

UNIVERSIDADE FEDERAL DE SANTA CATARINA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

CONRADO VIVEIROS JORDAN

CONDITION MONITORING COM BEACONS BLUETOOTH DE BAIXO CUSTO

FLORIANÓPOLIS

2018

CONRADO VIVEIROS JORDAN

CONDITION MONITORING COM BEACONS BLUETOOTH DE BAIXO CUSTO

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina **DAS 5511: Projeto de Fim de Curso** do curso de Graduação em Engenharia de Controle e Automação.

Orientador(a): Prof. Felipe Gomes de Oliveira Cabral

FLORIANÓPOLIS

2018

Conrado Viveiros Jordan

CONDITION MONITORING COM BEACONS BLUETOOTH DE BAIXO CUSTO

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 31 de julho de 2018

Banca Examinadora:

Gustavo Yugo Shinzato Rodrigues da Cunha
Orientador na Empresa
Byond

Prof. Felipe Gomes de Oliveira Cabral
Orientador no Curso
Universidade Federal de Santa Catarina

Prof. Rodolfo César Costa Flesch
Avaliador
Universidade Federal de Santa Catarina

Artur Straub
Debatedor
Universidade Federal de Santa Catarina

Kaio Siqueira de Souza
Debatedor
Universidade Federal de Santa Catarina

Dedico este trabalho a todos que contribuíram para a minha formação acadêmica assim como para a elaboração deste trabalho.

AGRADECIMENTOS

Agradeço à minha família, aos meus amigos da Escola Autonomia e aos meus colegas da engenharia de controle e automação.

Ao professor Felipe Cabral, ao professor Jomi Hübner e ao professor Ricardo Rabelo pelas orientações e conselhos.

Ao Lucas de Sousa Reibnitz, à Marina da Silveira Tagliari Hoffmann, ao professor Paulo Wendhausen e aos demais integrantes do MAGMA (Grupo de Materiais Magnéticos) na UFSC pela oportunidade de realização de testes em seus laboratórios.

Ao Gustavo Yugo pela oportunidade de estágio.

Ao Gauss e ao Euler.

Por último, mas não menos importante, agradeço à Sabrina Iara.

*“Two things are infinite: the universe
and human stupidity.”*

(Autor desconhecido)

RESUMO

A empresa Byond vende soluções de rastreabilidade para a indústria, principalmente usando tecnologias de identificação por radiofrequência (RFID, do inglês *Radio-Frequency Identification*). Essas soluções englobam a localização de produtos em uma linha de produção, o controle de estoque, o monitoramento de ativos de uma empresa (por exemplo, ter a informação de quantas e quais máquinas estão em manutenção), entre outras. Além disso, tendo ciência da dificuldade de se fazer manutenção preditiva em máquinas devido ao alto custo quando há sensores embutidos, a empresa começou a trabalhar com *beacons Bluetooth* de baixo custo para resolver esse problema. Os *beacons* são acoplados a máquinas e conseguem transmitir constantemente informações do estado atual dos equipamentos através da informação dos sensores: um acelerômetro de 3 eixos, um magnetômetro, um sensor de temperatura e um sensor de luminosidade. Neste trabalho foi feito um estudo da aquisição dessa nova tecnologia para a empresa e foi estudada a viabilidade de se fazer manutenção preditiva de máquinas através de técnicas de *condition monitoring* (monitoramento de parâmetros/condições de uma máquina) usando os *beacons* de baixo custo. Inicialmente foi feito um estudo preliminar das funcionalidades dos *beacons* e um teste em ambiente controlado com um ventilador de pequeno porte. Em seguida, foram aplicados dois algoritmos de verificação de falhas através dos dados obtidos dos sensores: um no domínio do tempo e outro no domínio da frequência. Por último, a solução foi ampliada para um ambiente industrial com a utilização dos *beacons* em uma bomba de vácuo de um laboratório na UFSC. Os resultados confirmaram que a utilização dos *beacons* é vantajosa e prática para as práticas de manutenção preditiva através de *condition monitoring*, além de serem eficientes em relação a custo se comparadas a outras soluções.

Palavras-chave: Manutenção preditiva. *Condition monitoring*. *Beacon*. *Bluetooth*. *Bluetooth Low Energy*.

ABSTRACT

Byond is a company that sells solutions in traceability for industrial environments using mainly Radio-Frequency Identification (RFID) technologies. These solutions include locating products in a production line, stock control, asset tracking (for example, extracting the information of how many and which assets are currently in maintenance), and others. Moreover, knowing that performing predictive maintenance in machines can be expensive due to the high prices of sensor-embedded machines, Byond started working with low-cost Bluetooth Beacons to solve this issue. The beacons are attached to the machines and can transmit constantly the current state of these equipment by using the information obtained from its sensors: a 3-axis accelerometer, a magnetometer, a temperature sensor and a luminosity sensor. In this project, a study of the acquisition of this new technology to the company and the viability of predictive maintenance with condition monitoring techniques (monitoring of some selected parameters of a machine) with these low-cost beacons were made. Firstly, a preliminary study of the beacons main features and specifications and a test in a controlled environment with a reduced size fan were conducted. After these tests were made, two algorithms for recognition of failures in equipment were applied to the data: one in the time domain and the other in the frequency domain. The solution was applied to a more industrial environment with the use of the beacons in a vacuum pump of a laboratory at UFSC. The main results corroborated with the idea that the use of the Bluetooth beacons in industrial environment for predictive maintenance via condition monitoring is practical and useful, while also being cost efficient compared to other solutions.

Key-words: Predictive maintenance. Condition monitoring. Beacon. Bluetooth. Bluetooth Low Energy.

LISTA DE ILUSTRAÇÕES

Figura 1 - <i>Beacons</i> industriais Bluvision	20
Figura 2 - Fluxo de dados desde os <i>beacons</i> até a plataforma Bluzone	21
Figura 3 - Plataforma Bluzone	22
Figura 4 – Página de criação de <i>templates</i>	23
Figura 5 - Página de configuração de um <i>template</i> para <i>beacons</i>	24
Figura 6 - Página de configuração de um <i>template</i> para <i>beacons</i>	25
Figura 7 - Tela de criação de uma <i>policy</i> de <i>condition monitoring</i>	28
Figura 8 - Tela de criação das demais configurações da <i>policy</i> de <i>condition monitoring</i>	29
Figura 9 - Comparação entre sensores internos e <i>beacons</i> de <i>Bluetooth</i>	31
Figura 10 - Aquecedor termoventilador A1 da marca Ventisol utilizado no estudo.	33
Figura 11 - Aquecedor com provocado desequilíbrio para alteração do padrão de vibração.....	34
Figura 12 - <i>Beacon</i> modelo BEEKs industrial utilizado no estudo.	35
Figura 13 - Tela de configurações do <i>beacon</i> com período de duração do modo de alta velocidade destacado em vermelho.	36
Figura 14 – Exemplo de alerta gerado pela <i>policy</i> de <i>condition monitoring</i> através da plataforma Bluzone.....	38
Figura 15 - Valores de aceleração nos 3 eixos do <i>beacon</i> acoplado ao ventilador durante testes no ambiente controlado.	39
Figura 16 - Gráfico do sinal $S(t)$ no tempo.	45
Figura 17 - Espectro de frequência do sinal.	46
Figura 18 – Sinais no domínio do tempo para o ventilador normal e para o ventilador com defeito.	47
Figura 19 - Características estatísticas do sinal no eixo X, na situação normal e com defeito.....	47
Figura 20 - Sinal no tempo para o eixo Y da situação normal e da com defeito.	48
Figura 21 - Sinal no tempo para o eixo Z da situação normal e da com defeito.	49

Figura 22 - Características estatísticas do sinal no eixo Y da situação normal e da com defeito.....	49
Figura 23 - Características estatísticas do sinal no eixo Z da situação normal e da com defeito.....	49
Figura 24 - Espectros de frequência do eixo X em situação normal e com defeito.	50
Figura 25 - Espectros de frequência do eixo Y em situação normal e com defeito.	51
Figura 26 - Espectros de frequência do eixo Z em situação normal e com defeito.	51
Figura 27 - Bomba de vácuo do laboratório MAGMA.	54
Figura 28 - Um dos recipientes usados para armazenamento de ímãs.....	55
Figura 29 - Segundo recipiente usado para armazenamento dos ímãs.	55
Figura 30 - Local de instalação do <i>beacon</i>	56
Figura 31 - Local de instalação do <i>gateway</i> BluFi.	57
Figura 32 – Imagem mais aproximada do BluFi.	57

LISTA DE TABELAS

Tabela 1 - Algumas <i>features</i> comumente utilizadas no método no domínio do tempo.....	43
Tabela 2 - Parâmetros no tempo do primeiro período de funcionamento para os três eixos.....	64
Tabela 3 - Parâmetros no tempo do segundo período de funcionamento para os três eixos.....	64

LISTA DE ABREVIATURAS E SIGLAS

RFID - *Radio-frequency identification*

BLE - *Bluetooth Low Energy*

CM - *Condition Monitoring*

IoT - *Internet of Things* (“Internet das coisas”)

RTLS - *Real-Time Location System*

PLS - *Proximity Location System*

RSSI - *Received Signal Strength Indication*

API - *Application Programming Interface*

SVM - *Support Vector Machine*

RMS - *Root Mean Square*

AWS - *Amazon Web Services*

FFT - *Fast Fourier Transform*

SUMÁRIO

AGRADECIMENTOS.....	4
RESUMO	6
ABSTRACT.....	7
LISTA DE ILUSTRAÇÕES.....	8
LISTA DE TABELAS.....	10
LISTA DE ABREVIATURAS E SIGLAS.....	11
1 INTRODUÇÃO.....	14
2 <i>CONDITION MONITORING</i> E <i>BEACONS</i> DE <i>Bluetooth</i>	18
2.1 <i>BEACONS</i>	18
2.2 PRODUTOS BLUVISION UTILIZADOS.....	19
2.2.1 <i>Beacons</i> Bluvision	20
2.2.2 BluFis.....	20
2.2.3 Bluzone.....	21
2.3 MÉTODO DE <i>CONDITION MONITORING</i> COM OS <i>BEACONS</i> BLUVISION	22
2.3.1 Configuração dos <i>beacons</i>	23
2.3.2 Criação da <i>policy</i> de <i>condition monitoring</i>	27
2.3.3 <i>Condition monitoring</i>	29
2.4 VANTAGENS E LIMITAÇÕES DO USO DE <i>BEACONS</i>	30
3 PROTÓTIPO E TESTES	33
3.1 CONFIGURAÇÃO INICIAL	33
3.2 PERÍODO DE APRENDIZADO do <i>Beacons</i>	35
3.3 TESTES E RESULTADOS	37
4 ALGORITMOS DE <i>CONDITION MONITORING</i>	40
4.1 ALGORITMOS EXISTENTES	40

4.2 DOMÍNIO DO TEMPO	41
4.3 DOMÍNIO DA FREQUÊNCIA.....	44
4.4 IMPLEMENTAÇÃO DOS ALGORITMOS NO PROTÓTIPO	46
5 IMPLEMENTAÇÃO EM MÁQUINA INDUSTRIAL.....	53
5.1 CONFIGURAÇÃO DA MÁQUINA E DO HARDWARE	53
5.2 REALIZAÇÃO DOS TESTES	56
5.3 ANÁLISE NO TEMPO	58
5.4 ANÁLISE NA FREQUÊNCIA.....	60
6 CONSIDERAÇÕES FINAIS E PERSPECTIVAS	61
REFERÊNCIAS	63

1 INTRODUÇÃO

A manutenção de equipamentos é uma prática que gera grande economia de recursos para uma empresa e esse tema tem recebido cada vez mais atenção por elas.

Dentre os diferentes tipos de manutenção que podem ser realizados, há quatro categorias principais: manutenção corretiva, manutenção preventiva, manutenção preditiva e manutenção prescritiva (a última também chamada de manutenção detectiva) [1]. De forma resumida, pode-se explicar as diferentes categorias da seguinte maneira:

- **Manutenção corretiva:** realizada somente após a falha da máquina, quando, então, é consertada ou trocada. É uma das principais formas de manutenção utilizadas atualmente na indústria e a mais simples de todas.
- **Manutenção preventiva:** baseada em uma frequência predeterminada de verificação das condições atuais de funcionamento da máquina. Com essas manutenções agendadas, pode-se trocar peças desgastadas ou realizar ações de manutenção, tais como aplicação de mais lubrificante ou a troca de óleo de um motor. Essa manutenção promove um grande aumento na vida útil dos equipamentos.
- **Manutenção preditiva:** visa reduzir o número de paradas agendadas da manutenção preventiva. Basicamente, usa-se alguma técnica para verificar o estado atual da máquina e qual a previsão de tempo que ela consegue se manter em funcionamento até algum tipo de falha. Desse modo, pode-se fazer a manutenção somente quando necessário e há menos perda com paradas da linha de produção. Nesse contexto o conceito de “*condition monitoring*” é aplicado. *Condition monitoring* é a técnica de se monitorar alguns parâmetros de uma máquina, quando esses parâmetros interferem diretamente sobre o comportamento atual de funcionamento da mesma. Entre esses parâmetros, os mais utilizados são: som (sensoriamento de emissões acústicas da máquina), luz, vibração, temperatura, análise da lubrificação do sistema e

parâmetros elétricos, tais como corrente e tensão [3]. *Condition monitoring* é, muitas vezes, tomado erroneamente como sinônimo de manutenção preditiva. No entanto, existem diferentes técnicas de se fazer manutenção preditiva além de *condition monitoring*. Além do mais, *condition monitoring* pode ser usado em outros casos além de manutenção preditiva, como, por exemplo, no controle de qualidade de produtos [4].

- **Manutenção prescritiva ou detectiva:** esse tipo de manutenção engloba muitas das mesmas técnicas usadas em manutenção preditiva, porém leva a um passo adiante: seu objetivo é, além de perceber uma falha em desenvolvimento, reconhecer qual é essa falha e já gerar um prognóstico de manutenção do equipamento. Esse tipo de manutenção é mais dificilmente aplicado devido às dificuldades de se ter uma “biblioteca” de diferentes falhas para que o sistema reconheça qual delas está se desenvolvendo. Outra maneira que poderia se saber qual falha está para acontecer é conhecer os modelos físicos do sistema suficientemente bem para que consiga se inferir pelos parâmetros o que está acontecendo de errado. Essa opção, devido à complexidade de alguns processos e à quantidade de máquinas e componentes envolvidos, muitas vezes também não é viável.

Levando-se em conta que existem esses diferentes tipos, surge a dúvida de qual é ideal para cada aplicação. Algumas fatores podem ser levados em consideração para a tomada de decisão, como por exemplo: tipo e tamanho da máquina, o quão crítico é o processo ou a máquina para a produção, se a falha da máquina pode gerar consequências catastróficas ou perigosas, preço de substituição da peça/máquina danificada por causa de falhas, redundância, custo e disponibilidade de peças de reposição, entre outros [4]. De modo geral, a manutenção corretiva é usada quando a falha da máquina não gera nenhum risco de segurança, quando a máquina é redundante, quando a máquina tem baixo custo de troca e/ou reparação ou quando a falha não gera nenhum outro dano ou perda considerável para a empresa. Desse modo, utiliza-se esse tipo de manutenção por ser o mais simples de ser realizado. A manutenção preventiva é usada quando se tem uma noção do comportamento estatístico das falhas e pode-se estipular a frequência de manutenção

adequada. Além disso, esse tipo de manutenção é usado quando a variedade das falhas não é muito grande e elas são bem conhecidas, bem como quando, apesar das falhas não gerarem consequências catastróficas, elas podem ser custosas ou inconvenientes. Considera-se usar os dois últimos tipos de manutenção em processos ou manufaturas em que a falha da máquina pode ser perigosa, gerar muitas perdas de recursos ou a substituição é muito demorada e/ou difícil.

De maneira geral, as manutenções preditiva e detectiva parecem ser as melhores soluções considerando-se a minimização das falhas e a maximização da vida útil dos equipamentos, porém a implementação delas traz algumas desvantagens. Além de muitas vezes requerer um histórico de falhas e nem sempre isso ser possível e da necessidade de se fazer um investimento em mão-de-obra especializada, a principal desvantagem por enquanto continua sendo o alto preço da implantação e subsequente operação dos sistemas de *condition monitoring* (incluindo o alto preço de máquinas que possuem sensores embutidos).

Por conta desses altos custos relacionados às manutenções preditiva e prescritiva, a utilização de *beacons* de *Bluetooth* visa trazer uma solução barata para esse problema. Dessa forma, em vez de se investir em poucas máquinas com sensores (que são um grande investimento para a empresa) pode-se fazer a compra desses *beacons* em larga escala e aplicá-los a centenas de diferentes máquinas e dentro dos mais diversos setores da empresa. Adicionalmente, os *beacons* que foram estudados são facilmente acoplados às máquinas sem nenhuma intervenção ou parada no processo de produção, o que também não seria possível se fosse necessário trocar máquinas já existentes por máquinas com sensores embutidos.

Este documento está dividido em cinco capítulos. O capítulo dois apresenta um panorama geral sobre os *beacons* e as ferramentas utilizadas (tanto *hardware* como *software*). No Capítulo 3 é apresentado o protótipo para testes feitos dentro da empresa Byond, em ambiente de escala reduzida, e os resultados obtidos com esses testes. No Capítulo 4 são explorados dois algoritmos de detecção de falhas dentre os vários existentes na literatura, um no domínio do tempo e outro no domínio da frequência. Esses dois algoritmos foram aplicados ao protótipo em ambiente controlado. No Capítulo 5 é mostrado o teste feito em laboratório na UFSC em uma

máquina industrial e, por último, no capítulo 6 são expostas as considerações finais e perspectivas a respeito do projeto desenvolvido.

2 CONDITION MONITORING E BEACONS DE BLUETOOTH

A empresa Byond, onde foram feitos o estágio e o desenvolvimento deste projeto de fim de curso, é especializada em soluções de rastreabilidade na indústria normalmente implantando-as com tecnologias RFID. Visando aproveitar o crescimento de aplicações de formas mais modernas de manutenção (a manutenção preditiva e a prescritiva), a empresa resolveu se especializar também em soluções voltadas aos tipos de manutenção usando técnicas de *condition monitoring*. Devido às vantagens já mencionadas, maior escalabilidade e menor custo, a aplicação das técnicas de *condition monitoring* foi feita utilizando-se *Beacons* de *Bluetooth* de baixo custo.

2.1 BEACONS

“*Beacon*” é uma palavra em inglês que significa algum tipo de objeto que possui como objetivo avisar ou sinalizar tudo que está à sua volta sobre sua existência, podendo servir como aviso de algum perigo ou para enviar alguma informação. Um farol que sinaliza navegações da presença de uma costa ou península é exemplo de algo que pode ser chamado de *Beacon*. *Beacon*, no contexto de indústria e de automação, possui um significado similar: é o nome que se dá a dispositivos, normalmente de pequeno porte, que continuamente transmitem sinais de *Bluetooth* à sua volta para interagir com qualquer outro dispositivo que esteja preparado para receber e processar esses sinais. Esses sinais normalmente incluem algum identificador único do *Beacon*. Os *Beacons*, em geral, possuem aplicações muito diversas como em varejo, eventos, educação, aeroportos, hospitais etc. Essa tecnologia abre muitas possibilidades, como, por exemplo, um cliente andando na rua, passando na frente de uma loja e recebendo automaticamente, em seu smartphone, informações da loja, tais como promoções de produtos. A tecnologia se popularizou principalmente a partir de 2013, com o surgimento do protocolo da Apple para *Beacons* chamado “iBeacon” e, mais tarde, com a criação do padrão *open-source* “Eddystone”, desenvolvido pela empresa Google. Esses *beacons* normalmente são alimentados por bateria e possuem duração de uso de até alguns anos e, para isso,

utilizam uma tecnologia de *Bluetooth* chamada “*Bluetooth Low Energy*” (BLE), por vezes também chamado de “*Bluetooth Smart*”. A tecnologia BLE, apesar de muito similar ao *Bluetooth* convencional, tem como objetivo reduzir consideravelmente o consumo de energia mantendo um alcance similar ao do *Bluetooth* comum, o que aumenta a independência de uso sem troca ou recarga de bateria. Em âmbito industrial, os *Beacons* BLE foram sendo introduzidos para diferentes cenários: *condition monitoring* e manutenção preventiva, gerenciamento de equipamentos e salas, monitoramento do fluxo de funcionários, rastreamento de bens, otimização de processos, entre outros. Neste projeto, lidamos com o uso de *condition monitoring*, usando os *beacons* como sensores para medição de parâmetros desejados e subsequente envio desses dados através de BLE para os *gateways* que, em sequência, os envia à plataforma online para processamento e análise. Os *beacons* utilizados foram desenvolvidos e produzidos pela empresa Bluvision, que oferece alguns tipos diferentes de *beacons* industriais, dependendo da aplicação.

2.2 PRODUTOS BLUVISION UTILIZADOS

A Bluvision possui alguns diferentes produtos que fazem parte de soluções integradas na área de IoT (*Internet of things*). As principais soluções disponibilizadas pela empresa são de: RTLS (*Real-Time Locating System*), que são sistemas de localização de *Beacons* por triangulação, PLS (*Proximity Location System*), sistemas de localização de *Beacons* por proximidade a alguns pontos ou regiões específicas, e *Condition monitoring*, que é o método já explicado de monitoramento de parâmetros de equipamentos. Para essas soluções mencionadas, a empresa oferece três linhas de produtos: os *beacons*, os *gateways* (chamados pela empresa de “*BluFi*”s) e sua plataforma online, chamada Bluzone. A seguir, os produtos serão explicados em mais detalhes.

2.2.1 Beacons Bluvision

Os *beacons* industriais da Bluvision, ilustrados na Figura 1, vêm em diferentes tipos e tamanhos. Eles possuem sensores para detectar diferentes grandezas, bateria que pode durar até nove anos, precisão em usos de localização, autenticação e encriptação de dados, suportam os protocolos iBeacon e Eddystone e possuem resistência a água com classificação IP67.

Figura 1 - *Beacons* industriais Bluvision.



Fonte: site da Bluvision¹.

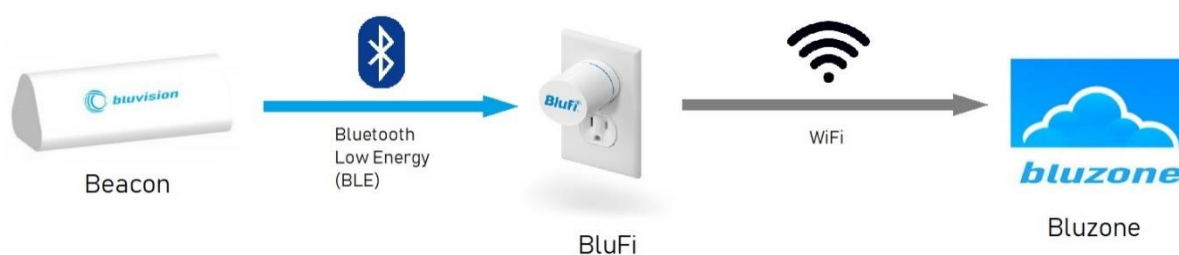
2.2.2 BluFis

Para os dados dos *beacons* serem processados, eles precisam ser enviados à plataforma *online* da Bluvision, a Bluzone. Os “BluFi”s são os *gateways* utilizados para essa tarefa. O nome BluFi remete à sua utilidade de fazer a ponte entre *Bluetooth* e WiFi. Os BluFis ficam conectados em alguma tomada ou ligados através de uma

¹ Disponível em: <<http://bluvision.com/our-products/>>. Acesso em 20 maio 2018.

bateria. Enquanto ligados, os BluFis recebem contínua e repetidamente os pacotes de dados dos *Beacons* através da comunicação *Bluetooth Low Energy*. Depois de receberem esses pacotes, eles o enviam através da sua comunicação WiFi (com algum roteador de internet que eles estejam conectados) até a plataforma Bluzone. O esquema da Figura 2 ilustra esse fluxo de dados.

Figura 2 - Fluxo de dados desde os *beacons* até a plataforma Bluzone.

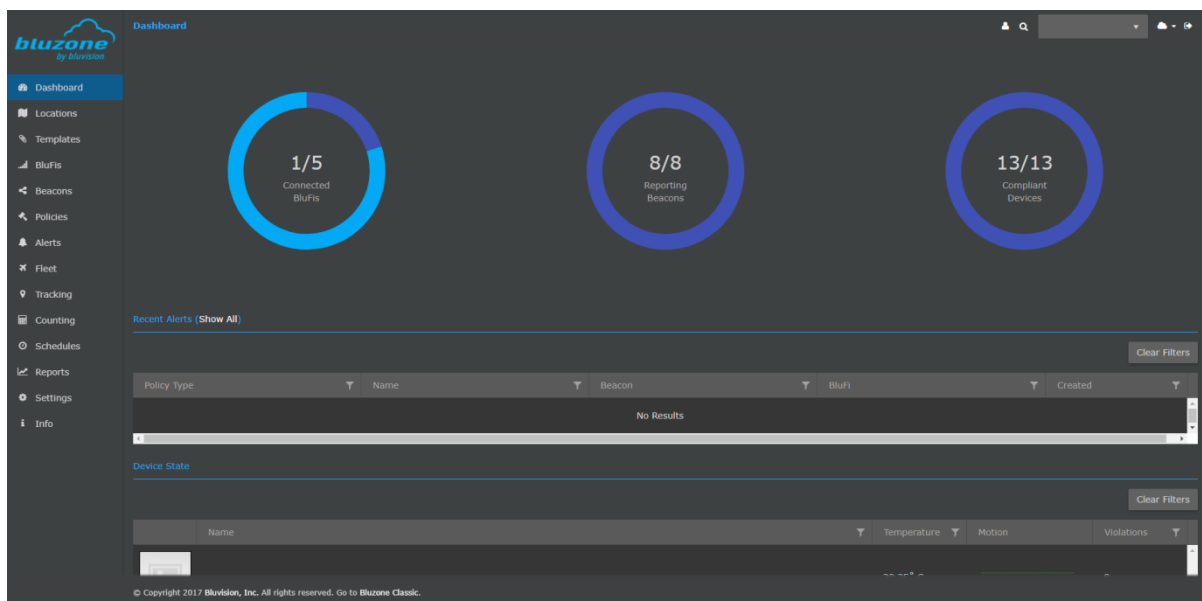


Fonte: elaborada pelo autor.

2.2.3 Bluzone

Por último, tem-se a plataforma *online* da Bluvision chamada de “Bluzone”. Essa plataforma engloba as muitas funcionalidades que podem ser usadas com os *beacons*, como as já antes citadas de rastreabilidade, localização e *condition monitoring*. A Figura 3 mostra a tela inicial de plataforma, acessada pelo link <https://bluzone.io/console/#/dashboard>. Ao entrar na plataforma há um painel de controle, inicial com algumas informações importantes a serem expostas para o usuário, como alertas recentes e estado de alguns dispositivos. À esquerda existe um menu lateral com diversas opções de telas diferentes que podem ser acessadas. Entre essas telas estão as telas de gerenciamento dos *beacons*, de gerenciamento dos BluFis, dos *templates* de configuração, das chamadas “*policies*” (que são regras que os *beacons* devem seguir e se não as seguir há a geração de alertas), tela de alertas e outras telas como as de rastreamento de *beacons*.

Figura 3 - Plataforma Bluzone.



Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

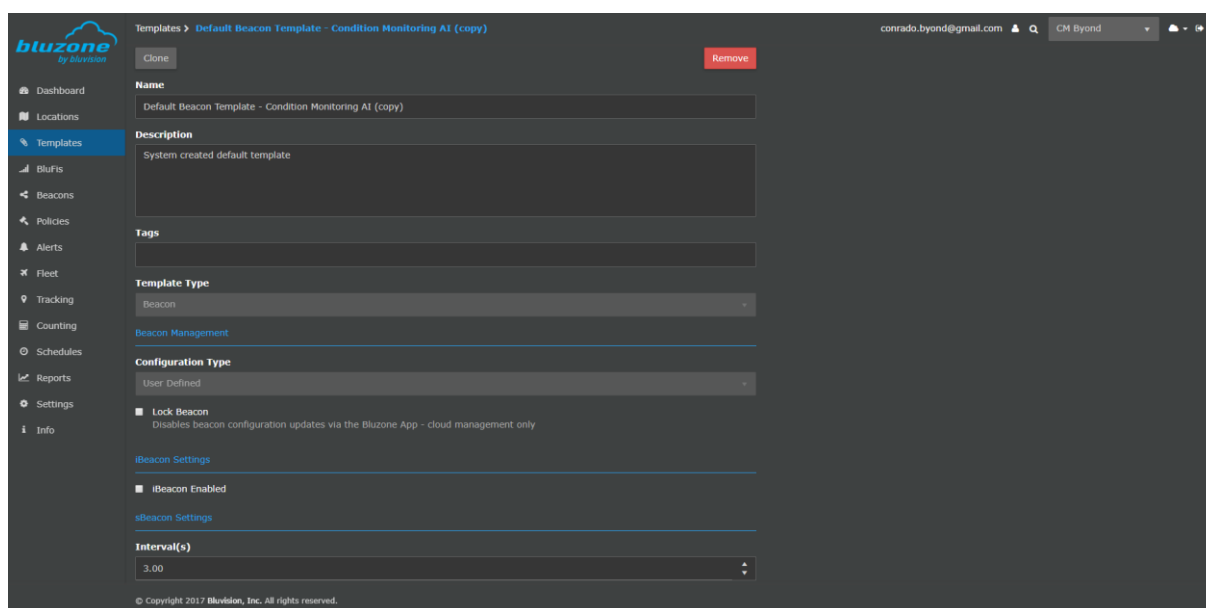
2.3 MÉTODO DE *CONDITION MONITORING* COM OS *BEACONS* BLUVISION

Para utilizar os *beacons* para *condition monitoring*, antes é necessário fazer algumas etapas de configuração e de outros fatores utilizando-se a plataforma online.

Existem duas opções para configuração dos *beacons*: pode-se configurar um modelo (chamado de *template*) que pode ser depois aplicado a inúmeros *beacons*, ou pode-se configurar cada *beacon* individualmente. De ambos os modos, são apresentadas as mesmas telas com as mesmas opções de configurações, portanto será explicado apenas um deles.

A página de configuração de *templates* dos *beacons* é mostrada na Figura 4. Nela é possível dar um nome e uma descrição ao *template* criado, assim como atribuir algumas *tags* de identificação, caso desejado.

Figura 4 – Página de criação de *templates*.



Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

2.3.1 Configuração dos *beacons*

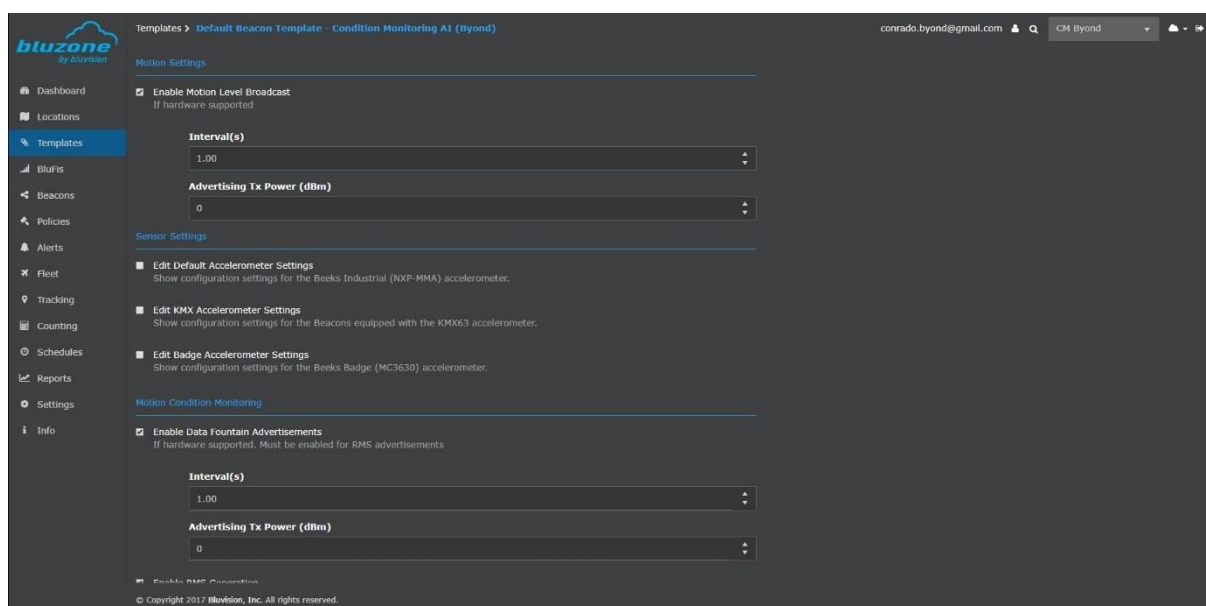
Na parte de tipo de *template* (“*Template type*”) escolhe-se entre *beacon* ou BluFi, neste caso foi selecionado o *beacon*. Além dessas configurações iniciais existem algumas outras para iBeacon ou Eddystone, que não serão necessárias por se tratarem dos protocolos da Apple e do Google, respectivamente.

A seguir, apresenta-se o que deve ser configurado em cada seção e em cada opção, que serão ilustrados na Figura 5 e 6:

2.3.1.1 Motion Settings

- **Enable Motion Level Broadcast:** essa opção, que se refere a envios de dados do *beacon* no modo de baixa velocidade (“*Low Speed*”), deve ser selecionada.
- **Interval(s):** intervalo entre transmissões de dados.
- **Advertising Tx Power (dBm):** potência do sinal de transmissão.

Figura 5 - Página de configuração de um *template* para beacons.



Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

2.3.1.2 Motion Condition monitoring

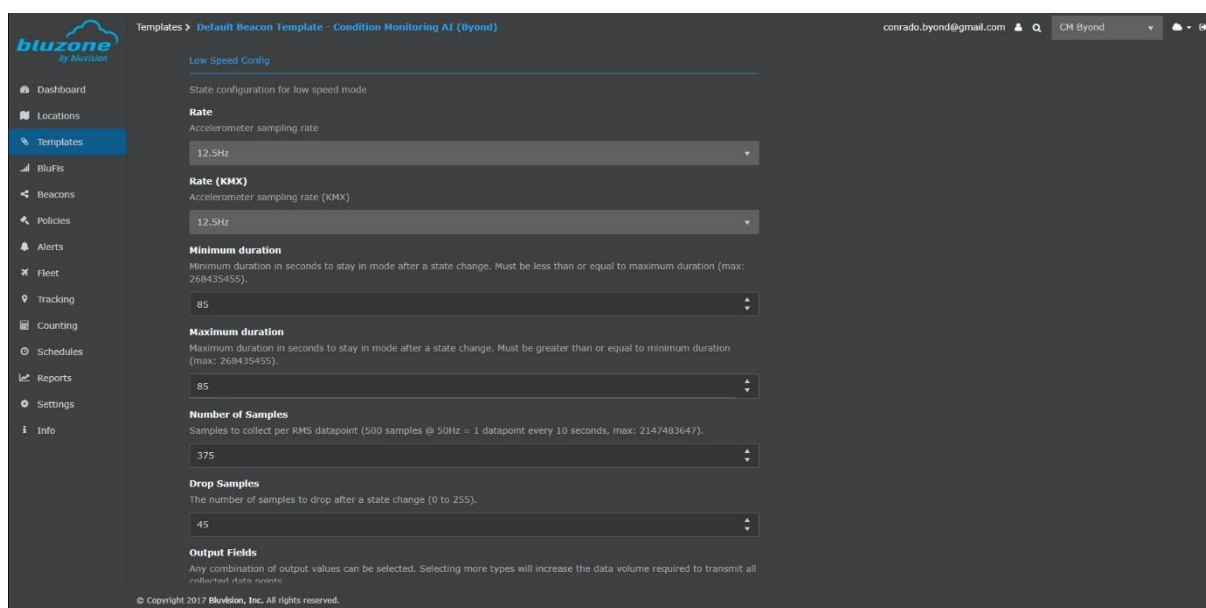
- **Enable Data Fountain Advertisements** também deve ser ativada. Essa opção, ao contrário da anterior, se refere à transferência de pacotes no modo de alta velocidade, onde o *beacon* fornece uma maior taxa de amostragem nos sensores.

Para se certificar que as informações dos *beacons* estão sendo enviadas para a plataforma Bluzone de forma segura e redundante, as amostras individuais são coletadas e agrupadas nos chamados "*Data Fountains*". Esses pacotes são continuamente codificados e transmitidos à plataforma, que os decodifica e extrai as informações.

- **Interval (s)**: intervalo entre transmissões de dados.
- **Advertising Tx Power (dBm)**: potência do sinal de transmissão.
- **Enable RMS Generation**: essa opção também deve ser ativada, pois é necessária para a aplicação de *Condition monitoring*.

Os *beacons* possuem dois modos de operação: "*Low Speed*" (baixa velocidade) e "*High Speed*" (alta velocidade). No modo de alta velocidade, a taxa de amostragem do acelerômetro do *beacon* é maior, podendo chegar a até 800 Hz. O ideal seria manter essa taxa de amostragem alta para obter mais informação a respeito da vibração do motor ou da máquina que está sendo monitorados. Porém, como os *beacons* possuem um intervalo de vida útil que depende da duração da bateria (que não é substituível), o modo de menor taxa de amostragem é usado para economizar bateria e aumentar essa vida útil. Portanto, o *beacon* usa esses dois modos para ambos possuírem períodos com alta taxa de amostragem, que é importante, como para economizar sua bateria, mas continuar recebendo informações de frequências um pouco menores. O *Beacon* fica constantemente alternando entre esses dois modos, dependendo de quanto tempo ele está configurado para permanecer em cada um deles.

Figura 6 - Página de configuração de um *template* para *beacons*.



Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

2.3.1.3 Low Speed Config & High Speed Config

Essas duas seções possuem as mesmas opções de configuração, uma para o modo de "*low speed*" e outra para o de "*high speed*":

- **Rate:** aqui pode-se alterar a taxa de amostragem do acelerômetro quando o *Beacon* está no modo que se está configurando.
- **Minimum duration:** duração mínima de tempo que o *Beacon* vai permanecer no modo.
- **Maximum duration:** duração máxima de tempo que o *Beacon* vai permanecer no modo.

Caso sejam deixados os valores de *minimum duration* e *maximum duration* iguais, o *beacon* permanece exatamente na mesma quantidade de segundos no modo com essa configuração.

- **Number of samples:** número de amostras presentes em cada pacote a ser enviado para a plataforma. Com isso, pode-se também calcular de quanto em quanto tempo cada pacote é enviado (1600 amostras, a uma amostragem de 800 Hz equivale a um pacote a cada 2 minutos).
- **Drop samples:** número das primeiras amostras a não serem usadas ao se mudar o modo de operação do *Beacon* para descartar um período de transição entre modos.
- **Output fields:** nessas opções, pode-se escolher quais informações dos acelerômetros devem ser geradas e enviadas à plataforma. Dentre elas, valores de máximo, mínimo, média e RMS.
- **Comparator Mask e Comparators:** além de se providenciar uma duração mínima e máxima, o modo de operação do *Beacon* pode ser alterado baseado na comparação das saídas do *Beacon* com determinados valores limites. Esses valores e essas comparações podem ser modificados nessas seções.

2.3.1.4 Demais seções

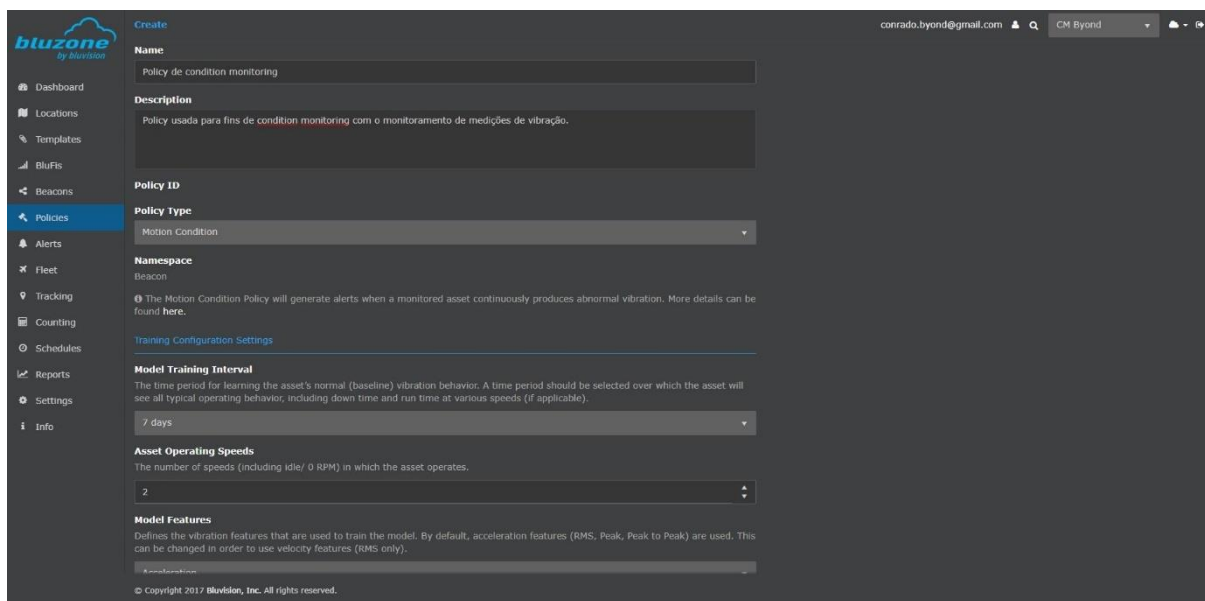
As seções restantes referem-se a configurações que não afetam diretamente a aplicação de *condition monitoring*, com exceção de *Remote Condition monitoring*, que não será explicado neste projeto, que é usado em casos onde não se tem acesso constante à internet no local da instalação dos *beacons*.

2.3.2 Criação da política de *condition monitoring*

Depois de configurados os *beacons*, é necessária a criação de uma política (*policy*). As *policies* são espécies de regras que os *beacons* devem seguir e são relações de muitos para muitos, ou seja, cada *Beacon* pode ser cadastrado em inúmeras *policies*, assim como cada *policy* pode ter vários *beacons* cadastrados. Uma *policy* pode estar relacionada a fatores como a temperatura do *Beacon*, a localização, o movimento, o nível de sua bateria, entre outros. As *policies* mais simples, como as relacionadas à temperatura, verificam os dados do sensor de temperatura do *beacon* e comparam esses dados a alguma faixa de temperatura em que o *beacon* deve se encontrar. Caso esteja a uma temperatura fora dessa faixa, a *policy* gera um alerta que pode ser visualizado pelo usuário diretamente na plataforma ou esse alerta pode desencadear outras ações, através do uso das APIs da Bluzone ou da integração de serviços de *stream* de dados, como o AWS (*Amazon Web Services*) Kinesis.

Apesar das *policies* mais básicas requererem apenas algumas informações, tais como limite superior e inferior de uma certa faixa de operação, a *policy* de *condition monitoring* é um pouco mais complexa, envolvendo análises mais profundas com técnicas de inteligência artificial e possuindo configurações adicionais como a janela de tempo de análise das informações recebidas. A Figura 7 mostra algumas das configurações da *policy* de *condition monitoring*.

Figura 7 - Tela de criação de uma *policy* de *condition monitoring*.

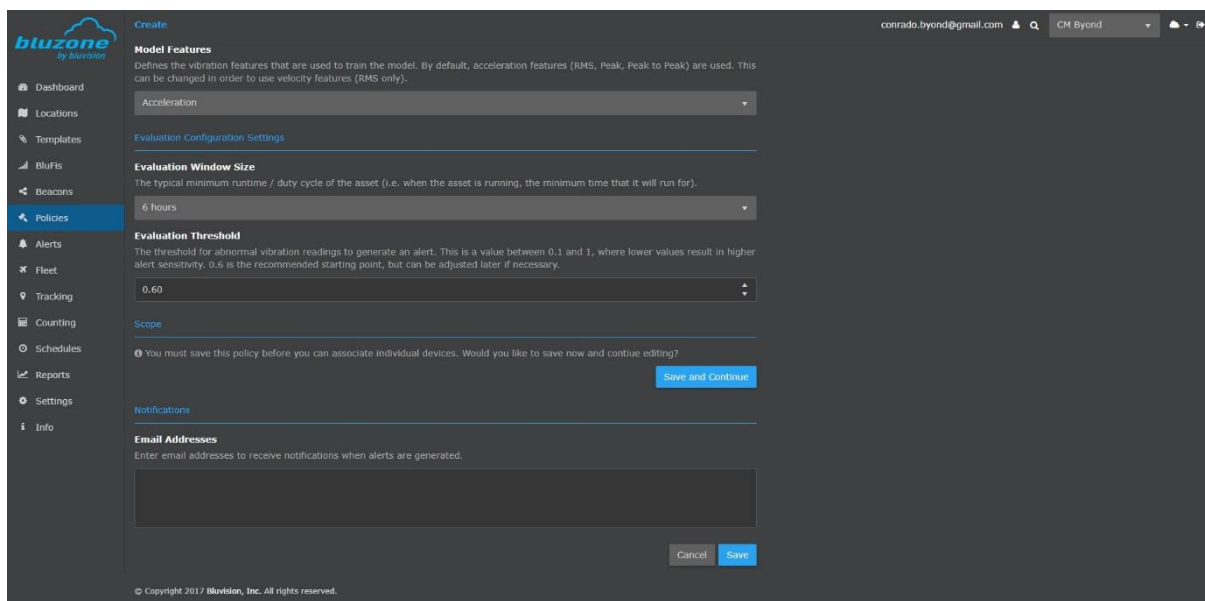


Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

Algumas opções estão relacionadas ao modelo de funcionamento da máquina, como “*Model Training Interval*”, que especifica qual o período de tempo em que a plataforma recolherá os dados de vibração dos *beacons* e os utilizará para gerar um modelo de funcionamento da máquina. Em “*Asset Operating Speeds*” pode-se especificar o número de velocidades de operação da máquina. O repouso conta como uma velocidade de operação, ou seja, se a máquina possui apenas os modos “em repouso” e “em funcionamento”, coloca-se o número 2. Caso a máquina possua mais velocidades de operação, como por exemplo algum processo que requeira diferentes velocidades de rotação de um motor, coloca-se o número total incluindo o repouso.

A Figura 8 ilustra as demais configurações da *policy*. Em “*Model Features*” é possível escolher entre as opções de fazer o *condition monitoring* baseado em dados de aceleração ou de velocidade. Neste projeto, foi utilizada apenas a aceleração, pois o *condition monitoring* baseado em velocidade através da plataforma ainda está em versão beta. Além disso, as informações de aceleração são mais diretamente obtidas pelos acelerômetros do *beacon*.

Figura 8 - Tela de criação das demais configurações da *policy* de *condition monitoring*.



Fonte: captura de tela da plataforma Bluzone usando sistema operacional Windows 8.1 e navegador Google Chrome Versão 67.0.3396.99.

Em “*Evaluation Configuration Settings*” são ajustadas as configurações de como serão avaliados os dados que são constantemente providos pelos *beacons* depois de passado o processo de aprendizado. Em “*Evaluation Window Size*” se especifica a janela de tempo em que as medidas de aceleração deverão ser avaliadas e em “*Evaluation Treshold*” se determina um limite de leitura de medidas anormais, que deverá gerar um alerta. Esse limite é dado como um número entre 0 e 1 e determina uma porcentagem. Desse modo, caso o *Evaluation Treshold* seja 0,5 e haja mais de 50% das medidas de vibração consideradas como “anormais” (significativamente diferentes do modelo de aprendizado) dentro da janela de tempo avaliada, a *policy* gerará um alerta.

2.3.3 *Condition monitoring*

Além de configurar o *Beacon* e a *policy* que serão utilizados na aplicação, é necessário cadastrar os *beacons* que medirão a vibração da máquina. Isso é feito também na página de criação de *policy* mostrada na Figura 8, na seção “*Scope*”. Depois de cadastrar os *beacons* desejados na *policy* e salvar a *policy*, o processo de

aprendizado do modelo de funcionamento normal da máquina já entra em ação automaticamente e dura o tempo determinado anteriormente em *Model Training Interval*. Finalizado esse período de tempo, os *beacons* já entram no modo de *condition monitoring* e passam a constantemente enviar as medidas de vibração para análise.

2.4 VANTAGENS E LIMITAÇÕES DO USO DE *BEACONS*

Uma solução alternativa ao uso dos *beacons* (e a mais tradicional da indústria) é a utilização de máquinas e motores com sensores embutidos. Essa solução possui alguns pontos positivos quando comparada ao uso dos *Beacons* e também alguns pontos negativos.

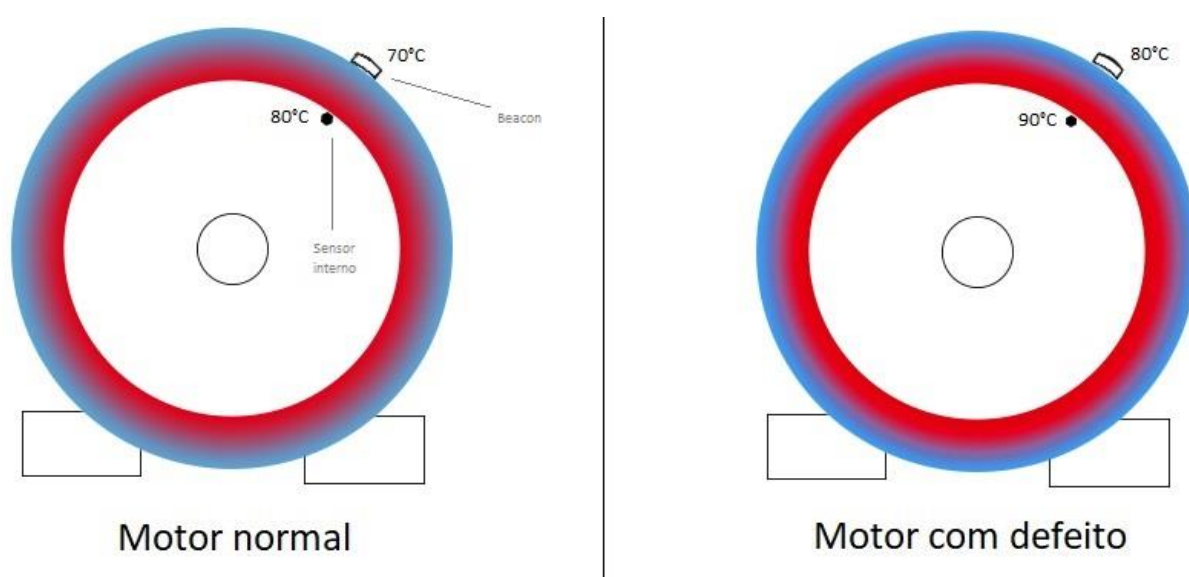
A fabricação de máquinas com sensores requer um projeto mais elaborado de engenharia e também envolve o custo dos próprios sensores, principalmente quando se trata de sensores de boa qualidade (resolução, taxa de aquisição, robustez, etc). Com isso, o custo de tais máquinas e motores costuma ser mais elevado. Essa é a principal vantagem do uso dos *beacons*: por se tratarem de aparelhos de pequeno porte e relativamente simples, eles possuem custo baixo. Além disso, seu uso não adiciona grandes custos de instalação, já que os *beacons* usados neste projeto são facilmente acoplados às máquinas e motores através de cola ou fita dupla face de nível industrial. Considerando-se então o baixo custo comparado a máquinas com sensores integrados, os *beacons* também têm a vantagem de poderem ser comprados e usados em grande escala, aumentando a quantidade de bens sendo monitorados e reduzindo ainda mais a perda com quebras e manutenções. Essa redução no custo para fazer a manutenção preditiva é o grande aspecto positivo da utilização dos *beacons* e foi a principal motivação para este projeto, além da própria vantagem de se fazer a manutenção preditiva em equipamentos.

Não obstante às vantagens do uso dos *beacons*, outras soluções devem também ser consideradas. Assim como mostra a Figura 9, sensores embutidos internamente às máquinas possuem como principal vantagem o fato de estarem mais próximos das fontes das grandezas, sejam elas calor, vibração, ruído ou outras.

Sendo assim, eles medem mais diretamente a grandeza desejada, com duas vantagens principais: a menor atenuação devido à propagação da grandeza através do meio até o local do sensor, bem como a menor sensibilidade a ruídos. A primeira vantagem é notável, principalmente em grandezas como temperatura. Ao ser medida externamente, o calor passa pela carcaça do motor ou por alguma eventual camada de ar e chega com uma menor quantidade de energia ao sensor, que mede uma temperatura menor. A outra vantagem é facilmente notada quando se pensa em sistemas de medição de sons. Estando mais longe da fonte principal de som que se quer medir, o sensor estará mais próximo a outras fontes de sons e a razão entre sinal significativo e ruído pode, em muitos casos, não ser ideal.

Ainda assim, a atenuação da grandeza até o local do sensor não faz muita diferença, a não ser em casos extremos.

Figura 9 - Sensores internos e *beacons* de *Bluetooth*.



Fonte: elaborada pelo autor.

Isso é diretamente concluído do fato de que o valor absoluto da grandeza não é importante normalmente, mas sim a diferença entre o comportamento do valor da grandeza durante funcionamento normal do motor e o seu comportamento durante funcionamento defeituoso. Essa consideração também é válida durante a escolha do local de instalação do *beacon*. Se o *beacon* for instalado o mais perto possível da fonte de grandeza, seu exato local ou orientação não é importante. Usando as técnicas

de *condition monitoring* apropriadas, pode-se aprender os padrões habituais de funcionamento da máquina e eventuais desvios desses padrões já são suficientes para se inferir um comportamento anormal e, dependendo do caso, concluir qual problema específico está acontecendo no momento.

3 PROTÓTIPO E TESTES

Com o objetivo de se fazer uma espécie de prova de conceito e validar a possibilidade de futura aplicação em escala industrial, foi feito um protótipo para testes dentro da empresa Byond. Esse protótipo foi projetado para que, de forma simples, mostre o funcionamento dos *beacons* em aplicação de *condition monitoring*.

3.1 CONFIGURAÇÃO INICIAL

Algumas opções de motores de pequeno porte foram exploradas e, por fim, utilizou-se um aquecedor portátil como máquina a ser testada. O aquecedor é composto de um ventilador e de um sistema de aquecimento de ar por resistência elétrica. Para o nosso protótipo, o sistema de aquecimento foi desligado e o foco foi apenas no movimento rotacional do ventilador e seu comportamento de vibração. A Figura 10 mostra uma imagem do aquecedor utilizado.

Figura 10 - Aquecedor termoventilador A1 da marca Ventisol utilizado no estudo.



Fonte: Manual do consumidor - aquecedor de ambientes².

² **Manual do consumidor:** Aquecedor de ambientes. [S.l.: s.n.], [201-]. 1 p. Disponível em: <http://www.ventisol.com.br/fmanager/ventisol/produtos/arquivo_manual157_1.pdf>. Acesso em: 01 maio 2018.

O aquecedor teve sua proteção frontal aberta para possibilitar modificações necessárias ao teste. Com isso, além de desabilitar o sistema de aquecimento, abriu-se a possibilidade de serem feitas modificações no motor ou nas hélices do ventilador, de modo a simular uma falha em seu funcionamento. A Figura 11 mostra a configuração feita. Na imagem é possível ver pedaços de fita adesiva que foram utilizados para acoplar pesos nas pás da hélice, com o objetivo de forçar um desequilíbrio nelas e mudar seu padrão de vibração.

Figura 11 - Aquecedor com provocado desequilíbrio para alteração do padrão de vibração.



Fonte: Elaborada pelo autor.

Dois *beacons* modelo “BEEKs Industrial VER 1.6” foram instalados na parte traseira com fita dupla face e, com isso, pode-se começar os testes. O modelo de *beacon* utilizado é ilustrado na Figura 12.

Figura 12 - *Beacon* modelo BEEKs industrial utilizado no estudo.



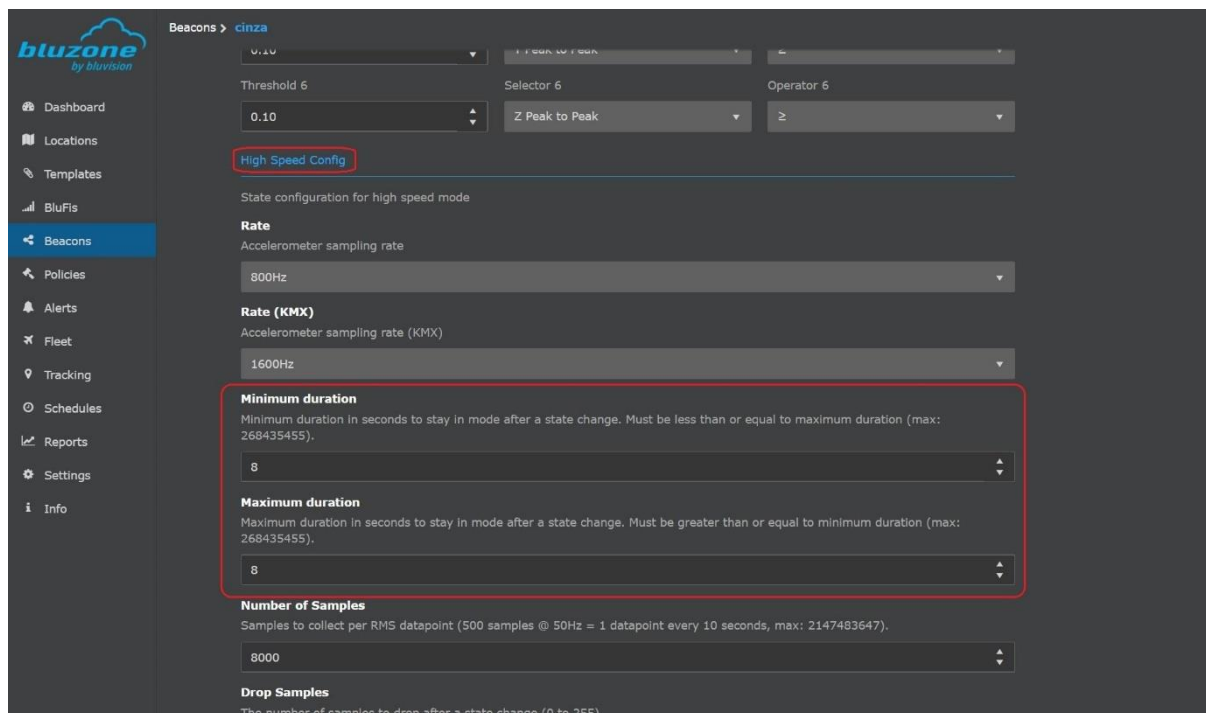
Fonte: datasheet do *Beacon*³.

3.2 PERÍODO DE APRENDIZADO DOS BEACONS

Depois de instalados, os *beacons* passam por um processo de aprendizado onde vão sendo armazenados os padrões usuais de comportamento de vibração para depois serem comparados com as novas vibrações medidas com o objetivo de classificar o comportamento da máquina em “normal” ou “falho”. Inicialmente as primeiras tentativas de aprendizado não foram bem-sucedidas, devido a configurações não bem dimensionadas dos *beacons*. Como explicado na parte de configuração dos *beacons* (capítulo 2), os *beacons* possuem dois modos de funcionamento, um modo chamado de “modo de alta velocidade” e outro chamado de “modo de baixa velocidade”. O modo de alta velocidade possui uma taxa de amostragem maior (de até 800 Hz) e consegue, portanto, captar componentes de vibração de mais alta frequência. A primeira configuração de aprendizado usou o padrão utilizado para nível industrial, em que o período de aprendizado é de aproximadamente uma semana de treinamento. Contudo os testes feitos na Byond foram em menores escalas temporais e o *beacon* não estava configurado para permanecer tempo necessário no modo de alta velocidade para gerar dados suficientes para o aprendizado. Depois de um ajuste nas configurações, aumentando a duração do período de tempo do modo de alta velocidade, pode-se fazer o aprendizado com sucesso em um período de 24 horas. A Figura 13 mostra a tela na qual se pode modificar essas configurações.

³ **SPECIFICATION sheet - BEEKs industrial ver 1.6.** [S.l.: s.n.], 2017. 1 p. Disponível em: <http://bluvision.com/wp-content/uploads/2017/09/Specs-BEEKs-Industrial_1.pdf>. Acesso em: 13 jun. 2018.

Figura 13 - Tela de configurações do *beacon* com período de duração do modo de alta velocidade destacado em vermelho.



Fonte: captura de tela da plataforma Bluzone com sistema operacional Windows 10 e navegador Google Chrome Versão 67.0.3396.99.

Na tela de configuração dos *beacons* e na de criação dos *templates* para os *beacons* pode-se fazer algumas outras configurações que não entram muito no escopo deste projeto, mas que foram ajustadas para a aplicação no teste.

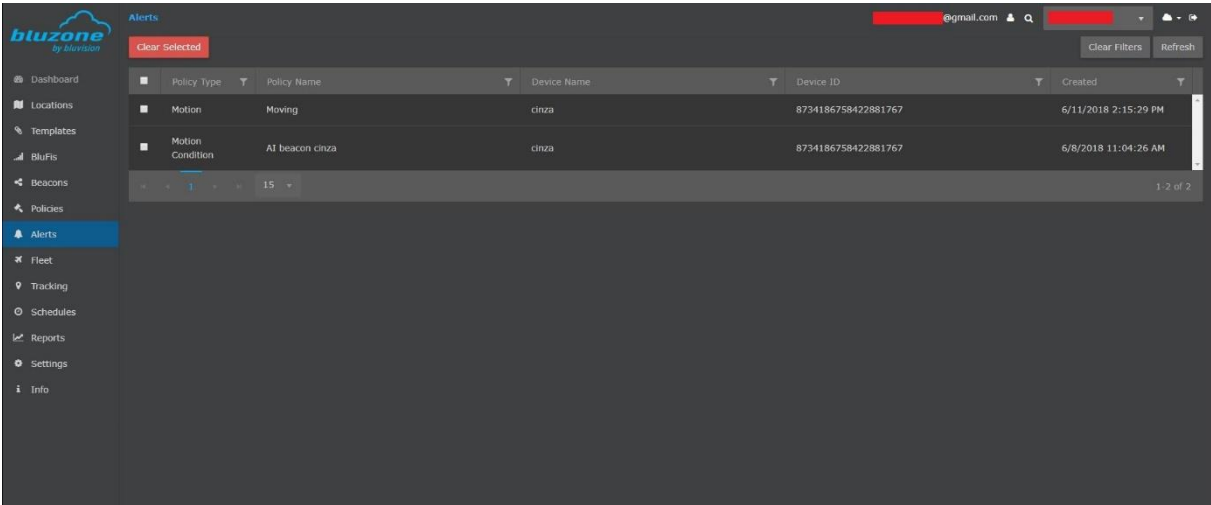
O método de aprendizado e de armazenamento dessas informações é feito, de acordo com a empresa Bluvision, usando técnicas de *machine learning*, que significa “aprendizado de máquina” e é uma área de estudo que se situa dentro do campo de inteligência artificial. Os algoritmos e técnicas usadas são de propriedade da Bluvision e não são explicitados nas documentações oficiais. Deste modo, o algoritmo não será analisado em detalhes.

3.3 TESTES E RESULTADOS

Como explicado na seção 2.3.2, é necessário configurar também a *policy* de *condition monitoring*. Por se tratar de testes de escala de tempo reduzida, a janela de avaliação das medidas de vibração foi de 10 minutos e o limite de leituras anormais foi de 0,3. Isso significa que dentro dos últimos 10 minutos de medidas de vibração recebidas pela plataforma, se 30% dos dados fossem considerados de comportamento anormal comparado com os padrões aprendidos, seria gerado um alerta.

Após essas configurações, os testes estavam aptos a serem realizados. No começo deixou-se o ventilador funcionando de forma habitual para validar o aprendizado e verificar que a plataforma reconheceria esse comportamento como o esperado. Não houve a geração de nenhum alerta e pode-se concluir que a plataforma de fato classificou corretamente o funcionamento durante esse período. Em seguida, foi feito o desequilíbrio nas pás da hélice do ventilador com alguns pesos. Os locais de fixação, mais precisamente a distância entre o local de fixação e o eixo de rotação, foram escolhidos após alguns testes de modo que o padrão de rotação da hélice fosse significativamente diferente do normal sem colocar em risco a estrutura física do ventilador ou do motor. Depois da fixação dos pesos usados para simular um defeito, também foi feita uma leve inclinação do ventilador para gerar diferentes valores de componente da aceleração da gravidade nos 3 eixos. Ele então foi colocado em funcionamento durante dezenas de minutos para que a plataforma detectasse o comportamento anormal. O teste foi repetido algumas vezes e, em todas as ocasiões, a plataforma detectou comportamento anormal e gerou os alertas como desejado. A Figura 14 ilustra um desses alertas.

Figura 14 – Exemplo de alerta gerado pela *policy* de *condition monitoring* através da plataforma Bluzone.



The screenshot shows the Bluzone Alerts interface. The left sidebar contains navigation options: Dashboard, Locations, Templates, BluFis, Beacons, Policies, Alerts (selected), Fleet, Tracking, Schedules, Reports, Settings, and Info. The main area displays a table of alerts with the following data:

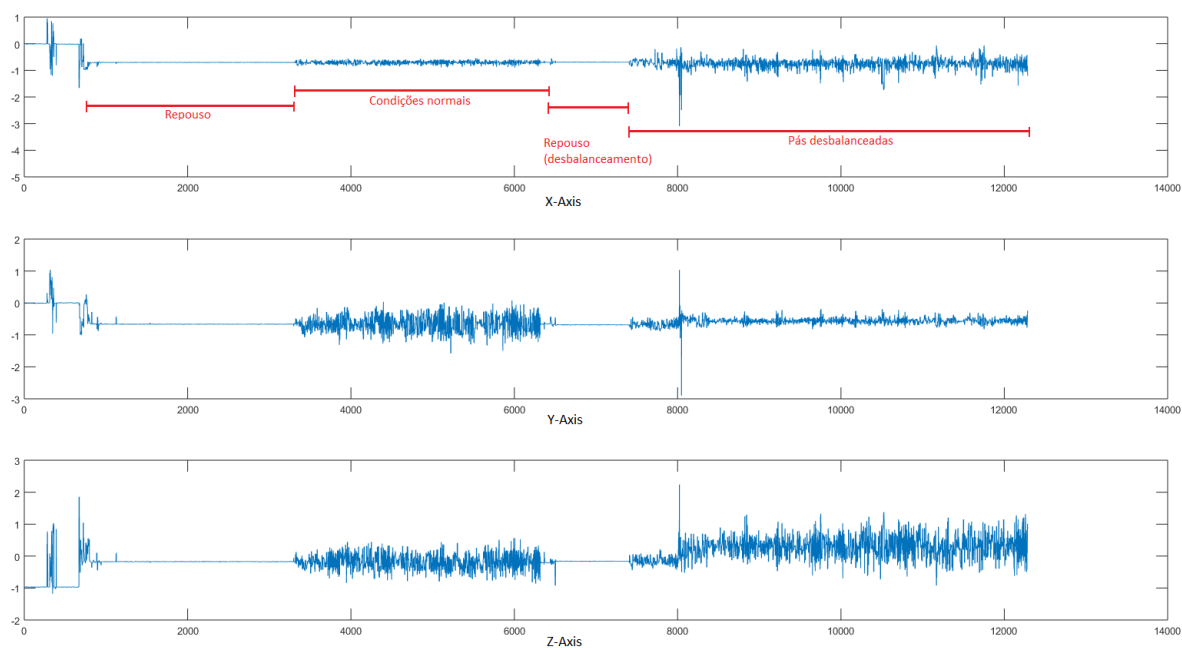
Policy Type	Policy Name	Device Name	Device ID	Created
Motion	Moving	cinza	8734186758422881767	6/11/2018 2:15:29 PM
Motion Condition	At beacon cinza	cinza	8734186758422881767	6/8/2018 11:04:26 AM

Fonte: captura de tela da plataforma Bluzone com sistema operacional Windows 10 e navegador Google Chrome Versão 67.0.3396.99.

Com isso, pôde-se validar o funcionamento dos *beacons* para aplicação de *condition monitoring*, pelo menos em escala reduzida e em ambiente controlado. Assim é possível justificar testes em escala industrial e adoção dessa tecnologia para futuras soluções com clientes da Byond.

Além dos resultados através da geração de alertas mostrando que o ventilador possuía um “defeito”, também foram exportados os dados crus do acelerômetro para a análise no Matlab. A Figura 15 ilustra os valores de aceleração ao longo do tempo com a divisão entre um período de funcionamento normal do ventilador e o funcionamento após o desequilíbrio das pás. O valor mostrado no eixo Y está relativo à aceleração da gravidade na Terra (g), que é de cerca de $9,8 \text{ m/s}^2$, ou seja, o valor 1 no gráfico corresponde a $1 g \approx 9,8 \text{ m/s}^2$.

Figura 15 - Valores de aceleração nos 3 eixos do *beacon* acoplado ao ventilador durante testes no ambiente controlado.



Fonte: Figura gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Os períodos de tempo estão divididos em:

- Repouso – período em que o ventilador se encontrava em repouso e desligado.
- Condições normais – período em que o ventilador estava em funcionamento normal.
- Repouso (desbalanceamento) – período em que estava sendo realizado o acoplamento dos pesos para desequilíbrio das pás.
- Pás desbalanceadas – período em que o ventilador estava funcionando de forma desequilibrada, simulando uma situação de defeito.

Para a geração do gráfico, os dados de um arquivo de formato *.csv* (*comma-separated values*) foi importado à área de trabalho do Matlab código 1 foi executado (ver apêndice).

4 ALGORITMOS DE *CONDITION MONITORING*

Em sequência do estudo da tecnologia dos *beacons* e suas funcionalidades e da sua validação em testes em ambiente controlado, foi feito um estudo da possibilidade de implantação de diferentes algoritmos de *condition monitoring* nos dados obtidos pelos sensores.

4.1 ALGORITMOS EXISTENTES

Dentre as técnicas já criadas e utilizadas para manutenção preventiva através de *condition monitoring* pode-se citar os métodos baseados em frequências naturais, métodos baseados em forma modal, métodos baseados em energia de tensão modal, métodos estatísticos [11], regressão linear múltipla, modelos de mistura Gaussiana, aprendizagem bayesiana esparsa [12] e modelos baseados em técnicas de inteligência artificial, como redes neurais, *fuzzy logic*, algoritmos evolucionários e SVM (*support vector machine*) [13]. Esses métodos se agrupam em algumas categorias, sendo alguns algoritmos utilizados no domínio do tempo, alguns no domínio da frequência e outros com outros tipos de análise, tais como por transformação de *wavelet* ou análise modal.

Inicialmente tinha sido considerada a implementação de técnicas de inteligência artificial baseadas em redes neurais artificiais, mais precisamente usando-se o modelo de perceptron de múltiplas camadas. Para a utilização correta dessa técnica, é necessário o treinamento da rede neural com diversos exemplos de funcionamento da máquina. Com os testes realizados no protótipo construído, foi possível gerar uma quantidade razoável de dados, entretanto todos os dados vêm do funcionamento normal do ventilador ou do funcionamento com as pás desequilibradas. Com isso a rede neural teria a capacidade de classificar futuras amostras entre essas duas opções baseado em qual se encaixaria melhor. Caso houvesse algum defeito diferente desse provido como exemplo nos treinamentos, a rede neural não seria capaz de classificá-lo corretamente, o que limitaria a capacidade dessa técnica. Esse problema poderia ser resolvido se existisse uma grande quantidade de exemplos de

diferentes tipos de defeitos para que a rede neural pudesse aprender todos e fazer com sucesso, não só uma manutenção preditiva como uma manutenção prescritiva. Porém, isso requer uma grande quantidade de exemplos de medições do ventilador, desenvolvendo diferentes tipos de falhas, o que não é possível na prática.

Como consequência, foram testados outros algoritmos já utilizados na literatura, um deles no domínio do tempo e outro no domínio da frequência. Com isso pôde-se verificar a possibilidade da utilização de alguns algoritmos de *condition monitoring* já conhecidos nos dados de vibração dos sensores.

4.2 DOMÍNIO DO TEMPO

No domínio do tempo, o principal método utilizado é a análise de algumas características estatísticas do sinal no tempo. A obtenção dessas características do sinal é chamada de extração de característica ou extração de *features* (do inglês “*feature selection*”).

Alguns dos parâmetros mais popularmente utilizados são: média do sinal, RMS, variância, coeficiente de assimetria, curtose, fator de crista, entropia e valor de pico [14], [15] e [16].

Na Tabela 1 estão ilustrados alguns desses parâmetros, sua significância e como são calculados.

Tabela 1 - Algumas *features* comumente utilizadas no método no domínio do tempo.

Feature Name	Description	
	Brief Definition	Formula
RMS	The RMS value increase gradually as fault developed. However, RMS is unable to provide the information of incipient fault stage while it increases with the fault development [11].	$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$
Variance	Variance measures the dispersion of a signal around their reference mean value.	$Var = \frac{\sum_{i=1}^N (x_i - m)^2}{(N-1)\sigma^2}$
Skewness	Skewness quantifies the asymmetry behavior of vibration signal through its probability density function (PDF).	$Sk = \frac{\sum_{i=1}^N (x_i - m)^3}{(N-1)\sigma^3}$
Kurtosis	Kurtosis quantifies the peak value of the PDF. The kurtosis value for normal rolling element bearing is well-recognized as 3.	$Ku = \frac{\sum_{i=1}^N (x_i - m)^4}{(N-1)\sigma^4}$
Shape factor	Shape factor is a value that is affected by an object's shape but is independent of its dimensions [12].	$SF = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}{\frac{1}{N} \sum_{i=1}^N x_i }$
Crest factor	Crest factor (CF) calculates how much impact occur during the rolling element and raceway contact. CF is appropriate for "spiky signals" [12].	$CF = \frac{\max x_i }{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}$
Entropy	Entropy, $e(p)$, is a calculation of the uncertainty and randomness of a sampled vibration data. Given a set of probabilities, (p_1, p_2, \dots, p_n) , the entropy can be calculated using the formulas as shown in the right column.	$e(p) = - \sum_{i=1}^n p(z_i) \log_2 p(z_i)$

Fonte: Caesarendra e Tjahjowidodo (2017) [14].

Dentre esses parâmetros, os utilizados na análise no domínio do tempo são: média do sinal, valor de pico, valor RMS, variância e coeficiente de assimetria. Essas características mencionadas, que serão explicadas a seguir, conseguem descrever em diversos níveis o sinal da grandeza em função do tempo e já são suficientes para reconhecimento de alguma falha caso seus valores se diferenciem do normal.

- **Média do sinal** – esse parâmetro é simplesmente a média algébrica das medidas de vibração dentro de um certo período de tempo. É calculado pela seguinte fórmula:

$$m = \frac{1}{N} \sum_{i=1}^N x_i$$

Em que m é a média, N é o número de amostras e x_i é uma amostra de índice i .

- **Valor de pico** – indica qual o máximo valor que aconteceu em um certo período. Pode indicar se o nível de vibração está atingindo novos patamares ao longo do tempo, o que pode indicar uma falha. É simplesmente dado por:

$$V_p = \max(x_i)$$

Em que V_p é o valor de pico e x_i é a amostra de índice i .

- **RMS** – é uma forma de indicar a potência de um sinal. Neste caso, indica a intensidade da vibração. Calculado por:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

Em que N e x_i são o número de amostras e a amostra do índice i respectivamente.

- **Variância** – mede o quanto o sinal tende a se distanciar do seu valor de média. Calcula-se pela seguinte fórmula:

$$V = \frac{1}{N-1} \sum_{i=1}^N |x_i - m|^2$$

Sendo N é o número de amostras, x_i é uma amostra de índice i e m é a média do sinal.

- **Coeficiente de assimetria (C_a)** - é uma forma de se quantificar a assimetria do sinal. O valor é resultante da seguinte fórmula:

$$C_a = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - m)^3}{\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - m)^2}^3}$$

Estipulados os parâmetros a serem usados como forma de análise, criou-se um programa no *software* Matlab para importação dos valores de vibração e extração dessas características. O código deste programa encontra-se no apêndice referenciado por “Código 2”.

O resultado da execução desse código é o gráfico das medições de vibração no período de tempo analisado e a exibição de todas as características mencionadas para esse período.

4.3 DOMÍNIO DA FREQUÊNCIA

Feito o algoritmo e a análise dos sinais de vibração no tempo, agora passa-se a trabalhar no domínio da frequência. A principal forma de análise de um sinal na frequência é através de sua transformada de Fourier. A transformada de Fourier de um sinal no tempo leva ao domínio da frequência e traz a informação de todas as suas componentes senoidais, assim como as respectivas amplitudes e fases. A definição da transformada de Fourier para um sinal em tempo contínuo é dada por:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{j\omega t} dt \quad (1)$$

Em que a variável t representa o tempo e a variável ω representa a frequência. Para se trabalhar em um computador é necessário fazer a amostragem do sinal e trabalhar com dados discretos. Com isso, utiliza-se a transformada discreta de Fourier para fazer a análise na frequência, definida como:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad (2)$$

$$k = 0, \dots, N - 1$$

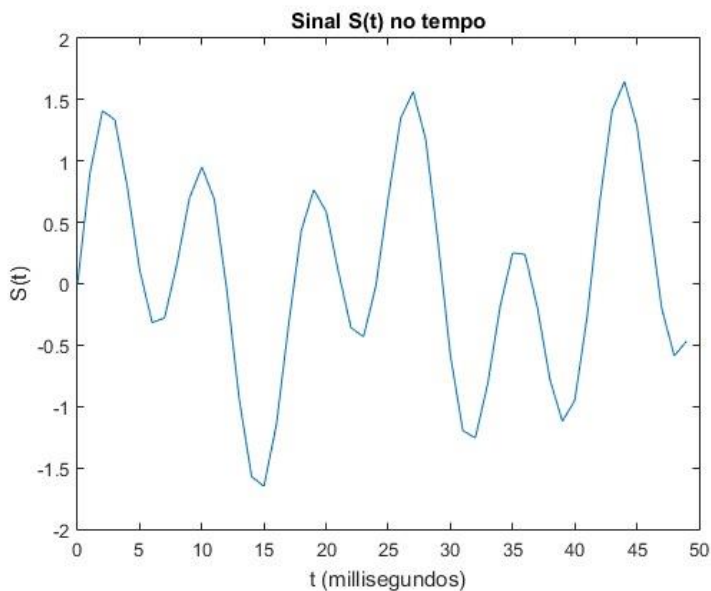
Com o uso da transformada discreta de Fourier, decomparamos o sinal em suas diferentes componentes e podemos analisar quais frequências são mais proeminentes no sinal. A evolução de defeitos em uma máquina é normalmente acompanhada de uma mudança no espectro de frequência de sua vibração, ampliando algumas frequências e reduzindo outras. Com isso, a análise de frequência é importante para o processo de manutenção preditiva via *condition monitoring*.

A implantação da transformada discreta de Fourier foi feita em um programa de Matlab e com um algoritmo chamado transformada rápida de Fourier (FFT - do

inglês *Fast Fourier Transform*). Esse algoritmo gera exatamente os mesmos resultados do que o cálculo direto pela definição da transformada discreta de Fourier da equação (2), com a vantagem de ser mais eficiente e, portanto, calculado mais rapidamente por computadores. Pela sua eficiência, a aplicação da transformada discreta de Fourier através da FFT é muito popularmente utilizada. Pode-se ver na Figura 16 e Figura 17, respectivamente, o gráfico no tempo e o espectro de amplitude na frequência da função exemplo especificada a seguir:

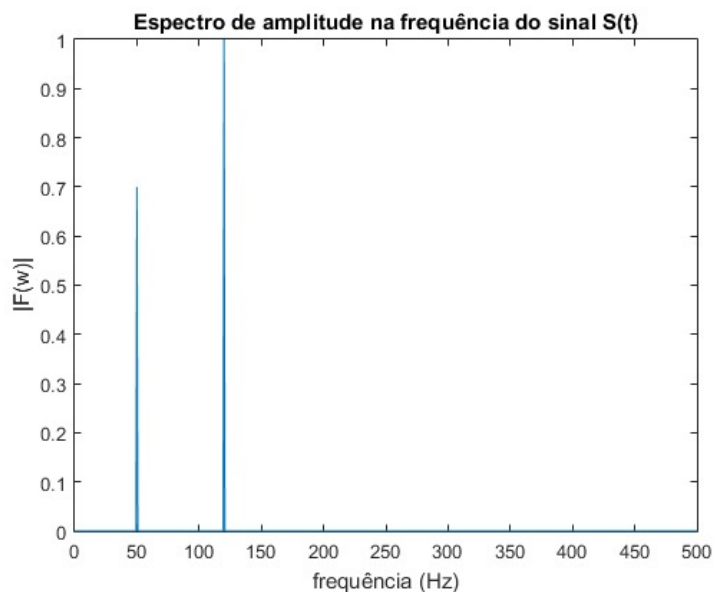
$$S(t) = 0.7 \sin(2\pi * 50 * t) + \sin(2\pi * 120 * t)$$

Figura 16 - Gráfico do sinal S(t) no tempo.



Fonte: Figura gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 17 - Espectro de frequência do sinal.



Fonte: Figura gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Na Figura 17 ficam claros os picos na frequência 50 Hz e 120 Hz, sendo as frequências das senoides usadas para gerar esse sinal exemplo.

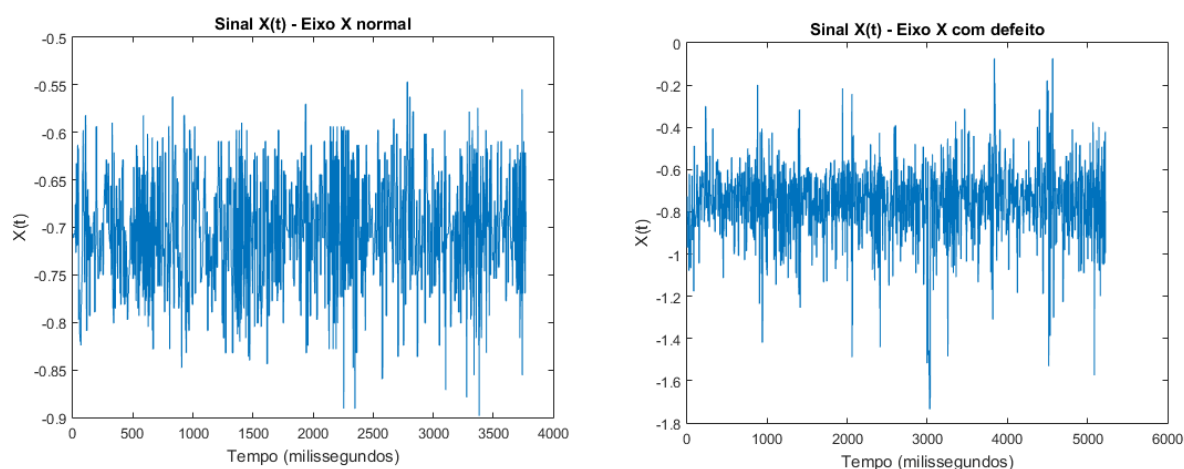
O código 3 do apêndice mostra a implementação do programa em Matlab usado para analisar os sinais na frequência.

4.4 IMPLEMENTAÇÃO DOS ALGORITMOS NO PROTÓTIPO

Tendo os algoritmos prontos e implementados, pôde-se fazer a primeira etapa, que é utilizá-los nos dados do ventilador e mostrar a análise no tempo e na frequência para o protótipo, tanto em funcionamento normal quanto para o funcionamento com as pás da hélice desbalanceadas.

Na sequência, os dados são exportados da plataforma Bluzone, em formato .csv, e importados à área de trabalho do Matlab. Depois disso, são salvos em variáveis e usados na execução do programa. Os resultados dos sinais, no tempo, encontram-se apresentados na Figura 18.

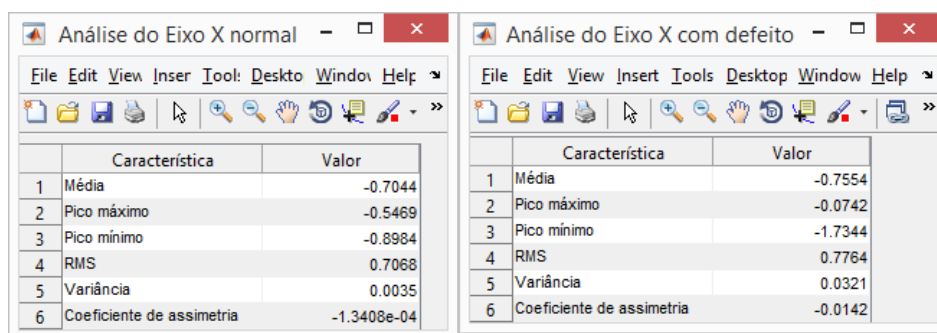
Figura 18 – Sinais no domínio do tempo para o ventilador normal e para o ventilador com defeito.



Fonte: Figura gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Na Figura 18 já pode-se notar uma diferença entre os gráficos do sinal no tempo. Relembrando que o valor 1 na vertical do gráfico se refere a $1\text{ g} \approx 9,8\text{ m/s}^2$. Ao se calcular as *features* do sinal, temos os resultados apresentados na Figura 19.

Figura 19 - Características estatísticas do sinal no eixo X, na situação normal e com defeito.

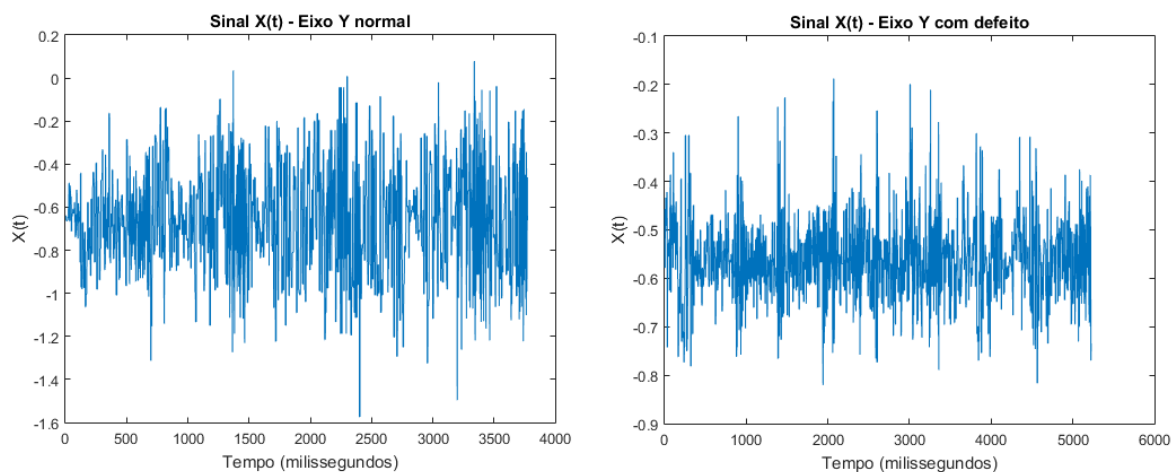


Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

A média do sinal teve um aumento em módulo de cerca de 7% no sentido negativo, o que é uma quantidade considerável. Isso indica que, no geral, os valores de vibração tiveram um deslocamento no gráfico para baixo, com um certo aumento da aceleração no sentido negativo. Ao se analisarem os valores de pico do sinal, nota-se que houve um significativo aumento nos picos de aceleração ambos no sentido negativo como positivo, mostrando uma faixa maior de valores de aceleração, sendo

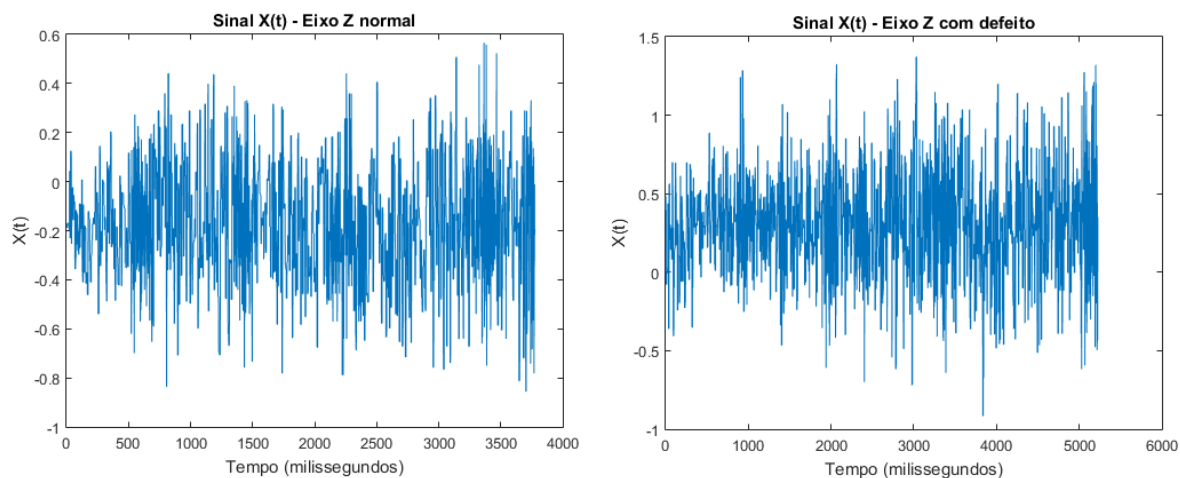
que o pico máximo passou de -0,5469 para -0,0742 e o pico mínimo passou de -0,8984 para -1,7344. O valor RMS do sinal teve um aumento de cerca de 10%, o que indica maior intensidade geral na vibração do protótipo. A variância também teve um aumento significativo, de cerca de 917%. Esse valor em porcentagem ficou muito elevado devido à proximidade ao número 0 do valor da variância em funcionamento normal. De qualquer modo, essa alteração nos indica que houve uma maior inconsistência nos valores de vibração, existindo uma quantidade maior de picos distantes do valor médio de vibração. Por último, o coeficiente de assimetria teve um aumento em módulo também indicando uma assimetria maior do sinal em relação à sua média. O aumento foi no sentido negativo, podendo se concluir que houve um aumento na aceleração da máquina no sentido negativo do eixo X maior do que houve um aumento na aceleração da máquina no sentido positivo. As Figura 20 e Figura 21 mostram, lado a lado, os gráficos, em situação normal e com defeito, no tempo do eixo Y e Z, respectivamente. Em sequência nas Figura 22 e Figura 23 pode-se visualizar a diferença entre os parâmetros do eixo Y e eixo Z.

Figura 20 - Sinal no tempo para o eixo Y da situação normal e da com defeito.



Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 21 - Sinal no tempo para o eixo Z da situação normal e da com defeito.



Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 22 - Características estatísticas do sinal no eixo Y da situação normal e da com defeito.

Análise do Eixo Y normal		Análise do Eixo Y com defeito	
Característica	Valor	Característica	Valor
1 Média	-0.6592	1 Média	-0.5620
2 Pico máximo	0.0781	2 Pico máximo	-0.1875
3 Pico mínimo	-1.5742	3 Pico mínimo	-0.8203
4 RMS	0.7053	4 RMS	0.5682
5 Variância	0.0628	5 Variância	0.0071
6 Coeficiente de assimetria	-3.7064e-05	6 Coeficiente de assimetria	0.0014

Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 23 - Características estatísticas do sinal no eixo Z da situação normal e da com defeito.

Análise do Eixo Z normal		Análise do Eixo Z com defeito	
Característica	Valor	Característica	Valor
1 Média	-0.1760	1 Média	0.3039
2 Pico máximo	0.5664	2 Pico máximo	1.3711
3 Pico mínimo	-0.8555	3 Pico mínimo	-0.9141
4 RMS	0.3001	4 RMS	0.4549
5 Variância	0.0591	5 Variância	0.1146
6 Coeficiente de assimetria	-9.3488e-04	6 Coeficiente de assimetria	0.0044

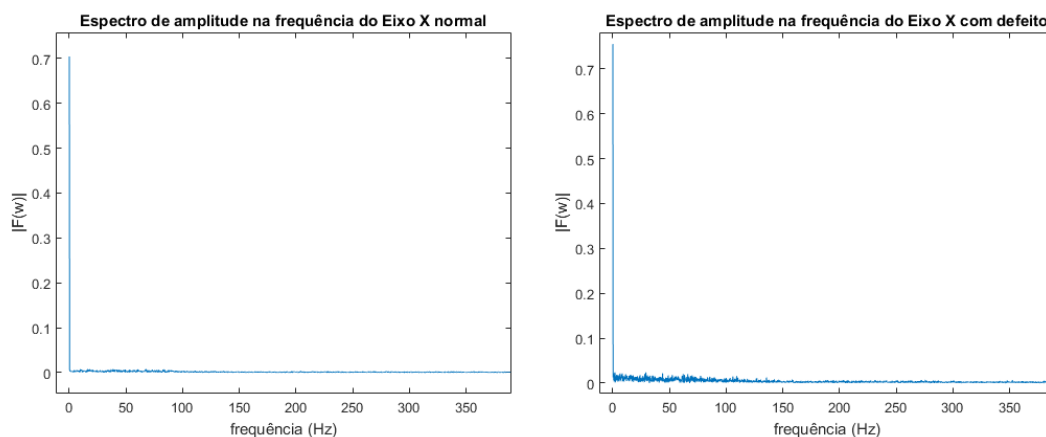
Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

As análises para o eixo Y e Z não são feitas detalhadamente, pois são muito semelhantes à análise feita para o eixo X. O importante de se notar é apenas que também houve uma diferença considerável nos parâmetros escolhidos para análise nesses dois eixos.

Para se concluir que a diferença nesses valores foi significativa o suficiente para indicar uma falha no funcionamento da máquina, é necessário fazer inúmeros testes, tanto em funcionamento normal quanto com diferentes defeitos. A partir daí, pode se traçar um limite em porcentagem ou em valor absoluto cujo funcionamento dentro desse limite é considerado normal e, além desse limite, é um bom indicativo de que há uma falha se desenvolvendo. Neste caso não se tem muitos dados de diferentes testes e falhas, porém uma diferença de 5% a 10% de alguns desses valores já é uma diferença considerável dos valores originais e uma diferença desse tamanho na maioria dos casos já é um indicativo de que o funcionamento está além de uma faixa normal de variação e representativo de um funcionamento defeituoso. Ademais, a visualização dos gráficos no tempo auxilia, empiricamente, à chegada dessa conclusão. Por último, é importante lembrar que o objetivo aqui não é encontrar esses limites e sim mostrar que com a utilização dos *beacons* de *Bluetooth* e os algoritmos usados é possível fazer isso, o que foi demonstrado.

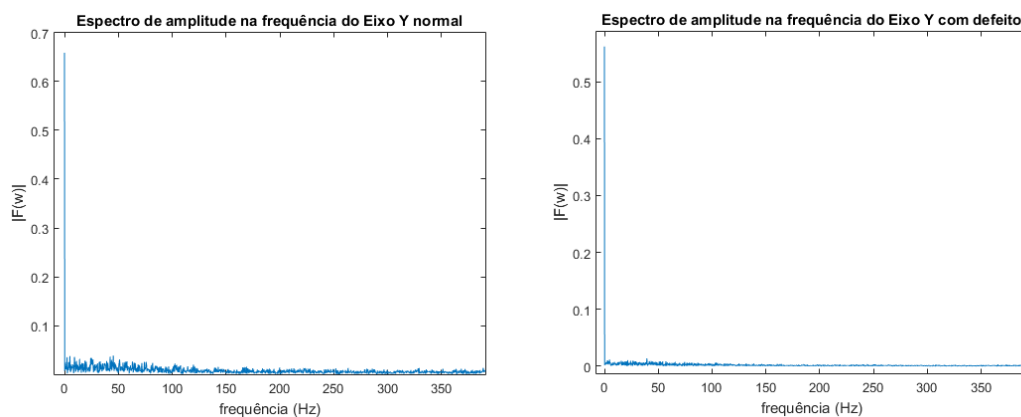
Passando-se para a análise na frequência, os seguintes resultados foram gerados depois de executado o programa no Matlab, apresentados nas Figuras 24, 25 e 26.

Figura 24 - Espectros de frequência do eixo X em situação normal e com defeito.



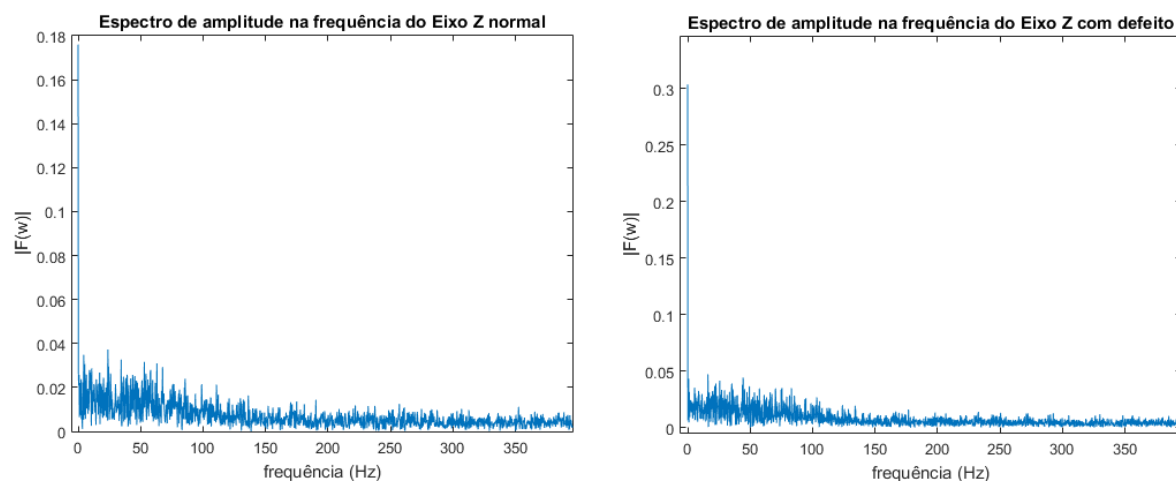
Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 25 - Espectros de frequência do eixo Y em situação normal e com defeito.



Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Figura 26 - Espectros de frequência do eixo Z em situação normal e com defeito.



Fonte: Figuras gerada pelo software Matlab versão R2015a com sistema operacional Windows 8.1 Pro.

Começando a análise pelo eixo X, tem-se que a componente de maior frequência do sinal em funcionamento normal é na frequência de 0 Hz e representa a componente da aceleração da gravidade nesse eixo. A amplitude nessa frequência é de aproximadamente 0,704. Já com as pás da hélice desbalanceadas e inclinação do ventilador, a componente nessa mesma frequência teve amplitude de aproximadamente 0,755. Pode-se ver que houve um certo aumento na amplitude dessa componente principal, indicando uma inclinação no *beacon* diferente da original. Além disso, nota-se um aumento visível na amplitude das demais frequências ao longo do espectro, indicando que o sinal no tempo da vibração do ventilador com

defeito foi, em geral, mais ruidoso. Nos eixos Y e Z, pode-se fazer análises semelhantes em relação às componentes de maior amplitude e às componentes ao longo do espectro com o aumento da frequência. Ademais, é possível fazer análises mais profundas no espectro da frequência como a análise do envelope do gráfico na frequência. Essas análises não fazem parte do escopo deste trabalho e não foram incluídas.

Novamente, vê-se que o ideal para a certificação de que houve falha ou não seria a realização de um número maior de testes com funcionamento normal para delimitar uma faixa normal de variação do espectro de vibração e encontrar-se um limite para essa faixa. Porém, já é possível notar diferenças claras nos espectros em todos os eixos. Destaque-se, também, que assim como no tempo, o objetivo deste estudo não é encontrar tais limites.

Com a análise tanto no tempo quanto na frequência foi possível concluir que o uso dessas tecnologias, tanto de hardware como de software, e os algoritmos aplicados são suficientes para realizar aplicações de manutenção preditiva via *condition monitoring*, pelo menos na escala reduzida testada. No capítulo seguinte uma análise de testes realizados com uma máquina industrial é feita.

5 IMPLEMENTAÇÃO EM MÁQUINA INDUSTRIAL

Com os testes realizados em ambiente controlado já há boa indicação da aplicabilidade dos *beacons* de *Bluetooth* em métodos de manutenção, entretanto, há algumas diferenças consideráveis entre o protótipo criado e uma aplicação em máquina dentro de fábrica. Dentre as diferenças, pode-se citar: o material do que é feita a máquina (o nosso protótipo é feito majoritariamente de plástico enquanto máquinas industriais são, em sua maioria, de metal), a intensidade de vibração, a quantidade de fontes de ruídos que seria maior em ambiente industrial já que se estaria perto de outras máquinas e nosso protótipo estava em ambiente controlado, entre outros fatores. Desse modo, foi procurada uma forma de se testar os *beacons* em um ambiente um pouco mais próximo do industrial. Com a supervisão de integrantes do laboratório MAGMA (Laboratório de Materiais Magnéticos, pertencente à UFSC) e do professor Paulo Wendhausen, foi possível acessar uma bomba de vácuo e acoplar um exemplar de *Beacon* em sua carcaça. Essa bomba de vácuo é utilizada em anexo a dois compartimentos e serve para manter uma atmosfera controlada para armazenamento de ímãs. A bomba pode se anexar apenas a um desses compartimentos de cada vez, através de um tubo que é conectado no compartimento desejado.

5.1 CONFIGURAÇÃO DA MÁQUINA E DO HARDWARE

A Figura 27 mostra a bomba de vácuo (que possui um número '5' em sua lateral) anexada a um dos recipientes de armazenamento de ímãs. A Figura 28 e a Figura 29 mostram os dois recipientes de armazenamento e, a Figura 30, mostra o local de acoplamento do *beacon*. O *beacon* não foi instalado diretamente na carcaça do motor, pois ele estaria sujeito às altas temperaturas que a carcaça do motor atinge. Foi escolhida a parte traseira de um painel de metal rigidamente fixado ao motor, de modo que a vibração do motor é propagada ao painel fazendo com que este vibre de forma quase idêntica ao motor. A vibração não é exatamente igual, pois podem haver frequências de ressonância diferentes entre o painel e o motor, assim como pode

haver uma diminuição na energia de vibração devido à propagação, entre outros fatores.

Figura 27 - Bomba de vácuo do laboratório MAGMA.



Fonte: Elaborada pelo autor.

Não obstante, esses fatores não alteram os resultados, já que o valor absoluto de vibração do motor não é necessário e sim como os comportamentos de vibração variam a curto e longo prazo. Como as variações em como o motor vibra certamente implicarão variações em como o painel vibra, a instalação do *beacon* no local escolhido é justificada.

A existência de dois recipientes de armazenamento, apesar de ter sido detalhada aqui para melhor entendimento do funcionamento rotineiro da bomba, não afeta os testes realizados.

Figura 28 - Um dos recipientes usados para armazenamento de ímãs.



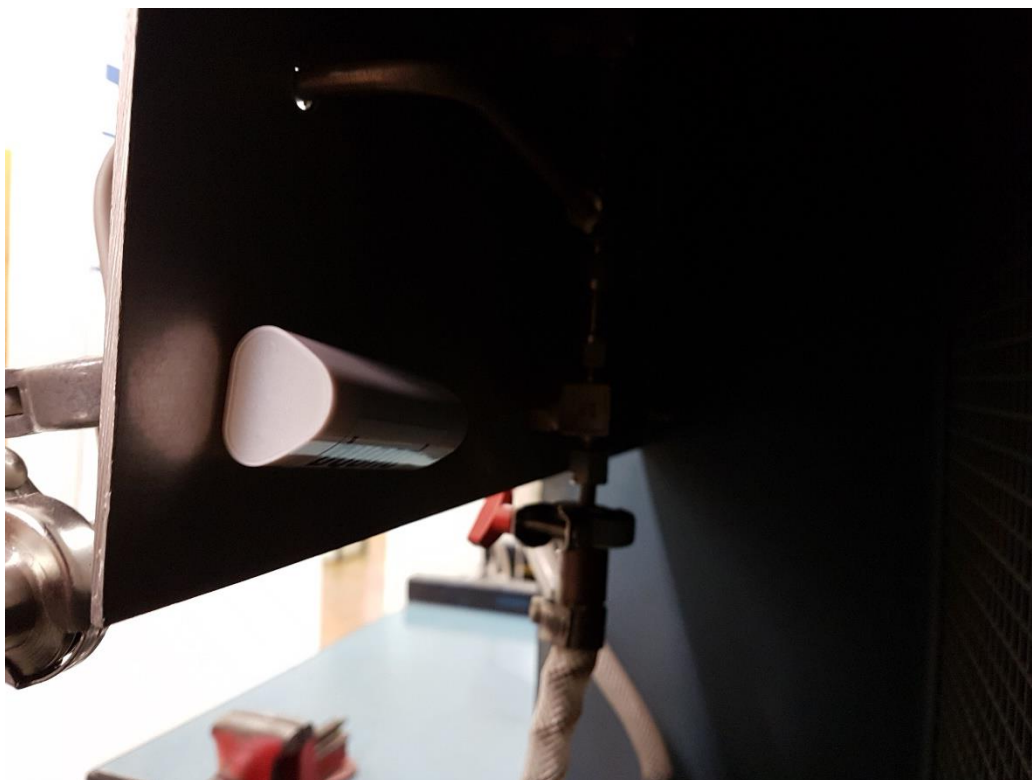
Fonte: Elaborada pelo autor.

Figura 29 - Segundo recipiente usado para armazenamento dos ímãs.



Fonte: Elaborada pelo autor.

Figura 30 - Local de instalação do *beacon*.

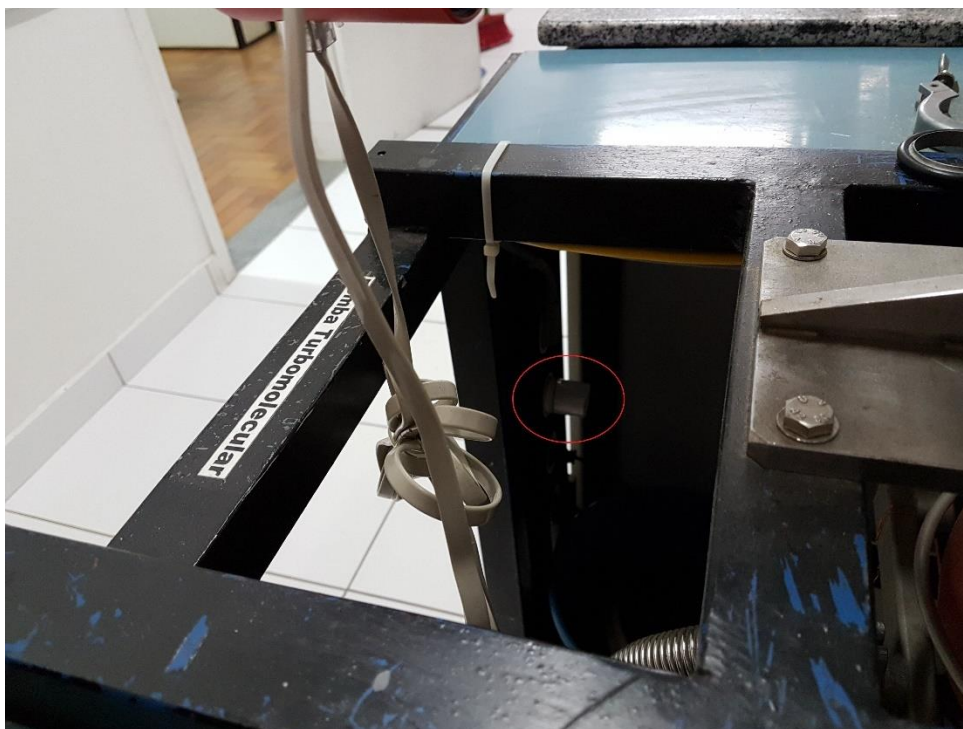


Fonte: Elaborada pelo autor.

5.2 REALIZAÇÃO DOS TESTES

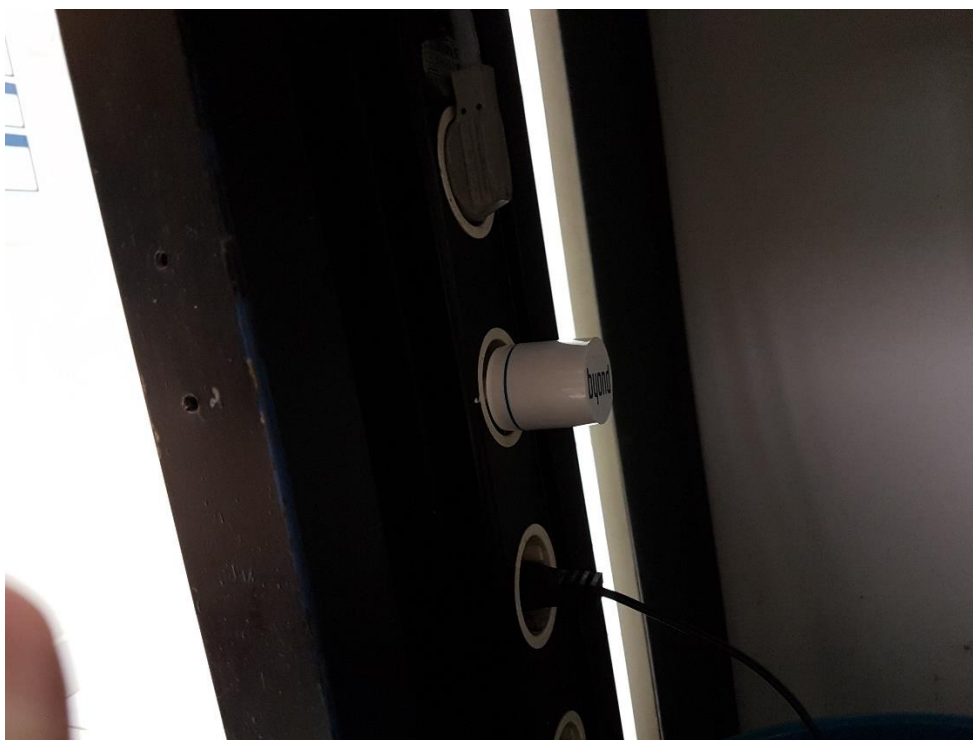
O *beacon* foi deixado por dois dias junto à bomba de vácuo onde, nessa janela de tempo, houve dois períodos de utilização da bomba para bombeamento de ar para fora dos recipientes. Similarmente ao teste no protótipo, depois desses períodos de funcionamento, já se pôde extrair o arquivo .csv da plataforma e realizar as análises dos dados.

Figura 31 - Local de instalação do gateway BluFi.



Fonte: Elaborada pelo autor.

Figura 32 – Imagem mais aproximada do BluFi.



Fonte: Elaborada pelo autor.

5.3 ANÁLISE NO TEMPO

Na etapa de exportação dos dados foi encontrado um problema onde a API de exportação dos dados brutos de vibração dos sensores foi descontinuada pela empresa Bluvision no período entre os testes com o protótipo e os testes na bomba de vácuo. A exportação dos dados ainda foi possível, porém foi feita em pacotes agrupados de medidas de aceleração de aproximadamente 10 segundos cada uma. Essas amostras não englobavam os valores absolutos de vibração, mas sim os parâmetros RMS, valores de pico, e pico a pico, chamado em inglês de *peak to peak*, que é a diferença entre o pico máximo e o pico mínimo durante o período. Desse modo, algumas das análises foram limitadas, porém ainda se pôde fazer uma análise no tempo.

Para a extrapolação desses parâmetros dos pacotes de amostras para todo o período, fez-se algumas modificações, descritas a seguir.

Primeiro, em relação ao valor RMS, fez-se a média de todos os valores RMS dos pacotes. Para encontrar o valor de pico de todo o período, simplesmente foi selecionado o maior valor de pico de todos os pacotes do período. De forma semelhante, para encontrar o maior valor pico a pico no período, selecionou-se o maior valor pico a pico de todos os pacotes.

Os valores desse modo foram calculados no Matlab de acordo com o código 4 incluído no apêndice.

Obtendo-se os valores, criou-se a Tabela 2 e a Tabela 3, apresentadas a seguir, onde é possível ver os parâmetros pico, pico a pico e RMS para os três eixos, nos dois períodos em que a bomba esteve em funcionamento.

Tabela 2 - Parâmetros no tempo do primeiro período de funcionamento para os três eixos.

	Eixo X	Eixo Y	Eixo Z
Pico	2,0024	1,6245	1,9873
Pico a Pico	3,9814	2,9951	3,3896
RMS	0,7471	0,6055	0,3173

Tabela 3 - Parâmetros no tempo do segundo período de funcionamento para os três eixos.

	Eixo X	Eixo Y	Eixo Z
Pico	1,8608	1,4158	1,8926
Pico a Pico	3,5709	2,7406	3,2896
RMS	0,7023	0,5750	0,2811

Fazendo-se a comparação dos valores das Tabelas 2 e 3, vê-se que eles não se distanciam muito entre os dois períodos de uso da bomba, sendo o segundo conjunto de dados um pouco menor em termos de intensidade de vibração, que pode ser observado pelos valores menores de RMS dos eixos e, também, pela redução nos valores de pico. Como é sabido que ambos os conjuntos de dados se relacionam a um funcionamento da bomba sem defeito, pode-se tomar esses valores como hipótese inicial para limites de uma faixa de funcionamento normal da bomba.

Desse modo, tem-se, por exemplo, que a faixa de funcionamento normal para valores de pico de aceleração do eixo X é de 1,86 a 2 g, a faixa de funcionamento normal de pico a pico é de 3,57 a 3,98 g e a faixa de funcionamento normal de valor RMS é de 0,70 a 0,74 g. Com subsequentes testes poderia se ter um maior grau de certeza a respeito dessa faixa e, a partir daí, monitorar os valores de aceleração para classificar os que saíssem muito dessa faixa como um possível comportamento defeituoso.

5.4 ANÁLISE NA FREQUÊNCIA

Como mencionado anteriormente, a perda de acesso a valores individuais de medidas de aceleração do *Beacon* prejudicou a análise durante os testes com a bomba de vácuo. A análise na frequência ficou especialmente debilitada, já que as amostras individuais de aceleração são essenciais para o cálculo da transformada discreta de Fourier do sinal e só existem os valores de pico, pico a pico e RMS. Além disso, esses valores são calculados com uma média de 10 segundos, então, mesmo se houvesse os valores absolutos de aceleração com essa taxa de amostragem, ela não teria uma frequência de amostragem suficientemente alta pra extrair as informações das componentes frequenciais do sinal, já que é necessário, no mínimo, que a taxa de amostragem seja maior que a frequência de Nyquist do sinal.

6 CONSIDERAÇÕES FINAIS E PERSPECTIVAS

Uma tarefa importante de um engenheiro em uma empresa, principalmente uma que possui setores industriais, é lidar com a aquisição e familiarização de novas tecnologias existentes no mercado. Essas tecnologias podem ser provenientes de recentes estudos acadêmicos em universidades e afins, bem como também da oferta de novos produtos vendidos por outras empresas. Uma parte considerável do trabalho feito durante o período na empresa foi baseada no estudo e familiarização da utilização de uma dessas tecnologias: os *beacons* industriais de *Bluetooth low energy*. Por serem pequenos e acoplados externamente, esses *beacons* são facilmente adicionados a máquinas em linhas de produção já existentes, sem requerer nenhuma modificação significativa a elas ou ao processo de produção. Além disso, eles são conectados à internet via os gateways BluFis e podem ser integrados a outras soluções baseadas em conceitos da indústria 4.0 como IoT (“internet das coisas”) e computação na nuvem. Esses conceitos são muito visados atualmente pelas empresas que querem se manter alinhadas com as tecnologias recentes e fazer uso do grande potencial de soluções que existe no mercado.

Também foi objetivo deste trabalho ressaltar o uso de técnicas mais elaboradas de manutenção em máquinas e motores industriais. As manutenções preditiva e prescritiva (ou detectiva) possuem um grande potencial positivo de mudança na forma como as empresas lidam com a segurança de seus equipamentos e funcionários assim como a otimização de suas linhas de produção.

Neste projeto pôde-se verificar a eficácia da utilização desses *beacons* nos testes realizados em escala reduzida para esses tipos de manutenção, conseguindo-se detectar a existência de uma falha simulada em um aquecedor de pequeno porte usando os algoritmos prontos disponibilizados pela Bluvision. Visando aprofundamento na parte teórica e maior exploração da capacidade da solução, foram testados dois algoritmos adicionais de detecção de falha via *condition monitoring*: um no domínio do tempo e outro no domínio da frequência. Esses algoritmos mostraram-se capazes de atingir os objetivos de manutenção preditiva com técnicas de *condition monitoring* no teste com o protótipo.

Entretanto, passando-se para o teste em máquina industrial, houve o problema da falta de suporte da Bluvision em relação a API utilizada anteriormente afetando diretamente o resultado dos testes e a possibilidade de análise no domínio da frequência. Esses problemas obtidos mostram a dependência que a Byond tem em relação a essa API o que pode ser um empecilho caso deseje realizar o desenvolvimento de sua própria plataforma baseada nos dados extraídos da Bluzone. Houve um contato com o suporte da Bluvision, porém o problema mesmo assim não foi resolvido. Dessa forma, caso a Byond opte por continuar utilizando os beacons da Bluvision ela está restrita a usar o algoritmo desenvolvido pela Bluvision até que a API volte a se tornar utilizável (se é que isso de fato acontecerá). Caso a API possa novamente ser usada, uma opção para aprimorar soluções futuras da Byond é fazer um estudo mais aprofundado de quais algoritmos são mais apropriados para diferentes máquinas e processos industriais. Além disso, analisando diferentes falhas pode-se até tomar o próximo passo que é a realização de manutenção prescritiva. Desse modo seria possível indicar com mais detalhes qual é o defeito que está sendo desenvolvido na máquina e poder-se-ia indicar melhor qual é a vida útil restante dela.

Alternativamente a Byond tem a opção de fabricar seus próprios *beacons* com sensores ou encontrar outra fabricante de *beacons* para ser sua fornecedora de modo a ter maior flexibilidade no uso dos dados dos sensores.

REFERÊNCIAS

- [1] TIPOS DE MANUTENÇÃO. Disponível em: <<http://www.dee.ufrn.br/~joao/manut/05%20-%20Cap%EDtulo%203.pdf>>. Acesso em: 02 ago. 2018.
- [2] IEEE INTERNATIONAL CONGRESS ON NEURAL NETWORKS, 1996, Washington, D.C.. **A hybrid neural network based machine condition forecaster and classifier by using multiple vibration parameters**. Washington, D.C.: [s.n.], 1996. 2096-2100 p. v. 4.
- [3] *Condition monitoring*. In: Wikipédia: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Condition_monitoring> Acesso em: 20 jun. 2018.
- [4] MECHEFSKE, C. K. *Machine Condition monitoring and Fault Diagnosis*. In: **Machine Condition monitoring and Fault Diagnosis, Vibration and Shock Handbook**. Boca Raton: CRC Press, 2005. p. 25.
- [5] GAO, Robert. Chapter 8 Neural Networks for Machine *Condition monitoring* and Fault Diagnosis. In: **Neural Networks for Machine Condition monitoring and Fault Diagnosis**. Amherst: [s.n.], 2018. cap. 8, p. 167-185.
- [6] WHAT is a *Beacon*?. Disponível em: <<https://kontakt.io/Beacon-basics/what-is-a-Beacon>>. Acesso em: 20 jun. 2018.
- [7] *Bluetooth low energy Beacon*. In: Wikipédia: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Bluetooth_low_energy_Beacon>. Acesso em: 20 jun. 2018.
- [8] PRODUCTS. 2018. Disponível em: <<http://bluvision.com/our-products/>>. Acesso em: 21 jul. 2018.
- [9] **MANUAL do consumidor VENTISOL**: Aquecedor de ambientes, modelo A1. [S.l.: s.n.], 2018. 1 p.
- [10] **SPECIFICATION sheet - BEEKs industrial ver 1.6**. [S.l.: s.n.], 2017. 1 p. Disponível em: <http://bluvision.com/wp-content/uploads/2017/09/Specs-BEEKs-Industrial_1.pdf>. Acesso em: 13 jun. 2018.
- [11] CARDEN, E. Peter; FANNING, Paul. Vibration Based *Condition monitoring*: A Review. **Structural Health Monitoring**, Dublin, v. 3, n. 4, p. 355-377, dez. 2004.
- [12] BUTLER, Shane. **Prognostic Algorithms for Condition monitoring and Remaining Useful Life Estimation**. 2012. 33-65 p. Dissertação (Doutorado)-Engenharia eletrônica, National University of Ireland, Maynooth, 2012.
- [13] ZHANG, Wan et al. Comprehensive Overview on Computational Intelligence Techniques for Machinery *Condition monitoring* and Fault Diagnosis. **Chinese Journal of Mechanical Engineering**, Pequim, v. 30, n. 4, p. 782-795, jul. 2017.
- [14] CAESARENDRA, Wahyu; TIAHJOWIDODO, Tegoeh. A Review of Feature Extraction Methods in Vibration-Based *Condition monitoring* and Its Application for

Degradation Trend Estimation of Low-Speed Slew Bearing. **Machines**, [S.l.], p. 1-28, set. 2017. Disponível em: <<http://www.mdpi.com/2075-1702/5/4/21>>. Acesso em: 21 jun. 2018.

[15] P, Lakshmi Pratyusha; V, Shanmukha Priya; NAIDU, VPS. Bearing Health *Condition monitoring*: Time Domain Analysis. **International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering**, [S.l.], p. 75-82, dez. 2014. Disponível em: <http://www.ijareeie.com/upload/2014/icetes/10_lakshmi%20v.p.pdf>. Acesso em: 22 jun. 2018.

[16] KIM, Eric Y. et al. Experimental Study on *Condition monitoring* of Low Speed Bearings: Time Domain Analysis. In: Fifth Australasian Congress on Applied Mechanics (ACAM 2007), 2007, Brisbane, Austrália. **Proceedings of the Fifth Australasian Congress on Applied Mechanics...** Brisbane, Austrália: Engineers Australia, 2007. p. 108-113. v. 1. Disponível em: <<https://espace.library.uq.edu.au/view/UQ:131913/B1.3.pdf>>. Acesso em: 23 jun. 2018.

[17] SKEWNESS. Disponível em: <<https://www.mathworks.com/help/stats/skewness.html>>. Acesso em: 15 jun. 2018.

[18] VAR. Disponível em: <<https://www.mathworks.com/help/matlab/ref/var.html>>. Acesso em: 15 jun. 2018.

[19] FAST Fourier transform. In: Wikipédia: the free encyclopedia. Disponível em: <https://en.wikipedia.org/wiki/Fast_Fourier_transform>. Acesso em: 16 jun. 2018.

[20] FOURIER transform. In: Wikipédia: the free encyclopedia. Disponível em: <https://en.wikipedia.org/wiki/Fourier_transform>. Acesso em: 16 jun. 2018.

APÊNDICE

Código 1 – programa que gera os gráficos de vibração do ventilador:

```

%% Universidade Federal de Santa Catarina
%% Centro Tecnológico (CTC)
%% Conrado Viveiros Jordan
%% 2018
%%
%% Programa que traça os gráficos de vibração do ventilador
funcionando em modo normal e com defeito simulado

% Carregamento do workspace com os dados importados de um arquivo
.csv
load('Ventilador balanceado e desbalanceado.mat')

figure

% Geração do gráfico para o eixo X
subplot(3,1,1)
plot(EixoX)
xlim([0 12500])
ylim([-3 2])
line([3310 3310], [-5 5], 'color', [0 1 0])
line([6330 6330], [-5 5], 'color', [0 1 0])
line([7400 7400], [-5 5], 'color', [1 0 0])
line([12286 12286], [-5 5], 'color', [1 0 0])

% Geração do gráfico para o eixo Y
subplot(3,1,2)
plot(EixoY)
xlim([0 12500])
ylim([-3 2])
line([3310 3310], [-5 5], 'color', [0 1 0])
line([6330 6330], [-5 5], 'color', [0 1 0])
line([7400 7400], [-5 5], 'color', [1 0 0])
line([12286 12286], [-5 5], 'color', [1 0 0])

% Geração do gráfico para o eixo Z
subplot(3,1,3)
plot(EixoZ)
xlim([0 12500])
ylim([-3 2])
line([3310 3310], [-5 5], 'color', [0 1 0])
line([6330 6330], [-5 5], 'color', [0 1 0])
line([7400 7400], [-5 5], 'color', [1 0 0])
line([12286 12286], [-5 5], 'color', [1 0 0])

```

Código 2 – programa que calcula as características do sinal no tempo:

```

%% Universidade Federal de Santa Catarina
%% Centro Tecnológico (CTC)
%% Conrado Viveiros Jordan
%% 2018
%%
%% Programa que calcula as características desejadas de um sinal no
tempo para análise de Condition Monitoring

% Carregamento do workspace com os dados importados de um arquivo
.csv
load('Ventilador balanceado e desbalanceado.mat')

% Variável com as medidas de vibração
dados = {'Eixo X normal', 'Eixo Y normal', 'Eixo Z normal', 'Eixo X
com defeito', 'Eixo Y com defeito', 'Eixo Z com defeito';
        EixoXnormal, EixoYnormal, EixoZnormal, EixoXdefeito,
EixoYdefeito, EixoZdefeito};

for var = 1:6
    nomeEixo = dados{1,var};
    eixo = dados{2,var};

    % Cálculo da soma e do número de elementos do vetor de medidas de
vibração
    soma = sum(eixo);
    N = length(eixo);
    tempo = (1:N).*1.25; % Vetor tempo

    % Característica número 1: média do sinal
    media = soma/N;

    % Característica número 2: valores de pico (positivo e negativo)
    maximo = max(eixo);
    minimo = min(eixo);

    % Característica número 3: valor RMS
    X2 = eixo.^2; % Quadrado dos valores
    somaX2 = sum(X2); % Soma do quadrado dos valores
    mq = somaX2/N; % Média da soma dos quadrados
    RMS = sqrt(mq);

    % Característica número 4: variância
    dist = eixo-media;
    distAbs = abs(dist);
    distA2 = distAbs.^2;
    somaDA2 = sum(distA2);
    V = somaDA2/(N-1);

    % Característica número 5: coeficiente de assimetria
    A = (sum((eixo-media).^3)/N);
    B = ((sum((eixo-media).^2))/N)^(1/3);
    Ca = A/B;

```

```
% Tracando os gráficos e mostrando os resultados
figure;
plot(tempo,eixo)
title(sprintf('Sinal X(t) - %s',nomeEixo));
xlabel('Tempo (milissegundos)');
ylabel('X(t)');
f = figure('NumberTitle', 'off', 'Name', sprintf('Análise do
%s',nomeEixo));
t = uitable(f);
f.Position = [750 500 280 130];
d = {'Média',media;'Pico máximo',maximo;'Pico
mínimo',minimo;'RMS',RMS;'Variância',V;'Coeficiente de
assimetria',Ca};
t.Data = d;
t.Position = [0 0 280 130];
t.ColumnName = {'Característica','Valor'};
set(t,'ColumnWidth',{109})
var = var + 1;
end
```

Código 3 – programa que gera a transformada de Fourier dos sinais:

```

%% Universidade Federal de Santa Catarina
%% Centro Tecnológico (CTC)
%% Conrado Viveiros Jordan
%% 2018
%%
%% Programa que gera a FFT dos sinais de vibração dos eixos X, Y e Z

% Carregamento do workspace com os dados importados de um arquivo
.csv
load('Ventilador balanceado e desbalanceado.mat')

% Variável com as medidas de vibração
dados = {'Eixo X normal', 'Eixo Y normal', 'Eixo Z normal', 'Eixo X
com defeito', 'Eixo Y com defeito', 'Eixo Z com defeito';
        EixoXnormal, EixoYnormal, EixoZnormal, EixoXdefeito,
EixoYdefeito, EixoZdefeito};

Fs = 800;           % Frequência de amostragem
T = 1/Fs;          % Período de amostragem

for var = 1:6
    nomeEixo = dados{1,var};
    eixo = dados{2,var};
    L = length(eixo);           % Tamanho do sinal
    t = (0:L-1)*T;              % Vetor tempo
    f = Fs*(0:(L/2))/L;         % Vetor frequencia

    % Gráfico no tempo do sinal
    figure
    plot(1000*t,eixo)
    title(sprintf('X(t) - %s',nomeEixo))
    xlabel('Tempo (millissegundos)')
    ylabel('X(t)')

    % FFT do sinal
    Y = fft(eixo);
    P2 = abs(Y/L);
    P1 = P2(1:L/2+1);
    P1(2:end-1) = 2*P1(2:end-1);

    % Gráfico da FFT
    figure
    plot(f,P1)
    title(sprintf('Espectro de amplitude na frequência do
%s',nomeEixo))
    xlabel('frequência (Hz)')
    ylabel('|F(w)|')
end

```

Código 4 – programa que calcula as características do sinal no tempo:

```

%% Universidade Federal de Santa Catarina
%% Centro Tecnológico (CTC)
%% Conrado Viveiros Jordan
%% 2018
%%
%% Programa que calcula as características desejadas de um sinal no
tempo para análise de Condition Monitoring

% Carregamento do workspace com os dados importados de um arquivo
.csv
load('Bomba de vácuo MAGMA.mat')

% Primeiro período de funcionamento
xPeak1 = xPeak(5683:5775);
xPeakToPeak1 = xPeakToPeak(5683:5775);
xRms1 = xRms(5683:5775);

yPeak1 = yPeak(5683:5775);
yPeakToPeak1 = yPeakToPeak(5683:5775);
yRms1 = yRms(5683:5775);

zPeak1 = zPeak(5683:5775);
zPeakToPeak1 = zPeakToPeak(5683:5775);
zRms1 = zRms(5683:5775);

% Segundo período de funcionamento
xPeak2 = xPeak(8245:8320);
xPeakToPeak2 = xPeakToPeak(8245:8320);
xRms2 = xRms(8245:8320);

yPeak2 = yPeak(8245:8320);
yPeakToPeak2 = yPeakToPeak(8245:8320);
yRms2 = yRms(8245:8320);

zPeak2 = zPeak(8245:8320);
zPeakToPeak2 = zPeakToPeak(8245:8320);
zRms2 = zRms(8245:8320);

% Calculando os valores de pico do primeiro período
picoEixoX1 = max(xPeak1);
picoEixoY1 = max(yPeak1);
picoEixoZ1 = max(zPeak1);
% Calculando os valores de pico a pico do primeiro período
papEixoX1 = max(xPeakToPeak1);
papEixoY1 = max(yPeakToPeak1);
papEixoZ1 = max(zPeakToPeak1);
% Calculando os valores RMS do primeiro período
rmsEixoX1 = mean(xRms1);
rmsEixoY1 = mean(yRms1);

```

```
rmsEixoZ1 = mean(zRms1);

% Calculando os valores de pico do segundo período
picoEixoX2 = max(xPeak2);
picoEixoY2 = max(yPeak2);
picoEixoZ2 = max(zPeak2);
% Calculando os valores de pico a pico do segundo período
papEixoX2 = max(xPeakToPeak2);
papEixoY2 = max(yPeakToPeak2);
papEixoZ2 = max(zPeakToPeak2);
% Calculando os valores RMS do segundo período
rmsEixoX2 = mean(xRms2);
rmsEixoY2 = mean(yRms2);
rmsEixoZ2 = mean(zRms2);
```