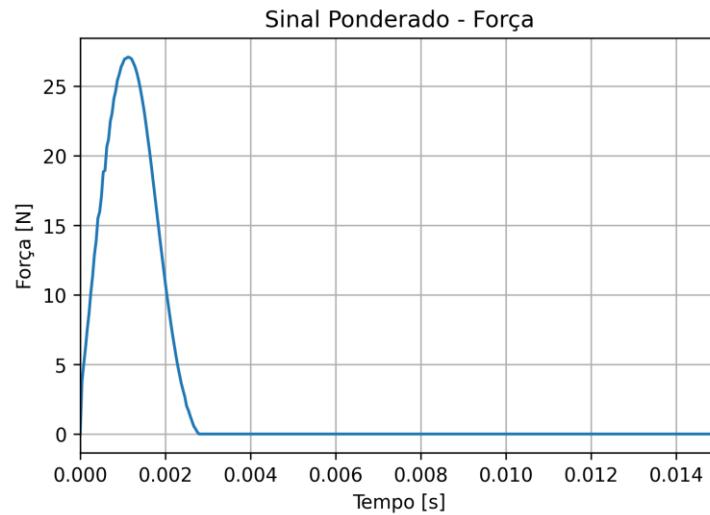


Figura 43 - Sinal da força ponderado no ponto 2

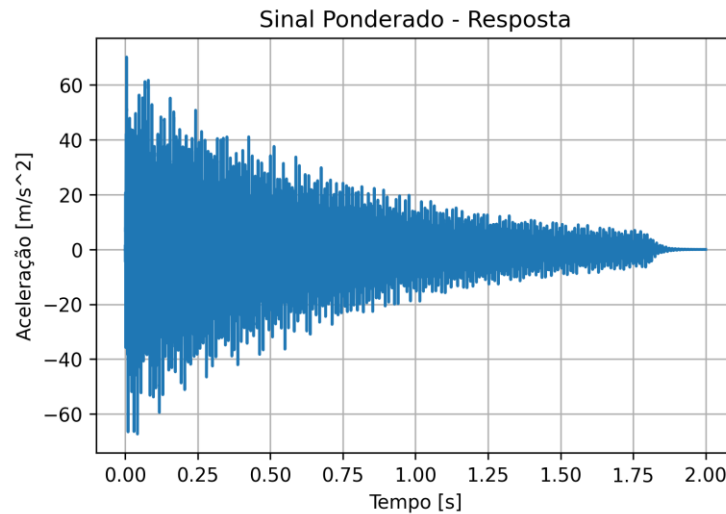


Figura 44 - Sinal da força ponderado no ponto 2

Definidas as condições de ensaio e consequentes parâmetros presentes no mesmo iniciou-se todo o processo com objetivo de obter os modos de vibração da viga apresentada anteriormente.

Foram realizados, tal como referido, 3 leituras por cada ponto definido, ou seja, um total de 27 leituras executadas. Dando origem a 9 funções de resposta em frequência médias. As leituras foram efetuadas em cada um dos pontos marcados na viga, sendo que posteriormente ao impacto, verificava-se no programa a existência ou não de repiques e a leitura era validada em caso de estes não existirem. Assim, na Figura 45, Figura 46, Figura 47, Figura 48, Figura 49, Figura 50, Figura 51, Figura 52 e Figura 53, pode-se observar as FRF médias obtidas através do ensaio experimental, do ponto 1 ao 9, respetivamente.

Resultados e discussão

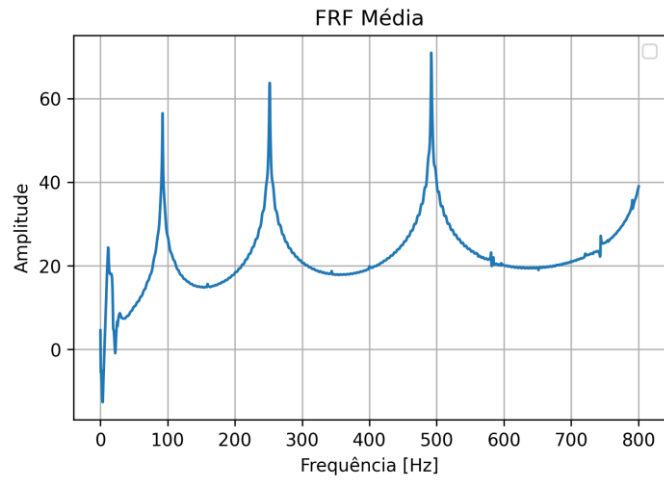


Figura 45 – FRF medida no ponto 1

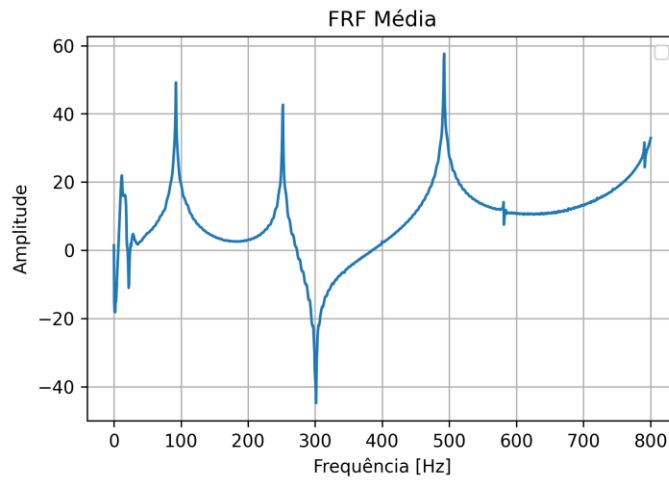


Figura 46 - FRF medida no ponto 2

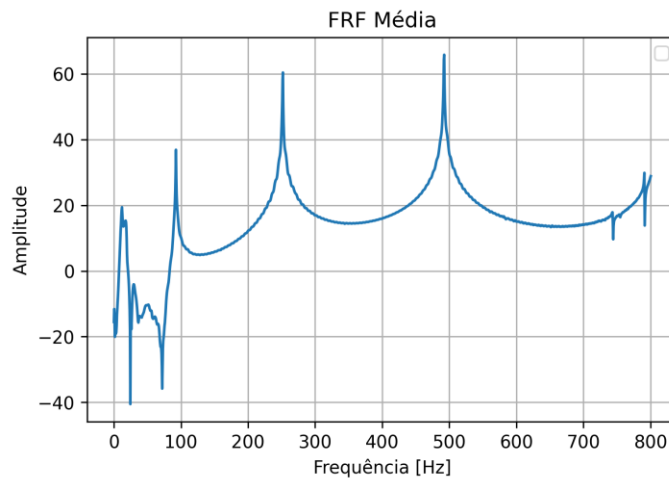


Figura 47 - FRF medida no ponto 3

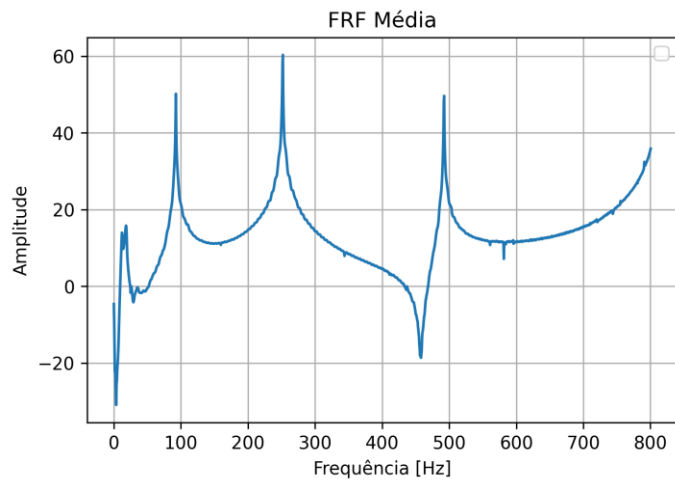


Figura 48 - FRF medida no ponto 4

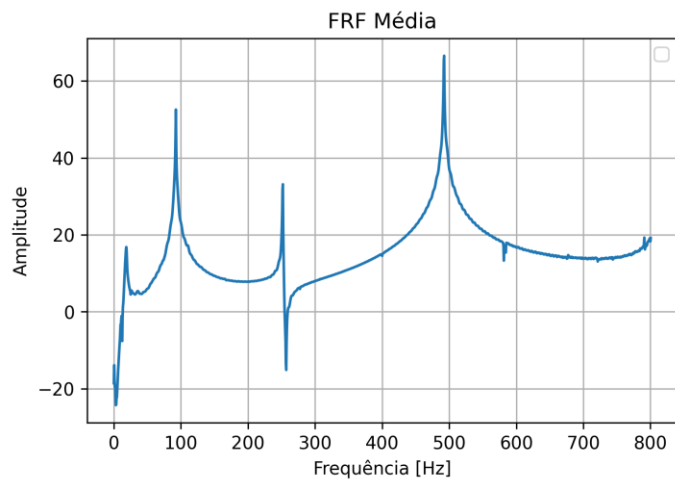


Figura 49 - FRF medida no ponto 5

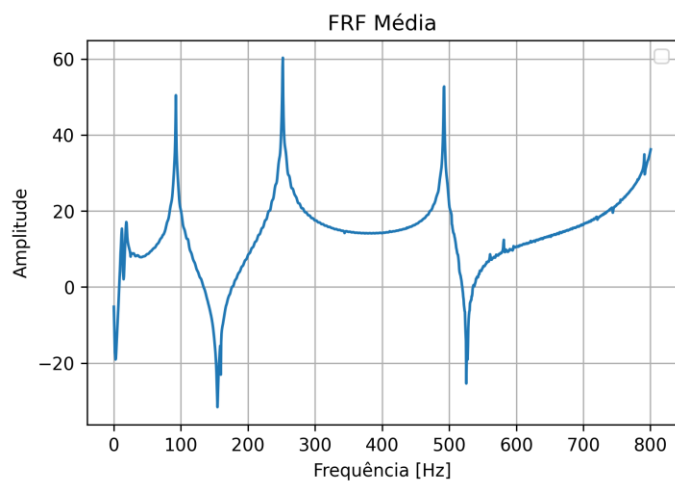


Figura 50 - FRF medida no ponto 6

Resultados e discussão

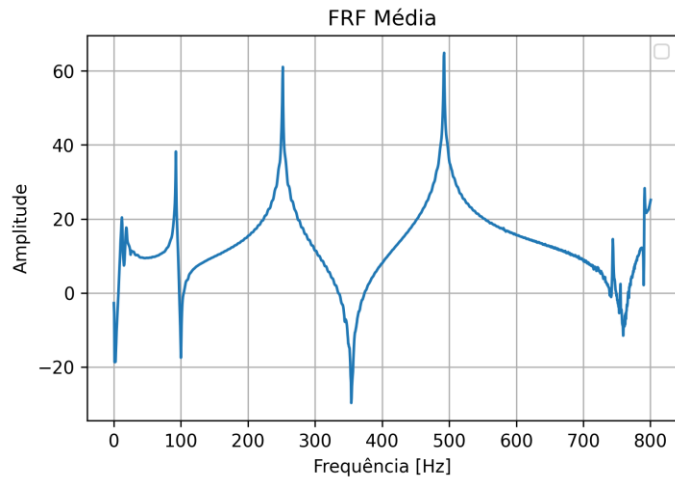


Figura 51 - FRF medida no ponto 7

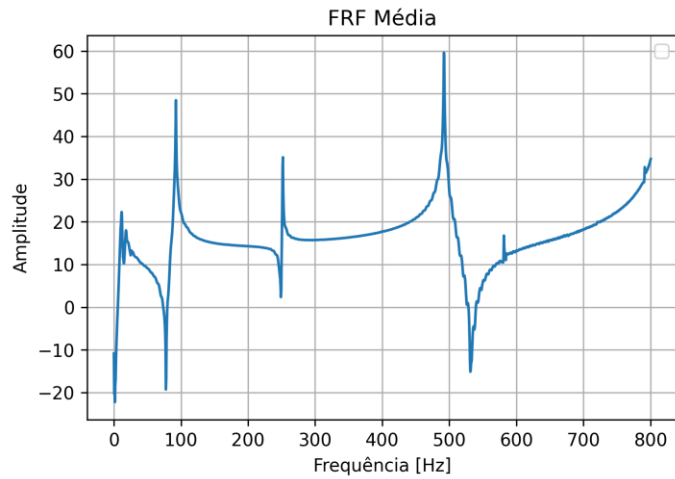


Figura 52 – FRF medida no ponto 8

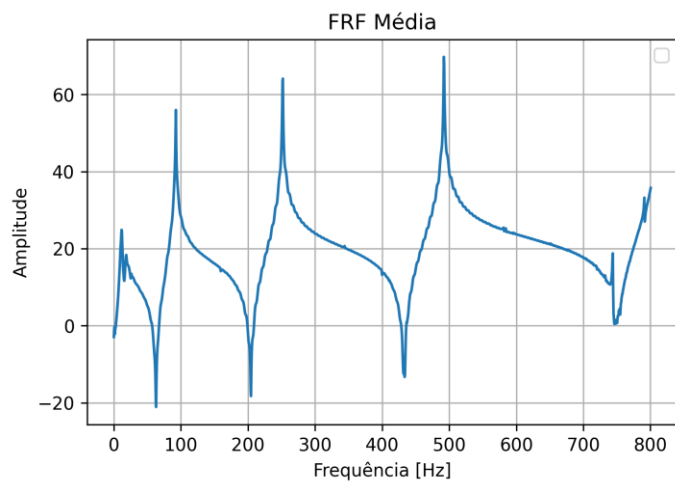


Figura 53 – FRF medida no ponto 9

Observando as FRF apresentadas anteriormente, é de notar a existência de quatro picos presentes nestas sendo primeiro muito próximo da frequência 0, correspondendo à frequência de vibração de corpo rígido da estrutura. Os restantes três picos dizem respeito às frequências naturais da viga em estudo.

Assim, pode-se observar na Tabela 2 as frequências obtidas através do ensaio experimental executado.

Tabela 2 - Frequências naturais obtidas experimentalmente

Ordem das frequências	Valor de frequência natural (Hz)
1ª	92,0
2ª	250,5
3ª	490,5

Posto isto, e identificadas as frequências naturais, obtiveram-se os modos de vibração da viga em estudo. Na Figura 54, Figura 55, Figura 56 encontram-se o primeiro, segundo e terceiro modo de vibração da viga, respetivamente

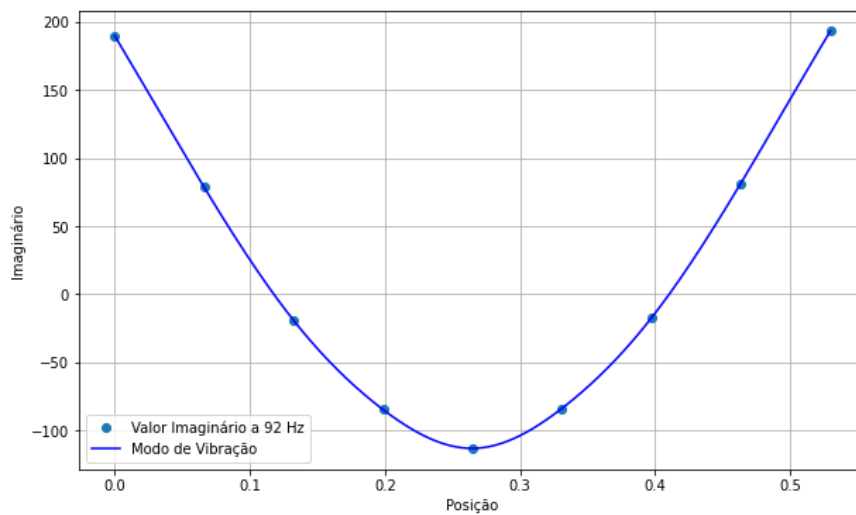


Figura 54 - Primeiro modo de natural de vibração - experimental

Resultados e discussão

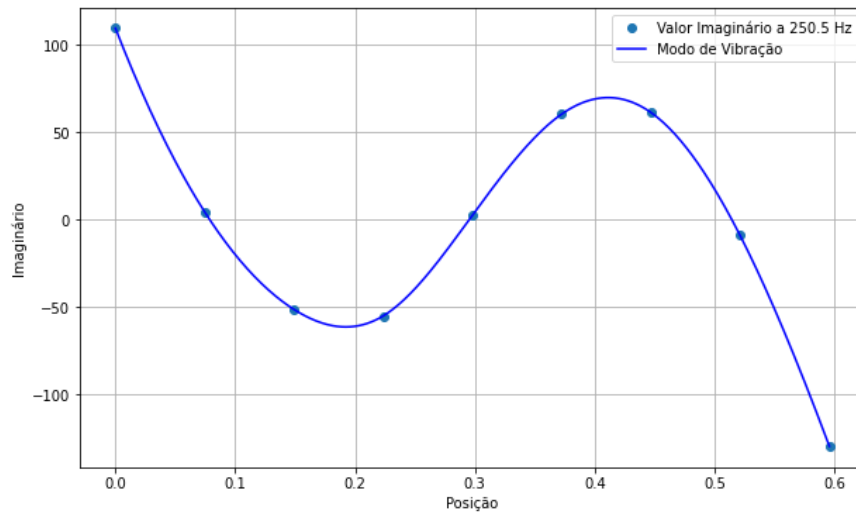


Figura 55 - Segundo modo de natural de vibração - experimental

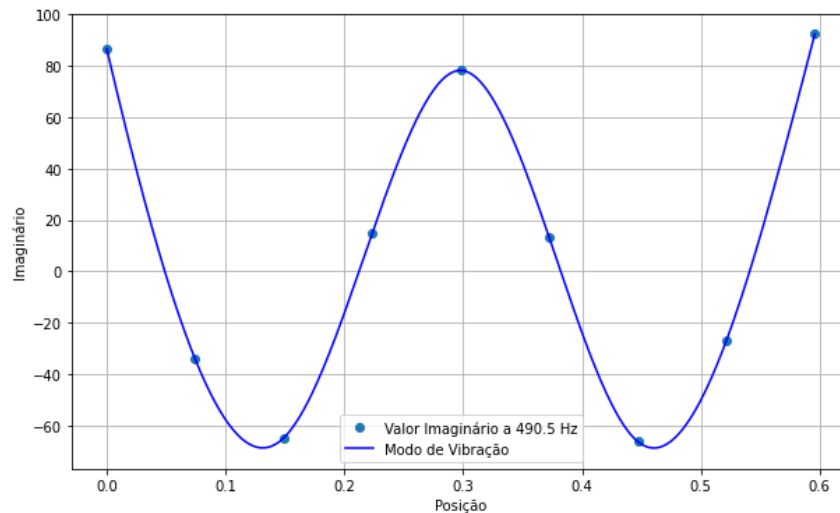


Figura 56 - Terceiro modo de natural de vibração - experimental

A partir da obtenção dos modos naturais de vibração é notório, principalmente no terceiro modo, que estes não apresentam uma simetria perfeita mostrando uma imperfeição nos resultados. Esta imperfeição pode ser justificada com erros de medição, pelo facto da geometria da viga não ser uniforme, por interferência da massa do acelerómetro e também pelo facto do método utilizado para a identificação dos modos não ser o mais adequado. A utilização de programas comerciais de análise permitiria eliminar alguns destes efeitos e obter resultados mais precisos. No entanto, os resultados apresentados, apesar de ligeiras imprecisões, garantem boa qualidade e cumprem plenamente o propósito do trabalho.

4.2. Solução numérica

O estudo da resposta vibratória através da simulação numérica baseado no método de elementos finitos e recorrendo ao programa comercial ANSYS®, teve como objetivo realizar um estudo comparativo dos resultados obtidos.

Deste modo, criou-se um modelo computacional da viga em estudo. Esse modelo pode visualizar-se na Figura 57.

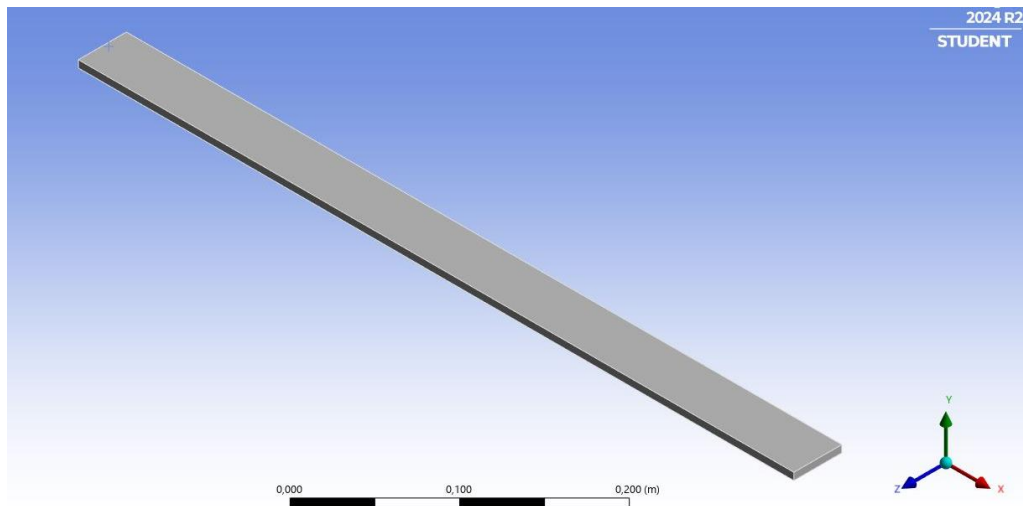


Figura 57 - Modelo geométrico da viga

O material atribuído ao modelo foi o aço estrutural. Assim, foram atribuídas as propriedades referidas anteriormente (Tabela 1) de forma a manter as propriedades semelhantes às da viga usadas no ensaio experimental. A construção do modelo numérico envolveu a sua discretização em elementos sólidos lineares com quatro nós e três graus de liberdade por nó, com o tamanho de 5 mm. A malha uniforme gerada automaticamente pelo programa, composta por esses elementos, é apresentada na Figura 58.

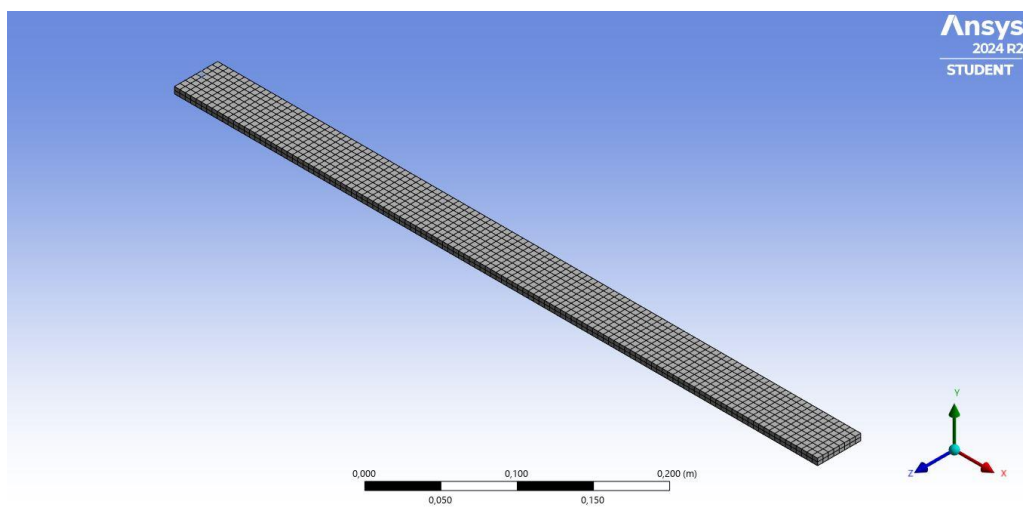


Figura 58 – Modelo numérico da viga

Uma vez que se pretende obter resultados nas mesmas condições do ensaio experimental executado anteriormente, não se impôs nenhuma condição fronteira ao modelo da viga, ficando, portanto, na condição livre-livre. Assim, resultado desta condição, as primeiras frequências obtidas na simulação correspondem à vibração de corpo rígido da estrutura. Na Figura 59 encontra-se um exemplo desse efeito, onde é notório que para uma frequência de 0 Hz, o corpo move-se na sua totalidade.

Resultados e discussão

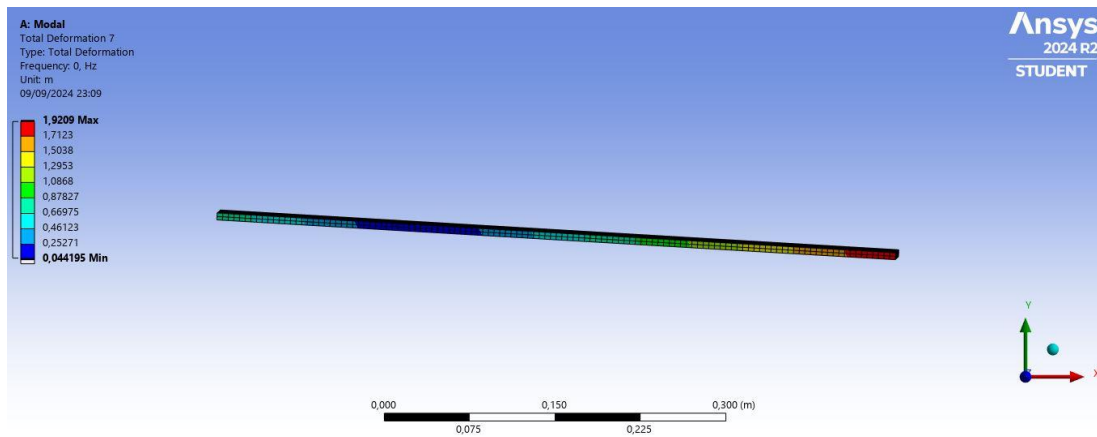


Figura 59 - Vibração Corpo Rígido

Depois de observados todos os modos de vibração de corpo rígido, analisaram-se os três seguintes modos de vibração, de modo a ser possível a comparação com os modos de vibração obtidos experimentalmente.

Na Tabela 3 são apresentadas as primeiras três frequências naturais de vibração obtidas numericamente, nas quais foram omitidas as frequências correspondentes aos modos de corpo rígido.

Tabela 3 - Frequências naturais obtidas numericamente

Ordem das frequências	Valor de frequência natural (Hz)
1ª	89,79
2ª	247,50
3ª	485,16

Na Figura 60, Figura 61 e Figura 62 são apresentados os três primeiros modos naturais vibração da viga obtidos através da simulação numérica.

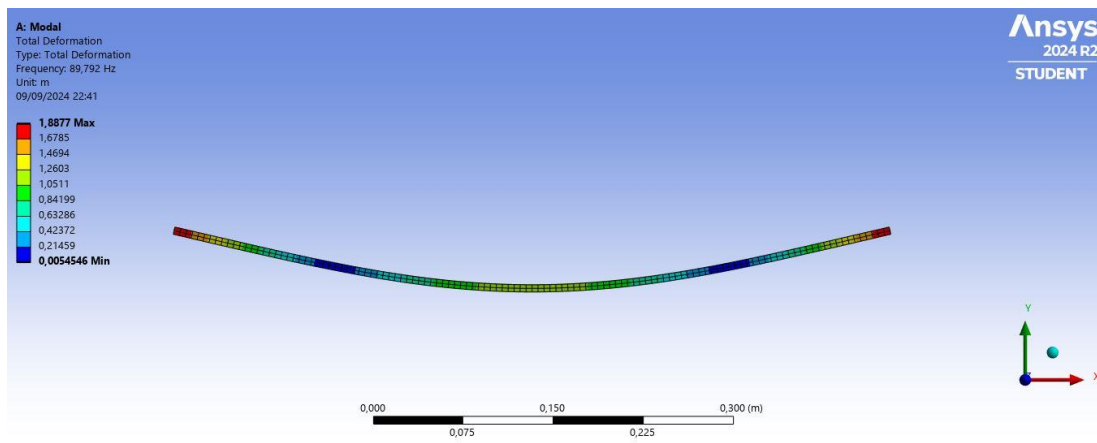


Figura 60 - Primeiro modo natural de vibração- numérico

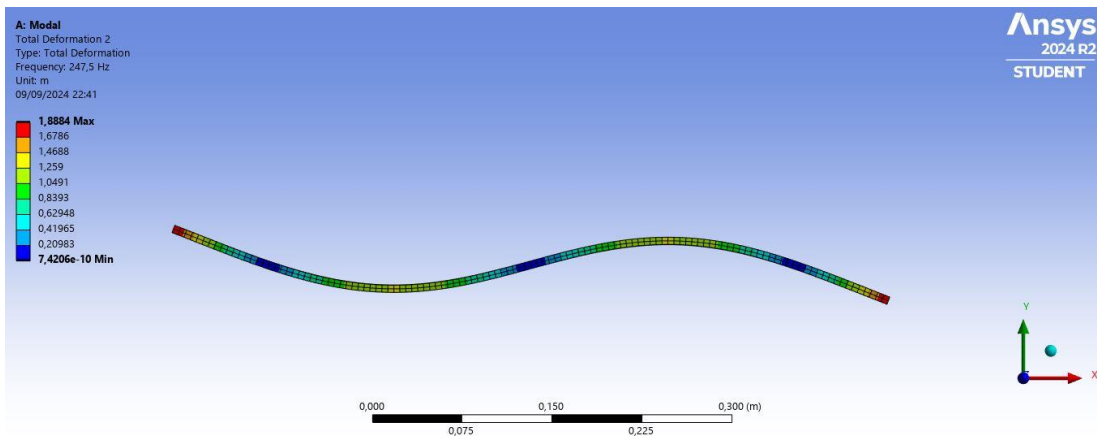


Figura 61 - Segundo modo natural de vibração- numérico

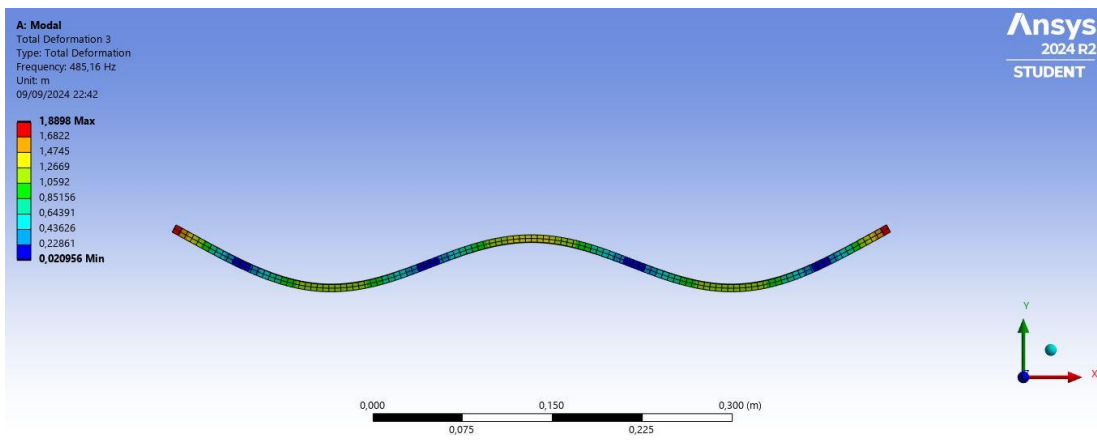


Figura 62 - Terceiro modo natural de vibração- numérico

A título de curiosidade, apresentam-se na Figura 63 o quarto modo de vibração, obtido nas mesmas condições que os modos anteriores, manifestando-se à frequência de 589,28 Hz. Este corresponde à combinação da flexão em torno dos eixos Y e Z.

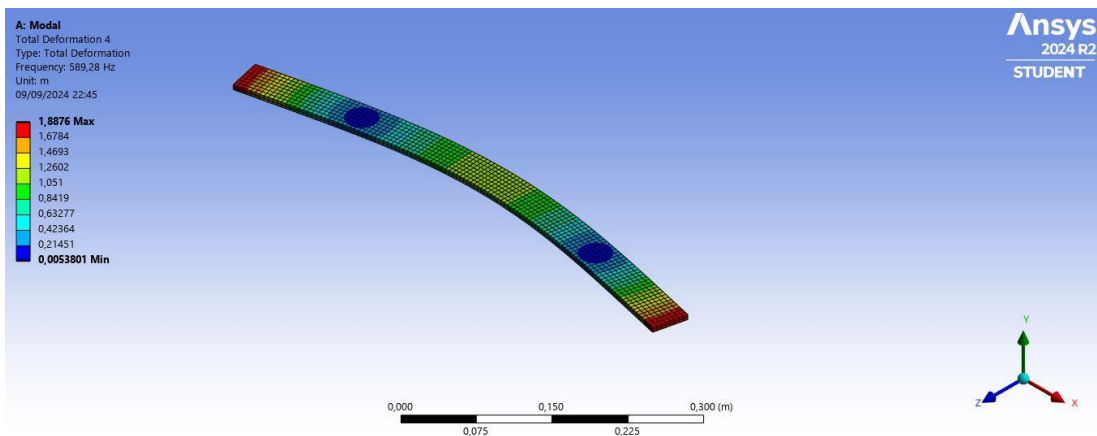


Figura 63 - Quarto Modo de Vibração – Análise Numérica

Já quanto ao quinto modo de vibração apresentado na Figura 64 e com a frequência de 760,73 Hz, corresponde ao modo de torção da viga.

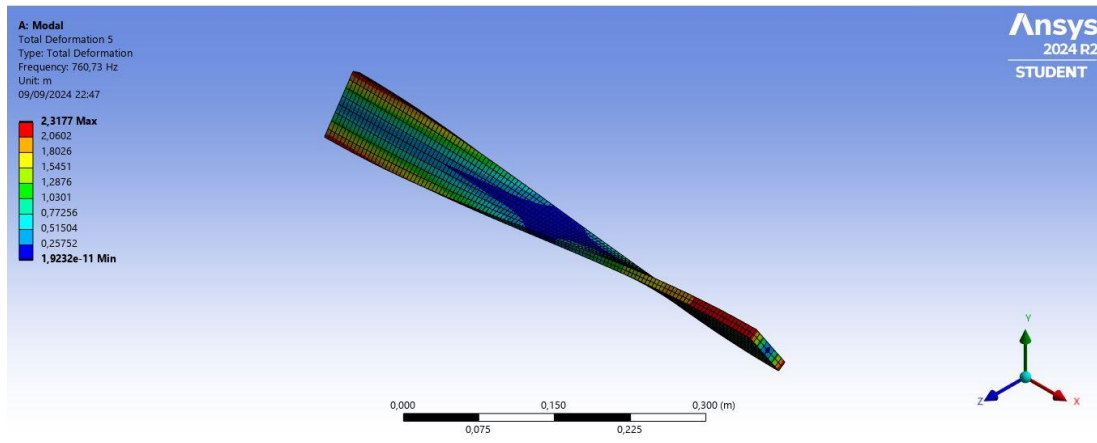


Figura 64 - Quinto Modo de Vibração - Análise Numérica

4.3. Solução analítica

Depois de obtidas as frequências naturais e os modos naturais de vibração, tanto através ensaio experimental e pelo método de elementos finitos (MEF), estes foram também obtidos a partir da formulação analítica de vibração de vigas finas. Deste modo, calculou-se as primeiras três frequências naturais de vibração para as propriedades apresentadas pela viga em estudo. Assim, pode-se determinar as frequências naturais de vibração de uma viga na condição livre-livre, através da equação (7) [40].

$$\omega_n = k_n^2 \sqrt{\frac{EI}{\rho AL^4}} \quad (rad/s) \quad (7)$$

Além da notação normalmente utilizada para a viga, é também introduzido o chamado número de onda, k_n . Este número assume valores específicos para diferentes condições de vibração da viga, sendo neste caso livre-livre [40]. Na Tabela 4 encontram-se os valores para a vibração em regime livre.

Tabela 4 - Números de onda

Frequência de vibração	k_n
0	0
1	4,7300
2	7,8532
3	10,9956

Estes valores são fundamentais para a análise das frequências naturais de vibração de vigas com diferentes condições de fronteira, permitindo prever o comportamento dinâmico da estrutura

com precisão. A introdução do número correlacionado com a frequência facilita a identificação de frequências modais em diferentes tipos de vigas, sendo assim uma ferramenta útil na análise de vigas.

Assim, as correspondentes frequências naturais de vibração são determinadas introduzindo o número de onda na equação (7) e conhecidas as propriedades da viga, onde o momento de inércia de área da viga I é dado pela equação (8).

$$I = \frac{b \times h^3}{12} \quad (8)$$

Deste modo, substituindo os valores de k_n , obtém-se as frequências naturais, as quais são apresentadas na Tabela 5.

Tabela 5 - Frequências naturais obtidas analiticamente

Ordem das frequências	Valor de frequência natural (Hz)
1ª	89,80
2ª	247,55
3ª	485,29

Os correspondentes modos naturais de vibração são determinados através da equação (9) [40].

$$W_n(x) = [\sinh(k_n x) + \sin(k_n x)] + \frac{\sin(k_n L) - \sinh(k_n L)}{\cosh(k_n L) - \cos(k_n L)} [\cosh(k_n x) + \cos(k_n x)] \quad (9)$$

Assim, foram obtidos os modos de vibração da viga com as condições fronteira definidas. Deste modo, pode-se observar na Figura 65 o primeiro modo de vibração obtido analiticamente. Posteriormente na Figura 66 pode-se observar o segundo e, por fim, na Figura 67 o terceiro modo.

Resultados e discussão

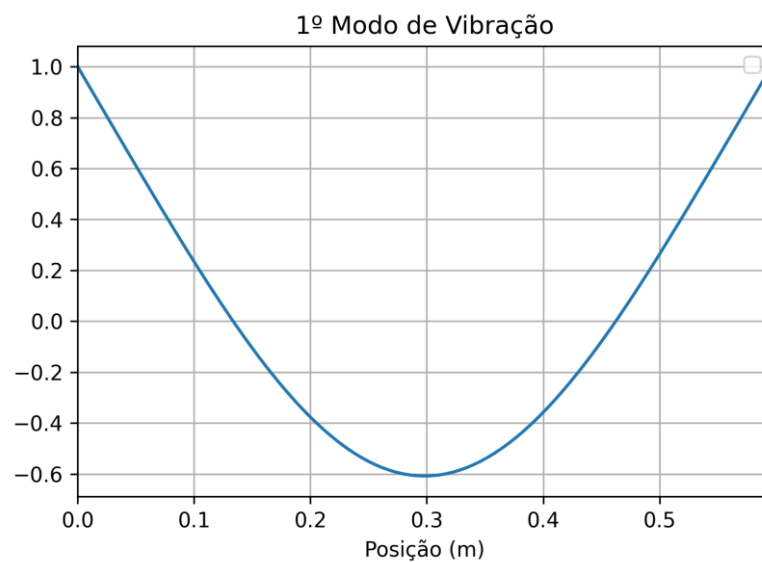


Figura 65 - Primeiro modo de natural de vibração – analítico

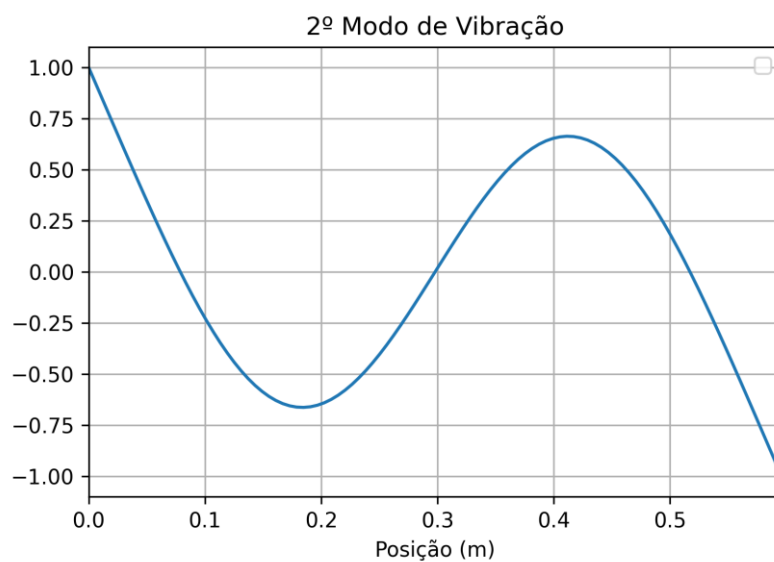


Figura 66 - Segundo modo de natural de vibração – analítico

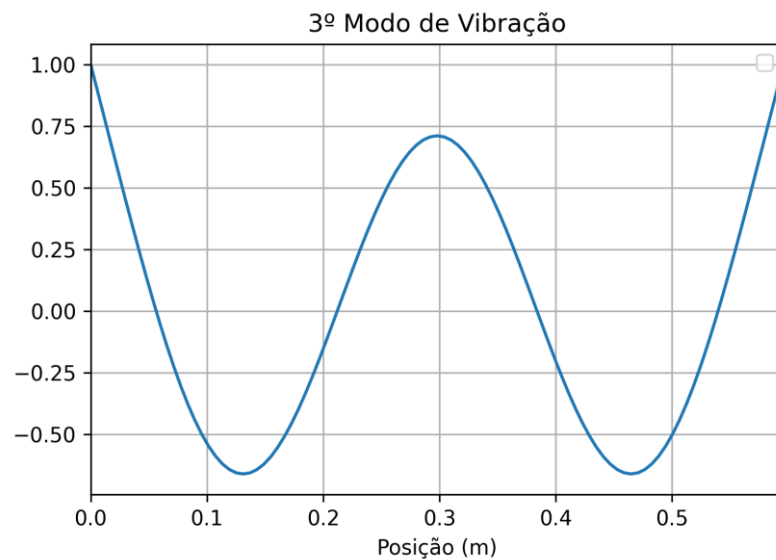


Figura 67 - Terceiro modo de natural de vibração – analítico

De notar que os valores máximos de deslocamento dos três primeiros modos de vibração ocorrem nas extremidades da viga, dado que a viga se encontra em vibração livre. Esta forma distingue-se, por exemplo, da viga simplesmente apoiada em que a amplitude de vibração das extremidades é nula.

4.4. Comparação e discussão de resultados

Na Tabela 6 são comparadas as três primeiras frequências naturais obtidas através do ensaio experimental, por via numérica e analítica.

Tabela 6 - Resumo de frequências naturais da viga

Ordem das frequências	Experimental (Hz)	Numérico (Hz)	Analítico (Hz)
1ª	92,0	89,79	89,80
2ª	250,5	247,50	247,55
3ª	490,5	485,16	485,29

Assim, pode-se observar uma diferença pouco significativa entre os valores numéricos e analíticos sendo estes bastantes próximos. Quanto aos valores obtidos experimentalmente, são ligeiramente diferentes sendo que a diferença não é significativa.

Na Tabela 7 é apresentado o erro relativo das frequências naturais numéricas e analíticas relativamente às frequências experimentais. Com isto, pretendeu-se comparar os valores obtidos numérico e analítico com os valores obtidos experimentalmente.

Tabela 7 - Erro relativo entre frequências naturais

Ordem das frequências	Numérico (%)	Analítico (%)
1ª	2,40	2,39
2ª	1,20	1,18
3ª	1,09	1,06

Uma vez que as frequências naturais numéricas e analíticas são muito próximas, o seu desvio relativamente às frequências experimentais é praticamente idêntico. Contudo, o desvio observado é pouco significativo, sendo consistentemente negativo, uma vez que as frequências experimentais são sempre superiores às frequências obtidas pelos outros dois métodos

Relativamente aos modos de vibração, procurou-se compará-los diretamente. Para este fim, foi necessário normalizá-los. Assim, procurou-se ajustar as suas amplitudes para o intervalo de -1 a 1, através da divisão pelo valor máximo associado, de modo a garantir uma comparação consistente. No que diz respeito aos valores analíticos, e uma vez que estes se calcularam para uma amplitude normalizada, não foi necessário nenhum ajuste. No entanto, tanto para os valores numéricos como experimentais foi necessário a normalização para o intervalo pretendido.

Deste modo, na Figura 68 é apresentado o primeiro modo natural normalizado, obtido pelos três métodos.

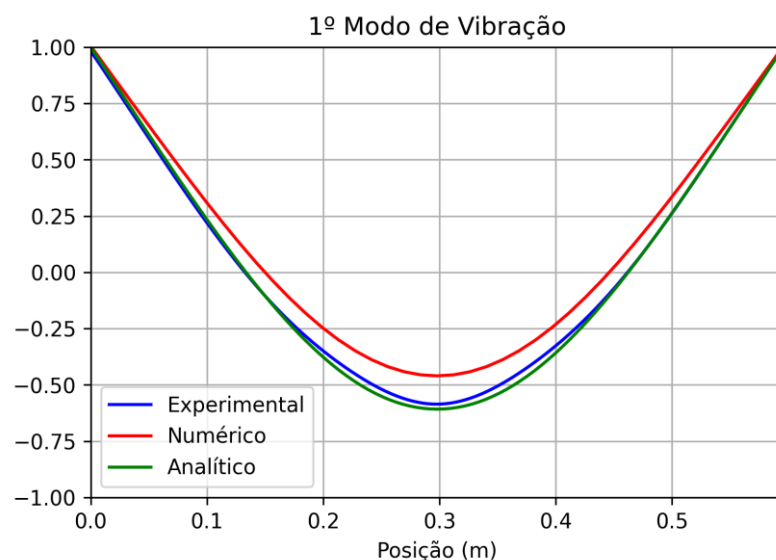


Figura 68 – Primeiro modo natural de vibração obtido pelos três métodos

Relativamente aos primeiros modos de vibração, é de notar que o modo obtido experimentalmente é bastante semelhante ao modo analítico. Sendo que a diferença mais notória é entre o modo numérico e os restantes, havendo um maior decaimento da amplitude quando esta converge para o centro da viga. No entanto, existe uma relação aceitável entre os três métodos.

De seguida, pode-se observar, na Figura 69 a configuração do segundo modo natural de vibração obtido pelos três métodos.

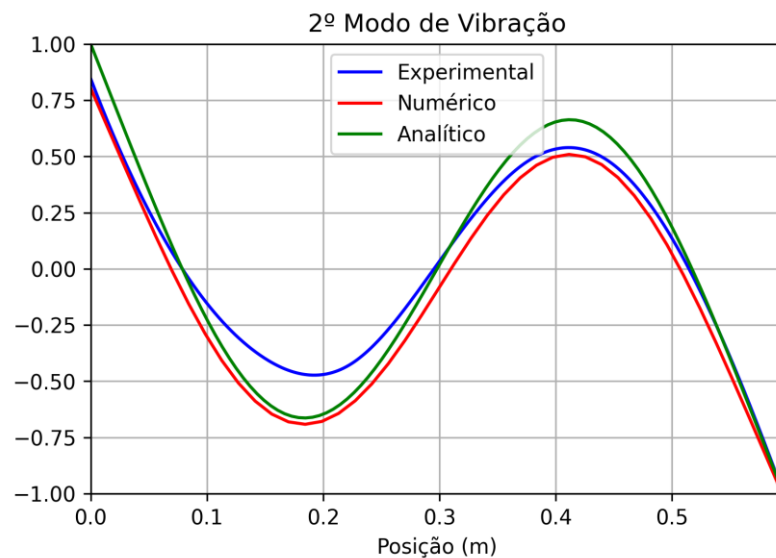


Figura 69 - Segundo modo natural de vibração obtido pelos três métodos

Neste caso, são observadas algumas discrepâncias na configuração do modo para os três métodos. Refira-se que o modo obtido numericamente apresenta uma amplitude menor nas extremidades e maior no centro da viga quando comparado com o método analítico. Porém, apesar destas ligeiras diferenças, a configuração dos modos é semelhante, isto é, apresenta três pontos nodais muito próximos.

Por último, na Figura 70 é representado o terceiro modo natural obtido pelos três métodos.

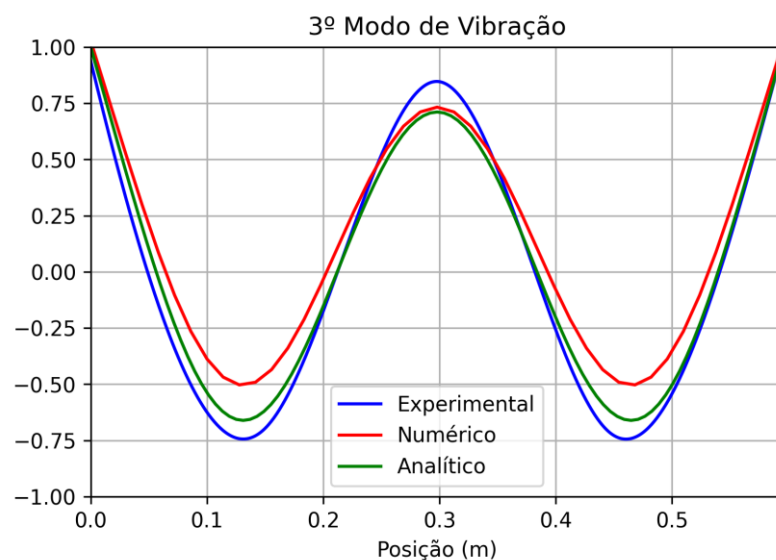


Figura 70 - Terceiro modo natural de vibração obtido pelos três métodos

No que diz respeito ao terceiro modo de vibração obtido pelos três métodos, verifica-se que estas apresentam uma configuração muito semelhante, apesar da translação e da diferença de

Resultados e discussão

amplitude. Refira-se que a solução numérica é a que apresenta menor amplitude, seguida pela solução analítica, sendo a medição experimental a que apresenta maior amplitude.

Em resumo, os resultados obtidos mostram que as frequências e os modos naturais de vibração apresentam, respetivamente, valores e configurações muito próximas. Além disso, estes resultados comprovam que os programas usados para a medição das FRF e identificação das frequências e modos naturais de vibração foram corretamente implementados.

5. Conclusão e proposta de trabalhos futuros

Neste capítulo apresentam-se as conclusões relativas ao trabalho realizado, tendo em conta o objetivo e mais valias do trabalho desenvolvido. Além disso, apresentam-se as dificuldades enfrentadas ao longo do desenvolvimento do trabalho e estratégias utilizadas para as ultrapassar.

Por fim, apresentam-se as propostas de trabalhos futuros com vista à constante evolução e melhoria do trabalho realizado.

5.1. Conclusões

Tendo em conta que o objetivo principal do trabalho era o desenvolvimento um programa funcional para a obtenção de parâmetros modais para uma estrutura unidimensional, este encontra-se alcançado. O foco, durante o desenvolvimento do programa, esteve centrado na criação e otimização tendo em conta as necessidades específicas de uma solução em código aberto. Como foi possível verificar anteriormente, foi possível analisar e obter os modos de vibração de uma viga unidimensional através do programa desenvolvido.

Deste modo, realizou-se o desenvolvimento de um programa, como referido, em código aberto, que garante uma maior flexibilidade e é ajustável ao caso de estudo, como por exemplo, definição dos graus de liberdade, adaptação ao equipamento de aquisição e de introdução de excitação e outras configurações necessárias. A solução desenvolvida integra todas as funcionalidades dentro do mesmo programa, evitando limitações associadas a licenças de software ou restrições temporais, enquanto mantém a simplicidade na configuração e utilização.

É importante referir que o desenvolvimento do programa foi muito desafiante devido às limitações impostas pela integração rígida entre o software *Labview* e o hardware, ambos desenvolvidos pela empresa *National Instruments*®, funcionando praticamente como um sistema fechado. Apesar da existência da biblioteca *nidaqmx*, que ajuda o programa a aproximar-se de outros já desenvolvidos, esta carece de falta de informação e de uma especificidade, que leva ao aumento da complexidade do programa, e consequentemente do trabalho desenvolvido. Esta falta de informação não permitiu a definição de um *trigger* de

excitação que, apesar de presente nos parâmetros a definir pela biblioteca disponível, não foi possível introduzir de forma funcional.

Não obstante as dificuldades encontradas pode-se afirmar que foi possível concluir com sucesso o trabalho proposto.

5.2. Proposta de trabalhos futuros

Um dos principais pontos fortes do programa desenvolvido é o facto de este ser de fonte aberta. Sendo assim, é possível a sua evolução e atualização, fazendo-o capaz de se adaptar a tarefas diferentes e a outros equipamentos, caracterizando-se assim como um programa versátil e flexível.

Assim, uma melhoria do trabalho presente seria a representação de modos de vibração tanto no plano como no espaço. Deste modo, o programa seria capaz de aumentar a sua aplicação em diversos cenários. Para além desta melhoria, uma melhor identificação dos parâmetros modais, utilizando metodologias idênticas à dos programas comerciais, através da construção de matrizes de massa, rigidez e amortecimento, constituiriam uma melhoria ao programa apresentado.

Outra melhoria com vista à valorização do programa desenvolvido seria a implementação de um *trigger* para o início da leitura dos dados, o que permitiria retirar a pré-preparação do sinal realizada depois da aquisição do sinal e uma leitura melhorada do mesmo. Realizada esta melhoria, seria interessante a realização de um ensaio através de excitadores alternativos, como por exemplo um excitador eletromagnético, de modo aumentar a precisão do ensaio com o propósito de melhorar os resultados obtidos.

Referências

- [1] R. Kendre, «Random Vibration Analysis for Starter Motor of Three Wheeler Automobile», em *Procedia Engineering*, Elsevier Ltd, 2016, pp. 1381–1388. doi: 10.1016/j.proeng.2016.05.168.
- [2] C. Smith, C. M. Akujuobi, P. Hamory, e K. Kloesel, «An approach to vibration analysis using wavelets in an application of aircraft health monitoring», *Mech Syst Signal Process*, vol. 21, n. 3, pp. 1255–1272, 2007, doi: 10.1016/j.ymssp.2006.06.008.
- [3] F. bin Zahid, Z. C. Ong, e S. Y. Khoo, «A review of operational modal analysis techniques for in-service modal identification», 1 de Agosto de 2020, *Springer*. doi: 10.1007/s40430-020-02470-8.
- [4] J. Justo, «Dinâmica de Estruturas», 2022.
- [5] S. Braun, S. Diego, S. Francisco, N. York, B. London, e S. Tokyo, *ENCYCLOPEDIA OF VIBRATION*. San Diego: Academic Press, 2002. doi: 10.1006/rwvb.200.
- [6] O. Døssing e B. & Kjaer, *STRUCTURAL TESTING Part I: Mechanical Mobility Measurements*.
- [7] O. Døssing e B. Kjaer, *STRUCTURAL TESTING Part II: Modal Analysis and Simulation*.
- [8] M. L. Adams, *Rotating machinery vibration: from analysis to troubleshooting*. Ohio: Marcel Dekker, 2001.
- [9] B. J. Schwarz e M. H. Richardson, «EXPERIMENTAL MODAL ANALYSIS», 1999.
- [10] P. Andrade, J. Santos, e P. Escórcio, «Direct Integration Methods versus Modal Superposition Method, on Predicting Staircases Vibrations», *Procedia Structural Integrity*, vol. 28, pp. 279–286, 2020, doi: 10.1016/j.prostr.2020.10.033.
- [11] J. Alvarellos e H. Metiu, «The evolution of the wave function in a curve crossing problem computed by a fast Fourier transform method», *J Chem Phys*, vol. 88, n. 8, pp. 4957–4966, 1988, doi: 10.1063/1.454707.
- [12] J. W. Cooley e J. W. Tukey, «An Algorithm for the Machine Calculation of Complex Fourier Series», *Math Comput*, vol. 19, n. 90, pp. 297–301, 1965, doi: 10.2307/2003354.
- [13] K. Shin e J. K. Hammond, *Fundamentals of Signal Processing: for Sound and Vibration Engineers*. England: Wiley, 2008.
- [14] J. V. Araújo dos Santos, C. M. Mota Soares, C. A. Mota Soares, e N. M. M. Maia, «Structural damage identification in laminated structures using FRF data», *Compos Struct*, vol. 67, n. 2 SPEC. ISS., pp. 239–249, 2005, doi: 10.1016/j.compstruct.2004.09.011.
- [15] M. Hagara, R. Huňady, P. Lengvarský, e P. Pavelka, «Analysis of Reliability of Modal Parameters Estimation Using High-speed Digital Image Correlation Method», *American Journal of Mechanical Engineering*, vol. 3, n. 6, pp. 190–194, 2015, doi: 10.12691/ajme-3-6-7.
- [16] Y. Wang, F. S. Egner, T. Willems, M. Kirchner, e W. Desmet, «Camera-based experimental modal analysis with impact excitation: Reaching high frequencies thanks

Referências

- to one accelerometer and random sampling in time», *Mech Syst Signal Process*, vol. 170, p. 108879, 2022, doi: 10.1016/j.ymsp.2022.108879.
- [17] M. Kodrič, T. Bregar, G. Čepon, e M. Boltežar, «An expansion based on System Equivalent Model Mixing: From a limited number of points to a full-field dynamic response», *Measurement (Lond)*, vol. 190, Fev. 2022, doi: 10.1016/j.measurement.2021.110522.
- [18] J. Pan, Z. Zhang, J. Wu, K. R. Ramakrishnan, e H. K. Singh, «A novel method of vibration modes selection for improving accuracy of frequency-based damage detection», *Compos B Eng*, vol. 159, pp. 437–446, 2019, doi: 10.1016/j.compositesb.2018.08.134.
- [19] S. Walunj Prashant, V. N. Chougule, e A. C. Mitra, «Investigation on Modal Parameters of Rectangular Cantilever Beam Using Experimental Modal Analysis», *Mater Today Proc*, vol. 2, pp. 2121–2130, 2015, doi: 10.1016/j.matpr.2015.07.214.
- [20] D. Pasquale Pasca, «Ambient vibration testing of civil engineering structures: development of a Python module and applications to case studies», Universidade Norueguesa de Ciências da Vida, 2021.
- [21] R. Duris, «A determination of material properties of flexible structures using EMA and FEM analysis», *Applied Mechanics and Materials*, vol. 693, pp. 293–298, 2014, doi: 10.4028/www.scientific.net/AMM.693.293.
- [22] V. Nguyen e S. Melkote, «Hybrid statistical modelling of the frequency response function of industrial robots», *Robot Comput Integr Manuf*, vol. 70, p. 102134, 2021, doi: 10.1016/j.rcim.2021.102134.
- [23] W. Silva, *Vibration Monitoring, Testing, and Instrumentation*. Boca Raton, 2007. doi: <https://doi.org/10.1201/9781420053203>.
- [24] C. Ratcliffe, D. Heider, R. Crane, C. Krauthauser, M. K. Yoon, e J. W. Gillespie, «Investigation into the use of low cost MEMS accelerometers for vibration based damage detection», *Compos Struct*, vol. 82, pp. 61–70, 2008, doi: 10.1016/j.compstruct.2006.11.012.
- [25] Y. Peng, B. Li, X. Mao, H. Liu, C. Qin, e H. He, «A method to obtain the in-process FRF of a machine tool based on operational modal analysis and experiment modal analysis», *International Journal of Advanced Manufacturing Technology*, vol. 95, n. 9–12, pp. 3599–3607, 2018, doi: 10.1007/s00170-017-1405-8.
- [26] N. Instruments, «USB-4432». Acedido: 3 de Julho de 2023. [Em linha]. Disponível em: <https://www.ni.com/pt-pt/support/model.usb-4432.html>
- [27] National Instruments, «USB-4431». Acedido: 11 de Janeiro de 2023. [Em linha]. Disponível em: <https://www.ni.com/pt-pt/support/model.usb-4431.html>
- [28] C. J. Kalkman, «LabVIEW: A SOFTWARE SYSTEM FOR DATA ACQUISITION, DATA ANALYSIS, AND INSTRUMENT CONTROL», *J Clin Monit*, vol. 11, pp. 51–58, 1995.
- [29] W. Wang, C. Li, E. W. Tollner, e G. C. Rains, «Development of software for spectral imaging data acquisition using LabVIEW», *Comput Electron Agric*, vol. 84, pp. 68–75, 2012, doi: 10.1016/j.compag.2012.02.010.
- [30] MathWorks, «What Is MATLAB?» Acedido: 29 de Dezembro de 2022. [Em linha]. Disponível em: <https://www.mathworks.com/discovery/what-is-matlab.html>
- [31] P. Perakakis, M. Joffily, M. Taylor, P. Guerra, e J. Vila, «KARDIA: A Matlab software for the analysis of cardiac interbeat intervals», *Comput Methods Programs Biomed*, vol. 98, n. 1, pp. 83–89, 2010, doi: 10.1016/j.cmpb.2009.10.002.
- [32] P. Lesage, «Interactive Matlab software for the analysis of seismic volcanic signals», *Comput Geosci*, vol. 35, n. 10, pp. 2137–2144, 2009, doi: 10.1016/j.cageo.2009.01.010.
- [33] Python, «What is Python? Executive Summary». Acedido: 9 de Janeiro de 2023. [Em linha]. Disponível em: <https://www.python.org/doc/essays/blurb/>

- [34] D. Parbat e M. Chakraborty, «A python based support vector regression model for prediction of COVID19 cases in India», *Chaos Solitons Fractals*, vol. 138, p. 109942, 2020, doi: <https://doi.org/10.1016/j.chaos.2020.109942>.
- [35] A. Sheela e Kp. Meena, «Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB-A Comparative analysis», 2019.
- [36] J. Ranjani, A. Sheela, e K. Meena, «Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB», *1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pp. 1–5, 2019, doi: 10.1109/ICIICT1.2019.8741475.
- [37] van der Walt Stefan, Colbert S. Chris, e Varoquaux Gael, «The NumPy Array: A Structure for Efficient Numerical Computation», *Comput Sci Eng*, vol. 13, n. 2, pp. 22–30, 2011, doi: 10.1109/MCSE.2011.37.
- [38] PCB Piezotronics, «Impact Hammer». Acedido: 2 de Agosto de 2024. [Em linha]. Disponível em: <https://www.pcb.com/products?m=086c01>
- [39] PCB Piezotronics, «Accelerometer». Acedido: 2 de Agosto de 2024. [Em linha]. Disponível em: <https://www.pcb.com/products?m=352a24>
- [40] M. Caresta, «Vibrations of a Free-Free Beam». Acedido: 1 de Setembro de 2024. [Em linha]. Disponível em: http://www.varg.unsw.edu.au/Assets/link%20pdfs/Beam_vibration.pdf

Referências

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade. Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Declaro que o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P. PORTO.

Miguel Fernando Pinto Queirós

ISEP, Porto, 14 de setembro de 2024

Declaração de Integridade

Apêndice A

```
import aquisicao
import analise

# definir as sub-rotinas
def final():
    print("Executar o Programa de Aquisição e Processamento")

def analise():
    print("Executar o Programa de Análise")

# função principal para as sub-rotinas
def main():
    aquisicao() # executa o programa de aquisição e processamento
    analise() # executa o programa de análise

# função principal
if __name__ == "__main__":
    main()
```

Apêndice A

Apêndice B

```
import nidaqmx
import time
import numpy as np
import matplotlib.pyplot as plt
from nidaqmx import constants
from nidaqmx.constants import TerminalConfiguration, Edge, AcquisitionType,
SampleTimingType, ForceUnits, AccelUnits, AccelSensitivityUnits

# parametros da placa NI-USB 4431

# parametros da placa NI-USB 4432
sens_martelo = 11.2e-3 # V/N
sens_ace1 = 9.92e-3 * 9.81 # V/G

frfs = []
incremento = 0

time_out = 10.0 # time out
num_pre_samples = 6 # número de pré-samples
freq_analise = 1600 # Hz
freq_aquisicao = 1.5 * freq_analise * 10 #sample/s # Na 4432 a Freq máxima de aquisição é
102.4 kHz. A minima é 1 kHz
N_leituras = int(freq_aquisicao * time_out) # numero de leituras a realizar
threshold_impacto = 0.1
amostragem = 2
```

Apêndice B

```
amostras = int(amostragem*freq_aquisicao) # número de amostras a ler depois do impacto
valores_de_tempo_frf = np.arange(0, amostragem, amostragem/amostras)
freq_max = freq_analise
dt = 1/(freq_aquisicao)
indice_freq = int(freq_max/2)*amostragem + 1
tempo = np.arange(0, N_leituras/freq_aquisicao, 1/freq_aquisicao) # variavel de tempo total de
aquisição

# definir um nome para ficheiro de resultados que incorpore a data
named_tuple = time.localtime() # get struct_time
time_string = time.strftime("%Y_%m_%d_%H_%M_%S", named_tuple)
nome_ficheiro_resultados = 'Resultados_p_NI_USB_4432_' + time_string + '.txt' # Titulo criado
e indexado ao tempo (nome unico)

# Input do número de leituras desejadas
while True:
    try:
        n_leituras = int(input("Insira o número de leituras desejadas: "))
        if n_leituras <= 0:
            print("O número de leituras deve ser maior que zero.")
        else:
            break
    except ValueError:
        print("Por favor, insira um número inteiro válido.")

while len(frf) < n_leituras:
    incremento += 1

# ler os 2 primeiros canais da placa
def readdaq(N_pedido, freq_aquisicao_ped):
    Analog_read_task = nidaqmx.task.Task('Analog_read_task') # Iniciar o DAQ device
    (Tarefa)
```

```

# configuracao canal 0
ai_channel = Analog_read_task.ai_channels.add_ai_force_iepe_chan(
    'Dev1/ai0', name_to_assign_to_channel='force',
    terminal_config=TerminalConfiguration.DEFAULT,
    units=constants.ForceUnits.NEWTONS,
    min_val=-10,
    max_val=+10,
    sensitivity=1000,
    sensitivity_units=constants.ForceIEPESensorSensitivityUnits.MILLIVOLTS_PER_NEWTON,
    current_excit_val=2.100000e-3,
    custom_scale_name="")

# configuracao canal 1
ai_channel_1 = Analog_read_task.ai_channels.add_ai_accel_chan(
    'Dev1/ai1', name_to_assign_to_channel='acel',
    units=constants.AccelUnits.G,sensitivity= 1000,
    sensitivity_units=constants.AccelSensitivityUnits.MILLIVOLTS_PER_G)
ai_channel_1.ai_accel_db_ref=1
ai_channel_1.ai_coupling=constants.Coupling.AC

ai_channel_1.ai_excit_voltage_or_current=constants.ExcitationVoltageOrCurrent.USE_CURRENT
ai_channel_1.ai_excit_src=constants.ExcitationSource.INTERNAL
ai_channel_1.ai_excit_val=0.0021
ai_channel_1.ai_max=+10
ai_channel_1.ai_min=-10
ai_channel_1.ai_term_cfg=constants.TerminalConfiguration.PSEUDO_DIFF
ai_channel_1.ai_accel_units.METERS_PER_SECOND_SQUARED

# definir taxa
Analog_read_task.timing.cfg_samp_clk_timing(rate = freq_aquisicao_ped,
sample_mode=constants.AcquisitionType.CONTINUOUS,

```

Apêndice B

```
        samps_per_chan = N_pedido)

    Analog_read_task.sample_clk_active_edge=constants.Edge.RISING
    Analog_read_task.samp_timing_type=constants.SampleTimingType.SAMPLE_CLOCK

    Analog_read_task.start()

    valores_lidos_aq =
    Analog_read_task.read(number_of_samples_per_channel=int(N_pedido))
    Analog_read_task.stop()

    Analog_read_task.close() # Encerrar a tarefa

    return valores_lidos_aq # devolve os valores obtidas através da placa de aquisição

i_impacto=0
def detetar_impacto(data, threshold):
    # usar numpy.where para encontrar os índices onde o valor excede o threshold
    indices = np.where(data > threshold)[0]
    # print(indices)

    # verificar se algum índice foi encontrado e retornar o primeiro, caso contrário retornar -1
    if indices.size > 0:
        return indices[0] # devolve o primeiro índice encontrado
    else:
        return -1 # devolve o valor -1 se nenhum valor exceder o threshold

def processar_dados(valores_lidos, threshold_impacto, num_pre_samples):
    i_impacto = detetar_impacto(valores_lidos[0], threshold_impacto)
    if i_impacto != -1:
        start_index = max(i_impacto - num_pre_samples, 0)
        valores_lidos = valores_lidos[:, start_index:]
```

```

    # print(f"Start index: {start_index}")
else:
    print("Impacto não detetado")
return valores_lidos, i_impacto

# Plot das figuras
resol_dpi = 300

# leitura dos dados
valores_lidos_aq = np.array(readdaq(N_leituras, freq_aquisicao)) # leitura dos dados através
da placa de aquisição

i_impacto = detetar_impacto(valores_lidos_aq[0], threshold_impacto) # determinação do
instante em que acontece o impacto do martelo

t = tempo[i_impacto]

# reformulação dos valores de tempo e da placa de acordo com o indice de impacto
valores_de_tempo = tempo[i_impacto:]-t
valores_lidos = valores_lidos_aq[:, i_impacto:]

for i, valores in enumerate(valores_lidos):

    plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
    ax = plot.gca()
    plt.plot(tempo, valores_lidos_aq[i], '-r', label=f'Canal {i}')
    plt.title(f'Canal {i} - Valores Lidos')
    plt.xlabel('Tempo [s]')
    plt.ylabel('Tensão [V]')
    plt.legend()
    plt.grid(True)
    plt.show()

# processamento de dados

```

Apêndice B

```
# valores_lidos_processados = processar_dados(valores_lidos, threshold_impacto,
num_pre_samples)

valores_lidos_processados, i_impacto = processar_dados(valores_lidos_aq,
threshold_impacto, num_pre_samples)

if i_impacto != -1:
    sinal_forca = valores_lidos_processados[0,:amostras]/sens_martelo
    sinal_resposta = valores_lidos_processados[1,:amostras] /sens_ acel * 9.81
    sinal_resposta = (valores_lidos_processados[1, :amostras]) / sens_ acel * 9.81

else:
    print("Impacto não detetado")

# ponderação resposta
d = 30 # frecuencia de declive Hz
ponderacao = np.ones(amostras)
ponderacao[int(0.9 * amostras):] = 1 * np.exp(-d * valores_de_tempo_frff[:int(0.1 *
amostras)])
ponderacao[-10:] = 0

# ponderação forca
threshold_estabilizacao = 0.1

# encontrar o índice de estabilização
indice = None
for i in range(amostras):
    i = i + num_pre_samples
    if abs(sinal_forca[i]) < threshold_estabilizacao:
        indice = i+1
        break

# verificar se encontramos um índice de estabilização
```

if indice is not None:

```
# criar ponderacao_forca e definir valores após o índice de estabilização como zero
```

```
ponderacao_forca = np.ones(amostras)
```

```
ponderacao_forca[0] = 0
```

```
ponderacao_forca[indice:] = 0
```

else:

```
print("Nenhum índice de estabilização encontrado.")
```

função de ponderação

```
def fun_ponderacao(sinal, funcao_ponderacao):
```

```
    sinal_ponderado = funcao_ponderacao * sinal
```

```
    return sinal_ponderado
```

```
sinal_resposta_ponderado = fun_ponderacao(sinal_resposta, ponderacao)
```

```
sinal_forca_ponderado = sinal_forca * ponderacao_forca
```

plot sinal força

```
plt=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
```

```
ax = plt.gca()
```

```
plt.plot(valores_de_tempo_frf,sinal_forca)
```

```
ax = plt.gca()
```

```
ax.axes.set_xlim([0, 0.015])
```

```
plt.title('Sinal Não Ponderado - Força')
```

```
plt.xlabel('Tempo [s]')
```

```
plt.ylabel('Força [N]')
```

```
# plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```

plot sinal força ponderado

Apêndice B

```
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()
plt.plot(valores_de_tempo_frf,sinal_forca_ponderado)
ax = plot.gca()
ax.axes.set_xlim([0, 0.015])
plt.title('Sinal Ponderado - Força')
plt.xlabel('Tempo [s]')
plt.ylabel('Força [N]')
# plt.legend()
plt.grid(True)
plt.show()

# plot sinal resposta ponderado
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()
plt.plot(valores_de_tempo_frf,sinal_resposta_ponderado)
plt.title('Sinal Ponderado - Resposta')
plt.xlabel('Tempo [s]')
plt.ylabel('Aceleração [m/s^2]')
# plt.legend()
plt.grid(True)
plt.show()

# transformada de Fourier
def calcular_fft(sinal_ponderado, amostras, freq_max):
    fourier = np.fft.fft(sinal_ponderado)
    amplitude = np.abs(fourier)[:indice_freq + 1] ** 2
    fase = np.angle(fourier)[:indice_freq + 1]
    freq = np.arange(0,freq_max/2 + 1, 1/amostragem)

    return fourier, freq, amplitude, fase
```

```

    fourier_forca, freq_forca, amplitude_forca, fase_forca = calcular_fft(sinal_forca_ponderado,
amostras, freq_max)

    fourier_resposta, freq_resposta, amplitude_resposta, fase_resposta =
calcular_fft(sinal_resposta_ponderado, amostras, freq_max)

# cálculo dos autoespectros associados à força e resposta
auto_espectro_forca = fourier_forca * np.conj(fourier_forca)
auto_espectro_resposta = fourier_resposta * np.conj(fourier_resposta)

# plot autoespectro da força
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()
plt.plot(freq_forca, 20 * np.log10(auto_espectro_forca[:indice_freq + 1] * 1e6))
plt.title('FFT - Força')
plt.ylabel('Amplitude [dB]')
plt.xlabel('Freq. [Hz]')
# plt.legend()
plt.grid(True)
plt.show()

# plot amplitude da resposta
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()
plt.plot(freq_resposta, 20 * np.log10 (amplitude_resposta)) # plot da amplitude em decibel
[dB]
plt.title('FFT - Resposta')
plt.xlabel('Frequência [Hz]')
plt.ylabel('Amplitude')
# plt.legend()
plt.grid(True)
plt.show()

```

Apêndice B

```
# calcular FRF

frf_numerador = fourier_forca * np.conj(fourier_resposta)
frf_denominador = fourier_forca * np.conj(fourier_forca)
frf = frf_numerador / frf_denominador

# calcular amplitude e fase da FRF

frf_amplitude = np.abs(frf)
frf_fase = np.angle(frf)

frf_imag = np.imag(frf) # parte imaginária da FRF
frf_real = np.real(frf) # parte real da FRF

# plot FRF

plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()

plt.plot(freq_resposta, 20 * np.log10(frf_amplitude[:indice_freq + 1])) # plot da amplitude em
decibel [dB]

plt.title('FRF')
plt.xlabel('Frequência [Hz]')
plt.ylabel('Amplitude')
# plt.legend()
plt.grid(True)
plt.show()

# calcular a função de coerência

auto_espectro_resposta = fourier_resposta * np.conj(fourier_resposta) # GXX
auto_espectro_forca = fourier_forca * np.conj(fourier_forca) # GFF

coerencia = (np.abs(frf_numerador))**2 / (auto_espectro_forca * auto_espectro_resposta)
coerencia_abs = np.abs(coerencia)
```

```

# vetor = (frf_numerador)**2

# plot da função de coerência
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
ax = plot.gca()
plt.plot(freq_resposta, coerencia_abs[:indice_freq + 1])
plt.title('Função de Coerência')
plt.xlabel('Frequência [Hz]')
plt.ylabel('Coerência')
# plt.legend()
plt.grid(True)
plt.show()

confirmar = input(f"Leitura {incremento} está correta? (s/n): ").strip().lower()
if confirmar == 's':
    frfs.append(frf)
    # reiniciar variáveis para próxima iteração
    valores_lidos_aq = None
    valores_lidos_processados = None
    sinal_forca = None
    sinal_resposta = None
    print(f"Leitura {incremento} adicionada.")

# calcular a média das FRFs
if frfs:
    frf_media = np.mean(frfs, axis=0)

# Plot da média da FRF
plot=plt.figure(num=None, figsize=(6, 4), dpi=resol_dpi)
plt.plot(freq_resposta, 20 * np.log10(frf_media[:indice_freq + 1]))
plt.title('FRF Média')

```

Apêndice B

```
plt.xlabel('Frequência [Hz]')
```

```
plt.ylabel('Amplitude')
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```

```
# guardar os dados
```

```
dados = np.column_stack((np.abs(freq_resposta), np.real(frf_media[:indice_freq + 1]), np.imag(frf_media[:indice_freq + 1])))
```

```
np.savetxt(nome_ficheiro_resultados, dados, delimiter=',', header='Frequência, FRF - parte real, FRF - parte imaginária', comments='')
```

Apendice C

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.interpolate import interp1d

# características da estrutura unidimensional
largura = 596e-3 # m
n_pontos = 9

# função para ler os dados de FRF de um arquivo
def ler_frf(arquivo):
    with open(arquivo, 'r') as f:
        linhas = f.readlines()

    freq_resposta = []
    valores_reais = []
    valores_imaginarios = []
    for linha in linhas[1:]: # saltar a primeira linha (cabeçalho)
        partes = linha.strip().split(',')
        freq = float(partes[0])
        real = float(partes[1])
        imag = float(partes[2])
        freq_resposta.append(freq)
        valores_reais.append(real)
```

Apêndice C

```
valores_imaginarios.append(imag)

return np.array(freq_resposta), np.array(valores_reais), np.array(valores_imaginarios)

# frequência alvo para a extração do imaginário
freq_modulo = [92, 250.5, 490.5] # Hz

# ler os arquivos da FRF
for j in range(0, len(freq_modulo)):
    # armazenar os valores imaginários da frequência alvo
    modulo_vibracao = []

    for i in range(1, n_pontos + 1):
        arquivo = f'frf_{i}_1.txt'
        freq_resposta, val_reais, val_complexos = ler_frf(arquivo)

        # encontrar o índice da frequência
        ind_alvo = np.argmin(np.abs(freq_resposta - freq_modulo[j]))

        # obter o imaginário correspondente
        imaginario_freq = val_complexos[ind_alvo]
        real_freq = val_reais[ind_alvo]
        modulo_vibracao.append(imaginario_freq)

    # plot do módulo de vibração
    posicao = np.arange(0, 1.1*largura, largura/(n_pontos-1))

    # criação da função de interpolação cúbica
    interp_func = interp1d(posicao, modulo_vibracao, kind='cubic')

    # criação de um array de posições para a interpolação
```

```
posicao_fina = np.linspace(posicao.min(), posicao.max(), num=500)
modulo_vibracao_interpolado = interp_func(posicao_fina)

# plot dos dados originais e dos dados interpolados
plt.figure(figsize=(10, 6))
plt.plot(posicao, modulo_vibracao, 'o', label=f'Valor Imaginário a {freq_modulo[j]} Hz')
plt.plot(posicao_fina, modulo_vibracao_interpolado, '-', label='Modo de
Vibração', color='blue')
# plt.title("")
plt.xlabel('Posição')
plt.ylabel('Imaginário')
plt.legend()
plt.grid(True)
plt.show()
```

