

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

Marcos Vinícius Quinteiro Silveira

**SIMULAÇÃO NUMÉRICA DE MOTORES ELÉTRICOS E
CONSTRUÇÃO DE BANCO DE DADOS PARA
PLATAFORMA DE PROGNÓSTICO DE FALHAS**

Santa Maria, RS
2022

Marcos Vinícius Quinteiro Silveira

**SIMULAÇÃO NUMÉRICA DE MOTORES ELÉTRICOS E CONSTRUÇÃO DE
BANCO DE DADOS PARA PLATAFORMA DE PROGNÓSTICO DE FALHAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do título de **Bacharel em Engenharia Mecânica**

Orientador: Prof. Dr. Carlos Eduardo de Souza

Santa Maria, RS
2022

Quinteiro Silveira, Marcos Vinícius

Simulação Numérica de Motores Elétricos e Construção de Banco de Dados para Plataforma de Prognóstico de Falhas / por Marcos Vinícius Quinteiro Silveira. – 2022.

65 f.: il.; 30 cm.

Orientador: Carlos Eduardo de Souza

Trabalho de Conclusão de Curso - Universidade Federal de Santa Maria, Centro de Tecnologia, Curso de Graduação em Engenharia Mecânica, RS, 2022.

1. Simulação. 2. Motores Elétricos. 3. Banco de Dados. 4. Aprendizado de Máquinas. 5. Prognóstico de Falhas. I. de Souza, Carlos Eduardo. II. Simulação Numérica de Motores Elétricos e Construção de Banco de Dados para Plataforma de Prognóstico de Falhas.

© 2022

Todos os direitos autorais reservados a Marcos Vinícius Quinteiro Silveira. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: marcos.v.q.silveira@gmail.com

Marcos Vinícius Quinteiro Silveira

**SIMULAÇÃO NUMÉRICA DE MOTORES ELÉTRICOS E CONSTRUÇÃO DE
BANCO DE DADOS PARA PLATAFORMA DE PROGNÓSTICO DE FALHAS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia Mecânica
da Universidade Federal de Santa Maria
(UFSM, RS), como requisito parcial para a obtenção
do título de **Bacharel em Engenharia
Mecânica**

Aprovado em 1 de agosto de 2022:

Carlos Eduardo de Souza, Dr. (UFSM)
(Presidente/Orientador)

Marcos Daniel de Freitas Awruch, Dr. (UFSM)

Cesar Gabriel dos Santos, Dr. (UFSM)

Santa Maria, RS
2022

RESUMO

SIMULAÇÃO NUMÉRICA DE MOTORES ELÉTRICOS E CONSTRUÇÃO DE BANCO DE DADOS PARA PLATAFORMA DE PROGNÓSTICO DE FALHAS

AUTOR: Marcos Vinícius Quinteiro Silveira

ORIENTADOR: Carlos Eduardo de Souza

Em praticamente todas as máquinas utilizadas dentro de uma fábrica existe um motor elétrico, responsável pela movimentação de componentes. Assim como demais mecanismos, os motores estão sujeitos a falhas em operação. Com a extensa utilização destes equipamentos para suprir as necessidades atuais de manufatura, é imprescindível que a realização de uma parada de manutenção, mesmo quando programada e rotineira, é um evento atípico à normalidade do dia a dia de uma grande empresa ou indústria e, com isso, gera perdas econômicas. Com a ascensão da indústria 4.0, a utilização de métodos computacionais para prever e impedir paradas inesperadas está cada vez mais abundante. Métodos de aprendizado de máquinas estão sendo desenvolvidos a todo momento para suprir a necessidade de sistemas que preveem falhas de equipamentos. Um dos principais passos para o desenvolvimento de tais técnicas é o aprendizado propriamente dito. Estes aprendizados dependem de um conjunto de dados de treinamento, que devem ser o mais eficaz possível para ser desenvolvido um modelo preditivo eficiente e confiável. O objetivo deste trabalho é desenvolver um modelo computacional para simular e extrair dados de um motor elétrico sob diferentes condições de funcionamento, para assim, se estudar quais parâmetros extraídos desse são os mais aptos para o desenvolvimento de um banco de dados eficaz. Para atingir este objetivo, foi feito um estudo de modelos numéricos de motores elétricos com cinco graus de liberdade, bem como um estudo da estatística de dados para se ter um melhor entendimento quantitativo dos dados extraídos do sistema computacional desenvolvido. Foram realizadas simulações onde o motor foi colocado sob diferentes condições de funcionamento variando características estruturais e extraídos diferentes tipos de dados e avaliado tais dados de forma quantitativa. Para este experimento, a metodologia utilizada abrangeu a linguagem de programação de utilização aberta Python para a aplicação dos modelos numéricos, além de validação através de dados bibliográficos para o modelo proposto. Nele, o resultado de um dos graus de liberdade desenvolvido no modelo proposto não observou a sensibilidade ao parâmetro estrutural, sendo que dois parâmetros se mostraram mais sensíveis do que os outros dois, mostrando-se o método como eficaz no estudo e desenvolvimento de banco de dados, porém sem utilização em casos reais, devido às diversas suposições realizadas no estudo.

Palavras-chave: Simulação. Motores Elétricos. Banco de Dados. Aprendizado de Máquinas. Prognóstico de Falhas.

ABSTRACT

NUMERICAL SIMULATION OF ELECTRIC MOTORS AND DATABASE CONSTRUCTION FOR FAULT PROGNOSIS PLATFORM

AUTHOR: Marcos Vinícius Quinteiro Silveira

ADVISOR: Carlos Eduardo de Souza

In practically all machines used inside a factory there is an electric motor, responsible for moving components. As with other mechanisms, engines are subject to operating failures. With the extensive use of these equipments to meet current manufacturing needs, it is essential that a maintenance stop, even when scheduled and routine, is an event that is atypical of the day-to-day routine of a large company or industry and, as a result, generates economic losses. With the rise of Industry 4.0, the use of computational methods to predict and prevent unexpected stops is becoming more and more abundant. Machine learning methods are being developed regularly to meet the needs for systems that predict equipment failures. One of the main steps for the development of such techniques is the learning itself. These learnings depend on a training dataset, that must be as effective as possible to develop efficient and reliable predictive systems. The objective of this work is to develop a computational model to simulate and extract data from an electric motor under different operating conditions, in order to study which parameters extracted from this are the most suitable for the development of an effective database. To achieve this objective, a study of numerical models of electric motors with five degrees of freedom was carried out, as well as a study of data statistics to have a better quantitative understanding of the data extracted from the developed computational system. Simulations were carried out where the engine was placed under different operating conditions, varied structural characteristics and different types of data were extracted, and such data evaluated in a quantitative way. For this experiment, the methodology used covered the open use programming language Python for the application of numerical models, in addition to validation through bibliographic data for the proposed model. In it, the result of one of the degrees of freedom developed in the proposed model did not observe the sensitivity to the structural parameter, two parameters were more sensitive than the other two, showing the method to be effective in the study and development of a database, but not used in real cases, due to the different assumptions made in the study.

Keywords: Simulation. Eletrical Motors. Database. Machine Learning. Fault Prognosis.

LISTA DE FIGURAS

| | | |
|----|--|----|
| 1 | Aspectos construtivos de motores elétricos. | 18 |
| 2 | Componentes de mancais de rolamento. | 18 |
| 3 | Diagrama de corpo livre do motor de indução trifásico. | 19 |
| 4 | Diagrama de corpo livre do rotor com rotor e eixo rígido e mancais flexíveis. | 20 |
| 5 | Diagrama do rotor para diferentes casos. | 20 |
| 6 | Classificações de sinais. | 23 |
| 7 | Sinal periódico seno. | 23 |
| 8 | Correlação entre sinal no domínio do tempo e da frequência. | 24 |
| 9 | Transformada de Fourier de um sinal periódico com período (T_p) de 0,5. | 25 |
| 10 | Transformada de Fourier de um sinal quase periódico. | 26 |
| 11 | Distribuição dos valores de obliquidade. | 28 |
| 12 | Distribuição dos valores de curtose. | 29 |
| 13 | Fluxograma do desenvolvimento de um sistema preditivo em AM. | 30 |
| 14 | Fluxograma do procedimento realizado para o desenvolvimento de banco de dados. | 32 |
| 15 | Amplitude de resposta do sistema em vibração, resultado analítico obtido do livro FRISWELL et al. (2010) em preto, resultado obtido na simulação computacional em vermelho. | 37 |
| 16 | Valores de desalinhamento de eixo para o estudo de caso. | 39 |
| 17 | Resultados da simulação com todos os dados em 100% do seu valor. | 40 |
| 18 | Resultado da frequência da simulação com todos os dados em 100% do seu valor. ... | 41 |
| 19 | Resultado da frequência da simulação com todos os dados em 100% do seu valor e sem variação do torque. | 42 |
| 20 | Variáveis que não tiveram variação significativa. | 43 |
| 21 | Varição da média, intervalo e desvio padrão de u , v , θ e ψ | 44 |
| 22 | Varição da obliquidade e curtose de u , v , θ e ψ | 45 |
| 23 | Varição do valor máximo, média e desvio padrão das amplitudes das frequências de u , v , θ e ψ | 46 |
| 24 | Varição da obliquidade e curtose das amplitudes das frequências de u , v , θ e ψ | 48 |
| 25 | Cascata das pastas criadas pelo programa desenvolvido. | 49 |

LISTA DE TABELAS

| | | |
|---|---|----|
| 1 | Dados obtidos da simulação. | 35 |
| 2 | Valores da simulação de validação..... | 36 |
| 3 | Valores de entrada da simulação..... | 39 |
| 4 | Valores iniciais de obliquidade e curtose. | 45 |
| 5 | Valores iniciais de obliquidade e curtose das amplitudes das frequências..... | 48 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------------|--|
| AM | Aprendizado de Máquinas |
| TF | Transformada de Fourier |
| TDF | Transformada Discreta de Fourier |
| EDO | Equação Diferencial Ordinária |
| FFT | <i>Fast Fourier Transform</i> |
| CSV | <i>Comma-Separated-Values</i> |
| GD | Gêmeo Digital |
| T | Energia Cinética |
| U | Energia Potencial |
| m | Massa |
| k | Constante Elástica |
| x | Variável Dependente do Sistema |
| t | Tempo |
| F | Força |
| M | Massa em sua forma matricial |
| q | Graus de liberdade do sistema em sua forma vetorial |
| C | Coefficiente de amortecimento em sua forma matricial |
| K | Constante elástica em sua forma matricial |
| f | Forças externas em sua forma vetorial |
| A | Primeira constante do método Runge-Kutta |
| B | Segunda constante do método Runge-Kutta |
| x | Vetor de estados |
| J | Momento de inércia da carga e do rotor |
| B_n | Coefficiente de atrito |
| T_e | Torque eletromagnético |
| T_m | Torque mecânico |
| θ_p | Posição angular do eixo |
| ω_r | Velocidade angular do eixo |
| I_d | Momento de inércia do eixo central |
| I_p | Momento de inércia polar do rotor |
| η | Raio de desbalanceamento de massa |
| β | Ângulo de deslocamento do eixo |

| | |
|-----------|--|
| u | Grau de liberdade de translação na direção x |
| v | Grau de liberdade de translação na direção y |
| θ | Grau de liberdade de rotação na direção x |
| ψ | Grau de liberdade de rotação na direção y |
| k_1 | Constante elástica do mancal a esquerda |
| k_2 | Constante elástica do mancal a direita |
| c_1 | Coefficiente de amortecimento do mancal a esquerda |
| c_2 | Coefficiente de amortecimento do mancal a direita |
| k_T | Constante elástica de translação |
| k_C | Constante elástica de acoplamento |
| k_R | Constante elástica de rotação |
| c_T | Coefficiente de amortecimento de translação |
| c_C | Coefficiente de amortecimento de acoplamento |
| c_R | Coefficiente de amortecimento de rotação |
| T_p | Período |
| \bar{x} | Média |
| s | Desvio Padrão |
| dt | Passo de tempo |
| A_{mp} | Amplitude |
| f | Frequência |

SUMÁRIO

| | | |
|--------------|---|----|
| 1 | INTRODUÇÃO | 12 |
| 1.1 | OBJETIVOS | 14 |
| 2 | REVISÃO TEÓRICA | 15 |
| 2.1 | VIBRAÇÕES MECÂNICAS | 15 |
| 2.1.1 | Equação do Movimento para Sistemas de um Grau de Liberdade | 15 |
| 2.1.2 | Múltiplos Graus de Liberdade | 16 |
| 2.2 | MOTORES ELÉTRICOS | 17 |
| 2.2.1 | Modelagem do Motor | 19 |
| 2.2.2 | Sistema Completo com 5 Graus de Liberdade | 21 |
| 2.3 | PROCESSAMENTO DE SINAL | 22 |
| 2.3.1 | Classificação de Sinal | 22 |
| 2.3.2 | Transformada de Fourier | 23 |
| 2.3.3 | Transformadas de Fourier e Classificação de Sinal | 25 |
| 2.3.4 | Transformada Rápida de Fourier | 26 |
| 2.3.5 | Análise Estatística | 26 |
| 2.3.5.1 | Obliquidade | 27 |
| 2.3.5.2 | Curtose | 28 |
| 2.4 | APRENDIZADO DE MÁQUINAS | 29 |
| 2.4.1 | Data Lake | 30 |
| 2.4.2 | Banco de Dados de Treinamento | 30 |
| 2.4.3 | Algoritmo de AM | 30 |
| 2.4.4 | Modelo Preditivo | 31 |
| 3 | METODOLOGIA PROPOSTA | 32 |
| 3.1 | DADOS DE ENTRADA | 33 |
| 3.2 | CÓDIGO DE AGRUPAMENTO | 33 |
| 3.3 | CÓDIGO DA SIMULAÇÃO ESTRUTURAL | 34 |
| 3.3.1 | Processamento Estatístico | 34 |
| 3.3.2 | Integração das EDO (<i>odeint</i>) | 34 |
| 3.4 | CÓDIGO PARA SALVAR GRÁFICOS | 35 |
| 4 | RESULTADOS NUMÉRICOS | 36 |
| 4.1 | VALIDAÇÃO DO MODELO | 36 |
| 4.2 | ESTUDO DE CASO | 37 |
| 4.2.1 | Potência, Torque Eletromagnético e Velocidade Dinâmica | 38 |
| 4.2.2 | Propriedade Variável no Torque Mecânico | 38 |
| 4.2.3 | Variáveis de Desbalanceamento e Desalinhamento | 39 |
| 4.2.4 | Simulação com os dados Nominais | 39 |
| 4.2.5 | Variáveis sem influência da variação | 42 |
| 4.2.6 | Análise dos Resultados Estatísticos | 43 |
| 4.2.6.1 | Média, Intervalo e Desvio Padrão | 43 |
| 4.2.6.2 | Obliquidade e Curtose | 45 |
| 4.2.7 | Análise dos Resultados Estatísticos das FFTs | 46 |
| 4.2.7.1 | Valor Máximo, Média e Desvio Padrão | 46 |
| 4.2.7.2 | Obliquidade e Curtose | 47 |
| 4.3 | BANCO DE DADOS | 49 |

| | | |
|----------|------------------------------------|-----------|
| 5 | CONCLUSÕES..... | 51 |
| 5.1 | SUGESTÃO DE TRABALHOS FUTUROS..... | 51 |
| | REFERÊNCIAS | 53 |
| | APÊNDICES..... | 56 |

1 INTRODUÇÃO

Dentre as máquinas mais utilizadas na indústria estão o torno, fresadora, prensa e furadeira (CRAVO, 2019), e todas estas máquinas possuem motores elétricos em sua composição. Visto que hoje em dia é esperado de empresas sempre manter uma cota de manufatura, é esperado que estas máquinas trabalhem sem parar por horas afim. Com a utilização extensa de equipamentos, esses estão condenados a diminuição de sua eficiência e quebras, assim é sempre necessário atenção para se fazer manutenções.

Convencionalmente, as técnicas de manutenção em máquinas consistem em corretivas, preventivas e preditivas. Manutenção corretiva é a correção após uma falha ou quando o desempenho é menor que o esperado, manutenção preventiva é a atuação para reduzir ou evitar uma falha, baseado em dados históricos elaborados definidos por tempo de uso e manutenção preditiva é a atuação para evitar falhas baseados em dados reais retirados do equipamento em questão (PINTO; XAVIER, 2001). Mesmo que manutenções preventivas sejam as mais populares atualmente, estas podem ser ineficientes por se trocar peças antes do fim de tempo de vida real, causando grandes perdas financeiras.

No processo de manutenção existem diferentes métodos aplicáveis. O método de manutenção preditiva é o considerado mais eficiente por se utilizar o maior tempo possível de uma peça antes da falha acontecer. Para aplicar este método é necessário estudar a máquina e prever tais falhas, o estudo de prever é denominado prognóstico de falhas. Os métodos de prognósticos podem ser subdividido em três tipos, sendo eles: o prognóstico baseado em experiências, que realiza previsões a partir de modelos probabilísticos de experiências passadas, o prognóstico baseado em dados, que utiliza dados previamente observados para fazer uma projeção da máquina e o prognóstico baseado em modelos matemáticos, onde se faz uma modelagem matemática dinâmica da máquina para calcular e simular eventos futuros (LIAO; KÖTTIG, 2014).

O desenvolvimento de um sistema de Aprendizado de Máquina (em inglês, *Machine Learning*) para a detecção de falhas é imprescindível para o futuro da indústria 4.0. Sendo essa um conceito que representa a automação industrial e a integração de diferentes tecnologias computacional nas indústrias (INDÚSTRIA 4.0: ENTENDA SEUS CONCEITOS E FUNDAMENTOS, 2019). Para o desenvolvimento de algoritmos preditivos e utilização de AM são necessários conjuntos de dados de treinamento rotulados para induzir um modelo preditivo capaz de prever, para um novo objeto apresentado pelos valores de seus atributos preditivos, o

valor de seu atributo alvo (FACELI et al., 2021).

Um novo nicho de tecnologia está emergindo com a indústria 4.0, Gêmeos Digitais (em inglês, *Digital Twin*) o desenvolvimento de réplicas virtuais de sistemas físicos, como máquinas, ou digitais, como processos. Estes sistemas podem ser utilizados para simular estruturas e prever suas falhas. Para se desenvolver um GD é necessário muitos dados da estrutura que se quer simular (HAAG; ANDERL, 2018). Notável o aumento da necessidade de se extrair, tratar e armazenar dados no mundo de hoje.

Máquinas rotativas elétricas estão entre as mais utilizadas no setor industrial (BOUDINAR et al., 2016), além disto estas são responsáveis por utilizar cerca de 26% de toda energia gerada no Brasil (WEG, 2019). Considerando isto, é notável a importância destes equipamentos para a produção, e mantê-los em pleno funcionamento é necessário para sua alta eficiência constante, diminuindo assim os gastos energéticos.

Dentre as falhas possíveis em motores elétricos, as mais abundantes são as falhas em rolamentos, responsáveis por cerca de 40% a 90% das falhas, dependendo do tamanho dessa (IMMOVILLI et al., 2013). É notável que identificar tais falhas antes de causarem estragos irreversíveis é de grande importância para a indústria.

Na literatura têm-se diversas técnicas de monitoramento e diagnóstico de falhas em motores elétricos. Dentre elas tem-se a análise de vibrações mecânicas. Pode ser feita um estudo analítico e experimental (IMMOVILLI et al., 2013; BOUDINAR et al., 2016), também têm-se estudos de como avaliar os valores de experimentos e como processá-los (ABDULBARY; EMBABY; GOMAA, 2021). Notável que estes envolvem experimentos com motores físicos.

Tendo em vista isto, observa-se a necessidade do desenvolvimento de um banco de dados para o AM baseado em modelagem matemática de motores elétricos. Assim, tem-se um estudo em torno das variáveis de interesse para um AM eficaz sem a necessidade de realizar testes em bancada em um motor elétrico. O estudo das variáveis de interesse se dá por meios estatísticos, um modo de se calcular estes são por sistemas computacionais, dentre os mais utilizados está a linguagem de programação Python, uma linguagem de grande aceitação e de aplicação livre, existem diversos trabalhos que se utilizam desta linguagem para produzir simulações computacionais como GALVAO (2017), FERNANDES (2021) e PORTELA (2022).

1.1 OBJETIVOS

O objetivo geral deste trabalho é mostrar modelos numéricos e simular motores elétricos baseando-se em modelos matemáticos na literatura e desenvolver banco de dados com as variáveis mais importantes para futuramente utilizar os dados em um AM.

Para atingir este objetivo principal, foram estabelecidos os seguintes objetivos específicos:

- Realizar uma revisão teórica dos principais temas do trabalho como AM, as componentes móveis de motores elétricos, modelagem matemática e análises vibracional;
- Montar um código dedicado a realizar a simulação numérica do modelo matemático obtido;
- Simular o modelo matemático em Python e visualizar graficamente como os dados variam de acordo com parâmetros funcionais;
- Fazer o tratamento estatístico dos resultados da simulação;
- Avaliar como estes parâmetros mecânicos variam dentro de um sistema simulado e analisar quais dos dados tratados são relevantes para o estudo do sistema;
- Verificar se uma simulação computacional é o suficiente para avaliar o funcionamento de um motor real.

2 REVISÃO TEÓRICA

Neste capítulo é apresentado os principais conceitos teóricos utilizadas no decorrer deste trabalho, necessários para o entendimento da metodologia aplicada. Primeiramente será realizado uma breve explicação de vibrações e das partes mecânicas de motores elétricos, bem como seu modelo matemático e sua dinâmica estrutural. Posteriormente, é apresentado meios de processar sinais e por fim será explicado o conceito de AM e os passos necessários para sua aplicação.

2.1 VIBRAÇÕES MECÂNICAS

Vibração é denominado como qualquer movimento que se repita após um intervalo de tempo. Um sistema pode ser denominado vibratório se possuir três principais meios, um meio de armazenar energia potencial (mola ou elasticidade), um meio para armazenar energia cinética (massa ou inércia) e um meio de perda gradual de energia (amortecedor) (RAO, 2009).

A vibração de um sistema envolve a transferência alternada de sua energia potencial para a energia cinética e de energia cinética para energia potencial. Se o sistema for amortecido, certa quantidade de energia é dissipada em cada ciclo de vibração (RAO, 2009).

2.1.1 Equação do Movimento para Sistemas de um Grau de Liberdade

Sabendo que a energia de um sistema vibratório é composto por energia cinética (T) e energia potencial (U), e que, considerando a conservação de energia, a soma delas é constante, temos:

$$T + U = \text{constante} \quad (2.1)$$

de forma que sua derivada no tempo é dada por:

$$\frac{d}{dt}(T + U) = 0 \quad (2.2)$$

Sabendo que as energias são dadas por:

$$T = \frac{1}{2}m\dot{x}^2 \quad U = \frac{1}{2}kx^2 \quad (2.3)$$

com x sendo a variável dependente do sistema, sendo em função do tempo (t), e representa o deslocamento, m a massa do sistema e k a constante elástica do sistema. Combinando as Equações 2.3 na 2.2 pode-se obter:

$$m\ddot{x} + kx = 0 \quad (2.4)$$

Caso o sistema seja amortecido, tem-se mais uma variável no sistema, sendo uma força ligada a velocidade do deslocamento e a uma constante de amortecimento c , tem-se então a equação final do movimento como:

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (2.5)$$

para sistemas livres e

$$m\ddot{x} + c\dot{x} + kx = F(t) \quad (2.6)$$

para sistemas excitados por força $F(t)$ externa.

2.1.2 Múltiplos Graus de Liberdade

Para sistemas com múltiplos graus de liberdade, têm-se diversas equações como a equação 2.6 para diferentes deslocamentos, os quais podem ser acoplados entre si. É possível então escrever a função do movimento matricialmente. Sendo ela:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f} \quad (2.7)$$

tal que, para n graus de liberdade:

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{pmatrix} \quad (2.8)$$

A equação diferencial 2.7 será resolvida pelo método Runge-Kutta, sendo um método iterativo implícito e explícito para resolução numérica de equações diferenciais ordinárias (BUTCHER, 1996). Para se utilizar este método é necessário transformar a equação de segunda ordem em um de primeira ordem, para isto, será utilizado o método do espaço de estados, que deixara a equação arranjada da seguinte forma:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (2.9)$$

onde \mathbf{x} é o vetor de estados, incluindo todos os graus de liberdade de interesse. Para "n" graus de liberdade, tem-se:

$$\mathbf{x} = \begin{pmatrix} q_1 \\ q_2 \\ \cdot \\ q_n \\ \dot{q}_1 \\ \dot{q}_2 \\ \cdot \\ \dot{q}_n \end{pmatrix} = \begin{Bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{Bmatrix} \quad (2.10)$$

Para se encontrar as matrizes \mathbf{A} e \mathbf{B} deve-se primeiro isolar a matriz $\ddot{\mathbf{q}}$ na sequência a seguir:

$$\mathbf{M}\ddot{\mathbf{q}} = -\mathbf{K}\mathbf{q} - \mathbf{C}\dot{\mathbf{q}} + \mathbf{f} \quad (2.11)$$

$$\mathbf{M}^{-1}\mathbf{M}\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\mathbf{K}\mathbf{q} - \mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{q}} + \mathbf{M}^{-1}\mathbf{f} \quad (2.12)$$

$$\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\mathbf{K}\mathbf{q} - \mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{q}} + \mathbf{M}^{-1}\mathbf{f} \quad (2.13)$$

Organizando a equação 2.13 na forma matricial temos:

$$\begin{Bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{Bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{Bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{M}^{-1} \end{bmatrix} \begin{Bmatrix} 0 \\ \mathbf{f} \end{Bmatrix} \quad (2.14)$$

Com isto tem-se que, sabendo as matrizes de massa (\mathbf{M}), rigidez (\mathbf{K}), amortecimento (\mathbf{C}) e o vetor de forças externas (\mathbf{f}), é possível calcular todas as variáveis dependentes.

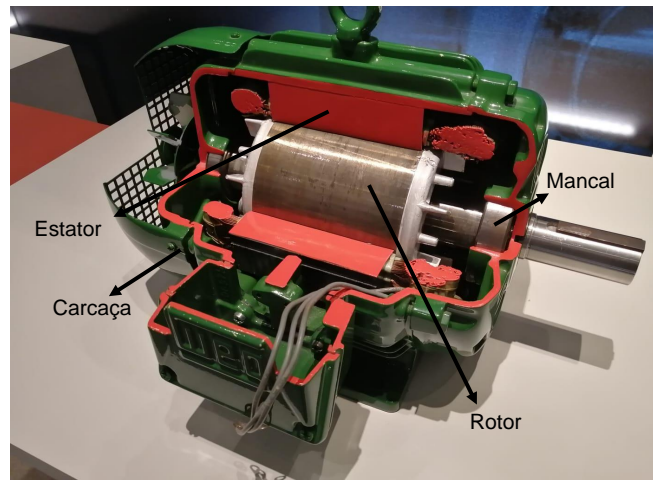
2.2 MOTORES ELÉTRICOS

Motores Elétricos são máquinas rotativas capazes de transformar energia elétrica em energia mecânica. Estes são feitos de diversos componentes, mas para se ter uma modelagem dinâmica suficientemente completa são necessários o entendimento de três principais partes, o rotor, o estator e a carcaça, essas podem ser observadas na Figura 1 (FRISWELL et al., 2010).

Considerado o coração de qualquer máquina rotativa, o rotor é responsável por efetuar o movimento rotativo. Para máquinas pequenas o rotor é consideravelmente simples de modelar, visto apenas como um eixo com um disco rígido, mas para rotores maiores é necessário considerar a flexibilidade de tal disco (FRISWELL et al., 2010).

O estator é a parte estacionária do motor, por questões de simplicidade ele é considerado rígido, engloba o rotor e é o responsável por transmitir os campos que fazem com que o rotor rotacione. A carcaça é o suporte onde o rotor e estator estão incorporados, responsável por manter a máquina no local.

Figura 1 – Aspectos construtivos de motores elétricos.

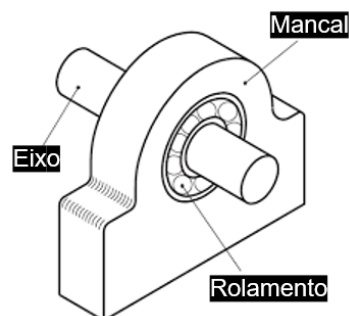


Fonte: Autor.

O rotor está acoplado aos mancais de rolamento, que no que lhe concerne estão acoplados à carcaça. Os mancais tem como principal objetivo conectar um eixo girante a um componente estacionário, perdendo o mínimo possível de carga.

A utilização de mancais de rolamento estão entre os mais utilizados para rotores de médio porte, já que providenciam boa capacidade de carga e são rígidos. Porém, como possuem partes interna móveis, estes contribuem bastante na dinâmica do equipamento (FRISWELL et al., 2010).

Figura 2 – Componentes de mancais de rolamento.



Fonte: GORDO; FERREIRA (2000).

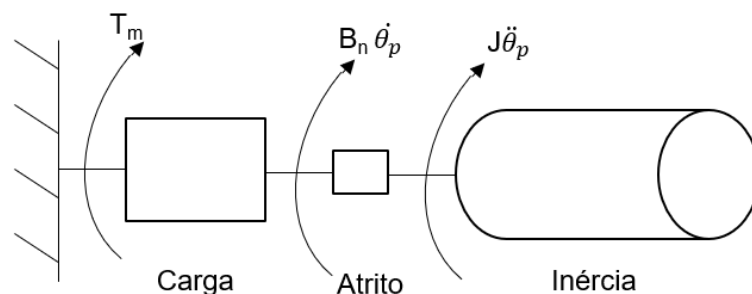
2.2.1 Modelagem do Motor

A modelagem de um motor elétrico foi feita em duas etapas. Primeiramente a modelagem funcional mecânica, considerando um grau de liberdade para a posição angular do eixo, bem como sua velocidade. E posteriormente uma modelagem dinâmica ligada diretamente à rigidez dos mancais de rolamento.

O torque eletromagnético desenvolvido (T_e) pela máquina é igual a soma do torque da carga (T_m) desta. O torque desenvolvido pelo motor é dividido em dois, sendo estes o torque de aceleração e o torque de perdas por atrito. O torque de aceleração é dado em função do momento de inércia da carga e do rotor (J). E as perdas por atrito se dão em função do coeficiente de atrito (B_n) entre a carga e o motor, tal que, em função da posição angular do eixo (θ_p) (CÂMARA, 2007):

$$J\ddot{\theta}_p + B_n\dot{\theta}_p = T_e - T_m \quad (2.15)$$

Figura 3 – Diagrama de corpo livre do motor de indução trifásico.



Fonte: Autor.

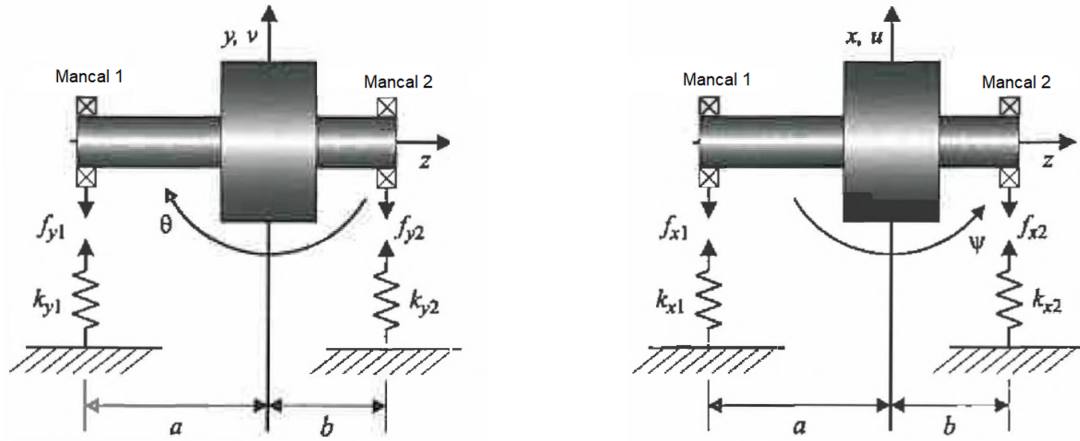
Sabendo que a derivada da posição em função do tempo é a velocidade, temos a velocidade angular do motor de estudo (Ω):

$$\dot{\theta}_p = \Omega \quad (2.16)$$

A dedução completa para o sistema de quatro graus de liberdade é extensa e pode ser vista com mais detalhes em (FRISWELL et al., 2010). Neste trabalho será apresentado o sistema final para um rotor e eixo rígidos para mancais isotrópicos e flexível com desbalanceamento de massa e deslocamento do eixo.

O rotor possui quatro graus de liberdade, pois pode transladar nas direções x e y , bem como rotacionar nestas mesmas direções θ na direção x e ψ na direção y .

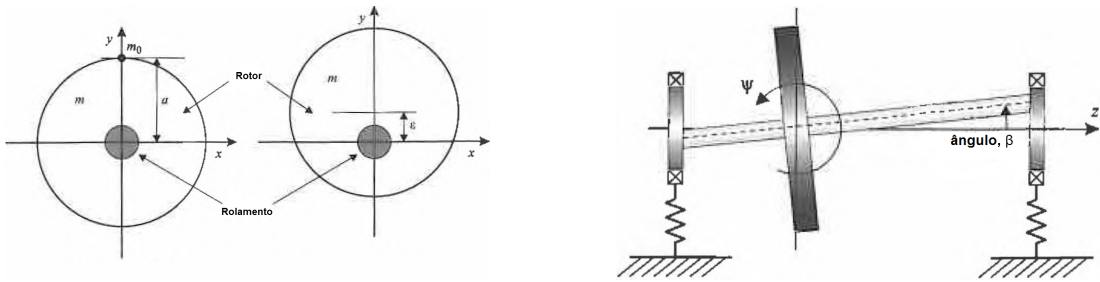
Figura 4 – Diagrama de corpo livre do rotor com rotor e eixo rígido e mancais flexíveis



(a) Visão frontal (b) Visão superior
 Fonte: ADAPTADO DE FRISWELL et al. (2010).

Todo tipo de máquina rotativa possui falhas estruturais que afetam seu funcionamento dinâmico, dentre os principais tem-se o desbalanceamento de massa (Figura 5a) e deslocamento do eixo rotativo (Figura 5b).

Figura 5 – Diagrama do rotor para diferentes casos.



(a) Desbalanceamento de massa (b) Eixo deslocado
 Fonte: ADAPTADO DE FRISWELL et al. (2010).

Considerando os mancais isotrópicos, desbalanceamento de massa e deslocamento do eixo, temos que a rigidez na direção x e na direção y são iguais, tem-se:

$$\begin{cases} m\ddot{u} + c_T\dot{u} + c_C\dot{\psi} + k_Tu + k_C\psi & = m\epsilon\Omega^2 \cos \Omega t \\ m\ddot{v} + c_T\dot{v} - c_C\dot{\theta} + k_Tv - k_C\theta & = m\epsilon\Omega^2 \sin \Omega t \\ I_d\ddot{\theta} + I_p\Omega\dot{\psi} - c_C\dot{v} + c_R\dot{\theta} - k_Cv + k_R\theta & = -(I_d - I_p) \beta\Omega^2 \sin \Omega t \\ I_d\ddot{\psi} - I_p\Omega\dot{\theta} + c_C\dot{u} + c_R\dot{\psi} + k_Cu + k_R\psi & = (I_d - I_p) \beta\Omega^2 \cos \Omega t \end{cases} \quad (2.17)$$

Onde m é a massa do rotor, I_d o momento de inércia no eixo central, I_p o momento de inércia polar do rotor, ϵ o raio de desbalanceamento de massa e β o ângulo de deslocamento

do eixo. O efeito giroscópico do rotor pode ser visto nos termos que se tem Ω , onde este é a velocidade angular do rotor, possível ser obtido através da equação 2.15. Os subscritos T , C e R para o coeficiente de rigidez k e coeficiente de amortecimento c foram escolhidos para indicar translação, acoplamento entre deslocamento e rotação e coeficiente de rigidez rotacional, respectivamente. Sendo esse calculado como:

$$\begin{cases} k_1 + k_2 & = k_T \\ -ak_1 + bk_2 & = k_C \\ a^2k_1 + b^2k_2 & = k_R \end{cases} \quad \begin{cases} c_1 + c_2 & = c_T \\ -ac_1 + bc_2 & = c_C \\ a^2c_1 + b^2c_2 & = c_R \end{cases} \quad (2.18)$$

2.2.2 Sistema Completo com 5 Graus de Liberdade

Combinando o sistema mecânico do motor (Eq. 2.15) com o sistema vibracional (Eq. 2.17), sabendo-se que a derivada da posição angular em função do tempo é a velocidade ($\dot{\theta}_p = \Omega$), se obtém um sistema de 5 equações:

$$\begin{cases} J\ddot{\theta}_p + B_n\dot{\theta}_p & = T_e - T_m \\ m\ddot{u} + c_T\dot{u} + c_C\dot{\psi} + k_Tu + k_C\psi & = m\epsilon\dot{\theta}_p^2 \cos \dot{\theta}_p t \\ m\ddot{v} + c_T\dot{v} - c_C\dot{\theta} + k_Tv - k_C\theta & = m\epsilon\dot{\theta}_p^2 \sin \dot{\theta}_p t \\ I_d\ddot{\theta} + I_p\dot{\theta}_p\dot{\psi} - c_C\dot{v} + c_R\dot{\theta} - k_Cv + k_R\theta & = -(I_d - I_p)\beta\dot{\theta}_p^2 \sin \dot{\theta}_p t \\ I_d\ddot{\psi} - I_p\dot{\theta}_p\dot{\theta} + c_C\dot{u} + c_R\dot{\psi} + k_Cu + k_R\psi & = (I_d - I_p)\beta\dot{\theta}_p^2 \cos \dot{\theta}_p t \end{cases} \quad (2.19)$$

Tendo todo o sistema completo é possível descrever isto na forma matricial (Eq. 2.7), com os componentes definidos a seguir:

$$\mathbf{q} = \begin{pmatrix} \theta_p \\ u \\ v \\ \theta \\ \psi \end{pmatrix} \quad (2.20)$$

$$\mathbf{M} = \begin{bmatrix} J & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 \\ 0 & 0 & 0 & I_d & 0 \\ 0 & 0 & 0 & 0 & I_d \end{bmatrix} \quad (2.21)$$

$$\mathbf{C} = \begin{bmatrix} B_n & 0 & 0 & 0 & 0 \\ 0 & c_T & 0 & 0 & c_C \\ 0 & 0 & c_T & -c_C & 0 \\ 0 & 0 & -c_C & c_R & \dot{\theta}_p I_p \\ 0 & c_C & 0 & -\dot{\theta}_p I_p & c_R \end{bmatrix} \quad (2.22)$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & k_T & 0 & 0 & k_C \\ 0 & 0 & k_T & -k_C & 0 \\ 0 & 0 & -k_C & k_R & 0 \\ 0 & k_C & 0 & 0 & k_R \end{bmatrix} \quad (2.23)$$

$$\mathbf{f} = \begin{pmatrix} T_e - T_m \\ m\epsilon\dot{\theta}_p^2 \cos(\dot{\theta}_p t) \\ m\epsilon\dot{\theta}_p^2 \sin(\dot{\theta}_p t) \\ -(I_d - I_p)\beta\dot{\theta}_p^2 \sin(\dot{\theta}_p t) \\ (I_d - I_p)\beta\dot{\theta}_p^2 \cos(\dot{\theta}_p t) \end{pmatrix} \quad (2.24)$$

2.3 PROCESSAMENTO DE SINAL

Ao medir sinais de vibração, é comum observar certos padrões, e é de grande interesse do estudo de vibrações observar e compreender estes padrões de repetições. Existem diversas maneiras de se analisar sinais, desde métodos visuais ao se comparar com a literatura, ou com métodos analíticos de tratamento do sinal. Para melhor visualização do sinal é possível efetuar o processamento através de softwares como a Transformada Rápida de Fourier (*Fast Fourier Transform* — FFT). Para entender melhor como este tipo de software funciona, é necessário compreender o que é a transformada de Fourier.

2.3.1 Classificação de Sinal

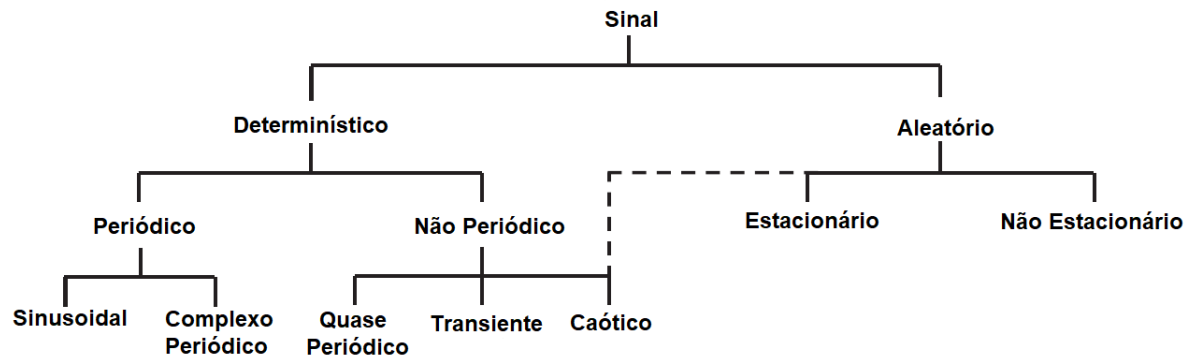
De acordo com SHIN; HAMMOND (2008), sinais em função do tempo podem ser categorizados como na Figura 6. A diferença fundamental entre sinais é se este é determinístico ou aleatório e o método de análise é determinado conforme o tipo de sinal.

Normalmente sinais são misturas de várias classificações, assim, o tipo de análise pode não ser tão aparente. Com isso, um conhecimento prévio do tipo de sistema é importante para a análise desse. No caso de estudo temos um sistema todo determinístico, visto que todas as variáveis estão sob controle do usuário.

Sinais periódicos são definidos por sinais que se repetem de forma exata em intervalos regulares de tempo (SHIN; HAMMOND, 2008) ditos como período (T_p), como visto na Figura 7. Sinais não periódicos não seguem este padrão.

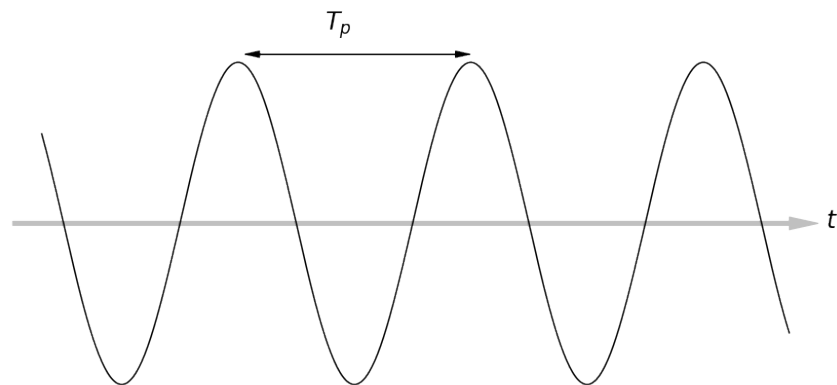
Sinais quase periódicos são aqueles que aparentam ser periódicos, mas após inspecionar mais afundo é visto que não são assim, tornando o sistema mais imprevisível, este tipo de

Figura 6 – Classificações de sinais.



Fonte: ADAPTADO DE SHIN; HAMMOND (2008).

Figura 7 – Sinal periódico seno.



Fonte: Autor.

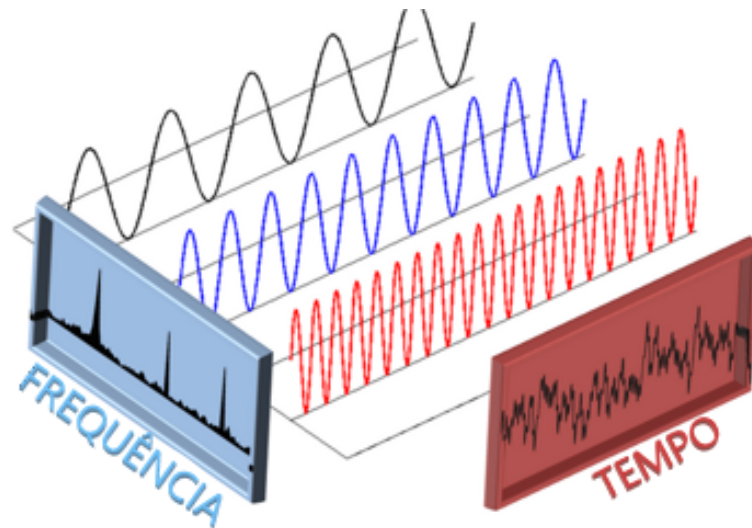
sinal é o mais abundante na natureza. Esta inspeção é efetuada com o auxílio de análises de frequências. É possível confirmar o tipo de sinal através da análise da transformada de Fourier.

2.3.2 Transformada de Fourier

Praticamente todo tipo de sinal pode ser descrito na forma de ondas, uma função de tempo, espaço e outra variável. A Transformada de Fourier (TF) demonstra que qualquer onda pode ser representadas em somas de senos e cossenos (BEVELACQUA, 2010).

A TF é um procedimento matemático que decompõe qualquer função no domínio do tempo em uma função ou série no domínio de sua frequência como pode ser observado na Figura 8.

Figura 8 – Correlação entre sinal no domínio do tempo e da frequência.



Fonte: ADAPTADO DE ENSUS ENGENHARIA (2016).

A TF pode ser calculada para uma função contínua $x(t)$ como:

$$S(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (2.25)$$

onde $S(f)$ é o resultado da transformada de Fourier no domínio da frequência f .

Para uma função periódica e contínua, $x(t)$, é possível calcular sua forma em somas de senos e cossenos, feita por Séries de Fourier:

$$x(t) = a_0 + \sum_{m=0}^{\infty} a_m \cos\left(\frac{2\pi mt}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi nt}{T}\right) \quad (2.26)$$

onde a_m e b_n são os coeficientes da Série de Fourier.

Apesar desta forma ser válida para avaliar funções, ela se utiliza de meios contínuos, ou seja, infinitos. Para modelos computacionais, é necessário a discretização desta, ou seja, com um número finito de amostras. Para isto, se utiliza de técnicas como a Transformada Discreta de Fourier (TDF).

Tendo isto, é possível ver que uma aproximação da Transformada de Fourier de sinais periódicos é a TDF. A equação para encontrar a TDF é:

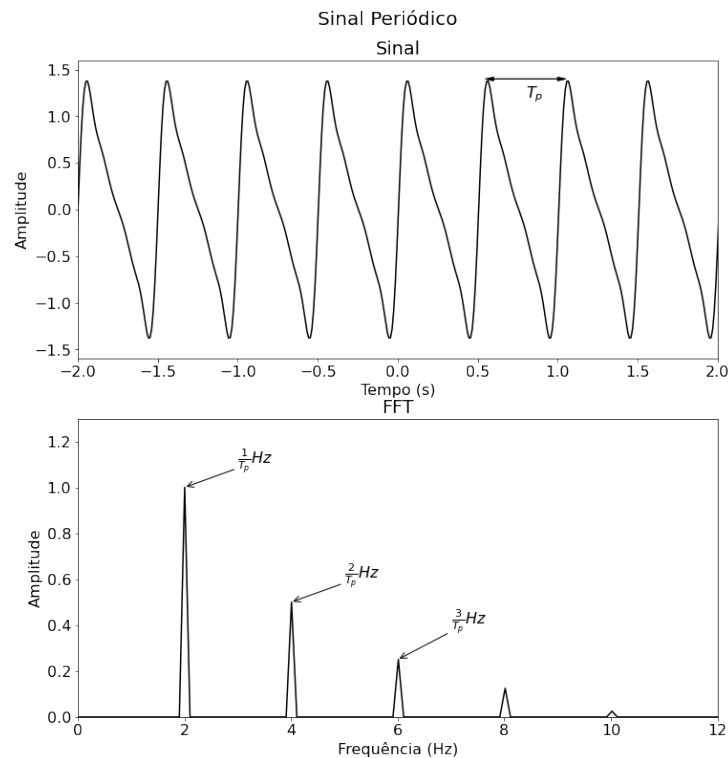
$$A_r = \sum_{k=0}^{N-1} X_K e^{(-2\pi jr k/N)} \quad (2.27)$$

A dedução completa para a TDF pode ser vista em COCHRAN et al. (1967)

2.3.3 Transformadas de Fourier e Classificação de Sinal

Através da análise da transformada de Fourier é possível dizer qual a classificação do sinal. Para sinais determinísticos e periódicos é possível observar as frequências fundamentais do sinal relacionadas de forma harmônica, onde cada pico de amplitude representa uma frequência fundamental, sendo esta um fator da frequência natural do sistema (SHIN; HAMMOND, 2008). Esta transformada pode ser constatada na Figura 9, com as frequências fundamentais marcadas e seus valores apresentados.

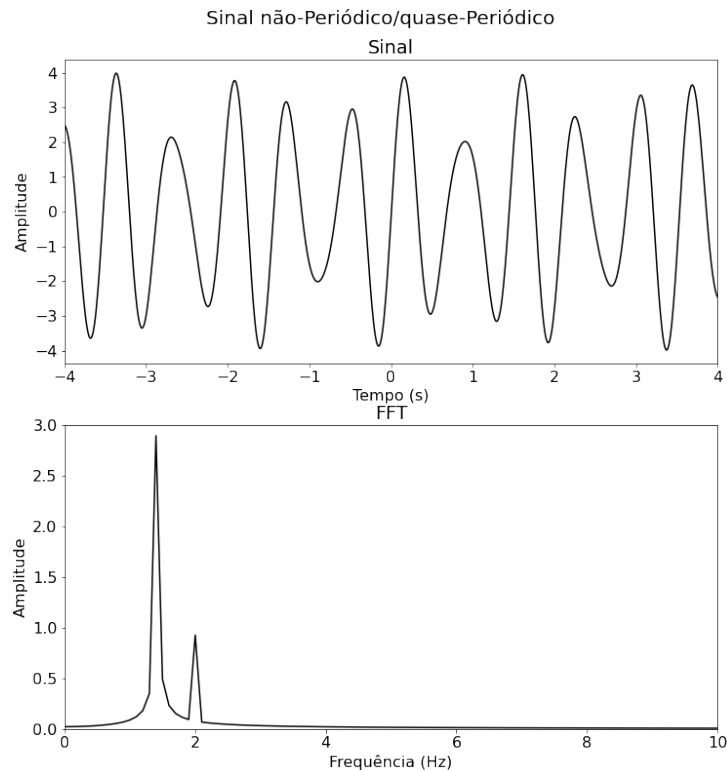
Figura 9 – Transformada de Fourier de um sinal periódico com período (T_p) de 0,5.



Fonte: Autor.

Analisando a transformada de Fourier é possível confirmar se o sinal é periódico ou quase periódico, visto que é possível analisar que as frequências fundamentais não seguem a mesma lógica de sinais periódicos, exemplo pode ser observado na Figura 10. Os picos de amplitude ocorre em dois pontos sem ligação entre si, assim, este sinal nunca se repetira de forma exata.

Figura 10 – Transformada de Fourier de um sinal quase periódico.



Fonte: Autor.

2.3.4 Transformada Rápida de Fourier

A FFT é um algoritmo que se utiliza do método TDF para calcular a Transformada de Fourier da maneira mais eficiente possível. Ele nos dá o mesmo resultado que a TDF, porém com muito menos tempo computacional e também diminui substancialmente o número de erros por arredondamento (COCHRAN et al., 1967). Neste trabalho será utilizado o método de FFT a partir da biblioteca *numpy* do Python (HARRIS et al., 2020).

2.3.5 Análise Estatística

Existem diversas maneiras de se processar os dados, sendo que algumas maneiras de se interpretá-los são por estatística descritiva (FACELI et al., 2021). A estatística descritiva resume quantitativamente as principais características de um conjunto de dados. As medidas estatísticas descritivas assumem que os dados são gerados por um processo estatístico.

É possível encontrar valores que determinam a tendência central dos valores gerados. Estes valores definem pontos de referência nos dados, em termos numéricos podem ser definidos

como:

- Frequência;
- Média (\bar{x}); calculada por:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.28)$$

Alguns parâmetros estatísticos determinam o espalhamento ou dispersão dos dados gerados, estes valores são utilizados para observar se os dados estão amplamente espalhados ou relativamente concentrados em torno de um valor. Alguns parâmetros que podem nos dar estes dados quantitativos são:

- Intervalo; calculado por:

$$\text{intervalo}(x) = x_{\text{máx}} - x_{\text{mín}} \quad (2.29)$$

- Variância; calculada por:

$$\text{variância}(x) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.30)$$

- Desvio padrão (s); calculado por:

$$s(x) = \sqrt{\text{variância}(x)} \quad (2.31)$$

Outros parâmetros estatísticos determinam a distribuição dos dados gerados, como obliquidade e curtose, esses serão vistos com mais detalhes abaixo.

2.3.5.1 Obliquidade

Obliquidade mede a simetria de distribuição em torno da média, em uma distribuição simétrica, um histograma com a quantidade de valores em cada intervalo tem a mesma aparência a direita e à esquerda do centro. A equação 2.32 utiliza o desvio padrão ao cubo, s^3 , para tornar a medida independente da escala (FACELI et al., 2021).

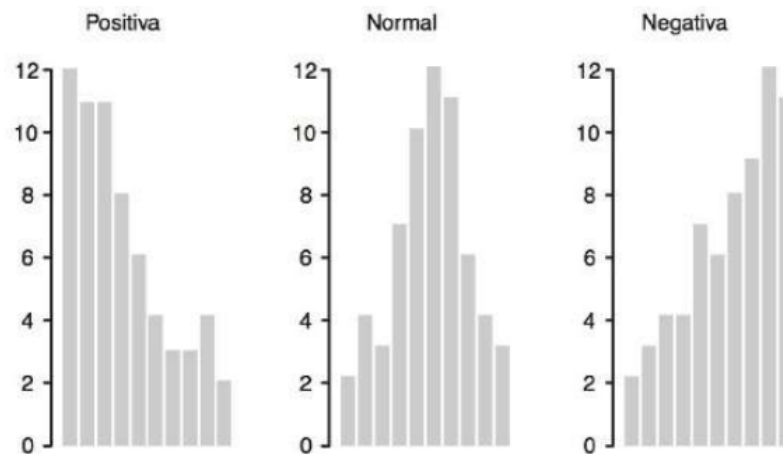
$$\text{obliquidade} = \frac{1}{(n-1)s^3} \sum_{i=1}^n (x_i - \bar{x})^3 \quad (2.32)$$

A distribuição dos valores em um conjunto de dados está associada ao valor da obliquidade da seguinte de forma que:

- Obliquidade = 0: a distribuição é aproximadamente simétrica;
- Obliquidade > 0: a distribuição concentra-se mais no lado esquerdo;
- Obliquidade < 0: a distribuição concentra-se mais no lado direito;

A Figura 11 mostra a obliquidade para os três tipos de distribuição.

Figura 11 – Distribuição dos valores de obliquidade.



Fonte: FACELI et al. (2021).

2.3.5.2 Curtose

Curtose mede a dispersão que captura o achatamento da função de distribuição. A medida de curtose, calculada na equação 2.33 verifica se os dados apresentam um pico ou são achatados em relação a uma distribuição normal (FACELI et al., 2021).

$$\text{curtose} = \frac{1}{(n-1)s^4} \sum_{i=1}^n (x_i - \bar{x})^4 - 3 \quad (2.33)$$

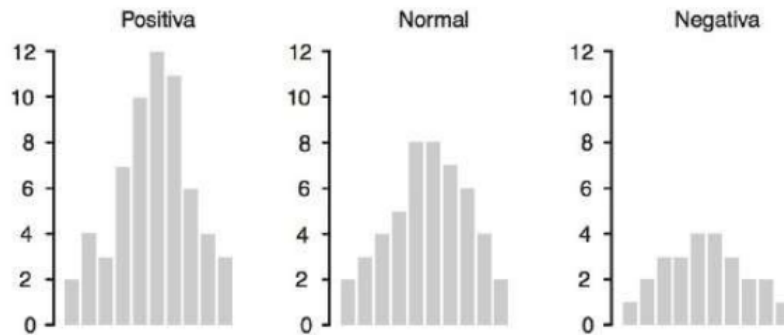
A distribuição dos valores em um conjunto de dados está associada ao valor da curtose da seguinte forma que:

- Curtose = 0: o histograma de distribuição dos dados apresenta o mesmo achatamento que uma distribuição normal;
- Curtose > 0: o histograma de distribuição dos dados apresenta uma distribuição mais alta e concentrada que a distribuição normal;

- Curtose < 0 : o histograma de distribuição dos dados apresenta uma distribuição mais achatada que a distribuição normal.

A Figura 12 mostra a curtose para os três tipos de distribuição.

Figura 12 – Distribuição dos valores de curtose.



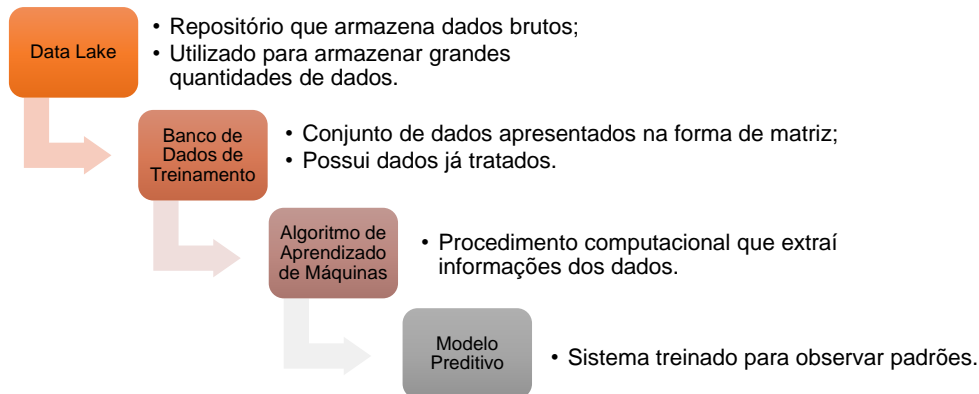
Fonte: FACELI et al. (2021).

2.4 APRENDIZADO DE MÁQUINAS

Aprendizado de máquinas é o procedimento, o processo de aprendizado de um algoritmo através de padrões de um banco de dados entregue para este com o intuito de fazer previsões e tomar decisões de um novo conjunto de dados (FACELI et al., 2021). Os algoritmos de AM preditivos são funções que, dado um conjunto de exemplos rotulados, constrói um estimador. Para o desenvolvimento deste é necessário um conjunto de dados de treinamento, que, para um algoritmo de previsão de falhas, devem ser divididos entre um conjunto de dados saudável e um conjunto de dados com falhas (FACELI et al., 2021).

Os conjuntos de dados devem ser rotulados entre saudável e com falha, tendo assim, um sistema de AM supervisionado. Este termo vem da simulação da presença de um supervisor externo, que conhece o verdadeiro diagnóstico do conjunto de dados apresentado. A informação do supervisor é usada para guiar o processo de aprendizado na extração de um modelo com boa capacidade preditiva (FACELI et al., 2021). O processo pode ser visualizado na Figura 13. Os termos são descritos nas seções a seguir.

Figura 13 – Fluxograma do desenvolvimento de um sistema preditivo em AM.



Fonte: Autor.

2.4.1 Data Lake

Data Lake é um tipo de repositório que armazena os dados brutos de um sistema sem nenhum processamento. Utilizado para armazenar grandes quantidade de dados para ser usado futuramente sem nenhum tipo de perda por processamento (MILOSLAVSKAYA; TOLSTOY, 2016).

2.4.2 Banco de Dados de Treinamento

Banco de dados é o conjunto de dados apresentados na forma de matriz, com cada linha simbolizando um objeto e nas colunas os tipos de atributos. Considerando o caso do desenvolvimento de um modelo preditivo, ainda se tem como último atributo o alvo, como valor binário para máquinas saudáveis e máquinas defeituosas. Para casos de valores quantitativos, como em análises dinâmicas, estes atributos assumem números e são considerados contínuos, ou seja, não possuem limites inferiores ou superiores.

Tendo em vista os meios de processamento de sinal, é possível criar um banco de dados eficaz calculando os tipos de processamento para cada variável de estudo. Com isto tem-se tudo o que é necessário para o desenvolvimento de um banco de dados para motores elétricos.

2.4.3 Algoritmo de AM

Sistema computacional desenvolvido para analisar dados do banco de dados de treinamento e aprender os padrões desses. Estes algoritmos não necessitam de um modelo matemático

definido, aprendem a desenvolver e definir padrões a partir de dados extraídos, estes podem ser previamente rotulados com a resposta ou não (FONTANA, 2020).

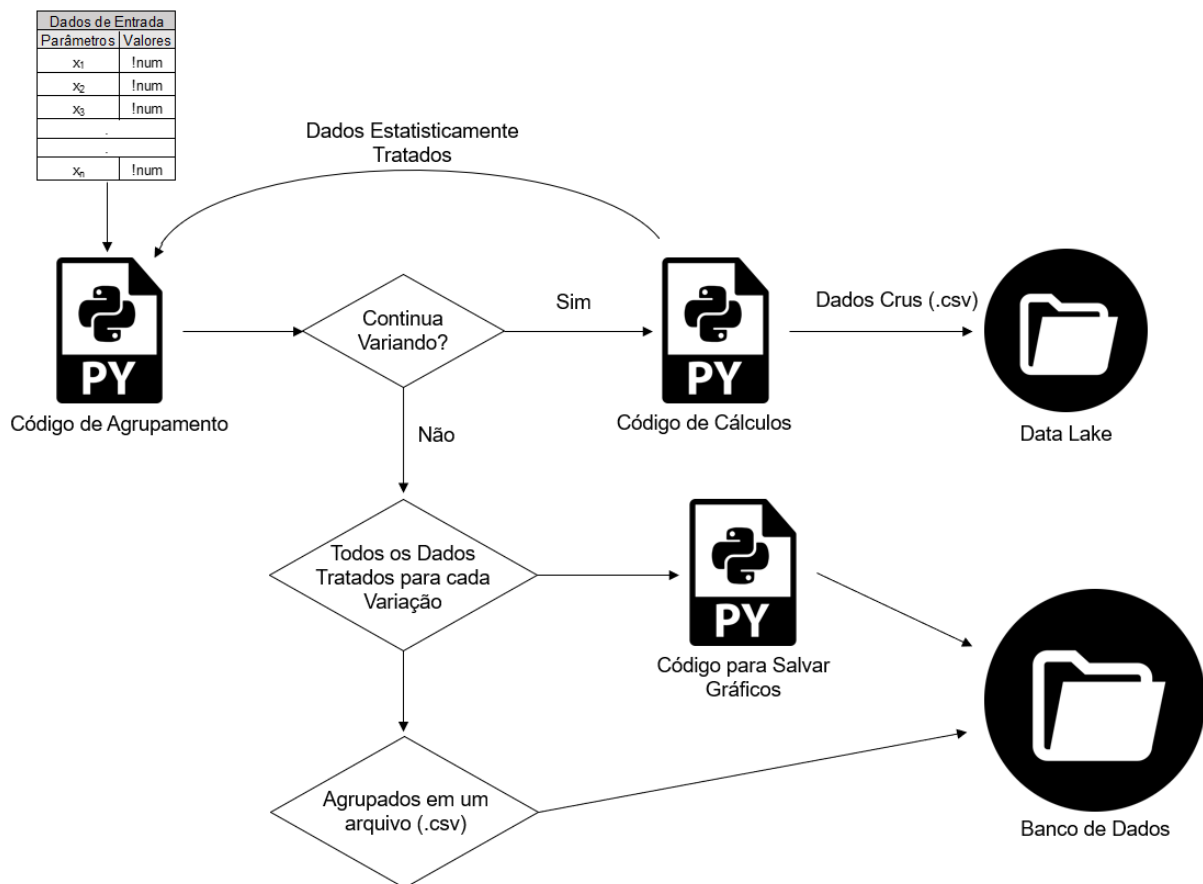
2.4.4 Modelo Preditivo

Sistema computacional treinado a partir do algoritmo de AM, pronto para analisar dados reais e definir uma resposta com certa confiabilidade, dependendo da qualidade do sistema e dos dados apresentados a ele.

3 METODOLOGIA PROPOSTA

Neste capítulo é abordado os procedimentos feitos para simular, processar os dados e rotular o motor elétrico do estudo. O procedimento para o desenvolvimento do banco de dados pode ser visto na Figura 14. Todos os códigos necessários foram feitos na linguagem de programação aberta Python.

Figura 14 – Fluxograma do procedimento realizado para o desenvolvimento de banco de dados.



Fonte: Autor.

Considerando tempo e despesas inviáveis no âmbito acadêmico para a realização de testes com motores reais, a metodologia deste estudo foi escolhida pelo mérito de propiciar uma simulação próxima da realidade, fornecendo dados e possibilitando a criação de um banco de dados para os mesmos. No geral, a metodologia exerce o papel de alternativa para provar a utilização de métodos matemáticos para o desenvolvimento de banco de dados, a qual pode ser aplicada na realidade com motores reais, entretantes o tempo e investimento demandado para isso cairá com o aperfeiçoamento deste método.

3.1 DADOS DE ENTRADA

Os dados de entrada são todos os valores necessários para montar as matrizes \mathbf{M} , \mathbf{C} , \mathbf{K} e \mathbf{f} (Eq. 2.7), bem como qual parâmetro destes irá variar, onde este inicia, termina e qual o passo de variação. Considerando o fato desta simulação ser em função do tempo, também é necessário a definição do tempo e o passo de simulação.

Os dados de entrada serão todos aqueles necessários para o desenvolvimento do sistema de 5 graus de liberdade, bem como todos os parâmetros temporais e de variação. Os temporais serão:

- Tempo de simulação (t);
- Passo de tempo (dt).

Os parâmetros de variação serão:

- Parâmetro estrutural que será estudado (como componente elástica de um dos mancais, k_1 ou k_2);
- Início com relação a um valor nominal;
- Final com relação a um valor nominal;
- Passo de variação.

3.2 CÓDIGO DE AGRUPAMENTO

O código de agrupamento é responsável por organizar os dados de entrada e organizá-los para enviar para os outros códigos. Este também é responsável por variar o parâmetro previamente escolhido.

Este código recebe de volta os dados tratados do código de cálculo e agrupa estes, no fim da variação este manda os dados tratados em função da variação para um código responsável por criar e salvar os gráficos bem como salvar estes na forma numérica em arquivos *CSV*¹ na pasta de banco de dados. Ele pode ser visto no Apêndice A.1.

¹ Por definição são arquivos com valores separados por vírgula (do inglês *comma-separated-values*)

3.3 CÓDIGO DA SIMULAÇÃO ESTRUTURAL

O código de cálculos é responsável por montar as matrizes da Equação 2.7 e da Equação 2.14 com os valores recebidos. Com toda a formulação montada é feito a solução da equação diferencial ordinária (EDO) com a função `scipy.integrate.odeint` (abreviado para `odeint`), uma função de soluções para equações diferenciais que se utiliza do método de Runge-Kutta.

Nesta etapa também ocorre o cálculo das frequências de excitação de cada grau de liberdade, utilizando a função `numpy.fft.fftn`, função responsável por calcular a TDF para várias variáveis em simultâneo (NUMPY.FFT.FFTN, 2022). Por fim, têm-se os processamentos estatísticos, todos feitos com funções prontas no pacote estatístico da biblioteca `numpy` (STATISTICS, 2022).

Após a simulação os dados não tratados são agrupados em um arquivo *CSV* com cada linha representando um estado de tempo e cada coluna uma variável do grau de liberdade. Este arquivo é então guardado na pasta de Data Lake. Ele pode ser visto no Apêndice A.2.

3.3.1 Processamento Estatístico

O processamento estatístico aplicado nos dados da simulação foram feitos para obter o maior número de dados relevantes possíveis. A escolha de quais são relevantes foi obtido na literatura de inteligência artificial (FACELI et al., 2021). Na literatura é afirmado que a estatística descritiva resume quantitativamente as principais características de um certo conjunto de dados.

Após cada simulação para cada variação do parâmetro são obtidos todos os valores estatístico proposto no subcapítulo 2.3.5 para todos os graus de liberdade. Então para cada grau de liberdade tem-se 11 dados tratados obtidos, subdivididos em valores diretos da vibração e dos dados da frequência, todos os dados de estudo podem ser vistos na Tabela 1.

Com isso, o banco de dados final, para os 5 graus de liberdade de estudo, terá 55 colunas no total, ou seja, 11 dados.

3.3.2 Integração das EDO (*odeint*)

Para utilizar esta função são necessários a função que rege a EDO ($f(x)$ que para este sistema é a Equação 2.14), os chutes iniciais de cada grau de liberdade (q_0) e o espaço de tempo de estudo (t) (SCIPY.INTEGRATE.ODEINT, 2009).

Tabela 1 – Dados obtidos da simulação.

| Dados estatísticos das vibrações | Dados estatísticos das amplitudes |
|----------------------------------|-----------------------------------|
| Intervalo | Máxima |
| Média | Onde ocorre a amplitude máxima |
| Desvio padrão | Média |
| Obliquidade | Desvio padrão |
| Curtose | Obliquidade |
| | Curtose |

Fonte: Autor.

3.4 CÓDIGO PARA SALVAR GRÁFICOS

Este código é responsável por receber os dados tratados em formato CSV, lê-los utilizando a biblioteca Pandas e plotar gráficos destes em função do valor variado utilizando a biblioteca *matplotlib* e após a plotagem salvar estes como imagens. Estes gráficos serão em relação ao primeiro valor obtido, assim, obtém-se um gráfico em função da porcentagem de variação. Ele pode ser visto no Apêndice A.3.

4 RESULTADOS NUMÉRICOS

Este capítulo será separado em três partes. Primeiramente será feito uma validação do código de simulação, construindo um gráfico conhecido utilizando os dados de um exercício do livro de estudo. Depois será feito um estudo de caso, demonstrando os dados de entrada da simulação de estudo e analisado um caso da variação bem como apresentado os resultados e feita uma análise de cada parâmetros estatístico. Por fim será demonstrado o banco de dados e as pastas resultantes da variação do parâmetro mecânico.

A análise de cada parâmetro será feita na forma de análise de gráficos, estes serão plotados para se poder visualizar qual dos graus de liberdade foi o mais sensível a mudanças mecânicas do mancal. Esta análise será feita quantitativamente, visto que todos os dados estão no controle da simulação, tendo-se assim, um ambiente de testes completamente controlado.

4.1 VALIDAÇÃO DO MODELO

Para a validação do método proposto, foi desenvolvido uma comparação de gráficos da magnitude do raio de giro em função de diferentes rotações com os dados propostos na tabela 2 obtidos do exercício 6.2.1 de FRISWELL et al. (2010). Para o desenvolvimento deste gráfico foi utilizado o sistema de 4 graus de liberdade, abandonando assim o parâmetro θ_p . Utilizado o código para simular o motor por 10 segundo e variando a velocidade de 0 até 3500 RPM a um passo de 10 RPM e calculado o raio de giro máximo para cada caso como a raiz dos quadrados de u e v .

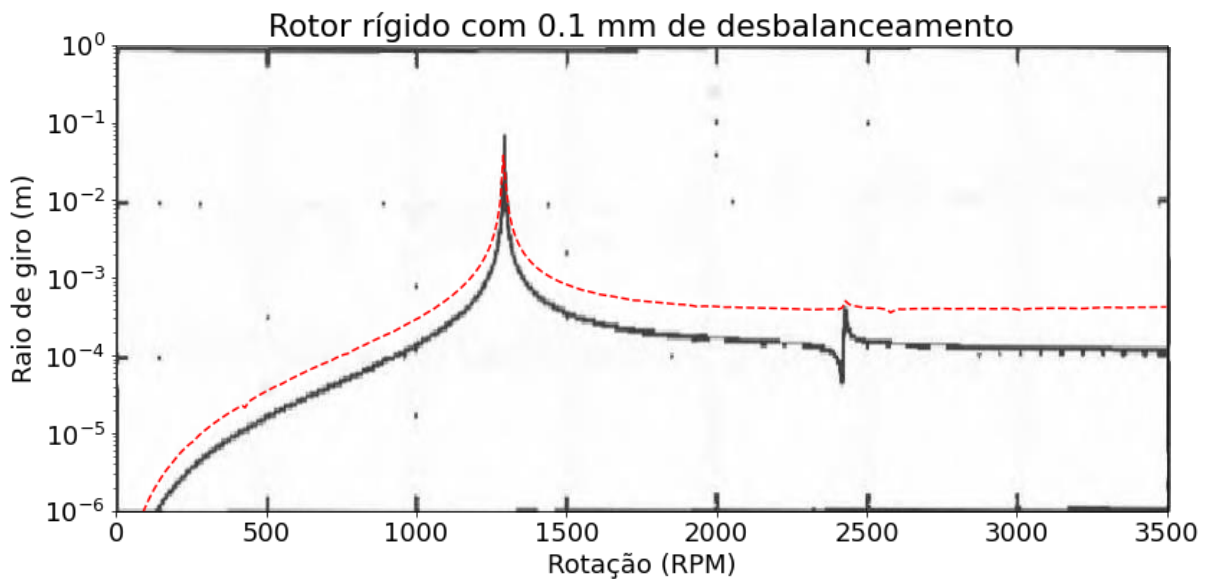
Tabela 2 – Valores da simulação de validação.

| | | | | | |
|-------|-------------------------|-------|----------|------------|---------|
| r | 0,1 m | a | 0,25 m | c_1 | 10 Ns/m |
| m | 122,68 kg | b | 0,25 m | c_2 | 13 Ns/m |
| I_p | 0,6134 kgm ² | k_1 | 1 MN/m | ϵ | 0,1 mm |
| I_d | 2,8625 kgm ² | k_2 | 1,3 MN/m | | |

Fonte: (FRISWELL et al., 2010)

O resultado pode ser visto na figura 15.

Figura 15 – Amplitude de resposta do sistema em vibração, resultado analítico obtido do livro FRISWELL et al. (2010) em preto, resultado obtido na simulação computacional em vermelho



Fonte: FRISWELL et al. (2010), Autor.

Após a visualização do gráfico gerado, é possível notar a semelhança entre esse e o gráfico gerado utilizando métodos analíticos, como feito no livro FRISWELL et al. (2010). O pico ocorrido em aproximadamente 1250 rotações por minuto é notado na simulação, confirmando o funcionamento do sistema desenvolvido. Os gráficos não são iguais devido a diferenças entre uma simulação dinâmica e um método analítico com número complexos, além de perdas devido ao passo de velocidade bem como a erros de arredondamento devido ao método de equação diferencial utilizado.

4.2 ESTUDO DE CASO

Considerando que a maior quantidade de erros acontecem em rolamentos (IMMOVILLI et al., 2013), o parâmetro variado para o estudo de caso foi o coeficiente de rigidez do rolamento à esquerda (k_1). A simulação ocorrerá por 10 segundos com um passo de tempo de 0,01 segundos e com a variação de k_1 de 90% à 110% do valor nominal.

O modelo estabelecido para o estudo de caso foi o mesmo para a validação do código, visto que os dados mecânicos de mancais não são usuais de se encontrar em documentação. Além dos dados mecânicos do motor foram feitas considerações para a obtenção de um sistema mais realístico, estes sendo:

- Potência e torque elétrico de um motor real com massa igual ao de estudo;
- Velocidade dinâmica;
- Propriedade variável no torque mecânico;
- Pior caso de desbalanceamento e desalinhamento conforme a norma brasileira.

4.2.1 Potência, Torque Eletromagnético e Velocidade Dinâmica

Para a obtenção da potência e torque nominal do motor de estudo foram obtidos do catálogo de um motor da empresa WEG, o W22 IR3 Premium 20CV (W22 IR3 PREMIUM 20 CV 4P 160M 3F 220/380/440 V 60 HZ IC411 - TFVE - B3D, 2017). Como a velocidade depende do torque eletromagnético é possível que obter um valor dinâmico da velocidade (θ_p) utilizando a equação 4.1 para calcular o torque eletromagnético.

$$T_e = \frac{P}{\dot{\theta}_p} \quad (4.1)$$

Assim, com o decorrer do motor, com a velocidade aumentando, terá uma variação no torque eletromagnético, obtendo-se assim um parâmetro variável da velocidade com o tempo de simulação.

4.2.2 Propriedade Variável no Torque Mecânico

Em motores reais o torque mecânico aplicado na entrada não é constante, existe bastante variação derivado de irregularidades e outros fatores externos. No caso de estudo todos os parâmetros são controlados, então para observar como uma variação no torque de entrada interage com os graus de liberdade foi introduzido uma variação no torque mecânico.

A variação foi com uma função senoidal para poder ver a amplitude dessa variação na velocidade na frequência dos graus de liberdade. Para o estudo de caso foi posto uma variação de amplitude (A_{mp}) de 5 Nm e frequência (f) de 10 Hz, fazendo com que o torque de entrada fica igual a:

$$T_m = T_m - A_{mp} \sin(f2\pi t) \quad (4.2)$$

4.2.3 Variáveis de Desbalanceamento e Desalinhamento

Para observar a influência do desbalanceamento de massa foi utilizada a norma ISO 1940–1, que rege limites de qualidade de desbalanceamento ($\epsilon\Omega$) que, para o caso de estudo deve ser igual a $2,5 \text{ mm/s}$ (ISO, 2003). Calculando para o estudo de caso, tem-se um limite de desbalanceamento igual a $8,3 \times 10^{-3} \text{ mm}$.

Na análise da influência de desalinhamento de eixo radialmente do rotor, foi utilizado valores limites do livro EASY-LASER (2008). Valores para diferentes velocidades de rotação podem ser vistos na Fig. 16. Para o caso de estudo, com velocidade em torno de 187 rad/s , tem-se um limite angular β de $2,6 \times 10^{-6} \text{ rad}$.

Figura 16 – Valores de desalinhamento de eixo para o estudo de caso.

| rpm | Excelente | | Aceitável | |
|-----------|--------------|-----------|--------------|-----------|
| | milésimos/'' | mm/100 mm | milésimos/'' | mm/100 mm |
| 0000-1000 | 0,6 | 0,06 | 1,0 | 0,10 |
| 1000-2000 | 0,5 | 0,05 | 0,8 | 0,08 |
| 2000-3000 | 0,4 | 0,04 | 0,7 | 0,07 |
| 3000-4000 | 0,3 | 0,03 | 0,6 | 0,06 |
| 4000-5000 | 0,2 | 0,02 | 0,5 | 0,05 |
| 5000-6000 | 0,1 | 0,01 | 0,4 | 0,04 |

Fonte: EASY-LASER (2008).

4.2.4 Simulação com os dados Nominais

Com todas as considerações apresentadas até agora, tem-se todos os dados de entrada necessários para fazer a simulação na tabela 3

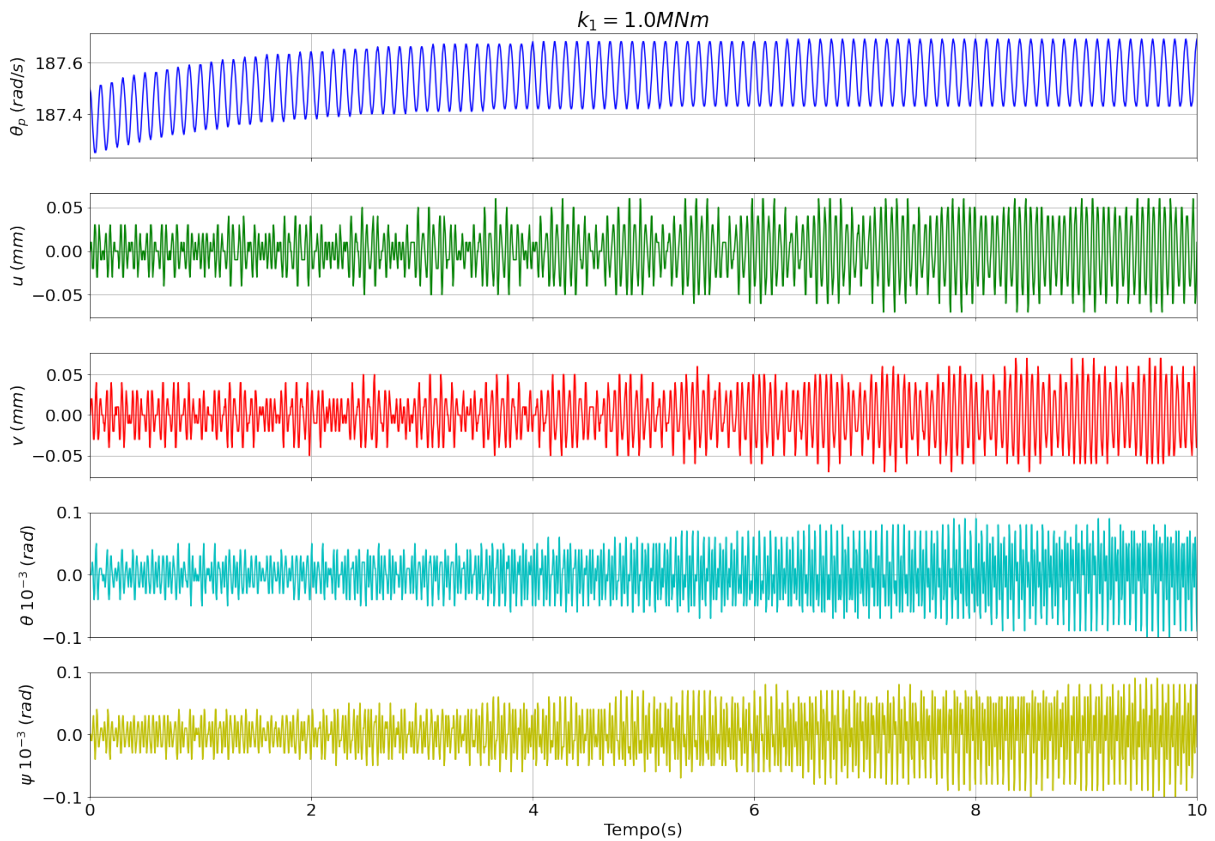
Tabela 3 – Valores de entrada da simulação.

| | | | | | | | |
|----------|-------------------------|-------|-----------|---------|----------------------------------|----------------|---------------------------------|
| P_{ot} | 15 kW | k_1 | 1 MN/m | I_d | 2,8625 kgm ² | ϵ | $8,3 \times 10^{-3} \text{ mm}$ |
| T_e | 80 Nm | k_2 | 1,3 MN/m | I_p | 0,6134 kgm ² | t | 10 s |
| T_m | 79,6 Nm | c_1 | 10 | a | 0,25 m | dt | 0,01 s |
| J | 0,6134 kgm ² | c_2 | 13 | b | 0,25 m | $Inicio_{var}$ | 90% |
| B_n | 0,002 | m | 122,68 kg | β | $2,6 \times 10^{-6} \text{ rad}$ | $Final_{var}$ | 110% |

Fonte: Autor.

Primeiro será analisado a simulação em sua forma nominal, com os dados em 100% de seu valor. Este gráfico pode ser visto na Figura 17.

Figura 17 – Resultados da simulação com todos os dados em 100% do seu valor.



Fonte: Autor.

É notável que ao decorrer do tempo o motor tende a acelerar, visto que o torque de carga atuando na máquina é menor que o torque produzido por ela. A variação na velocidade é devido à variação no torque com a função senoidal apresentada anteriormente.

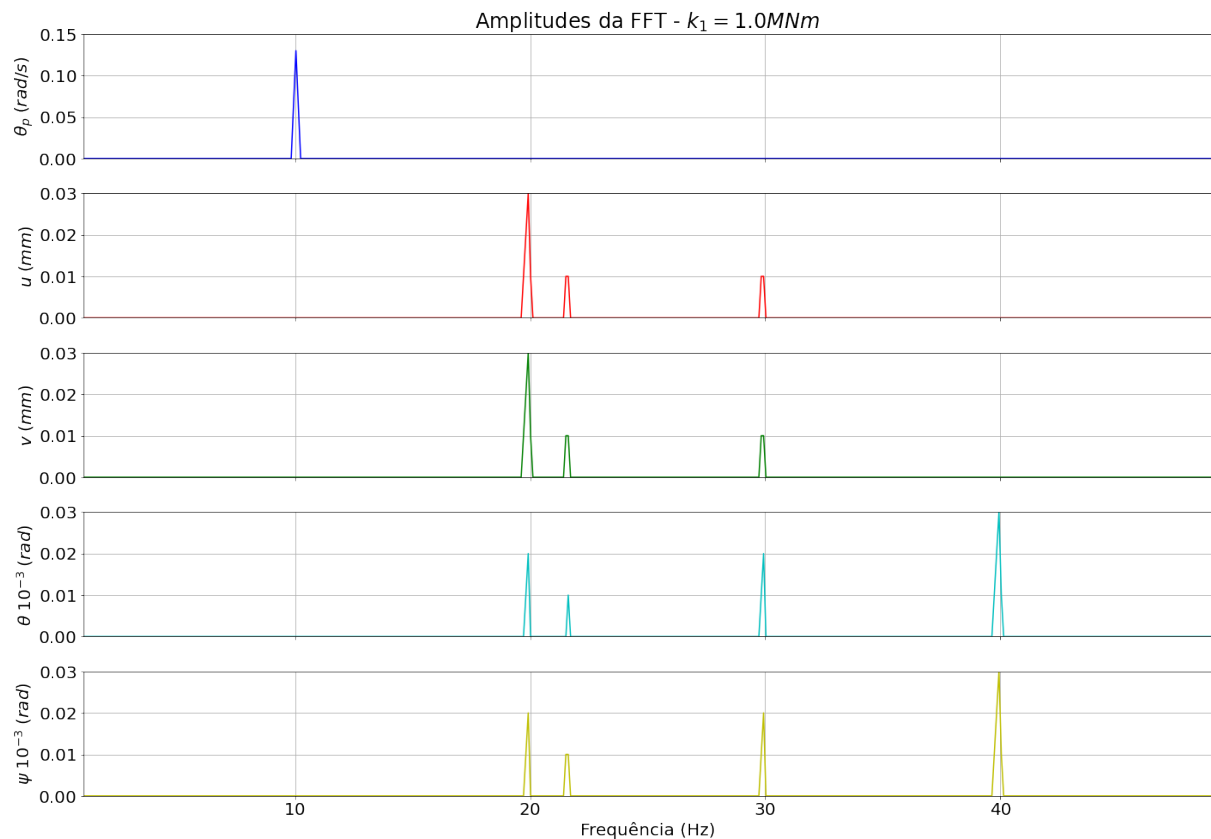
A vibração nos graus de liberdade aparentam ser aleatórias e caóticas, mas é possível ver que com o decorrer do tempo existe certos padrões observáveis no sinal, podendo assim assumir que este gráfico é determinístico e periódico. É possível notar que as 4 variáveis de espaço do motor tem uma forma espelhada entre si. As variáveis u e v são as de movimento na direção x e y e, após o início aleatório, é possível enxergar que certos padrões de batimento aparecerem.

Nas variáveis de ângulo ψ e θ é possível ver um espelhamento de um para o outro, ou seja, quando um está girando para um lado, o outro está para o outro. O formato do gráfico é parecido com o das variáveis u e v , após o início caótico, é notável o aparecimento de padrões

repetitivos, podendo ser categorizado como determinístico e periódico.

Nos gráficos de frequência, é possível obter mais informações sobre os resultados.

Figura 18 – Resultado da frequência da simulação com todos os dados em 100% do seu valor.



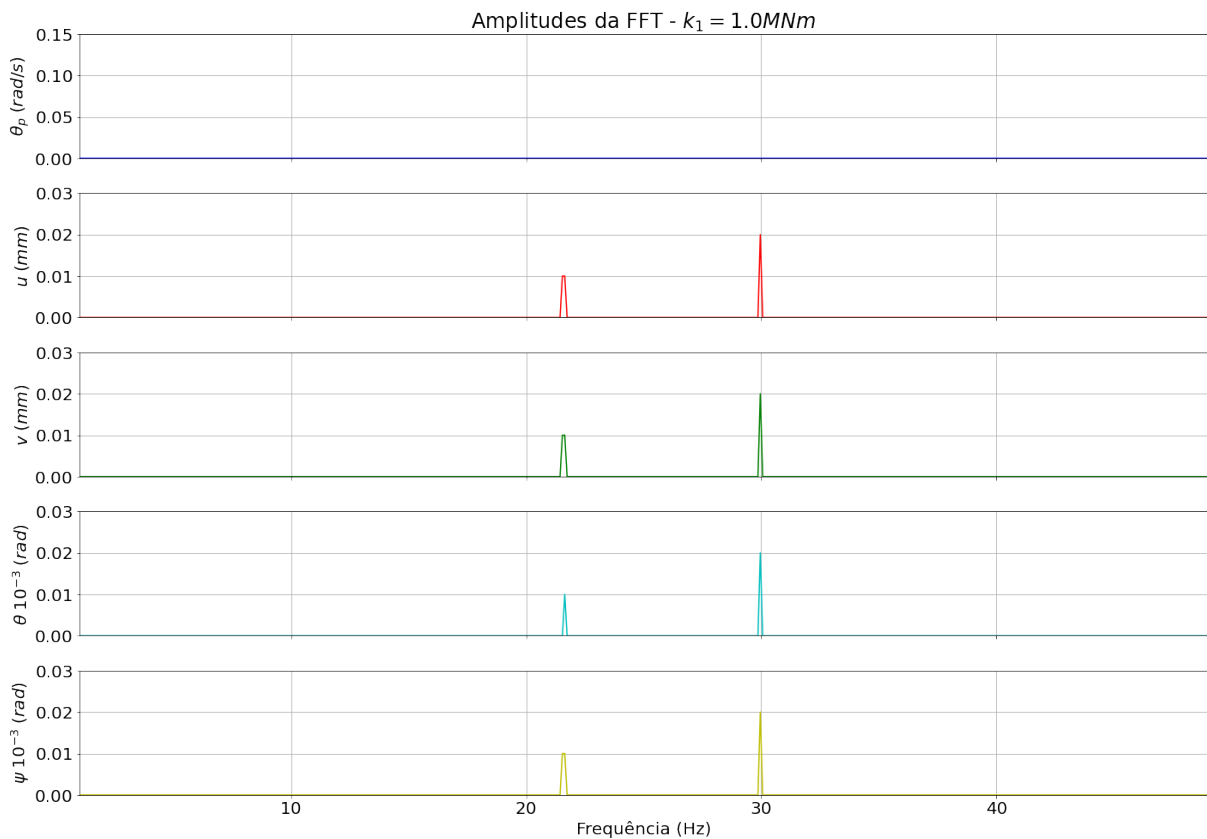
Fonte: Autor.

No gráfico das frequências (Figura 18) é possível ver as frequências que dominam os sinais gerados. Como esperado na velocidade θ_p tem-se a predominância da frequência de 10 Hz oriundo da variação senoidal proposto anteriormente.

Nas variáveis u e v é possível notar o espelhamento das variáveis, as duas possuem o mesmo domínio na frequência, com a frequência natural sendo em 20 Hz . Para as variáveis de giro ψ e θ , é notável a mesma interação de espelhamento entre as variáveis, porém a frequência natural é de 40 Hz . Com este gráfico é confirmado que o sistema não é periódico, visto que não tem as frequências fundamentais sendo fatores de uma frequência dominante, tendo-se assim, um sistema não periódico ou quase periódico.

Simulando o motor sem a variação do torque apresentado na Equação 4.2, é possível comparar o gráfico da FFT desse (Figura 19) com a Figura 18 para observar o acoplamento da velocidade com a vibração.

Figura 19 – Resultado da frequência da simulação com todos os dados em 100% do seu valor e sem variação do torque.



Fonte: Autor.

É notável que com a implementação de uma variação do torque é possível ver que se tem maiores frequências influenciando a vibração do sistema. Com isto, é confirmado o acoplamento da Equação 2.15 com a Equação 2.17. Além disso, sem a variação do torque de carga, o sistema se torna periódico com frequências fundamentais de 23 Hz e 30 Hz .

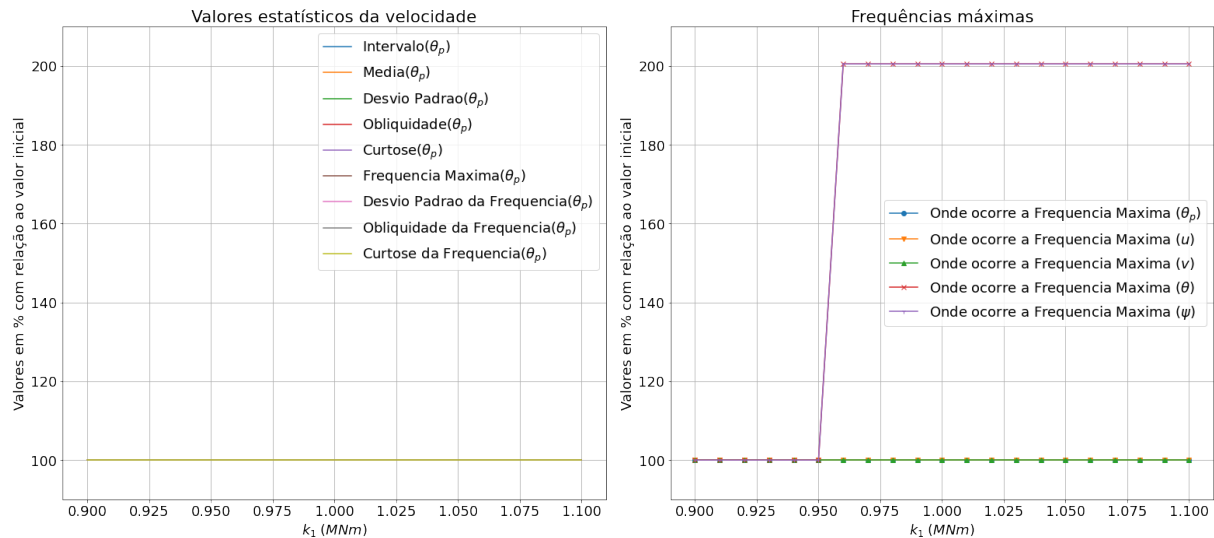
4.2.5 Variáveis sem influência da variação

Após a simulação foi observado que alguns parâmetros não foram influenciados pela variação do coeficiente elástico do rolamento à direita. A velocidade de rotação θ_p não variou com o rolamento isto ocorreu, pois, como visto na modelagem dinâmica, a velocidade não é influenciada pela vibração ou parâmetros físicos dos rolamentos, porém a vibração é acoplada com a velocidade, como visto na Figura 19.

Outro parâmetro que não foi observado variação significativa ou proporcional à mudança do coeficiente elástico foi a frequência onde ocorre a maior amplitude de vibração. Estes dois

parâmetros podem ser vistos na Figura 20.

Figura 20 – Variáveis que não tiveram variação significativa



Fonte: Autor.

Foi notável que apenas θ e ψ que variaram para o dobro de seu valor inicial, porém não foi uma variação significativa para se estudar mais a fundo.

4.2.6 Análise dos Resultados Estatísticos

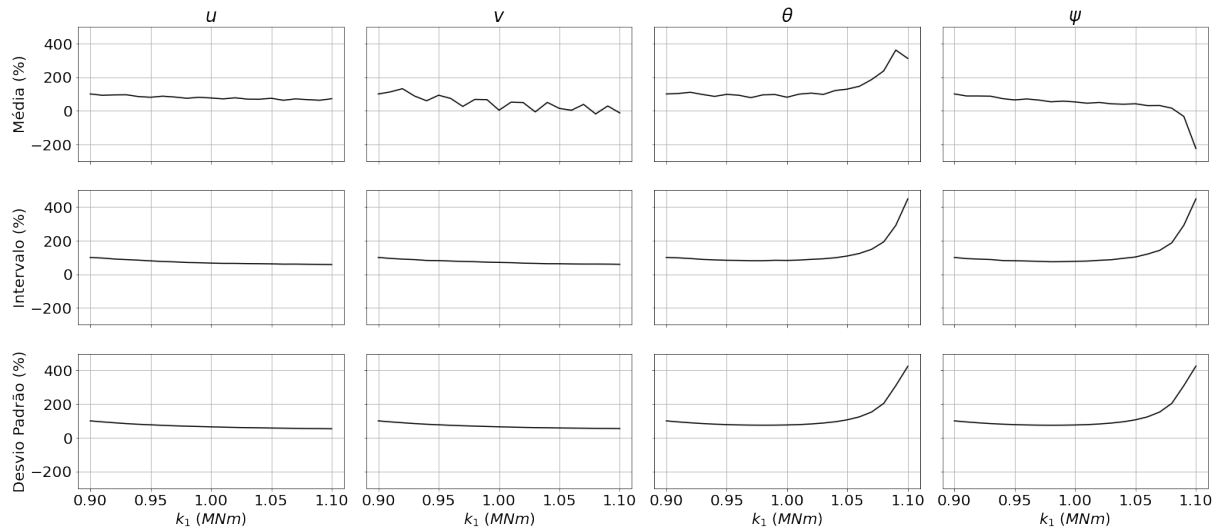
A análise dos resultados será subdividido entre os parâmetros que delimitam a localidade ou tendência central dos valores e espalhamento ou dispersão. Após será analisado os parâmetros que determinam a distribuição dos dados gerados. Todos os gráficos foram feitos utilizando o valor em porcentagem com relação ao valor inicial de simulação, ou seja, com o 100% sendo o valor obtido em k_1 igual à 90% do seu valor nominal.

4.2.6.1 Média, Intervalo e Desvio Padrão

Os resultados da média, intervalo e desvio padrão dos 4 graus de liberdade que foi observado variação podem ser vistos na Figura 21

As variáveis u e v obtiveram pouca variação da média, intervalo e desvio padrão comparando com θ e ψ . θ teve sua média aumentada em até 400% do valor inicial, mostrando que com o aumento da constante elástica do mancal de rolamento à esquerda θ tende a ter seu valor central maior, passando maior parte do tempo mais longe da posição estacionária. ψ tem

Figura 21 – Variação da média, intervalo e desvio padrão de u , v , θ e ψ



Fonte: Autor.

sua média diminuindo em até 300% do valor inicial, indo para o valor negativo comparado ao inicial, assim, ψ tende a aumentar a média, porém seu valor tende a ficar mais longe do ponto estacionário, mas para a direção contrária.

O intervalo mostra a diferença entre o maior e menor valor de uma variável, θ e ψ tem seus intervalos aumentados em até 400% do valor inicial, assim, estas variáveis tendem a cada vez ficarem com maior oscilação com o aumento do coeficiente de elástico. Este aumento aparenta ser exponencialmente, ou seja, a diferença entre o maior valor e menor valor destas variáveis tendem a subir muito com um pouco aumento da variável elástica.

O desvio padrão representa o quanto as variáveis estão espalhadas com relação à média, θ e ψ tem seus valores aumentados exponencialmente com o aumento do coeficiente elástico, ou seja, quanto pior está o rolamento, mais caótico ficam estas variáveis, visto que a vibração tende a fugir bastante da média.

As variáveis u e v tem pouca variação comparado com θ e ψ , porém, ainda é observado bastante variação, de até 100% do valor inicial. Todos os parâmetros estatísticos tenderam a diminuir, ou seja, a vibração caiu com o aumento do coeficiente elástico do mancal de rolamento, o que é esperado, visto que quando maior o coeficiente elástico, mais força é necessário para variar a posição do mancal.

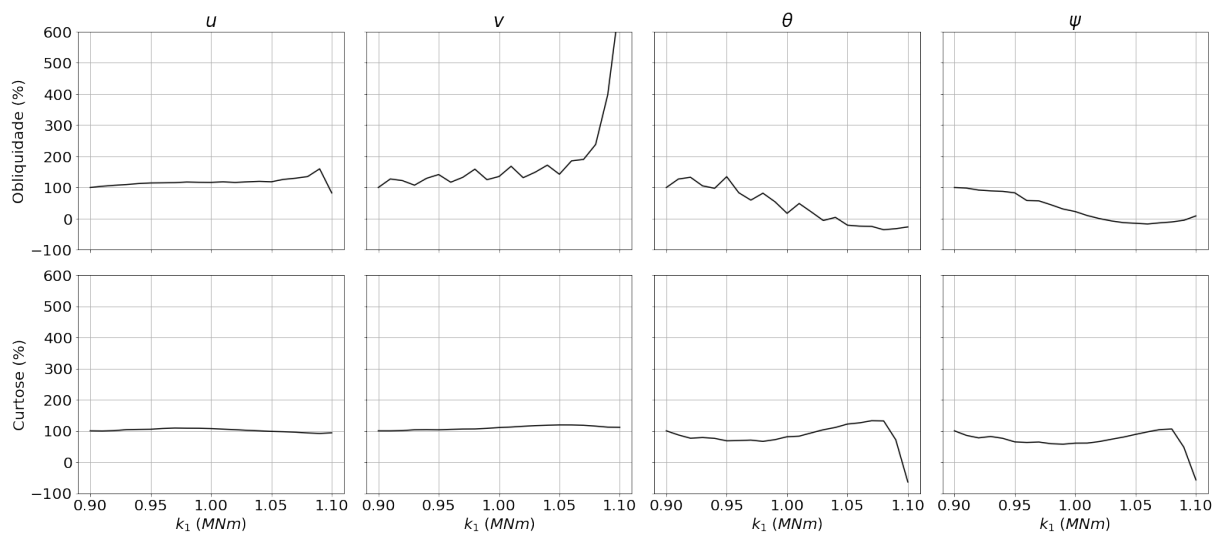
Mesmo com o aumento da constante elástica aumentando que faria o sistema mais rígido, as variáveis θ e ψ tenderam a aumentar, isto se deve ao fato que apenas um lado do sistema

tendeu a ficar mais rígido e o outro se manteve constante, obtendo-se assim ângulos de giro do rotor mais predominante.

4.2.6.2 Obliquidade e Curtose

Para melhor entendimento é necessário saber o valor inicial da obliquidade e curtose, visto que o significado destes depende se é maior, menor ou igual a zero. Os valores iniciais podem ser vistos na Tabela 4. Os resultados da obliquidade e curtose dos 4 graus de liberdade que foi observado variação podem ser vistos na Figura 22

Figura 22 – Variação da obliquidade e curtose de u , v , θ e ψ



Fonte: Autor.

Tabela 4 – Valores iniciais de obliquidade e curtose.

| | u | v | θ | ψ |
|-------------|-------------------------|--------------------------|-------------------------|-------------------------|
| Obliquidade | $-38,4 \times 10^{-3}$ | $8,6 \times 10^{-3}$ | $-43,1 \times 10^{-3}$ | $-192,1 \times 10^{-3}$ |
| Curtose | $-609,2 \times 10^{-3}$ | $-622,76 \times 10^{-3}$ | $-433,3 \times 10^{-3}$ | $-400,1 \times 10^{-3}$ |

Fonte: Autor

Com os dados apresentados, é notável que todas as variáveis estão bem próximas à distribuição simétrica normal, visto que todas as obliquidades estão próximas de 0, com ψ mais longe. Com a variação de k_1 , tem-se uma pequena variação na simetria de u , uma mudança alta de simetria de θ e ψ , caindo para mais perto de 0, obtendo cada vez uma distribuição mais simétrica. Na variável v é notável um aumento brusco do valor, considerando que começou

perto de zero, significa que com o aumento de k_1 a variável está fugindo de um gráfico simétrico.

Com os valores iniciais apresentados de curtose, é notável que todas as variáveis começaram com um histograma mais achatado que a distribuição normal. Com o aumento de k_1 é notável que u e v não sofreram alterações significativas nos valores. θ e ψ não sofreram alterações consideráveis do valor, porém na última iteração é possível ver que o valor sofreu uma brusca variação, indo de negativo para positivo, sofrendo uma mudança de um histograma achatado para um histograma com uma distribuição mais alta e concentrada que a distribuição normal.

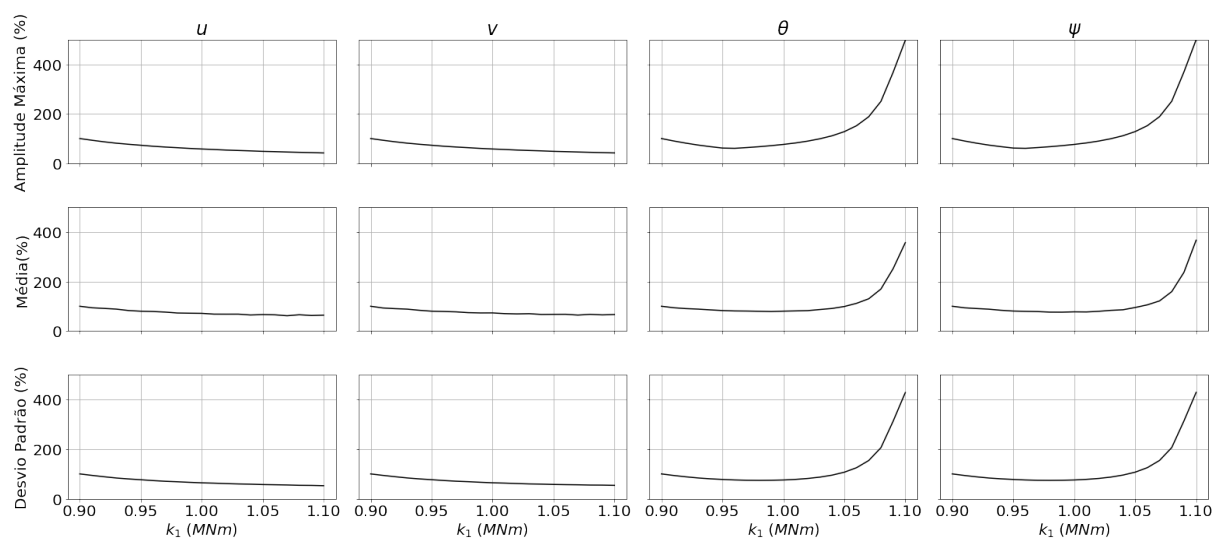
4.2.7 Análise dos Resultados Estatísticos das FFTs

A análise dos resultados será subdividido entre os parâmetros que delimitam a localidade ou tendência central dos valores e espalhamento ou dispersão. Após será analisado os parâmetros que determinam a distribuição dos dados gerados.

4.2.7.1 Valor Máximo, Média e Desvio Padrão

Os resultados do valor máximo, média e desvio padrão das amplitudes das transformadas de Fourier dos 4 graus de liberdade que foi observado variação podem ser vistos na Figura 23

Figura 23 – Variação do valor máximo, média e desvio padrão das amplitudes das frequências de u , v , θ e ψ



Fonte: Autor.

O motivo de não ter sido utilizado o intervalo como parâmetro de localidade foi por,

como visto na Figura 18 é notável que o valor mínimo da amplitude de frequência é sempre zero, então o intervalo seria apenas o valor máximo observado.

A amplitude máxima das frequências representa o quanto da vibração vem da frequência natural de excitação do sistema. Com o aumento do coeficiente elástico do mancal de rolamento à esquerda, a amplitude máxima de u e v cai devido à diminuição da vibração nessas direções. Para as variáveis θ e ψ o valor da amplitude máxima aumenta até em torno de 500% do valor inicial de forma, aparentemente, exponencial com o aumento de k_1 , mostrando assim que estes parâmetros são sensíveis ao aumento do coeficiente elástico.

A média das amplitudes, assim como a média dos valores, representa um valor central dos dados. Para u e v é possível observar a tendência decrescente das médias, visto uma diminuição das vibrações. Para os valores de θ e ψ , é notável um aumento exponencial das médias em até em torno de 350% do valor inicial, tendo assim, um aumento significativo da vibração com o aumento de k_1 .

Assim como o desvio padrão dos valores estatísticos, o desvio padrão representa o quanto as variáveis estão espalhados com relação à média, θ e ψ tem seus valores aumentados exponencialmente com o aumento do coeficiente elástico, ou seja, quanto pior está o rolamento, mais caótico ficam estas variáveis, visto que a vibração tende a fugir bastante da média.

Assim como no Subcapítulo 4.2.6.1, em comparação às variáveis θ e ψ , u e v não representam grandes variáveis, porém, sozinhas é possível notar que decaem em até 70% do seu valor inicial. O decaimento destas variáveis é notável visto que um aumento do coeficiente elástico torna o sistema mais rígido, assim, mais vibração.

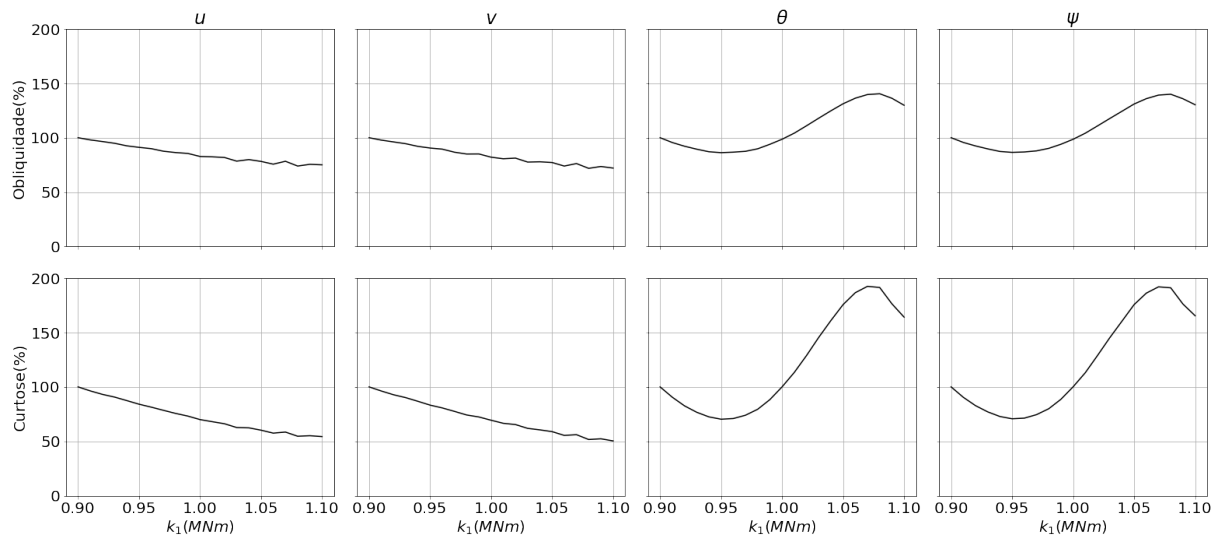
Assim como as variáveis dos resultados estatísticos, com o aumento da constante elástica aumentando que faria o sistema mais rígido, as variáveis θ e ψ tenderam a aumentar, isto se deve ao fato que apenas um lado do sistema tendeu a ficar mais rígido e o outro se manteve constante, obtendo-se assim ângulos de giro do rotor mais predominante.

4.2.7.2 *Obliquidade e Curtose*

Os resultados da obliquidade e curtose das amplitudes das transformadas de Fourier dos 4 graus de liberdade que foi observado variação podem ser vistos na Figura 24

Para melhor entendimento é necessário saber o valor inicial da obliquidade e curtose, visto que o significado destes depende se é maior, menor ou igual a zero. Os valores iniciais podem ser vistos na Tabela 5.

Figura 24 – Variação da obliquidade e curtose das amplitudes das frequências de u , v , θ e ψ



Fonte: Autor.

Tabela 5 – Valores iniciais de obliquidade e curtose das amplitudes das frequências.

| | u | v | θ | ψ |
|-------------|--------|--------|----------|--------|
| Obliquidade | 13,49 | 13,52 | 10,45 | 10,45 |
| Curtose | 213,86 | 214,71 | 131,21 | 130,88 |

Fonte: Autor.

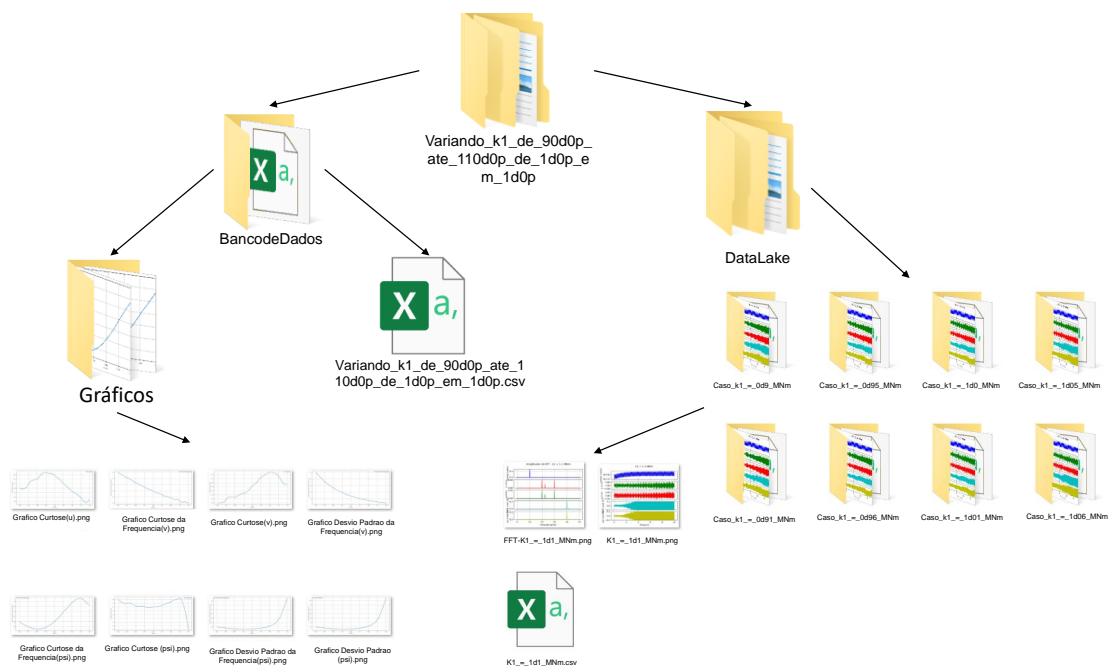
É notável o alto valor de obliquidade e curtose das amplitudes das frequências de excitação, isto se deve ao fato que a maioria das frequências possui uma amplitude de zero. A obliquidade sendo um valor muito alto mostra que a maioria dos dados se concentram à esquerda da média, neste caso, no valor zero. A curtose muito alta demonstra que se tem uma distribuição muito alta em comparação com a distribuição normal, novamente, isto se deve ao fato que a maioria dos valores são iguais a zero, assim, tem-se um pico alto onde tem-se amplitudes diferentes de zero.

Assim como os outros parâmetros estatísticos das amplitudes, é notável que u e v tem um decaimento com o aumento de k_1 e θ e ψ tem uma oscilação alta dos valores, porém, nunca chegam próximos de zero, tendo-se assim, sempre uma alta diferença entre o histograma observado e a distribuição normal. Assim, é possível notar que a distribuição dos valores de amplitudes de frequências fogem bastante da distribuição normal.

4.3 BANCO DE DADOS

O banco de dados foi salvo em pastas utilizando arquivos em formato de CSV, resultando em uma cascata de pastas como vista na Figura 25. Desta forma é possível reutilizar os dados gerados, tendo-se assim, um banco de dados estáticos para estudos futuros.

Figura 25 – Cascata das pastas criadas pelo programa desenvolvido.



Fonte: Autor.

Foi substituído o ponto de números reais, como em 1.10, pela letra "d" para evitar problemas de caracteres que podem ser confundidos com tipos de arquivos do sistema operacional Windows. O símbolo de porcentagem (%) foi substituído por "p" e o espaço por "_", pois algumas linguagens de programação utilizam estes símbolos para operações.

O arquivo "Variando k1 de 90.0% ate 110.0% de 1.0% em 1.0%" foi renomeado para "Variando_k1_de_90d0p_ate_110d0p_de_1d0p_em_1d0p" e os arquivos de cada caso foram renomeados de "Caso k1 = 0.96 MNm" para "Caso_k1_=_0d96_MNm".

Ao final do processo a pasta principal ocupa 70,1 MB de memória do computador, com os arquivos que mais ocupam espaço sendo as figuras geradas com um total de 66,3 MB, com o banco de dados "Variando_k1_de_90d0p_ate_110d0p_de_1d0p_em_1d0p.csv" ocupando 25 KB e o csv do data lake de cada caso ocupando 152 KB.

O tempo de processamento total da simulação foi de 429,30 segundos, ou em torno de 7

minutos, com o maior tempo sendo ocupado com a geração e arquivamento das imagens, com a geração dos gráficos finais de cada parâmetro variado ocupando um tempo de 181 segundos, ou em torno de 3 minutos. Removendo esta etapa do processo e bem como a geração de qualquer figura, o tempo de simulação fica em 38 segundos, obtendo-se assim, para 20 iterações de 10 segundos de simulação cada, para cada segundo de processamento é possível simular mais de 5,26 segundos.

A estrutura de arquivos proposta é de fácil entendimento para público externo do âmbito acadêmico, podendo assim, ser analisado por usuários sem experiência em Python e outras linguagens de programação de análise de dados. O programa desenvolvido também se mostrou versátil ao que se quer salvar no fim das iterações, tendo-se tempo de processamento menor, além de retirar dados que possam não ser relevantes.

5 CONCLUSÕES

A vibração produzida por imperfeições em máquinas rotativas é prejudicial para a vida útil desta, mas também contém muita informação referente a este equipamento. A utilização de métodos computacionais para análise da vibração de motores elétricos já é utilizado na indústria com métodos preditivos e utilização de AM bem como com a utilização de GD.

O método utilizado se mostrou eficaz para o desenvolvimento de banco de dados, bem como o estudo de variáveis estatísticas e sua sensibilidade ao funcionamento físico do motor. A simulação computacional se mostrou satisfatória quanto ao estudo das variáveis da máquina rotativa.

Após a análise do sistema de estudo, é possível notar que todas as variáveis são sensíveis à mudança de um parâmetro mecânico do mancal de rolamento. As variáveis θ e ψ se mostraram mais sensíveis devido ao fato que apenas um lado sofreu alteração tendo-se assim, maior giro do eixo.

É notável que a extração dos parâmetros estatísticos se mostraram eficazes em demonstrarem a variação de parâmetros mecânicos do rotor. As variáveis θ e ψ tiveram grande importância no estudo de qual dos dois mancais está sofrendo alterações físicas. Assim todas as variáveis de vibração se mostraram úteis no estudo de prognóstico de falha.

A utilização de simulação computacional para o estudo de máquinas rotativas se mostrou viável. Porém, é necessário um estudo de caso a caso quando se trata de máquinas reais. Assim, para se obter um sistema confiável de estudo é preciso obter todos os dados construtivos do motor de estudo.

Para o desenvolvimento de modelos preditivos o desenvolvimento de banco de dados de aprendizado é uma etapa crucial, visto que este é o que determinará o nível de confiabilidade das previsões feitas pelo sistema preditivo. Para o uso real de modelos de treinamento o estudo de caso usado neste trabalho não é suficiente para sua utilização em casos reais, visto que os dados utilizados foram obtidos de exercícios teóricos.

5.1 SUGESTÃO DE TRABALHOS FUTUROS

Foi visto neste trabalho que o uso de simulações computacionais para o desenvolvimento de banco de dados é viável. Porém, devido a limitações de dados construtivos disponíveis não

foi possível desenvolver sistemas que possam ser utilizados em casos reais. Desta forma ficam sugestões para futuros trabalhos que forem ser realizados nesta área:

- Utilizar dados de um motor de estudo real;
- Comparar os resultados obtidos na simulação com resultados obtidos de sensores em motores reais;
- Com os dados realísticos, fazer uma análise de relevância estatística;
- Emprego de Monte Carlo para avaliar influências dos parâmetros construtivos;
- Utilizar janelamento de sinais para casos realísticos para obter resultados mais "limpo".

REFERÊNCIAS

- ABDULBARY, M.; EMBABY, A.; GOMAA, F. Fault Diagnosis in Rotating Systems Based on Vibration Analysis. **Engineering Research Journal**, [S.l.], v.44, p.285–294, 07 2021.
- BEVELACQUA, P. J. Fourier Transform. **TheFourierTransform**, 2010. Disponível em: <<https://www.thefouriertransform.com>>. Acesso em 01 fev. 2022.
- BOUDINAR, A. et al. Induction motor bearing fault analysis using Root-MUSIC method. **IEEE Transactions on Industry Applications**, [S.l.], v.52, p.1–1, 06 2016.
- BUTCHER, J. A history of Runge-Kutta methods. **Applied Numerical Mathematics**, [S.l.], v.20, n.3, p.247–260, 1996.
- COCHRAN, W. et al. What is the fast Fourier transform? **Proceedings of the IEEE**, [S.l.], v.55, n.10, p.1664–1674, 1967.
- CRAVO, E. Equipamentos industriais: o que são, tipos e como escolhe. **Kalatec**, junho 2019. Disponível em: <<https://blog.kalatec.com.br/equipamentos-industriais/>>. Acesso em 11 ago. 2022.
- CÂMARA, H. T. **A contribution to three phase induction motor control without mechanical sensors**. 2007. Universidade Federal de Santa Maria, Santa Maria.
- EASY-LASER. **MANUAL E540 05-0751**. [S.l.: s.n.], 2008.
- ENSUS ENGENHARIA. **Análise de Vibração Tipos de Sinais Transformada de Fourier e PSD**, 2016. Disponível em: <<https://ensus.com.br/analise-de-vibracao-tipos-de-sinais-transformada-de-fourier-e-psd/>>. Acesso em 01 fev. 2022.
- ESTATISTICS. **NumPy**. Disponível em: <numpy.org/doc/stable/reference/routines.statistics>. Acesso em 07 fev. 2022.
- FACELI, K. et al. **Inteligência Artificial: uma abordagem de aprendizado de máquina**. 2.ed. Rio de Janeiro, RJ, BR: LTC, 2021.
- FERNANDES, I. C. B. **Desenvolvimento e implementação de linguagem de programação em Python para parametrizar e monitorar um controle de velocidade com cicloconversor de frequência**. 2021. Universidade Estadual Paulista - Unes).

FONTANA Éliton. **Introdução aos Algoritmos de Aprendizagem Supervisionada**. [S.l.: s.n.], 2020.

FRISWELL, M. I. et al. **Dynamics of Rotating Machines**. [S.l.]: Cambridge University Press, 2010. (Cambridge Aerospace Series).

GALVAO, L. C. F. **SIMULAÇÃO NUMÉRICA DE UM ROTOR DE JEFFCOTT COM CONTATO ROTOR-ESTATOR**. 2017. Universidade Federal do Rio de Janeiro.

GORDO, N.; FERREIRA, J. **Elementos de Máquinas**. [S.l.]: Escola SENAI - SP, 2000.

HAAG, S.; ANDERL, R. Digital twin – Proof of concept. **Manufacturing Letters**, [S.l.], v.15, p.64–66, 2018. Industry 4.0 and Smart Manufacturing.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, [S.l.], v.585, n.7825, p.357–362, Sept. 2020.

IMMOVILLI, F. et al. Bearing Fault Model for Induction Motor With Externally Induced Vibration. **IEEE Transactions on Industrial Electronics**, [S.l.], v.60, p.3408–3418, 2013.

INDÚSTRIA 4.0: entenda seus conceitos e fundamentos. **Portal da Indústria**. Disponível em: <<https://www.portaldaindustria.com.br/industria-de-a-z/industria-4-0/>>. Acesso em 07 jun. 2022.

ISO. ISO 1940-1: mechanical vibration - balance quality requirements for rotors in a constant (rigid) state - part 1: specification and verification of balance tolerances. **International Standard**, [S.l.], 2003.

LIAO, L.; KÖTTIG, F. Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. **IEEE Transactions on Reliability**, [S.l.], v.63, n.1, p.191–207, 2014.

MILOSLAVSKAYA, N.; TOLSTOY, A. Big Data, Fast Data and Data Lake Concepts. **Procedia Computer Science**, [S.l.], v.88, p.300–305, 2016. 7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016, held July 16 to July 19, 2016 in New York City, NY, USA.

NUMPY.FFT.FFTN. **NumPy**. Disponível em: <shorturl.at/eIEU6>. Acesso em 07 fev. 2022.

PINTO, A. K.; XAVIER, J. A. N. **Manutenção: função estratégica**. 2.ed. 20921-400, Rio de Janeiro, RJ: Qualitymark Editora Ltda, 2001.

PORTELA, K. M. **Desenvolvimento e aplicação de uma biblioteca Python para modelagem, análise e simulação de aeronaves**. 2022. Universidade Federal de Santa Maria - UFSM.

RAO, S. **Vibrações mecânicas**. [S.l.]: Pearson Prentice Hall, 2009.

SCIPY.INTEGRATE.ODEINT. **scipy**. Disponível em: <<https://docs.scipy.org/doc/scipy-0.7.x/reference/generated/scipy.integrate.odeint.html>>. Acesso em 07 fev. 2022.

SHIN, K.; HAMMOND, J. **Fundamentals of signal processing for sound and vibration engineers**. [S.l.]: John Wiley & Sons, 2008.

W22 IR3 Premium 20 cv 4P 160M 3F 220/380/440 V 60 Hz IC411 - TFVE - B3D. **WEG**. Disponível em: <<https://www.weg.net/catalog/weg/BR/pt/Motores-Elétricos/Trifásico—Baixa-Tensão/Usos-Gerais/W22/W22-/W22-IR3-Premium-20-cv-4P-160M-3F-220-380-440-V-60-Hz-IC411—TFVE—B3D/p/11405420>>. Acesso em 03 abr. 2022.

WEG. Motores eficientes geram economia de energia na indústria. **Valor.Globo**, dezembro 2019. Disponível em: <<https://valor.globo.com/patrocinado/weg/weg/noticia/2019/12/02/motores-eficientes-geram-economia-de-energia-na-industria.ghtml>>. Acesso em 13 dez. 2021.

APÊNDICES

APÊNDICE A – Códigos

A.1 AGRUPAMENTO

```

1 import os
2 import time
3 from numpy import array, vstack, savetxt, concatenate
4 from Calculo import Calculos
5 from SalvarGraficos import salvarfig
6
7 #-----
8 #PARÂMETROS DE VARIAÇÃO
9 #-----
10
11 Pot = 15e3 #Potência do Motor (W)
12 Te = 80 #Torque eletromagnético (Nm)
13 Tm = 79.6 #Torque de Carga (Nm)
14 J = 0.6134 #Momento de inércia (kgm2) motor da weg = 0.0586
15 Bn = 0.002 #Coeficiente de atrito
16 k1 = 1e6 #Coeficiente elástico do mancal à esquerda
17 k2 = 1.3e6 #Coeficiente elástico do mancal à direita
18 c1 = 10 #Amortecimento no mancal à esquerda
19 c2 = 13 #Amortecimento no mancal à direita
20 m = 122.68 #Massa do motor
21 Id = 2.8625 #momento de inércia do eixo central
22 Ip = 0.6134 #momento de inércia polar do rotor
23 a = 0.25
24 b = 0.25
25 beta = 2.6e-6 #Ângulo de desbalanceamento em rad
26 eps = 8.3e-6 #Distância do desbalanceamento de massa
27 t = 10 #tempo de simulação
28 dt = 0.01 #passo de tempo
29 Amp = 5.0 #amplitude da função seno do Torque de Carga
30 freq = 10.0 #Frequencia do seno que varia o torque de carga (Hz), só
    funciona se Tem for cte
31
32 Inicio = 0.9 #inicio em porcentagem
33 Final = 1.1 #final em porcentagem
34 Passo = 0.01 #passo em porcentagem
35
36 ql_val = k1 #Determina qual valor vai ser considerado o 100% do valor de
    velocidades, aceleração, etc.
37
38 Nome = "Variando k1 de {}% ate {}% de {}% em {}%".format(round(Inicio
    *100,2), round(Final*100,2), round(Passo*100,2), round(Passo*100,2)) #Cria
    ou abre a pasta que o código ja criou com o nome que quiser (CUIDADO, SE
    A PASTA JA EXISTE VAI SOBREPOR TUDO QUE FOR DE VALOR IGUAL!)
39
40 FazerImg = True #faz os graficos de todos os dados tratados ao fim
41
42 #-----
43 #-----
44
45 if __name__ == "__main__":
46     start_time = time.time()

```

```

47 Nome = Nome.replace(" ", "_")
48 Nome = Nome.replace(".", "d")
49 Nome = Nome.replace("%", "p")
50
51 Caso = Nome
52 k1_data = array(["k1"])
53 intervalo = array(["Intervalo(thetap)", "Intervalo(u)", "Intervalo(v)", "
Intervalo(theta)", "Intervalo(psi)"])
54 freq_max = array(["Frequencia Maxima(thetap)", "Frequencia Maxima(u)", "
Frequencia Maxima(v)", "Frequencia Maxima(theta)", "Frequencia Maxima(psi)
"])
55 onde_freqmax = array(["Onde ocorre a Frequencia Maxima(thetap)", "Onde
ocorre a Frequencia Maxima(u)", "Onde ocorre a Frequencia Maxima(v)", "
Onde ocorre a Frequencia Maxima(theta)", "Onde ocorre a Frequencia Maxima
(psi)"])
56 med = array(["Media(thetap)", "Media(u)", "Media(v)", "Media(theta)", "
Media(psi)"])
57 desv = array(["Desvio Padrao(thetap)", "Desvio Padrao(u)", "Desvio Padrao
(v)", "Desvio Padrao(theta)", "Desvio Padrao(psi)"])
58 media_freq = array(["Media da Frequencia(thetap)", "Media da Frequencia
(u)", "Media da Frequencia(v)", "Media da Frequencia(theta)", "Media da
Frequencia(psi)"])
59 desv_freq = array(["Desvio Padrao da Frequencia(thetap)", "Desvio
Padrao da Frequencia(u)", "Desvio Padrao da Frequencia(v)", "Desvio
Padrao da Frequencia(theta)", "Desvio Padrao da Frequencia(psi)"])
60 obliquidade = array(["Obliquidade(thetap)", "Obliquidade(u)", "
Obliquidade(v)", "Obliquidade(theta)", "Obliquidade(psi)"])
61 curtose = array(["Curtose(thetap)", "Curtose(u)", "Curtose(v)", "Curtose(
theta)", "Curtose(psi)"])
62 obliquidade_freq = array(["Obliquidade da Frequencia(thetap)", "
Obliquidade da Frequencia(u)", "Obliquidade da Frequencia(v)", "
Obliquidade da Frequencia(theta)", "Obliquidade da Frequencia(psi)"])
63 curtose_freq = array(["Curtose da Frequencia(thetap)", "Curtose da
Frequencia(u)", "Curtose da Frequencia(v)", "Curtose da Frequencia(theta)
", "Curtose da Frequencia(psi)"])
64
65 if os.path.isdir(Nome):
66     print("Pasta (" + Nome + ") foi aberta com sucesso!")
67 else:
68     os.makedirs(Nome)
69     print("Pasta (" + Nome + ") criada com sucesso!")
70     os.makedirs(Nome + "/DataLake")
71     os.makedirs(Nome + "/BancodeDados")
72     os.makedirs(Nome + "/BancodeDados/Graficos")
73
74
75 while (Inicio <= Final):
76     k1 = ql_val*Inicio
77
78     NumSTR = str(round(k1*10**(-6), 4))
79     NumSTR = NumSTR.replace(".", "d")
80     Iter = "Caso_k1=_{}_MNm".format(NumSTR)
81     if os.path.isdir(Nome + "/DataLake/" + Iter):
82         print("\nJa foi calculado para este k1\n")
83     else:
84         os.makedirs(Nome + "/DataLake/" + Iter)
85
86     print("\nCalculando Para k1 = {} MNm \n".format(round(k1*10**(-6),

```

```

4)))
87     qintervalo, qmedia, qstd, max_freq, onde_maxfreq, fmean, fdesv,
      qobliquidade, qcurtose, obliquidade_freq1, curtose_freq1 = Calculos.
      calculos(J, Bn, m, Id, Ip, k1, k2, a, b, Nome, Iter, Pot, Te, Tm, c1, c2
      , beta, eps, Amp, freq, dt, t)
88
89     karray = array([k1])
90     k1_data = vstack((k1_data, karray))
91     intervalo = vstack((intervalo, qintervalo))
92     freq_max = vstack((freq_max, max_freq))
93     onde_freqmax = vstack((onde_freqmax, onde_maxfreq))
94     med = vstack((med, qmedia))
95     desv = vstack((desv, qstd))
96     media_freq = vstack((media_freq, fmean))
97     desv_freq = vstack((desv_freq, fdesv))
98     obliquidade = vstack((obliquidade, qobliquidade))
99     curtose = vstack((curtose, qcurtose))
100    obliquidade_freq = vstack((obliquidade_freq, obliquidade_freq1))
101    curtose_freq = vstack((curtose_freq, curtose_freq1))
102
103    Inicio = Inicio + Passo
104
105    tudo = concatenate((k1_data, intervalo), axis=1)
106    tudo = concatenate((tudo, med), axis=1)
107    tudo = concatenate((tudo, desv), axis=1)
108    tudo = concatenate((tudo, obliquidade), axis=1)
109    tudo = concatenate((tudo, curtose), axis=1)
110    tudo = concatenate((tudo, freq_max), axis=1)
111    tudo = concatenate((tudo, onde_freqmax), axis=1)
112    tudo = concatenate((tudo, media_freq), axis=1)
113    tudo = concatenate((tudo, desv_freq), axis=1)
114    tudo = concatenate((tudo, obliquidade_freq), axis=1)
115    tudo = concatenate((tudo, curtose_freq), axis=1)
116
117    savetxt(Nome + "/BancodeDados/" + Caso + ".csv", tudo, fmt = '%s',
      delimiter = ',') #Salvar o Banco de Dados
118
119    if FazerImg:
120        salvarfig.salv_fig(Nome, Caso)
121    print("Demorou {} segundos para terminar as iterações".format(time.time
      () - start_time))

```

A.2 CÁLCULO

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4 import os
5 import time
6 import pandas
7
8 from numpy import mean, max, diff, min, std
9 from math import pi, sin, cos
10 from scipy.integrate import odeint
11 from numpy.random import rand
12 from numpy.fft import fftn
13 from random import random

```

```

14
15 #-----
16 #function that returns dy/dt
17
18 def compute_dqdt(q,t,M,K,B,Ip,Bn,u,Te,Tm,Pot,cT,cC,cR,eps,m,beta,Id,amp,
    freq):
19     dqdt = np.empty((10,))
20
21     u[5] = Pot/q[5] - Tm - amp*sin(freq*pi*2*t)
22     u[6] = m*eps*q[5]**2*cos(q[5]*t)
23     u[7] = m*eps*q[5]**2*sin(q[5]*t)
24     u[8] = -(Id-Ip)*beta*q[5]**2*sin(q[5]*t)
25     u[9] = (Id-Ip)*beta*q[5]**2*cos(q[5]*t)
26     #u[5]=Te-Tm
27     C = np.array([[Bn, 0, 0, 0, 0],
28                  [ 0, cT, 0, 0, cC],
29                  [ 0, 0, cT, -cC, 0],
30                  [ 0, 0, -cC, cR, q[5]*Ip],
31                  [ 0, cC, 0, -q[5]*Ip, cR]]) #Matriz amortecimento
32
33     #Montando a matriz A
34     A = np.zeros((10,10))
35
36     A[5:10,0:5] = np.dot(-M,K)
37     A[0:5,5:10] = np.identity(5)
38     A[5:10,5:10] = np.dot(-M,C)
39
40     dqdt = np.dot(A,q) + np.dot(B,u)
41     return dqdt
42
43 #Função FFT
44 def ffthalfamp(t, y, T, N):
45
46     yf = fftn(y)
47
48     amp = np.abs(yf) # get amplitude spectrum
49
50     amp = (2/amp.size)*amp[0:amp.size//2]
51
52     freq = np.linspace(0.0, 1.0 / (2.0*T) , N//2) # get freq axis
53
54     return freq, amp
55
56 #Função obliquidade
57 def obliquidade(x, xmed, s, n):
58     sum = 0
59     for i in range (0,n):
60         sum = sum + (x[i]-xmed)**3
61     return sum/((n-1)*s**3)
62
63 #Função curtose
64 def curtose(x, xmed, s, n):
65     sum = 0
66     for i in range (0,n):
67         sum = sum + (x[i]-xmed)**4
68     return sum/((n-1)*s**4)-3
69
70

```

```

71 #-----
72 class Calculos:
73     def calculos(J, Bn, m, Id, Ip, k1, k2, a, b, Nome, Iter, Pot, Te, Tm,
74         c1, c2, beta, eps, amp, freq, dt, tf):
75
76         matplotlib.rcParams.update({"font.size": 11})
77         #J_or = 0.0563
78         #Bn_or = 0.002
79
80         kT =      k1 +      k2
81         kC =  -a*k1 +      b*k2
82         kR = a**2*k1 + b**2*k2
83         cT =      c1 +      c2
84         cC =  -a*c1 +      b*c2
85         cR = a**2*c1 + b**2*c2
86         omega_0 = Pot/Te
87
88         Caso = "k1 = {} MNm\n".format(round(k1*10**(-6),4))
89         k1STR = str(round(k1*10**(-6),4))
90         k1STR = k1STR.replace(".", "d")
91         CasoSTR = "k1 = {} MNm".format(k1STR)
92         CasoSTR = CasoSTR.replace(" ", "_")
93
94
95         #initial conditions
96         x10 = 0.0 #theta_p
97         x20 = 0.0 #u
98         x30 = 0.0 #v
99         x40 = 0.0 #theta
100        x50 = 0.0 #psi
101        x60 = omega_0 #\dot{theta} Velocidade
102        x70 = 0.0
103        x80 = 0.0
104        x90 = 0.0
105        x100 = 0.0
106
107        M = np.array([[J, 0, 0, 0, 0],
108                    [0, m, 0, 0, 0],
109                    [0, 0, m, 0, 0],
110                    [0, 0, 0, Id, 0],
111                    [0, 0, 0, 0, Id]]) #Matriz massa
112
113        print (M)
114
115        K = np.array([[0, 0, 0, 0, 0],
116                    [0, kT, 0, 0, kC],
117                    [0, 0, kT, -kC, 0],
118                    [0, 0, -kC, kR, 0],
119                    [0, kC, 0, 0, kR]]) #Matriz rigidez
120
121        print (K)
122
123        invM = np.linalg.inv(np.matrix(M))
124
125        #Montando a matriz B
126        B = np.zeros((10,10))
127        B[5:10,5:10] = invM

```

```

128
129     q0 = np.empty((10,))
130
131     q0[0] = x10
132     q0[1] = x20
133     q0[2] = x30
134     q0[3] = x40
135     q0[4] = x50
136     q0[5] = x60
137     q0[6] = x70
138     q0[7] = x80
139     q0[8] = x90
140     q0[9] = x100
141
142     #input vector
143     u = np.zeros((10,))
144
145     #time points
146     dt = dt
147     tf = tf
148     N = round(tf/dt)
149     t = np.linspace(0,tf,N)
150     Ni = round(N/2)
151
152     q = odeint(compute_dqdt,q0,t,args=(invM, K, B, Ip, Bn, u, Te, Tm,
Pot, cT, cC, cR, eps, m, beta, Id, amp, freq) )
153
154     tudo = np.empty((N,6))
155     tudo[:N,0] = np.transpose(t)
156     tudo[:N,1] = q[:N,5]
157     tudo[:N,2] = q[:N,1]*10**3
158     tudo[:N,3] = q[:N,2]*10**3
159     tudo[:N,4] = q[:N,3]
160     tudo[:N,5] = q[:N,4]
161
162     #estatística dos dados em arrays
163
164     qmax = max(q[:,1:6],0) #máximos
165     qmin = min(q[:,1:6],0) #mínimos
166     qmedia = mean(q[:,1:6],0) #médias
167     qstd = std(q[:,1:6],axis = 0, ddof = 1) #desvio padrão
168     qintervalo = np.array([qmax[4]-qmin[4],qmax[0]-qmin[0],qmax[1]-qmin
169 [1],qmax[2]-qmin[2],qmax[3]-qmin[3]]) #intervalo
170     qmedia = np.array([qmedia[4],qmedia[0],qmedia[1],qmedia[2],qmedia
171 [3]])
172     qstd = np.array([qstd[4],qstd[0],qstd[1],qstd[2],qstd[3]])
173     qshort = q[Ni:-1,:]
174     qm = mean(qshort, axis = 0)
175
176     #Obliquidade
177     qobliquidade = np.array([obliquidade(q[:,5],qmedia[0],qstd[0],N),
obliquidade(q[:,1],qmedia[1],qstd[1],N), obliquidade(q[:,2],qmedia[2],
qstd[2],N), obliquidade(q[:,3],qmedia[3],qstd[3],N), obliquidade(q[:,4],
qmedia[4],qstd[4],N)])
178
179     #Curtose
180     qcurtose = np.array([curtose(q[:,5],qmedia[0],qstd[0],N), curtose(q
[:,1],qmedia[1],qstd[1],N), curtose(q[:,2],qmedia[2],qstd[2],N), curtose

```

```

(q[:,3],qmedia[3],qstd[3],N), curtose(q[:,4],qmedia[4],qstd[4],N)])
179
    tshort = t[Ni:-1]
180
    thetapmei = q[Ni:-1,5] - qm[5]
181
182
    #FFTs
183
    freq_thetap, yf_thetap = ffthalfamp(tshort, thetapmei, dt, N-Ni-1)
184
    freq_umei, yf_umei = ffthalfamp(t, q[:,1]*10**3, dt, N)
185
    freq_vmei, yf_vmei = ffthalfamp(t, q[:,2]*10**3, dt, N)
186
    freq_thetamei, yf_thetamei = ffthalfamp(t, q[:,3], dt, N)
187
    freq_psimei, yf_psimei = ffthalfamp(t, q[:,4], dt, N)
188
189
    #Valores estatísticos das FFTs
190
    media_freq = np.array([mean(yf_thetap,0), mean(yf_umei,0), mean(
191 yf_vmei,0), mean(yf_thetamei,0),mean(yf_psimei,0)])
    desv_freq = np.array([std(yf_thetap,axis = 0, ddof = 1), std(
192 yf_umei,axis = 0, ddof = 1), std(yf_vmei,axis = 0, ddof = 1), std(
    yf_thetamei,axis = 0, ddof = 1), std(yf_psimei,axis = 0, ddof = 1)])
193
    max_freq = np.array([max(yf_thetap,0), max(yf_umei,0), max(yf_vmei
194 ,0), max(yf_thetamei,0), max(yf_psimei,0)])
    curtose_freq = np.array([curtose(yf_thetap,mean(yf_thetap,0),
195 desv_freq[0],Ni//2-1), curtose(yf_umei,mean(yf_umei,0),desv_freq[1],N
    //2-1), curtose(yf_vmei,mean(yf_vmei,0),desv_freq[2],N//2-1), curtose(
    yf_thetamei,mean(yf_thetamei,0),desv_freq[3],N//2-1), curtose(yf_psimei,
    mean(yf_psimei,0),desv_freq[4],N//2-1)])
196
    obliquidade_freq = np.array([obliquidade(yf_thetap,mean(yf_thetap
197 ,0),desv_freq[0],Ni//2-1), obliquidade(yf_umei,mean(yf_umei,0),desv_freq
198 [1],N//2-1), obliquidade(yf_vmei,mean(yf_vmei,0),desv_freq[2],N//2-1),
199 obliquidade(yf_thetamei,mean(yf_thetamei,0),desv_freq[3],N//2-1),
200 obliquidade(yf_psimei,mean(yf_psimei,0),desv_freq[4],N//2-1)])
201
    #procurar onde no yf é o máximo
202
    onde_thetap = np.where(yf_thetap == max(yf_thetap,0))
203
    onde_thetap = onde_thetap[0]
204
    onde_umei = np.where(yf_umei == max(yf_umei,0))
205
    onde_umei = onde_umei[0]
206
    onde_vmei = np.where(yf_vmei == max(yf_vmei,0))
207
    onde_vmei = onde_vmei[0]
208
    onde_thetamei = np.where(yf_thetamei == max(yf_thetamei,0))
209
    onde_thetamei = onde_thetamei[0]
    onde_psimei = np.where(yf_psimei == max(yf_psimei,0))
    onde_psimei = onde_psimei[0]
    onde_maxfreq = np.array([freq_thetap[onde_thetap[0]], freq_umei[
210 onde_umei[0]], freq_vmei[onde_vmei[0]], freq_thetamei[onde_thetamei[0]],
    freq_psimei[onde_psimei[0]]])
211
    #Gráficos das FFTs
212
    fig1,axs1 = plt.subplots(5)
213
    axs1[0].plot(freq_thetap,np.round_(yf_thetap, decimals = 2), 'b')
214
    axs1[0].title.set_text("Amplitudes da FFT - " + Caso)
215
    axs1[0].set_ylabel("thetap (rad/s)")
216
    axs1[0].grid()
217
    axs1[0].axes.xaxis.set_ticklabels([])
218
    axs1[1].plot(freq_umei,np.round_(yf_umei, decimals = 2), 'r')
219
    axs1[1].set_ylabel("u (mm)")
220
    axs1[1].grid()
221

```



```

222     axs1[1].axes.xaxis.set_ticklabels([])
223
224     axs1[2].plot(freq_vmei,np.round_(yf_vmei, decimals = 2), 'g')
225     axs1[2].set_ylabel("v (mm)")
226     axs1[2].grid()
227     axs1[2].axes.xaxis.set_ticklabels([])
228
229     axs1[3].plot(freq_thetamei,np.round_(yf_thetamei*10**3, decimals =
230 2), 'c')
231     axs1[3].set_ylabel("theta (rad)  $\mathregular{10^{-3}}$ ")
232     axs1[3].grid()
233     axs1[3].axes.xaxis.set_ticklabels([])
234
235     axs1[4].plot(freq_pstimei,np.round_(yf_pstimei*10**3, decimals = 2),
236 'y')
237     axs1[4].set_ylabel("psi (rad)  $\mathregular{10^{-3}}$ ")
238     axs1[4].grid()
239     plt.xlabel(r"Frequência(Hz)")
240     plt.savefig(Nome + "/DataLake/" + Iter + "/" + "FFT-" + CasoSTR + ".
241 .png", format = 'png', bbox_inches = 'tight', dpi = 1200)
242     plt.close()
243     plt.clf()
244
245     #Gráficos dos Graus de Liberdade
246     fig,axs = plt.subplots(5)
247
248     axs[0].plot(t,np.round_(q[:,5], decimals = 2),'b')
249     axs[0].set_ylabel('thetap (rad/s)')
250     axs[0].grid()
251     axs[0].title.set_text(Caso)
252     axs[0].axes.xaxis.set_ticklabels([])
253
254     axs[1].plot(t, np.round_(q[:,1]*10**3, decimals = 2), 'g')
255     axs[1].set_ylabel('u (mm)')
256     axs[1].grid()
257     axs[1].axes.xaxis.set_ticklabels([])
258
259     axs[2].plot(t, np.round_(q[:,2]*10**3, decimals = 2), 'r')
260     axs[2].set_ylabel('v (mm)')
261     axs[2].grid()
262     axs[2].axes.xaxis.set_ticklabels([])
263
264     axs[3].plot(t, np.round_(q[:,3]*10**3,decimals = 2), 'c')
265     axs[3].set_ylabel('theta (rad)  $\mathregular{10^{-3}}$ ')
266     axs[3].set_ylim(-0.1,0.1)
267     axs[3].grid()
268     axs[3].axes.xaxis.set_ticklabels([])
269
270     axs[4].plot(t, np.round_(q[:,4]*10**3, decimals = 2), 'y')
271     axs[4].set_ylabel('psi (rad)  $\mathregular{10^{-3}}$ ')
272     axs[4].set_ylim(-0.1,0.1)
273     axs[4].grid()
274
275     plt.xlabel('Tempo(s)')
276     plt.savefig(Nome + "/DataLake/" + Iter + "/" + CasoSTR + ".png",
277 format = 'png', bbox_inches = 'tight', dpi = 1200)
278
279     #Salvar Data Lake

```

```

276     np.savetxt(Nome+ "/DataLake/" + Iter + "/" + CasoSTR + ".csv", tudo
, delimiter = ',')
277     texto = open(Nome+ "/DataLake/" + Iter + "/" + CasoSTR + ".csv", "r
")
278     conteudo = texto.read()
279     texto = open(Nome+ "/DataLake/" + Iter + "/" + CasoSTR + ".csv", "w
")
280     texto.write("tempo (s), thetap (rad/s), u (mm), v (mm), theta (rad), psi (rad)
\n" + conteudo)
281     texto.close()
282     plt.close()
283
284     #return qmax, qmin, qmedia, qstd, max_freq, onde_maxfreq,
media_freq, desv_freq
285     return qintervalo, qmedia, qstd, max_freq, onde_maxfreq, media_freq
, desv_freq, qobliquidade, qcurtose, obliquidade_freq, curtose_freq

```

A.3 SALVAR GRÁFICOS

```

1 import numpy as np
2 import csv
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import matplotlib
6 import time
7
8 class salvarfig:
9     def salv_fig(Nome, Caso):
10         start_time_fig = time.time()
11         matplotlib.rcParams.update({"font.size": 5})
12         data = pd.read_csv(Nome + "/BancodeDados/" + Caso + ".csv")
13         data['k1'] = data['k1'] / 10**6
14         data.set_index('k1', inplace=True, drop=True)
15         data = (data/data.values[0]) * 100
16         row_count, col_count = data.shape
17         plt.rcParams["figure.figsize"] = [7.50, 3.50]
18         plt.rcParams["figure.autolayout"] = True
19
20         for x in range(0, col_count):
21             fig = plt.figure()
22             data.plot(y=data.columns[x], use_index=True)
23             plt.grid()
24             plt.ylabel(data.columns[x] + "(%) do valor inicial")
25             plt.xlabel("k1 (MNm)")
26             plt.savefig(Nome + "/BancodeDados/Graficos/Grafico " + data.
columns[x] + ".png", format='png', bbox_inches='tight', dpi=1200)
27             plt.clf()
28             plt.close(fig)
29             print("Para fazer todas as figuras das variações, demorou {}
segundos".format(time.time() - start_time_fig))

```