

**UNIVERSIDADE FEDERAL DE SÃO CARLOS - UFSCar
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA - CCET
DEPARTAMENTO DE ENGENHARIA MECÂNICA - DEMec**

ORLANDO DE SOUSA BARBOSA NETO

**Desenvolvimento de ferramenta computacional baseada em
Multiple Instance Learning para análise de vibrações mecânicas
através de uma câmera digital**



São Carlos - SP
2022

ORLANDO DE SOUSA BARBOSA NETO

**Desenvolvimento de ferramenta computacional baseada em
Multiple Instance Learning para análise de vibrações mecânicas
através de uma câmera digital**

Trabalho de conclusão de curso apresentado ao curso de graduação em Engenharia Mecânica da Universidade Federal de São Carlos, para obtenção do título de bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. Sidney Bruce Shiki

São Carlos - SP

2022



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905

Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 14/2022/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

ORLANDO DE SOUSA BARBOSA NETO

**DESENVOLVIMENTO DE FERRAMENTA COMPUTACIONAL BASEADA EM MULTIPLE INSTANCE LEARNING
PARA ANÁLISE DE VIBRAÇÕES MECÂNICAS ATRAVÉS DE UMA CÂMERA DIGITAL**

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 29 de abril de 2022

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	Sidney Bruce Shiki
Membro da Banca 1	Flávio Yukio Watanabe
Membro da Banca 2	Luis Antonio Oliveira Araujo



Documento assinado eletronicamente por **Sidney Bruce Shiki, Docente**, em 29/04/2022, às 11:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Flavio Yukio Watanabe, Docente**, em 29/04/2022, às 11:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luis Antonio Oliveira Araujo, Docente**, em 29/04/2022, às 11:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **0665178** e o código CRC **12154D36**.



Referência: Caso responda a este documento, indicar expressamente o Processo nº
23112.011454/2022-68

SEI nº 0665178

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

DEDICATÓRIA

Dedico este trabalho à minha família, a quem agradeço todo o carinho e suporte durante este ciclo da minha vida.

AGRADECIMENTOS

Agradeço a todas as pessoas que de alguma forma participaram da minha graduação, incluindo minha família e amigos, por tornarem este momento da minha vida inesquecível.

Agradeço aos meus pais, Maria Filomena Barbosa e Reginaldo Barbosa pelo carinho, suporte ao longo do meu período de graduação, em especial nos momentos de maior dificuldade e tensão. Grato também por me proporcionarem todos os meios para eu alcançar meus objetivos.

Agradeço também o meu orientador Prof. Dr. Sidney Bruce Shiki, sem o qual a elaboração deste trabalho não teria sido possível. Grato por toda a sua ajuda ao longo da produção desta monografia.

Agradeço ainda meus companheiros de turma, em especial Lucas Bonventi, Gabriel Blanco, Fabrício Uemura e Vitor Duarte, pelo companheirismo durante os anos da graduação.

"A persistência é o melhor caminho para o êxito."

(Charles Chaplin)

RESUMO

No campo da engenharia existe uma necessidade de monitoramento e avaliação de equipamentos que estão em constante funcionamento. Um dos problemas comumente enfrentados é a variação de vibrações em componentes, de modo que isso possa causar algum tipo de falha precoce no elemento mecânico, pois variando as vibrações, também ocorre uma mudança nas forças dinâmicas aplicadas ao equipamento. Porém, existe um alto custo de aquisição, instalação e pessoas bem preparadas para realizar este tipo de manutenção. Dessa maneira, é importante um estudo e desenvolvimento de novas técnicas que possibilitem este tipo de operação de prevenção de falhas prematuras. Pensando nisso, é proposto um método de baixo custo pouco invasivo, no qual necessita-se de apenas uma câmera para realização das filmagens do equipamento em funcionamento e um computador para o tratamento da imagem e análise de dados, de modo que a medição seja sem contato. Neste presente trabalho é desenvolvido um algoritmo em Python para identificar o posicionamento de um objeto de interesse na filmagem, para posteriormente poder fazer uma análise do espectro em frequência. Espera-se poder comparar os resultados obtidos através da técnica de imagem, com algum modelo de acelerômetro profissional, de maneira que possam ser encontrados valores de frequências satisfatoriamente próximas. Tendo isso em vista, foram realizados testes em uma viga engastada. Uma entrada impulsiva foi aplicada com um martelo de impacto de modo que foi possível realizar a medição da vibração em pontos equi-espaçados da viga. Essa medição foi realizada tanto pelo acelerômetro, quanto pela técnica de imagem, posteriormente as frequências foram comparadas.

Palavras-chave: Vibrações. Técnica de imagem. Medição sem contato.

ABSTRACT

In the engineering field, there is a need for monitoring and evaluating equipment that is in constant operation. A type of mechanism can cause problems in some components, so it can also cause a failure in the mechanical element, varying early as changes, a change in the operations applied to the equipment can occur. However, there is a high cost of acquisition, installation and people well prepared to perform this type of maintenance. Thus, it is important to study and develop new techniques that enable this type of operation to prevent premature failures. With this in mind, a low-cost method is proposed, without the need for only a low cost to carry out the filming of the equipment in operation and a computer for the treatment of the image and the analysis of data, so that it can be thought without contact. In this work, a spectrum is developed in Python to identify the positioning of an object of interest in the footage, in order to later make a frequency analysis. It is expected to compare the results obtained through the imaging technique, with a professional accelerometer model, so that frequency values can be found satisfactorily close. With this in mind, tests were carried out on a cantilevered beam. An impulsive input was applied with an impact hammer so that it was possible to perform a vibrating motion at equi-spaced points of vibration. This was performed differently, both by the accelerometer and by the imaging technique, later performed as frequency.

Palavras-chave: Vibrations. Image technique. Non-contact measurement.

Lista de Figuras

Figura 1 – Sistema massa-mola-amortecedor	4
Figura 2 – Exemplo espectro e domínio do Tempo	5
Figura 3 – Amostragem de sinal com e sem <i>Aliasing</i>	7
Figura 4 – Ilustração do algoritmo de <i>block matching</i>	8
Figura 5 – Representação de regiões positivas e negativas	9
Figura 6 – Fluxograma do algoritmo	10
Figura 7 – Bancada de Testes	13
Figura 8 – Vista superior da viga	14
Figura 9 – Fixação do acelerômetro na viga	15
Figura 10 – Interface	16
Figura 11 – Obtenção do comprimento em pixel	17
Figura 12 – Seleção do alvo	18
Figura 13 – Caixa de seleção do objeto	19
Figura 14 – Enumeração das Etiquetas	20
Figura 15 – Resposta no Domínio do Tempo	21
Figura 16 – Sinal temporal do acelerômetro	22
Figura 17 – Espectro em frequência através do acelerômetro	23
Figura 18 – Representação visual do modelo criado	24
Figura 19 – Espectro em frequência das etiquetas	26
Figura 20 – <i>Error Bar</i> para as duas frequências nos 5 testes realizados	29

Lista de Tabelas

Tabela 1 – Especificações do Acelerômetro PCB333B30	14
Tabela 2 – Características do módulo NI9234	15
Tabela 3 – Informações geradas da malha	24
Tabela 4 – Frequência e modos de vibração, simulação	25
Tabela 5 – Frequências predominantes para as 6 etiquetas encontradas no teste 1	27
Tabela 6 – Frequências encontradas para os demais testes	28

Lista de Siglas

BSD	<i>Berkeley Software Distribution</i>
CCD	Dispositivo de carga acoplada (<i>Charge Coupled Device</i>)
DFT	<i>Discrete Fast Fourier Transform</i>
FPS	<i>Frames per second</i>
FFT	<i>Fast Fourier Transform</i>
GUI	<i>Graphical User Interface</i>
IEPE	<i>Integrated Electronics Piezo-Electric</i>
HCI	<i>Human-Computer Interaction</i>
MIL	<i>Multiple Instance Learning Track</i>
USB	<i>Universal Serial Bus</i>

Lista de Símbolos

$x(t)$	Deslocamento
m	Massa
C	Constante de amortecimento
ζ	Fator de amortecimento
k	Rigidez
ω	Frequência angular
ωd	Frequência amortecida
ωn	Frequência natural não amortecida
A	Amplitude
N	Quantidade de amostras do sinal
n	Índice do dado coletado no sinal
Xk	k-ésima frequência

Sumário

1	INTRODUÇÃO	1
	1.1 Objetivos gerais	2
	1.2 Objetivos específicos	2
2	REVISÃO BIBLIOGRÁFICA	3
	2.1 Vibrações Mecânicas	4
	2.2 Teorema de Nyquist.....	6
	2.3 Algoritmo de <i>block matching</i> para medição de movimento	7
	2.4 Algoritmo de <i>Multiple Instance Learning</i> (MIL)	8
3	FERRAMENTAS COMPUTACIONAIS PARA MEDIÇÃO DE DESLOCAMENTOS	
	10	
	3.1 Python	11
	3.2 NumPy	11
	3.3 Matplotlib.....	11
	3.4 Tkinter.....	12
	3.5 OpenCV	12
4	DESCRIÇÃO DO TESTE EXPERIMENTAL	13
	4.1 Bancada de testes	13
	4.2 Interface desenvolvida em Python	16
	4.3 Funcionamento do Algoritmo	18

5	RESULTADOS E DISCUSSÃO.....	20
6	CONSIDERAÇÕES FINAIS	30
	REFERÊNCIAS.....	31
	APÊNDICE A – ALGORITMO UTILIZADO NO PROCESSAMENTO DOS VÍDEOS	33
	APÊNDICE B – FFT APLICADO AO ACELERÔMETRO.....	39

1 INTRODUÇÃO

A vibração mecânica pode ser definida como um tipo de movimento no qual se considera uma massa reduzida a um ponto submetido a uma força. A ação dessa força obriga o ponto a executar um movimento oscilatório. Para que o movimento oscilatório do ponto se consitua numa vibração, ele deverá percorrer uma trajetória denominada trajetória completa ou ciclo, conhecida pelo nome de período de oscilação (Benedetti, 2002).

A partir dos dados obtidos através de uma análise de vibração, é possível identificar problemas capazes de prejudicar o desempenho dos equipamentos, o que permite reconhecer quais máquinas necessitam de mais atenção da manutenção. As informações obtidas são fundamentais, uma vez que possibilitam uma avaliação efetiva das condições dos ativos. Dessa forma, o gestor de manutenção consegue evitar defeitos e falhas inesperadas no maquinário, garantindo seu funcionamento adequado. Carneiro (2003) avalia as vantagens de diversas destas técnicas e conclui que a análise de vibrações é a técnica mais usada, porque se adequa a diversos tipos de equipamentos e permite identificar um elevado número de anomalias sem influenciar a disponibilidade das máquinas.

De modo geral, são utilizados acelerômetros profissionais que são fixados em alguma região de interesse permitindo a leitura do sinal da aceleração imprimida ao conjunto. Muitas das técnicas de processamento de sinais consistem de transformações de funções entre dois espaços de definição. A transformada de Fourier, transforma uma função no espaço temporal para um espaço cuja base são funções seno e cosseno complexas, se tornando a ferramenta mais importante no processamento de sinais devido ao uso amplo e consolidado do espectro de frequências para análise de um sinal (Silva, 2013). Porém, o uso de acelerômetros possuem algumas limitações. A adição do instrumento de medição ao objeto pode influenciar nas propriedades do sistema e interferir nos valores medidos, já que podem alterar a massa do sistema. Além disso, é possível citar o alto custo de um acelerômetro profissional e hardware necessários para medições. Dessa forma, o presente trabalho reside principalmente na verificação da estratégia de baixo custo para o cálculo de deslocamento, confrontando o sinal obtido com um acelerômetro.

1.1 Objetivos gerais

O trabalho tem como objetivo implementar e verificar um algoritmo para medição de deslocamentos baseados no método *Multiple Instances Learning* para análise de sinais de vibrações. Dessa maneira, sabendo o deslocamento é possível construir o espectro em frequência para identificar as frequências de vibração. As frequências encontradas são comparadas com os valores obtidos através de um acelerômetro comercial.

1.2 Objetivos específicos

- Estudar técnicas de rastreamento para medição de movimento em vídeo;
- Implementar o algoritmo de *Multiple Instances Learning* (MIL) para medição de movimento;
- Implementar a transformada de Fourier para obter o espectro em frequência a partir do sinal de deslocamento;
- Efetuar ensaios em uma estrutura vibratória simplificada para teste da metodologia;
- Confrontar características do sinal medido a partir de processamento de vídeo com as vibrações mecânicas obtidas a partir de um acelerômetro profissional de referência;

2 REVISÃO BIBLIOGRÁFICA

Sistemas mecânicos e estruturais são encontrados em diferentes áreas da engenharia e estão suscetíveis aos fenômenos de vibração, os quais em sua maioria devem ser medidos e analisados. O entendimento do comportamento vibratório desses sistemas é de fundamental importância no desenvolvimento de projetos, na obtenção de construções seguras, na utilização e operação de uma variedade de máquinas e estruturas (RAO, 2007). Em alguns casos, as vibrações são esperadas e desejadas no funcionamento de alguns equipamentos. No entanto, em alguns casos, as vibrações causam muitos problemas indesejáveis, podendo citar fadigas estruturais, desgaste excessivo em rolamentos e engrenagens, que ocasionam a diminuição da vida útil do sistema e de seus elementos.

Devido a esses fatores, diferentes técnicas são utilizadas para a medição de vibrações, podendo ser divididas em sensoriamento de contato, que seria o caso por exemplo dos acelerômetros e extensômetros. Estes sensores necessitam estar acoplados aos sistemas medidos, gerando alterações como por exemplo a mudança de massa do sistema. E também temos o sensoriamento sem contato, que não necessita de nenhum tipo de contato físico com o sistema, como por exemplo sensores de aproximação e o vibrômetro laser Doppler, que é utilizado para medição de vibração através da reflexão de um feixe que é apontado para a superfície de interesse (LIU et al., 2012). Estes sensores possuem um custo mais elevado, quando comparado com os sensores de contato tradicionais.

A facilidade na obtenção, na utilização e a significativa redução nos preços das câmeras digitais comerciais, contribuíram para o surgimento de novos trabalhos envolvendo análise de vibrações utilizando imagens. Olaszek (1999) foi um dos primeiros a utilizar câmeras de vídeo convencionais em conjunto com técnicas de visão computacional para analisar as características dinâmicas de pontes. Outros estudos abordaram as medições de vibrações no espaço tridimensional (3D), podendo-se citar como exemplos os trabalhos desenvolvidos por Ryall e Fraser (2002) que, com apenas uma câmera CCD perfeitamente sincronizada com a iluminação estroboscópica, determinaram, através da triangulação de pontos do objeto em diferentes fases, os modos de vibração tridimensional (3D) da asa de um avião.

Portanto, nota-se que o estudo dos conceitos relacionados a vibrações mecânicas é de extrema importância para o conhecimento dos Engenheiros Mecânicos que atuam principalmente na área de manutenção preditiva, com ênfase em análise de vibrações, pois estes

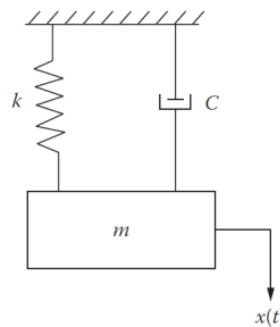
irão se deparar com máquinas ou estruturas que estão submetidas a diversos fenômenos vibratórios.

2.1 Vibrações Mecânicas

A vibração pode ser definida como uma oscilação de um corpo em torno de uma posição de referência. Esse movimento em uma máquina é a resposta às forças dinâmicas que a excitam, e frequentemente é um processo destrutivo, como alerta (ALMEIDA e ALMEIDA, 2014), ocasionando falhas nos elementos de máquinas por fadiga, trincas, falhas por ruptura, dentre outros.

A Figura 1 representa um sistema mecânico simples (massa-mola-amortecedor) que contém os três elementos básicos que interagem entre si e são responsáveis pelo comportamento dinâmico. Esses elementos são: mola (rigidez), amortecedor (dissipação de energia), e massa (inércia). Quando submetidos a uma excitação eles reagem com deslocamento, velocidade e aceleração.

Figura 1 – Sistema massa-mola-amortecedor



Fonte: MOHANTY, 2015

Quando a massa é afastada da sua posição de equilíbrio e largada, sem interferência de forças externas, o seu deslocamento é dado pela equação (1):

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0 \quad (1)$$

Onde, m é a massa, c a constante de amortecimento do conjunto, e k a rigidez do sistema.

Resolvendo essa equação, a resposta desse movimento é descrita por (2):

$$x(t) = A^{-\zeta\omega t} \sin(\omega_d t + \phi) \quad (2)$$

Onde, o fator de amortecimento é expresso conforme (3):

$$\zeta = \frac{c}{2\sqrt{km}} \quad (3)$$

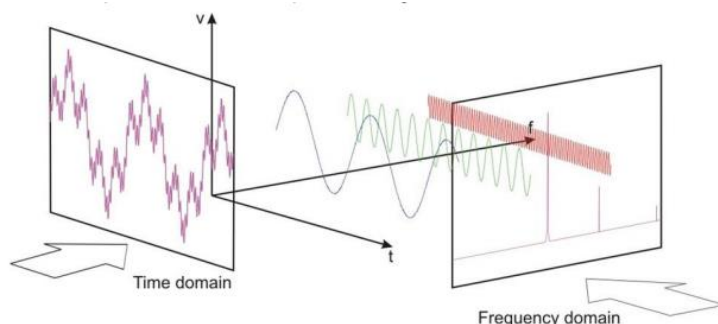
E a frequência natural pela equação (4):

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}, \text{ onde, } \omega_n = \sqrt{\frac{k}{m}} \quad (4)$$

Em um sistema mais aproximado da realidade, não existe apenas uma frequência característica. Nestas condições a forma de onda resultante não será uma senoidal pura e pode ser bem mais complexa. A representação dos fenômenos vibratórios no domínio do tempo, especialmente quando as fontes de sinal são diversas, torna difícil a interpretação do que está acontecendo.

Então, para solucionar este problema, o sinal no domínio do tempo pode ser passado para o domínio da frequência, facilitando a identificação de algumas características do sinal, como por exemplo amplitudes e frequências dominantes. A forma mais comum de se fazer isso nas vibrações é utilizando o espectro em frequência. Dentro da análise de vibrações, a avaliação do espectro de frequência é uma das técnicas mais utilizadas. Os sinais de vibração adquiridos em uma série de amostras em função do tempo podem ser representados por uma soma de senoides de várias frequências e fases, conforme apresenta de forma simplificada a Figura 2.

Figura 2 – Exemplo espectro e domínio do tempo



Fonte: ARTECH ENGENHARIA E INSTRUMENTAÇÃO, 2016

Com a evolução dos sistemas digitais, adaptou-se a teoria de Fourier para o tratamento de dados amostrados (sinais discretos). No tratamento de dados discretos, usa-se a Transformada Discreta de Fourier (DFT – *Discrete Fourier Transform*). Supondo que a sequência discreta $x[n]$ é nula para $n < 0$ e $n \geq N$ onde N é um inteiro positivo. A DFT X_k de $x[n]$ é definida por (KAMEN & HECK, 2000):

$$X_k = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, k = 0, 1, \dots, N - 1 \quad (5)$$

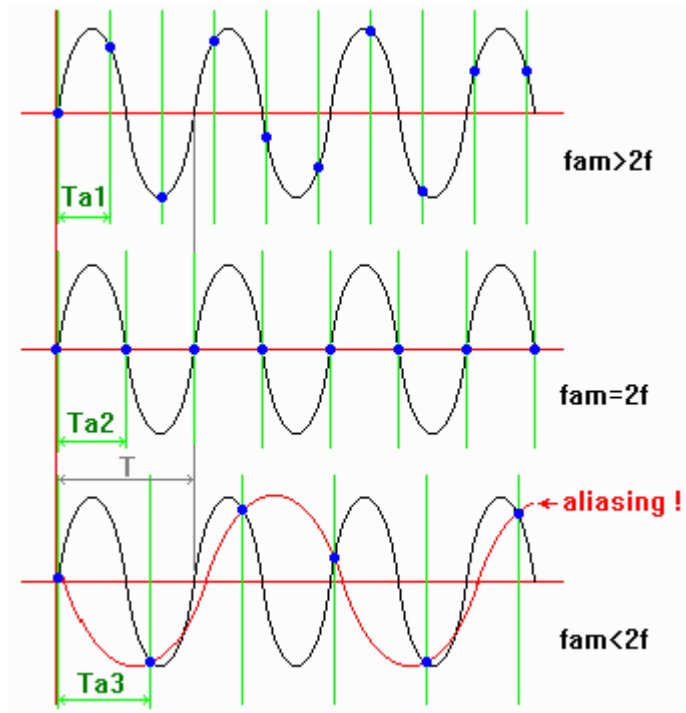
Para o cálculo da DFT existe um algoritmo bastante eficiente que ficou conhecido como FFT (*Fast Fourier Transform*), sendo o algoritmo que implementa o procedimento matemático que permite a partir de um sinal no domínio temporal $x(t)$ obter uma função $X(f)$ ou na forma complexa $X(j\omega)$, descrevendo o comportamento em frequência desse sinal (KAMEN & HECK, 2000). Esta atribuição facilitou muito o tratamento de sinais amostrados com grande número de pontos. O algoritmo da FFT está descrito em DeFatta (1998).

2.2 Teorema de Nyquist

Com a finalidade de garantir que o processo de amostragem seja sempre realizado de forma adequada, deve-se observar o teorema de Nyquist, o qual recomenda que para a aquisição correta de um sinal que varia continuamente no tempo, é necessária uma taxa de amostragem mínima de duas vezes a frequência máxima que se deseja estudar. Esta taxa de amostragem mínima é conhecida como “frequência de Nyquist” (LATHI, 2000).

Durante a amostragem de um sinal, para obter a relação no domínio da frequência, pode surgir o fenômeno *aliasing* que ocorre quando um sinal é amostrado com uma frequência menor que o dobro da frequência do próprio sinal. Quando isso ocorre, existe um “batimento” entre a frequência de amostragem e o sinal amostrado que gera uma frequência inferior aquela efetivamente aplicada. (OPPENHEIM, 2012). Isto pode ser visualizado na Figura 3, onde “ f_m ” indica a frequência de amostragem e “ f ” indica a frequência do sinal amostrado. Para eliminar esse problema, a frequência de amostragem deve ser no mínimo duas vezes maior que a maior frequência do sinal.

Figura 3 – Amostragem de sinal com e sem *Aliasing*



Fonte: <https://www.qsl.net/py4zbx/teoria/digitaliz.htm>

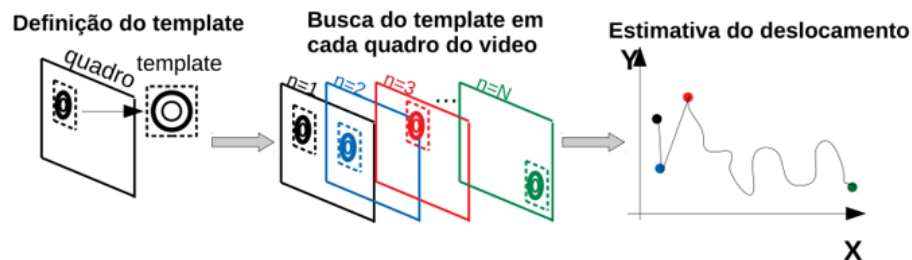
Portanto, como a câmera utilizada neste trabalho é uma comum de celular, cuja taxa de amostragem é de 30 fps, a análise fica limitada a apenas vibrações que possuem no máximo uma frequência de 15 Hz, de acordo com o teorema de Nyquist.

2.3 Algoritmo de *block matching* para medição de movimento

O algoritmo de *block matching* é frequentemente utilizado para estimativa de movimento a partir de filmagens (JOSEPH et al., 2017). Este método consiste em definir uma porção da imagem, que pode ser denominada como *template* o qual será definido em um dos quadros do video capturado. Os *templates* são porções da imagem que serão buscados em cada um dos *frames* que formam o video de modo que sua posição ao longo do tempo representa a posição instantânea de um determinado ponto. Nessa técnica, os quadros da sequência de video são analisados aos pares, a fim de detectar os movimentos de um quadro para o outro. O *template* é dividido em varios blocos, em que cada bloco será comparado, dentro de uma janela de busca própria na imagem anterior, com outros blocos a fim de encontrar o melhor *matching*

(casamento). O algoritmo de *block matching* é ilustrado na Figura 4 para a estimativa do deslocamento de uma determinada região.

Figura 4 – Ilustração do algoritmo de *block matching*



Fonte: Autor

Porém, este método possui algumas limitações. Se o *template* estiver em um local com uma iluminação um pouco menos favorecida ou que contenha qualquer outro objeto que de certa forma seja parecido com o alvo que se deseja capturar o movimento, o rastreamento do *template* é prejudicado, perdendo totalmente a referência. Além disso, este algoritmo implementado no *software* Matlab foi testado em trabalhos anteriores e constatou-se que exigia um grande tempo de processamento do computador devido à necessidade de um mapeamento de um vídeo quadro-a-quadro buscando-se o *template* de forma exaustiva (BARBOSA, 2019). Dessa maneira, foi implementado em um algoritmo alternativo em linguagem Python, conhecido como *Multiple Instance Learning* (MIL), pois funciona bem mesmo quando ocorre oclusão parcial do objeto e apresentou não ter dificuldades no rastreamento das etiquetas utilizadas neste trabalho.

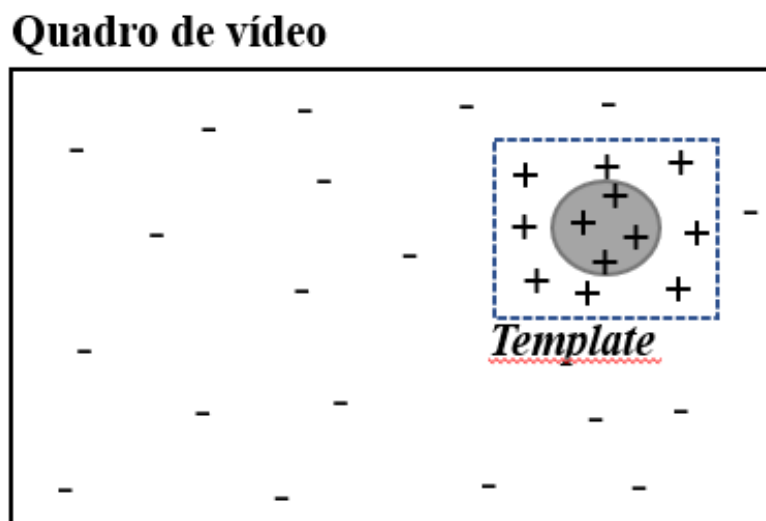
2.4 Algoritmo de *Multiple Instance Learning* (MIL)

Este algoritmo, apesar de parecido com *block matching*, funciona com base em *Machine Learning* ou aprendizado de máquinas. Esse tipo de Sistema tem o objetivo de aprender à tomar decisões com o mínimo de intervenção humana. Existem diversos algoritmos de *machine learning* e abordagens, neste caso em específico, um algoritmo preditivo ou supervisionado.

O algoritmo supervisionado é uma função que, a partir de conjuntos de dados já rotulados e conhecidos, constrói um modelo que consegue estimar rótulos de novos dados (CARVALHO et al., 2011).

Neste método, dada uma imagem, deve-se decidir se ela pertence ou não a uma classe de alvos, aqui chamado de *template*, com base em seu conteúdo visual. Tome como exemplo uma parte de uma imagem que contenha uma etiqueta circular, ou seja, o *template* neste caso será uma etiqueta. A região da imagem que está contida no *template*, será rotulada como positiva, enquanto as imagens rotuladas negativas serão as que apresentam qualquer outro tipo de conteúdo visual, como é mostrado na Figura 5. Portanto, cada região será composta por um conjunto de rótulos positivos e negativos, que são representados de forma binária 1 ou 0 respectivamente.

Figura 5 – Representação de regiões positivas e negativas



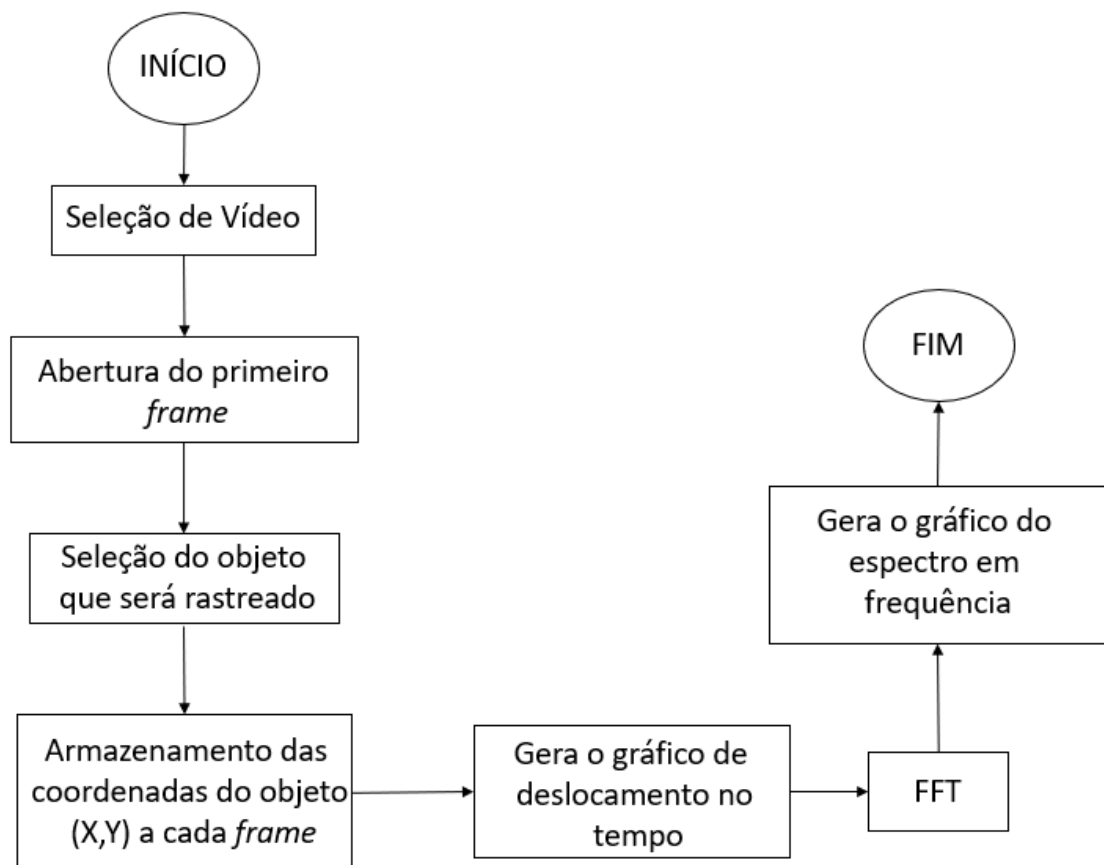
Fonte: Autor

A partir disso, é estabelecido um modelo de classificação, com base na rotulação obtida pelo primeiro *frame*. Este modelo é aplicado a cada *frame* de vídeo, fazendo a captura da nova posição do *template*. Desse modo, o *template* consegue ser atualizado a cada *frame*, deixando o algoritmo mais robusto. Por fim, o algoritmo retorna um vetor com as posições do *template* ao longo de todo o tempo. Vale lembrar que o *template* neste trabalho será definido como sendo as etiquetas.

3 FERRAMENTAS COMPUTACIONAIS PARA MEDIÇÃO DE DESLOCAMENTOS

A aplicação de ferramentas computacionais foi imprescindível para o desenvolvimento dos métodos e a geração dos resultados deste trabalho. O uso da linguagem de programação Python acompanhada principalmente da biblioteca OpenCV permitiram o tratamento das imagens de forma adequada. Também é possível citar a Numpy e Matplotlib que foram fundamentais para manipulações de dados e plotagem. Por fim, foi usado a biblioteca Tkinter para desenvolver uma interface simples para que o usuário possa se utilizar deste algoritmo de forma mais intuitiva. O fluxograma que apresenta de forma sintetizada o funcionamento do algoritmo pode ser visto na Figura 6.

Figura 6 – Fluxograma do algoritmo



Fonte: Autor

3.1 Python

A escolha de utilizar Python ao invés de usar uma ferramenta mais específica, como o Matlab, baseia-se em alguns fatores. Python é uma linguagem que possui seu desenvolvimento em código aberto, sendo mantida pela *Python Software Foundation*, uma organização sem fins lucrativos. Por outro lado, o *software* Matlab é pago, sendo que sua licença possui um custo elevado. Além disso, a implementação do algoritmo de medição de deslocamento em Python, que é utilizado neste trabalho, se torna mais rápida e prática. Dessa forma, um possível protótipo se torna livre do uso de uma ferramenta paga, o que contribui para o conceito de ferramenta de baixo custo.

Um outro ponto importante é o fato que o uso de Python tem aumentado de forma significativa nos últimos anos. A revista IEEE Spectrum, que anualmente publica a lista das melhores linguagens de programação baseada em diversos fatores, posicionou Python em primeiro lugar em 2017, a frente de outras linguagens como C, C++ e Java (CASS; 2017)

3.2 NumPy

Numpy é um pacote para computação científica com Python, disponível no site <http://www.numpy.org/>, com licença BSD (do inglês, *Berkeley Software Distribution*) e o uso com poucas restrições. Possui funções sofisticadas e é bem empregado nas operações de matrizes n-dimensionais, faz a integração entre códigos escritos em C/C++ e Fortran, faz a combinação fácil e rápida e entre diversos tipos de bancos de dados, eficiente em álgebra linear, transformada de Fourier e números aleatórios.

3.3 Matplotlib

Matplotlib é uma biblioteca de plotagem 2D do Python, disponível no site <https://matplotlib.org/>. Essa ferramenta é muito prática, pois gera gráficos dos mais variados tipos (barras, dispersão, erro), histogramas, espectros de potência com algumas linhas de código.

3.4 Tkinter

Tkinter é uma biblioteca da linguagem Python que acompanha a instalação padrão e permite desenvolver interfaces gráficas. Isso significa que qualquer computador que tenha o interpretador Python instalado é capaz de criar interfaces gráficas usando o Tkinter.

3.5 OpenCV

OpenCV (do inglês, Open Source Computer Vision Library) é uma biblioteca de código aberto, disponível no site <http://opencv.org/> com licença BSD composta por centenas de algoritmos de Visão Computacional. Foi oficialmente lançado como um projeto de pesquisa pela *Intel Research*. É totalmente livre para uso acadêmico e comercial, tornando-se popular e possibilitando o desenvolvimento de programas em visão computacional. Apresenta interfaces C, C++, Python e Java e suporta Windows, Linux, Mac OS, iOS e Android.

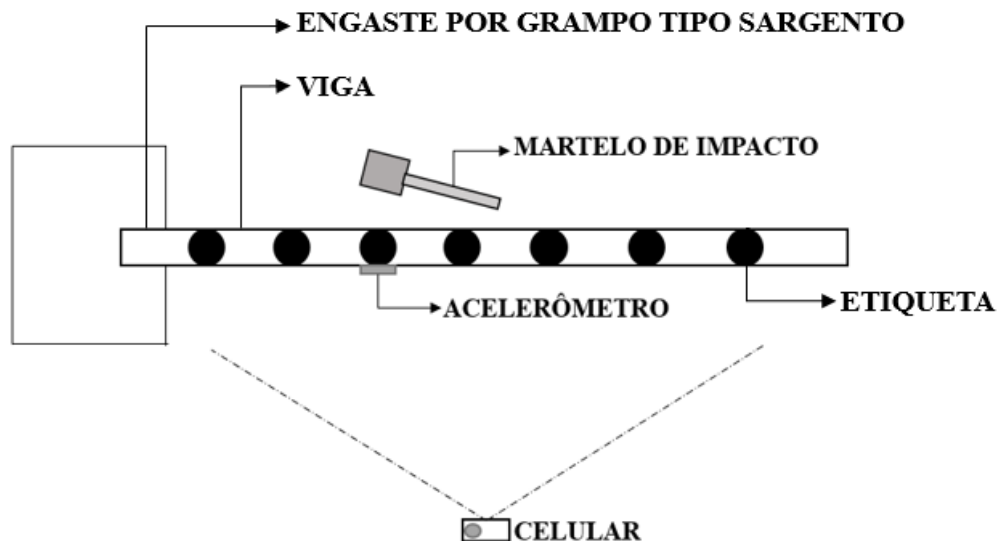
A sua biblioteca possui mais de 2500 algoritmos otimizados e mais de 14 milhões de downloads. É utilizada em todo o mundo e possui uma comunidade de usuários superando 47 mil pessoas. Possui módulos de processamento de imagens, vídeo, estrutura de dados, álgebra linear, GUI (do inglês, *Graphical User Interface*) básica, controle de mouse e teclado, algoritmos de visão computacional (filtros de imagem, calibração de câmeras, reconhecimento de objetos, análise estrutural, entre outros). Esta ferramenta possui diversas aplicações como: identificação de objetos, Interface Humano-Computador (HCI, do inglês, *Human-Computer Interaction*), sistema de reconhecimento facial, reconhecimento de movimentos, gravação de vídeos, robôs móveis, reconstrução 3D, realidade virtual, realidade aumentada e realidade misturada. Dessa biblioteca foi possível extrair e adaptar o algoritmo *MIL-Tracker* para fazer o rastreamento em vídeo.

4 DESCRIÇÃO DO TESTE EXPERIMENTAL

4.1 Bancada de testes

Foi desenvolvida uma bancada para realizar os testes utilizados neste trabalho, sendo todos feitos no Laboratório de Engenharia Mecânica da UFSCar em São Carlos. A Figura 7 ilustra a bancada por uma vista superior. Foi desenvolvida com a finalidade de medir a vibração através do acelerômetro e para captura de filmagens para posteriormente poder comparar os resultados.

Figura 7 – Bancada de Testes



Fonte: Autor

Dessa forma, uma viga com as dimensões de 670 (Comprimento em balanço) x 29,6 x 1,5 mm foi engastada com um sargento. Uma entrada impulsiva foi aplicada com um martelo de impacto instrumentado à 200 mm do engaste. Nessa mesma posição, a vibração da viga em termos de aceleração foi capturada.

A viga possui 15 etiquetas coladas para fins de captura de movimento, elas estão espaçadas ao longo do comprimento da viga, a Figura 8 ilustra a viga através de uma vista superior, observe que ela já está um pouco deformada. Para filmagem foi usado um Motorola Moto G5S plus, o vídeo foi capturado com 30 fps e resolução 1080x1920 pixels. Os testes foram repetidos 5 vezes.

Figura 8 – Vista superior da viga



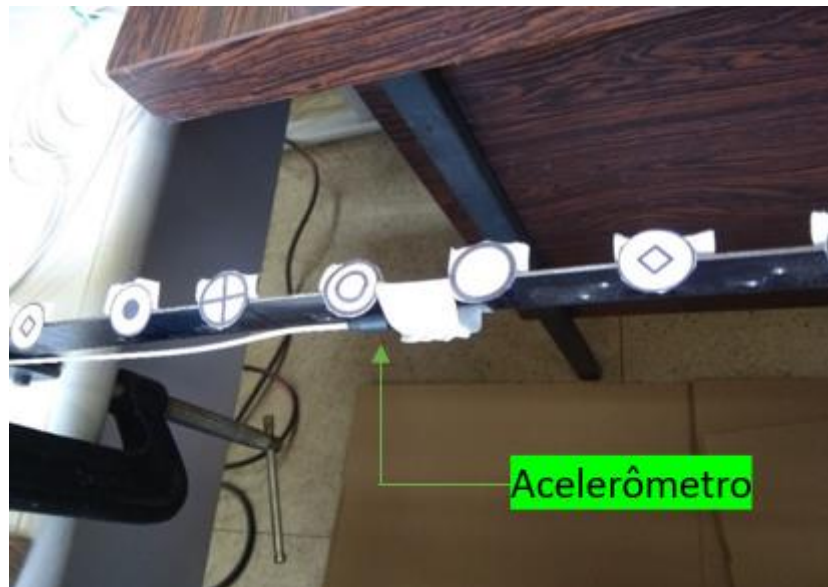
Fonte: Autor

Para aplicações em análise de vibração o tipo piezoelétrico é um dos mais utilizados. Este tipo de acelerômetro é composto, normalmente, uma cerâmica ferroelétrica polarizada artificialmente. Quando mecanicamente tensionada, proporciona uma carga proporcional à força aplicada, gerando uma carga elétrica que polariza suas faces. Os acelerômetros piezoelétricos não necessitam de fonte de alimentação, ou seja, o sinal de saída pode ser conectado diretamente ao medidor de vibrações, além disso, não possuem partes móveis e geram um sinal proporcional à aceleração. O modelo instalado na viga é o PCB333B30 e suas principais características estão descritas na Tabela 1. A fixação do acelerômetro pode ser melhor visualizada através da Figura 9. Importante ressaltar que o acelerômetro mediu acelerações na mesma direção que o deslocamento da viga.

Tabela 1 – Especificações do Acelerômetro PCB333B30

Parâmetro	Valor
Sensibilidade ($\pm 10\%$)	100 mV/(m/s ²)
Faixa de medição	± 50 m/s ² pk
Faixa de frequência ($\pm 5\%$)	0.5 to 3000 Hz
Frequência de Ressonância	≥ 40 KHz
Banda Larga de Resolução	0.00015 m/s ² rms
Peso	0.14 oz (\cong 396.9 g)

Figura 9 – Fixação do acelerômetro na viga



Fonte: Autor

Também foi utilizada a placa de aquisição NI9234. Esta placa é um módulo de aquisição de sinais dinâmicos de quatro canais para medições de alta precisão de sensores IEPE. Apresenta conectividade USB, resolução de 24 bits e uma faixa dinâmica de mais de 100dB. Os quatro canais de entrada adquirem taxas de 2 a 51,2 kHz simultaneamente. Além disso, o hardware é equipado com condicionamento de sinais IEPE (2 mA de corrente constante) para acelerômetros. E, por fim, a faixa de tensão se encontra em $\pm 5V$. Na Tabela 2 segue um resumo das principais características da placa:

Tabela 2 – Características do módulo NI9234

Conexão	Faixa de resolução (bits)	Faixa de operação (dB)	Taxa de amostragem por canal (kHz)	Entradas analógicas	Faixa de entrada
USB	24	102	51,4	4	± 5

4.2 Interface desenvolvida em Python

Para desenvolver a programação e consequentemente a interface, foi utilizado o *PyCharm* que é um ambiente de desenvolvimento integrado usado em programação, especificamente para a linguagem de programação em Python. Sendo desenvolvido pela empresa JetBrains. A interface feita para gerar o espectro em frequência através das filmagens da viga pode ser visualizada pela Figura 10.

Figura 10 – Interface



Fonte: Autor

São oferecidos alguns campos que devem ser preenchidos devidamente pelo usuário para que o algoritmo consiga funcionar de forma correta. Primeiramente deve ser feito a seleção do local do arquivo, clicando em procurar teremos a opção de selecionar o vídeo que será utilizado para a análise. Depois deve ser colocada a frequência de amostragem, ou seja, deverá ser informado o FPS (*frames per second*) da câmera utilizada, como no caso deste trabalho foi utilizado um celular Motorola Moto G5S que filma em 30 fps, neste campo deverá ser colocado, então, o valor de 30. Posteriormente, para gerar o espectro de frequência com o eixo da amplitude em milímetros, é necessário saber a relação milímetro por pixel. Então para auxiliar o usuário, foi colocado a opção de selecionar vídeo, em que é possível abrir novamente o vídeo, a partir disso será mostrado na tela o primeiro frame do vídeo de forma estática, em que será possível clicar sobre a imagem e na tela será informado as coordenadas (X, Y) em pixels. Selecionando dois pontos distintos, pode ser extraído, por exemplo, o comprimento da viga em pixels. Este processo pode ser melhor entendido olhando a Figura 11.

Figura 11 – Obtenção do comprimento em pixel



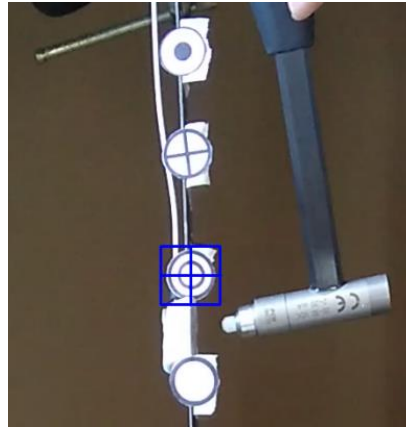
Fonte: Autor

Por fim, no último campo deve ser informado o comprimento em mm, apertando em “Calcular relação mm/pixel”, assim, será computado a relação e informado ao usuário. Em seguida basta apertar em “Analisar” para rodar o algoritmo.

Logo após apertar o botão de “Analisar”, será aberto o primeiro frame de vídeo de forma estática. Dessa forma é possível selecionar o alvo que se deseja fazer o rastreamento por

todo o vídeo. O alvo deverá ficar dentro do retângulo selecionado. A Figura 12 exemplifica este processo de seleção.

Figura 12 – Seleção do alvo

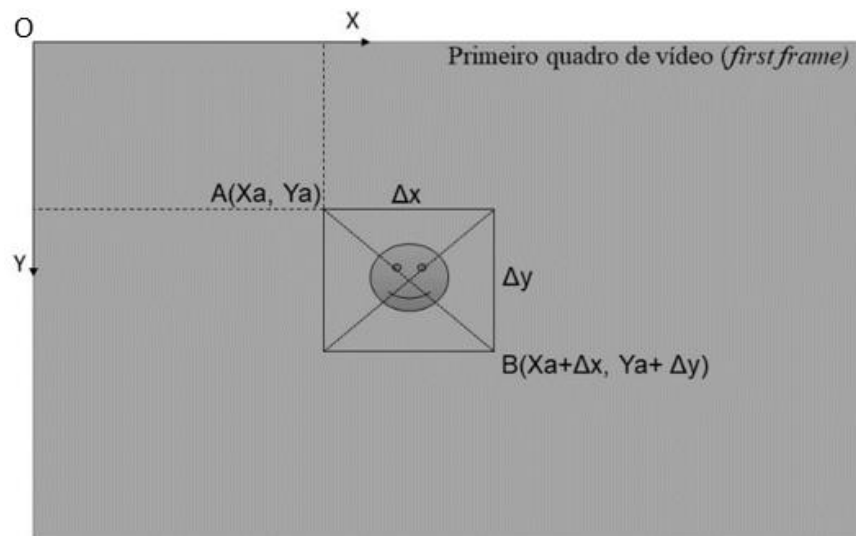


Fonte: Autor

4.3 Funcionamento do Algoritmo

Primeiramente, com a utilização da biblioteca OpenCV foi possível selecionar e abrir o vídeo. Posteriormente, o primeiro frame foi armazenado na variável “*first_frame*”. Este passo é importante pois será usado para a criação da caixa de seleção do objeto, possibilitando o rastreamento durante os próximos quadros. Agora a programação é feita de forma que a rotina consiga identificar as coordenadas do primeiro ponto A (momento do primeiro clique do usuário) e as coordenadas do segundo ponto B (momento em que o usuário solta o botão do *mouse*). Sabendo as coordenadas de A e B, o retângulo é formado e o que estiver dentro dele será o objeto que o rastreador irá buscar durante todo o vídeo através do algoritmo MIL. Aqui é importante lembrar que a origem do sistema de coordenadas é colocada no canto superior esquerdo, como pode ser visto na Figura 13.

Figura 13 – Caixa de seleção do objeto



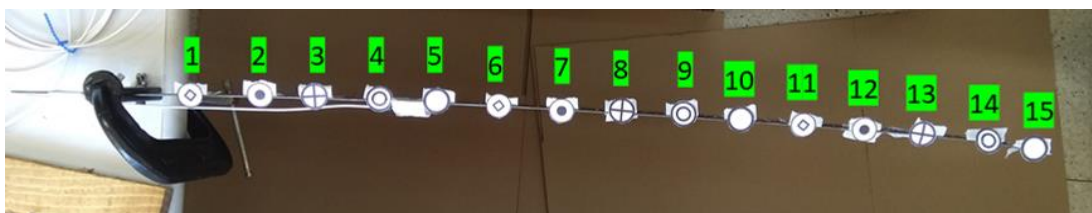
Fonte: Autor

O ponto “O” é a origem, sendo \overrightarrow{OX} e \overrightarrow{OY} o sentido positivo dos eixos coordenados. Observe que o ponto A possui coordenadas (X_a, Y_a) e o ponto B possui coordenadas que é função das coordenadas de A, acrescido de um deslocamento horizontal Δx e vertical Δy . Dessa maneira, a coordenada em B é descrita por $(X_a + \Delta x, Y_a + \Delta y)$.

5 RESULTADOS E DISCUSSÃO

Neste trabalho, foram realizados 5 testes. Em cada um deles, foi aplicada uma entrada impulsiva com um martelo de impacto à 200 mm do engaste. Para uma maior facilidade de entendimento dos resultados, as 15 etiquetas equi-espaçadas na viga foram enumeradas a partir do engaste até na extremidade oposta, como pode ser visto na Figura 14.

Figura 14 – Enumeração das Etiquetas

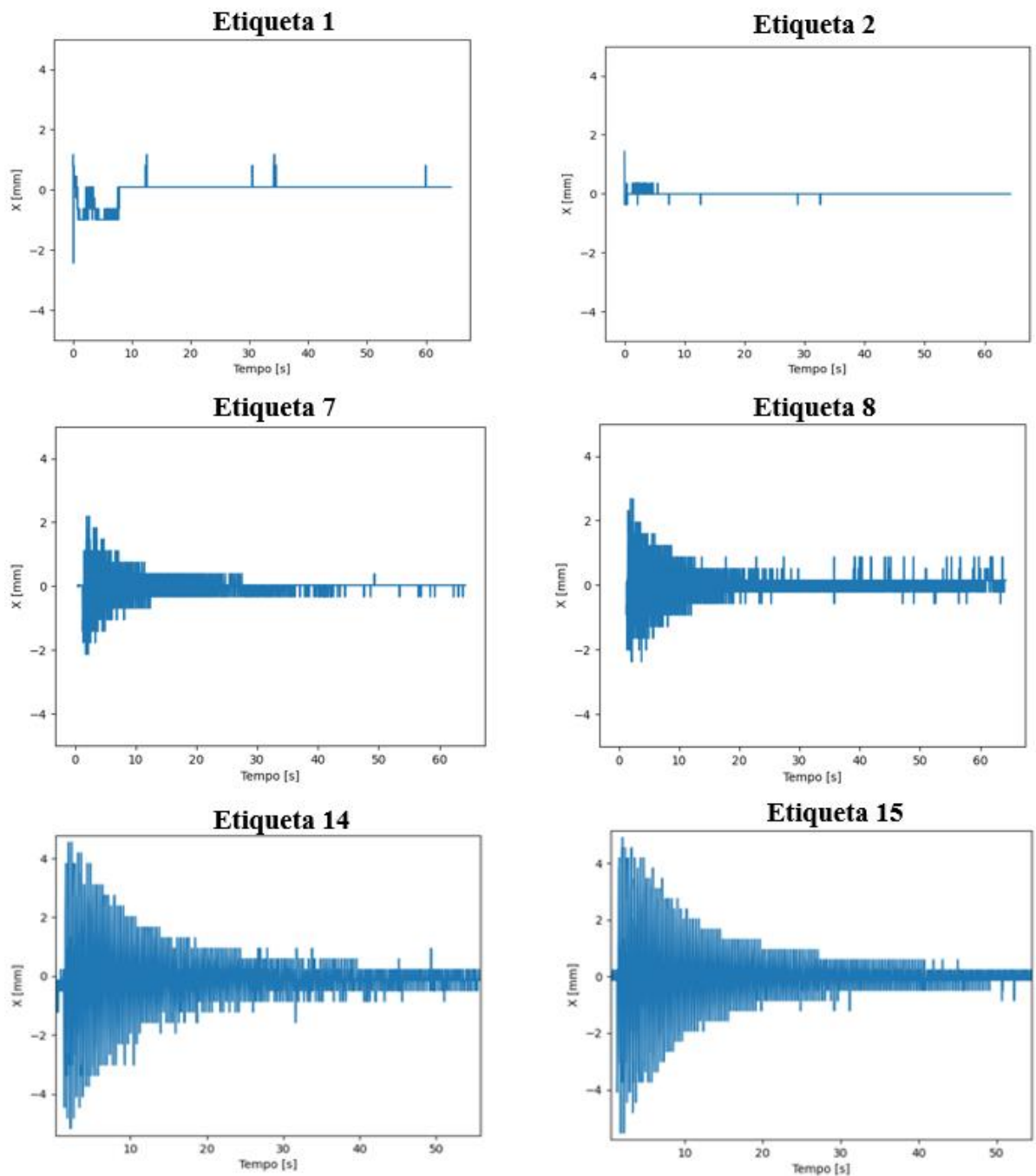


Fonte: Autor

Em um primeiro momento, será detalhado os resultados obtidos para o primeiro teste. Para isso, foram escolhidas 6 etiquetas de forma que se pudesse ter a uma boa representatividade dos resultados por toda a viga. Dessa forma, escolheu-se as duas primeiras etiquetas mais próximas do engaste, sendo estes dois casos os mais problemáticos devido a baixa amplitude de oscilação. Posteriormente, foram obtidos os resultados para as duas etiquetas centrais, neste caso foram escolhidas a sétima e a oitava etiqueta, que seria o caso “intermediário” em relação à dificuldade da câmera identificar os deslocamentos, por motivos de resolução. Por fim, as duas últimas próximas da extremidade livre da viga, que é o caso que teoricamente seria mais fácil de identificar os deslocamentos pela câmera foram analisadas, para este caso é esperado que o algoritmo consiga lidar melhor, já que temos deslocamentos maiores devido a uma maior distância do engaste, facilitando o rastreamento pelo algoritmo. Para o primeiro teste, será mostrado os gráficos no domínio do tempo e depois o gráfico obtido do espectro de frequência, após aplicação do algoritmo de FFT. Para os demais testes, serão apresentadas apenas as frequências com amplitudes predominantes em forma de tabela, já que os gráficos são bem similares de um teste para o outro.

O gráfico do deslocamento no tempo obtido para as 6 etiquetas pode ser visualizado através da Figura 15. Importante dizer que a relação entre milímetro e *pixels* encontrada foi de $0,36 \text{ mm/pixel}$. Importante dizer que este valor então, é a resolução da medição, ou seja, é o menor deslocamento possível que o algoritmo consegue detectar.

Figura 15 – Resposta no Domínio do Tempo

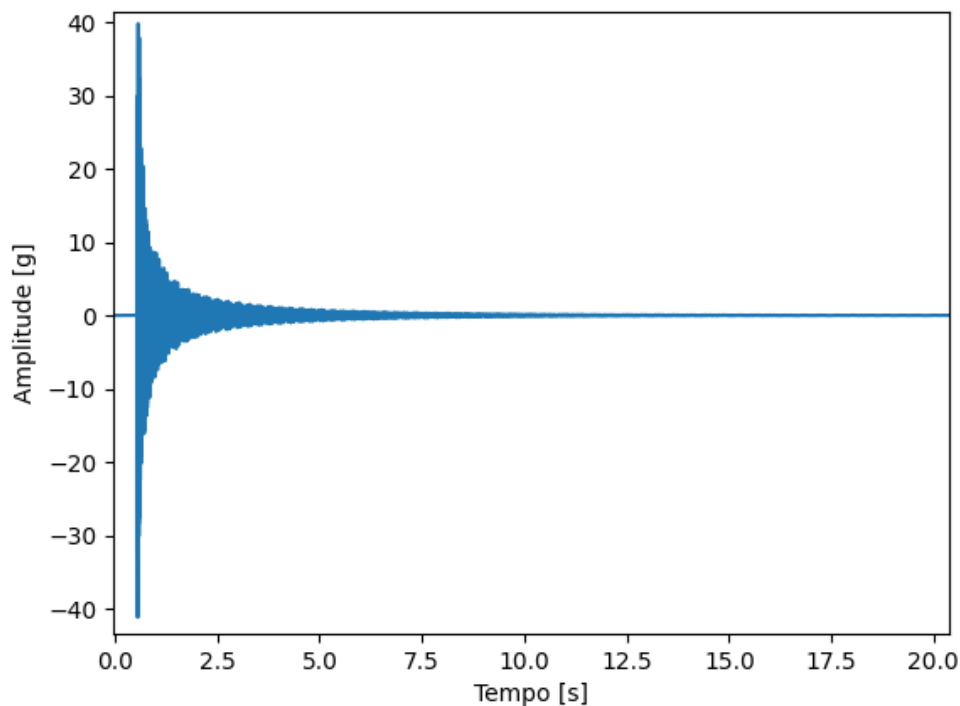


Fonte: Autor

Observa-se na Figura 15 que os resultados fazem sentido. Nas duas primeiras etiquetas, o deslocamento é muito pequeno em relação à câmera utilizada, não conseguindo identificar os deslocamentos. Este resultado é esperado visto que as amplitudes de oscilação perto do engaste são pequenas, não sendo identificadas de forma correta pelo algoritmo. Já as etiquetas que estão na parte do meio da viga, apresentam resultados melhores, nota-se que o gráfico toma uma forma de movimento oscilatório, cujas amplitudes vão diminuindo com o tempo. Por fim, observe que nas etiquetas mais afastadas (14 e 15), tem-se os melhores resultados, o que já era esperado visto que os deslocamentos maiores são esperados na extremidade livre da viga.

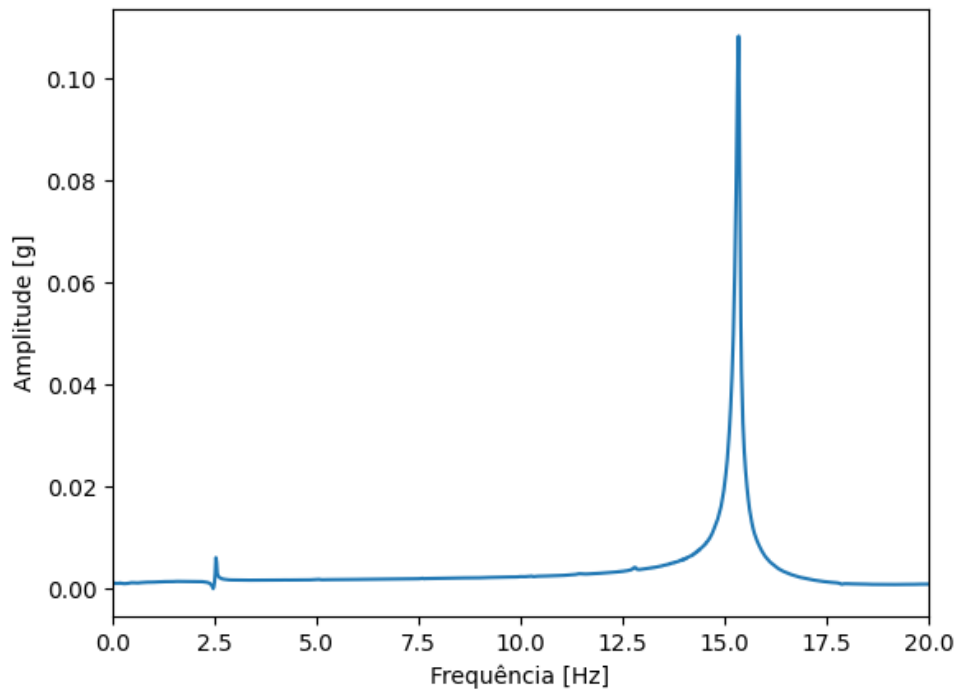
Em seguida, é possível visualizar através da Figura 16 o sinal temporal, e pela Figura 17 o gráfico do espectro em frequência obtido a partir dos dados coletados pelo acelerômetro. As frequências dominantes encontradas até 20 Hz vão ser confrontadas com as frequências encontradas a partir do sinal obtido pela técnica de processamento de vídeo. É importante dizer que a grandeza medida pelo acelerômetro é aceleração, mas espera-se que as frequências sejam coerentes com às medidas no deslocamento.

Figura 16 – Sinal temporal do acelerômetro



Fonte: Autor

Figura 17 – Espectro em frequência através do acelerômetro

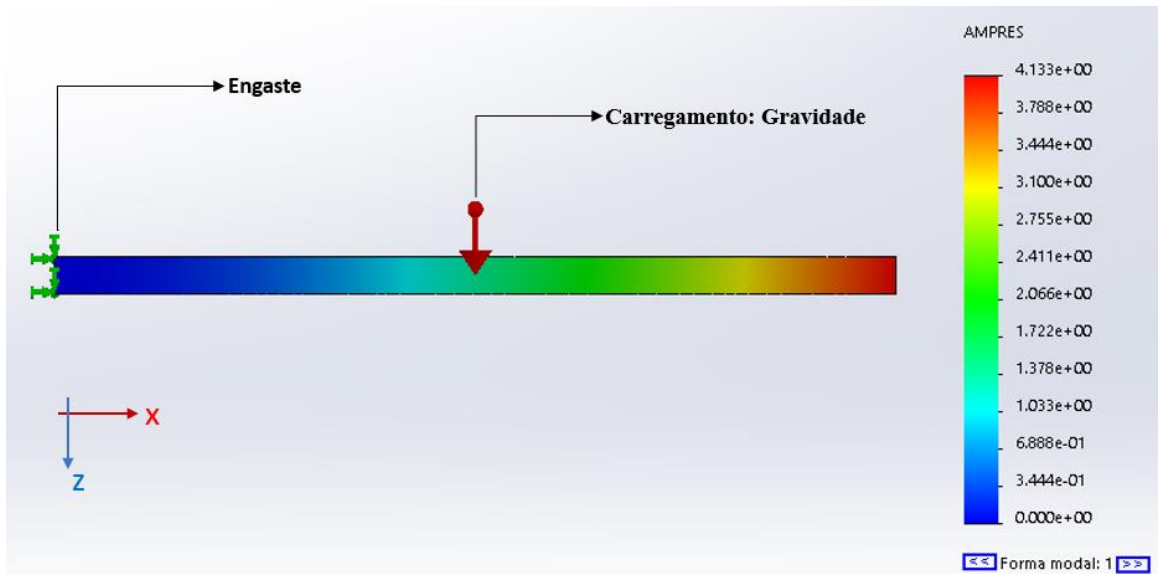


Fonte: Autor

A partir do gráfico da Figura 17, foi possível obter as frequências de 2,54 Hz e 15,32 Hz.

Para ter uma maior garantia dos resultados encontrados pelo acelerômetro, foi realizado uma simulação em elementos finitos da viga, utilizando análise modal para extrair as frequências da estrutura testada, foi utilizando o *software* SolidWorks. O modelo utilizado com algumas informações pode ser visto pela Figura 18.

Figura 18 – Representação visual do modelo criado



Fonte: Autor

As características detalhadas da malha foram geradas. Essas informações podem ser vistas através da Tabela 3.

Tabela 3 – Informações geradas da malha

Tipo de malha	Malha sólida
Gerador de malhas usado	Malha padrão
Pontos Jacobianos	4 pontos
Tamanho do elemento	3,67847 mm
Tolerância	0,183924 mm
Qualidade da malha	Alta
Total de Nós	19268
Total de elementos	9231

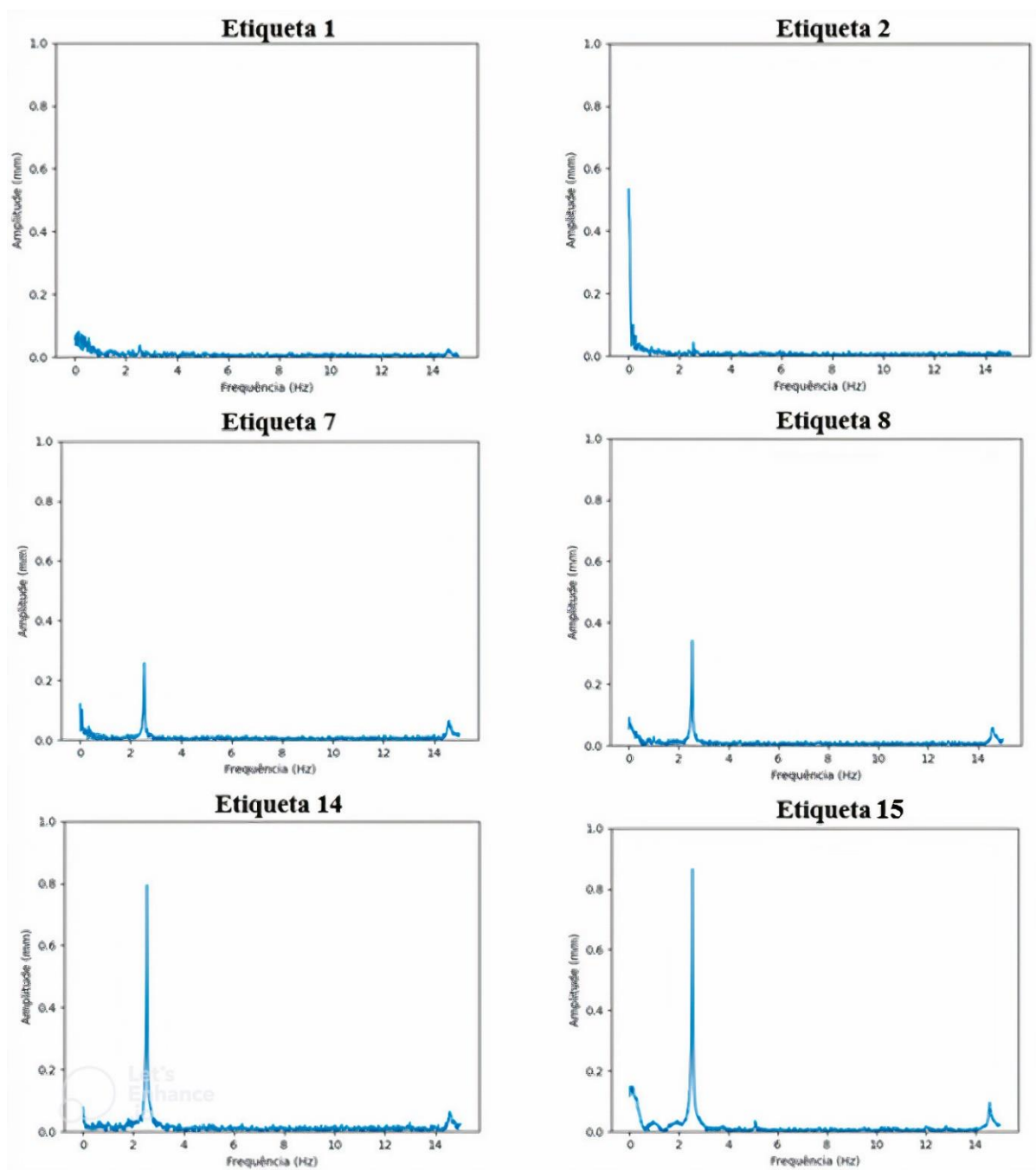
O *software* informa as frequências correspondentes aos cinco primeiros modos de vibração. O resultado numérico pode ser visto na Tabela 4

Tabela 4 – Frequência e modos de vibração, simulação

N° do Modo	Frequência [Hz]
1°	2,72
2°	17,07
3°	47,82
4°	53,55
5°	93,75

É possível notar que as frequências encontradas para o primeiro e segundo modos de vibrações são bem próximas com as obtidas pelo acelerômetro, tendo uma diferença percentual de 6,6% e 10,25% respectivamente. Essa diferença pode ser explicada por algumas possibilidades, como por exemplo engaste imperfeito, deformação na viga, dentre outros fatores. Dessa forma, as frequências de 2,54 Hz e 15,32 Hz, encontradas pelo acelerômetro, vão ser adotadas como referências para serem comparadas com os próximos resultados encontrados através da técnica de processamento de vídeo, independentemente da etiqueta analisada, pois são frequências naturais do sistema, que dependem apenas de sua massa e rigidez mas não da posição. A Figura 19 mostra o espectro em frequência para as 6 etiquetas monitoradas.

Figura 19 – Espectro em frequência das etiquetas



Fonte: Autor

Observe que apesar das frequências terem ficado próximas das obtidas pelo acelerômetro, as amplitudes ficaram invertidas, com amplitudes maiores por volta de 2,5 Hz e menores em torno de 15 Hz. Isso se deve ao fato de que a grandeza medida pelo acelerômetro é aceleração e não deslocamento. Quanto maior a frequência de vibração, maior será a aceleração medida pelo acelerômetro.

O resultado numérico pode ser visto pela Tabela 5. Considere “NI” como sendo um resultado não identificado.

Tabela 5 – Frequências predominantes para as 6 etiquetas encontradas no teste 1

Etiqueta	Frequência 1 [Hz]	Frequência 2 [Hz]
1 ^a	NI	NI
2 ^a	NI	NI
7 ^a	2,53	15,42
8 ^a	2,55	15,42
14 ^a	2,54	15,35
15 ^a	2,53	15,38

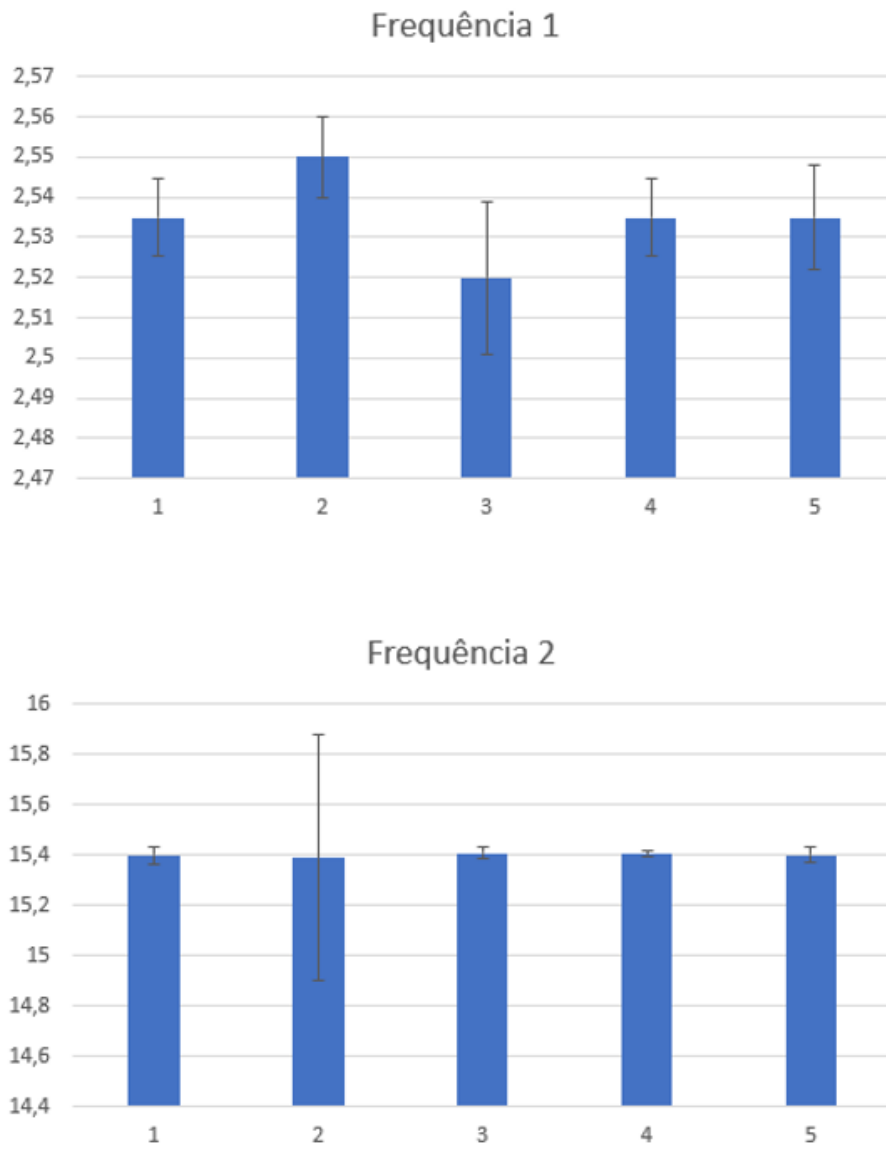
Analisando o resultado, observe que para as duas etiquetas mais próximas do engaste, não foi possível identificar quais as frequências predominantes no sinal, devido ao baixo deslocamento observado em regiões próximas ao engaste da viga. Para as demais, que possuem um deslocamento maior, as frequências obtidas ficaram bem próximas das obtidas pelo acelerômetro. Por fim, note também na Figura 19 que apareceram picos de amplitude perto de 0 Hz, isso pode ter se dado pelo fato de que a viga, no momento inicial de todos os testes, já se apresentar um pouco defletida, outro fator que pode ter contribuído é a própria programação do algoritmo, apresentando erros para frequências próximas de 0 Hz.

Por fim, para verificar repetibilidade dos resultados, as frequências naturais encontradas em outros 4 testes são apresentadas pela Tabela 6 e a Figura 20 mostra o gráfico de *Error Bar* para as duas frequências obtidas.

Tabela 6 – Frequências encontradas para os demais testes

2° TESTE		
Etiqueta	Frequência 1	Frequência 2
1 ^a	NI	NI
2 ^a	NI	NI
7 ^a	2,53	15,42
8 ^a	2,55	15,42
14 ^a	2,55	15,40
15 ^a	2,55	15,38
3° TESTE		
1 ^a	NI	NI
2 ^a	NI	NI
7 ^a	2,53	15,42
8 ^a	2,51	15,43
14 ^a	2,51	15,38
15 ^a	2,55	15,40
4° TESTE		
1 ^a	NI	NI
2 ^a	NI	NI
7 ^a	2,52	15,42
8 ^a	2,54	15,41
14 ^a	2,54	15,40
15 ^a	2,53	15,40
5° TESTE		
1 ^a	NI	NI
2 ^a	NI	NI
7 ^a	2,53	15,42
8 ^a	2,55	15,42
14 ^a	2,52	15,38
15 ^a	2,54	15,36

Figura 20 – *Error Bar* para as duas frequências nos 5 testes realizados.



Fonte: Autor

6 CONSIDERAÇÕES FINAIS

Este trabalho cumpriu as expectativas como um todo, e foi capaz de cumprir seu objetivo principal que reside em contribuir para a verificação de uma estratégia alternativa de cálculo de deslocamento, confrontando o sinal com um sensor de referência que neste caso é o acelerômetro. Conseqüentemente, possibilitando a medição de vibrações em diferentes etiquetas equi-distantes em uma viga engastada, sendo feita a comparação das frequências obtidas pelo método de contato, que seria a partir do acelerômetro e pelo método de não contato, utilizando apenas uma câmera de celular.

A partir dos dados coletados, foi possível observar em todos os testes realizados, que o acelerômetro identificou duas frequências em que temos um pico de amplitude, sendo a menor frequência sempre bem próxima de 2,5 Hz e a maior por volta de 15 Hz. Vale lembrar que foi limitada as frequências obtidas pelo acelerômetro numa faixa que não ultrapasse 20 Hz, uma vez que o objetivo era comparar as frequências obtidas com o sinal obtido através da câmera do celular que possui uma taxa de amostragem de 30 fps, portanto, vibrações com uma frequência maior que 15 Hz não são devidamente detectadas de acordo com o teorema de Nyquist. Dessa forma, para frequências próximas ou menores que 15 Hz o método por processamento de vídeo se mostrou bem eficaz, já que também conseguiu detectar as mesmas duas frequências encontradas no acelerômetro. Entretanto, foi possível constatar que quanto mais próxima a etiqueta ficava da parte engastada da viga, mais o sinal apresentava valores que não condizem com a realidade do movimento. Na primeira e segunda etiqueta do primeiro teste foi possível visualizar bem isso, nestes dois casos não foi possível a identificação de nenhuma frequência.

Portanto, é possível dizer que o método por imagem, apesar de apresentar bons resultados, possui limitações. A primeira e segunda etiqueta, por estarem mais próximas do engaste, possui a menor amplitude de vibração, dificultando a tarefa do algoritmo de identificar diferenças de posicionamento do marcador durante os quadros seguintes, fazendo com que o resultado não seja o esperado. Este problema talvez poderia ser solucionado utilizando uma câmera com uma definição melhor e com uma taxa de amostragem maior. Entretanto, o método se concentra em ser de baixo custo, indo contra a proposta inicial do trabalho. Pois câmeras de alta definição e taxa de atualização não são usualmente utilizadas e possuem um alto preço.

REFERÊNCIAS

ALMEIDA, M. T.; ALMEIDA, F. R. D. V. **Princípios de diagnósticos de defeitos em máquinas rotativas**. 27. Ed. Itajubá: IVMTA, 2014.

AQTECH ENGENHARIA E INSTRUMENTAÇÃO. **Treinamento em aquisição de dados**. AQTech. Florianópolis, p. 80. 2016.

BENEDETTI, Julinho Alberto. **Manutenção centrada em confiabilidade e análise de vibração**. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, 2002.

BARBOSA, Orlando. “**Estudo da dinâmica inversa em manipuladores robóticos a partir de medição de movimento usando processamento de imagens digitais**”. Relatório de Projeto de Iniciação Científica – ICT, Universidade Federal de São Carlos, 2019.

Babenko, B. (2008). **Multiple instance learning: Algorithms and applications. Technical report**, University of California San Diego, San Diego.

Carnero, M. C. (2003). “**Selection of diagnostic techniques and instrumentation in a predictive maintenance program. A case study.**” *Decision Support Systems* (38):539-555.

CARVALHO, André et al. **Inteligência Artificial: Uma abordagem de Aprendizado de Máquina**. Rio de Janeiro: Ltc, 2011.

DEFATTA, D. J.; et al. **Digital Signal Processing: A System Design Approach**, John Wiley and Sons. 1998.

Dietterich, T. G.; Lathrop, R. H. e Lozano-Perez, T. (1997). **Solving the multiple instance problem with axis-parallel rectangles**. *Artificial Intelligence*, Vol. 89(1-2):p. 31-71

FARIA, Alexandre. “**Uma nova abordagem para aprendizado de múltiplas instâncias, baseada em seleção de instâncias via estimador de densidade por kernel**”. Tese de Doutorado em Engenharia Elétrica, Universidade Federal de Minas Gerais, 2016

JOSEP, K.; RAJ, A. N. J.; FAN, Z.; VIDHYAPATHI, C. M. A time eficiente vídeos tabilization algorithm based on Block Matching in a restricted Search space. **Real time Computing and Robotics (RCAR)**, 2017 IEEE International Conference on, V. 2017. Available in: <http://ieeexplore.ieee.org/abstract/document/8311>.

KAMEN, E. W.; HECK, B. S. **Fundamentals of Signals and Systems, Using the Web and Matlab**, second edn, Prentice Hall, 2000.

KONAR, Amit. **Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain**. [S. 1.]: Crc Press, 1999.

LATHI, Bhagawandas P. **Signal processing and linear systems**. Oxford University Press, 2000

LIU, T.; BURNER, A. W.; JONES, T. W.; BARROWS, D. A. Photogrammetric techniques for Aerospace applications. **Progress in Aerospace Sciences**, v. 54, p. 1-58, October 2012.

MOHANTY, A. R. **Machinery Condition Monitoring: Principles and Practices**. Kharagpur: CRC Press, 2015.

ENDO, Marcos. “**Medição de vibrações Estruturais e de Sistemas Rotativos Utilizando Imagens Fotográficas Subamostradas**”. Dissertação de Mestrado em Engenharia Mecânica, Escola de Engenharia de São Carlos, 2013.

OPPENHEIM, A. V. **Processamento em Tempo Discreto de Sinais**. 3ª Edição. São Paulo: Pearson Education, 2012

OLASZEK, P. Investigation of the dynamic characteristic of bridge structures using a computer vision method. **Measurement**, v. 25, n. 3, p. 227-236, April 1999.

RYALL, T. G.; FRASER, C. S. Determination of structural modes of vibration using digital photogrammetry. **Journal of Aircraft**, v. 39, n. 1, p. 114-119, January-February 2002.

SANTOS, Derick. **Processamento de sinais para monitoramento de vibrações em unidades geradoras hidrelétricas**, 2013. Disponível em <http://www.biblioteca digital.ufmg.br>. Último acesso em: 11 de Abril de 2022.

APÊNDICE A – ALGORITMO UTILIZADO NO PROCESSAMENTO DOS VÍDEOS

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import time
from tkinter import *
from tkinter import filedialog

#FUNÇÕES

def relacao():
    global rel
    global relacao
    milimetro = float(mm.get())
    pixels = float(pixel.get())
    rel = milimetro/pixels
    Label(app, text = str(rel)).place(x=230, y=210, width=60, height=20)

def comparar():
    def click_event(event, x, y, flagas, params):
        if event == cv2.EVENT_LBUTTONDOWN:
            print(x, ', ',y)
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(first_frame,str(x) + ', ' + str(y), (x,y), font, 1, (255, 0, 0), 2)
            cv2.imshow('Primeiro frame', first_frame)
    dirpath = filedialog.askopenfilenames()
    cap = cv2.VideoCapture(dirpath[0])
    ret, first_frame = cap.read()
    cv2.imshow('Primeiro frame', first_frame)
```



```

cv2.setMouseCallback('Primeiro frame', click_event)

cv2.waitKey(0)

cv2.destroyAllWindows()

def sair():
    app.destroy()
    exit()

def procurar():
    global dirpath
    dirpath = filedialog.askopenfilenames()
    global arq
    arq = str(dirpath[0])
    Label(app, text=str(arq)).place(x=130, y=30, width=460, height=20)

def analisar():
    #VARIAVEIS USADAS
    C = [] #Matriz que guarda o centro de coordenadas do retangulo delimitado a cada frame [
(xc1,yc1) , (xc2,yc2) , .... ]
    cordC = [] #Matriz de apoio pra ir adicionando as cordenadas na matriz C
    cont_loss = 0
    prev_frame_time = 0
    new_frame_time = 0
    tempo = []
    fs = int(vfreq.get()) #Freq.Amostragem
    "relacao = float(rel.get())"

    #ABERTURA DO VIDEO E CONTAGEM DE FRAMES TOTAIS
    cap = cv2.VideoCapture(dirpath[0])
    nframes = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

```

#SELEÇÃO DO OBJETO À SER RASTREADO UTILIZANDO O PRIMEIRO FRAME DO VÍDEO.

```
ret, first_frame = cap.read()
largura = first_frame.shape[1] #1080
altura = first_frame.shape[0] #1920
proporcao = float(largura/altura)
altura_nova = 1000
largura_nova = int(altura_nova*proporcao)
first_frame = cv2.resize(first_frame,(largura_nova,altura_nova))
box = cv2.selectROI('Selecione o Objeto', first_frame, fromCenter=False) #Retorna
(x0,y0,^x (largura do retangulo), ^y (altura do retangulo))
tracker = cv2.TrackerMIL.create()
tracker.init(first_frame, box) #(Cordenadas na imagem modelo, retangulo que será
procurado no video)
cv2.destroyAllWindows()
start = time.time()
```

#RASTREAMENTO A CADA FRAME, EXTRANDO AS COORDENADAS E ARMAZENANDO NA LISTA C

```
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame = cv2.resize(frame, (largura_nova, altura_nova))
    ok, box = tracker.update(frame)
    if ok:
        #gray = frame
        #frame = cv2.resize(frame,(largura_nova,altura_nova))
        font = cv2.FONT_HERSHEY_SIMPLEX
        new_frame_time = time.time()
        fps = 1/(new_frame_time-prev_frame_time)
        prev_frame_time = new_frame_time
```

```

fps = int(fps)
fps = str(fps)
pt1 = (box[0], box[1])
pt2 = ((box[0]+ box[2]), (box[1] + box[3]))
cv2.rectangle(frame, pt1, pt2, (255,0,0), 2, 1)
xc = box[0]+(box[2]/2)
cordC.append(xc)
yc = box[1]+(box[3]/2)
tempo.append(time.time() - start)
cordC.append(yc)
C.append(cordC[:])
cordC.clear()

if cv2.waitKey(25) & 0xFF == ord('q'):
    break

if not ok:
    cont_loss += 1

    cv2.putText(frame, 'FPS: ' + str(int(fps)), (100, 50), cv2.FONT_HERSHEY_SIMPLEX,
.75, (200, 50, 50), 2)

    cv2.putText(frame, 'Tempo: ' + str(int(time.time()-start)), (100, 80),
cv2.FONT_HERSHEY_SIMPLEX, .75, (200, 50, 50), 2)

    cv2.imshow('Tracking', frame)

cv2.destroyAllWindows()

print(f'A rotina perdeu {cont_loss} de um total de {nframes}, {(cont_loss/nframes)*100}%
de perca')

#TRATAMENTO DE DADOS
X = list() #vetor_X armazena apenas as coordenadas X do objeto
Y = list() #vetor_Y armazena apenas as coordenadas Y do objeto
for i in range(0,len(C)):
    X.append(C[i][0])
    Y.append(C[i][1])
arr_X = rel*np.array(X)

```

```

arr_Y = rel*np.array(Y)

#fft = Fast Fourier Transformer
arr_X = arr_X - np.mean(arr_X)
freq = np.fft.fftfreq(np.size(X), d=1/fs) #Criar o eixo da frequencia
freq_pos = freq > 0
fft_calculo = np.fft.fft(arr_X) #FFT do sinal que quero calcular
fft_abs = 2*np.abs(fft_calculo/np.size(X))

```

```

plt.figure()
plt.xlabel('Frequência (Hz)')
plt.ylabel('Amplitude (mm)')
plt.plot(freq[freq_pos], fft_abs[freq_pos])
plt.show()

```

```

plt.figure()
tempo = np.linspace(0,(len(X)-1)/30,len(X))
plt.xlabel('Tempo [s]')
plt.ylabel('X [mm]')
plt.plot(tempo, arr_X)
plt.show()

```

```

#INTERFACE
global app
app = Tk()
app.title('Análise de Vibração')
app.geometry('600x300')

```

```
app.configure(background='#dde')
```

```
#anchor N=Norte, S=Sul, E=Leste, W=Oeste, posição dentro do quadrado formado pelo  
place
```

```
Label(app, text='Local do Arquivo', background='#dde', foreground='#009',  
anchor=W).place(x=10, y=10, width=100, height=20)
```

```
Label(app, text='Frequencia de Amostragem Fs [Hz] ', background='#dde',  
foreground='#009', anchor=W).place(x=10, y=60, width=200, height=20)
```

```
vfreq = Entry(app)
```

```
vfreq.place(x=10, y=80, width=200, height=20)
```

```
Label(app, text='Comprimento em Pixel', background='#dde', foreground='#009',  
anchor=W).place(x=10, y=110, width=200, height=20)
```

```
pixel = Entry(app)
```

```
pixel.place(x=10, y=130, width=200, height=20)
```

```
Label(app, text='Comprimento em mm', background='#dde', foreground='#009',  
anchor=W).place(x=10, y=160, width=200, height=20)
```

```
mm = Entry(app)
```

```
mm.place(x=10, y=180, width=200, height=20)
```

```
Button(app, text='Procurar', command=procurar).place(x=10, y=30, width=100, height=20)
```

```
Button(app, text='Analisar', command=analisar).place(x=10, y=270, width=100, height=20)
```

```
Button(app, text='Sair', command=sair).place(x=120, y=270, width=100, height=20)
```

```
Button(app, text='Selecionar Vídeo', command=comparar).place(x=215, y=130, width=100,  
height=20)
```

```
Button(app, text='Calcular relação mm/pixel', command=relacao).place(x=10, y=210,  
width=200, height=20)
```

```
app.mainloop()
```

APÊNDICE B – FFT APLICADO AO ACELERÔMETRO

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

tabela = pd.read_csv('teste04.txt', sep = '\t', header = 0).to_numpy()

x = tabela[:, 0]

y = tabela[:, 2]

arr_X = y

arr_X = y - np.mean(y)

freq = np.fft.fftfreq(np.size(y), d=1/3200) #Criar o eixo da frequencia

freq_pos = freq > 0

fft_calculo = np.fft.fft(arr_X) #FFT do sinal que quero calcular

fft_abs = 2*np.abs(fft_calculo/np.size(y))

plt.xlabel('Frequency')

plt.xlim(0,20)

plt.ylabel('Amplitude (m/s2)')

plt.title('Frequency Domain (Hz)')

plt.plot(freq[freq_pos], fft_abs[freq_pos])

plt.show()
```