



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Luciane Agnoletti dos Santos Pedotti

**Dispositivo IoT de baixo custo para Diagnóstico de Falhas
em Máquinas Rotativas**

Campinas

2019



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Luciane Agnoletti dos Santos Pedotti

Dispositivo IoT de baixo custo para Diagnóstico de Falhas em Máquinas Rotativas

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Engenharia Elétrica, na Área de Eletrônica, Microeletrônica e Optoeletrônica.

Orientador: Prof. Dr. Fabiano Fruett

Este exemplar corresponde à versão final da tese defendida pela aluna Luciane Agnoletti dos Santos Pedotti, e orientada pelo Prof. Dr. Fabiano Fruett

Campinas

2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

P342d Pedotti, Luciane Agnoletti dos Santos, 1982-
Dispositivo IoT de baixo custo para Diagnóstico de Falhas em Máquinas Rotativas / Luciane Agnoletti dos Santos Pedotti. – Campinas, SP : [s.n.], 2019.

Orientador: Fabiano Fruett.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Acelerômetros. 2. Vibração - Medição. 3. Microcontroladores. 4. Sistemas embarcados (Computadores). 5. Máquinas elétricas - Manutenção e reparos. I. Fruett, Fabiano, 1970-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Low-cost IoT device for Fault Diagnosis in Rotating Machines

Palavras-chave em inglês:

Accelerometers

Vibration - Measurement

Microcontrollers

Embedded computer systems

Electric machinery - Maintenance and repair

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Doutora em Engenharia Elétrica

Banca examinadora:

Fabiano Fruett [Orientador]

Rafael Ferrari

José Antenor Pomilio

Roberto Ribeiro Neli

Maria Bernadete de Moraes França

Data de defesa: 11-12-2019

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0003-0205-366X>
- Currículo Lattes do autor: <http://lattes.cnpq.br/4009726335510654>

COMISSÃO JULGADORA – TESE DE DOUTORADO

Candidata: Luciane Agnoletti dos Santos Pedotti **RA:** 180859

Data da Defesa: 11 de dezembro de 2019.

Título da Tese: “Dispositivo IoT de baixo custo para Diagnóstico de Falhas em Máquinas Rotativas”.

Prof. Dr. Fabiano Fruett

Prof. Dr. José Antenor Pomilio

Prof. Dr. Rafael Ferrari

Prof. Dr. Roberto Ribeiro Neli

Prof. Dr. Maria Bernadete de Moraes França

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Dedico esta tese aos meus pais, Cleusa e Antonio.

Agradecimentos

Agradeço primeiramente ao projeto DINTER entre a UTFPR (campus Campo Mourão) e a UNICAMP. Graças ao projeto tive a oportunidade de realizar este trabalho. Agradeço ao professor Dr. Fabiano Fruett pela orientação, atenção e paciência a mim dedicada, sempre com muita confiança, sabedoria e motivação. Agradeço aos colegas de UTFPR, em especial ao professor Dr. Roberto R. Neli pela oportunidade. Agradeço a UTFPR campus Curitiba por conceder-me afastamento de minhas atividades docentes para que eu pudesse passar preciosos anos na Unicamp me dedicando ao trabalho.

Agradeço também aos colegas de laboratório que foram praticamente minha família durante os anos que passei no LSM: José Ramirez, Rafael Montali, Diego Deotti, Carlo Giuliano, Audrey, Carolina, Pedro e Luiz Eduardo, muito obrigada por tudo. Agradeço ao colega Rodrigo Daniel da Silva pela ajuda com o controlador do BLDC que foi muito importante para a coleta de dados. Um agradecimento especial ao colega e parceiro de trabalho Ricardo Zago cujas contribuições foram grandiosas.

Agradeço aos professores Dr. Mateus Giesbrecht e Dr. José Gilberto Dalfré Filho (FEC) pela preciosa colaboração e parceria. Agradeço ao Jefferson Cutrim Rocha pela sua ajuda nos ensaios realizados na FEC.

Agradeço à equipe da Inova - Unicamp, por todo o suporte no momento em que buscamos registrar patente para o dispositivo.

Agradeço aos meus pais, Antonio e Cleusa, irmãos, Simone e Eduardo, sobrinhos, cunhados, sogro (in memorian), sogra e meu marido Arthur, que sempre me deram motivação para continuar meu trabalho e foram meu alicerce e colo sempre.

Um agradecimento especial à minha família campineira, que me acolheram com carinho nos anos que estive em Campinas, tia Suely e tio Jamiro e os primos Carol, Hugo, Ricardo, Mariana, Lucas, Antonio e José. Eles fizeram meus dias mais felizes. Agradeço a todos que de forma direta ou indiretamente fizeram parte deste trabalho.

Agradeço a todos os ensinamentos recebidos na Casa da Vó Maria.

Obrigada a Deus por mais esta conquista! Obrigada minha mãe Iansã e meu pai Xangô!

"Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota".

Madre Teresa de Calcutá

"Fio, desejar é o primeiro passo para a conquista de nossos sonhos".

Vovó Maria Conga

Resumo

Este trabalho apresenta um dispositivo de baixo custo com capacidade de mensuramento de vibrações utilizando como dispositivo transdutor de vibração um acelerômetro triaxial MEMS (*Microelectromechanical Systems*) aplicado a máquinas que operam numa velocidade de até 4000 RPM. Este dispositivo é capaz de obter o espectro com qualidade até a sexta ordem, de fácil instalação, com conectividade sem fio, facilitando sua inserção à internet das coisas (*IoT - Internet of Things*) e totalmente *opensource*.

Para selecionar o acelerômetro adequado à nossa aplicação, testes de bancada foram realizados utilizando um motor BLDC e uma bomba centrífuga. Diversas perturbações foram inseridas nos testes, como desbalanceamento, carga, cavitação, entre outros. A resolução do sensor escolhido foi capaz de fornecer dados suficientes para o processamento. Diversas técnicas de análise de dados foram utilizadas, tais como análise de gráfico tipo *waterfall*, regressão logística, linear SVM e redes neurais artificiais, e em todas delas foi possível diagnosticar as perturbações inseridas.

O sistema apresentado utiliza dados pré-processados em um microcontrolador ESP 32 de baixo custo. O microcontrolador recebe os dados temporais de vibração e velocidade da máquina objeto de estudo, calcula a FFT do sinal de vibração nos eixos X e Y e envia os dados para um servidor de IoT em nuvem que os armazenada. O envio desse sinal é feito através de uma comunicação sem fio utilizando a tecnologia IEEE 802.11, conhecida popularmente como Wi-Fi.

O sistema final consiste em uma placa de aquisição dos sinais de vibração e um software que faz o pré-processamento do sinal de vibração. O custo da produção do protótipo do nosso dispositivo é inferior a US\$30,00 e atendeu aos objetivos propostos de custo, integração IoT, processamento da FFT do sinal de vibração e armazenamento de dados em nuvem. O software foi desenvolvido utilizando apenas linguagens de programação gratuitas. Todo o projeto está disponível em um repositório *online* (*open hardware* e *open software*), sendo este aspecto parte da contribuição geral dessa tese.

Palavras-chaves: acelerômetro; MEMS; medição de vibração; FFT do sinal de vibração; ESP32; MMA8451Q.

Abstract

In this work is presented a low-cost device with capability of vibration measurement using as vibration transducer device a MEMS (Microelectromechanical Systems) triaxial accelerometer applied in machinery which operates under speed of until 4000 RPM. This device is able to obtain a quality spectrum until sixth order, with easy installation, with wireless connection, facilitating its insertion into the internet of things (IoT) and totally opensource.

Bench tests were made to select the accelerometer suitable for our application. It was used one BLDC motor and one centrifugal pump. Several disturbances were inserted in the tests, as unbalance, load, cavitation, among others. The resolution of the sensor chosen was able to provide data enough for processing. Different data analysis techniques were used, such as waterfall chart analysis, logistic regression, linear SVM and artificial neural networks, and in all of them were possible to diagnosis the disturbances inserted.

The presented system uses preprocessed data in a low-cost ESP32 microcontroller. The microcontroller receives the time domain vibration data and the speed of the study machine object, and calculates the FFT from the vibration signal in the X and Y-axis and sends these data to an IoT cloud server that stores it. This signal is sent through wireless communication using the IEEE 802.11 technology, popularly known as Wi-Fi.

The final system consists of an acquisition vibration signal board and software that do the pre-processing of the vibration signal. The prototype manufacture cost of our device is less than US\$ 30.00 and met the proposed objectives of cost, IoT integration, vibration signal FFT processing, and cloud storage. The software was developed using only free programming languages. All the project is available in an online repository (open hardware and open software), being this aspect a parcel of the general contribution of this thesis.

Keywords: accelerometer; MEMS; vibration measurement; FFT from the vibration signal; ESP32; MMA8451Q.

Lista de ilustrações

Figura 1 – Principais causas de falhas em máquinas rotativas reportadas. Fonte: Adaptado de Celulose Online (2019)	20
Figura 2 – Volume de pesquisas efetuadas no Google sobre <i>Wireless Sensor Networks</i> e <i>Internet of Things</i> . Fonte: Adaptado de Santos <i>et al.</i> (2016)	27
Figura 3 – Instrumentos de medição e análise de vibração comerciais pesquisados	28
Figura 4 – Representação gráfica da amplitude do espectro da vibração utilizando: (a) waterfall (b) espectrograma. Fonte: Autoria própria.	33
Figura 5 – Características do espectro da vibração para diversos tipos de falhas em máquinas rotativas	35
Figura 6 – Classificação dos Níveis de Severidade de Vibração segundo a ISO 10816-3. Fonte: ISO10816-3:2009 (2009).	35
Figura 7 – Efeito da força centrífuga no eixo de uma estrutura em rotação com uma massa de desbalanceamento	37
Figura 8 – Possíveis condições de desalinhamento: (a) angular (b) paralelo (c) combinado. Fonte: Adaptado de Braga (2019)	37
Figura 9 – Estrutura típica de montagem de mancal e rolamento. Fonte: Adaptado de (EUROPE, 2019).	39
Figura 10 – Curva PF. Fonte: Adaptado de Teles (2019a).	42
Figura 11 – Curva PF x Custos de Reparo. Fonte: Adaptado de Teles (2019a).	42
Figura 12 – Movimento harmônico simples com projeção do vetor girante	43
Figura 13 – Adição vetorial de funções harmônicas	45
Figura 14 – Fenômeno de batidas	46
Figura 15 – Representação de funções no domínio do tempo e frequência: (a) $x_1(t)$ (b) DFT de $x_1(t)$ (c) $x_2(t)$ (d) DFT de $x_2(t)$	48
Figura 16 – Discretização do sinal contínuo: (a) sinal contínuo (b) processo de amostragem (c) sinal discretizado. Fonte: autoria própria	49
Figura 17 – Acelerômetro piezoeletrico típico. Fonte: Adaptado de MEDIÇÃO... (2019)	53
Figura 18 – Acelerômetro piezoresistivo triaxial de quatro <i>strain gauges</i>	54
Figura 19 – Acelerômetro MEMS capacitivo diferencial: (a) Esquemático (b) Formação da capacidade diferencial sob aceleração. Fonte: Extraído de Sinha <i>et al.</i> (2014)	56
Figura 20 – Linha de tempo da quantidade de processamento ao longo dos anos baseado na Lei do Moore. Fonte: Adaptado de Kurzweil (2006)	57

Figura 21 – Evolução dos acelerômetros MEMS em tamanho e custo. Fonte: Extraído de Developpement (2014)	57
Figura 22 – Chip MMA8451Q da NXP. Fonte: Extraído de Semiconductors (2017)	58
Figura 23 – Módulo ESP32-WROOM-32 da Espressif	59
Figura 24 – Linha de tempo de revolução industrial	59
Figura 25 – Perspectiva de Dispositivos Conectados até 2050. Fonte: Baseado em Phan (2016)	60
Figura 26 – Funcionamento do Publish, Broker e Subscribers no MQTT	63
Figura 27 – Visão geral de um sistema de computação em nuvem	64
Figura 28 – Esquema da solução desenvolvida	67
Figura 29 – Esquema de módulos necessários no dispositivo	69
Figura 30 – Esquema de alimentação por uma fonte externa	70
Figura 31 – Esquema de alimentação em 3,3V para o ESP32 e Acelerômetro	70
Figura 32 – Esquema de ligação do ESP32	71
Figura 33 – Esquema de ligação do MMA8451Q	72
Figura 34 – Esquema ligação dos botões de gravação	72
Figura 35 – Layout da PCB dispositivo desenvolvido	73
Figura 36 – Dispositivo desenvolvido (escala em centímetro)	74
Figura 37 – Fluxograma do código desenvolvido - Core 0	75
Figura 38 – Fluxograma do código desenvolvido - Core 1	76
Figura 39 – Bancada de teste	79
Figura 40 – Detalhes do acionamento do BLDC: (a) fluxograma do driver e DAQ (b) manopla de liga e desliga	80
Figura 41 – Medição da velocidade do motor: (a) Sensor instalado no eixo do motor (b) detalhe do optoacoplador LM393	81
Figura 42 – Simulador de carga: (a) Desenho do todo (b) Detalhe da disposição dos ímãs	81
Figura 43 – Fotografia do aparato de simulação de carga	82
Figura 44 – Simulação de desbalanceamento: (a) Contrapeso (b) Roda com contrapeso	83
Figura 45 – Objeto para simular obstrução: (a) Obstrutor (b) Roda com obstrutor .	83
Figura 46 – Bancada de testes com 2 sensores coletando informações simultaneamente, formando uma rede de sensores	84
Figura 47 – Instalação do Sensor Node 2 na bancada de testes	85
Figura 48 – Primeiro protótipo de testes	86
Figura 49 – Magnitude de desbalanceamento em função do peso aplicado	87
Figura 50 – Primeira montagem do aparato experimental do BLDC	88
Figura 51 – Gráfico Waterfall Frequência vs RPM vs Magnitude da Vibração com desbalanceamento de 90 g: (a) sem carga (b) com carga a 20 mm	88

Figura 52 – Planta experimental de testes com bomba centrífuga	90
Figura 53 – Bomba centrífuga utilizada no experimento: parafusos de fixação da base e posicionamento do dispositivo no mancal	90
Figura 54 – Pás da bomba utilizadas no experimento: Nova (à esquerda), Avariada (no centro) e Detalhe da Avariada (à direita)	92
Figura 55 – Modelo da RNA utilizado nesta análise	95
Figura 56 – Listas de dispositivos e seu estado	97
Figura 57 – Listas de opções do IBM Watson	97
Figura 58 – <i>Dashboard</i> para o Sensor Node 1	98
Figura 59 – Visualização amplificada dos dados de amplitude das seis harmônicas cor- respondentes ao eixo y do sensor node 1	98
Figura 60 – Ambiente de programação do Node-RED	99
Figura 61 – Login para acessar os dados criados com o Node-RED	99
Figura 62 – Visualização da dados do Sensor Node 1	100
Figura 63 – Ambiente Node-RED: (a) Detalhe da visualização dos gráficos (b) Alertas criados.	100
Figura 64 – Elementos do dispositivo: (a) Pino tipo Jack (b) Mini-jumper para 2 pinos	115
Figura 65 – Esquema de ligação do dispositivo Vibration Monitor com adaptador UART116	
Figura 66 – Modelo matemático de um neurônio artificial	146

Listas de tabelas

Tabela 1 – Custos dos principais dispositivos comerciais	29
Tabela 2 – Frequências de Vibração e Suas Causas Prováveis. Fonte: autoria própria através de dados compilados de Mais (2019) e Bloch e Geitner (1999)	34
Tabela 3 – Especificações do Dispositivo Desenvolvido	68
Tabela 4 – Custo do Dispositivo	73
Tabela 5 – Lista de Materiais do Dispositivo <i>Vibration Monitor</i>	77
Tabela 6 – Testes realizados	91
Tabela 7 – Condição de Cavitação x Pressão Medida	92
Tabela 8 – Comparação dos resultados obtidos utilizando três formas de análise e números diferentes de entradas de dados	93
Tabela 9 – Condições de Operações Testadas	94
Tabela 10 – Precisão na detecção de falhas utilizando Regressão Logística como classificador	95
Tabela 11 – Precisão na detecção de falhas utilizando RNA como classificador	95
Tabela 12 – Pinagem entre o Vibration Monitor e Adaptador UART	116
Tabela 13 – Pinagem entre a barra de pinos P4 e o ESP32	116
Tabela 14 – Denominação das Frequências das Ondas Sonoras	157

Lista de Acrônimos e Abreviações

AES Advanced Encryption Standard

AI Analog Input

API Application Programming Interface

BLDC Brushless Direct Current

BLE Bluetooth Low Energy

bps bits por segundo

DAQ Data Acquisition

DI Digital Input

FFT Fast Fourier Transform

GND Ground

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IIoT Industrial Internet of Things

IoT Internet of Things (Internet das coisas)

ISM Industrial, Scientific and Medical Band

LAN Local Area Network

M2M Machine to Machine

MCU Movimento Circular Uniforme

MEMS Microelectromechanical Systems

MHS Movimento Harmônico Simples

MLP Multi-Layer Perceptron

MQTT Message Queue Telemetry Transport

NFC Near Field Communication

OEE Overall Equipment Effectiveness

PCB Printed Circuit Board

RCM Reliability Centered Maintenance

RF Radio Frequency

RFID Radio-Frequency IDentification

RMS Root Mean Square

RNA Rede Neural Artificial

RPM Rotações por minuto

RX Receive

SMD Surface Mount Device

THT Through-Hole Technology

TI Tecnologia da Informação

TPM Total Productive Maintenance

TTL Transistor-Transistor Logic

TX Transmit

UART Universal Asynchronous Receiver-Transmitter

UNB Ultra Narrow Band

USB Universal Serial Bus

WAN Wide Area Network

WEP Wired Equivalent Privacy

WPA Wired Protected Access

WSN Wireless Sensor Network

Sumário

Sumário	
1 Introdução	19
1.1 Objetivo Geral	21
1.2 Objetivos Específicos	21
1.3 Contribuições deste trabalho	22
1.4 Organização da Tese	22
2 Revisão Bibliográfica	24
2.1 Monitoramento da Vibração	24
2.2 Comunicação sem fio para IoT	25
2.3 Dispositivos de Medição da Vibração Existentes	26
2.4 Dispositivos similares com patente reconhecida	29
2.5 Conclusões deste capítulo	30
3 Conceitos de Vibração e Falhas Associadas	32
3.1 Diagnóstico de Avarias através da Análise da Vibração	36
3.1.1 Desbalanceamento Mecânico	36
3.1.2 Desalinhamento do Eixo	37
3.1.3 Ressonância	38
3.1.4 Rolamentos	38
3.1.5 Sobrecarga / <i>Overload</i>	39
3.1.6 Falhas Elétricas em Máquinas Rotativas	40
3.2 Seleção dos Pontos de Medição	40
3.3 Manutenção Preditiva	40
3.4 Conclusões deste capítulo	41
4 Embasamento Matemático	43
4.1 Análise Harmônica	46
4.2 Transformada de Fourier	47
4.3 Sinais Discretos	48
4.3.1 Transformada Discreta de Fourier	49
4.3.1.1 Transformada Rápida de Fourier	49
4.4 Conclusões deste capítulo	50
5 Tecnologias Envolvidas	51
5.1 Acelerômetros	51
5.1.1 Piezoelétrico	52

5.1.2	Piezoresistivo	53
5.1.3	Capacitivo	53
5.1.4	Acelerômetro de efeito Hall	54
5.1.5	Magnetoresistivo	54
5.1.6	Redes de Bragg em fibras ópticas	54
5.2	Acelerômetros MEMS	55
5.2.0.1	MMA8451Q	56
5.3	Hardware de processamento	58
5.4	Indústria 4.0	59
5.5	Internet das coisas - IoT	61
5.5.1	Tecnologia de comunicação sem fio para IoT	62
5.5.1.1	Protocolo MQTT	63
5.5.2	Armazenamento em Nuvem	64
5.6	Solução Desenvolvida	66
5.7	Conclusões deste capítulo	66
6	Desenvolvimento do Dispositivo	68
6.1	Especificações do Dispositivo	68
6.2	Diagrama do Circuito Eletrônico	69
6.3	Procedimento de Inicialização e Download de Firmware	72
6.4	Prototipagem	73
6.5	Código Fonte	74
6.6	Lista de Materiais	77
6.7	Conclusões deste capítulo	78
7	Bancada de Testes	79
7.1	Simulador de Carga	80
7.2	Simulador de Desbalanceamento	82
7.3	Simulador de Obstrução	83
7.4	Rede de Sensores	83
7.5	Conclusões deste capítulo	84
8	Resultados	86
8.1	Avaliação do acelerômetro MEMS utilizando o kit de desenvolvimento da NXP FRDM-KL25Z	86
8.2	Avaliação de distúrbios em um motor BLDC	87
8.3	Avaliação de distúrbios em uma bomba centrífuga	89
8.4	Utilização da solução desenvolvida na análise de distúrbios em um motor BLDC	93
8.5	Visualização dos dados na nuvem	96
8.6	Conclusões deste capítulo	100

Conclusão	102
Publicações	104
Referências	105
Apêndices	114
APÊNDICE A Descritivo operacional do dispositivo	115
A.1 Pinagem	115
A.2 Instalação do Hardware do ESP32 no IDE do Arduino	116
APÊNDICE B Código Fonte do Dispositivo	118
APÊNDICE C Técnicas de Análise de Dados	145
C.1 Regressão Logística	145
C.2 Rede Neural Artificial	145
C.2.1 Multi-Layer Perceptron	146
APÊNDICE D Códigos de Análise de Dados em Python	148
D.1 Regressão Logística	148
D.2 Rede Neural - Multi-layer Perceptron	151
Anexos	156
ANEXO A Ultrassom	157
A.1 O que é ultrassom?	157
A.2 Análise de Avarias por Ultrassom	157

1 Introdução

Falhas no monitoramento da vibração podem resultar em consequências catastróficas como, por exemplo, o colapso que ocorreu com a Ponte de Tacoma Narrows em 1940, onde o vento induziu uma oscilação na ponte até atingir a frequência de oscilação natural da estrutura, iniciando um movimento oscilatório destrutivo. A estrutura da ponte não aguentou os efeitos da oscilação e caiu (BILLAH; SCANLAN, 1991). Um grave acidente em uma das turbinas da usina hidrelétrica de Sayano-Shushenskaya na Rússia foi reportado em 2009. Um reparo realizado nas pás da turbina deixaram-nas desbalanceadas, gerando vibrações que aumentavam substancialmente com o tempo. Devido à trepidação crescente, a tampa da turbina abriu, deslocando seu rotor, fazendo jorrar água no salão de máquinas, inundando rapidamente o edifício e matando 75 pessoas (SULAIMA *et al.*, 2014). Estes episódios evidenciam a importância do monitoramento da vibração.

A maioria dos equipamentos industriais possuem elementos rotativos que geram vibrações. Alguns níveis de vibração fazem da parte natureza do equipamento, entretanto, quando os níveis de vibração operam fora de seus padrões normais, significa que a máquina pode estar funcionando com alguma irregularidade, como desalinhanamento do eixo, desbalanceamento, falhas de lubrificação, desgaste de rolamentos, entre outros. Tais vibrações podem ser monitoradas para avaliar o estado do equipamento.

A análise de vibração em máquinas rotativas tem sido utilizada com sucesso na manutenção preditiva nas indústrias desde a década de 70 (BOON; BAUW, 2012). A presença da vibração pode significar várias avarias em máquinas, como desbalanceamento, desgaste dos mancais, trincas, afrouxamento de parafusos, fontes de tensão com mau funcionamento (desequilíbrio de fase, harmônicos, etc), entre outros (RAO, 2014). Frente a isso, diversas ferramentas de análise de vibração vêm sendo utilizadas para auxiliar na manutenção preditiva. A medição de vibração pode ser feita através de transdutores, como por exemplo, acelerômetros.

Segundo o artigo publicado por Celulose Online (2019), uma pesquisa realizada com os participantes da International Maintenance Conference IMC-2012 constatou que 97% das falhas em máquinas tem como causa: desalinhanamento (32%), falhas de rolamento (31%), desbalanceamento (18%), desgastes (16%), conforme ilustrado na Figura 1. Todos estes problemas alteram as frequências e amplitudes de vibração da máquina de forma particular, portanto, se bem utilizado, um sistema de detecção e predição de falhas através da análise da vibração é suficientemente capaz de diagnosticar a maioria das falhas em máquinas rotativas.

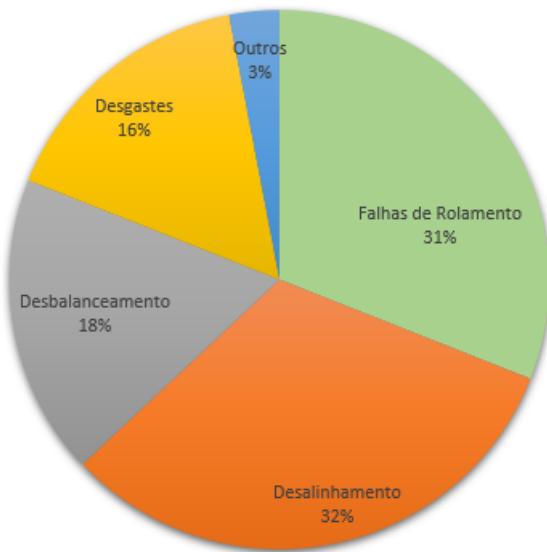


Figura 1 – Principais causas de falhas em máquinas rotativas reportadas. Fonte: Adaptado de Celulose Online (2019)

Devido às falhas de desalinhamento, por exemplo, a máquina apresenta aumento de temperatura, ruído e vibração que por sua vez, consomem parte da energia que deveria ser convertida em trabalho, levando a uma diminuição do rendimento da mesma. Quanto menor a eficiência, maior é o custo operacional do equipamento, e estudos apontam que um alinhamento correto pode reduzir em até 15% o consumo de energia elétrica da máquina (Celulose Online (2019)).

Além disso, o desalinhamento impacta diretamente em outros componentes da máquina que começam a sofrer deterioração precoce, levando à substituição do mesmo antes da vida útil total estimada, como acontece por exemplo com rolamentos, vedações e acoplamentos (Celulose Online (2019)).

Dessa forma, torna-se atrativa para a indústria a aquisição de ferramentas que auxiliem em suas análises para definições de manutenção preditiva. Equipamentos responsáveis por mensurar vibrações em processos produtivos possuem um custo consideravelmente alto, o que leva algumas empresas a contratar estes serviços de medição de maneira periódica. Entretanto, este tipo de prática não possibilita um levantamento histórico da evolução da variável medida ao longo do tempo de forma mais sistemática.

Segundo a última pesquisa realizada pela Associação Brasileira de Manutenção (Abra-man) em 2011, estima-se que a indústria nacional gastou em média 5% de seu faturamento bruto em manutenção, portanto, toda ferramenta eficiente no controle das manutenções gera um impacto significativo no lucro final (CRIATIVAS, 2018).

Assim sendo, este trabalho apresenta um dispositivo de baixo custo com capacidade de medir vibrações utilizando um acelerômetro MEMS (*Microelectromechanical Systems*) aplicado a máquinas que operam numa velocidade de até 4.000 RPM. Além disso, o dispositivo desenvolvido é capaz de obter o espectro com qualidade até a sexta ordem e é de simples instalação. Sua comunicação é sem fio, o que favorece sua inserção à internet das coisas (*IoT - Internet of Things*) e é completamente *opensource* (*hardware e software*).

1.1 Objetivo Geral

Desenvolvimento de um sistema de aquisição da vibração em máquinas rotativas que seja capaz de transmitir estes sinais em um meio sem fio e que esteja integrado a um sistema de computação em nuvem e ao IoT com um custo final máximo de US\$ 35,00.

1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são os seguintes:

- Identificar e validar acelerômetros MEMS de baixo custo.
- Selecionar uma tecnologia sem fio (*wireless*) de transmissão dentro das principais utilizadas no meio industrial.
- Desenvolver uma placa que adquira o sinal de vibração, processe o sinal e o transmita via comunicação *wireless*.
- Desenvolver a comunicação com um servidor que receba estes sinais e os disponibilize em uma plataforma de computação em nuvem.
- Parametrizar alguns padrões de leitura que possibilitem a comparação entre as leituras atuais que possibilite envio de alertas caso alguma anomalia seja diagnosticada na máquina rotativa.
- Validar os resultados obtidos na classificação das anomalias utilizando os dados coletados pelo dispositivo.
- Verificar, através de estudos de campo, para quais tipos de máquinas rotativas o equipamento desenvolvido se aplica.

1.3 Contribuições deste trabalho

As principais contribuições deste trabalho consistem em:

- Apresentar uma ferramenta desenvolvida, testada e validada, capaz de mensurar a vibração de máquinas rotativas por meio de um acelerômetro triaxial, processar o sinal desta vibração através de um cálculo da Transformada Rápida de Fourier (*FFT - Fast Fourier Transform*) da vibração medida nos eixos X e Y, calcular as seis primeiras principais harmônicas através da referência da leitura da velocidade da máquina (feita pelas entradas analógica e digital disponibilizadas na placa);
- Permitir o envio dos dados processados para um ambiente de computação em nuvem por meio de comunicação sem fio (Wi-Fi);
- Compartilhar todo o projeto de hardware e software do dispositivo (*opensource*).

1.4 Organização da Tese

Para uma melhor organização do trabalho, este foi dividido em oito capítulos. No Capítulo 2 é apresentada uma revisão bibliográfica, abordando os principais artigos relacionados ao monitoramento da vibração utilizando acelerômetros, as principais tecnologias e características IoT disponíveis atualmente. O levantamento de dispositivos similares disponíveis comercialmente e com patentes reconhecidas.

Uma abordagem sobre os principais conceitos de vibração e as principais falhas associadas à vibração é feita no Capítulo 3. Também destaca-se onde é mais apropriada a instalação de sensores de vibração em máquinas rotativas e o impacto que a análise de vibração tem na manutenção preditiva. No Capítulo 4 um embasamento matemático da vibração e as formas de análise é feito. Além disso, é argumentada a utilização da análise espectral utilizando a Transformada Rápida de Fourier e quais são os requisitos para que este algoritmo possa ser implementado no sistema.

No Capítulo 5, um levantamento sobre as tecnologias envolvidas no projeto foi elaborado, apresentando um estudo sobre os acelerômetros e os principais tipos disponíveis e a aplicabilidade de cada tipo para cada necessidade. Também discute-se sobre a escolha do hardware de processamento e suas principais características. É abordado o panorama atual da indústria na era 4.0, o crescimento de soluções IoT, as principais tecnologias de comunicação sem fio para a indústria e as plataformas de computação em nuvem mais populares.

No Capítulo 6, é apresentado todo o desenvolvimento do dispositivo. As especificações de projeto são apresentadas e a partir disso os diagramas eletrônicos foram desenvolvidos. Uma vez validado o circuito eletrônico, o projeto da placa foi prototipado. Foram necessários dois protótipos até a versão final validada. O fluxograma do código fonte e sua estrutura e principais detalhes também são abordados.

No Capítulo 7, é apresentado o desenvolvimento da bancada de testes. Trata-se de um sistema controlado onde diversas anomalias foram inseridas, como por exemplo, desbalanceamento, carga e obstrução. Foram coletados dados do sistema em perfeito funcionamento e as anomalias foram inseridas uma a uma e depois em conjunto.

No Capítulo 8, todos os resultados obtidos durante a elaboração desse trabalho são apresentados, desde a utilização de uma placa de desenvolvimento inicial com o acelerômetro adotado já embarcado até a placa final desenvolvida. Os resultados deram origem a alguns artigos que foram submetidos, conforme apresentado na seção *Artigos Publicados*. O aprendizado adquirido e as contribuições acadêmicas e industriais concluem esta tese em seu capítulo final.

2 Revisão Bibliográfica

Neste capítulo será apresentado o estado da arte dos assuntos relevantes ao desenvolvimento dessa tese. As pesquisas foram direcionadas à análise e medição de vibração e diagnóstico de falhas, comunicação sem fio e dispositivos existentes.

2.1 Monitoramento da Vibração

Diversos trabalhos apresentam a utilização da análise de vibração para diagnóstico de falhas, dentre eles podemos destacar:

- Em Marcal *et al.* (2000) é apresentada a implementação de um sistema de detecção de falhas através da análise espectral da vibração. Neste artigo, as baixas frequências constituíam as variáveis de interesse e a aquisição do sinal de vibração se deu em uma bancada de testes utilizando um acelerômetro ADXL202. Um software foi desenvolvido em Matlab para coletar e analisar estes sinais através de um sistema Fuzzy.
- Em Tzeng (2018) é apresentado um sistema de detecção e análise do sinal de vibração em turbinas eólicas baseado em uma rede LoRa sem fio. A análise é baseada na FFT do sinal coletado que é transmitida para um servidor LoRa que pode ser acessado remotamente. Neste trabalho, a avaria nos rolamentos da turbina foi avaliado.
- Em Mo *et al.* (2017) é apresentado um sistema de monitoramento de vibração baseado em análise do espectro da vibração medida em um fuso de alta velocidade. As características mais significativas do espectro da vibração são transmitidas a uma plataforma em nuvem utilizando uma comunicação wireless. Este dispositivo foi capaz de detectar vibrações normais e anormais no fuso de alta velocidade e serve como indicativo para manutenção preditiva. O acelerômetro utilizado é do tipo piezoelétrico de grande dimensão.
- Em Upadhye *et al.* (2015) é apresentado um sistema que utiliza um acelerômetro MEMS MMA7455L comunicando com um microcontrolador Freeduino e utilizando um Xbee para transmissão wireless dos dados. A unidade de monitoramento e a unidade de sensoriamento se comunicam através de módulos de RF do Xbee. O sistema de monitoramento foi feito no LabView onde foram adquiridos os sinais dos eixos X, Y e Z no domínio do tempo e transmitidos via protocolo ZigBee. Neste sistema, o motor estava girando a 1.425 RPM.

- Em Mohanty *et al.* (2014) é apresentado um acelerômetro capacitivo ADXL210 como sensor de vibração cujos dados de vibração dos três eixos são transmitidos através de uma rede ZigBee. Neste trabalho, os autores confirmam que o dispositivo foi capaz de medir a vibração em uma máquina rotativa em diversas velocidades de teste.
- Em Sabato *et al.* (2017) é apresentado um levantamento dos diversos dispositivos publicados na literatura para aquisição de sinal de vibração e transmissão desses dados via comunicação sem fio. Os dispositivos são dedicados ao monitoramento de estruturas.
- Em Patil e Gaikwad (2013) é apresentado um sistema de análise de vibração utilizando FFT. Os dados são coletados utilizando uma DAQ (Data Acquisition) conectado a um computador. A máquina foi testada em 900 RPM e foram inseridas falhas de desbalanceamento, falhas de fixação da base do motor e defeitos de engrenagem (neste caso a 1.200 RPM). Foram medidos os harmônicos de até terceira ordem. Os resultados obtidos podem ser utilizados na manutenção preditiva e foram capazes de diferenciar os tipos de falhas inseridas individualmente no sistema.
- Em Chen e Li (2017), uma técnica de diagnóstico de falhas em máquinas rotativas utilizando os sinais de vibração baseada em redes neurais é apresentada. Neste trabalho, a análise também é feita utilizando o espectro do sinal de vibração medido. Cinco falhas são inseridas no motor: desalinhamento, desbalanceamento, fricção, fixação da base e lubrificação. O método de aprendizado proposto pelos autores foi capaz de diagnosticar com 98% de precisão todas as falhas inseridas.

Com base nestes trabalhos relacionados, constata-se que este assunto possui relevância e vem sendo estudado de forma crescente nos últimos anos, corroborando com a temática deste trabalho.

2.2 Comunicação sem fio para IoT

Diversos protocolos de redes sem fio estão disponíveis no mercado. Dentre eles, podemos destacar (VIDAL, 2017), (SANTOS *et al.*, 2016):

- *LoRaWAN*: desenvolvido em aplicações para redes WAN (*Wide Area Network*), o LoRa foi projetado para prover baixo consumo de energia na comunicação e oferecer características específicas para segurança em comunicação M2M (*Machine to Machine*). O LoRa WAN tornou-se um protocolo muito popular.

- *Sigfox*: é uma alternativa intermediária entre Wi-Fi e redes de longo alcance (como 3G, 4G, etc). Utiliza a banda ISM (*Industrial, Scientific and Medical Band*). A tecnologia da Sigfox é UNB (*Ultra Narrow Band*) e seu alcance está entre 30 a 50 km.
- *NFC (Near Field Communication)*: é uma tecnologia para troca de dados entre dois dispositivos. É uma extensão da tecnologia RFID (*Radio-Frequency IDentification*) é especialmente interessante para *smartphones*. Seu alcance é de 10 cm.
- *Wi-Fi*: é amplamente difundido e conhecido. Devido a este sucesso, é uma opção muito utilizada por projetistas de sistemas IoT. Oferece velocidades de até 1 Gbps, operam no padrão 802.11, porém possuem um consumo de energia um pouco maior. Seu alcance está em torno de 50 m.
- *Zigbee*: é um dos protocolos mais utilizados em ambientes industriais. Baseado no padrão IEEE 802.15.4 e opera na faixa de 2.4 GHz. Seu alcance vai de 10 a 100 metros com taxa de transmissão máxima de 250 kbps.
- *Bluetooth*: outro protocolo largamente conhecido, o Bluetooth possui uma versão BLE (*Bluetooth Low Energy*) cujo foco está na IoT. De acordo com as especificações de seus circuitos integrados, seu alcance varia entre 10 m (2,5 mW de consumo) e 100 m (100 mW de consumo). A versão mais moderna é a 5.0 e pode chegar a 240 m de alcance com uma taxa de transmissão de 50 Mbits/s.

Um número crescente de pesquisas relacionadas à *Internet of Things* evidenciam o interesse do público em geral por esta tecnologia, conforme apresentado na Figura 2. Isso deve-se ao fato do termo ter ganho muita popularidade e ao crescimento das expectativas sobre IoT, ultrapassando a pesquisa pelo termo *Wireless Sensor Networks* em 2.010. A IoT foi identificada como uma tecnologia emergente em 2.012 (GARTNER, 2015).

Com base nesta pesquisa, conclui-se que é crescente a pesquisa de protocolos de comunicação para os mais diversos tipos de aplicação. Portanto, desenvolver um dispositivo que apresente uma destas tecnologias de comunicação é fundamental atualmente. O dispositivo desenvolvido contempla comunicação sem fio, seguindo as tendências e tecnologias consideradas mais atrativas.

2.3 Dispositivos de Medição da Vibração Existentes

A seguir serão apresentados alguns dispositivos comerciais, seus respectivos custos e suas principais características:

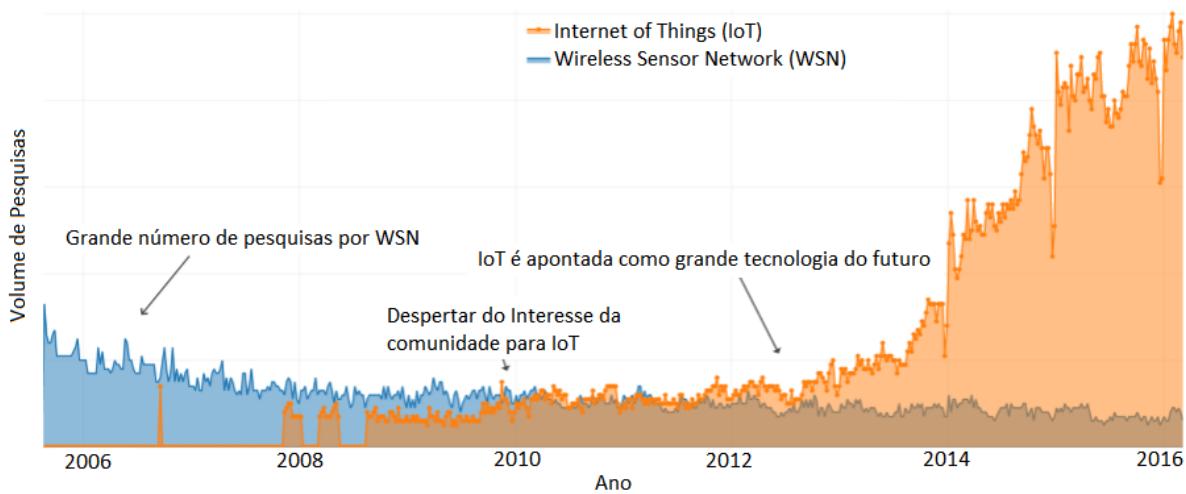


Figura 2 – Volume de pesquisas efetuadas no Google sobre *Wireless Sensor Networks* e *Internet of Things*. Fonte: Adaptado de Santos *et al.* (2016)

- VM220 Portable Vibration Meter: Custa aproximadamente US\$ 785. Variação de aceleração de 200 g. Lê os valores RMS (*Root Mean Square*), pico a pico e fator de crista. Apresenta os dados em um display. Tem dimensões de 133x78x28 mm e é apresentado na Figura 3a (INSTRUMENTS, 2018c).
- VM-6360 Hand Held Vibration Meter: Custa aproximadamente US\$ 380. Variação de aceleração de 0,02 a 20 g. Lê os valores True RMS, pico a pico, RPM (Rotações por Minuto) e pode ser conectado a um computador para apresentar os gráficos das medições e é apresentado na Figura 3b (INSTRUMENTS, 2018b).
- HS-620 & HS-630 Vibration Meter Kits: Custa aproximadamente US\$ 1.015 é apresentado na Figura 3c (INSTRUMENTS, 2018a).
- Sensor de aceleração VSA101 e Eletrônica de Diagnóstico VSE002. Custam aproximadamente US\$ 1.460. Variação de aceleração de 3,3 g. Sensor capacitivo. Faixa de frequência de até 1.000 Hz. Possui duas entradas analógicas, opera em 4 a 20 mA, comunicação Ethernet crossover, peso de 238,4 g + 53,2 g (módulo + acelerômetro), totalizando 291,6 g, memória interna, 12 bits de resolução, armazena até 1 minuto, é apresentado na Figura 3d e Figura 3e (ELECTRONICS, 2018a), (ELECTRONICS, 2018b).
- Medidor de vibração Fluke 805: custo aproximado de US\$ 2.200. Faixa de sensibilidade de frequência de 10 Hz a 1.000 Hz. Faz as medições e grava no próprio dispositivo para análise externa posterior, é apresentado na Figura 3f (FLUKE, 2018), (LIVRE, 2018).

- Medidor de Vibração 3 Eixos Tecnoferramentas Vm-6380: custo aproximado de US\$ 1.090. Faixa de frequência 10 Hz a 1 kHz, 10Hz a 10 kHz utilizando um acelerômetro piezoeletrico de 3 Eixos, é apresentado na Figura 3g (WALMART, 2018).
- Medidor de Vibração Omega HHVB82: custo aproximado de US\$ 1.435. Faixa de frequência entre 10 Hz a 1.000 Hz; Peso: 274 g e 38 g do sensor. Apenas realiza a leitura da vibração e é apresentado na Figura 3h (OMEGA, 2018).
- OneProd Falcon SMART: custo aproximado de US\$ 17.000 a US\$ 25.000. Comunicação sem fio. Sensor tri-axial. Grava os dados na unidade. Possui um pacote de análise de dados. Variação de frequência: 50 a 80 kHz (Na versão básica é de até 2.000 Hz). Realiza análise no domínio do tempo e espectral. É limitado a operar em máquinas com velocidade de 120 a 12.000 RPM e é apresentado na Figura 3i (ONEPROD, 2018).



Figura 3 – Instrumentos de medição e análise de vibração comerciais pesquisados

A Tabela 1 apresenta um resumo dos custos dos dispositivo comerciais listados. Os dispositivos comerciais apresentados possuem um custo mínimo de US\$ 380. A solução apresentada neste trabalho tem como objetivo apresentar um custo final inferior a US\$ 35, ou seja, menos de 10% do valor do dispositivo mais barato. Um dispositivo de baixo custo e

Tabela 1 – Custos dos principais dispositivos comerciais

Instrumento Comercial	Custo por unidade
VM220 Portable	US\$ 785
VM-6360 Hand Held	US\$ 380
HS-620 & HS-630	US\$ 1.015
VSA101 & VSE002	US\$ 1.460
FLUKE 805	US\$ 2.200
Tecnoferramentas VM-6380	US\$ 1.090
Omega HHVB82	US\$ 1.435
OneProd Falcon SMART	mais de US\$ 17.000

que possua funcionalidades similares aos comerciais existentes, é um atrativo para empresas praticantes da manutenção preditiva e engajadas nas tecnologias da indústria 4.0.

2.4 Dispositivos similares com patente reconhecida

Com o objetivo de avaliar o ineditismo do dispositivo desenvolvido, uma busca detalhada por patentes registradas com as características desejadas foi feita com auxílio da Inova. As patentes que mais se aproximaram foram as seguintes:

- *Data acquisition unit for remote monitoring system and method for remote monitoring* (HUNT, 2000): trata-se de um sistema de aquisição de sinais dedicado, onde qualquer tipo de dado de processo pode ser coletado. O armazenamento de dados é feito no computador local. O dispositivo envia os dados ao computador via protocolo de comunicação *RS/Ethernet* (via cabo) e o computador permite enviar os dados por uma conexão à internet. Não possui acelerômetro e sim, uma entrada disponível para a instalação de um acelerômetro (*dynamic data sensors*). A placa de expansão do dispositivo apresenta um algoritmo de análise de sinal (FFT). Esta placa de expansão é instalada em algum *slot* do computador local, ou seja, a placa não faz o processamento local da FFT e sim, faz o processamento remotamente. A diferença relevante entre este dispositivo e o apresentado neste trabalho consiste na capacidade do dispositivo proposto em conectar-se à internet e enviar os dados a qualquer repositório disponível e previamente configurado.
- *Machine condition indication system* (BOUSE *et al.*, 2005): os dados medidos são transmitidos via rede de comunicação industrial para um centro de controle onde são armazenados e consultados. Também trata-se de um dispositivo que habilita a conexão de sensores de vibração, porém não os tem embutidos no sistema. A comunicação dos

transdutores é feita via cabo. A placa de comunicação pode ser adaptada para comunicação industrial sem fio.

- *System for diagnosing facility apparatus, managing apparatus and diagnostic apparatus* (YOSHIE *et al.*, 2004): trata-se de um sistema que analisa os dados enviados para ele, e tem a capacidade de realizar diversas análises, como FFT, wavelets, tudo no computador local. Utiliza dados recebidos de uma placa de aquisição de sinais genérica. A chave da invenção está no software.
- *Distributed diagnostic system* (QUIST *et al.*, 1998): é apresentado um dispositivo capaz de medir diversas variáveis do processo, inclusive a vibração através de um acelerômetro embutido. Este armazena localmente os dados coletados e envia para o computador os dados processados (no domínio do tempo e no domínio da frequência (FFT) do sinal). Este talvez seja o invento mais parecido com o que apresenta-se neste trabalho. Utiliza um modelo já treinado localmente para predizer a saúde da máquina. É possível fazer comunicação sem fio através de um módulo transceptor de RF utilizando o protocolo HART, adaptando este módulo ao dispositivo.
- *Vibration monitoring system* (HASELEY; KIRKPATRICK, 1998): apresenta um sistema de monitoramento de vibração utilizando análise espectral, analisando a assinatura da máquina para predizer e detectar mudanças nas condições de operação da máquina. Ele possibilita uma análise espectral que relaciona frequência × velocidade. Ele não vem com os acelerômetros embutidos, eles são instalados à parte e conectados a placa de aquisição.

Frente às similaridades encontradas nessas soluções já registradas, infelizmente não foi possível patentear o dispositivo proposto neste trabalho. Entretanto, a grande contribuição do dispositivo proposto está no baixo-custo em relação aos demais encontrados no mercado além de apresentar um sistema totalmente *opensource*, o que permite ao usuário customizar o código de acordo com suas necessidades.

2.5 Conclusões deste capítulo

Através deste levantamento do estado da arte, ficou evidente que a análise através da FFT é um caminho seguro no diagnóstico de falhas e que a transmissão dos dados via comunicação sem fio é uma tendência em crescimento nos últimos anos. Um diferencial significativo observado é que os sistemas estudados apresentam tecnologias de aquisição que necessitam de softwares licenciados, além de utilização de dispositivos de alto custo. Outra observação importante é que a utilização de métodos computacionais de diagnóstico de falhas vêm

sendo utilizados com sucesso e os métodos vêm sendo aperfeiçoados. Tais métodos utilizam a FFT do sinal de vibração como parâmetro de análise. Além disso, nota-se a tendência de que os dispositivos possuam comunicação sem fio e que os protocolos de comunicação para instrumentos estão em expansão. Também, evidenciou-se que algumas contribuições deste trabalho são inéditas, sendo elas: *open hardware*, *open software* e baixo custo. Será criado um repositório onde todas essas informações serão disponibilizadas.

3 Conceitos de Vibração e Falhas Associadas

O desgaste de diversos equipamentos muitas vezes é percebido através de sua vibração ou em função de sua mudança no padrão de vibração. Dessa forma, a análise de vibração é uma ferramenta de diagnóstico poderosa, auxiliando na detecção de falhas em diferentes tipos de máquinas. Existem várias formas de se obter os dados de vibração para que, através de sua análise, possa ser possível identificar problemas específicos em máquinas rotativas (WOWK, 1991; RAO, 2014; BLOCH; GEITNER, 1999). Algumas técnicas incluem:

- Amplitude *versus* frequência;
- Amplitude *versus* tempo;
- Amplitude *versus* frequência *versus* tempo;
- Amplitude *versus* frequência *versus* velocidade;
- Forma de onda no tempo;
- Padrões de órbita;
- Amplitude e fase *versus* velocidade, e
- Análise de fase (movimento relativo).

Dentre as técnicas citadas, a mais importante é a amplitude *versus* frequência (BLOCH; GEITNER, 1999). Quando uma análise do histórico de vibração de uma máquina é realizado, é possível aprender muito sobre seu comportamento. A maioria das falhas podem ser observadas analisando o histórico de vibração da máquina, associando cada tipo de mudança a uma variável que impacta em seu aumento de vibração (RAO, 2014; BLOCH; GEITNER, 1999). Por exemplo, a substituição de acoplamentos e rolamentos pode aumentar a probabilidade de desbalanceamento e desalinhamento em máquinas rotativas. Até mesmo mudanças como temperatura ambiente, carga, velocidade, pressão, podem causar mudanças significativas no padrão de vibração (RAO, 2014). Seja qual for a causa, quando a vibração da máquina aumenta, em geral, é devido ao desgaste de componentes, deterioração mecânica ou mudanças estruturais.

O gráfico em cascata (*waterfall*) é outra técnica bem conhecida. Ela utiliza um gráfico tridimensional em função da amplitude, frequência e velocidade, conforme apresentado na

Figura 4a. Essa ferramenta de análise é uma sequência de análise espectral da vibração para cada velocidade medida na máquina. Uma forma de analisar a densidade da amplitude do gráfico *waterfall* é convertê-lo em um espectrograma, conforme apresentado na Figura 4b. A Figura 5 apresenta uma síntese de algumas falhas relacionando-as a algumas partes de um motor.

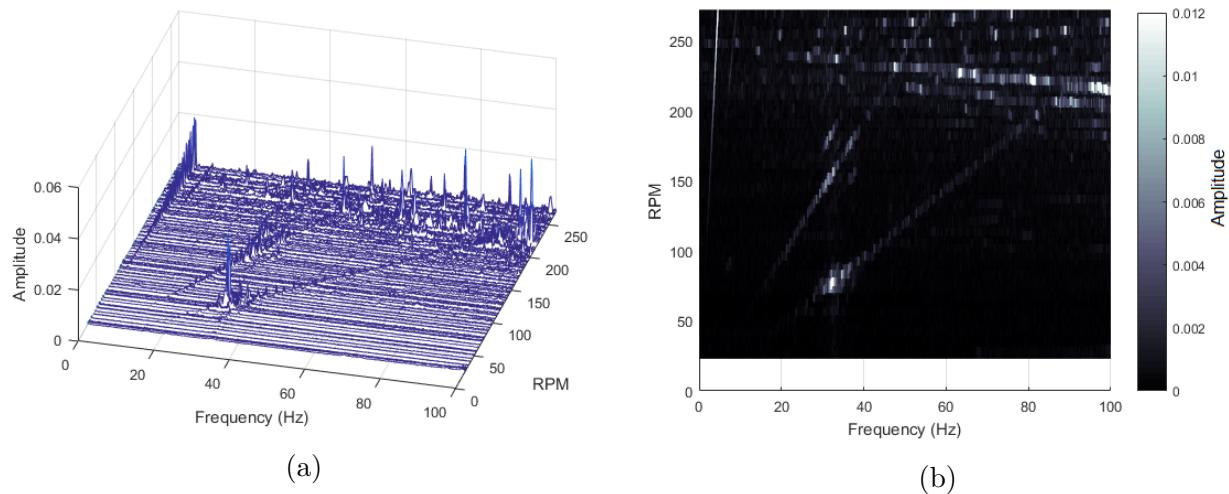


Figura 4 – Representação gráfica da amplitude do espectro da vibração utilizando: (a) waterfall (b) espectrograma. Fonte: Autoria própria.

Cada máquina possui um comportamento característico de vibração que depende de suas características construtivas. Tal padrão permite identificar o comportamento de cada máquina em condições normais de operação ao longo do tempo, como uma espécie de *assimilatura*. Caso haja mudança nesse padrão, é possível avaliar qual é a causa desse aumento na vibração. Por exemplo, se for um rolamento entrando em desgaste, é possível verificar esta tendência e prever que futuramente o rolamento irá se danificar por completo, causando uma parada inesperada e gerando custos indesejados no processo. Através dessa previsão, o responsável pela manutenção pode elaborar um plano de manutenção e realizar as devidas correções/ajustes antes que a falha ocorra, reduzindo custos e aumentando a produtividade (WOWK, 1991; RAO, 2014).

Uma vez que os dados da vibração são coletados, uma forma de análise é comparar a leitura atual com a leitura de características típicas de cada problema. A comparação de frequência baseada também na velocidade da máquina é capaz de identificar cada problema específico. A Tabela 2 relaciona as avarias e suas características em função de algumas variáveis da vibração medida. Cada problema está diretamente relacionado à velocidade de rotação que é um múltiplo da frequência de vibração. Esta comparação na maioria das vezes é suficiente para identificar a provável causa do problema (MAIS, 2019; BLOCH; GEITNER, 1999). As principais falhas relacionadas na Tabela 2 serão detalhadas nas próximas seções.

Tabela 2 – Frequências de Vibração e Suas Causas Prováveis. Fonte: autoria própria através de dados compilados de Mais (2019) e Bloch e Geitner (1999)

Frequência em relação à velocidade	Causa mais provável
$1 \times \text{RPM}$	Desbalanceamento Eixo Encurvado
$2 \times \text{RPM}$	Desalinhamento
de $2 \times$ até $10 \times \text{RPM}$	Desgastes mecânicos (os mais comuns são diagnosticados em até $6 \times \text{RPM}$)
Menor que $1 \times \text{RPM}$	Falta de lubrificação
Síncrona com a frequência AC da rede	Problemas Elétricos
$2 \times$ Velocidade Síncrona	Pulsos de torque
$3 \times, 4 \times \text{RPM}$ ou mais	Engrenagens Forças Hidráulicas Forças Aerodinâmicas
Altas Frequências	Rolamento e elementos de rolamento

Através da análise espectral da vibração é possível realizar diversos tipos de diagnósticos, podendo comparar cada componente harmônica entre pares de máquinas e definir variações aceitáveis. Entretanto, métodos computacionais de análise vem sendo utilizados cada vez mais como ferramentas que auxiliam a obtenção de um diagnóstico mais preciso e assertivo. Nesses métodos, a variável mais indicada para análise é o espectro da vibração.

Se forem analisadas as características de vibração para o caso de desbalanceamento, por exemplo, será possível verificar um aumento considerável na amplitude de vibração em $1 \times \text{RPM}$. Em 90% dos casos isso é um indicativo de desbalanceamento (BLOCH; GEITNER, 1999). Diversos padrões de vibração também apresentam coeficientes que podem ser utilizados como guias para avaliar a severidade de cada índice de vibração medido no equipamento, como ISO10816-1:1995 (1995) e ABNT-NBR:10082 (2011). Neste tipo de análise é medido o valor RMS da vibração, o valor medido é comparado com valores tabelados obtidos nas normas. Embora esta análise seja simples de implementar, ela pode ser insuficiente para alertar sobre danos iminentes às máquinas (RAO, 2014). A Figura 6 apresenta uma tabela contida na ISO 10816 onde é possível observar os índices de vibração RMS permitidos para cada classe de equipamentos que operam de 600 a 12.000 RPM.

Entretanto, analisar a amplitude do espectro da vibração comparando com a velocidade de rotação é, sem dúvida, o método mais acertivo (RAO, 2014; BLOCH; GEITNER, 1999). Dessa forma, analisar o sinal do espectro de vibração torna-se necessário.

A maioria das máquinas rotativas produzem baixos níveis de vibração quando estão

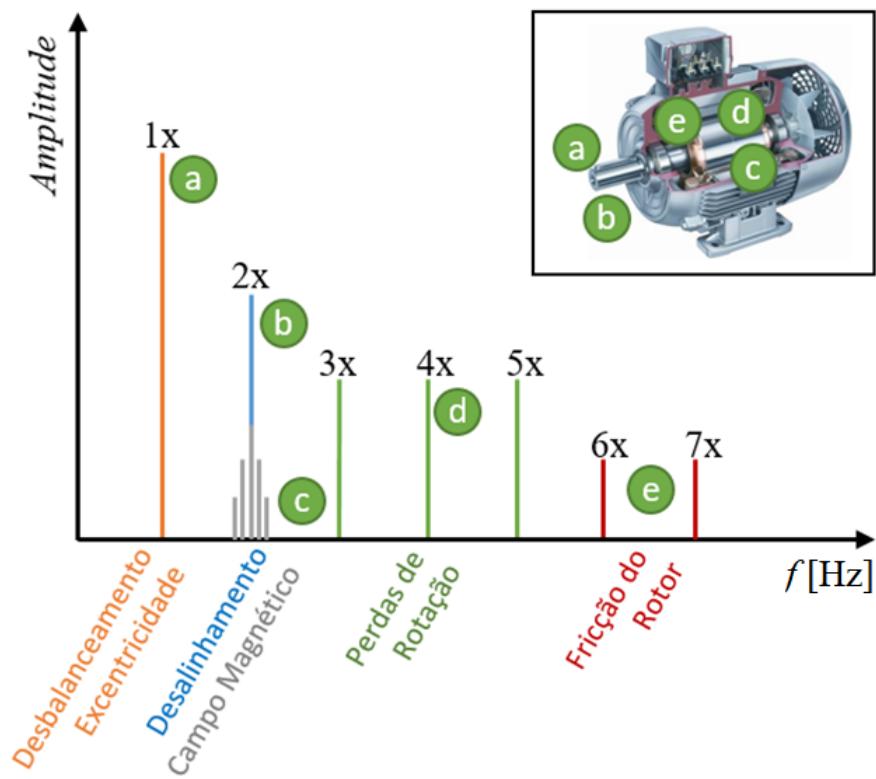


Figura 5 – Características do espectro da vibração para diversos tipos de falhas em máquinas rotativas

VIBRATION SEVERITY PER ISO 10816					
Machine		Class I small machines	Class II medium machines	Class III large rigid foundation	Class IV large soft foundation
	in/s mm/s				
0.01	0.28				
0.02	0.45				
0.03	0.71		good		
0.04	1.12				
0.07	1.80				
0.11	2.80		satisfactory		
0.18	4.50				
0.28	7.10		unsatisfactory		
0.44	11.2				
0.70	18.0				
0.71	28.0		unacceptable		
1.10	45.0				

Figura 6 – Classificação dos Níveis de Severidade de Vibração segundo a ISO 10816-3. Fonte: ISO10816-3:2009 (2009).

operando de forma adequada e sem falhas. Conforme o tempo de operação aumenta, todas elas ficam sujeitas a desgastes, fadiga, desalinhamento, entre outros modos de falha. Estes efeitos provocam aumento das folgas, desbalanceamento, trincas, etc. Estas falhas fazem com

que o nível de vibração da máquina aumente, desencadeando outros problemas mecânicos. O nível de vibração continua a aumentar com o passar do tempo, podendo resultar em falha ou avaria da máquina, reduzindo sua vida útil. Os tipos mais comuns de falhas que provocam aumento nos níveis de vibração são: eixos empenados e/ou excêntricos, desalinhamento, desbalanceamento, bombas com pás defeituosas, mancais defeituosos, peças frouxas e engrenagens defeituosas (RAO, 2014).

3.1 Diagnóstico de Avarias através da Análise da Vibração

Diversas avarias podem ser diagnosticadas utilizando análise do sinal de vibração, a seguir serão detalhadas algumas delas.

3.1.1 Desbalanceamento Mecânico

O desbalanceamento é caracterizado por uma distribuição assimétrica de massa em torno do eixo de rotação. Essa anomalia pode ser causada por diversos fatores, como: construção assimétrica, distorções permanentes devido ao estresse térmico ou esforços mecânicos, incrustações, desgaste ou corrosão, entre outros (COSTA, 2011).

Fazendo uma análise física do desbalanceamento, consideramos que um sistema que possui uma força centrífuga rotativa, causa uma vibração gerada pela excentricidade do centro de massa do rotor e sua rotação e dada por:

$$F_{cent} = mr\omega^2, \quad (3.1)$$

sendo que m é a massa de desbalanceamento em quilograma (kg), r é a distância entre o eixo do centro de gravidade e a massa em metros (m) e ω é a velocidade angular em radianos por segundo (rad/s). A força é equilibrada pela soma das reações que ocorrem no eixo de suporte da superfície circular. Então, se um dispositivo que faça a medição de tais forças for colocado próximo ao eixo, consequentemente acumulará neste ponto a maior concentração de vibração do sistema. Este efeito faz com que qualquer desequilíbrio existente na estrutura em rotação gere vibrações em diferentes frequências na estrutura do eixo. A Figura 7 ilustra o efeito da força centrífuga (HALLIDAY *et al.*, 2012). Sendo assim, é aconselhável que o dispositivo de medição seja inserido o mais próximo possível do eixo.

Observa-se na equação (3.1) que quanto maior a velocidade de rotação, mesmo para uma pequena massa de desbalanceamento, a força centrífuga resultante aumenta. Realizando uma análise no espectro da vibração, será observada uma elevação na amplitude de mesma frequência da velocidade da máquina devido ao efeito da força centrífuga.

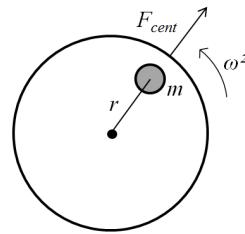


Figura 7 – Efeito da força centrífuga no eixo de uma estrutura em rotação com uma massa de desbalanceamento

A vibração devido ao desbalanceamento é caracterizada por uma elevação na amplitude do espectro da vibração em $1 \times \text{RPM}$. Tal aumento de amplitude é proporcional à quantidade de desbalanceamento existente (SOARES, 2014; BLOCH; GEITNER, 1999).

3.1.2 Desalinhamento do Eixo

O desalinhamento no acoplamento do eixo é uma condição tão comum quanto o desbalanceamento. O desalinhamento pode ser:

- Angular: as linhas dos dois eixos fazem um ângulo entre si, conforme mostrado de forma aumentada na Figura 8a;
- Paralelo: os eixos são paralelos, mas deslocados um do outro, conforme mostrado de forma aumentada na Figura 8b;
- Combinado: Os eixos encontram-se tanto desalinhados em ângulo como em paralelo, conforme mostrado de forma aumentada na Figura 8c.

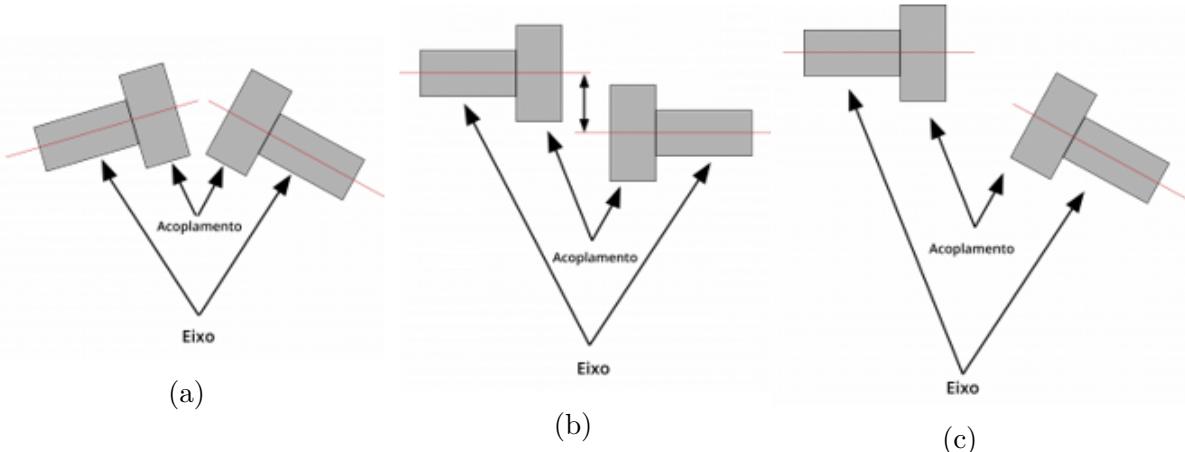


Figura 8 – Possíveis condições de desalinhamento: (a) angular (b) paralelo (c) combinado.
Fonte: Adaptado de Braga (2019)

Normalmente, o desalinhamento do eixo implica em um aumento na frequência de segunda ordem ($2 \times \text{RPM}$) e até de terceira ordem ($3 \times \text{RPM}$) quando as vibrações já atingiram um grande nível de severidade (ALMEIDA; ALMEIDA, 2011; BRAGA, 2019).

3.1.3 Ressonância

Todo corpo possui uma frequência natural própria relacionada à sua construção física. Quando a frequência de oscilação coincide com a frequência natural do sistema, ocorre uma condição conhecida como ressonância, e o sistema sofre oscilações elevadas que podem destruir a máquina (PETROBRÁS, 2000).

Uma maneira simples de descrever ressonância é a ampliação da vibração, que pode ser originária de várias fontes como desbalanceamento, desalinhamento, rolamento, entre outros. Todo corpo mecânico pode ressoar se forem excitadas em suas diversas frequências naturais. Um exemplo de ressonância é o diapasão que se for projetado para vibrar em uma frequência específica e for aproximado a uma máquina que está vibrando na frequência à qual o diapasão foi projetado, o diapasão irá ressoar (TRANTER, 2015). O diapasão é utilizado principalmente para afinar instrumentos musicais.

É comum confundir ressonância com batimento devido à forma de manifestação, já que o ruído de ambos é similar. A ressonância é permanente e o batimento é transitório, dando à ressonância um efeito destrutivo maior (PETROBRÁS, 2000). Através da análise espectral é possível identificar situações de ressonância que muitas vezes são a causa de deterioração prematura da máquina e seus componentes.

3.1.4 Rolamentos

Os rolamentos são utilizados como um dispositivo para troca de fricção de deslizamento entre duas ou mais partes e são fundamentais para que o equipamento trabalhe dentro do rendimento esperado (NORTON, 2010). Serve para manter firme um membro rotativo da máquina. São inseridos no mancal que dá suporte ao eixo (FRANCESCHI; ANTONELO, 2014). Elementos como poeira, sobrecarga, corrosão, defeito de montagem, temperatura e lubrificação deficiente podem ser fatores que influenciam na vida útil deste elemento.

O rolamento deixa de cumprir corretamente sua função com o passar do tempo e é possível notar seu desgaste ou avaria pelo aumento do ruído e vibração. Caso o rolamento apresente algum defeito, é possível detectar e diagnosticar as falhas analisando até o sexto harmônico do espectro de vibração da máquina, considerando como frequência fundamental a que representa a velocidade no eixo de rotação (Nivesrangsan; Jantarajirojkul, 2018). A

Figura 9 apresenta um exemplo de rolamento acoplado a um mancal de uma máquina rotativa (EUROPE, 2019).

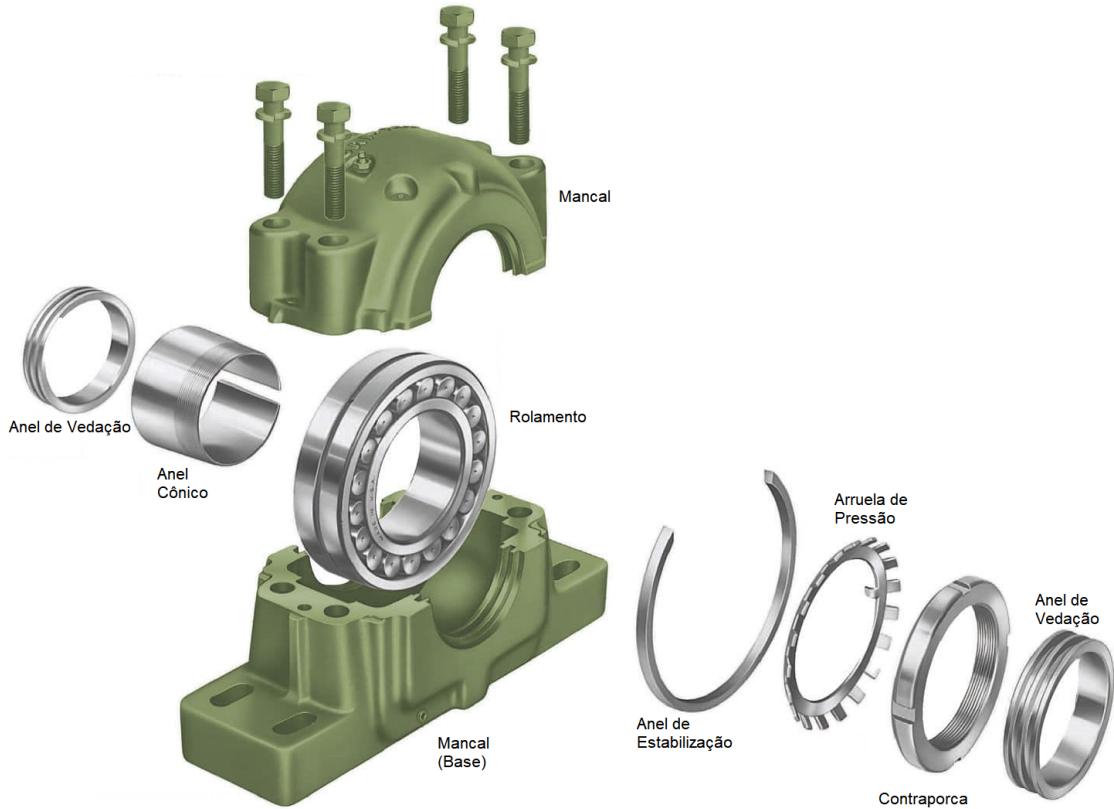


Figura 9 – Estrutura típica de montagem de mancal e rolamento. Fonte: Adaptado de (EUROPE, 2019).

3.1.5 Sobrecarga / Overload

Quando um motor é submetido a uma carga que excede seu torque nominal, é denominado que este está submetido a uma *sobrecarga*. Nessa condição, o motor opera sob uma corrente maior do que projetada, causando superaquecimento. Quando um motor trabalha superaquecido, sua vida útil diminui e dependendo do nível de sobrecarga, o motor pode fazer acionar o circuito de proteção de sua alimentação, provocando uma parada inesperada. Este tipo de situação também é indesejada em qualquer ambiente industrial (MOTORES, 2019). A sobrecarga também produz mudanças no padrão de vibração da máquina, sendo uma variável também passível de diagnóstico através da análise da vibração.

3.1.6 Falhas Elétricas em Máquinas Rotativas

O sistema de alimentação de energia elétrica de uma máquina rotativa também é importante na manutenção de sua vida útil. Quando um motor trifásico sofre a interrupção no fornecimento em uma de suas fases, o motor opera de forma irregular, aumentando sua vibração. Esta falha é conhecida como *falta de fase* e geralmente é provocada por uma queda em um dos fusíveis de alimentação (SOARES, 2014; MOTORES, 2019).

3.2 Seleção dos Pontos de Medição

Nem todo equipamento oferece condições favoráveis para inserção de sensores. Na escolha do ponto de medição é necessário levar em consideração alguns aspectos como: custo, acesso, segurança do operador, eficiência, entre outros. A fonte de vibração de qualquer máquina rotativa é junto aos elementos rotativos. Em diversas aplicações as partes rotativas são fixadas utilizando mancais. Dessa forma, as vibrações excitadoras irão do rotor para a carcaça passando pelo mancal (PETROBRÁS, 2000; RAO, 2014).

Para a manutenção preditiva é importante que os pontos de medição sejam sempre no mesmo local para referência. O melhor ponto de fixação dos sensores deve ser junto ao ponto rígido mais próximo da força de excitação, diminuindo assim a influência da impedância mecânica (PETROBRÁS, 2000; RAO, 2014).

3.3 Manutenção Preditiva

A manutenção tem grande impacto nos custos e consequentemente no balanço financeiro de uma empresa. Um indicador chamado OEE (*Overall Equipment Effectiveness*), que é proposto na metodologia TPM (*Total Productive Maintenance*) (NAKAJIMA, 1988). Com este indicador é possível verificar o quanto a empresa está utilizando de forma eficiente seus recursos disponíveis. Nessa metodologia, um dos indicadores é o de disponibilidade, que se refere ao tempo de trabalho comparado ao tempo disponível. As paradas podem ser classificadas como: não planejadas e planejadas. As paradas não planejadas influenciam diretamente no fator OEE (STEEGE, 1996). Frente a isso, diversas metodologias que auxiliam na previsão de falhas vêm sendo utilizadas. Dessa forma, uma manutenção pode ser programada de forma a corrigir tais problemas antes que a parada não-planejada ocorra e cause mais prejuízos.

Uma técnica utilizada para o monitoramento de equipamentos é a manutenção preditiva, que é baseada na tentativa de prever as condições futuras de cada máquina ou sistema

por meio de dados coletados ao longo do tempo e através de medições específicas. Análises desses dados são feitas com o objetivo de identificar tendências. Tais dados podem ser de temperatura, vibração, ultra-som e termografia, por exemplo (TELES, 2019b; ALBUQUERQUE, 2013).

O conceito já vem sendo utilizado na indústria desde a década de 1.970 e é uma ferramenta efetiva na produtividade. Na década de 1.990, teve um aumento expressivo, sendo que mais estudiosos se envolveram nessa área específica da manutenção. Entre os principais benefícios da manutenção preditiva destacam-se:

- Antecipação de serviços de manutenção no equipamento;
- Redução da probabilidade de desmontagem desnecessária;
- Aumento da disponibilidade de equipamentos;
- Aumento da vida útil, confiabilidade e desempenho de equipamentos;
- Redução das paradas de emergência;
- Determinação de paradas de produção programadas.

Segundo os padrões da RCM (*Reliability Centered Maintenance*), uma ferramenta analítica para um plano de manutenção adequado é a curva do Potencial Funcional (PF). Essa curva apresenta a performance do equipamento em relação ao seu tempo de funcionamento. Através dela, é possível identificar o intervalo PF que é o tempo entre a falha **potencial** e a falha **funcional**. A Figura 10 apresenta a curva PF. Nota-se que a análise de vibração é a segunda técnica mais rápida na identificação de mau funcionamento da máquina, perdendo apenas para a análise de ultrassom (Apêndice A).

Quanto mais tardar a correção da anomalia, maior será o custo do reparo. A curva de custos é inversamente proporcional à curva PF, conforme Figura 11. Pode-se concluir que na maioria dos casos, o custo da manutenção preditiva de qualidade é menor que o custo da parada não programada do equipamento. Fatores como indisponibilidade de peças de reposição, logística e pessoal qualificado impactarão ainda mais na disponibilidade e na capacidade produtiva do equipamento avariado.

3.4 Conclusões deste capítulo

Este capítulo apresentou uma análise da vibração em máquinas rotativas que desenvolvem vibrações periódicas para a maioria das avarias. Como a vibração tem característica

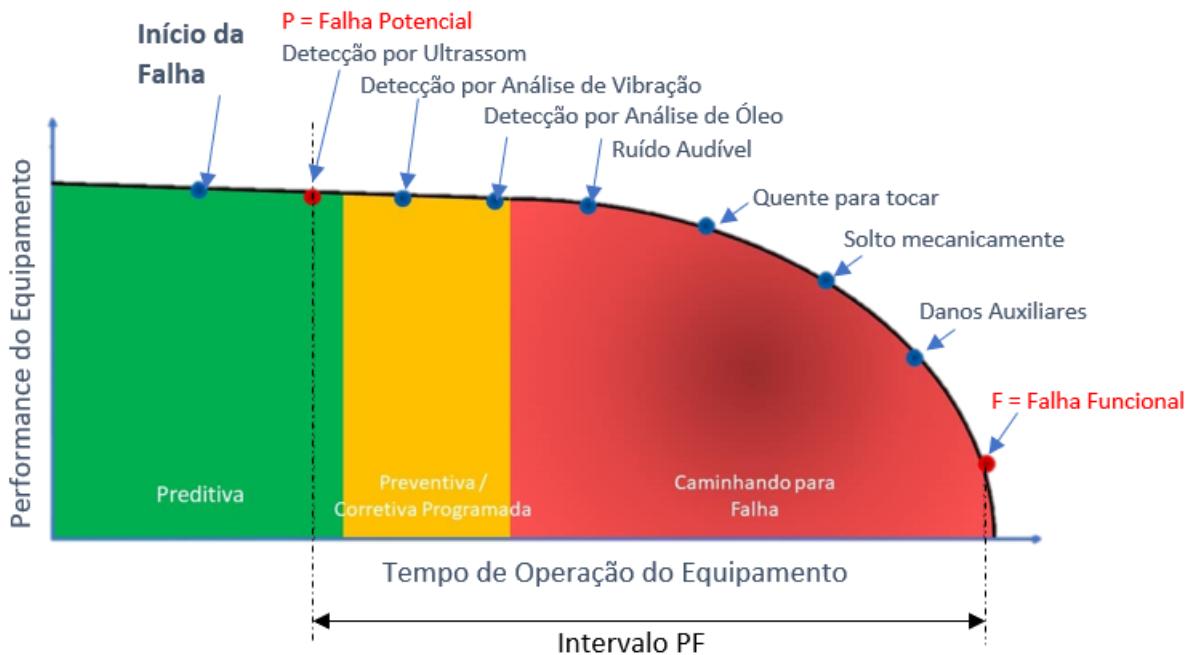


Figura 10 – Curva PF. Fonte: Adaptado de Teles (2019a).

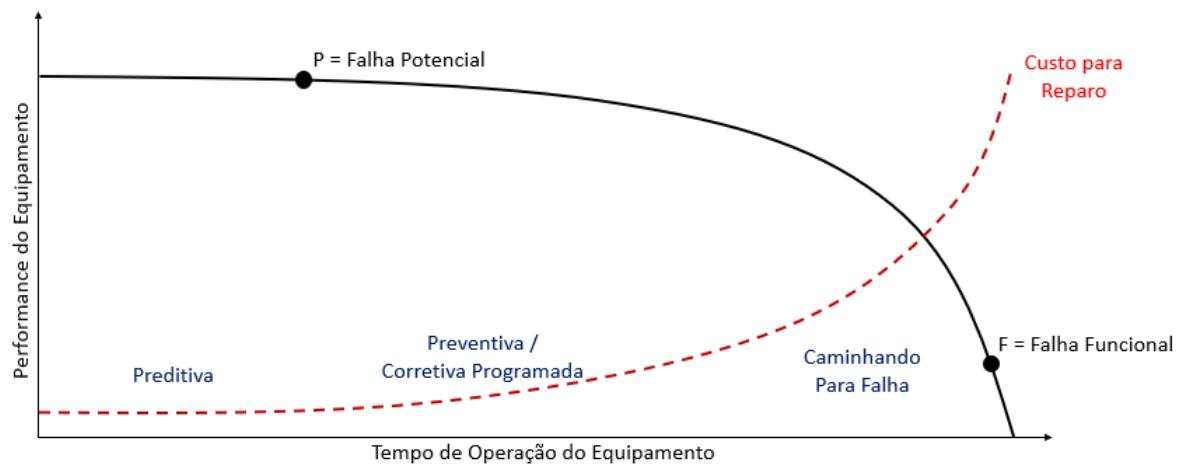


Figura 11 – Curva PF x Custos de Reparo. Fonte: Adaptado de Teles (2019a).

harmônica, realizar a análise do espectro de frequência da vibração é mais eficaz, pois é possível obter dados mais assertivos em relação a cada tipo de avaria. Alguns tipos de falhas e suas consequências no espectro de vibração foram apresentados, indicando que a maioria delas podem ser detectáveis antecipadamente através da análise de vibração.

Um panorama sobre a importância da manutenção e o impacto que ela representa na indústria foi apresentado, justificando a utilização de ferramentas que auxiliem no diagnóstico precoce de falhas futuras, possibilitando manutenções preditivas com menor impacto no processo produtivo.

4 Embasamento Matemático

A vibração é uma oscilação de um corpo sólido em relação ao seu centro de equilíbrio e está presente em diversas situações, como abalos sísmicos, instrumentos musicais, ação do vento, acústica, choques mecânicos, entre outros. A vibração está presente em nosso cotidiano, entretanto em algumas situações ela pode oferecer riscos ou representar algum tipo de problema. Quando uma estrutura é exposta a uma vibração externa cuja frequência coincide com sua frequência de oscilação natural, ocorre um fenômeno conhecido como ressonância resultando em deflexões excessivas (RAO, 2014).

Um sistema vibratório é um sistema dinâmico onde as excitações e as respostas são dependentes do tempo. A complexidade de um sistema vibratório e o grande número de variáveis que definem sua condição inicial tornam sua modelagem matemática inviável. Por este motivo, é utilizado um modelo físico simplificado, que considera apenas os elementos mais importantes.

Devido ao fato dos motores e geradores possuírem um elemento girante, as vibrações que eles originam são em sua maioria periódicas. Em máquinas rotativas, tem-se um movimento rotacional que é regido pelas leis cinemáticas do Movimento Circular Uniforme (MCU) (RAO, 2014). Ao registrar graficamente este movimento, tem-se um traçado senoidal dessa oscilação, que obedece às leis cinemáticas do Movimento Harmônico Simples (MHS), conforme apresentado na Figura 12.

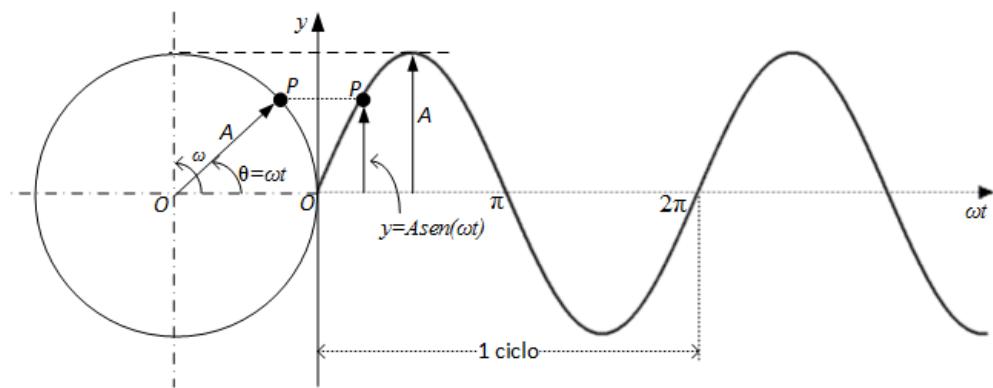


Figura 12 – Movimento harmônico simples com projeção do vetor girante

O movimento oscilatório pode repetir-se regularmente ou não. Se o movimento for repetido em intervalos iguais, denomina-se movimento periódico, sendo o mais simples dele o movimento harmônico. Esse movimento harmônico pode ser representado por um vetor \vec{OP} de magnitude A girando a uma velocidade constante ω (RAO, 2014). Se projetarmos

$\vec{X} = \vec{OP}$ sobre os eixos x e y , temos:

$$x = A \cos(\omega t) \quad (4.1)$$

e:

$$y = A \sin(\omega t) \quad (4.2)$$

Considerando que qualquer vetor \vec{X} no plano xy pode ser representado por um número complexo $\vec{X} = a + ib$, onde $i = \sqrt{-1}$ e a e b são os componentes x e y e também denominados componentes *real* e *imaginário*, respectivamente, é possível representar o vetor \vec{X} como:

$$\vec{X} = A \cos \theta + iA \sin \theta, \quad (4.3)$$

sendo que θ é o ângulo entre o vetor e o eixo x . Utilizando a representação por números complexos, o vetor girante \vec{X} pode ser escrito como:

$$\vec{X} = Ae^{i\omega t}, \quad (4.4)$$

sendo que a rotação do \vec{X} está em sentido anti-horário. A diferenciação do movimento harmônico resulta em:

$$\frac{d\vec{X}}{dt} = \frac{d}{dt}(Ae^{i\omega t}) = i\omega Ae^{i\omega t} = i\omega \vec{X} \quad (4.5)$$

e:

$$\frac{d^2\vec{X}}{dt^2} = \frac{d}{dt}(i\omega Ae^{i\omega t}) = -\omega^2 Ae^{i\omega t} = \omega^2 \vec{X} \quad (4.6)$$

Dessa forma, o deslocamento d do ponto P , a velocidade v e a aceleração a podem ser expressos como:

$$d = \operatorname{Re}[Ae^{i\omega t}] = A \cos(\omega t) \quad (4.7)$$

e:

$$v = \operatorname{Re}[i\omega Ae^{i\omega t}] = -\omega A \sin(\omega t) = \omega A \cos(\omega t + 90^\circ) \quad (4.8)$$

e:

$$a = \operatorname{Re}[-\omega^2 Ae^{i\omega t}] = -\omega^2 A \cos(\omega t) = \omega^2 A \cos(\omega t + 180^\circ) \quad (4.9)$$

As funções harmônicas podem ser somadas vetorialmente, conforme apresentado na Figura 13. Se $\text{Re}\vec{X}_1 = A_1 \cos(\omega t)$ e $\text{Re}\vec{X}_2 = A_2 \cos(\omega t + \theta)$ então a magnitude do vetor resultante \vec{X} é dada por:

$$A = \sqrt{A_1^2 + (A_2 \cos \theta)^2 + (A_2 \sin \theta)^2}. \quad (4.10)$$

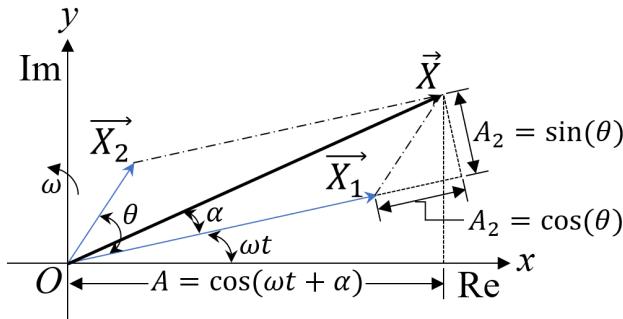


Figura 13 – Adição vetorial de funções harmônicas

Algumas definições e terminologia são úteis no trato do sinal de vibração, como:

- **Ciclo:** é uma vibração completa. Analisando a Figura 12, um ciclo trata-se do objeto P sair da posição O até 2π de forma cíclica.
- **Período:** é o tempo que o objeto P leva para concluir um ciclo. O período de oscilação (τ) é dado por:

$$\tau = \frac{2\pi}{\omega} \quad (4.11)$$

onde ω é a frequência angular e sua unidade é dada em radianos por segundo (rad/s).

- **Amplitude:** é o máximo deslocamento do corpo vibratório em relação à sua posição de equilíbrio.
- **Frequência de oscilação:** é o número de ciclos por unidade de tempo, e é dado por:

$$f = \frac{1}{\tau} = \frac{\omega}{2\pi} \quad (4.12)$$

e sua unidade é dada em Hertz (Hz).

- **Frequência natural:** se após uma perturbação inicial, um sistema continuar a vibrar por si próprio sem a ação de forças externas, esta frequência com que ele oscila é chamada de frequência natural.

- **Batimentos:** quando dois corpos estão oscilando em movimentos harmônicos de frequências, essas são somadas e o movimento resultante exibe um fenômeno denominado batimento. A frequência (δ) onde a amplitude aumenta gradativamente e depois diminui até desaparecer entre 0 e $2A$ é conhecida como frequência de batimento. A Figura 14 ilustra este fenômeno.

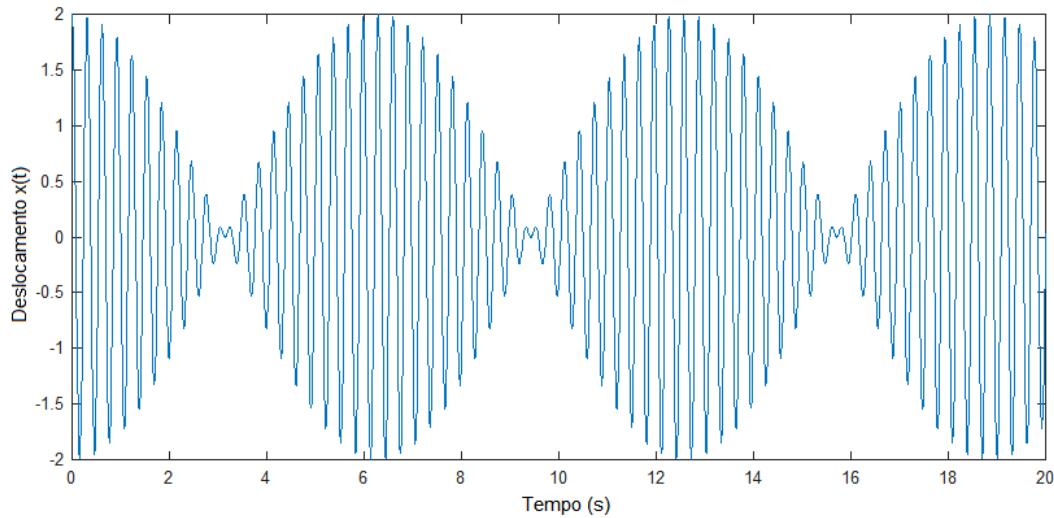


Figura 14 – Fenômeno de batidas

4.1 Análise Harmônica

Apesar do movimento harmônico ser periódico e relativamente simples de se tratar, o movimento de muitos sistemas vibratórios não são harmônicos, porém são periódicos. Qualquer função periódica pode ser representada por série de Fourier como uma soma infinita de termos de senos e cossenos, assim como apresentado na equação:

$$\begin{aligned} x(t) &= \frac{a_o}{2} + a_1 \cos(\omega t) + a_2 \cos(2\omega t) + \dots + b_1 \sin(\omega t) + b_2 \sin(2\omega t) \dots \\ &= \frac{a_o}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \end{aligned} \quad (4.13)$$

A série de Fourier também pode ser representada através de números complexos (RAO, 2014). Dessa forma, o $\cos(\omega t)$ 4.14 e $\sin(\omega t)$ 4.15 podem ser expressos como:

$$\cos(\omega t) = \frac{e^{i\omega t} + e^{-i\omega t}}{2} \quad (4.14)$$

e:

$$\sin(\omega t) = \frac{e^{i\omega t} - e^{-i\omega t}}{2i} \quad (4.15)$$

As funções harmônicas $a_n \cos(n\omega t)$ e $b_n \sin(n\omega t)$ são denominadas harmônicas de ordem n da função periódica $x(t)$. A harmônica de ordem n possui um período de τ/n . Essas harmônicas podem ser representadas como linhas verticais em um diagrama de amplitudes em relação a cada ordem de frequência $n\omega$ denominado *espectro de frequência* (RAO, 2014). Exemplos de funções representadas no domínio do tempo e no domínio da frequência são apresentadas na Figura 4.1 para as funções $x_1(t) = \sin(\omega t)$ e $x_2(t) = \sin(\omega t) + 0,5\sin(2 \cdot \omega t) + 0,25\sin(3 \cdot \omega t) + 0,125\sin(4 \cdot \omega t)$ onde nos dois casos a $f = 10$ Hz.

A Série de *Fourier* permite a representação de qualquer função periódica como séries infinitas de senos e cossenos, o que simplifica a visualização e manipulação de funções complexas. Essa expansão em harmônicas senoidais/cossenoidais, permite identificar quais são as frequências presentes no sinal, permitindo que a função originalmente obtida no domínio do tempo possa ser representada no domínio da frequência. Nas Figuras 15a e 15c temos funções representadas no domínio do tempo e nas Figuras 15b e 15d no domínio da frequência.

4.2 Transformada de Fourier

A transformada de Fourier decompõe uma função temporal em frequências, transformando a função temporal em uma função de valor complexo da frequência onde o valor absoluto representa a soma das frequências presentes na função original e o argumento complexo e a fase de deslocamento da base senoidal na frequência. A transformada de Fourier também é conhecida como representação no domínio da frequência de uma função temporal original. A transformada de Fourier é uma extensão da Série de Fourier. A transformada de Fourier de uma função $f = f(t)$ integrável é definida por:

$$\mathcal{F}\{f(t)\} \equiv F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt, \quad (4.16)$$

onde $F(\omega)$ é o sinal de $f(t)$ representado no domínio da frequência ω e $\omega \in \mathbb{R}$, onde $\omega = 2\pi f$ e f é a frequência do sinal.

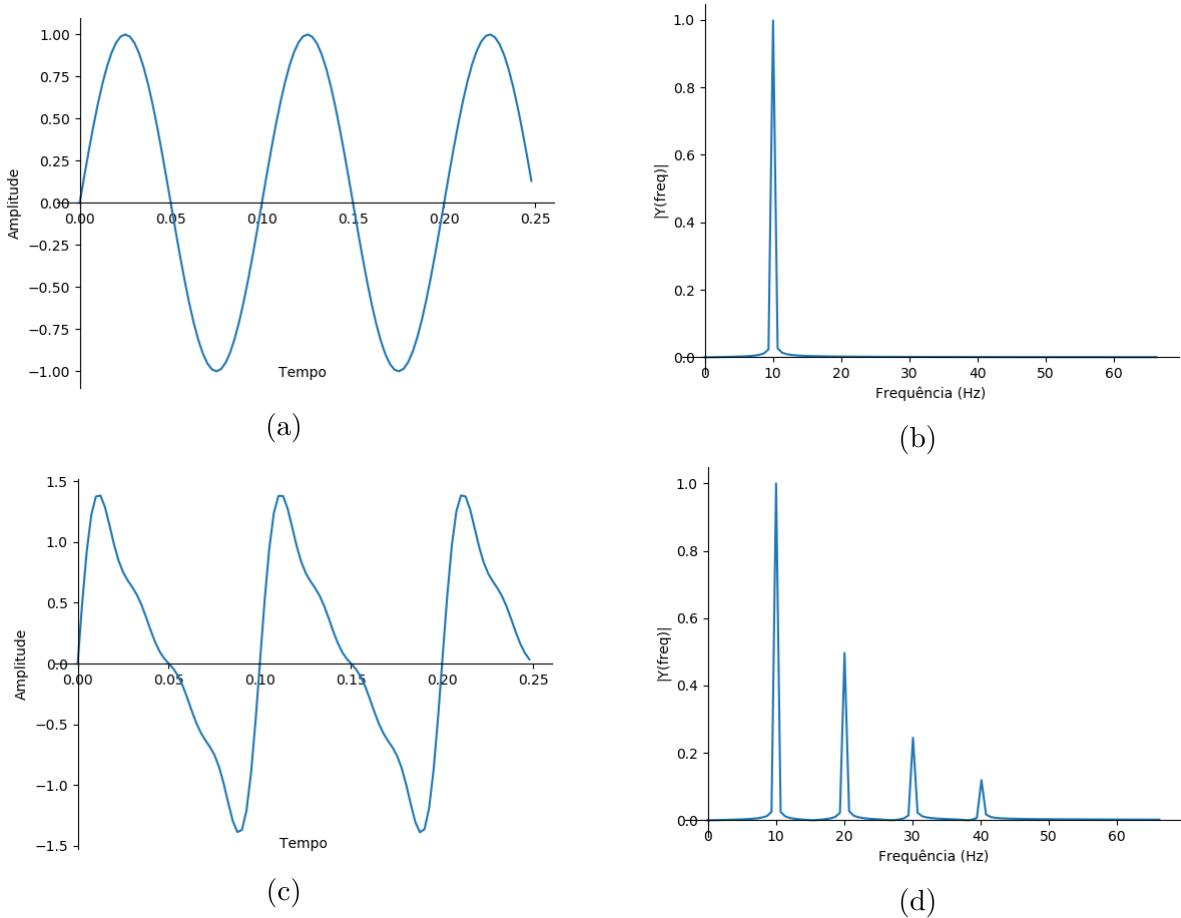


Figura 15 – Representação de funções no domínio do tempo e frequência: (a) $x_1(t)$ (b) DFT de $x_1(t)$ (c) $x_2(t)$ (d) DFT de $x_2(t)$

4.3 Sinais Discretos

A grande maioria dos sistemas físicos são de natureza analógica. Um sinal analógico é caracterizado por um sinal contínuo cuja variação em relação ao tempo é a representação proporcional de uma variável temporal, como por exemplo um termômetro de mercúrio. Entretanto, com o advento da eletrônica digital foi possível digitalizar estes sinais para processá-los computacionalmente através de microcontroladores e microprocessadores (OPPENHEIM; SCHAFER, 2009). Para digitalizar um sinal contínuo no tempo passamos por um processo de amostragem, onde este sinal é transformado numa sequência finita de pontos discretos, assim como apresentado na Figura 16. É possível observar na figura que o sinal discretizado não é igual ao sinal contínuo, sendo ele uma representação desse sinal, neste caso do exemplo da figura, uma representação grosseira.

Discretizar um sinal é vantajoso pois é possível adequar o tamanho do sinal a ser processado dentro da capacidade de memória do processador. A discretização de um sinal é

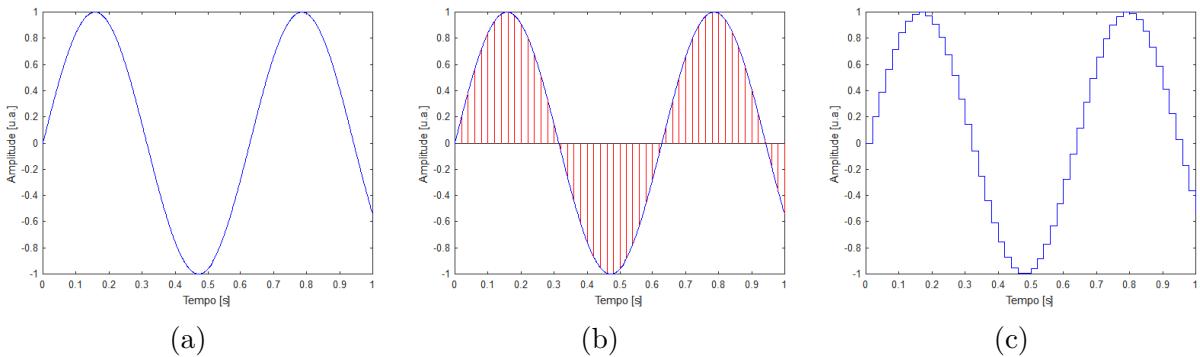


Figura 16 – Discretização do sinal contínuo: (a) sinal contínuo (b) processo de amostragem (c) sinal discretizado. Fonte: autoria própria

obtida através de três processos: amostragem, quantização e codificação. Para uma aquisição de sinal com qualidade é necessário levar em consideração fatores como: taxa de amostragem e resolução.

4.3.1 Transformada Discreta de Fourier

A Transformada Discreta de Fourier (TDF), também conhecida como DFT (*Discrete Fourier Transform*), é aplicada em sinais amostrados. Para o sinal $f(t)$ amostrado com intervalos regulares T , representado por (J.A.POMILIO; S.M.DECKMANN, 2019):

$$f(kT) = x_0\delta(t) + x_1\delta(t - T) + x_2\delta(t - 2T) + \dots + x_k\delta(t - kT), \quad (4.17)$$

onde $k = 0, 1, 2\dots$ e $\omega \in \Re$.

No caso de sinais amostrados, ocorre uma repetição desse espectro a cada $2\pi/T$. O espectro de interesse se encontra na faixa $-\pi/T < \omega \leq \pi/T$ e como o sinal é simétrico, basta conhecer o espectro na faixa de $0 \leq \omega \leq \pi/T$. Assim sendo, a condição suficiente para uma reconstrução a partir das amostras em uma taxa de amostragem f_s é $f_s > 2B$, onde B é o limite de banda, $2B$ é conhecido como taxa de Nyquist e $f_s/2$ como frequência de Nyquist. Em resumo, para a reconstrução do sinal é necessário uma taxa de amostragem duas vezes maior que a frequência máxima do sinal (OPPENHEIM; SCHAFER, 2009).

4.3.1.1 Transformada Rápida de Fourier

Conhecida como FFT (*Fast Fourier Transform*), esse é um algoritmo eficiente para se calcular a DFT de um sinal. A análise de Fourier converte um sinal do domínio do tempo para uma representação em domínio da frequência e vice-versa. Uma FFT calcula essa transformação fatorando a matriz da DFT em um produto de fatores esparsos. Dessa forma, é

possível reduzir a complexidade de calcular a DFT na ordem quadrática do número de dados. O algoritmo baseia-se no método de dobramentos sucessivos, onde é possível expressar a transformada de Fourier como:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux}, \quad (4.18)$$

onde $W_N^{ux} = e^{-j2\pi ux/N}$ e $N = 2^n$ com n inteiro positivo. Escrevendo $N = 2M$, com M inteiro positivo, temos:

$$F(u) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) W_{2M}^{ux}. \quad (4.19)$$

É possível desmembrar a equação 4.19 em duas parcelas, uma par e outra ímpar e assim computar a transformada de N pontos pela divisão da expressão original em duas partes. O código escrito em linguagem C para o cálculo da FFT utilizado neste trabalho está disponível no Apêndice B.

O algoritmo desenvolvido retorna os valores reais e imaginários de frequência e amplitude do sinal onde é passado a frequência de amostragem e o número de amostras. O teorema da amostragem de Nyquist-Shannon especifica que um sinal deve ser amostrado com uma frequência de amostragem de no mínimo o dobro da do sinal de interesse (OPPENHEIM; SCHAFER, 2009). Assim sendo, se o sistema coleta dados à uma frequência de 800 Hz, significa que irá amostrar com qualidade sinais de até 400 Hz.

O algoritmo que calcula a FFT trabalha com um número finito de amostras que deve ser da ordem de 2^n e quanto maior for o valor de n , mais lento é o algoritmo, porém, maior será a resolução dos resultados.

4.4 Conclusões deste capítulo

Neste capítulo foi apresentado como um sinal de vibração é interpretado como um movimento oscilatório e quais são as características desse sinal. Também apresentou-se o embasamento matemático para o cálculo do espectro do sinal de vibração. A forma de como efetuar o cálculo foi descrita, desde a análise harmônica e a transformada de Fourier até a análise de sinais discretos e como utilizar a transformada discreta de Fourier sobre eles.

O algoritmo mais utilizado para calcular a transformada discreta de Fourier, o FFT foi apresentado. A FFT está aplicada como um algoritmo embarcado no microcontrolador do dispositivo desenvolvido. Estes detalhes serão aprofundados no Capítulo 6.

5 Tecnologias Envolvidas

Para desenvolver o sistema, foi necessário realizar um levantamento das principais tecnologias utilizadas atualmente, analisando os aspectos técnicos e econômicos de viabilidade. A solução apresentada é baseada em um microcontrolador, comunicação sem fio e um acelerômetro MEMS. As análises de cada requisito serão apresentadas nas próximas seções.

5.1 Acelerômetros

Como apresentado no Capítulo 3, a vibração de um corpo é uma informação importante em diversos segmentos, necessitando ser medida e monitorada frequentemente. Normalmente a medição dessa grandeza é feita utilizando um acelerômetro.

Acelerômetros são dispositivos transdutores utilizados para medição de vibrações, detecção de impactos mecânicos, monitoramento de velocidade, detecção de inclinação e forças de aceleração. Existem muitos tipos de acelerômetros e os mais utilizados são os baseados em cristais piezoelétricos, entretanto estes são grandes e de difícil instalação (ANDREJASIC, 2008).

Geralmente eles são montados diretamente em contato com os elementos onde a vibração será medida, como por exemplo, em mancais de elementos rolantes, caixas de engrenagens, pás giratórias, entre outros. Também são instalados em locais onde há necessidade de medições de choque mecânicos. O grande benefício de utilizar acelerômetros é sua linearidade ao longo de uma extensa faixa de frequência.

Outro sensor utilizado para medir vibrações é a sonda de proximidade. Este tipo de transdutor opera sem contato direto, respeitando-se as distâncias mínimas do projeto de medição. Essas sondas são quase que exclusivamente utilizadas em medições de eixos em maquinários rotativos (MEDIÇÃO..., 2019; LALLY, 2019).

Os acelerômetros podem ser aplicados em diversas áreas, como por exemplo:

- Análise preditiva por vibrações: empregados geralmente para monitoramento da saúde estrutural de elementos de máquinas, como rolamentos, correias, bombas, entre outros, baseando esta análise no princípio da assinatura espectral de cada componente.
- Monitoramento sísmico: alguns acelerômetros de alta sensibilidade podem ser utilizados como sismógrafos, auxiliando no monitoramento de terremotos.

- Dispositivos eletrônicos: é comum serem aplicados em smartphones ou tablets para determinação da orientação de imagens, mudando automaticamente a posição conforme o usuário movimenta o dispositivo.
- Aplicações médicas: utilizados para estudo da cinética articular de tornozelos, pulsos, joelhos, quadris e demais articulações, auxiliando na medição de impactos em atletas de alta performance.

Alguns parâmetros de desempenho para a escolha de um acelerômetro também são relevantes, de modo que ele possa atender à demanda exigida para a aplicação escolhida. Os principais parâmetros a serem considerados na escolha de um acelerômetro são:

- Intervalo de medição: o intervalo dinâmico do acelerômetro deve ser maior do que o intervalo de amplitude do sistema;
- Sensibilidade: depende do nível do sinal que será medido, por exemplo, acelerômetros de alta sensibilidade costumam ser empregados em medições de baixa amplitude;
- Intervalo de frequência: deve ser selecionado de acordo com quais frequências deseja-se medir;
- Número de eixos: geralmente uniaxias ou triaxiais;
- Massa: deve ser menor que o objeto que se deseja monitorar;
- Condições ambientais: sua construção deve ser robusta suficiente para ser utilizada no ambiente em que será aplicada, que pode estar sujeito a intempéries como temperatura, umidade, oxidação, entre outros;
- Custo: opta-se por um dispositivo de melhor custo-benefício para a aplicação pretendida.

Existem diversos tipos de acelerômetros que utilizam diferentes tipos de princípios que são capazes de medir a aceleração. Dentre os tipos existentes, pode-se destacar: piezoeletrico, piezoresistivo, capacitivo, MEMS, efeito hall, magnetoresistivo e redes de Bragg em fibras ópticas. Cada um deles será descrito brevemente, e o acelerômetro MEMS será mais detalhado por ser o foco de estudo.

5.1.1 Piezoeletrico

Este tipo de acelerômetro emprega o efeito piezoeletrico de certos materiais para medir mudanças mecânicas dinâmicas. Seu princípio de funcionamento é baseado na segunda

lei de Newton ($F = m.a$). Quando uma força é exercida no acelerômetro, a massa sísmica carrega o elemento piezoelétrico. Tal força exercida pode ser observada na mudança da força eletrostática ou da tensão gerada pelo material piezoelétrico. Tal força pode ser classificada como: flexão ou compressão.

Os materiais utilizados com este propósito piezoelétrico são divididos em duas categorias: cristais e cerâmicos. Os cristais geralmente são do tipo monocristalino (quartzo). Os cerâmicos apresentam uma constante piezoelétrica mais alta e são mais baratos e os materiais mais utilizados são: titanato de bário, titanato zirconato de chumbo (PZT), metaniobato de chumbo, porém sua sensibilidade se degrada com o tempo. A Figura 17 apresenta um acelerômetro piezoelétrico típico. São capazes de medir frequências numa faixa de 1 Hz a 30 kHz, porém são tipicamente adequados para aplicações de alta frequência.

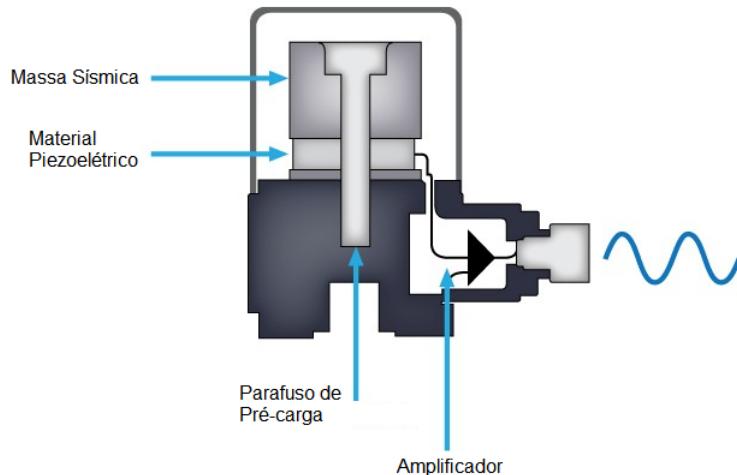


Figura 17 – Acelerômetro piezoelétrico típico. Fonte: Adaptado de MEDIÇÃO... (2019)

5.1.2 Piezoresistivo

No efeito piezoresistivo a resistividade do material é modificada quando este é submetido a tensões mecânicas. Quando ocorre uma deflexão no diafragma de material piezoresistivo (p.e. Silício), estas podem ser mensuradas, geralmente implementando dois ou quatro extensômetros de resistência elétrica (*strain gauges*), com meia ponte de *Wheatstone* ou ponte completa respectivamente, conforme exemplo apresentado na Figura 18. São adequados para medição de baixas frequências, operando de 0 Hz a 750 Hz.

5.1.3 Capacitivo

Baseiam-se no princípio da mudança de capacitância em resposta à mudança de aceleração. São formados por uma massa sísmica que se movimenta quando uma aceleração é

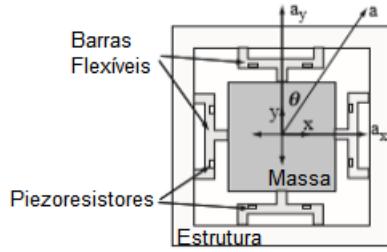


Figura 18 – Acelerômetro piezoresistivo triaxial de quatro *strain gauges*.

aplicada. O capacitor é formado por uma placa estacionária e outra ligada a massa sísmica. Como a capacitância é função da distância entre as placas, a mesma é alterada quando a massa entra em movimento. Dessa forma, é possível projetar acelerômetros com intervalos de medida de 0,2 g a 1.000 g capaz de medir uma frequência de até 3.000 Hz.

5.1.4 Acelerômetro de efeito Hall

Neste caso, quando o dispositivo é submetido a uma aceleração, esta movimenta uma fita que está conduzindo corrente elétrica por um campo magnético não uniforme. Dessa forma, quanto maior for o deslocamento, maior será o campo magnético e portanto, maior a diferença de potencial transversal a corrente devido ao efeito Hall (POPA *et al.*, 2019).

5.1.5 Magnetoresistivo

Neste tipo de transdutor, a aceleração causa um deslocamento de uma massa de material magnético e na parte fixa do dispositivo são posicionados materiais que têm sua resistência alterada quando submetidos a um campo elétrico (PHAN *et al.*, 2008).

5.1.6 Redes de Bragg em fibras ópticas

Redes de Bragg em fibras ópticas consistem de fibras com regiões específicas de variação periódica de índice de refração, conferindo a ela a propriedade de transmissão de diversos comprimentos de onda e de refletir um determinado comprimento, funcionando como um filtro de comprimento de onda. Ao sofrer uma deformação, a densidade da fibra óptica é alterada consequentemente acarretando mudanças em seu índice refração, mudando assim o comprimento de onda filtrado.

Desse modo, de forma construtiva similar ao piezoresistivo, porém fixando uma fibra a uma base fixa e a parte da fibra onde está a rede de Bragg fixada à massa sísmica, fazendo com que caso seja aplicada uma aceleração, a mesma sofra uma tensão, esticando-a. Este tipo de dispositivo é frequentemente utilizado para detecção de atividade sísmica devido à

sua altíssima sensibilidade e baixo ruído. É capaz de medir frequências na ordem de 0,1 Hz a 1 Hz com aceleração de até 0,1 g dando como resposta nesta faixa de aceleração variações no comprimento de onda na faixa de 90 a 600 pm/g (BASUMALLICK *et al.*, 2013).

5.2 Acelerômetros MEMS

Os acelerômetros MEMS foram desenvolvidos através de estudos de microeletrônica, cujo objetivo era produzir dispositivos de dimensões reduzidas e vasta extensão de aplicações. O primeiro acelerômetro MEMS foi proposto em 1960 mas apenas comercializados em 1982 quando as indústrias de *airbag* automotivo introduziram este dispositivo para detecção de impacto (WIKI, 2017). A Analog Devices Corporation começou em 1991 uma produção em larga escala de acelerômetros MEMS para sistemas de *airbag*. A partir de então a indústria deste tipo de dispositivo vem crescendo e alguns fabricantes incorporaram no mesmo chip um giroscópio (TDK, 2018). Atualmente, as principais aplicações para os acelerômetros MEMS são: ignição de *airbag*, medição de vibração em máquinas rotativas, verificação da orientação em smartphones, medição de movimento em joystick de videogames, monitoramento da aceleração no lançamento de alguns foguetes, entre outros.

Um acelerômetro MEMS pode ser do tipo: piezoresistivo e de capacitância variável. Os de capacitância variável são dispositivos de grande sensibilidade e um pequeno intervalo de medida e são comumente utilizados para monitoramento estrutural e em sistemas estáticos (com aceleração constante). Os piezoresistivos possuem maior intervalo de medida porém com menor sensibilidade e são aplicados comumente em medição de choques mecânicos e explosões (PIESOTRONICS, 2018).

A aceleração pode ser mensurada através da utilização de uma massa de prova, ancorada por feixes de suspensão. Uma aceleração externa desloca a massa de prova de sua posição inicial onde este deslocamento é proporcional à magnitude da aceleração (SINHA *et al.*, 2014). Os acelerômetros MEMS capacitivos medem a aceleração baseado na mudança de capacitância durante o movimento da massa de prova. A Figura 19a apresenta um esquemático de um acelerômetro capacitivo MEMS. Quando a aceleração é zero, a lacuna entre os eletrodos é igual, portanto $C_1 = C_2$. Quando o acelerômetro é submetido a uma aceleração, a massa de prova se movimenta e a capacitância C_1 aumenta e a C_2 diminui (Figura 19b). A sensibilidade do acelerômetro é proporcional ao deslocamento da massa de prova e inversamente proporcional ao espaço entre os eletrodos, dado por:

$$\frac{\Delta C}{C} \approx \frac{x_{max}}{d}, \quad (5.1)$$

onde $\Delta C = |C_1 - C_2|$, C é a capacidade inicial em aceleração igual a zero, x_{max} é o deslocamento máximo e d é a distância entre a massa de prova e os eletrodos em estado inicial (SINHA *et al.*, 2014). A sensibilidade de cada acelerômetro dependerá de uma série de fatores e associados ao projeto do fabricante.

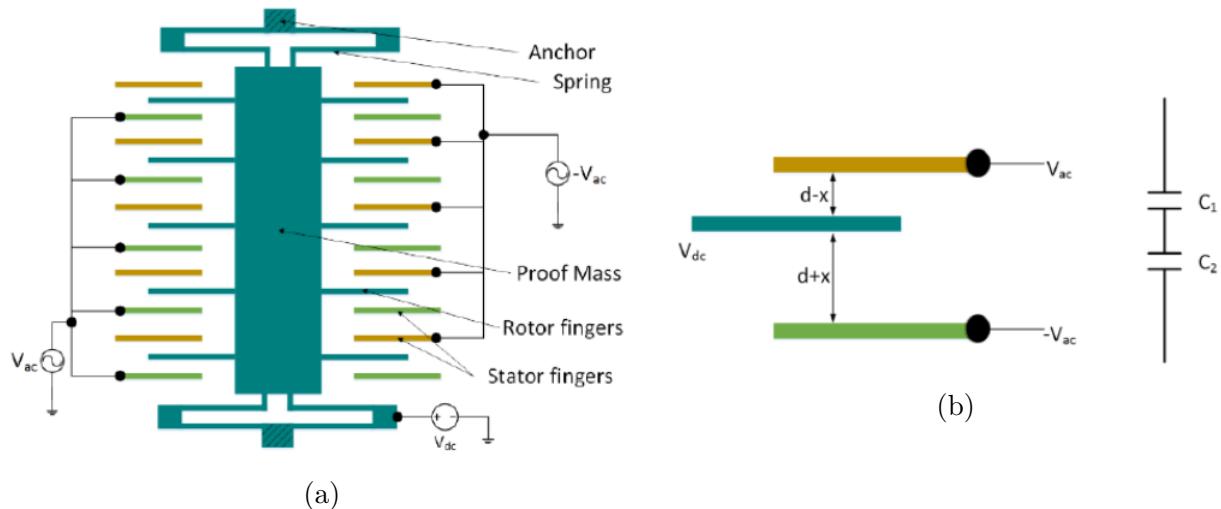


Figura 19 – Acelerômetro MEMS capacitivo diferencial: (a) Esquemático (b) Formação da capacidade diferencial sob aceleração. Fonte: Extraído de Sinha *et al.* (2014)

Os dispositivos microeletrônicos vêm ao longo dos anos ficando cada vez mais baratos e com menor tamanho, assim como previu Moore (1965), cujo conceito chamado de Lei de Moore, sofreu uma atualização em 1975. A lei de Moore previu que o número de transistores dentro de processador dobraria a cada 18 meses. Essa lei vem se concretizando até os dias de hoje. A Figura 20 apresenta a evolução da capacidade de processamento ao longo dos anos e uma projeção futura.

Os acelerômetros MEMS também acompanham a Lei de Moore, conforme apresentado na Figura 21 onde é possível observar que estão ficando cada vez menores e com menor custo (DEVELOPPEMENT, 2014).

5.2.0.1 MMA8451Q

O acelerômetro MEMS escolhido para o desenvolvimento deste trabalho é o MMA8451Q da NXP que é um acelerômetro capacitivo, 3-axial, 14/8 bits. A Figura 22 apresenta o dispositivo escolhido (SEMICONDUCTORS, 2017). As características mais relevantes deste dispositivo estão listadas a seguir:

- 1,95 V a 3,6 V de tensão de alimentação;
- ± 2 g, ± 4 g, ± 8 g escala selecionável dinamicamente;

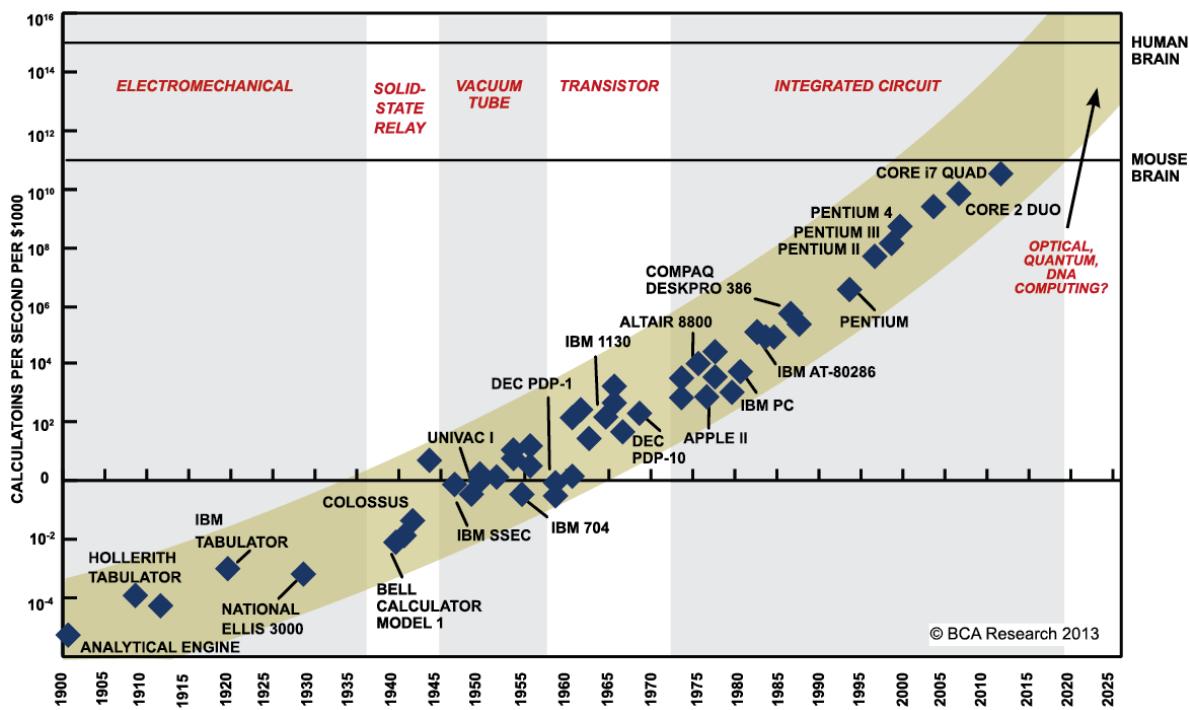


Figura 20 – Linha de tempo da quantidade de processamento ao longo dos anos baseado na Lei do Moore. Fonte: Adaptado de Kurzweil (2006)

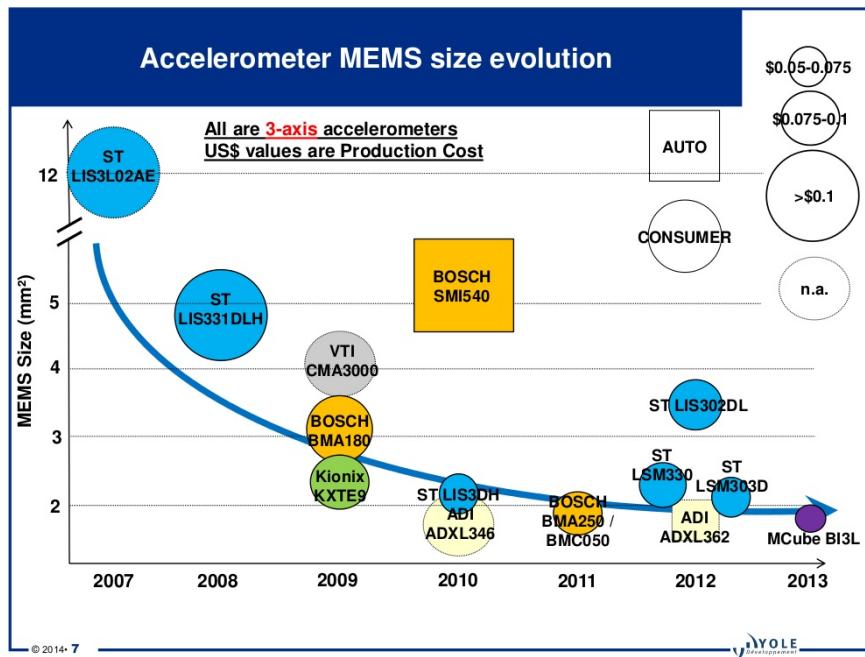


Figura 21 – Evolução dos acelerômetros MEMS em tamanho e custo. Fonte: Extraído de Developpement (2014)

- Output data rates (ODR) de 1,56 Hz até 800 Hz;

- Interface de saída digital I²C.



Figura 22 – Chip MMA8451Q da NXP. Fonte: Extraído de Semiconductors (2017)

5.3 Hardware de processamento

A aceleração medida pelo acelerômetro MEMS é recebida por um microcontrolador que processa essa informação. Diversos microcontroladores foram avaliados para realizar esta tarefa. O custo é uma variável importante para a avaliação, já que uma das especificações do sistema é ser de baixo custo.

O Microcontrolador escolhido para esta finalidade foi o ESP32 da Espressif. Este dispositivo encontra-se em expansão no mercado atualmente e vem sendo utilizado para diversas aplicações devido à sua capacidade de processamento e custo atrativo. O módulo escolhido foi o ESP32-WROOM-32, apresentado na Figura 23, que tem como características:

- Dois núcleos independentes de clock ajustável de 80 MHz até 240 MHz;
- Conexão Bluetooth com suporte a BLE (Bluetooth Low Energy);
- Conexão Wi-Fi 802.11 n (2,4 GHz), com velocidade de até 150 Mbps;
- Periféricos: capacitive touch sensors, Hall sensor, low-noise sense amplifiers, SD card interface, Ethernet, high-speed SPI, UART, I²S e I²C;
- Chip ESP32-D0WDQ6;
- Flash 4 MB, 38 Pins;
- Dimensão: 18x25,5x2,8 mm;
- Antenas PCB certificadas.



Figura 23 – Módulo ESP32-WROOM-32 da Espressif

5.4 Indústria 4.0

A indústria está passando por mais uma revolução atualmente denominada quarta revolução industrial. Fazendo um breve histórico, tivemos a primeira revolução industrial com o advento da máquina a vapor em 1784, quando a mecanização possibilitou um salto na produtividade. A segunda revolução tem como marco o Fordismo com a implantação da produção seriada e a eletrificação. A terceira é conhecida como era da automação graças à revolução eletrônica que transformou a indústria. Já a quarta revolução industrial tem como característica a digitalização do mundo atual, ou seja, o sistema não é apenas automático mas também autônomo (TU, 2017), (ABID, 2018). A Figura 24 apresenta um esquema sobre o histórico da revolução industrial.

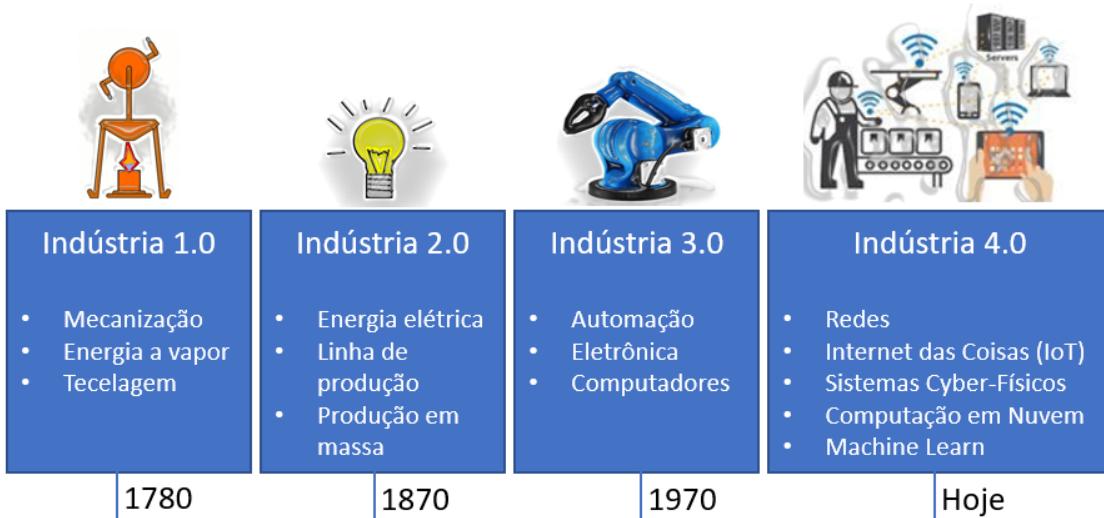


Figura 24 – Linha de tempo de revolução industrial

Na indústria 4.0, a Internet das Coisas contribui para a integração dos ambientes físico e digital, conhecido como ambiente Cyber-Físico. Um ambiente Cyber-Físico pode ser entendido como uma rede de equipamentos conectados de acordo com suas inter-relações. De modo global, existe conexão entre a TI, componentes mecânicos e eletrônicos das máquinas que também se comunicam por meio de seus protocolos (SANTOS *et al.*, 2016). Dentro da

Indústria, esta comunicação vem sendo denominada Internet Industrial das Coisas (IIoT) (MADEIRA, 2016). O número de dispositivos conectados vem crescendo ao longo dos anos e uma perspectiva é que em 2050 este número ultrapasse 100 bilhões de dispositivos conectados (PHAN, 2016), conforme apresentado na Figura 25.

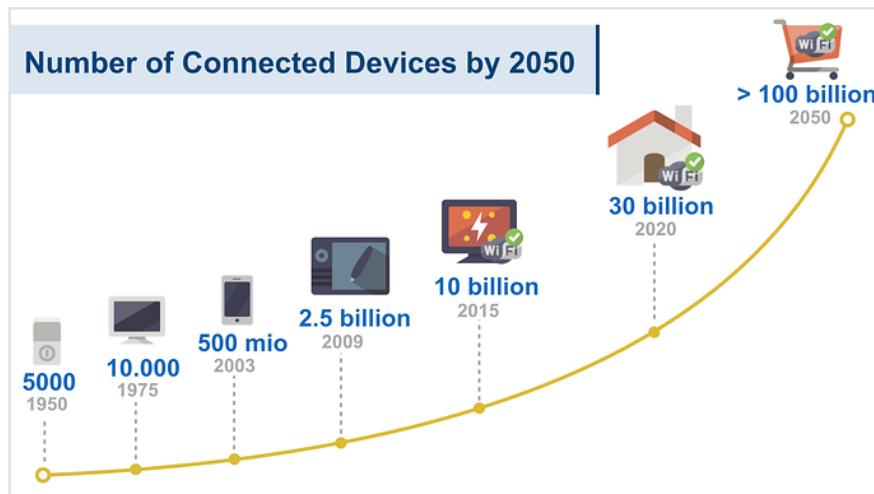


Figura 25 – Perspectiva de Dispositivos Conectados até 2050. Fonte: Baseado em Phan (2016)

O conceito da indústria 4.0 engloba as principais inovações tecnológicas de vários segmentos, aplicando-as nos processos de fabricação e de serviços. Essas tecnologias têm permitido uma mudança no modelo de negócios, permitindo melhorias significativas nos modelos atuais. Essas tecnologias são o que diferenciam a quarta revolução industrial da terceira, que são:

- *Tomada de Decisão em Tempo Real*: a gestão ocorre em tempo real, quando os dados de cada etapa do processo são obtidos de forma precisa, ampliando as possibilidades de análises e tomada de decisão. A inteligência artificial é importante neste processo tanto para análise dos dados quanto para reconhecimento de padrões, auxiliando na tomada de decisão automática.
- *Virtualização*: é a representação digital do processo de produção, simulando o processo, suas velocidades, consumo, entre outros, de modo a identificar e prevenir problemas e encontrando uma solução ótima para operação do processo.
- *Descentralização*: os processos ficam menos dependentes de interferências humanas, tomando decisões mais seguras geradas por sistemas cyber-físicos e baseados nas análises dos dados de processo.

- *Orientação a serviços*: este conceito está associado a softwares que permitem que o cliente interaja com o processo requisitando novas aplicações de acordo com sua necessidade.
- *Interoperabilidade*: a comunicação entre os sistemas cyber-físicos, sensores, produtos, humanos, atuadores, fábricas inteligentes (M2M - *Machine to Machine*) é feita através de diversos protocolos de comunicação, nos quais a IoT é a facilitadora de tal interconectividade.
- *Modularidade*: é a divisão do processo em subunidades que podem ser conectadas ou desconectadas do processo de acordo com a necessidade, permitindo assim que cada parte do processo seja um sistema independente porém com interação com os demais.

As grandes empresas acreditam que a indústria 4.0 é a grande oportunidade de tornar a indústria mais produtiva, sendo auxiliada por tecnologias digitais que ajudarão as empresas a serem mais ágeis na tomada de decisões.

5.5 Internet das coisas - IoT

O termo Internet das Coisas – IoT (*Internet of Things*) foi criado em 1999 por Kevin Ashton, pesquisador do Instituto de Tecnologia de Massachusetts (MIT) que concebeu um sistema de sensores. A aposta da tecnologia está na possibilidade de se etiquetar eletronicamente os produtos da linha de produção de uma empresa, facilitando a logística por meio de identificadores de radiofrequência (FOOTE, 2016). Em 2013, a IoT evoluiu para um sistema que utiliza várias tecnologias, desde comunicação sem fio e sistemas microeletromecânicos (MEMS) a sistemas embarcados, aplicados em áreas tradicionais de automação, redes de sensores, GPS, sistemas de controle e entre outros que suportam IoT.

Esta ideia vem sendo utilizada na interligação direta entre dispositivos para que possam se comunicar entre si (M2M - *Machine to Machine*). Um exemplo prático é utilizar um *smartphone* para enviar sinais de acionamento de um portão residencial assim que o usuário estiver próximo. Segundo Santos *et al.* (2016), a IoT é uma extensão da internet atual, proporcionando aos objetos que possuam capacidade computacional e de comunicação a se conectar à internet, permitindo aos usuários controlar os objetos remotamente ou torná-los provedores de serviços.

O termo *objetos* refere-se a elementos que possuem capacidade de comunicação e/ou processamento aliados a sensores. Neste quesito, podemos relacionar computadores conven-

cionais, TVs, automóveis, *smartphones*, sensores ou qualquer outro equipamento que possua uma forma de conexão à rede.

Os requisitos básicos de um sistema IoT são:

- *Sensores*: responsáveis por coletar as informações necessárias para o qual foram desenvolvidos (temperatura, pressão, vibração, etc).
- *Processamento*: responsável por processar as informações recebidas, podendo ser feitas por microcontroladores, processadores e FPGAs.
- *Comunicação*: protocolo de comunicação através do qual o objeto irá se comunicar (bluetooth, WiFi, etc).
- *Serviços*: o modelo de negócios no qual o dispositivo será inserido, como transporte, comércio, agronegócio, etc.

5.5.1 Tecnologia de comunicação sem fio para IoT

Como já mencionado na seção 2.2, existem várias tecnologias de comunicação para IoT. Para este trabalho, optou-se pela comunicação WiFi pois está presente no *hardware* de processamento escolhido e detalhado na seção 5.3. O protocolo de comunicação WiFi disponível no *hardware* utilizado atende ao padrão IEEE 802.11n que estabelece uma frequência de 2,4 GHz e/ou 5 GHz com capacidade de 150 a 600 Mbps de velocidade de transmissão com potência de 100 mW (MORAES, 2010).

O que também torna a rede WiFi atrativa é seu alcance que fica em torno de 30 a 100 m. Além disso, o WiFi permite uma implantação de uma rede local (LAN), cada dia mais barata, e pode ser aplicado ao ar livre. No quesito segurança, o WiFi dispõe de diversos mecanismos de proteção da rede, evitando a utilização da mesma por dispositivos não autorizados (MORAES, 2010). Os principais mecanismos de segurança de uma rede WiFi são: WEP (*Wired Equivalent Privacy*), WPA (*Wired Protected Access*) e WPA2 que é uma variação do WPA baseado no protocolo AES (*Advanced Encryption Standard*).

No entanto, as redes WiFi podem sofrer interferências que, dependendo da intensidade, podem fazer com que a rede deixe de funcionar. Tais interferências podem ser barreiras físicas ou eletromagnéticas, estas últimas que podem ser causadas por microondas, motores elétricos, entre outros. Como o sistema será aplicado em ambientes industriais e, mais especificamente, em motores elétricos, estudar a viabilidade da rede WiFi é importante (MAHANTI *et al.*, 2010).

Qualquer ambiente industrial contém ruído elétrico, porém, a utilização de métodos de aterramento adequado, blindagem, fios trançados, filtros e amplificadores diferenciais podem controlar o ruído na maioria dos casos. Por exemplo, inversores de frequência podem gerar interferência eletromagnética na faixa de 10 MHz a 300 MHz devido aos seus circuitos de comutação. A maioria dos fabricantes tomam cuidado no projeto de tais dispositivos de modo a minimizar este efeito (CASSIOLATO, 2018). Outro aspecto importante para a instalação adequada é a disposição adequada dos roteadores a fim de minimizar os efeitos de possíveis interferências (PLETS *et al.*, 2016), (CISCO, 2018).

5.5.1.1 Protocolo MQTT

O MQTT (*Message Queue Telemetry Transport*), é um protocolo desenvolvido pela IBM e atualmente é um protocolo de comunicação aberto OASIS. A arquitetura desse protocolo é do tipo *publish-subscribers*, no qual o dispositivo é responsável pelo envio dos dados (*publish*) ao servidor (*broker*), que intermedia essa operação. O servidor então pode enviar as informações recebidas para os clientes interessados (*subscribers*) (CONCEIÇÃO; COSTA, 2019; YUAN, 2017). A Figura 26 apresenta o esquema de cada funcionalidade na comunicação.

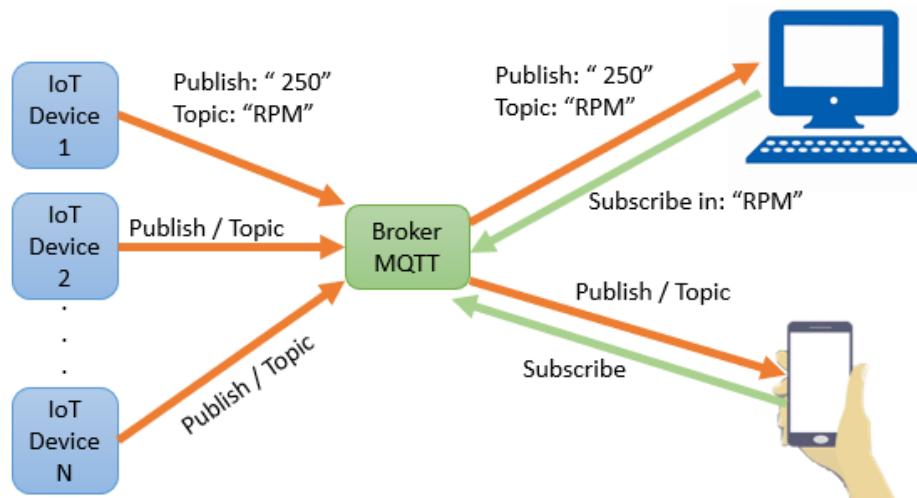


Figura 26 – Funcionamento do Publish, Broker e Subscribers no MQTT

Situado na camada de arquitetura do TCP/IP, o MQTT define a forma de interação entre os dispositivos, especificando o papel de cada um, o formato e a ordem da mensagem. Esse protocolo tornou-se padrão para comunicações IoT (CONCEIÇÃO; COSTA, 2019; YUAN, 2017).

5.5.2 Armazenamento em Nuvem

O armazenamento em nuvem é um modelo no qual os dados são armazenados na internet por meio de um provedor de computação em nuvem, responsável por gerenciar e operar o armazenamento físico de dados, possibilitando acesso aos dados a qualquer momento e de qualquer lugar. O armazenamento em nuvem é adquirido de um fornecedor externo de nuvem ou pela criação de seu próprio ambiente. Os aplicativos acessam o armazenamento em nuvem através de protocolos de armazenamento ou usando uma API (*Application Programming Interface*). Muitos fornecedores desse serviço também disponibilizam serviços complementares para auxílio na coleta, proteção e análise dos dados (SOUSA *et al.*, 2009).

O armazenamento de dados na nuvem possibilita diversos benefícios, como diminuição do custo total de propriedade, pois não é necessária aquisição de *hardware*, diminuição no tempo de implantação, pois a infra-estrutura estará completa e maior gerenciamento de informações devido à centralização dos dados. Além disso, alguns quesitos são importantes para a escolha ou implantação de um sistema de armazenamento em nuvem, como durabilidade dos dados, disponibilidade e segurança (SOUSA *et al.*, 2009). A Figura 27 apresenta uma visão geral de um sistema de computação em nuvem. Como apresentado na figura, os usuários alimentam a nuvem com dados e aplicações e podem acessar a nuvem utilizando qualquer dispositivo que tenha acesso à internet. Os dados estão centralizados em uma infraestrutura com diversos recursos computacionais e podem ser acessados em tempo real e de qualquer lugar.

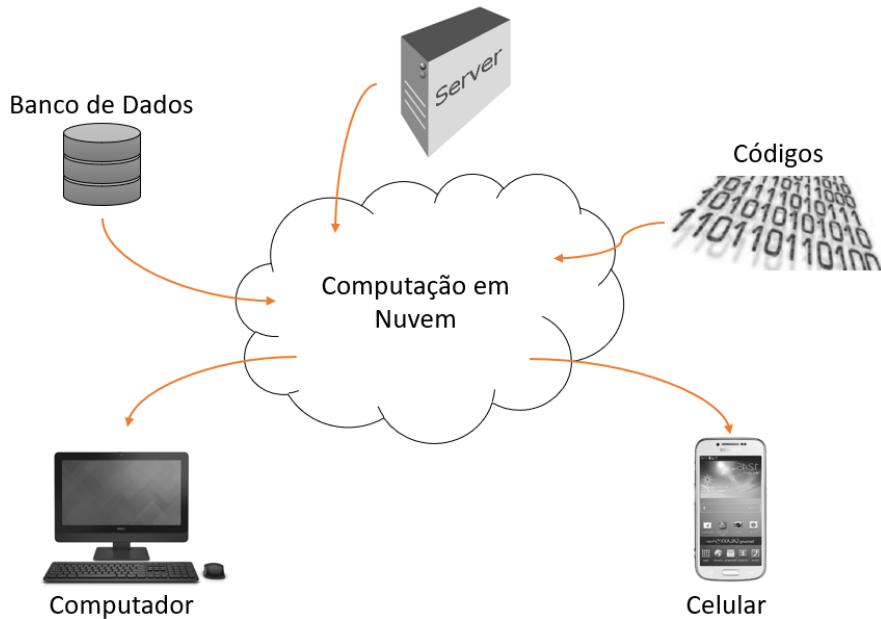


Figura 27 – Visão geral de um sistema de computação em nuvem

Existem diversos modelos de implantação de ambientes de computação em nuvem. A

restrição ou acesso depende do modelo de negócio que se deseja trabalhar. Na maioria das empresas não é desejável que todos os usuários possam ter acesso a determinadas informações ou recursos, assim, surge a necessidade de ambientes mais restritos onde somente usuários com devida autorização pode acessar dados e utilizar serviços. Os modelos de implantação de computação em nuvem são divididos em (SOUZA *et al.*, 2009).:

- Nuvem Privada: A infraestrutura de nuvem é utilizada exclusivamente por uma determinada organização, podendo ser remota ou local, sendo administrada pela empresa ou por terceiros. Políticas de acesso são empregadas nesse modelo;
- Nuvem Pública: A infraestrutura de nuvens é disponibilizada para o público em geral, podendo ser acessado por qualquer usuário, não permitindo aplicação de gerenciamento e controle de acesso;
- Nuvem Comunitária: Neste modelo, diversas organizações compartilham uma nuvem, sendo esta administrada por entidades que atendam o interesse comum.
- Nuvem Híbrida: É uma composição de duas ou mais nuvens, podendo ser privadas, públicas ou comunitária que permite a portabilidade de dados e aplicações.

A computação em nuvem envolve diversos conceitos e tecnologias. Empresas como Amazon, Microsoft e Google estão oferecendo serviços computacionais seguindo as diretrizes de computação em nuvem. A comunidade acadêmica também tem apresentado interesse neste ambiente, podendo destacar o projeto Eucalyptus (LIU *et al.*, 2007) e o projeto MiniCloud desenvolvido na Unicamp por meio de uma parceria com a IBM e do laboratório Linux Technology Center. Dentro desse laboratório está o OpenPOWER Lab cujo foco é em pesquisa e desenvolvimento da arquitetura *open source Power* (BICAS, 2017).

Dentro do universo de ambiente de computação proprietária, a lista a seguir resume as principais existentes:

- Amazon Web Services IoT Platform
- Microsoft Azure IoT Hub
- IBM Watson IoT Platform
- Google Cloud Platform
- Oracle
- Salesforce

- Temboo (*free*)
- Carriots (*free*)
- NearBus (*free*)
- Ubidots (*free*)
- ThingSpeak (*free*)

Dentre as plataformas listadas, a Microsoft Azure foi testada durante 1 mês gratuitamente e apresenta muitas ferramentas e funcionalidades interessantes, porém este projeto irá direcionar-se a plataformas gratuitas. Entre as plataformas gratuitas, foram analisadas e testadas a ThingSpeak e Ubidots. O primeiro possui integração com o Matlab, o que a torna interessante para exportação de dados para uma futura análise utilizando aprendizado de máquinas. Já o último possui recursos de utilização simples e intuitivos, o que inclui a inserção de índices para comparação, podendo ativar alertas em relação a um determinado comportamento do sinal.

Até então o Ubidots havia sido selecionado para a execução de todos os testes finais, entretanto, pesquisando a plataforma IBM Watson IoT Platform, descobriu-se que oferece um plano Lite, gratuito e sem restrição no tempo de uso, onde vários serviços são disponibilizados para que o usuário possa experimentar suas vantagens, com algumas restrições. É possível registrar até 500 dispositivos com um armazenamento máximo de 200 Mb por dispositivo na plataforma IoT (IBM, 2018). Além disso, possui muito mais material para aprender a utilizar a ferramenta e um canal de suporte muito competente. Assim sendo, para os testes finais a plataforma IBM Watson IoT Platform foi escolhida e utilizada de forma satisfatória para a aplicação desejada.

5.6 Solução Desenvolvida

Um esquemático com a solução desenvolvida é apresentado na Figura 28. Nela é possível observar as escolhas dos *hardwares*: o acelerômetro, adotou-se o MMA8451Q, para o microcontrolador, adotou-se o ESP32. A comunicação sem fio escolhida foi o Wi-Fi e a plataforma de computação em nuvem adotada foi a IBM Watson IoT Platform.

5.7 Conclusões deste capítulo

Este capítulo apresentou as tecnologias envolvidas no trabalho, como o acelerômetro e o microcontrolador escolhidos. Também foi apresentado o panorama atual da indústria 4.0, a

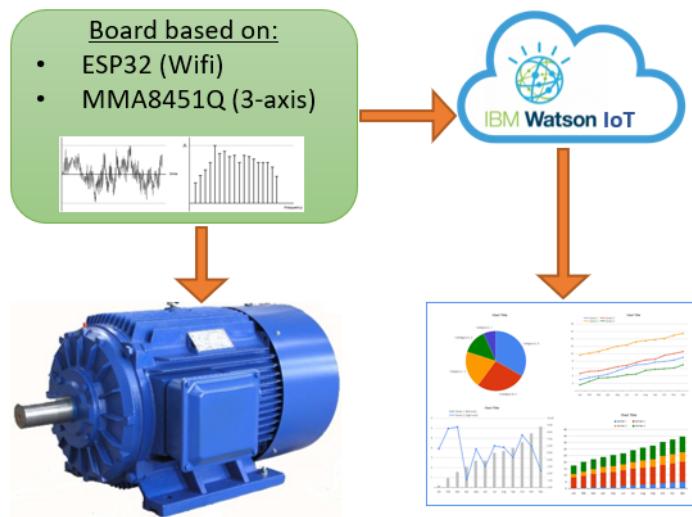


Figura 28 – Esquema da solução desenvolvida

IoT e a tecnologia de comunicação escolhida (Wi-Fi), bem como suas vantagens e limitações. Também foi abordada a computação em nuvem e quais plataformas foram testadas.

6 Desenvolvimento do Dispositivo

Neste capítulo é apresentado o projeto do dispositivo desenvolvido a partir de especificações definidas. A forma como o projeto foi concebido e implementado será detalhada. Detalhes como tamanho das trilhas, dispositivos de proteção, condicionamento de sinais e os sinais de comunicação foram projetados conforme recomendações de boas práticas de projetos de PCB (*Printed Circuit Board*) (CIRCUIBRAS, 2009).

O processo para utilização da IDE do Arduino como ferramenta de escrita e compilação do código é apresentado e a forma de conexão do dispositivo ao adaptador UART/USB é detalhada. O processo de inicialização do dispositivo para o modo de upload de software também é apresentado neste capítulo. A bancada de testes montada em laboratório é detalhada com os itens que compõe as simulações de perturbações aplicadas no motor BLDC (*Brushless Direct Current*).

6.1 Especificações do Dispositivo

As especificações necessárias para que o dispositivo possa atender os objetivos propostos estão resumidas na Tabela 3.

Tabela 3 – Especificações do Dispositivo Desenvolvido

Característica	Definição
Tamanho	menor que 60~mm de diâmetro
Formato	Circular
Número de Layers	2 (Top e Bottom)
Acelerômetro	MEMS, Triaxial, com taxa de amostragem maior ou igual a 800~Hz
Microcontrolador	Capaz de processar a FFT do sinal pelo menos a cada 1~s
Comunicação	Sem fio
Alimentação	Fonte CC de 9 a 12V
Operação do Equipamento Medido	Velocidade de Até 4.000 RPM
Armazenamento de Dados	Em nuvem

A velocidade máxima de rotação da máquina estar limitada a 4.000 RPM deve-se ao fato de ser um sinal amostrado e portanto necessitar de uma taxa de amostragem $2 \times$ superior ao número de amostras, conforme apresentado na seção 4.3.1.1. Dessa forma, como o acelerômetro escolhido apresenta uma taxa de amostragem de 800 Hz, a reconstrução do

espectro se dá com qualidade na frequência de até 400 Hz. Como apresentado na equação 4, é possível determinar que uma máquina girando a 4.000 RPM está girando em 66,67 Hz. Dessa forma, a harmônica de ordem seis dessa frequência é de 400 Hz, justificando assim sua limitação.

A partir das especificações foi desenvolvido um diagrama identificando os principais módulos de circuitos, conforme apresentado na Figura 29. Dessa forma, foi projetado um dispositivo em formato circular onde todo o circuito eletrônico foi distribuído na placa de forma a minimizar o tamanho das trilhas e balancear o centro de gravidade. O diagrama do circuito eletrônico de cada bloco identificado será detalhado nas próximas seções.

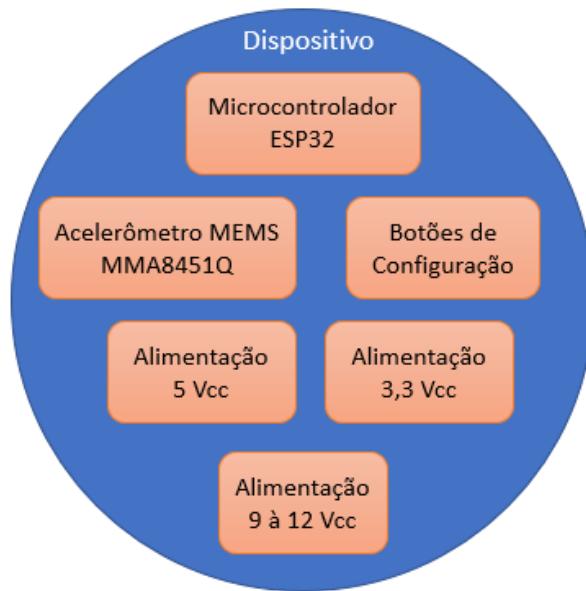


Figura 29 – Esquema de módulos necessários no dispositivo

6.2 Diagrama do Circuito Eletrônico

O projeto da placa foi desenvolvido baseado nos *datasheets* disponibilizados pelos fabricantes dos componentes MMA8451Q¹ e ESP32².

A ligação da fonte externa no dispositivo é apresentada na Figura 30. A tensão de entrada deve ser fornecida por uma fonte externa de 9 V à 12 V com uma capacidade de pelo menos 500 mA de corrente.

¹ Disponível em: <https://os.mbed.com/media/uploads/sam_grove/frdm-kl05z_user-s_manual_-rev_1.1-.pdf>

² Disponível em: <https://www.smart-prototyping.com/image/data/2_components/development_board/101839%20NodMCU-32S/1.png>

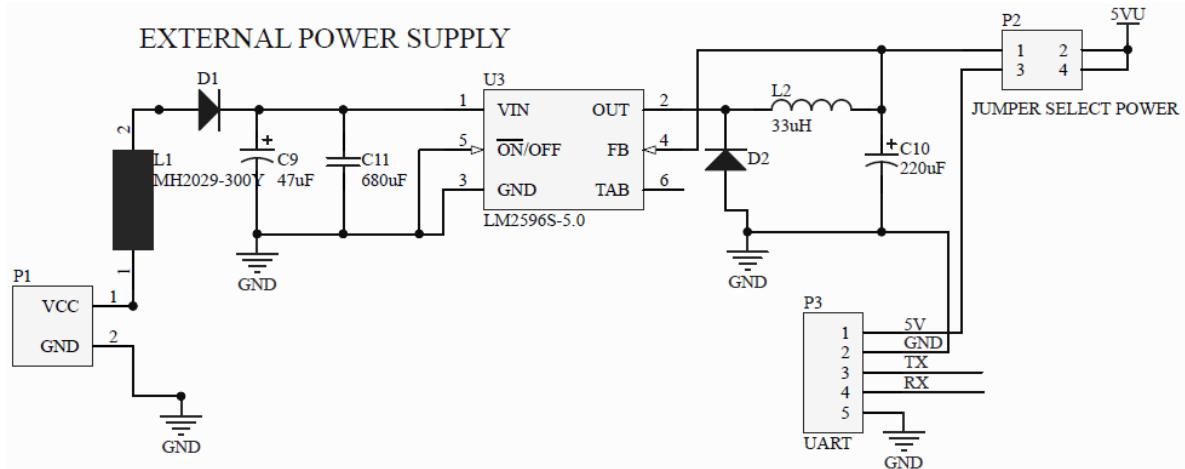


Figura 30 – Esquema de alimentação por uma fonte externa

O Pino P2 serve para selecionar qual será a fonte principal de alimentação do dispositivo, seja ela externa ou placa de comunicação UART.

A Figura 31 apresenta a ligação do circuito de conversão de 5 V em 3,3 V para alimentar os elementos do acelerômetro MMA8451Q e o microcontrolador Esp32. Estes dispositivos devem ser alimentados em 3,3V conforme especificações do fabricante.

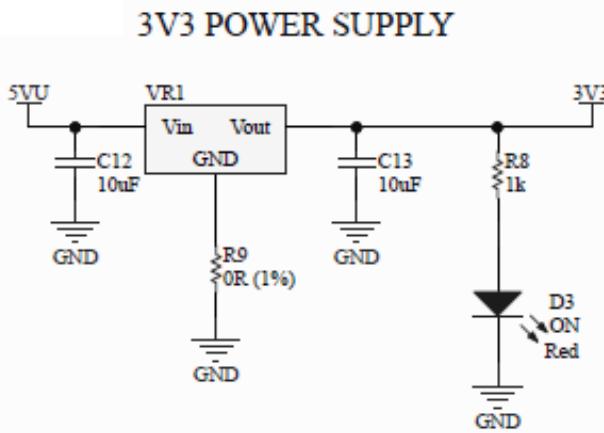


Figura 31 – Esquema de alimentação em 3,3V para o ESP32 e Acelerômetro

Ressalta-se a adição de capacitores de desacoplamento C12 e C13 no elemento VR1. Esses capacitores fornecem um caminho de baixa impedância para sinais de alta frequência. A proximidade desses capacitores com o circuito que se deseja proteger é fundamental pois quanto mais próximos, menores são os efeitos de indutâncias em série. Dessa forma, é importante adicionar capacitores de desacoplamento nos circuitos onde se deseja filtrar ruídos de alta frequência na alimentação do microcontrolador e do acelerômetro.

A Figura 32 mostra o esquema de ligação do ESP32, os capacitores de desacopla-

mento e os pinos de acesso às entradas disponibilizadas (ESPRESSIF, 2016). Na Figura 33 é mostrado o esquema de ligação do acelerômetro MMA8451Q conforme especificações e recomendações do fabricante (SEMICONDUCTORS, 2017). Na Figura 34 é mostrado o esquema dos botões de *boot* do ESP32.

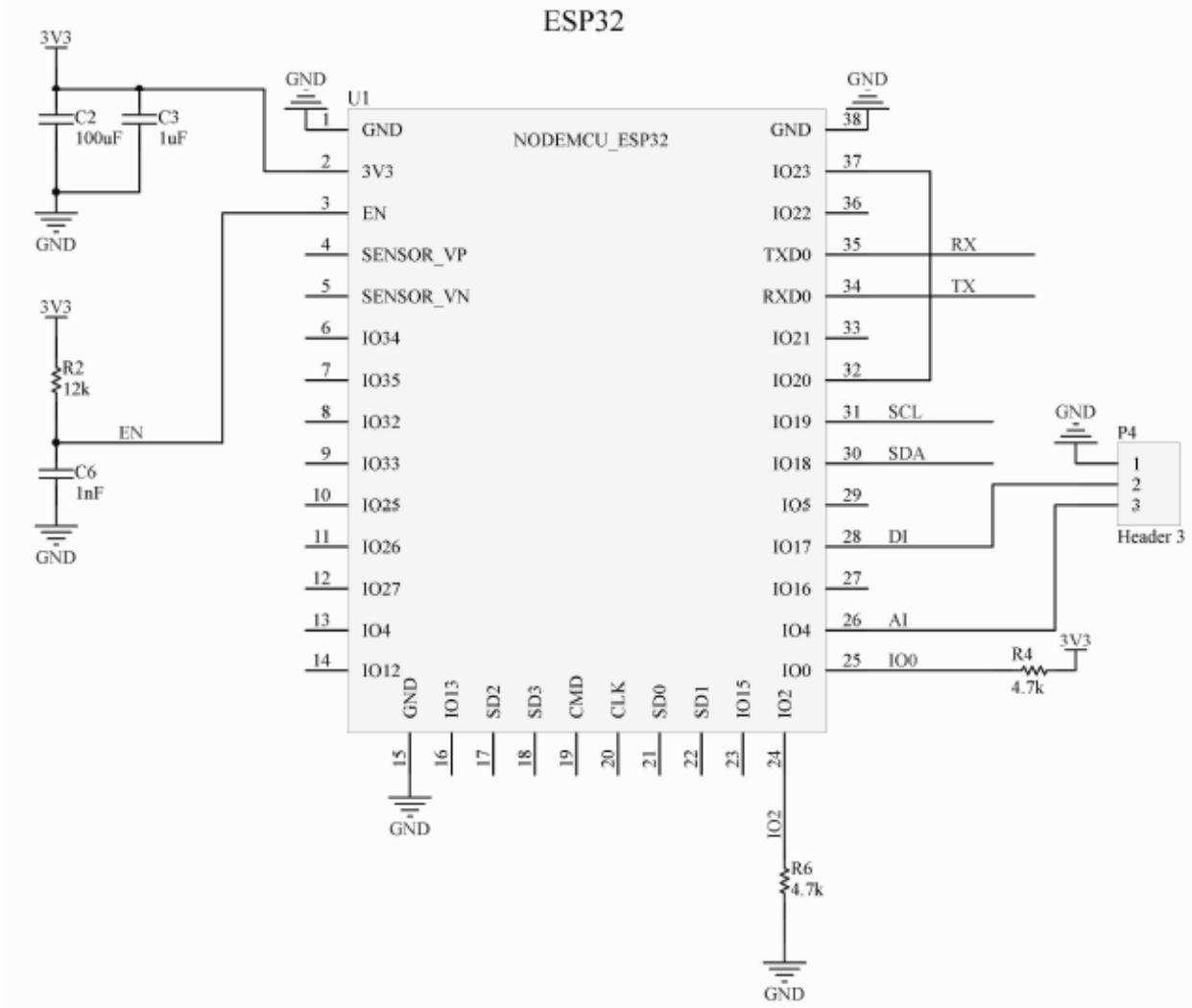


Figura 32 – Esquema de ligação do ESP32

Conforme orientações do fabricante, os botões são importantes para que se possa iniciar o dispositivo no modo *upload* ou no modo *run*. O procedimento necessário para cada modo está detalhado na seção 6.3.

A maioria dos elementos escolhidos são do tipo SMD (*Surface Mount Device*), onde somente as conexões dos pinos e do conector tipo *Jack* são do tipo THT (*Through-Hole Technology*). A escolha de elementos tipo SMD deve-se ao fato de serem mais compactos, o que permite uma placa de tamanho reduzido. O descritivo dos pinos e suas funções estão detalhados no Apêndice A.1.

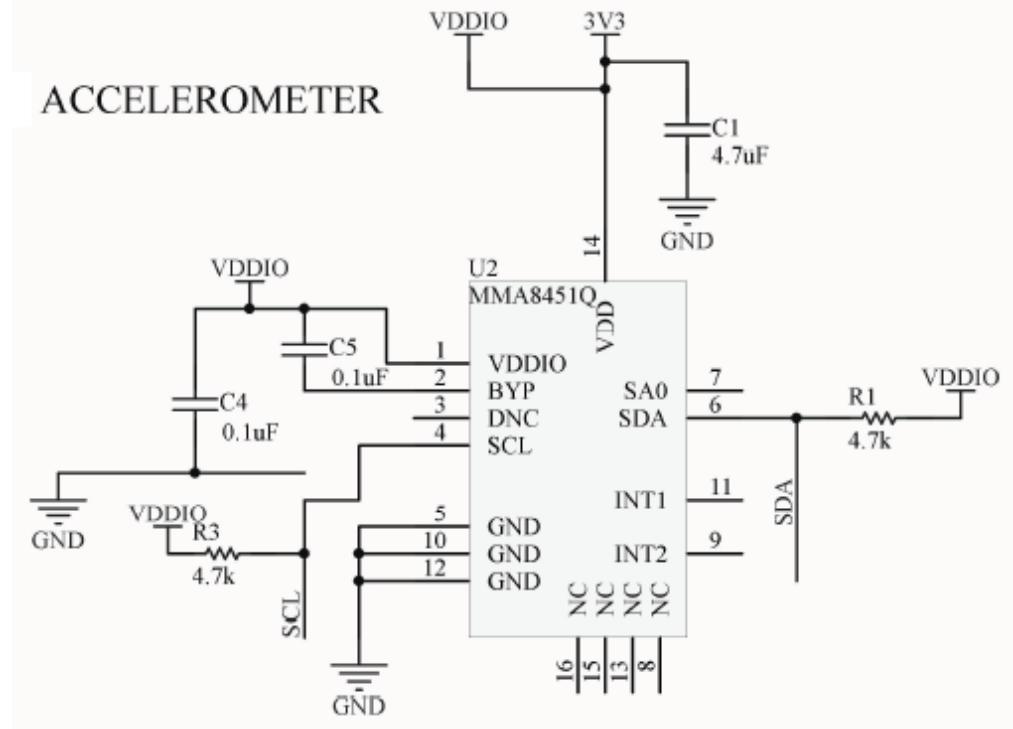


Figura 33 – Esquema de ligação do MMA8451Q

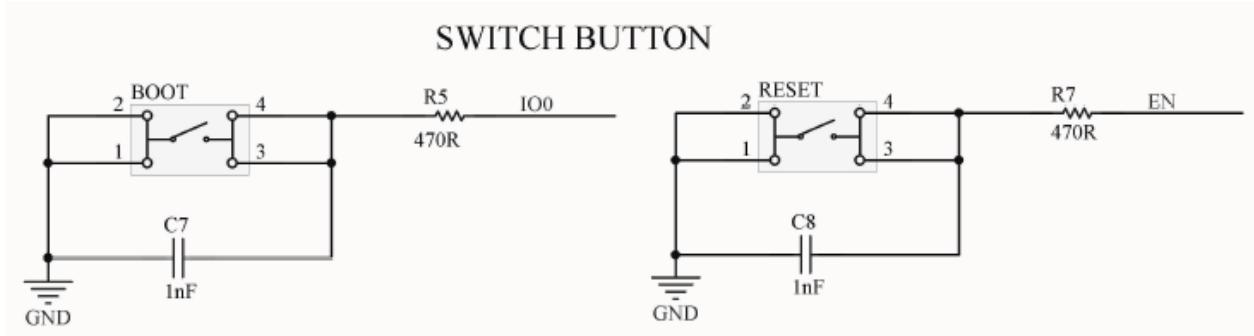


Figura 34 – Esquema ligação dos botões de gravação

6.3 Procedimento de Inicialização e Download de Firmware

A seleção do modo de inicialização do dispositivo é feita utilizando os botões RESET e BOOT. Para entrar no modo de gravação, é necessário ligar o dispositivo, pressionar o botão BOOT, manter o botão pressionado e em seguida pressionar o botão RESET, soltar o botão RESET e em seguida soltar o botão BOOT. O dispositivo ficará pronto para receber um novo firmware. Após enviar o novo firmware, pressione e solte o botão RESET para entrar no modo inicialização padrão. Este procedimento atende a especificação presente no *datasheet* (ESPRESSIF, 2017) e no esquemático do fabricante (ESPRESSIF, 2016). Todo o procedimento operacional de instalação do dispositivo está detalhado no Apêndice A.2.

6.4 Prototipagem

Para a conclusão do dispositivo final, foi necessária a confecção de dois protótipos. No primeiro, três inconsistências foram detectadas e corrigidas. O segundo e último protótipo foi desenvolvido e testado com sucesso em diversas situações. Na Figura 35a é apresentada a frente da placa e na Figura 35b a parte traseira. A placa foi projetada em formato circular, com 54 mm de diâmetro pesando aproximadamente 20 g (já montada). Uma lista de custos da placa é apresentada na Tabela 4, portanto, cada placa pode ser manufaturada com um investimento inferior a US\$ 30,00, conferindo ao dispositivo uma característica de baixo custo de material. Considerando uma produção deste dispositivo em grande escala, este custo cai consideravelmente. A Figura 36 apresenta uma foto do dispositivo desenvolvido, denominado *Vibration Monitor*.

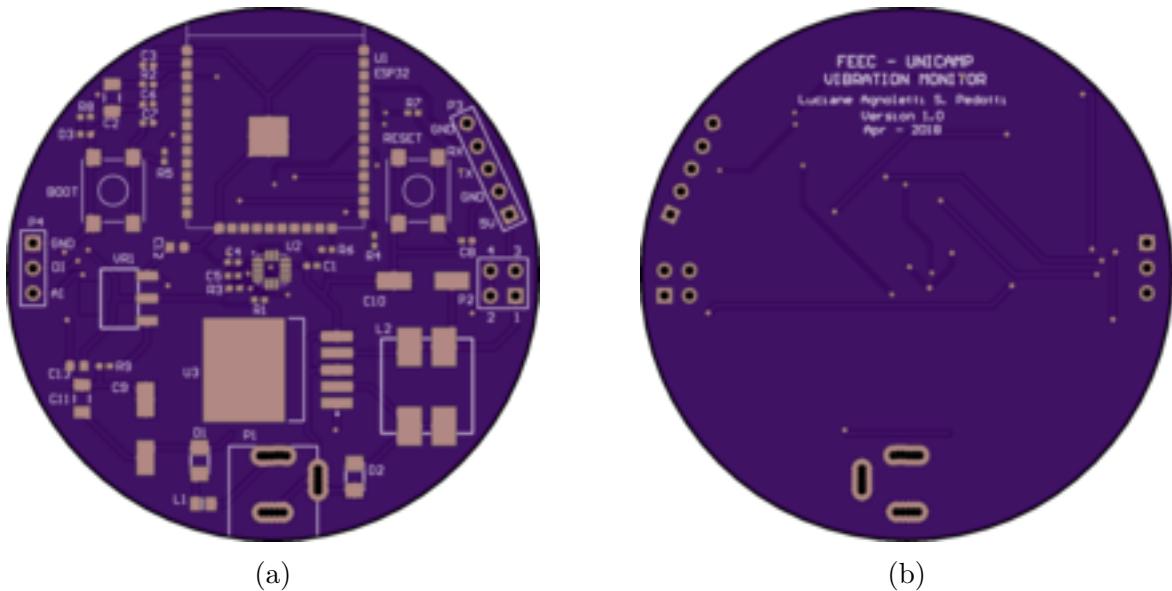


Figura 35 – Layout da PCB dispositivo desenvolvido

Tabela 4 – Custo do Dispositivo

Dispositivo	Custo por unidade
Espressif EPS32	US\$ 3,80
MMA8451Q	US\$ 1,46
Manufatura	US\$ 10,00
Outros Componentes	US\$ 12,74
Custo Final	US\$ 28,00

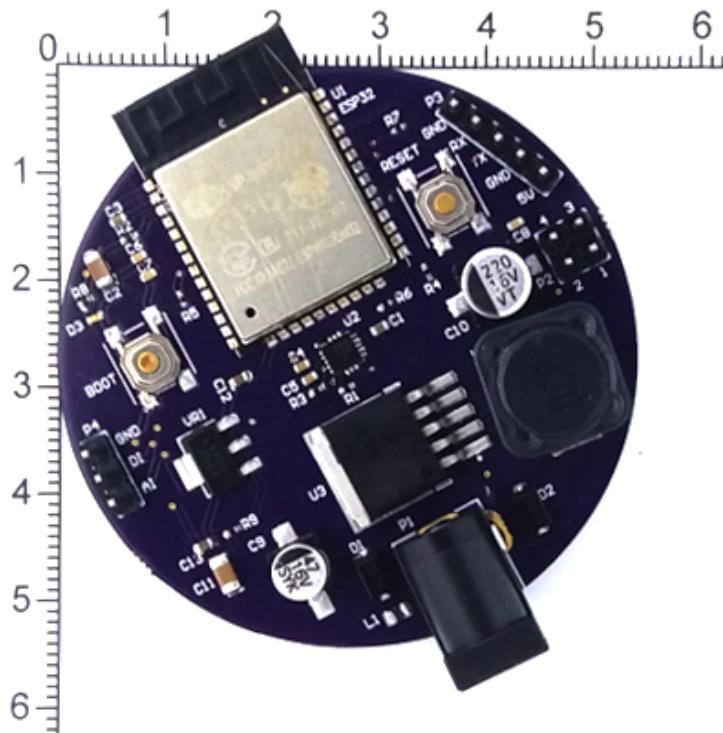


Figura 36 – Dispositivo desenvolvido (escala em centímetro)

6.5 Código Fonte

O fluxograma do código desenvolvido está apresentado nas Figuras 37 e 38.

A Figura 37 apresenta fluxograma do código para calcular a velocidade do dispositivo através de um pino de entrada digital cujo sinal de entrada é fornecido por um encoder. Este sinal pode ser fornecido de outras formas, como por leitura do valor de velocidade controlada por um inversor de frequência, por exemplo. Como o microcontrolador escolhido possui dois núcleos de processamento, foi escolhido o núcleo 0 (core 0) para ficar dedicado a este cálculo.

A Figura 38 apresenta fluxograma do código que roda no núcleo principal (core 1). Nele o dispositivo é configurado pelo *setup* e todas as variáveis são declaradas. O código está dividido em funções de forma estruturada. As funções deixam o código mais fácil para manutenções e melhorias futuras.

A habilitação da serial do microcontrolador foi feita com um *baud rate* de 250.000 bps. A comunicação do microcontrolador com o acelerômetro foi utilizando o protocolo I²C através dos pinos SDA(IO18) e SCL(IO19). O clock padrão do SCL é de 200 kHz, podendo ser configurado para até 400 kHz.

Na inicialização do acelerômetro, os registradores de configuração devem ser ajustados basicamente em 3 configurações: ativação do filtro passa-baixas, range de medida(2 g, 4 g ou

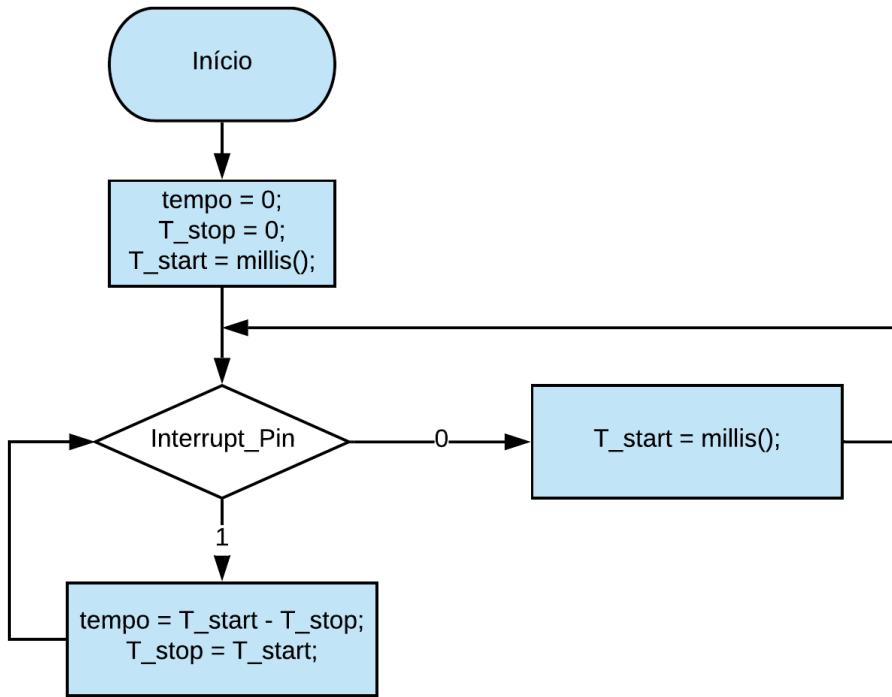


Figura 37 – Fluxograma do código desenvolvido - Core 0

8 g) e modo de leitura (rápida ou modo ativo). No caso do código desenvolvido, foi escolhida a desativação do filtro passa-baixas, range de ± 2 g e modo de leitura ativo (14 bits). A escolha do range de medição proporciona uma maior sensibilidade do sensor, neste caso de 4.096 contagens/s de acordo com as especificações do fabricante. A taxa de saída de dados do acelerômetro foi definida como seu valor máximo de 800 Hz.

Para habilitar o protocolo I²C, a biblioteca *wire.h* (disponível na IDE do Arduino) foi adicionada. O cálculo da FFT foi feito utilizando a biblioteca *myFFT.h*. O cálculo das seis harmônicas principais é feito baseado na velocidade da máquina. Através dessa informação, é possível saber qual é a frequência fundamental do sistema.

Em posse do vetor como resultado da FFT calculado anteriormente e conhecendo a frequência fundamental, o algoritmo faz uma varredura no vetor da FFT selecionando os valores máximos e frequência correspondente aos seis harmônicos fundamentais. No caso dos demais componentes harmônicos, basta multiplicar o valor de frequência da máquina pela componente harmônica desejada.

O processamento da FFT é feita diretamente no dispositivo para que a quantidade de informação transmitida seja a menor possível, de forma que o mesmo economize energia, viabilizando sua alimentação com baterias.

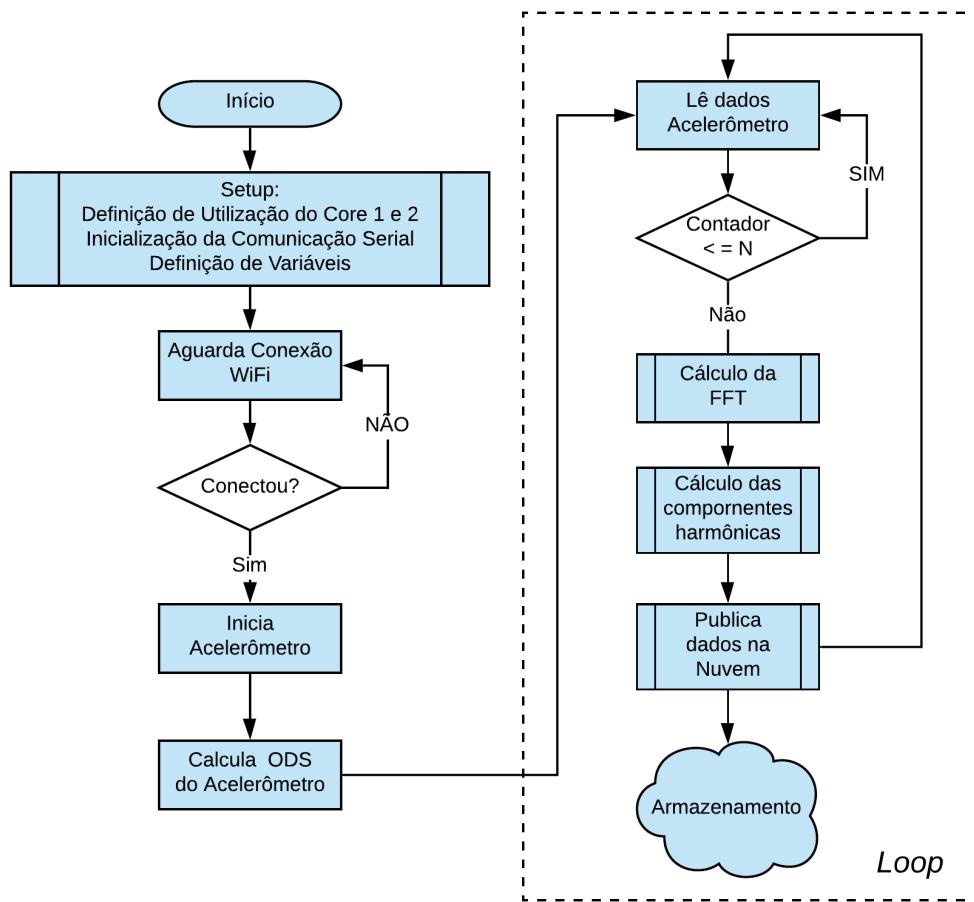


Figura 38 – Fluxograma do código desenvolvido - Core 1

Para habilitar a comunicação Wi-Fi, a biblioteca *WiFi.h* foi adicionada, e está disponível na IDE do Arduino. Para que a conexão com a rede Wi-Fi possa ser estabelecida, é necessário fornecer o **nome da rede** e a **senha** correspondente para o acesso. Para dispositivos IoT, a comunicação com a internet é algo fundamental.

Cada servidor em nuvem possui sua forma de receber os pacotes de dados que são transmitidos. Entretanto, em todos os casos, o endereço web do servidor deve ser informado, bem como o usuário previamente configurado no servidor e sua senha. Cada pacote transmitido (*payload*) possui uma estrutura que pode ser utilizada para outro dispositivo, sendo necessário apenas alterar o *DEVICE-ID*. Por exemplo, se em seu sistema tiver 4 sensores, o código será sempre o mesmo para comunicação em nuvem, basta mudar o código referente a cada dispositivo instalado.

Para realizar a transmissão dos pacotes é necessário adicionar a biblioteca *PubSubClient.h* (disponível na IDE do Arduino), uma biblioteca para o MQTT que envia os pacotes de dados para o servidor. O código fonte do *firmware* da placa *Vibration Monitor* está detalhado

no Anexo B deste documento.

6.6 Lista de Materiais

A lista de todos os componentes necessários para fabricação de cada placa do *Vibration Monitor* está especificada na Tabela 5.

Designação	Descrição	Valor
BOOT, RESET	SMD 4X4X1.5MM Tactile Tact Push Button Micro Switch Momentary	-
C1	Capacitor (Semiconductor SIM Model)	4.7uF
C2	Capacitor (Semiconductor SIM Model)	100uF
C3	Capacitor (Semiconductor SIM Model)	1uF
C4, C5	Capacitor (Semiconductor SIM Model)	0.1uF
C6, C7, C8	Capacitor (Semiconductor SIM Model)	1nF
C9	Polarized Capacitor (Surface Mount)	47uF
C10	Polarized Capacitor (Surface Mount)	220uF
C11	Capacitor (Semiconductor SIM Model)	680uF
C12, C13	Capacitor (Semiconductor SIM Model)	10uF
D1, D2	General Purpose Rectifier	Diode 1N4007
D3	Surface LED 0402	RED
L1	Ferrite Beads 30 ohms 25% HIGH CURRENT	MH2029-300Y
L2	Inductor	33uH
P1	Power Socket Jack DC Female Plug 13.8x10.8x8.8mm	Jack DC - Female
P2	Header, 2-Pin	-
P3	Header, 5-Pin	-
P4	Header, 3-Pin	-
R1, R3, R4, R6	Resistor	4.7k
R2	Resistor	12k
R5, R7	Resistor	470R
R8	Resistor	1k
R9	Resistor	0R (1%)
U1	Node MCU Esp32	Node MCU ESP 32
U2	Accelerometer MMA 8451Q	MMA8451Q
U3	SIMPLE SWITCHER® Power Converter 150 KHz 3A Step-Down Voltage Regulator, 5-pin TO-263	LM2596S-5.0
VR1	Voltage Regulator LM1117-3.3 N05A SOT223 800mA	LM1117-3.3

Tabela 5 – Lista de Materiais do Dispositivo *Vibration Monitor*

6.7 Conclusões deste capítulo

Este capítulo apresentou o projeto do dispositivo desenvolvido. A partir das especificações de projeto, o esquemático eletrônico foi desenvolvido. Uma vez validado o esquemático, o protótipo da PCB foi desenvolvido e no segundo protótipo todas as especificações e funcionalidades foram atendidas. Os detalhes do código implementado também foram apresentados, junto com o fluxograma e detalhamento de cada função utilizada.

7 Bancada de Testes

Uma bancada de testes formada por um motor BLDC com 350 W - 36 V, projetado para uma roda de bicicleta, controlado por um driver trifásico, foi montada sobre uma mesa rígida. Sobre a mesa, foi adicionada uma superfície de granito para aumentar a inércia do eixo à mesa de suporte, conforme apresentado na Figura 39.

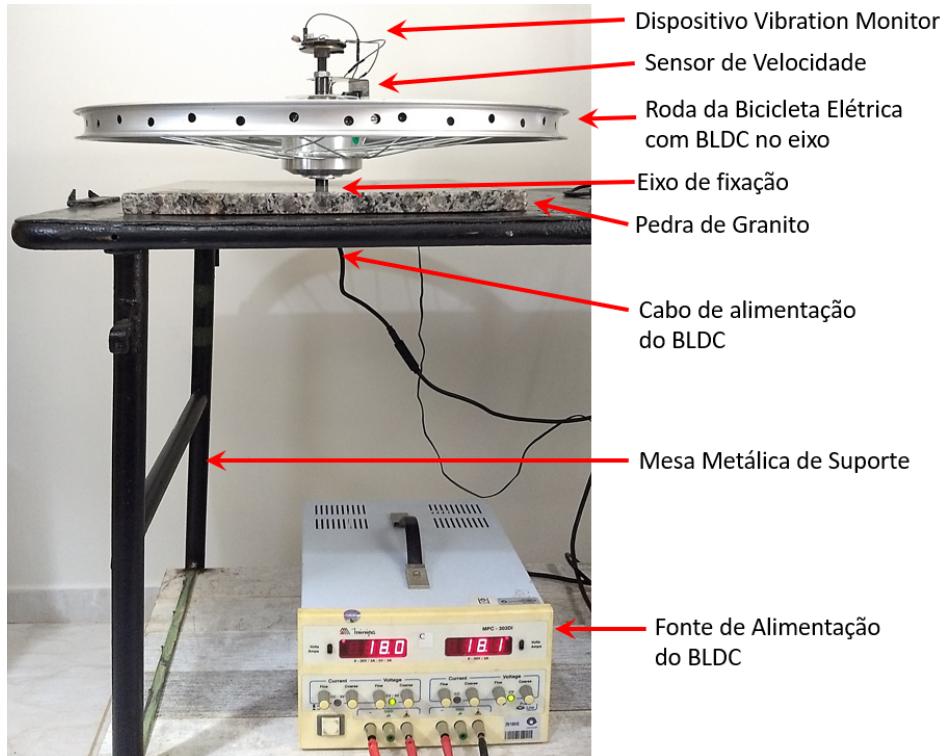


Figura 39 – Bancada de teste

Um motor BLDC como o próprio nome sugere, não utiliza escovas em sua comutação, sendo comutado eletronicamente. Ele apresenta diversas vantagens em relação ao motor AC convencional, como: menor ruído, maior vida útil, eliminação da ionização do comutador, redução da interreferência eletromagnética (EMI - *Electromagnetic interference*), maior confiabilidade e eficiência (YEDAMALE, 2003).

O BLDC vem sendo inserido cada vez mais no mercado, como por exemplo, no setor de compressores herméticos para refrigeração (BARATIERI, 2011), além de sua franca expansão na indústria automobilística, aeroespacial, médica, equipamentos de automação industrial, instrumentação, entre outras (YEDAMALE, 2003). O motor BLDC é um tipo de motor síncrono alimentado por inversor (*driver*) de corrente contínua. Este tipo de motor requer

um controlador eletrônico de velocidade. A Figura 40a apresenta o fluxograma do driver de acionamento e a Figura 40b a manopla de liga e desliga do motor BLDC utilizado na bancada de testes.

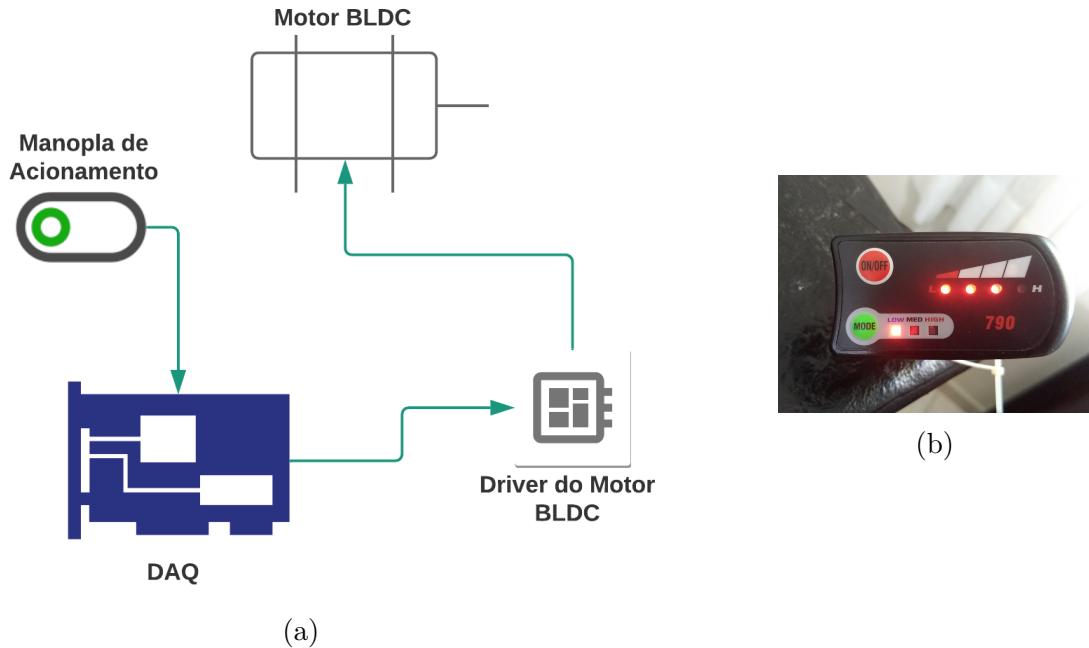


Figura 40 – Detalhes do acionamento do BLDC: (a) fluxograma do driver e DAQ (b) manopla de liga e desliga

O controlador do motor BLDC não faz parte do escopo deste projeto pois utilizou-se o mesmo controlador desenvolvido por Silva (2017). Para a medição da velocidade do motor BLDC, um sensor de velocidade do tipo interruptor óptico foi instalado no eixo no motor. Este sensor é responsável pela aquisição do sinal de velocidade através da interrupção do feixe infravermelho a cada volta. O sensor instalado é apresentado na Figura 41a e o detalhe do sensor é apresentado na Figura 41b.

7.1 Simulador de Carga

Um sistema de simulação de carga mecânica para o motor da bancada de teste foi desenvolvido. Um aspecto importante deste sistema é a inexistência de contato físico entre o motor e a carga. Assim sendo, uma estrutura metálica rígida foi projetada para suportar dez ímãs de neodímio (NdFeB N25) em formato circular com dimensões $\phi 30 \times 7$ mm, fixada sobre a bancada de teste e parafusada em um suporte vertical. Os ímãs foram fixados numa placa metálica circular com um orifício central para passagem do dispositivo de medição, em distâncias exatamente iguais entre si e com orientações dos pólos intercaladas, garantindo

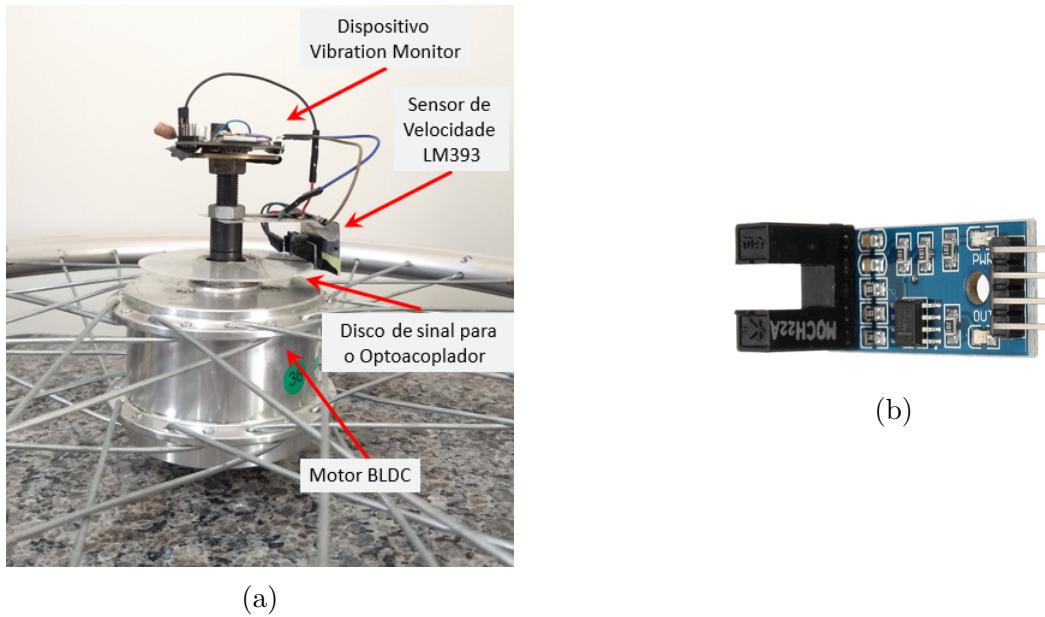


Figura 41 – Medição da velocidade do motor: (a) Sensor instalado no eixo do motor (b) detalhe do optoacoplador LM393

maior homogeneidade nas forças de atração magnética. O projeto do suporte de simulação de carga é mostrado na Figura 42a. Na Figura 42b, é apresentado o esquema de disposição dos ímãs e, na Figura 43, é apresentado o simulador de carga instalado no local.

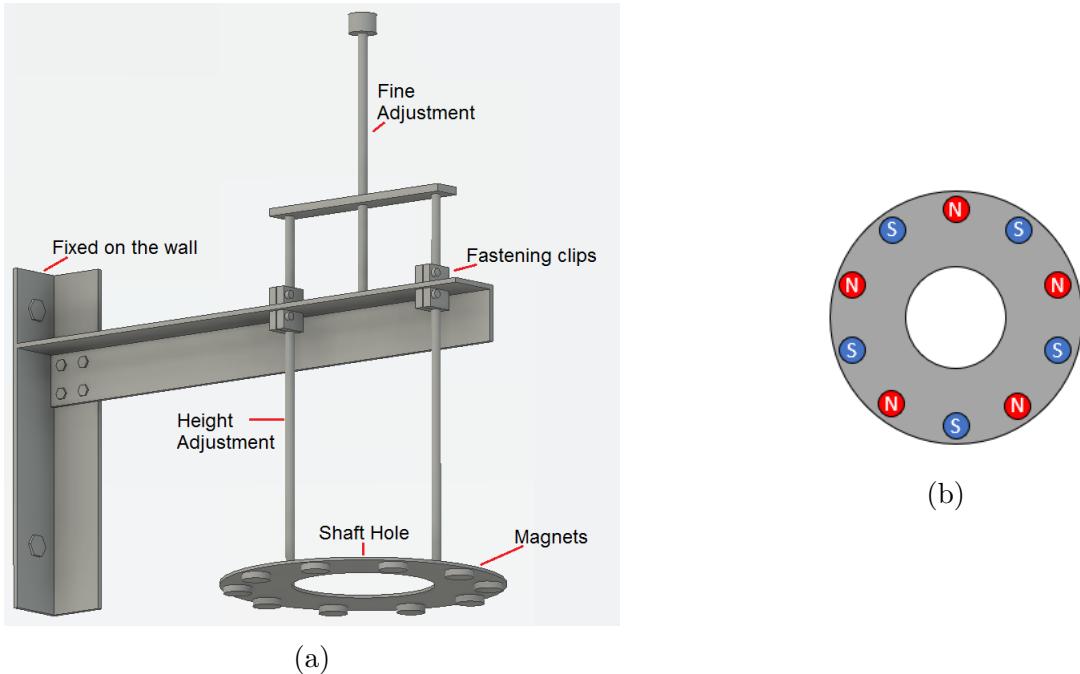


Figura 42 – Simulador de carga: (a) Desenho do todo (b) Detalhe da disposição dos ímãs

A carga é simulada alterando-se a altura da placa de ímãs do suporte. Neste esquema é



Figura 43 – Fotografia do aparato de simulação de carga

utilizado o mesmo princípio de frenagem magnética, que utiliza as interações eletromagnéticas entre um sistema em movimento e um sistema em repouso. A lei de Faraday estabelece que a variação de fluxo magnético (devido à rotação da roda) na área das raias condutoras, que é perpendicular ao fluxo, provoca o surgimento de forças eletromotrices induzidas, fazendo circular corrente elétrica induzida. De acordo com a lei de Lenz, o sentido de circulação dessas correntes é tal que elas tendem a anular o efeito que as induziu. O efeito que produz tais correntes é a passagem das raias através das linhas de campo dos ímãs suspensos (SADIKU, 2007).

As correntes aparecem então em cada lado das raias em um sentido que provocam o surgimento de um campo magnético cuja polaridade é contrária à do ímã que está sobre ele. Como consequência, ocorrerá atração entre a superfície das raias e cada um dos ímãs suspensos, fazendo com que a roda seja freada.

7.2 Simulador de Desbalanceamento

A inserção de desbalanceamento na roda foi feita adicionando-se contrapesos de chumbo em suas raias. Tais contrapesos são do mesmo tipo utilizado para balanceamento de pneu automotivo e ilustrado na Figura 44a. A posição do contrapeso e seu modo de fixação na roda está apresentado na Figura 44b.

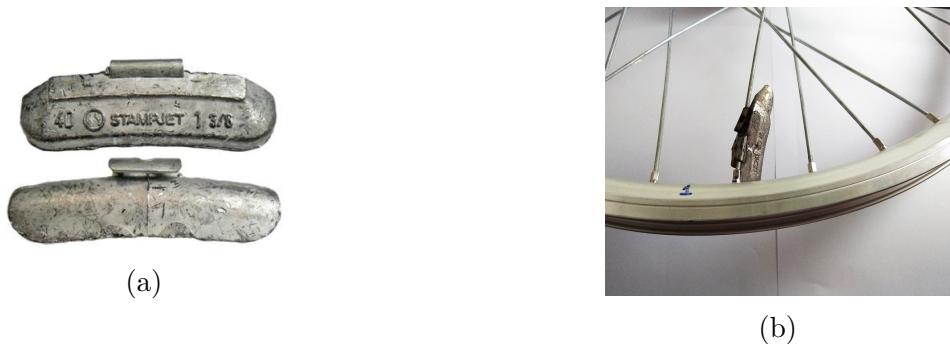


Figura 44 – Simulação de desbalanceamento: (a) Contrapeso (b) Roda com contrapeso

7.3 Simulador de Obstrução

A simulação de obstrução na roda é feita utilizando um aparato construído com fitas de nylon. Estas fitas simulam um ponto de obstrução à rotação, sem danificar a roda. O objeto desenvolvido está apresentado na Figura 45a e na Figura 45b ele está posicionado para realizar a obstrução na roda.



Figura 45 – Objeto para simular obstrução: (a) Obstrutor (b) Roda com obstrutor

Com isto é possível simular três tipos de interferências mecânicas, sendo: frenagem, desbalanceamento e obstrução no rotor.

7.4 Rede de Sensores

O dispositivo *Vibration Monitor* pode ser conectado a uma rede de sensores. O propósito desta aplicação é possibilitar ao usuário um monitoramento em diversos pontos do equipamento, permitindo um estudo de falhas e causas em relação à posição do sensor no

equipamento. Na bancada de testes foram utilizados dois sensores, que mediram as vibrações simultaneamente.

A Figura 46 apresenta um esquema de ligação de dois dispositivos *Vibration Monitor* (Sensor Node 1 e Sensor Node 2) instalados na bancada de testes e coletando simultaneamente os dados de vibração. A Figura 47 apresenta o detalhe do Sensor Node 2 que foi instalado na parte inferior da bancada de testes.

Neste caso, o eixo do motor BLCD é fixo e a parte girante é a parte externa. Dessa forma, foi possível inserir o dispositivo exatamente sobre o eixo do motor. Em motores convencionais, o dispositivo deverá ser instalado o mais próximo possível do eixo, no caso, sobre o mancal.

O sistema comporta quantos sensores o usuário desejar, já que o único limitante neste caso é sistema de armazenamento em nuvem. Cada sensor trabalha de forma independente, não interferindo na taxa de aquisição final. Como os dados são enviados à nuvem utilizando o protocolo MQTT, não há problemas de congestionamento de transmissão.

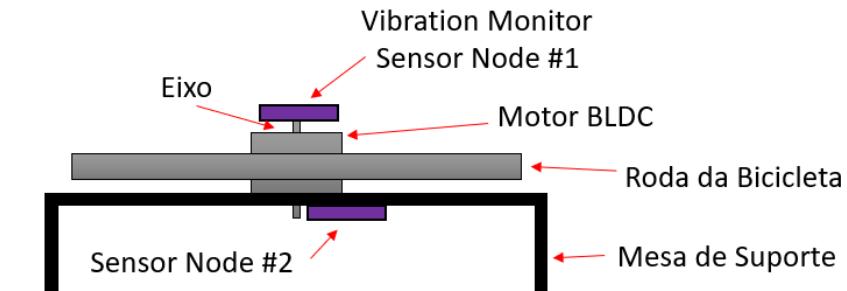


Figura 46 – Bancada de testes com 2 sensores coletando informações simultaneamente, formando uma rede de sensores

7.5 Conclusões deste capítulo

Este capítulo apresentou o desenvolvimento do dispositivo proposto e nomeado como *Vibration Monitor*. Além disso, a bancada de teste desenvolvida e suas especificações foram apresentadas. O projeto de simulação de carga também foi apresentado, detalhando o princípio físico de simulação de carga. Também foi abordada a utilização do dispositivo como uma rede de sensores, podendo ser utilizado para medir diversos pontos de vibração em um mesmo equipamento, o que poderá estudar a correlação dos sinais em diferentes pontos da máquina para cada interferência mecânica ou falha introduzida.

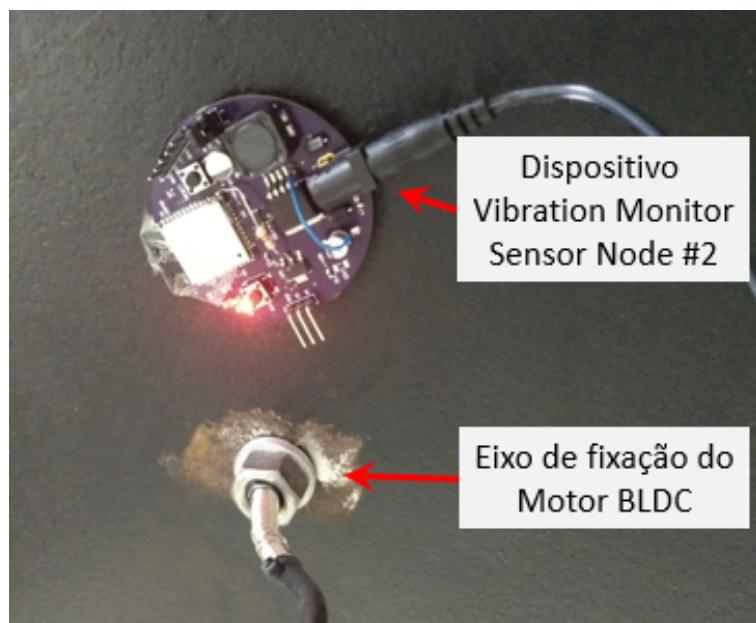


Figura 47 – Instalação do Sensor Node 2 na bancada de testes

8 Resultados

Neste capítulo serão detalhados os resultados experimentais obtidos e as conclusões.

8.1 Avaliação do acelerômetro MEMS utilizando o kit de desenvolvimento da NXP FRDM-KL25Z

Para realizar os testes experimentais preliminares, optou-se por utilizar uma placa de aquisição de dados de vibração de mercado que atendesse às especificações do projeto. A placa escolhida que atendeu a esses requisitos foi o kit de desenvolvimento da NXP, o FRDM-KL25Z que já possui o sensor MMA8451Q embarcado. Os testes experimentais preliminares foram feitos utilizando uma roda de bicicleta e a placa de desenvolvimento.

Os dados de aceleração nos eixos X e Y foram coletados através de uma comunicação serial RS232 e processado utilizando o Matlab. Diversos pesos foram inseridos sempre no mesmo ponto do aro da roda de bicicleta e os dados de aceleração foram coletados para cada peso inserido (de 0 g a 210 g). A FFT do sinal medido foi calculada para estudar a magnitude tanto no eixo X quanto no Y. A placa de aquisição foi montada com seu centro de massa alinhado com o eixo e os pesos na extremidade das raias da roda, conforme apresentado na Figura 48.

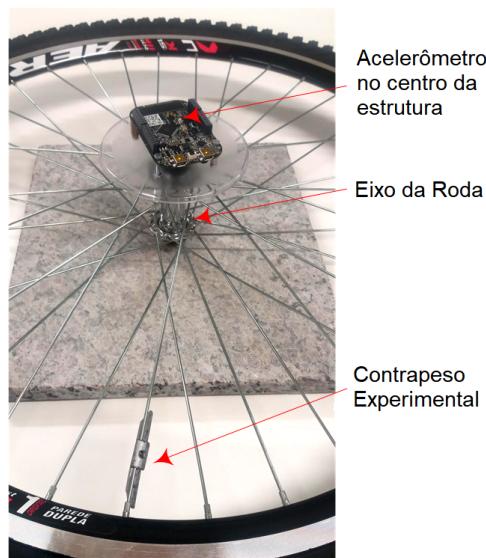


Figura 48 – Primeiro protótipo de testes

Através de resultados experimentais foi possível concluir que utilizar este sensor foi

viável e que o peso de desbalanceamento apresenta um aumento linear na amplitude da vibração analisando seu espetro. Neste caso a amplitude afetada corresponde a frequência de velocidade de rotação, ou seja, a frequência fundamental, conforme o gráfico apresentado na Figura 49. Estes resultados foram publicados e apresentados no *International Symposium on Instrumentation Systems, Circuits and Transducers* (INSCIT 2016). O aumento linear de desbalanceamento em função do peso aplicado condiz com o embasamento teórico apresentado na seção 3.1.1.

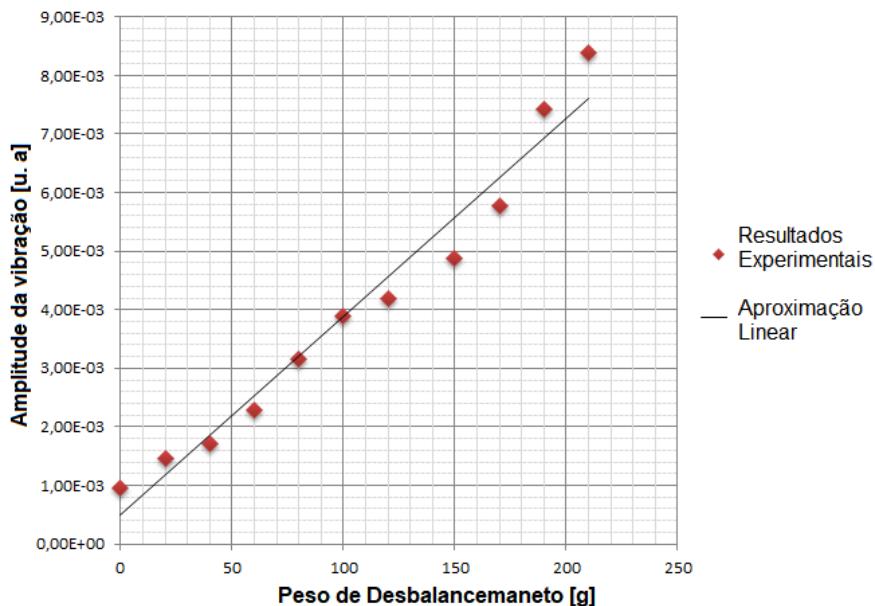


Figura 49 – Magnitude de desbalanceamento em função do peso aplicado

8.2 Avaliação de distúrbios em um motor BLDC

Medições de vibração foram realizadas com o objetivo de investigar distúrbios aplicados em um motor BLDC, utilizando o mesmo kit de desenvolvimento apresentado na seção 8.1. Além do desbalanceamento, foram inseridas simulações de carga utilizando o aparato apresentado na seção 7.1. A velocidade de rotação do motor foi medida através de um *reed-switch*. A montagem do experimento é apresentada na Figura 50.

Os dados de vibração foram coletados de forma similar ao apresentado na seção anterior. Entretanto, a estratégia de análise do resultado obtido utilizou uma análise visual através de gráficos tipo *waterfall*. A Figura 51a apresenta o gráfico waterfall com o motor submetido a um desbalanceamento de 90 g sem carga e a Figura 51b submetido ao mesmo desbalanceamento, porém com carga.

Através de resultados experimentais concluiu-se que foi possível observar cada distúrbio inserido de forma independente, utilizando análise visual através do gráfico waterfall. Estes resultados foram publicados e apresentados no *INSCIT 2017*. Uma versão estendida desse artigo foi publicado na revista *IEEE Instrumentation and Measurement Magazine* (Vol. 20, No. 6, December 2017) com o título: *Fault Diagnostics in Rotary Machines through Spectral Vibration Analysis Using Low-Cost MEMS Devices*.

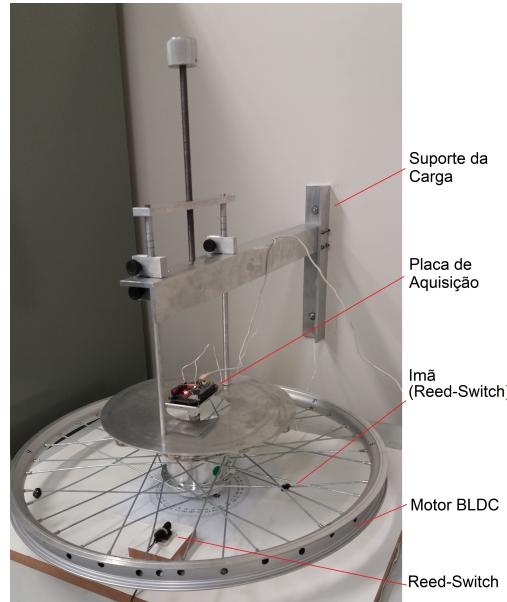


Figura 50 – Primeira montagem do aparato experimental do BLDC

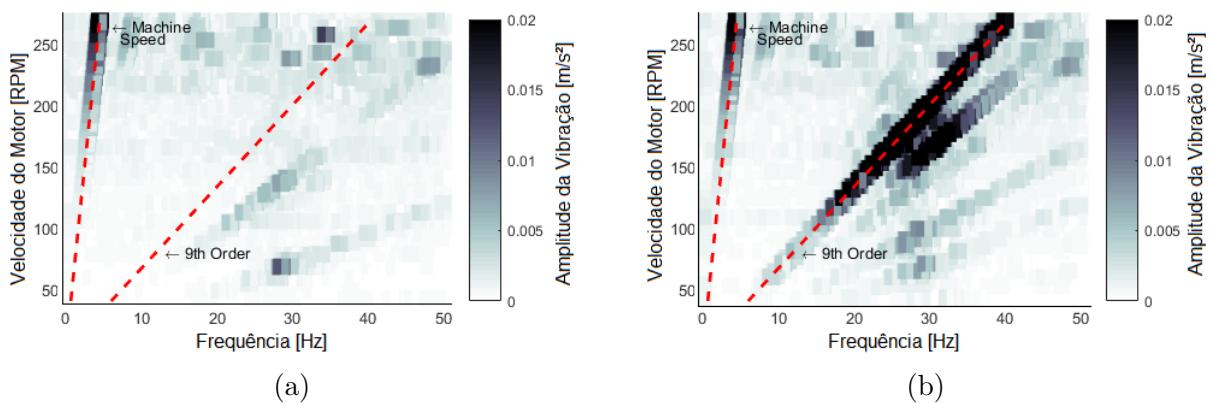


Figura 51 – Gráfico Waterfall Frequência vs RPM vs Magnitude da Vibração com desbalanceamento de 90 g: (a) sem carga (b) com carga a 20 mm

O fato do sistema sob carga afetar o nono ($9\ddot{\zeta}$) harmônico está relacionado ao número de fases que *driver* o motor BLDC possui, neste caso, trifásico.

8.3 Avaliação de distúrbios em uma bomba centrífuga

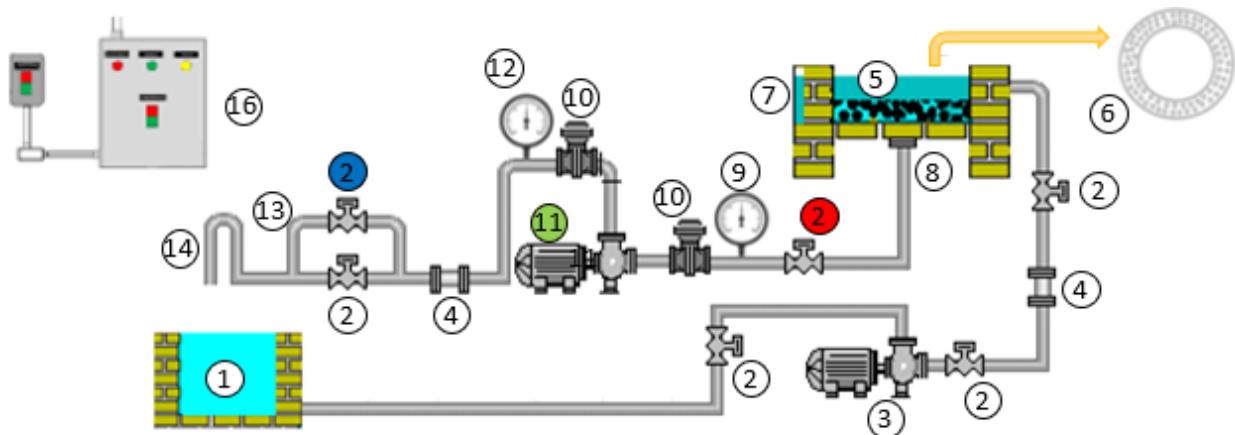
Uma bomba centrífuga também foi objeto de estudo em parceria com a FEC (Faculdade de Engenharia Civil), onde diversos distúrbios puderam ser inseridos e mensurados utilizando o mesmo kit de desenvolvimento apresentado na seção 8.1. Neste experimento, três tipos de distúrbios foram avaliados: estado do rotor (novo ou avariado), fixação da base da bomba (fixa ou frouxa) e presença de cavitação, este último sob três condições (sem cavitação, cavitação incipiente e cavitação severa).

Cavitação é um termo utilizado para indicar a formação de bolhas ou cavidades de vapor devido à redução de pressão durante o transporte de líquidos. Nos sistemas de bombeamento, o líquido transportado pode vaporizar na sucção da bomba ou em seus rotores, limitando a capacidade de sucção e causando efeitos indesejáveis como: flutuação de pressão, vibração, ruído adicional e erosão. Ou seja, a cavitação é uma variável indesejada em qualquer ambiente industrial, e sua observação pode ser determinante no aumento da vida útil das pás do rotor, acoplamentos, entre outros. A presença de cavitação gera uma vibração característica na bomba passível de detecção (SCHRÖDER *et al.*, 2015). A Figura 52 apresenta o fluxograma da planta de teste utilizada. O dispositivo foi fixado sobre o mancal conforme apresentado na Figura 53.

Um parâmetro comumente utilizado para caracterizar a quantidade de cavitação é o índice NPSH (*Net Positive Suction Head*). O cálculo do NPSH é o mais utilizado em projetos de bombas centrífugas (SCHRÖDER *et al.*, 2015). Nesse caso, dois índices são calculados, o NPSHa (*NPSH available*) e o NPSHr (*NPSH required*). O NPSHa refere-se à pressão total do sistema medida no flange da bomba de sucção em um fluxo e temperatura da água específicos. O NPSHr representa a energia do líquido no flange de sucção da bomba, necessária para alcançar o rotor, sem vaporização (SCHRÖDER *et al.*, 2015).

Por necessitar de instalação de medidores de pressão em dois pontos do sistema, este método é invasivo e requer parada do processo para instalação. É possível avaliar se um sistema está sob cavitação analisando o sinal de vibração na bomba. Dessa forma, o dispositivo desenvolvido é um atrativo nesse tipo de aplicação, principalmente por se tratar de um método não invasivo de análise.

Neste trabalho, exploramos outras técnicas de análise de dados também utilizando a FFT do sinal de vibração medido. Ressalta-se que o foco desse trabalho não é a análise dos dados e sim o dispositivo de coleta de dados de vibração. A utilização de diversas técnicas de análise possibilita demonstrar que o dispositivo é suficientemente capaz de medir dados de vibração de forma satisfatória, cabendo ao usuário utilizar a forma de análise que mais lhe agrada.

**LEGENDA**

- | | |
|------------------------|--|
| ① Tanque inferior | ⑨ Manovacuômetro; |
| ② Válvula | ⑩ Transmissor de Pressão; |
| ③ Bomba de alimentação | ⑪ Bomba centrífuga (Objeto de estudo); |
| ④ Medidor de vazão | ⑫ Manômetro; |
| ⑤ Tanque superior | ⑬ By-pass; |
| ⑥ Redutor de fluxo | ⑭ Sifão; |
| ⑦ Régua de nível | ⑮ Painel de comando do inversor de |
| ⑧ Coletor de água | frequência da bomba centrífuga. |

Figura 52 – Planta experimental de testes com bomba centrífuga

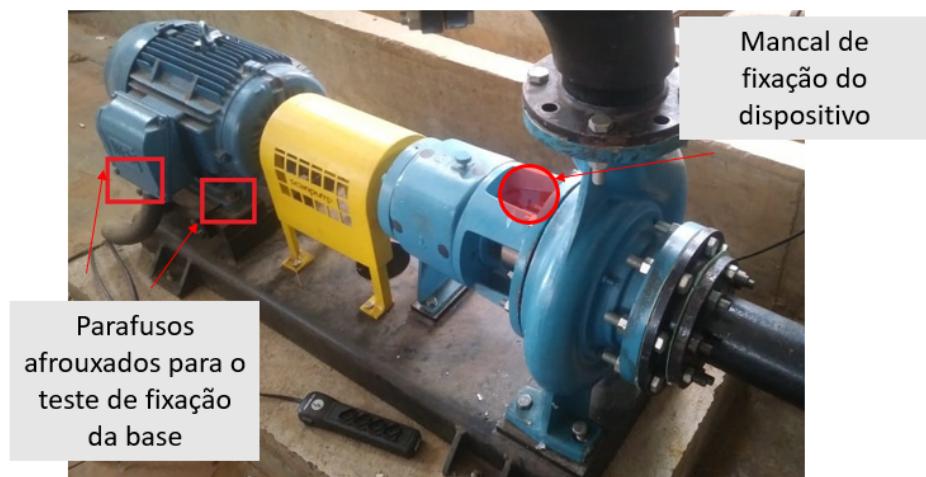


Figura 53 – Bomba centrífuga utilizada no experimento: parafusos de fixação da base e posicionamento do dispositivo no mancal

Tabela 6 – Testes realizados

Teste	Pás do Rotor	Fixação da Base	Cavitação	RPM	Rótulo da Classe
1	Novo	Fixa	Sem	76	[0,0,1,0,0]
2	Novo	Fixa	Incipiente	76	[0,0,0,1,0]
3	Novo	Fixa	Severa	76	[0,0,0,0,1]
4	Novo	Frouxa	Sem	76	[0,1,1,0,0]
5	Novo	Frouxa	Incipiente	76	[0,1,0,1,0]
6	Novo	Frouxa	Severa	76	[0,1,0,0,1]
7	Danificado	Fixa	Sem	76	[1,0,1,0,0]
8	Danificado	Fixa	Incipiente	76	[1,0,0,1,0]
9	Danificado	Fixa	Severa	76	[1,0,0,0,1]
10	Danificado	Frouxa	Sem	76	[1,1,1,0,0]
11	Danificado	Frouxa	Incipiente	76	[1,1,0,1,0]
12	Danificado	Frouxa	Severa	76	[1,1,0,0,1]
13	Novo	Fixa	Sem	50	[0,0,1,0,0]
14	Novo	Fixa	Incipiente	50	[0,0,0,1,0]
15	Novo	Fixa	Severa	50	[0,0,0,0,1]
16	Novo	Frouxa	Sem	50	[0,1,1,0,0]
17	Novo	Frouxa	Incipiente	50	[0,1,0,1,0]
18	Novo	Frouxa	Severa	50	[0,1,0,0,1]
19	Danificado	Fixa	Sem	50	[1,0,1,0,0]
20	Danificado	Fixa	Incipiente	50	[1,0,0,1,0]
21	Danificado	Fixa	Severa	50	[1,0,0,0,1]
22	Danificado	Frouxa	Sem	50	[1,1,1,0,0]
23	Danificado	Frouxa	Incipiente	50	[1,1,0,1,0]
24	Danificado	Frouxa	Severa	50	[1,1,0,0,1]

Neste caso, três tipos de classificadores foram empregados: regressão logística, Linear SVM (*Support Vector Machine*) e Rede Neural Artificial (RNA). Foram realizados 24 ensaios no total, onde 12 deles foram repetidos para duas velocidades de rotação: 76% (1368 RPM) e 50% (900 RPM). A combinação e características dos distúrbios aplicados estão resumidas na Tabela 6.

Na tabela 6, o *target binário* indica quais valores foram considerados para cada condições de treino dos classificadores. No vetor, o primeiro elemento indica as condições das pás do rotor, onde caso igual a ‘0’, as pás são novas, caso ‘1’, as pás estão danificadas. Uma pá danificada é uma pá que já sofreu avarias devido à cavitação, perdendo parte da massa da sua estrutura, que fica mais rugosa. Isso deixa as pás mais frágeis e susceptíveis a avarias mais graves. As pás utilizadas nos testes estão apresentadas na Figura 54.

A segunda posição indica qual é o estado de fixação da base, onde ‘0’ é fixa e ‘1’ é frouxa. A terceira, quarta e quinta posições indicam em qual estado de cavitação a máquina



Figura 54 – Pás da bomba utilizadas no experimento: Nova (à esquerda), Avariada (no centro) e Detalhe da Avariada (à direita)

se encontra, onde ‘1’ na terceira posição indica sem cavitação, ‘1’ na quarta indica cavitação incipiente e ‘1’ na quinta posição indica cavitação severa. Quando uma condição de cavitação for setada em ‘1’, obrigatoriamente as outras duas posições serão setadas em ‘0’.

As condições de cavitação consideradas foram baseadas na pressão indicada por um manômetro instalado na parte de sucção da bomba. Os valores considerados para cada perfil de operação são apresentados na Tabela 7. Uma vez ajustada a pressão na condição desejada, os dados eram coletados, ou seja, os dados eram coletados em regime permanente em cada condição.

Tabela 7 – Condição de Cavitação x Pressão Medida

Condição de Cavitação	Pressão Medida (Manômetro)
Sem	0,5 psi
Incipiente	0,3 psi
Severa	0,1 psi

Foi realizada uma comparação do processamento de dados utilizando todo o espectro de frequência do sinal e a seleção das dez primeiras freqüências harmônicas com a velocidade do rotor da bomba. O objetivo desse estudo foi investigar se a seleção de dados harmônicos é suficientemente robusta como entrada de dados em diversas formas de análise.

Quanto menor a quantidade de dados de entrada, maior a velocidade de processamento da análise e menor será o tempo de transmissão caso seja utilizada uma rede sem fio. Assim, constatou-se que a utilização dos valores de magnitude dos principais harmônicos não afetam de maneira significativa o resultado do diagnóstico obtido em algumas análises utilizadas, conforme apresentado na Tabela 8.

Os resultados foram comparados para avaliar qual deles obteve o melhor desempenho.

Tabela 8 – Comparaçao dos resultados obtidos utilizando trés formas de análise e nmeros diferentes de entradas de dados

Falhas	Regressão Logística			Linear SVM			MLP		
	Todo Espectro	10 prim. harm.	Var. Perc.	Todo Espectro	10 prim. harm.	Var. Perc.	Todo Espectro	10 prim. harm.	Var. Perc.
Pás do Rotor	82,0	79,5	3,0	91,2	85,2	6,6	100,0	89,0	11,0
Fixação da Base	63,2	56,0	11,4	69,9	64,1	8,3	92,4	77,8	15,8
Sem Cavitação	62,3	62,3	0,0	86,8	76,1	12,3	99,7	94,0	5,7
Cavitação Incipiente	67,9	68,1	0,3	71,9	69,4	3,5	99,1	92,3	6,9
Cavitação Severa	69,6	69,6	0,0	75,9	69,6	8,3	99,4	88,3	11,2

Portanto, a rede neural (RNA) MLP forneceu melhores resultados. Os resultados obtidos com a RNA-MLP foram capazes de diagnosticar com mais de 92% de precisão na previsão das falhas mecânicas aplicadas e com cerca de 99% de precisão na avaliação do status da cavitação.

O acelerômetro usado para adquirir os dados de vibração foi capaz de fornecer dados suficientes para extraer recursos para analisar a saúde da bomba centrífuga. Um processamento de dados deve ser feito para aplicação dos classificadores. A extração de características foi considerada fundamental para a obtenção destes resultados e, combinada com outras características possíveis, as previsões poderiam ser melhores ainda. Os objetivos foram alcançados e cada uma das falhas analisadas. Até a cavitação incipiente pode ser detectada com um alto grau de confiabilidade.

8.4 Utilização da solução desenvolvida na análise de distúrbios em um motor BLDC

O protótipo do dispositivo final, apresentado na Figura 36, foi utilizado para coletar e processar os dados, calcular e enviar os seis primeiros harmônicos para um servidor na nuvem utilizando conexão Wi-Fi, de acordo com o objetivo desse trabalho.

Para analisar a vibração no motor BLDC (bancada de testes detalhada na Seção 7), dois sensores foram adicionados à bancada de testes, formando uma WSN (*Wireless Sensor Network*). A inserção de um outro nó sensor no mesmo objeto de estudo tinha dois objetivos principais: aumentar a quantidade de dados para análise, verificar como os dados estão relacionados; avaliar o impacto da posição do sensor no equipamento de teste e analisar a conformidade condizente com a seção 3.2 que deve ser o mais próximo possível do eixo.

Foram coletados aproximadamente 2 h de dados em cada teste, sendo realizados oito testes inserindo-se três falhas: desbalanceamento na roda, carga e obstrução na roda.

A adição de carga no sistema foi feita utilizando o equipamento (freio magnético) apresentado na seção 7.1. A inserção de desbalanceamento na roda foi feita adicionando-se 90 g de contrapeso de chumbo em uma das raias, conforme detalhado na seção 7.2. A simulação de obstrução na roda é feita utilizando um aparato construído com fitas de nylon, conforme detalhado na seção 7.3.

O processamento dos dados foi feito utilizando a linguagem de programação Python e suas bibliotecas de *machine learning*: Keras e Tensorflow. Dois tipos de classificadores foram utilizados para realizar a análise desses dados: regressão logística e ANN-MLP (*Artificial Neural Network - Multi-Layer Perceptron*). Essas técnicas de análise estão disponíveis no Anexo C. Os dados foram divididos em conjuntos com a seguintes proporção: 70% para treinamento, 20% para teste e 10% para validação. Os códigos desenvolvidos em Python para processamento dos sinais estão disponíveis no Anexo D.

Os sensores foram fixados utilizando ímãs, o que torna o dispositivo fácil de instalar. O sensor node número 1 (sensor 1) foi fixado no eixo e o sensor node número 2 (sensor 2) foi fixado embaixo da mesa do suporte, próximo ao eixo, conforme apresentado na Figura 46. O sistema foi submetido a oito condições de operação conforme os itens listados na Tabela 9.

Tabela 9 – Condições de Operações Testadas

Teste	Desbalanceado	Carga	Obstrução
T1	0	0	0
T2	0	0	1
T3	0	1	0
T4	0	1	1
T5	1	0	0
T6	1	0	1
T7	1	1	0
T8	1	1	1

Utilizando a regressão logística para analisar os dados, foi possível obter os resultados apresentados na Tabela 10. Os dados obtidos através da utilização de uma RNA treinada com uma camada de 100 e outra de 50 neurônios, conforme modelo apresentado na Figura 55, são apresentados na Tabela 11.

Os nós sensores adquiriram os dados do MEMS dos eixos X e Y. Esses sinais foram processados utilizando um algoritmo FFT para obter seu espectro de vibração. Com esse espectro, o microcontrolador calcula as seis primeiras ordens harmônicas relativas à velocidade

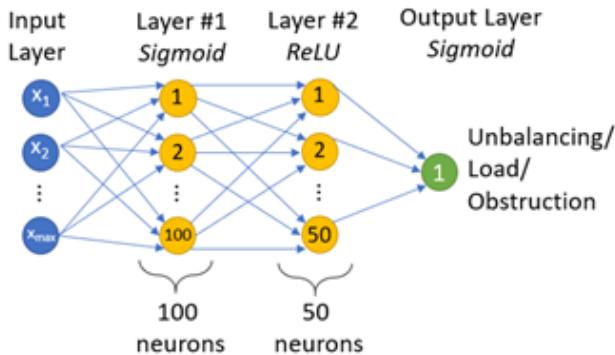


Figura 55 – Modelo da RNA utilizado nesta análise

Tabela 10 – Precisão na detecção de falhas utilizando Regressão Logística como classificador

	Sensor 1			Sensor 2		
	Desbal. (%)	Carga (%)	Obst. (%)	Desbal. (%)	Carga (%)	Obst. (%)
Treinamento	100,0	78,0	99,0	100,0	75,0	94,0
Diagnóstico	99,9	78,8	98,3	99,8	74,5	94,3

Tabela 11 – Precisão na detecção de falhas utilizando RNA como classificador

	Sensor 1			Sensor 2		
	Desbal. (%)	Carga (%)	Obst. (%)	Desbal. (%)	Carga (%)	Obst. (%)
Treinamento	99,7	90,0	99,0	100,0	82,0	96,0
Diagnóstico	100,0	88,2	99,1	100,0	79,6	94,4

da máquina. O sistema envia para a nuvem esses seis valores harmônicos calculados para os eixos X e Y de cada nó.

O sistema permaneceu estável e não sofreu interferência eletromagnética durante esse experimento, garantindo a integridade da transferência de dados para todos os testes. O nó do sensor foi capaz de calcular e enviar os dados conforme programado por meio de uma conexão Wi-Fi. Os dados foram coletados e armazenados usando uma plataforma de sistema em nuvem gratuita.

Foi possível diagnosticar cada tipo de perturbação aplicada na máquina utilizando a amplitude da FFT como parâmetro para comparar níveis aceitáveis em cada ordem harmônica. Esses mesmos dados foram aplicados a três tipos de classificadores: uma comparação do nível de amplitude da FFT, a regressão logística e um NN.

Os resultados mostram que, usando os dados coletados, o NN obteve a melhor precisão e foi capaz de diagnosticar 100% de precisão no desequilíbrio de detecção e acima de 99% na detecção de obstrução e acima de 83% na carga de diagnóstico na máquina.

Por outro lado, a posição do dispositivo *Vibration Monitor* foi essencial para obter melhores resultados, fato observado ao comparar as diferenças significativas de amplitudes entre os dois locais. A partir dos resultados, sugere-se que, nesse caso, a melhor posição esteja próxima ao eixo do rotor. As técnicas de análise de dados usadas neste estudo servem para verificar se o dispositivo coleta informações representativas. Como os dados coletados estão disponíveis na nuvem, o usuário pode aplicar qualquer outra técnica de análise.

O dispositivo foi capaz de detectar condições como desbalanceamento, obstrução no elemento girante e aplicação de carga. Para que o dispositivo torne-se um produto, ainda é necessário projetar um envólucro adequado para protegê-lo contra poeira, umidade e choque mecânico. Uma alternativa ao sensor de velocidade é usar algoritmos para estimar a velocidade do rotor a partir de sinais como as correntes de entrada. Isso poderia reduzir ainda mais os custos do sistema completo, tornando-o mais competitivo.

8.5 Visualização dos dados na nuvem

O primeiro passo para enviar os dados à nuvem é realizar uma conexão bem sucedida com a rede Wi-Fi disponível. Deste modo, observou-se que o dispositivo leva menos de 15 s para se conectar com sucesso a rede Wi-Fi escolhida, iniciando posteriormente a coleta e pré-análise dos dados.

Em seguida, uma vez configurada à nuvem, os dados enviados pelo dispositivo ao repositório online podem ser acessados remotamente. Foi utilizado o serviço *IBM Watson IoT Platform*, cujo controle de acesso é dado de forma convencional por meio de usuários e senhas específicos. Este serviço é gerenciado e hospedado em nuvem de forma a simplificar a extração de dados de dispositivos IoT. A escolha desta plataforma foi motivada por ser gratuita e oferecer diversos serviços para aplicações, como inteligência artificial e outras formas de análise de dados. O serviço gratuito é limitado em tráfego de dados mensais.

Neste ambiente é possível criar painéis de controle *dashboards* de acordo com a necessidade. A primeira coisa a verificar na plataforma é se os dispositivos encontram-se *online*. Para isso, basta verificar se um círculo verde aparece no nome do dispositivo já configurado, conforme apresentado na Figura 56.

Uma vez criados os *dashboards* desejados, basta acessá-los através do painel de navegação, conforme Figura 57.

A Figura 58 apresenta o *dashboard* criado para visualização dos dados do sensor node 1. Já a Figura 59 apresenta detalhes de navegação em um gráfico específico, onde é possível ampliar o tempo de apresentação e buscar dados históricos.

ID do dispositivo	Tipo de dispositivo	ID de classe	Data de inclusão
Sensor1	Esp32-S1	Dispositivo	27 de Jul de 2018 às 15:53
Sensor2	Esp32-S1	Dispositivo	31 de Jul de 2018 às 14:34

Figura 56 – Listas de dispositivos e seu estado

Figura 57 – Listas de opções do IBM Watson

Apesar do uso poderoso da ferramenta, esta não é a melhor escolha caso deseje disponibilizar dados de leitura de forma mais simples e fácil. Para isso, existe uma ferramenta muito versátil para o desenvolvimento de aplicações voltadas a IoT, o Node-RED. O Node-RED utiliza uma programação gráfica que facilita a elaboração de aplicações. A Figura 60 apresenta parte da aplicação desenvolvida com o Node-RED.

Os dados são acessados através de um endereço da web, administrados por um serviço



Figura 58 – Dashboard para o Sensor Node 1

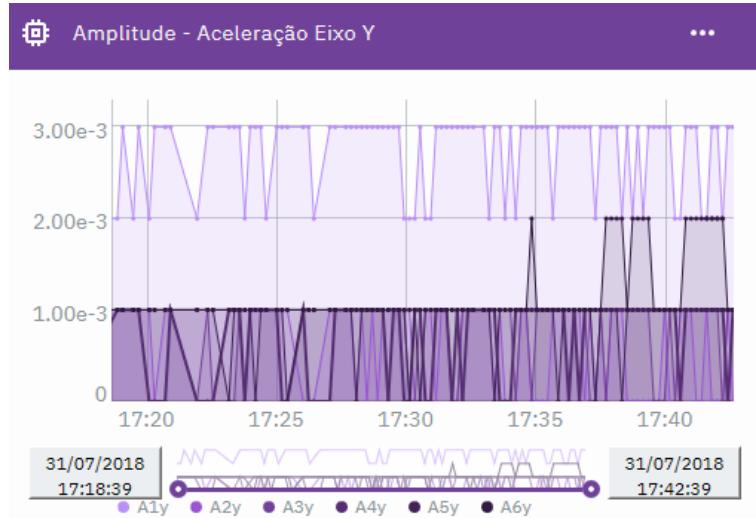


Figura 59 – Visualização amplificada dos dados de amplitude das seis harmônicas correspondentes ao eixo y do sensor node 1

chamado *bluemix*. Para proteção dos dados, uma página de login foi criada, conforme apresentado na Figura 61. A Figura 62 apresenta um apanhado de informações sobre o Sensor Node 1 e a Figura 63a apresenta uma visão maior do gráfico, onde é possível realizar análises específicas.

Além disso, a ferramenta também é capaz de ser ajustada para fornecer alguns alertas, por exemplo, se uma determinada amplitude extrapolar um valor máximo estipulado, o sistema pode emitir um alerta, contabilizando quantos alertas semelhantes ocorreram num determinado espaço de tempo. Isso pode ajudar a verificar quando um determinado patamar

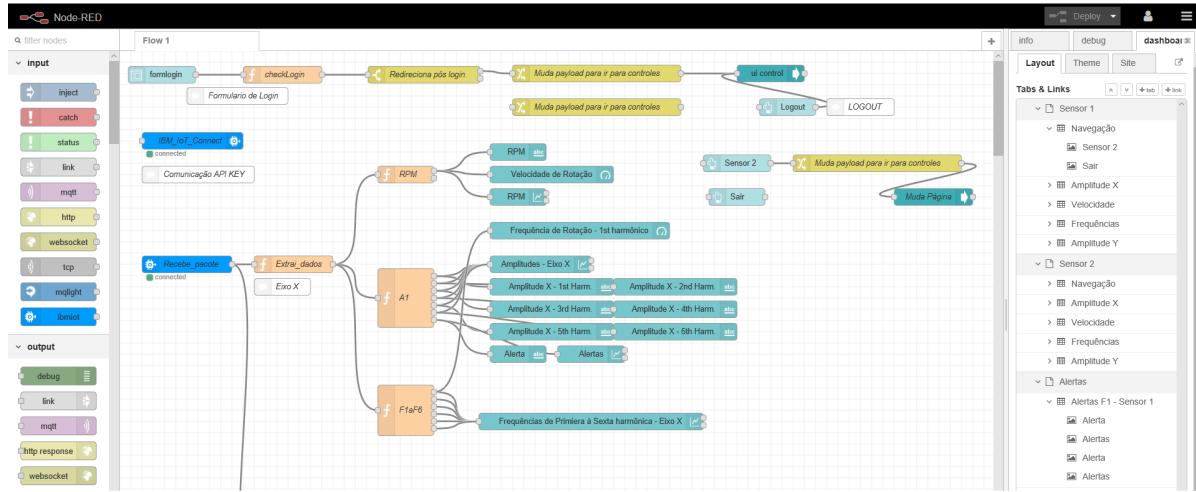


Figura 60 – Ambiente de programação do Node-RED

de vibração teve sua frequência de ocorrências aumentada, fazendo com que a atenção àquele evento seja redobrada e que análises mais profundas possam ser realizadas. A Figura 63b apresenta um exemplo de alertas criados.

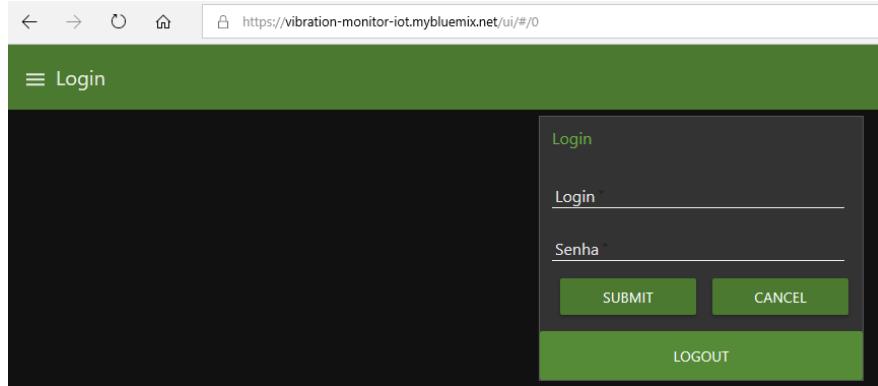


Figura 61 – Login para acessar os dados criados com o Node-RED

Como o ambiente é em nuvem, é possível criar um repositório de dados de maneira que um fabricante de máquinas rotativas possa analisá-las individualmente, mesmo instaladas em diferentes lugares e sob diferentes condições. Este tipo de repositório dá uma quantidade maior de variáveis que podem ser consideradas, aumentando o banco de dados para análise. Por exemplo, algumas falhas podem ocorrer em determinados ambientes de operação mas não em outros. Isso pode ser um indicativo de que melhorias no projeto podem ser feitos para desenvolver máquinas específicas para cada ambiente, aumento da confiabilidade e eficiência de cada equipamento.

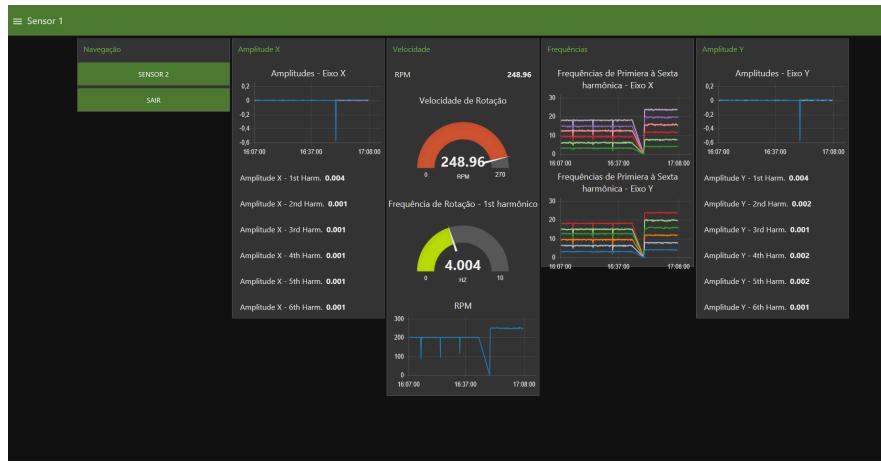


Figura 62 – Visualização da dados do Sensor Node 1

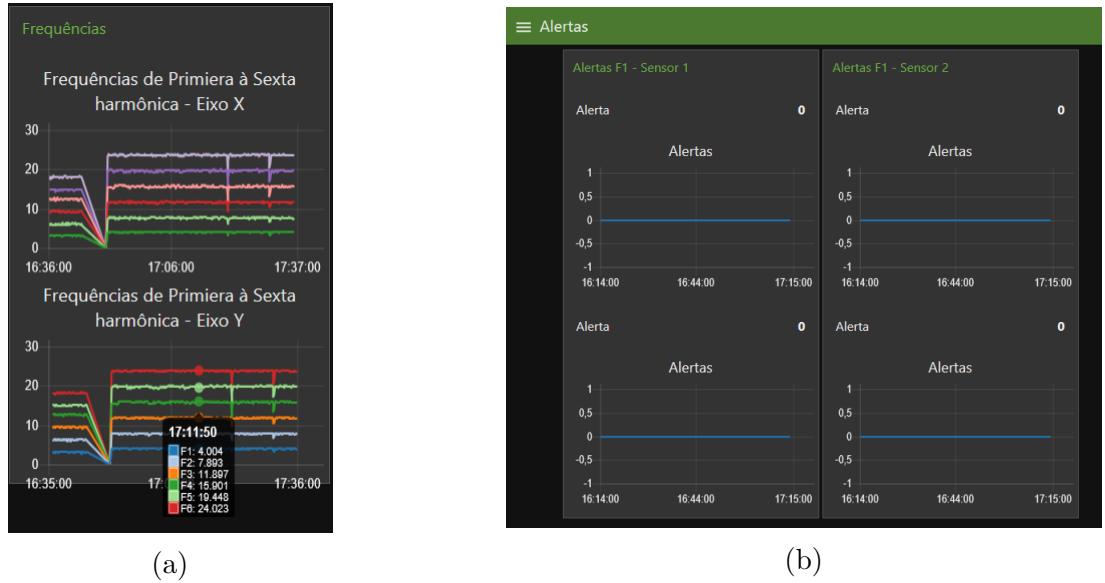


Figura 63 – Ambiente Node-RED: (a) Detalhe da visualização dos gráficos (b) Alertas criados.

8.6 Conclusões deste capítulo

Durante as análises dos dados, diversas técnicas foram empregadas, como waterfall, regressão logística, Linear SVM e RNA. Isso mostra que os dados podem e devem ser manipulados pelo usuário final utilizando a forma de análise que seja mais conveniente.

A conexão com a rede Wi-Fi foi realizada da forma rápida, e durante todos os testes o dispositivo permaneceu online, não sendo registrada nenhuma queda na conexão. Dependendo do ambiente de utilização do dispositivo, é necessário uma atenção a ruídos eletromagnéticos, conforme detalhado na seção 5.5.1.

O protótipo do dispositivo final permite que os dados adquiridos sejam enviados à nuvem sem complicaçāo. Apesar da escolha pelo IBM Cloud, ressalta-se que cabe ao usuário final escolher o servidor que melhor lhe atender, podendo inclusive utilizar um repositório próprio.

Criar as interfaces de visualização depende também das necessidades do usuário final. Entretanto, a interface elaborada apresenta as principais variáveis desejadas e necessárias para análise e/ou monitoramento das condições das máquinas rotativas.

Os testes experimentais mostraram que o elemento sensor escolhido, o acelerômetro MEMS triaxial MMA8451Q atendeu às especificações de projeto, fornecendo os dados de aceleração a uma taxa de amostragem superior à indicada pelo fabricante, atingindo 817 Hz.

O protótipo do dispositivo *Vibration Monitor* testado no motor BLDC atendeu todas as especificações de projeto.

Conclusão

Este trabalho apresentou um dispositivo de baixo custo para monitoramento de vibrações em máquinas rotativas utilizando um acelerômetro MEMS. Dispositivos comerciais similares foram analisados para a compreensão da solução e avaliação de suas limitações e vantagens. O dispositivo criado, nomeado como *Vibration Monitor*, foi projetado usando a concepção *open-hardware* e *open-software*. O projeto foi ainda estruturado para permitir uma fácil instalação em diversos tipos de máquinas rotativas.

O desenvolvimento do projeto levou à construção de dois protótipos. O segundo protótipo possui dimensão de 54 mm de diâmetro e possui as seguintes características:

- Custo de construção menor que U\$ 30,00;
- Taxa de amostragem do sinal de aceleração nos três eixos: 818 Hz;
- Pré-processamento da FFT do sinal de vibração a cada 1,25 s;
- Processa o sinal do sensor de velocidade;
- Consumo de potência médio: 600 mW
- Peso: 20 g; e
- Comunicação sem fio: Wi-Fi e BLE.

A partir dos dados que são enviados à nuvem, foi possível:

- Realizar o download para análise remota;
- Criar interfaces de visualização dos dados em tempo real e histórico; e
- Criar uma interface WEB de visualização de dados.

O protótipo pode ser facilmente encapsulado usando um molde projetado para cada aplicação específica e preenchido com Dimetil Polissiloxano (PMDS). O dispositivo não possui elementos que geram grande quantidade de calor, eliminando a necessidade de dissipadores térmicos e de ventilação forçada, simplificando seu encapsulamento. A instalação de um módulo de baterias possibilitaria a instalação do dispositivo de forma *stand-alone*.

Uma característica deste sistema é sua versatilidade com relação à análise. O usuário não está limitado a utilizar uma forma específica de análise, podendo escolher a que melhor lhe atenda. Os dados processados e enviados para o servidor se mostraram suficientes para diversos diagnósticos de distúrbios inseridos.

Conclui-se portanto que a principal contribuição deste trabalho é o projeto de um sistema para análise de vibração em máquinas rotativas, de baixo custo de construção (menor que U\$ 30,00), e que pode ser replicado a partir das informações aqui disponíveis.

Este sistema envia dados pré-processados à nuvem permitindo assim a comparação com dados de outras máquinas gêmeas que estejam distante geograficamente. Esta característica facilita o estudo do envelhecimento da máquina. Além disto, pode-se criar algoritmos que detectem a necessidade de manutenção preventiva, evitando quebras e reduzindo paradas desnecessárias.

Para finalizar, concluímos que este desenvolvimento tecnológico pode colaborar com a eficácia da manutenção e melhoria dos produtos de pequenas e médias empresas que utilizam ou fabricam máquinas rotativas para as mais variadas aplicações.

Publicações

Artigos Publicados

1. L. A. Dos Santos Pedotti, R. M. Zago and F. Fruett, "Fault diagnostics in rotary machines through spectral vibration analysis using low-cost MEMS devices,"in IEEE Instrumentation & Measurement Magazine, vol. 20, no. 6, pp. 39-44, December 2017.
2. L. A. S. Pedotti, R. M. Zago and F. Fruett, "Low-cost Wireless Vibration Analysis IoT System,"International Conference on Innovations in Engineering, Technology and Sciences (ICIETS 2018), Karnataka, India, 2018.
3. L. A. S. Pedotti, R. M. Zago and F. Fruett, "Application of low-cost MEMS accelerometer to measure unbalancing and load on a BLDC motor,"2017 2nd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT), Fortaleza, 2017, pp. 1-6.
4. L. A. S. Pedotti, R. M. Zago and F. Fruett, "Instrument based on MEMS accelerometer for vibration and unbalance analysis in rotating machines,"2016 1st International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT), Belo Horizonte, 2016, pp. 25-30.

Referências

- ABID, M. da Indústria Comércio e S. *Agenda brasileira para a Indústria 4.0.* 2018. Disponível em: <<http://www.industria40.gov.br/>>. Citado na página 59.
- ABNT-NBR:10082, A. B. D. N. T. Nbr 10082: Ensaio não destrutivo - análise de vibrações - avaliação da vibração mecânica de máquinas com velocidades de operação de 600 rpm a 15.000 rpm. 2011. Citado na página 34.
- ALBUQUERQUE, D. *O que é manutenção preditiva?* 2013. Disponível em: <<https://certificacaoiso.com.br/e-manutencao-preditiva-2/>. Acesso em: jun-19.> Citado na página 41.
- ALMEIDA, M. T. de; ALMEIDA, F. R. do V. *Módulo I – Princípios de Diagnósticos de Defeitos em Máquinas Rotativas.* Dissertação (Mestrado) — Fundação de Pesquisa e Assessoramento à Indústria - FUPAI, Itajubá, 2011. Citado na página 38.
- ANDREJASIC, M. *Seminar: MEMS Accelerometers.* [S.l.]: University of Ljubljana: Faculty for mathematics and physics, 2008. Citado na página 51.
- BARATIERI, C. L. *Controle de velocidade sensorless de motores brushless DC submetidos a variações periódicas de carga.* Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2011. Citado na página 79.
- BASUMALLICK, N.; BISWAS, P.; DASGUPTA, K.; BANDYOPADHYAY, S. Design optimization of fiber bragg grating accelerometer for maximum sensitivity. *Sensors and Actuators A: Physical*, v. 194, p. 31–39, 2013. Citado na página 55.
- BICAS, A. *MiniCloud – A Cloud gratuita OpenPOWER Linux da UNICAMP.* 2017. Disponível em: <<https://www.ibm.com/blogs/systems/br-pt/2017/01/16/minicloud-cloud-gratuita-openpower-linux-da-unicamp/>>. Citado na página 65.
- BILLAH, K. Y.; SCANLAN, R. H. Resonance, tacoma narrows bridge failure, and undergraduate physics textbooks. *American Journal of Physics*, v. 59, n. 2, p. 118–124, feb 1991. Citado na página 19.
- BLOCH, H. P.; GEITNER, F. K. *Machinery Failure Analysis and Troubleshooting.* third. Houston, Texas: Gulf Publishing Company, 1999. v. 2. Citado 5 vezes nas páginas , 32, 33, 34 e 37.
- BOON, G.; BAUW, K. D. A journey through 30 years of vibration analysis on large turbines: a history of progress in technology and experience. *Proceedings of ISMA2012-USD2012*, 2012. Citado na página 19.
- BOUSE, K.; BOWERS, S. V.; ROBINSON, J.; PIETY, R.; REEVES, T. *Machine condition indication system.* 2005. Disponível em: <<https://patents.google.com/patent/US20060181427A1/en?oq=US20060181427A1>>. Citado na página 29.

- BRAGA, D. *Como o Desalinhamento de Eixos Ocorre e Quais os Perigos Para o seu Equipamento.* 2019. Disponível em: <<https://dynamox.net/como-o-desalinhamento-de-eixos-ocorre-e-quais-os-perigos-para-o-seu-equipamento/>>. Citado 3 vezes nas páginas , 37 e 38.
- CASSIOLATO, C. *EMI - Interferência Eletromagnética em instalações industriais e muito mais.* 2018. Disponível em: <<http://www.smar.com/brasil/artigo-tecnico/emi-interferencia-eletromagnetica-em-instalacoes-industriais-e-muito-mais>>. Citado na página 63.
- CHEN, Z.; LI, Z. Research on fault diagnosis method of rotating machinery based on deep learning. In: *2017 Prognostics and System Health Management Conference (PHM-Harbin).* [S.l.: s.n.], 2017. p. 1–4. ISSN 2166-5656. Citado na página 25.
- CIRCUIBRAS. *Parametros Tecnicos Para o Desenvolvimento de Circuitos Impressos.* 2009. Disponível em: <http://www.eletrica.ufpr.br/mehl/te232/textos/CIRCUIBRAS_Manual_Tecnico.pdf>. Citado na página 68.
- CISCO. *20 Myths of Wi-Fi Interference: Dispelling Myths to Gain High-Performing and Reliable Wireless.* 2018. Disponível em: <<https://www.bradley.edu/dotAsset/887599c0-26bf-4be4-a5b9-3c0843b65d74.pdf>>. Citado na página 63.
- CONCEIÇÃO, W. N. E. da; COSTA, R. M. de R. Análise do protocolo mqtt para comunicação iot através de um cenário de comunicação. *Centro de Ensino Superior de Juiz de Fora (CESJF),* 2019. Citado na página 63.
- COSTA, E. da S. *Diagnóstico de Defeitos Mecânicos em Máquinas Rotativas Usando Lógica Fuzzy Baseada no Espectro Completo de Vibrações.* Dissertação (Dissertação (Mestrado em Engenharia Mecânica)) — Universidade Federal do Pará – UFPA, 2011. Citado na página 36.
- CRIATIVAS, D. S. *Quais segmentos gastam mais com manutenção de máquinas industriais?* 2018. Disponível em: <<https://www.industria40.ind.br/artigo/16472-quais-segmentos-gastam-mais-com-manutencao-de-maquinas-industriais>>. Citado na página 20.
- DEVELOPPEMENT, Y. *Inertial MEMS Manufacturing Trends 2014 Report by Yole Developpement.* 2014. Disponível em: <https://www.slideshare.net/Yole_Dev/yoletrendsample>. Citado 3 vezes nas páginas , 56 e 57.
- ELECTRONICS, I. *Eletrônica de diagnóstico para sensores de vibração: VSE002.* 2018. Disponível em: <<https://www.ifm.com/br/pt/product/VSE002>>. Citado na página 27.
- ELECTRONICS, I. *Sensor de aceleração: VSA101.* 2018. Disponível em: <<https://www.ifm.com/br/pt/product/VSA101>>. Citado na página 27.
- ENGEFAZ. *Inspeção por ultrassom acústico.* 2019. Disponível em: <<https://engefaz.com/manutencao-preditiva/inspecao-por-ultrassom>>. Citado na página 157.

- ESPRESSIF. *Schematic ESP32 module*. 2016. Disponível em: <https://dl.espressif.com/dl/schematics/ESP32-Core-Board-V2_sch.pdf>. Citado 2 vezes nas páginas 71 e 72.
- ESPRESSIF. *ESP32-WROOM-32 Datasheet*. 2017. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Citado na página 72.
- EUROPE, T. *braço de suporte de placa / de rolos esféricos / em ferro fundido / robusto*. 2019. Disponível em: <<http://www.directindustry.com/pt/prod/timken-europe/product-7626-1150319.html>>. Citado 2 vezes nas páginas 39.
- FLUKE. *Medidores de vibração e testadores de vibração portáteis Fluke 805*. 2018. Disponível em: <<https://www.fluke.com/pt-br/produto/manutencao-mecanica/analise-de-vibracoes/fluke-805>>. Citado na página 27.
- FOOTE, K. D. A brief history of the internet of things. *Data Education for Business and IT Professionals*, 2016. Disponível em: <http://www.dataversity.net/brief-history-internet-things/>. Acesso em: out-2018. Citado na página 61.
- FRANCESCHI, A. de; ANTONELLO, M. G. *Elementos de Máquinas*. [S.l.]: Biblioteca Cnetral da UFSM, 2014. Citado na página 38.
- GARTNER. *Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor*. 2015. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2015-08-18-gartners-2015-hype-cycle-for-emerging-technologies-identifies-the-computing-innovations-that-organizations-should-monitor>>. Citado na página 26.
- HALLIDAY, D.; RESNICK, R.; WALKER, J. *Fundamentos de Física: Mecânica*. nineth. [S.l.]: LTC, 2012. v. 1. Citado na página 36.
- HASELEY, R. K.; KIRKPATRICK, P. A. *Vibration monitoring system*. 1998. Disponível em: <<https://patents.google.com/patent/US5602757A/en?oq=US5602757A>>. Citado na página 30.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. second. [S.l.]: Prentice Hall, 1998. Citado na página 147.
- HAYKIN, S. *Neural Networks and Learning Machines*. third. [S.l.]: Prentice-Hall, 2008. Citado 2 vezes nas páginas 145 e 146.
- HELERBROCK, R. *O que é ultrassom?; Brasil Escola*. 2019. Disponível em: <<https://brasilescola.uol.com.br/o-que-e/fisica/o-que-e-ultrassom.htm>>. Citado na página 157.
- HUNT, S. R. *Data acquisition unit for remote monitoring system and method for remote monitoring*. 2000. Disponível em: <<https://patents.google.com/patent/US6556956B1/en?oq=D1+-+US6556956B1>>. Citado na página 29.
- IBM. *IBM Cloud Lite: Work with the best of IBM Cloud for an unlimited amount of time*. 2018. Disponível em: <https://www.ibm.com/cloud/lite-account?mhq=ibm%20cloud%20lite&mhsrc=ibmsearch_a>. Citado na página 66.

INFOSOLDA. *Ensaio não-destrutivo: Ultrassom.* 2019. Disponível em: <<https://www.infosolda.com.br/biblioteca-digital/livros-senai/ensaios-nao-destrutivos-e-mecanicos-livros-senai/214-ensaio-nao-destrutivo-ultrassom>>. Citado na página 158.

INSTRUMENTS, O. *HS-620 & HS-630 Vibration Meter Kits.* 2018. Disponível em: <<http://www.omniinstruments.co.uk/vibration-monitoring/portable-vibration-meters/hs-620-hs-630-vibration-meter-kits.html>>. Citado na página 27.

INSTRUMENTS, O. *VM-6360 Hand Held Vibration Meter.* 2018. Disponível em: <<http://www.omniinstruments.co.uk/vibration-monitoring/portable-vibration-meters/vm-6360-hand-held-vibration-meter-839.html#upsell-product-div>>. Citado na página 27.

INSTRUMENTS, O. *VM220 Portable Vibration Meter.* 2018. Disponível em: <<http://www.omniinstruments.co.uk/vibration-monitoring/portable-vibration-meters/vm220-portable-vibration-meter.html>>. Citado na página 27.

ISO10816-1:1995. Mechanical vibration - evaluation of machine vibration by measurements on non-rotating parts - part 1: General guidelines. 1995. Citado na página 34.

ISO10816-3:2009. Mechanical vibration — evaluation of machine vibration by measurements on non-rotating parts — part 3: Industrial machines with nominal power above 15 kw and nominal speeds between 120 r/min and 15 000 r/min when measured in situ. 2009. Citado 2 vezes nas páginas e 35.

J.A.POMILIO; S.M.DECKMANN. *Avaliação da Qualidade da Energia Elétrica.* 2019. Disponível em: <<http://www.dsce.fee.unicamp.br/~antenor/pdffiles/qualidade/b4.pdf>>. Citado na página 49.

KARNGA, A. E. Neural networks. In: *IJCNN.99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339).* Washington, DC, USA: [s.n.], 1999. v. 6, p. 4419–4421. Citado 2 vezes nas páginas 145 e 146.

KIM, J.; LEE, J.; LEE, C.; PARK, E.; KIM, J.; KIM, H.; LEE, J.; JEONG, H. Optimal feature selection for pedestrian detection based on logistic regression analysis. In: *2013 IEEE International Conference on Systems, Man, and Cybernetics.* [S.l.: s.n.], 2013. p. 239–242. ISSN 1062-922X. Citado na página 145.

KRAS, E. *CONHEÇA A IMPORTÂNCIA DO ULTRASSOM MANUTENÇÃO PREDITIVA PARA O ANDAMENTO DE SUA EMPRESA.* 2019. Disponível em: <<https://www.krasinspecao.com.br/ultrassom-manutencao-preditiva>>. Citado na página 157.

KURZWEIL, R. *The Singularity Is Near: When Humans Transcend Biology.* [S.l.]: Viking Prees, 2006. Citado 2 vezes nas páginas e 57.

LALLY, J. *ACCELEROMETER SELECTION CONSIDERATIONS: Charge and ICP Integrated Circuit Piezoelectric.* 2019. Disponível em: <https://www.pcb.com/techsupport/docs/vib/TN_17_VIB-0805.pdf>. Citado na página 51.

- LIU, S.; LIANG, Y.; BROOKS, M. Eucalyptus: A web service-enabled e-infrastructure. *17th Annual International Conference on Computer Science and Software Engineering (CASCON 2007)*, oct 2007. Citado na página 65.
- LIVRE, M. *Medidores de vibração e testadores de vibração portáteis Fluke 805*. 2018. Disponível em: <https://produto.mercadolivre.com.br/MLB-835208314-medidor-de-vibraco-de-motor-fluke-805-_JM>. Citado na página 27.
- MADEIRA, D. *IIoT - Industrial Internet of Things*. 2016. Disponível em: <<https://www.embarcados.com.br/iiot-industrial-internet-of-things/>>. Citado na página 60.
- MAHANTI, A.; CARLSSON, N.; WILLIAMSON, C.; ARLITT, M. Ambient interference effects in wi-fi networks. *NETWORKING 2010: 9th International IFIP TC 6 Networking Conference*, p. 160–173, may 2010. Citado na página 62.
- MAIS, J. *Spectrum Analysis: The key features of analyzing spectra*. 2019. Disponível em: <<https://www.skf.com/binary/tcm:12-113997/CM5118%20EN%20Spectrum%20Analysis.pdf>>. Citado 3 vezes nas páginas , 33 e 34.
- MARCAL, R. F. M.; NEGREIROS, M.; SUSIN, A. A.; KOVALESKI, J. L. Detecting faults in rotating machines. In: . [S.l.: s.n.], 2000. p. 24–26. Citado na página 24.
- MEDIÇÃO de vibração com acelerômetros. 2019. Disponível em: <<http://www.ni.com/pt-br/innovations/white-papers/06/measuring-vibration-with-accelerometers.html>>. Citado 3 vezes nas páginas , 51 e 53.
- MO, Y.; SU, K.; KANG, W.; CHEN, L.; CHANG, W.; LIU, Y. An fft-based high-speed spindle monitoring system for analyzing vibrations. In: *2017 Eleventh International Conference on Sensing Technology (ICST)*. [S.l.: s.n.], 2017. p. 1–4. ISSN 2156-8073. Citado na página 24.
- MOHANTY, S.; GUPTA, K. K.; RAJU, K. S.; MISHRA, V.; KUMAR, V.; PRASAD, P. B. Characterization of wireless accelerometer sensor and its industrial applications. In: *2014 Twentieth National Conference on Communications (NCC)*. [S.l.: s.n.], 2014. p. 1–5. Citado na página 25.
- MOORE, G. E. Cramming more components onto integrated circuits. *IEEE Solid-State Circuits Society Newsletter*, v. 38, n. 8, p. 114, apr 1965. Citado na página 56.
- MORAES, A. F. de. *Redes sem fio: Instalação, configuração e segurança*. first. São Paulo: Érica, 2010. Citado na página 62.
- MOTORES, B. *Queima de Motores Elétricos: Motores Trifásicos*. 2019. Disponível em: <<http://bezerramotores.com.br/queimas-de-motores-eletricos/causas-de-queima-de-motores-eletricos/>>. Citado 2 vezes nas páginas 39 e 40.
- NAIR, V.; HINTON, G. E. *Rectified linear units improve restricted boltzmann machines*. [S.l.]: ICML, 2010. Citado na página 147.

- NAKAJIMA, S. *Introduction to TPM: Total Productive Maintenance (Preventative Maintenance Series)*. [S.l.]: Eleventh Printing edition, 1988. ISBN 978-0915299232. Citado na página 40.
- Nivesrangsan, P.; Jantarajirojkul, D. Bearing fault monitoring by comparison with main bearing frequency components using vibration signal. In: *2018 5th International Conference on Business and Industrial Research (ICBIR)*. [S.l.: s.n.], 2018. p. 292–296. Citado na página 38.
- NORTON, R. L. *Machine Design*. fourth. [S.l.]: Prentice-Hall, 2010. Citado na página 38.
- OMEGA. *Medidor de Vibração? Monitoramento de Maquinário para Definir a Necessidade de Manutenção*. 2018. Disponível em: <https://br.omega.com/googlebase/product.html?pn=HHVB82&gclid=CjwKCAjwiPbWBRBtEiwAJakcpArcBAixshs8op_3GGuRKJZS2z_inNTPVgoUkk3QaxZmkb4dUV1c9BoC0rEQAvD_BwE>. Citado na página 28.
- ONEPROD. *FALCON: Smart Portable Solution*. 2018. Disponível em: <<https://vibralign.com/wp-content/uploads/2015/03/falcon-technical-datasheet-en.pdf>>. Citado na página 28.
- OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing*. third. [S.l.]: Pearson, 2009. Citado 3 vezes nas páginas 48, 49 e 50.
- PATIL, S. S.; GAIKWAD, J. A. Vibration analysis of electrical rotating machines using fft: A method of predictive maintenance. In: *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2013. p. 1–6. Citado na página 25.
- PETROBRÁS. *Curso de Vibrações*. [S.l.], 2000. Citado 2 vezes nas páginas 38 e 40.
- PHAN, K.; MAURITZ, A.; HOMBURG, F. A novel elastomer-based magnetoresistive accelerometer. *Sensors and Actuators A*, v. 145-145, p. 109–115, 2008. Citado na página 54.
- PHAN, T. *5 Things you should know about MQTT*. 2016. Disponível em: <<https://community.iotone.com/t/5-things-you-should-know-about-mqtt/55>>. Citado 2 vezes nas páginas 54 e 60.
- PIESOTRONICS, P. *Introduction to MEMS Accelerometers*. 2018. Disponível em: <<http://www.pcb.com/Resources/Technical-Information/mems-accelerometers>>. Citado na página 55.
- PLETS, D.; TANGHE, E.; PAEPENS, A.; MARTEENS, L.; JOSEPH, W. Wifi network planning and intra-network interference issues in large industrial warehouses. In: *2016 10th European Conference on Antennas and Propagation (EuCAP)*. [S.l.: s.n.], 2016. p. 1–5. Citado na página 63.
- POPA, R.; IAMURRI, J.; SIDRONIO, Y.; FLETCHER, A.; SIGLER, C. *Digital Accelerometers and Magnetometers*. 2019. Disponível em: <<https://www.ece.msu.edu/classes/ece480/capstone/spring12/group03/Documents/TechnicalLecturePresentation.pdf>>. Citado na página 54.

- QUIST, N. L.; BONNETT, A. H.; LYNCH, J. P.; KLINE, J. A.; HENDERSON, S. I.; HANNULA, R. I.; GRUDKOWSKI, J. W.; DIVLJAKOVIC, V.; BUCKLEY, G. W.; BAUER, R. P.; ALGUINDIGUE, I. E. *Distributed diagnostic system*. 1998. Disponível em: <<https://patents.google.com/patent/US6199018B1/en?oq=US6199018B1>>. Citado na página 30.
- RAO, S. *Vibrações Mecânicas*. fourth. [S.l.]: Pearson, 2014. Citado 9 vezes nas páginas 19, 32, 33, 34, 36, 40, 43, 46 e 47.
- ROJAS, R. *Neural Networks - A Systematic Introduction*. Berlin, New-York: Springer-Verlag, 1996. Citado na página 147.
- SABATO, A.; NIEZRECKI, C.; FORTINO, G. Wireless mems-based accelerometer sensor boards for structural vibration monitoring: A review. *IEEE Sensors Journal*, v. 17, n. 2, p. 226–235, Jan 2017. ISSN 1530-437X. Citado na página 25.
- SADIQU, M. N. O. *Elements of Electromagnetism*. fourth. New York: Oxford University Press, 2007. Citado na página 82.
- SANTOS, B. P.; SILVA, L. A. M.; CELES, C. S. F. S.; NETO, J. B. B.; PERES, B. S.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. A. F. Internet das coisas: da teoria à prática. *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016)*, 2016. Citado 5 vezes nas páginas , 25, 27, 59 e 61.
- SCHRÖDER, F. C.; LUCCA, Y. F. L. D.; DALFRÉ.FILHO, J. G. Bearing temperature effect analysis of centrifuge pumps operating with moderate cavitation. *American Journal of Hydropower, Water and Environment Systems*, aug 2015. Citado na página 89.
- SEMICONDUCTORS, N. *MMA8451Q, 3-axis, 14-bit/8-bit digital accelerometer*. 2017. Disponível em: <<https://www.nxp.com/docs/en/data-sheet/MMA8451Q.pdf>>. Citado 4 vezes nas páginas , 56, 58 e 71.
- SILVA, R. D. da. *IDENTIFICAÇÃO DE PARÂMETROS E PROJETO DO SISTEMA DE CONTROLE DE VELOCIDADE DE MOTORES DE CORRENTE CONTÍNUA SEM ESCOVAS BLDC*. Dissertação (Trabalho de Conclusão de Curso) — Universidade Estadual Paulista Júlio de Mesquita Filho, 2017. Citado na página 80.
- SINHA, S.; SHAKYA, S.; MUKHIYA, R.; GOPAL, R.; PANT, B. Design and simulation of mems differential capacitive accelerometer. *ISSS International Conference on Smart Materials, Structures and Systems*, jul 2014. Citado 3 vezes nas páginas , 55 e 56.
- SOARES, P. da S. *SISTEMA DE AVALIAÇÃO PREDITIVA DE FALHAS EM MÁQUINAS ELÉTRICAS USANDO LÓGICA FUZZY COM ANÁLISE DOS PARÂMETROS DE VIBRAÇÃO, CORRENTE E TEMPERATURA*. Dissertação (Tese (Doutorado)) — UNICAMP - Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação (FEEC), 2014. Citado 2 vezes nas páginas 37 e 40.

- SOUZA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. Capítulo 7: Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *III Escola Regional de Computação Ceará, Maranhão e Piauí ERCEMAPI*, p. 150–175, 2009. Citado 2 vezes nas páginas 64 e 65.
- STEEGE, P. Overall equipment effectiveness in resist processing equipment. In: *Advanced Semiconductor Manufacturing Conference and Workshop, 1996. ASMC 96 Proceedings. IEEE/SEMI 1996*. [S.l.: s.n.], 1996. p. 76–79. ISSN 1078-8743. Citado na página 40.
- SULAIMA, M. F.; OTHMAN, M. N.; JALI, M. H.; NOR, M. K.; SHAMSUDIN, N. H. Engineering ethics case study of hydroelectric sayanosheshenkaya accident: Lesson learned for malaysian engineer. *Journal of Engineering Technology*, v. 2, p. 64–74, jan 2014. Citado na página 19.
- TANG, J.; DENG, C.; HUANG, G. Extreme learning machine for multilayer perceptron. In: *IEEE Transactions on Neural Networks and Learning Systems*. [S.l.: s.n.], 2016. (4, v. 27), p. 809–821. Citado na página 145.
- TDK. *MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS Motion Tracking™ Devices*. 2018. Disponível em: <<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>>. Citado na página 55.
- TELES, J. *Curva PF: O que é e como usar*. 2019. Disponível em: <<https://engeteles.com.br/curva-pf/>>. Citado 2 vezes nas páginas e 42.
- TELES, J. *Manutenção Preditiva: O que é e como ela pode te ajudar!* 2019. Disponível em: <<https://engeteles.com.br/manutencao-preditiva/>>. Citado na página 41.
- TRANTER, J. Resonance the silent killer of rotating machinery. *MaintWorld Magazine for maintenance and asset management professionals*, aug 2015. Citado na página 38.
- TU, N. N. *Agronegócio: a aplicação da indústria 4.0 na melhora da produtividade e rendimento*. 2017. Disponível em: <<https://paineira.usp.br/aun/index.php/2017/06/29/agronegocio-a-aplicacao-da-industria-4-0-na-melhora-da-produtividade-e-rendimento/>>. Citado na página 59.
- TZENG, C. Vibration detection and analysis of wind turbine based on a wireless embedded microcontroller system. In: *2018 IEEE International Conference on Applied System Invention (ICASI)*. [S.l.: s.n.], 2018. p. 133–136. Citado na página 24.
- UPADHYE, M. Y.; BOROLE, P. B.; SHARMA, A. K. Real-time wireless vibration monitoring system using labview. In: *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. [S.l.: s.n.], 2015. p. 925–928. Citado na página 24.
- VIDAL, V. *10 protocolos de IoT que você deveria conhecer*. 2017. Disponível em: <<https://www.profissionaisti.com.br/2017/11/10-protocolos-de-iot-que-voce-deveria-conhecer/>>. Citado na página 25.

- WALMART. *Medidor de Vibração Digital 3 Eixos Deslocamento Velocidade Aceleração Tecnoferramentas Vm-6380*. 2018. Disponível em: <<https://www.walmart.com.br/medidor-de-vibracao-digital-3-eixos-deslocamento-velocidade-aceleracao-tecnoferramentas-vm-6380/5601179/pr>>. Citado na página 28.
- WIKI, E. T. H. *MEMS*. 2017. Disponível em: <<https://ethw.org/MEMS>>. Citado na página 55.
- WOWK, V. *Machinery Vibration*. [S.l.]: McGraw-Hill, 1991. Citado 2 vezes nas páginas 32 e 33.
- YEDAMALE, P. *Brushless DC (BLDC) Motor Fundamentals*. Microchip Technology Inc. AN885. 2003. Disponível em: <ww1.microchip.com/downloads/en/appnotes/00885a.pdf>. Citado na página 79.
- YOSHIE, O.; SATO, N.; FUKUNAGA, T. *System for diagnosing facility apparatus, managing apparatus and diagnostic apparatus*. 2004. Disponível em: <<https://patents.google.com/patent/US20060129358A1/en?oq=US20060129358A1>>. Citado na página 30.
- YUAN, M. *Conhecendo o MQTT*. 2017. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Citado na página 63.

Apêndices

APÊNDICE A – Descritivo operacional do dispositivo

A.1 Pinagem

As pinagens do dispositivo estão descritas a seguir:

- P1: Este é o pino tipo Jack (Figura 64a), que serve para alimentação do dispositivo através de uma fonte externa de 9 V a 12 V.
- P2: Este pino serve para selecionar qual será a fonte principal de alimentação do dispositivo. Deve ser utilizado um mini jumper 2,54 mm de dois pinos para realizar esta seleção, similar ao da Figura 64b. Caso queira-se energizar a placa utilizando uma fonte ligada ao pino P1, interligar os pinos 1 e 2 de P2 utilizando o jumper. Caso queira-se energizar a placa utilizando uma porta serial (através do Pino P3), interligar os pinos 3 e 4 de P2 utilizando o jumper.

*Obs: NUNCA interligue os pinos 1&2 e 3&4 simultaneamente. Por isso a importância de haver somente 1 jumper de dois pinos na placa.



Figura 64 – Elementos do dispositivo: (a) Pino tipo Jack (b) Mini-jumper para 2 pinos

- P3: Este pino permite a comunicação com um computador externo através de uma porta serial. Sugere-se a utilização de um conversor USB Serial TTL de 5 V (adaptador UART) para esta operação. Ligar os pinos do conversor exatamente conforme escritos na placa do dispositivo, conforme o esquema da Figura 65 e da Tabela 12.

*Obs: O cruzamento RX-TX já está feito na placa, portanto deve-se ligar exatamente como mostrado na Figura 65.

Adaptador UART	Dispositivo: Vibration Monitor
GND	GND
VCC / 5V	5V
RX	RX
TX	TX

Tabela 12 – Pinagem entre o Vibration Monitor e Adaptador UART

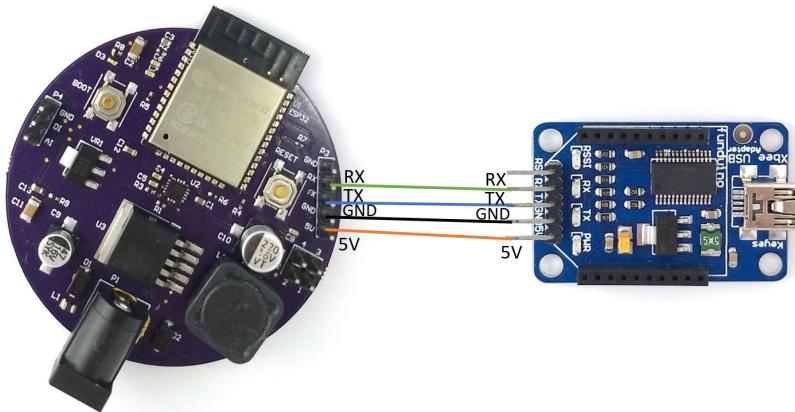


Figura 65 – Esquema de ligação do dispositivo Vibration Monitor com adaptador UART

- P4: Pino destinado à ligação de Entrada Digital e Entrada Analógica de um dispositivo externo, como um *reed-switch* por exemplo, no ESP32, conforme apresentado na Tabela 13

Pin P4	Pin ESP32
GND	GND
DI	IO17 / 28
AI	IO4 / 26

Tabela 13 – Pinagem entre a barra de pinos P4 e o ESP32

A.2 Instalação do Hardware do ESP32 no IDE do Arduino

Esta seção apresenta um passo a passo de instalação do ESP32 na IDE do Arduino.

Passo 1: Instalar o conversor USB-Serial do ESP32 para o Sistema Operacional desejado. O Driver está disponível em: <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>.

Passo 2: Com a IDE do Arduino já instalada em seu computador, clique em Arquivo -> Preferências.

Passo 3: Abrirá uma janela com as configurações. No Final da tela há um *textbox* com o seguinte *label*: URLs adicionais para Gerenciadores de Placas: Clique no botão na frente do *textbox*.

Passo 4: Abrirá uma tela para que seja inserido a URL adicional. Digite: `<https://dl.espressif.com/dl/package_esp32_index.json>` e clique em Ok.

Passo 5: Volte na tela principal da IDE do Arduino. Clique em Ferramentas -> Placa 'Arduino/Genuino Uno' -> Gerenciador de Placas

Passo 6: Faça uma pesquisa por **esp32**

Passo 7: Aparecerá uma opção de instalação **esp32 by Espressif Systems**. Clique em Instalar.

Passo 8: Quando for pesquisar novamente em Ferramentas -> Placa: 'xxxx' -> Aparecerá na listagem as opções relativas ao Esp32. Selecione a placa desejada e já é possível utilizar a IDE do Arduino para desenvolver seu código mais facilmente. FIM.

APÊNDICE B – Código Fonte do Dispositivo

Código para cálculo da FFT (myFFT.cpp)

```

1 #include <pgmspace.h>
2 #include "myFFT.h"
3 #include <Arduino.h>
4 #define pi 3.141592654
5 void myFFT(double* nXr, double* nXi, double real[], double imag[],
6           int N, int h, int sam_gap){
7     /* Variables:
8      nXr: real Freq
9      nXi: imagined Freq
10     r[]: real signal
11     i[]: imagined signal
12     N: # of samples
13     h: signal starting point
14     sam_gap: signal sampling gap */
15     if (N == 1){
16         nXr[0] = real[h];
17         nXi[0] = imag[h];
18         return;
19     }else{
20         double nXr1[N/2], nXi1[N/2], nXr2[N/2], nXi2[N/2];
21         // Interval recursion
22         myFFT(nXr1, nXi1, real, imag, N/2, h, sam_gap*2);
23         myFFT(nXr2, nXi2, real, imag, N/2, h+sam_gap, sam_gap*2);
24         for (int i = 0; i < (N/2); i++){
25             // t <- X_k
26             double temp_re = nXr1[i];
27             double temp_im = nXi1[i];
28             double temp_re_iN2 = nXr2[i];
29             double temp_im_iN2 = nXi2[i];
30             // X_k <- t + exp(-2*pi*i*k/N) X_(k+N/2)
31             nXr[i] = temp_re + cos(-2*pi*i/N)*temp_re_iN2 - sin(-2*
32                           pi*i/N)*temp_im_iN2;

```

```

31         nXi[i] = temp_im + cos(-2*pi*i/N)*temp_im_iN2 + sin(-2*
32             pi*i/N)*temp_re_iN2;
33             // X_(k+N/2) <- t - exp(-2*pi*i*k/N) X_(k+N/2)
34             nXr[i+N/2] = temp_re - (cos(-2*pi*i/N)*temp_re_iN2 - sin
35                 (-2*pi*i/N)*temp_im_iN2);
36             nXi[i+N/2] = temp_im - (cos(-2*pi*i/N)*temp_im_iN2 + sin
37                 (-2*pi*i/N)*temp_re_iN2);
38     }
39 }
40
41 void myFFT2(double* nXr, double* nXi, int fst, double real[], int N,
42     int h, int sam_gap){
43     /* Variables:
44         nXr: real Freq
45         nXi: imagined Freq
46         fst: Freq starting point
47         r[]: real signal
48         i[]: imagined signal
49         N: # of samples
50         h: signal starting point
51         sam_gap: signal sampling gap      */
52     if (N == 1){
53         nXr[fst] = real[h];
54         nXi[fst] = 0;
55         return;
56     }else{
57         // double nXr1[N/2], nXi1[N/2], nXr2[N/2], nXi2[N/2];
58         // Interval recursion
59         myFFT2(nXr, nXi, fst, real, N/2, h, sam_gap*2);
60         myFFT2(nXr, nXi, fst+N/2, real, N/2, h+sam_gap, sam_gap*2);
61         for (int i = 0; i < (N/2); i++){
62             // t <- X_k
63             double temp_re = nXr[i+fst];
64             double temp_im = nXi[i+fst];
65             double temp_re_iN2 = nXr[i+fst+N/2];
66             double temp_im_iN2 = nXi[i+fst+N/2];
67             // Calculation temp
68             double temp1 = cos(-2*pi*i/N)*temp_re_iN2 - sin(-2*pi*i/
69                 N)*temp_im_iN2;

```

```

65     double temp2 = cos(-2*pi*i/N)*temp_im_iN2 + sin(-2*pi*i/
66     N)*temp_re_iN2;
67     // X_k <- t + exp(-2*pi*i*k/N) X_(k+N/2)
68     nXr[i+fst] = temp_re + temp1;
69     nXi[i+fst] = temp_im + temp2;
70     // X_(k+N/2) <- t - exp(-2*pi*i*k/N) X_(k+N/2)
71     nXr[i+fst+N/2] = temp_re - temp1;
72     nXi[i+fst+N/2] = temp_im - temp2;
73 }
74 }
```

Código para cálculo da FFT (myFFT.h) - Biblioteca

```

1 // FFT Cooley-Turkey with array pointer
2 void myFFT(double* nXr, double* nXi, double real[], double imag[],
3             int N, int h, int sam_gap);
4 // FFT Cooley-Turkey w/o creating new arrays
5 void myFFT2(double* nXr, double* nXi, int fst, double real[], int N,
6             int h, int sam_gap);
```

Código para leitura da comunicação I²C - Biblioteca (wire.cpp)

```

1 /* TwoWire.cpp - TWI/I2C library for Arduino & Wiring Copyright (c)
2 2006 Nicholas Zambetti. All right reserved. This library is free
3 software; you can redistribute it and/or modify it under the
4 terms of the GNU Lesser General Public License as published by
5 the Free Software Foundation; either version 2.1 of the License,
6 or (at your option) any later version. This library is
7 distributed in the hope that it will be useful, but WITHOUT ANY
8 WARRANTY; without even the implied warranty of MERCHANTABILITY or
9 FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General
10 Public License for more details. You should have received a copy
11 of the GNU Lesser General Public License along with this library;
12 if not, write to the Free Software Foundation, Inc., 51 Franklin
13 St, Fifth Floor, Boston, MA 02110-1301 USA Modified 2012 by
14 Todd Krein (todd@krein.org) to implement repeated starts Modified
15 December 2014 by Ivan Grokhotkov (ivan@esp8266.com) - esp8266
```

```
support Modified April 2015 by Hrsto Gochkov (ficeto@ficeto.com)
- alternative esp8266 support*/

2 extern "C" {
3 #include <stdlib.h>
4 #include <string.h>
5 #include <inttypes.h>}
6 #include "esp32-hal-i2c.h"
7 #include "Wire.h"
8 #include "Arduino.h"

9 TwoWire::TwoWire(uint8_t bus_num)
10      :num(bus_num & 1)
11      ,sda(-1)
12      ,scl(-1)
13      ,i2c(NULL)
14      ,rxIndex(0)
15      ,rxLength(0)
16      ,txIndex(0)
17      ,txLength(0)
18      ,txAddress(0)
19      ,transmitting(0)
20 {}
21 void TwoWire::begin(int sdaPin, int sclPin, uint32_t frequency)
22 {
23     if(sdaPin < 0) {
24         if(num == 0) {
25             sdaPin = SDA;
26         } else {
27             return;
28         }
29     }
30     if(sclPin < 0) {
31         if(num == 0) {
32             sclPin = SCL;
33         } else {
34             return;
35         }
36     }
37     if(i2c == NULL) {
38         i2c = i2cInit(num, 0, false);
```

```
39         if(i2c == NULL) {
40             return;
41         }
42     }
43     i2cSetFrequency(i2c, frequency);
44
45     if(sda >= 0 && sda != sdaPin) {
46         i2cDetachSDA(i2c, sda);
47     }
48     if(scl >= 0 && scl != sclPin) {
49         i2cDetachSCL(i2c, scl);
50     }
51     sda = sdaPin;
52     scl = sclPin;
53     i2cAttachSDA(i2c, sda);
54     i2cAttachSCL(i2c, scl);
55     flush();
56     i2cInitFix(i2c);
57 }
58 void TwoWire::setClock(uint32_t frequency)
59 {
60     i2cSetFrequency(i2c, frequency);
61 }
62 size_t TwoWire::requestFrom(uint8_t address, size_t size, bool
63     sendStop)
64 {
65     if(size > I2C_BUFFER_LENGTH) {
66         size = I2C_BUFFER_LENGTH;
67     }
68     size_t read = (i2cRead(i2c, address, false, rxBuffer, size,
69         sendStop) == 0)?size:0;
70     rxIndex = 0;
71     rxLength = read;
72     return read;
73 }
74 uint8_t TwoWire::endTransmission(uint8_t sendStop)
75 {
76     int8_t ret = i2cWrite(i2c, txAddress, false, txBuffer, txLength,
77         sendStop);
```

```
75     txIndex = 0;
76     txLength = 0;
77     transmitting = 0;
78     return ret;
79 }
80 uint8_t TwoWire::requestFrom(uint8_t address, uint8_t quantity,
81     uint8_t sendStop)
82 {
83     return requestFrom(address, static_cast<size_t>(quantity),
84         static_cast<bool>(sendStop));
85 }
86 uint8_t TwoWire::requestFrom(uint8_t address, uint8_t quantity)
87 {
88     return requestFrom(address, static_cast<size_t>(quantity), true)
89         ;
90 }
91 uint8_t TwoWire::requestFrom(int address, int quantity)
92 {
93     return requestFrom(static_cast<uint8_t>(address), static_cast<
94         size_t>(quantity), true);
95 }
96 void TwoWire::beginTransmission(uint8_t address)
97 {
98     transmitting = 1;
99     txAddress = address;
100    txIndex = 0;
101    txLength = 0;
102 }
103 void TwoWire::beginTransmission(int address)
104 {
105     beginTransmission((uint8_t)address);
106 }
107 uint8_t TwoWire::endTransmission(void)
```

```
108 {
109     return endTransmission(true);
110 }
111 size_t TwoWire::write(uint8_t data)
112 {
113     if(transmitting) {
114         if(txLength >= I2C_BUFFER_LENGTH) {
115             return 0;
116         }
117         txBuffer[txIndex] = data;
118         ++txIndex;
119         txLength = txIndex;
120     }
121     return 1;
122 }
123 size_t TwoWire::write(const uint8_t *data, size_t quantity)
124 {
125     if(transmitting) {
126         for(size_t i = 0; i < quantity; ++i) {
127             if(!write(data[i])) {
128                 return i;
129             }
130         }
131     }
132     return quantity;
133 }
134 int TwoWire::available(void)
135 {
136     int result = rxLength - rxIndex;
137     return result;
138 }
139 int TwoWire::read(void)
140 {
141     int value = -1;
142     if(rxIndex < rxLength) {
143         value = rxBuffer[rxIndex];
144         ++rxIndex;
145     }
146     return value;
```

```
147 }
148 int TwoWire::peek(void)
149 {
150     int value = -1;
151     if(rxIndex < rxLength) {
152         value = rxBuffer[rxIndex];
153     }
154     return value;
155 }
156 void TwoWire::flush(void)
157 {
158     rxIndex = 0;
159     rxLength = 0;
160     txIndex = 0;
161     txLength = 0;
162 }
163 TwoWire Wire = TwoWire(0);
```

Biblioteca (wire.h)

```
1 /*TwoWire.h - TWI/I2C library for Arduino & Wiring Copyright (c)
   2006 Nicholas Zambetti. All right reserved. This library is
   free software; you can redistribute it and/or modify it under
   the terms of the GNU Lesser General Public License as published
   by the Free Software Foundation; either version 2.1 of the
   License, or (at your option) any later version. This library is
   distributed in the hope that it will be useful, but WITHOUT ANY
   WARRANTY; without even the implied warranty of MERCHANTABILITY or
   FITNESS FOR A PARTICULAR PURPOSE. See the GNU
   Lesser General Public License for more details. You should have
   received a copy of the GNU Lesser General Public License along
   with this library; if not, write to the Free Software
   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
   02110-1301 USA Modified 2012 by Todd Krein (todd@krein.org) to
   implement repeated starts Modified December 2014 by Ivan
   Grokhotkov (ivan@esp8266.com) - esp8266 support Modified April
   2015 by Hrsto Gochkov (ficeto@ficeto.com) - alternative esp8266
   support*/
4 #ifndef TwoWire_h
```

```
5 #define TwoWire_h
6 #include <esp32-hal.h>
7 #include "freertos/FreeRTOS.h"
8 #include "freertos/queue.h"
9 #include "Stream.h"
10 #define I2C_BUFFER_LENGTH 128
11 class TwoWire: public Stream
12 {
13 protected:
14     uint8_t num;
15     int8_t sda;
16     int8_t scl;
17     i2c_t * i2c;
18     uint8_t rxBuffer[I2C_BUFFER_LENGTH];
19     uint16_t rxIndex;
20     uint16_t rxLength;
21     uint8_t txBuffer[I2C_BUFFER_LENGTH];
22     uint16_t txIndex;
23     uint16_t txLength;
24     uint8_t txAddress;
25     uint8_t transmitting;
26 public:
27     TwoWire(uint8_t bus_num);
28     void begin(int sda=-1, int scl=-1, uint32_t frequency=400000);
29     void setClock(uint32_t);
30     void beginTransmission(uint8_t);
31     void beginTransmission(int);
32     uint8_t endTransmission(void);
33     uint8_t endTransmission(uint8_t);
34     size_t requestFrom(uint8_t address, size_t size, bool sendStop);
35     uint8_t requestFrom(uint8_t, uint8_t);
36     uint8_t requestFrom(uint8_t, uint8_t, uint8_t);
37     uint8_t requestFrom(int, int);
38     uint8_t requestFrom(int, int, int);
39     size_t write(uint8_t);
40     size_t write(const uint8_t *, size_t);
41     int available(void);
42     int read(void);
43     int peek(void);
```

```

44     void flush(void);
45     inline size_t write(const char * s)
46     {
47         return write((uint8_t*) s, strlen(s));
48     }
49     inline size_t write(unsigned long n)
50     {
51         return write((uint8_t)n);
52     }
53     inline size_t write(long n)
54     {
55         return write((uint8_t)n);
56     }
57     inline size_t write(unsigned int n)
58     {
59         return write((uint8_t)n);
60     }
61     inline size_t write(int n)
62     {
63         return write((uint8_t)n);
64     }
65 };
66 extern TwoWire Wire;
67 #endif

```

Código principal (main.ino)

```

1 #include "Wire.h"
2 #include "myFFT.h"
3 #include <WiFi.h>
4 #include <PubSubClient.h> //Lib de MQTT
5 //----- Dados de conexao no Wi-Fi -----
6 char ssid[] = "<WI-FI Network>";
7 char password[] = "<Password>";
8 //----- Nucleo para execucao do codigo -----
9 static int taskCoreZero = 0;
10 static int taskCoreOne = 1;
11 #define ORG "4znn9d" // your organization or "quickstart"
12 #define DEVICE_TYPE "Esp32-S1" // use this default for quickstart or

```

```
        customize to your registered device type
13 #define DEVICE_ID "Sensor1" // use this default for quickstart or
    customize to your registered device id
14 #define TOKEN "<DEVICE_TOKEN>" // your device token or not used with
    "quickstart"
15 //----- Customise the above values -----
16 #define N 1024 // Numero de valores para calculo da FFT
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";//
    internetofthings.ibmcloud.com
18 char topic[] = "iot-2/evt/status/fmt/json";
19 char authMethod[] = "use-token-auth";
20 char token[] = TOKEN;
21 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
22 //----- Conta Voltas -----
23 const byte interruptPin = 34;
24 unsigned long previousMillis=0;
25 WiFiClient wifiClient;
26 PubSubClient client(server, 1883, wifiClient);
27 // ----- VARIAVEIS PARA CALCULAR RPM -----
28 float rpm_machine = 0;
29 float freq_machine = 0;
30 unsigned long t_start = 0, t_stop = 0;
31 byte muda_status = 0;
32 float tempo = 0;
33 void coreTaskZero( void * pvParameters ){
34     String taskMessage = "Task running on core ";
35     taskMessage = taskMessage + xPortGetCoreID();
36     while(true){
37         // ----- CALCULANDO VELOCIDADE -----
38         if ((muda_status == 0) && (digitalRead(interruptPin) == 1))
39         {
40             t_start = millis();
41             muda_status = 1;
42         }
43         if ((muda_status == 1) && (digitalRead(interruptPin) == 0))
44         {
45             tempo = t_start - t_stop;
46             t_stop = t_start;
47             muda_status = 0;
```

```
48     }
49     if ((tempo > 15) && (tempo < 2000))
50     {
51         freq_machine = 1/(tempo/1000);
52         rpm_machine = freq_machine*60;
53         //Serial.print("RPM: ");
54         //Serial.println(rpm_machine);
55     }
56     if (millis() - t_start > 2000)
57     {
58         freq_machine = 0;
59         rpm_machine = 0;
60     }
61 }
62 }
63 void setup() {
64     pinMode(interruptPin, INPUT);
65 //----- Habilitando Serial -----
66     Serial.begin(250000);
67     delay(30);
68     Serial.println();
69 //----- Conexao WiFi -----
70     while (WiFi.status() != WL_CONNECTED) {
71         Serial.print("Attempting to connect to SSID: ");
72         Serial.println(ssid);
73         WiFi.begin(ssid, password); // Connect to WPA/WPA2 network.
74             // Change this line if using open or WEP network:
75         delay(5000); // wait 5 seconds for connection:
76     }
77     Serial.println(""); Serial.print("WiFi connected, IP address: ");
78     Serial.println(WiFi.localIP());
79 //----- Criando Tarefas para determinado Nucleo -----
80     Serial.print("Starting to create task on core ");
81     Serial.println(taskCoreZero);
82     xTaskCreatePinnedToCore(
83             coreTaskZero, /* Function to implement the
84                         task */
85             "coreTask", /* Name of the task */
86             10000, /* Stack size in words */
```

```
84             NULL,           /* Task input parameter */
85             1,                /* Priority of the task */
86             NULL,           /* Task handle. */
87             taskCoreZero); /* Core where the task should
88             run */
89     Serial.println("Task created... ");
90 }
91 int temperature_measurement = 0;
92 //Tempo em que o ultimo envio foi feito
93 long lastPublishTime = 0;
94 //Intervalo entre os envios
95 #define INTERVAL 10000
96 // Accelerometer Data
97 int16_t temp2[3];
98 int Fs = 0;
99 int Fs_calc = 0;
100 //FFT
101 double X[N], Y[N], Z[N];
102 double Xr[N], Xi[N], Fr[N/2];
103 double Yr[N], Yi[N];
104 // Calculo dos valores maximos
105 double six_max_freq[6];
106 double six_max_amp[6];
107 double amp1[6], freq1[6], amp2[6], freq2[6], amp3[6], freq3[6], amp4
108 [6], freq4[6], amp5[6], freq5[6], amp6[6], freq6[6], amp7[6],
109 freq7[6], amp_med[6], freq_med[6];
110 double six_max_freqy[6];
111 double six_max_ampy[6];
112 double amp1y[6], freq1y[6], amp2y[6], freq2y[6], amp3y[6], freq3y
113 [6], amp4y[6], freq4y[6], amp5y[6], freq5y[6], amp6y[6], freq6y
114 [6], amp7y[6], freq7y[6], amp_medy[6], freq_medy[6];
115 int contador;
116 int k = 0;
117 int kk = 0;
118 int roda = 1;
119 void loop() {
120     start_acc();
121     contador=0;
122     //Funcao para contar algumas vezes a velocidade do acelerometro e
```

```
    enviar o valor consolidado para o loop definitivo
118 while(k<=3)
119 {
120     get_xyz(temp2);
121     // Funcao para contar a frequencia
122     unsigned long currentMillis = millis();
123     // How much time has passed, accounting for rollover with
124     // subtraction!
125     if ((unsigned long)(currentMillis - previousMillis) >= 1000)
126     {
127         // It's time to do something!
128         Fs_calc = Fs;
129         // Use the snapshot to set track time until next event
130         previousMillis = currentMillis;
131         Fs=0;
132         k++;
133     }
134 }
135 while (1){
136     get_xyz(temp2);
137     X[contador]=(double)(2*temp2[0])/32768;
138     Y[contador]=(double)(2*temp2[1])/32768;
139     //Z[contador]=(double)(2*temp2[2])/32768;
140     contador++;
141     if (contador>=N){
142         calcula_FFT();
143         calcula_valores_maximos();
144         roda++;
145         if (roda == 9)
146             { // ----- ENVIANDO PACOTE DE DADOS AO WATSON
147                 -----
148                 envia_IOT();
149                 roda = 1;
150             }
151             contador=0;
152     }
153 // ----- CONECTANDO AO WATSON -----
154 if (!client.connected()) {
```

```
154     Serial.print("Reconnecting client to "); Serial.println(server);
155     while (!client.connect(clientId, authMethod, token)) {
156         Serial.print(".");
157         delay(500);
158     }
159     Serial.println();
160 }
161 }
162 }
163 void calcula_valores_maximos()
164 {
165     max_freq(Xr, Fs_calc, freq_machine);
166     max_freq(Yr, Fs_calc, freq_machine);
167
168     if (roda==1)
169     {
170         while ( kk < 6)
171         {
172             amp1[kk] = six_max_amp[kk];
173             freq1[kk] = six_max_freq[kk];
174             amp1y[kk] = six_max_ampy[kk];
175             freq1y[kk] = six_max_freqy[kk];
176             kk++;
177         }
178         kk = 0;
179     }
180     if (roda==2)
181     {
182         while ( kk < 6)
183         {
184             amp2[kk] = six_max_amp[kk];
185             freq2[kk] = six_max_freq[kk];
186             amp2y[kk] = six_max_ampy[kk];
187             freq2y[kk] = six_max_freqy[kk];
188             kk++;
189         }
190         kk = 0;
191     }
192     if (roda==3)
```

```
193 {
194     while ( kk < 6)
195     {
196         amp3[kk] = six_max_amp[kk];
197         freq3[kk] = six_max_freq[kk];
198         amp3y[kk] = six_max_ampy[kk];
199         freq3y[kk] = six_max_freqy[kk];
200         kk++;
201     }
202     kk = 0;
203 }
204 if (roda==4)
205 {
206     while ( kk < 6)
207     {
208         amp4[kk] = six_max_amp[kk];
209         freq4[kk] = six_max_freq[kk];
210         amp4y[kk] = six_max_ampy[kk];
211         freq4y[kk] = six_max_freqy[kk];
212         kk++;
213     }
214     kk = 0;
215 }
216 if (roda==5)
217 {
218     while ( kk < 6)
219     {
220         amp5[kk] = six_max_amp[kk];
221         freq5[kk] = six_max_freq[kk];
222         amp5y[kk] = six_max_ampy[kk];
223         freq5y[kk] = six_max_freqy[kk];
224         kk++;
225     }
226     kk = 0;
227 }
228 if (roda==6)
229 {
230     while ( kk < 6)
231     {
```

```
232     amp6[kk] = six_max_amp[kk];
233     freq6[kk] = six_max_freq[kk];
234     amp6y[kk] = six_max_ampy[kk];
235     freq6y[kk] = six_max_freqy[kk];
236     kk++;
237 }
238 kk = 0;
239 }
240 if (roda==7)
241 {
242     while ( kk < 6)
243     {
244         amp7[kk] = six_max_amp[kk];
245         freq7[kk] = six_max_freq[kk];
246         amp7y[kk] = six_max_ampy[kk];
247         freq7y[kk] = six_max_freqy[kk];
248         kk++;
249     }
250     kk = 0;
251 }
252 //calcular o valor medio das medidas
253 if (roda==8)
254 {
255     while ( kk < 6)
256     {
257         amp_med[kk] = (amp1[kk]+amp2[kk]+amp3[kk]+amp4[kk]+amp5[kk]+
258                         amp6[kk]+amp7[kk])/7;
259         freq_med[kk] = (freq1[kk]+freq2[kk]+freq3[kk]+freq4[kk]+freq5[
260                         kk]+freq6[kk]+freq7[kk])/7;
261         amp_medy[kk] = (amp1y[kk]+amp2y[kk]+amp3y[kk]+amp4y[kk]+amp5y [
262                         kk]+amp6y[kk]+amp7y[kk])/7;
263         freq_medy[kk] = (freq1y[kk]+freq2y[kk]+freq3y[kk]+freq4y[kk]+
264                         freq5y[kk]+freq6y[kk]+freq7y[kk])/7;
265         kk++;
266     }
267 /*Serial.println("Media");
268 text = String("{\"SR\":\"") + String(Fs_calc, DEC)
269     + String(",\"a1\":\"") + String(amp_med[0], 3) + String(",\"f1"
270     \":") + String(freq_med[0], 2)
```

```
266     + String(",\"a2\":"") + String(amp_med[1], 3) + String(",\"f2
267         \":"") + String(freq_med[1], 2)
268     + String(",\"a3\":"") + String(amp_med[2], 3) + String(",\"f3
269         \":"") + String(freq_med[2], 2)
270     + String(",\"a4\":"") + String(amp_med[3], 3) + String(",\"f4
271         \":"") + String(freq_med[3], 2)
272     + String(",\"a5\":"") + String(amp_med[4], 3) + String(",\"f5
273         \":"") + String(freq_med[4], 2)
274     + String(",\"a6\":"") + String(amp_med[5], 3) + String(",\"f6
275         \":"") + String(freq_med[5], 2)+ String("}\r\n");
276     Serial.println(text);*/
277     kk = 0;
278 }
279 }
280 void calcula_FFT(){
281     myFFT2(Xr, Xi, 0, X, N, 0, 1);
282     myFFT2(Yr, Yi, 0, Y, N, 0, 1);
283     //Serial.print("Hi: ");
284     //Serial.print(Fs_calc);
285     //Serial.println();
286     // Calcula o modulo a magnitude do sinal na frequencia faz um
287     // vetor com as frequencias e amplitudes
288     for (uint16_t i = 0; i < N; i++)
289     {
290         Xr[i]=2*sqrt(Xr[i]*Xr[i] + Xi[i]*Xi[i])/N;
291         Yr[i]=2*sqrt(Yr[i]*Yr[i] + Yi[i]*Yi[i])/N;
292         //Serial.println(Xr[i]);
293     }
294     if (Xr[0] > -1){
295         Xr[0] = 0;
296     }
297     for (uint16_t i = 0; i < N/2; i++)
298     {
299         Fr[i] = i*Fs_calc*1.0/N;
300     }
301     for (uint16_t i = 0; i < N/2; i++)
302     {
303         /*Serial.print(Fr[i]);
304         Serial.print(";");
305     }
```

```
299         Serial.print(Xr[i]);
300         Serial.println();*/
301     }
302 }
303 void envia_IOT()
304 { //Send data to IBM Watson
305     long now = millis();
306     if (now - lastPublishTime > INTERVAL){
307         lastPublishTime = now;
308         String payload = "{\"";
309         payload += "\"d\": {";
310         payload += "\"RPM\": ";
311         payload += String (rpm_machine);
312         payload += ",";
313         payload += "\"A1\": ";
314         payload += String (amp_med[0]);
315         payload += ",";
316         payload += "\"A2\": ";
317         payload += String (amp_med[1]);
318         payload += ",";
319         payload += "\"A3\": ";
320         payload += String (amp_med[2]);
321         payload += ",";
322         payload += "\"A4\": ";
323         payload += String (amp_med[3]);
324         payload += ",";
325         payload += "\"A5\": ";
326         payload += String (amp_med[4]);
327         payload += ",";
328         payload += "\"A6\": ";
329         payload += String (amp_med[5]);
330         payload += ",";
331         payload += "}}";
332         Serial.print("Sending payload 1: "); Serial.println(payload);
333         if (client.publish(topic, (char*) payload.c_str())) {
334             Serial.println("Publish ok");
335         } else {
336             Serial.println("Publish failed");
337         }
```

```
338     String payload2 = "{";
339     payload2 += "\"d\": {";
340     payload2 += "\"F1\":";
341     payload2 += String (freq_med[0]);
342     payload2 += ",";
343     payload2 += "\"F2\":";
344     payload2 += String (freq_med[1]);
345     payload2 += ",";
346     payload2 += "\"F3\":";
347     payload2 += String (freq_med[2]);
348     payload2 += ",";
349     payload2 += "\"F4\":";
350     payload2 += String (freq_med[3]);
351     payload2 += ",";
352     payload2 += "\"F5\":";
353     payload2 += String (freq_med[4]);
354     payload2 += ",";
355     payload2 += "\"F6\":";
356     payload2 += String (freq_med[5]);
357     payload2 += ",";
358     payload2 += "}}";
359     Serial.print("Sending payload 2: ");
360     Serial.println(payload2);
361     if (client.publish(topic, (char*) payload2.c_str())) {
362         Serial.println("Publish ok");
363     } else {
364         Serial.println("Publish failed");
365     }
366     String payload3 = "{";
367     payload3 += "\"d\": {";
368     payload3 += "\"A1y\":";
369     payload3 += String (amp_medy[0]);
370     payload3 += ",";
371     payload3 += "\"A2y\":";
372     payload3 += String (amp_medy[1]);
373     payload3 += ",";
374     payload3 += "\"A3y\":";
375     payload3 += String (amp_medy[2]);
376     payload3 += ",";
377     payload3 += "\"A4y\":";
```

```
377     payload3 += String (amp_medy[3]);
378     payload3 += ",";
379     payload3 += "\"A5y\":\"";
380     payload3 += String (amp_medy[4]);
381     payload3 += ",";
382     payload3 += "\"A6y\":\"";
383     payload3 += String (amp_medy[5]);
384     payload3 += ",";
385     payload3 += "}}";
386     Serial.print("Sending payload 3: "); Serial.println(payload3);
387     if (client.publish(topic, (char*) payload3.c_str())) {
388         Serial.println("Publish ok");
389     } else {
390         Serial.println("Publish failed");
391     }
392     String payload4 = "{";
393     payload4 += "\"d\": {";
394     payload4 += "\"F1y\":\"";
395     payload4 += String (freq_medy[0]);
396     payload4 += ",";
397     payload4 += "\"F2y\":\"";
398     payload4 += String (freq_medy[1]);
399     payload4 += ",";
400     payload4 += "\"F3y\":\"";
401     payload4 += String (freq_medy[2]);
402     payload4 += ",";
403     payload4 += "\"F4y\":\"";
404     payload4 += String (freq_medy[3]);
405     payload4 += ",";
406     payload4 += "\"F5y\":\"";
407     payload4 += String (freq_medy[4]);
408     payload4 += ",";
409     payload4 += "\"F6y\":\"";
410     payload4 += String (freq_medy[5]);
411     payload4 += ",";
412     payload4 += "}}";
413     Serial.print("Sending payload 4: "); Serial.println(payload4);
414     if (client.publish(topic, (char*) payload4.c_str())) {
415         Serial.println("Publish ok");
```

```
416     } else {
417         Serial.println("Publish failed");
418     }
419 }
420 }
421 // Funcao para retornar o valor maximo // inline indica ao
422 // compilador para colocar a funcao dentro do codigo normal
423 inline double max_freq(double Xr[], int16_t Fs, int freq_maq){
424     double maxk;
425     double maxvalue = -1;
426     double temp;
427     double nowFre;
428     int vet1 = 0;
429     for (int i = 1; i < N/2; i++){
430         nowFre = i*Fs*1.0/N;
431         temp = Xr[i];
432         if ((temp > maxvalue) && (abs(freq_maq-nowFre) < 0.8)){
433             maxk = nowFre;
434             maxvalue = temp;
435             vet1 = i;
436         }
437     six_max_amp[0] = maxvalue;
438     six_max_freq[0] = maxk;
439     //2x
440     maxvalue = -1;
441     for (int i = vet1; i < N/2; i++){
442         nowFre = i*Fs*1.0/N;
443         temp = Xr[i];
444         if ((temp > maxvalue) && (abs(freq_maq*2-nowFre) < 0.8)){
445             maxk = nowFre;
446             maxvalue = temp;
447             vet1=i;
448         }
449     }
450     six_max_amp[1] = maxvalue;
451     six_max_freq[1] = maxk;
452     //3x
453     maxvalue = -1;
```

```
454 for (int i = vet1; i < N/2; i++) {
455     double nowFre = i*Fs*1.0/N;
456     double temp = Xr[i];
457     if ((temp > maxvalue) && (abs(freq_maq*3-nowFre) < 0.8)){
458         maxk = nowFre;
459         maxvalue = temp;
460         vet1=i;
461     }
462 }
463 six_max_amp[2] = maxvalue;
464 six_max_freq[2] = maxk;
//4x
466 maxvalue = -1;
467 for (int i = vet1; i < N/2; i++) {
468     double nowFre = i*Fs*1.0/N;
469     double temp = Xr[i];
470     if ((temp > maxvalue) && (abs(freq_maq*4-nowFre) < 0.8)){
471         maxk = nowFre;
472         maxvalue = temp;
473         vet1=i;
474     }
475 }
476 six_max_amp[3] = maxvalue;
477 six_max_freq[3] = maxk;
//5x
479 maxvalue = -1;
480 for (int i = vet1; i < N/2; i++) {
481     double nowFre = i*Fs*1.0/N;
482     double temp = Xr[i];
483     if ((temp > maxvalue) && (abs(freq_maq*5-nowFre) < 0.8)){
484         maxk = nowFre;
485         maxvalue = temp;
486         vet1=i;
487     }
488 }
489 six_max_amp[4] = maxvalue;
490 six_max_freq[4] = maxk;
//6x
492 maxvalue = -1;
```

```
493 for (int i = vet1; i < N/2; i++) {
494     double nowFre = i*Fs*1.0/N;
495     double temp = Xr[i];
496     if ((temp > maxvalue) && (abs(freq_maq*6-nowFre) < 0.8)){
497         maxk = nowFre;
498         maxvalue = temp;
499         vet1=i;
500     }
501 }
502 six_max_amp[5] = maxvalue;
503 six_max_freq[5] = maxk;
504 }
505 // Funcao para retornar o valor maximo// inline indica ao compilador
      para colocar a funcao dentro do codigo normal
506 inline double max_freqy(double Yr[], int16_t Fs, int freq_maq){
507     double maxk;
508     double maxvalue = -1;
509     double temp;
510     double nowFre;
511     int vet1 = 0;
512     for (int i = 1; i < N/2; i++){
513         nowFre = i*Fs*1.0/N;
514         temp = Yr[i];
515         if ((temp > maxvalue) && (abs(freq_maq-nowFre) < 0.8)){
516             maxk = nowFre;
517             maxvalue = temp;
518             vet1 = i;
519         }
520     }
521     six_max_ampy[0] = maxvalue;
522     six_max_freqy[0] = maxk;
523 //2x
524     maxvalue = -1;
525     for (int i = vet1; i < N/2; i++){
526         nowFre = i*Fs*1.0/N;
527         temp = Yr[i];
528         if ((temp > maxvalue) && (abs(freq_maq*2-nowFre) < 0.8)){
529             maxk = nowFre;
530             maxvalue = temp;
```

```
531     vet1=i;
532 }
533 }
534 six_max_ampy[1] = maxvalue;
535 six_max_freqy[1] = maxk;
536 //3x
537 maxvalue = -1;
538 for (int i = vet1; i < N/2; i++){
539     double nowFre = i*Fs*1.0/N;
540     double temp = Yr[i];
541     if ((temp > maxvalue) && (abs(freq_maq*3-nowFre) < 0.8)){
542         maxk = nowFre;
543         maxvalue = temp;
544         vet1=i;
545     }
546 }
547 six_max_ampy[2] = maxvalue;
548 six_max_freqy[2] = maxk;
549 //4x
550 maxvalue = -1;
551 for (int i = vet1; i < N/2; i++){
552     double nowFre = i*Fs*1.0/N;
553     double temp = Yr[i];
554     if ((temp > maxvalue) && (abs(freq_maq*4-nowFre) < 0.8)){
555         maxk = nowFre;
556         maxvalue = temp;
557         vet1=i;
558     }
559 }
560 six_max_ampy[3] = maxvalue;
561 six_max_freqy[3] = maxk;
562 //5x
563 maxvalue = -1;
564 for (int i = vet1; i < N/2; i++){
565     double nowFre = i*Fs*1.0/N;
566     double temp = Yr[i];
567     if ((temp > maxvalue) && (abs(freq_maq*5-nowFre) < 0.8)){
568         maxk = nowFre;
569         maxvalue = temp;
```

```
570     vet1=i;
571 }
572 }
573 six_max_ampy[4] = maxvalue;
574 six_max_freqy[4] = maxk;
575 //6x
576 maxvalue = -1;
577 for (int i = vet1; i < N/2; i++){
578     double nowFre = i*Fs*1.0/N;
579     double temp = Yr[i];
580     if ((temp > maxvalue) && (abs(freq_maq*6-nowFre) < 0.8)){
581         maxk = nowFre;
582         maxvalue = temp;
583         vet1=i;
584     }
585 }
586 six_max_ampy[5] = maxvalue;
587 six_max_freqy[5] = maxk;
588 }
589 // Grava um registrador no acelerometro
590 void i2c_write(int16_t address, int16_t reg, int16_t data){
591     Wire.beginTransmission(address);
592     Wire.write(reg);
593     Wire.write(data);
594     Wire.endTransmission();
595 }
596 // Faz a leitura de um registrador do acelerometro
597 int16_t i2c_read(int16_t address, int16_t reg){
598     Wire.beginTransmission(address);
599     Wire.write(reg);
600     Wire.endTransmission(false);
601     Wire.requestFrom(address, 1);
602     if(Wire.available()){
603         return(Wire.read());
604     }
605 }
606 // Faz a leitura do de XYZ do acelerometro
607 inline void get_xyz(int16_t *temp2){
608     int16_t accel[6];
```

```
609     acc_read(0x1D, 0x01, accel, 6);
610     temp2[0]=((accel[0]<<8)|(accel[1]));
611     temp2[1]=((accel[2]<<8)|(accel[3]));
612     temp2[2]=((accel[4]<<8)|(accel[5]));
613 }
614 // Faz a leitura bruta dos registrados de XYZ do acelerometro
615 inline void acc_read(int16_t address, int16_t reg, int16_t *acc,
616   int16_t qty){
617   int16_t i;
618   while(((i2c_read(0x1D,0x00)&(0b00001000))&&(0b00001000))==0;
619   //Serial.println(i2c_read(0x1D,0x00));
620   // Wire.requestFrom(address, qty, reg, 1, true);
621   Wire.beginTransmission(address);
622   Wire.write(reg);
623   Wire.endTransmission(false); //endTransmission but keep the
624   //connection active
625   Wire.requestFrom(address, qty); //Ask for 1 byte, once done, bus
626   //is released by default
627   i=0;
628   while (Wire.available()) { // slave may send less than requested
629     acc[i] = Wire.read(); // receive a byte as character
630     i++;
631   }
632 }
633 // Inicia o acelerometro MMA8452Q
634 void start_acc()
635 {
636   //Pause Accel (necessario para alterar configuracoes)
637   i2c_write(0x1D, 0x2A, 0b00000000);
638   //XYZ_DATA_CFG Register //000a00bc
639   // a = 1 -> ativa filtro passa alta
640   // bc = 00 -> 2g // bc = 01 -> 4g // bc = 10 -> 8g
641   i2c_write(0x1D, 0x0E, 0b00000000);
642   //CTRL_REG1 //000000ab
643   //a = 1 -> modo de leitura rapida (8 bits) //b = 1 -> modo ativo
644   //Liga o Modo Ativo
645   i2c_write(0x1D, 0x2A, 0b00000001);
646 }
```

APÊNDICE C – Técnicas de Análise de Dados

C.1 Regressão Logística

A regressão logística é a técnica mais utilizada em classificadores de aprendizado de máquina e possui outras denominações, como modelo logístico (KIM *et al.*, 2013). A probabilidade de um evento é uma combinação linear de variáveis independentes ou preditoras, com saída binária, então o modelo de regressão logística prevê $P(Y = 1)$ como uma função de X. A função logística é uma função sigmoide onde cada entrada t , ($t \in \mathbb{R}$) e saída é um valor entre zero e um. A função logística é definida como na equação C.1:

$$\sigma(t) = \frac{1}{1 + e^{-t}} \quad (\text{C.1})$$

Assumindo que t é uma função linear, t pode ser expressado como $t = \beta_0 + \beta_1 x$, então, a função logística pode ser escrita como na equação C.2

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad (\text{C.2})$$

onde $p(x)$ é a probabilidade da variável dependente ser igual a verdadeiro ou falso.

C.2 Rede Neural Artificial

O uso de redes neurais artificiais é motivado pela possibilidade de elaborar mecanismos distintos para problemas intratáveis ou não resolvidos ainda que utilizando uma base computacional. Entretanto, é possível criar condições que reproduzem habilidade cognitivas e habilidades de processamento que são muito desejáveis em várias aplicações. Contudo, é necessário distinguir quais problemas são propensos à este tipo de tratamento (HAYKIN, 2008; KARNGA, 1999; TANG *et al.*, 2016).

Quando um sistema apresenta multidimensionalidade e variáveis sujeitas a interações não-lineares, quando é possível coletar os dados e aprender com eles, a rede neural artificial torna-se um caminho possível para modelar o sistema.

O modelo matemático de um neurônio artificial apresentam simplificações com o propósito de representar os aspectos mais significantes. Dessa forma, detalhes menos importantes são descartados, permitindo sua modelagem. A Fig. 66 apresenta o modelo matemático de um neurônio artificial (HAYKIN, 2008; KARNGA, 1999).

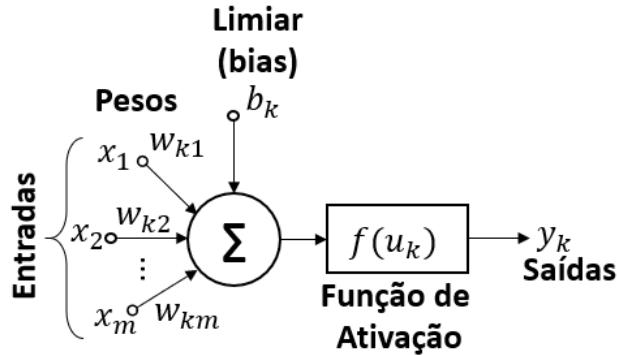


Figura 66 – Modelo matemático de um neurônio artificial

A saída y_k do neurônio k pode ser escrita pela equação C.3:

$$y_k = f(u_k) = f \left(\sum_{j=1}^m w_{kj} x_j + b_k \right) \quad (\text{C.3})$$

onde u_k é o sinal de entrada da função de ativação $f(u_k)$, w_{kj} é o peso associado a cada entrada j , x_j até a j -ésima entrada, b_k é o bias e m é o número de entradas.

Uma simplificação pode ser feita na equação C.3 incluindo o *bias* como um sinal de entrada de valor $x_0 = 1$ com peso associado $w_{k0} = b_k$, como apresentado na equação C.4.

$$y_k = f(u_k) = f \left(\sum_{j=0}^m w_{kj} x_j \right) = f(\mathbf{w}^T \mathbf{x}) \quad (\text{C.4})$$

no qual $\mathbf{w}^T = [w_{k0} \ w_{k1} \cdots w_{km}]$ e $\mathbf{x} = [x_0 \ x_1 \cdots x_m]^T$. O processo de conexão entre os neurônios artificiais leva à geração de sinapses e à construção de uma RNA (Rede Neural Artificial).

C.2.1 Multi-Layer Perceptron

O MLP é uma classe de RNA feedforward. Uma MLP consistem em pelo menos três nós: uma cada de entrada, uma camada oculta e uma camada de saída. Cada nó é um neurônio que usa funções de ativação não lineares. No MLP, a saída de cada neurônio do layer anterior é a entrada para cada neurônio da camada seguinte. Esta estrutura é conhecida como

rede neural *perceptron* com camada intermediária. A MLP usa uma técnica de aprendizado chamada *backpropagation* para treinamento. É capaz de classificar dados que são separáveis não linearmente (ROJAS, 1996; HAYKIN, 1998).

A função de ativação utilizada neste trabalho foi: Rectified Linear Units (ReLU), equação (C.5) e a Sigmoide, equação (C.6). A ReLU (NAIR; HINTON, 2010) é uma função de ativação geral usada na maioria dos casos atualmente e a sigmoide geralmente é melhor empregada em classificadores. Essas funções de ativação foram escolhidas devido a estas características.

$$f(x) = \max(0, x) \quad (\text{C.5})$$

$$f(x) = \frac{1}{(1 + e^{-x})} \quad (\text{C.6})$$

Uma RNA-MLP foi desenvolvida para obter resultados que podem representar o comportamento do sinal e relacioná-los com a condição atual da máquina. Dos valores de entrada amostrados, 70% deles são usados para treinar a rede neural, 20% para testá-la e 10% para validação.

APÊNDICE D – Códigos de Análise de Dados em Python

Será descrito detalhadamente os código em Python utilizados para obtenção dos resultados apresentados neste trabalho.

D.1 Regressão Logística

```

import numpy as np
import winsound
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn import model_selection
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
#####
#      VETORES DE TREINO E TESTE      #
#####
dataset = np.genfromtxt("Dados_S1_Treina.txt", delimiter=";")
X_train = dataset[:,0:25] # XY RPM, amplitude e frequencia
media = np.mean(X_train, axis=0)
X_train = X_train/media
y_train = dataset[:,27:28] #O TARGET NÃO MUDA
dataset = np.genfromtxt('Dados_S1_Testa.txt', delimiter=";")
X_test = dataset[:,0:25]
X_test = X_test/media
y_test = dataset[:,27:28]
dataset2 = np.genfromtxt("Dados_S2_Treina.txt", delimiter=";")
X_train2 = dataset2[:,0:25] # XY RPM, amplitude e frequencia
media2 = np.mean(X_train2, axis=0)
X_train2 = X_train2/media2
y_train2 = dataset2[:,27:28] #O TARGET NÃO MUDA

```

```
dataset2 = np.genfromtxt('Dados_S2_Teste.txt', delimiter=";")  
X_test2 = dataset2[:,0:25]  
X_test2 = X_test2/media2  
y_test2 = dataset2[:,27:28]  
print("Número de dados para treinamento {0} e número de dados para  
teste {1}.".format(len(X_train), len(X_test)))  
#####  
#          Regressão Logistica      #  
#####  
# fit a logistic regression model to the data  
logreg = LogisticRegression()  
logreg.fit(X_train, y_train)  
print(logreg)  
y_pred = logreg.predict(X_test)  
print('Accuracy of logistic regression classifier on test set: {:.2f}'  
     .format(logreg.score(X_test, y_test)))  
# cross validation  
kfold = model_selection.KFold(n_splits=100, random_state=7)  
modelCV = LogisticRegression()  
scoring = 'accuracy'  
results = model_selection.cross_val_score(modelCV, X_train, y_train,  
                                         cv=kfold, scoring=scoring)  
print("10-fold cross validation average accuracy: %.3f" % (results.mean()))  
#Confusion Matrix  
confusion_matrix = confusion_matrix(y_test, y_pred)  
print(confusion_matrix)  
from sklearn.metrics import roc_auc_score  
from sklearn.metrics import roc_curve  
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))  
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba  
                                 (X_test)[:,1])  
plt.figure(1)  
plt.subplot(121)  
plt.plot(fpr, tpr, label='Logistic Regression\n(area = %0.2f)' % logit_roc_auc)  
plt.plot([0, 1], [0, 1], 'r--')  
plt.xlim([0.0, 1.0])  
plt.ylim([0.0, 1.05])
```

```
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Sensor 1\n(Obstruction)')
plt.legend(loc="lower right")
#plt.savefig('Log_ROC')
#####
#      Sensor 2
#####
print("Número de dados para treinamento {0} e número de dados para
      teste {1}.format(len(X_train2), len(X_test2)))
#####
#      Regressão Logistica      #
#####
# fit a logistic regression model to the data
logreg2 = LogisticRegression()
logreg2.fit(X_train2, y_train2)
print(logreg2)
y_pred2 = logreg2.predict(X_test2)
print('Accuracy of logistic regression classifier on test set: {:.2f}'
      .format(logreg.score(X_test2, y_test2)))
# cross validation
kfold2 = model_selection.KFold(n_splits=100, random_state=7)
modelCV2 = LogisticRegression()
scoring2 = 'accuracy'
results2 = model_selection.cross_val_score(modelCV2, X_train2, y_train2,
                                           cv=kfold2, scoring=scoring2)
print("10-fold cross validation average accuracy: {:.3f} % (results2.mean())")
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc2 = roc_auc_score(y_test2, logreg2.predict(X_test2))
fpr, tpr, thresholds = roc_curve(y_test, logreg2.predict_proba(X_test2)
                                 [:,1])
plt.figure(1)
plt.subplot(122)
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc2)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0.0, 1.0])
```

```

plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Sensor 2\n(Obstruction)')
# plt.title('Receiver operating characteristic (ROC)')
plt.legend(loc="lower right")
#plt.savefig('Log_ROC')
plt.show()

```

D.2 Rede Neural - Multi-layer Perceptron

```

import numpy as np
import winsound
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.utils import to_categorical
from keras.layers import Dropout
from keras.models import load_model
from keras import layers
from keras import metrics
from sklearn.multioutput import MultiOutputClassifier
import matplotlib.pyplot as plt
#####
#      VETORES DE TREINO E TESTE      #
#####
dataset = np.genfromtxt("Dados_S1_Treina.txt", delimiter=";")
X_train = dataset[:,0:25] # XY RPM, amplitude e frequencia
media = np.mean(X_train, axis=0)
X_train = X_train/media
y_train = dataset[:,26:27] #O TARGET NÃO MUDA
dataset = np.genfromtxt('Dados_S1_Testa.txt', delimiter=";")
X_test = dataset[:,0:25]
X_test = X_test/media
y_test = dataset[:,26:27]
dataset2 = np.genfromtxt("Dados_S2_Treina.txt", delimiter=";")
X_train2 = dataset2[:,0:25] # XY RPM, amplitude e frequencia
media2 = np.mean(X_train2, axis=0)

```

```
X_train2 = X_train2/media2
y_train2 = dataset2[:,26:27] #O TARGET NÃO MUDA
dataset2 = np.genfromtxt('Dados_S2_Teste.txt', delimiter=";") 
X_test2 = dataset2[:,0:25]
X_test2 = X_test2/media2
y_test2 = dataset2[:,26:27]
#####
# Dados da rede de sensores #
#####
X_train_final = np.concatenate((X_train, X_train2), axis=0)
media_final = np.mean(X_train_final, axis=0)
X_train_final = X_train_final/media_final
y_train_final = np.concatenate((y_train, y_train2), axis=0)
X_test_final = np.concatenate((X_test, X_test2), axis=0)
X_test_final = X_test_final/media_final
y_test_final = np.concatenate((y_test, y_test2), axis=0)
print("Número de dados para treinamento {0} e número de dados para
      teste {1}.format(len(X_train), len(X_test)))
#####
#          REDE A PARTIR DAQUI          #
#####
# create model
model = Sequential()
# Input - Layer
model.add(Dense(50, input_dim=25, activation='sigmoid')) # LAYER 1
# Hidden - Layers
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
#model.add(Dense(30, activation='relu')) # LAYER 2
#model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
#model.add(Dense(50, activation='relu')) # LAYER 3
#model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
# Output - Layer
model.add(Dense(1, activation='sigmoid')) # SAÍDA
# For a multi-class classification problem
model.compile(optimizer='adam',
              loss='mean_squared_error',
              metrics=['binary_accuracy'])
```

```
model.summary()
epoca = 500
##### sensor 1 #####
# Fit the model
history = model.fit(X_train, y_train, validation_data=(X_test,y_test),
                      epochs=epoca, batch_size=30, verbose=0)
# calculate predictions
results = (model.predict(X_test, batch_size=None, verbose=0))
results = (results > 0.5)
# Creating the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, results)
validacao = (cm[0,0]+cm[1,1])/(cm[0,0] + cm[0,1] + cm[1,0] + cm[1,1])
#print("taxa de acerto treino S1: ",history.history['binary_accuracy',10])
print("taxa de acerto treino S1: ",round(max(history.history
['binary_accuracy']),2)*100, "%")
print("taxa de acerto validacao S1: ", round(validacao*100,2), "%")
#####
##### sensor 2 #####
# Fit the model
history2 = model.fit(X_train2, y_train2, validation_data=(X_test2,y_test2),
                      epochs=epoca, batch_size=30, verbose=0)
# calculate predictions
results2 = (model.predict(X_test2, batch_size=None, verbose=0))
results2 = (results2 > 0.5)
# Creating the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm2 = confusion_matrix(y_test2, results2)
validacao2 = (cm2[0,0]+cm2[1,1])/(cm2[0,0] + cm2[0,1] + cm2[1,0] + cm2[1,1])
print("taxa de acerto treino S2: ",round(max(history2.history
['binary_accuracy']),2)*100, "%")
print("taxa de acerto validacao S2: ", round(validacao2*100,2), "%")
#model.save('T5.h5')
#####
# Dados da rede de sensores #
#####
# Fit the model
history3 = model.fit(X_train_final, y_train_final, validation_data=
```

```
(X_test_final,y_test_final), epochs=epoca, batch_size=30, verbose=0)
# calculate predictions
results3 = (model.predict(X_test_final, batch_size=None, verbose=0))
results3 = (results3 > 0.5)

# Creating the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm3 = confusion_matrix(y_test_final, results3)
validacao3 = (cm3[0,0]+cm3[1,1])/(cm3[0,0] + cm3[0,1] + cm3[1,0] + cm3[1,1])
print("taxa de acerto treino NETWORK: ",round(max(history3.history
['binary_accuracy']),2)*100, "%")
print("taxa de acerto validacao NETWORK: ", round(validacao3*100,2), "%")
#winsound.Beep(700, 1000) # BEEP PARA INDICAR QUE O TERMINOU
# list all data in history
print(history.history.keys())
# summarize history for accuracy
plt.figure(1, figsize = (8.5,3.5))
plt.subplot(131)
plt.plot(history.history['binary_accuracy'], linewidth=2, color = 'red')
plt.plot(history.history['val_binary_accuracy'], linestyle=':',
         linewidth=2, color = 'blue')
plt.title('Sensor 1')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['train', 'test'], loc='lower right')
plt.ylim((0.78,1.01))
# list all data in history
#print(history2.history.keys())
# summarize history for accuracy
plt.figure(1)
plt.subplot(132)
plt.plot(history2.history['binary_accuracy'], linewidth=2, color = 'red')
plt.plot(history2.history['val_binary_accuracy'], linestyle=':',
         linewidth=2, color = 'blue')
plt.title('Sensor 2')
#plt.ylabel('accuracy (%)')
plt.xlabel('Epoch')
plt.legend(['train', 'test'], loc='lower right')
```

```
plt.ylim((0.78,1.01))
plt.figure(1)
plt.subplot(133)
plt.plot(history3.history['binary_accuracy'], linewidth=2, color = 'red')
plt.plot(history3.history['val_binary_accuracy'], linestyle=':', 
         linewidth=2, color = 'blue')
plt.title('Sensor Network')
# plt.ylabel('accuracy (%)')
plt.xlabel('Epoch')
plt.legend(['train', 'test'], loc='lower right')
plt.ylim((0.78,1.01))
plt.show()
```

Anexos

ANEXO A – Ultrassom

A.1 O que é ultrassom?

Ultrassom é o nome dados às ondas sonoras de frequências superiores as audíveis pelo ser humano. Os seres humanos são capazes de ouvir em faixas de frequência de 20 Hz à 20.000 Hz, chamado também de frequências de espectro audível (HELERBROCK, 2019), portanto, ultrassom é som sujas ondas sonoras possuem frequência superior à 20.000 Hz. A Tabela 14 apresenta as frequências sonoras e sua respectiva denominação. Para fins de obtenção de imagens (ultrassonografia), a frequência varia entre 1 MHz e 10 MHz (HELERBROCK, 2019).

Tabela 14 – Denominação das Frequências das Ondas Sonoras

Faixa de Frequência (Hz)	Denominação	Exemplo
0-20	Infrassom	Terremotos
20-20.000	Som audível	Músicas e Conversas
>20.000	Ultrassom	Morcegos, Cristais Osciladores

A.2 Análise de Avarias por Ultrassom

A inspeção por ultrassom permite localizar vazamentos em tubulações que transportam fluidos como ar comprimido, gás, vapor e líquidos em geral. É usado em locais que seja permitida a aproximação do instrumento de medição com a fonte do defeito ou falha exposta na atmosfera, identificando avarias que emitem sons característicos relevantes (não aplicável a merejamento ou gotejamento) (ENGEFAZ, 2019).

Também pode ser utilizado para inspeção em sistemas elétricos cujas falhas emitam sons característicos relevantes, como por exemplo, isoladores em redes elétricas ou terminais elétricos. As principais inspeções realizadas com essa técnica são (ENGEFAZ, 2019; KRAS, 2019):

- Vazamentos externos em tubulações e/ou conexões de ar comprimido e gases;
- Vazamentos internos em sistemas de purgadores de vapor e em válvulas com a função de estanque para fluidos compressíveis ou incompressíveis;;
- Sistemas elétricos (efeito corona), e

- Verificação das condições de operação e lubrificação de vários componentes de equipamentos e máquinas, como engrenagens e rolamentos em geral;

A inspeção por ultrassom é não invasiva, e deve ser feita por equipe especializada de forma periódica nos maquinários da empresa. Os instrumentos de inspeção por ultrassom são constituídos por circuitos eletrônicos especiais que permitem transmitir ao cristal piezoeétrico uma série de pulsos controlados, que se transformam em ondas ultrassônicas pelo cristal. Os sinais captados pelo cristal são visualizados em uma tela em forma de pulsos luminosos, denominados ecos. Estes ecos constituem o registro das descontinuidades no interior do material (INFOSOLDA, 2019).

O método possui alta sensibilidade em detecção de pequenas falhas, como trincas internas e fissuras. Entretanto, necessita de inspetores com grande conhecimento teórico e experiência de campo.