

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

**Desenvolvimento de um Dispositivo para Monitoramento
Dinâmico de Estruturas**

José Lucas Silva Borges

Porto Alegre
2021

JOSÉ LUCAS SILVA BORGES

**DESENVOLVIMENTO DE UM DISPOSITIVO PARA
MONITORAMENTO DINÂMICO DE ESTRUTURAS**

Dissertação apresentada ao Programa de Pós-Graduação em
Engenharia Civil da Universidade Federal do Rio Grande do Sul,
como parte dos requisitos para obtenção do título de Mestre em
Engenharia.

Porto Alegre

2021

CIP - Catalogação na Publicação

Borges, José Lucas Silva
Desenvolvimento de um Dispositivo para
Monitoramento Dinâmico de Estruturas / José Lucas
Silva Borges. -- 2021.
100 f.
Orientador: Marcelo Maia Rocha.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Escola de Engenharia, Programa de
Pós-Graduação em Engenharia Civil, Porto Alegre,
BR-RS, 2021.

1. Análise Dinâmica de Estruturas. 2. Monitoramento
de Integridade Estrutural. 3. Instrumentação. I.
Rocha, Marcelo Maia, orient. II. Título.

JOSÉ LUCAS SILVA BORGES

**DESENVOLVIMENTO DE UM DISPOSITIVO PARA
MONITORAMENTO DINÂMICO DE ESTRUTURAS**

Esta dissertação de mestrado foi julgada adequada para a obtenção do título de MESTRE EM ENGENHARIA, Estruturas, e aprovada em sua forma final pelo professor orientador e pelo Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, 17 de junho de 2021

Prof. Marcelo Maia Rocha

Dr. techn pela Universität Innsbruck, Áustria
Orientador

Prof. Nilo Cesar Consoli

Coordenador do PPGEC/UFRGS

BANCA EXAMINADORA

Prof. Herbert Martins Gomes (UFRGS)

Dr. pela Universidade Federal do Rio Grande do Sul

Prof. Zacarias Martin Chamberlain Pravia (FUPF)

Dr. pela Universidade Federal do Rio de Janeiro

Prof. Antônio Ernandes Macêdo Paiva (IFMA)

Dr. pela Universidade Federal de São Carlos

AGRADECIMENTOS

Agradeço primeiramente a Deus, por guiar e iluminar meus caminhos.

Aos meus pais, José Ferreira Borges e Maria Vanda Silva Borges pelo apoio incondicional, carinho, compreensão e dedicação. E ao meu irmão, José Felipe Silva Borges, pelo carinho e compreensão.

A minha namorada, Rayara Costa, por toda a ajuda, apoio, carinho e compreensão. Você foi essencial para esta conquista.

Ao meu orientador, professor Dr. Marcelo Maia Rocha, pela orientação, paciência e tutela desde o início do presente trabalho.

Ao professor do IFMA Dr. Ernandes Paiva, pela ajuda no desenvolvimento do trabalho ao fornecer acesso ao laboratório de ensaios.

Ao Matheus Carini por me ajudar no estudo e aplicação dos métodos de identificação de sistemas abordados em sua tese de doutorado.

Aos meus colegas e amigos, vocês todos contribuíram para minha formação.

À FAPEMA pelo apoio financeiro na forma de bolsa de estudos.

“A persistência é o menor caminho do êxito”

Charles Chaplin

RESUMO

BORGES, J.L.S. **Desenvolvimento de um Dispositivo para Monitoramento Dinâmico de Estruturas**. 2021. Dissertação (Mestrado em Estruturas) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre.

No âmbito das construções civis, a diminuição da capacidade portante juntamente com incertezas relacionadas ao estado de integridade pode ocasionar, em casos extremos, a inviabilidade da utilização das mesmas ou até mesmo o colapso. Consequentemente, evidencia-se a possibilidade de danos à manutenção da vida, perdas financeiras, econômicas, ambientais ou uma combinação entre todos. Como forma de combater tais empecilhos, o Monitoramento da Integridade Estrutural (MIE) emerge como uma prática de grande relevância. O monitoramento de uma estrutura possibilita estabelecer políticas mais eficientes de atuação tanto em âmbito emergencial quanto de manutenção preventiva, reduzindo, assim, custos e tempo de reparo. Em contrapartida, a implantação do MIE encontra alguns desafios na sua difusão. Dentre eles, destacam-se os altos custos relacionados a implantação e manutenção do sistema. Com o intuito de tornar a utilização do MIE mais atrativa, necessita-se desenvolver dispositivos que empreguem tecnologias atuais, compactas, de baixo custo e fácil instalação. Tomando como enfoque a detecção de danos baseada em vibração, o presente trabalho tem como objetivo desenvolver, programar e implementar um dispositivo de baixo custo baseado em acelerômetro MEMS a ser aplicado como ferramenta no monitoramento de integridade estrutural. A instrumentação sugerida caracterizou-se por ser um sistema de aquisição de dados composto por três unidades de medição simultânea. Cada uma delas possuía um sensor MPU 6500 gerenciado por um microprocessador ESP 32. A aquisição e interpretação dos dados oriundos de cada uma dessas unidades ficou a cargo de um Raspberry Pi 3. Como forma de avaliar o comportamento do sistema, o mesmo submeteu-se a ensaios de performance de ruído, *throughput* de rede, perdas no envio de dados e validação de leitura. Além disso, utilizou-se o dispositivo proposto para extrair as frequências naturais de um modelo de dimensões reduzidas. Por fim, avaliou-se o total gasto para construção do dispositivo proposto. Conclui-se por meio dos resultados que foi possível desenvolver um sistema de aquisição de dados confiável para emprego no monitoramento de integridade estrutural, utilizando equipamentos de baixo custo e grande disponibilidade no mercado.

Palavras-chave: *Integridade Estrutural; Vibração; Instrumentação; Baixo Custo.*

ABSTRACT

BORGES, J.L.S. **Desenvolvimento de um Dispositivo para Monitoramento Dinâmico de Estruturas**. 2021. Dissertação (Mestrado em Estruturas) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre.

In the context of civil constructions, the reduction of carrying capacity in addition to uncertainties related to their integrity can cause, in extreme cases, the impossibility of using them or even a collapse. Consequently, there is the possibility of damage considering financial, economic, and environmental aspects and the maintenance of life or a combination of all. As a way to combat such obstacles, Structural Health Monitoring (SHM) emerges as a highly relevant practice. Monitoring a structure enables the establishment of more efficient action policies in emergency cases and preventive maintenance, thus reducing costs and repair time. On the other hand, the implementation of the SHM faces some challenges in its diffusion. Among them, the high costs related to the implementation and maintenance of the system stand out. In order to make the use of the SHM more attractive, it is necessary to develop devices that employ current, compact, low-cost, and easy-to-install technologies. Focusing on vibration-based damage detection, this work aims to develop, program and implement a low-cost MEMS accelerometer-based device to be applied as a tool in structural health monitoring. The suggested instrumentation was characterized by being a data acquisition system composed of three simultaneous measurement units. Each one of them had a MPU 6500 sensor managed by an ESP 32 microprocessor. The acquisition and interpretation of data from each of these units was in charge of a Raspberry Pi 3. As a way of evaluating the behavior of the system, it was submitted to noise performance tests, network throughput, data sending losses and read validation. Furthermore, the proposed device was used to extract the natural frequencies of a model with reduced dimensions. Finally, the total expenditure for construction of the proposed device was evaluated. It is concluded through the results that it was possible to develop a reliable data acquisition system for use in structural health monitoring, using equipment of low cost and wide availability in the market.

Key-words: *Structural Integrity; Vibration; Instrumentation; Low cost.*

LISTA DE FIGURAS

Figura 1: Esquema de um sistema de MIE	20
Figura 2: Elementos de um sistema de medição.....	25
Figura 3: Esquema de funcionamento de um sensor	25
Figura 4: Esquema de funcionamento do acelerômetro	26
Figura 5: Comportamento em relação à frequência de excitação.....	27
Figura 6: <i>Strain Gauge</i> piezoresistivo	28
Figura 7: Acelerômetro <i>strain gauge</i>	28
Figura 8: Alguns modos de deformação para gerar carga elétrica	29
Figura 9: Exemplo de deslocamento de placas em sensores unidirecionais.....	30
Figura 10: Sensor de capacitância diferencial	30
Figura 11: Ilustração de um acelerômetro MEMS triaxial.	31
Figura 12: Representação da sensibilidade de um sensor	33
Figura 13: Exemplo de não linearidade em um acelerômetro	33
Figura 14: Tipos de filtros em relação ao modo de atenuação	34
Figura 15: Comportamento ideal de um conversor A/D	35
Figura 16: Aquisição de sinal a diferentes taxas de amostragem	36
Figura 17: Representação de onda quadrada	36
Figura 18: Comunicação SPI.....	37
Figura 19: SPI com vários <i>slaves</i>	38
Figura 20: Comunicação serial	38
Figura 21: Comunicação paralela.....	38
Figura 22: Estrutura do pacote de dados UART.....	39
Figura 23: Sinais de saída do módulo RS-485	40
Figura 24: Interpretação do valor diferencial.	40

Figura 25: Esquemático de um sistema de medição generalizado	41
Figura 26: MPU-6500 e regulador de tensão	42
Figura 27: Diagrama de blocos do MPU-6500.....	43
Figura 28: Módulo ESP-32.....	44
Figura 29: Raspberry Pi 3 B+ (a) modelo utilizado; (b) pinagem do modelo.....	44
Figura 30: Módulo RS-485.....	45
Figura 31: Esquema de ligação.....	46
Figura 32: Módulo USB-TTL.....	46
Figura 33: Cabo de dados Cat5e.....	47
Figura 34: Baterias empregadas.	47
Figura 35: Circuito de proteção BMS.....	47
Figura 36: Esquema elétrico de ligação.....	48
Figura 37: Regulador de tensão.	48
Figura 38: Esquema detalhado de ligação de um ponto de medição.....	49
Figura 39: Esquema completo do sistema de medição.....	49
Figura 40: Ponto de medição.	50
Figura 41: Módulo de aquisição de dados.	50
Figura 43: IDE do Arduino.....	51
Figura 42: Representação do mecanismo de sincronização da leitura	52
Figura 44: Ensaio de performance de ruído.....	54
Figura 45: Aparato para teste de validação da leitura	56
Figura 46; Representação esquemática do modelo ensaiado.....	56
Figura 47: Aparato para ensaio.....	57
Figura 48: Densidade Espectral de Potência.	58
Figura 49: Monitoramento do <i>buffer</i> de entrada.....	59
Figura 50: Tempo para aquisição de um conjunto de dados – dT.....	60

Figura 51: Utilização do processador.	60
Figura 52: Periodograma dos sinais medidos em cada sensor. (a) 30Hz; (b) 40Hz; (c) 50Hz e (d) 60Hz.	61
Figura 53: Coeficiente de correlação para 30Hz.	62
Figura 54: Coeficiente de correlação para 40Hz.	62
Figura 55: Coeficiente de correlação para 50Hz.	63
Figura 56: Coeficiente de correlação para 60Hz.	63

LISTA DE TABELAS

Tabela 1: <i>Gauge Factor</i> para diferentes materiais	28
Tabela 2: Configurações iniciais MPU-6500	51
Tabela 3: Estrutura do pacote de dados	52
Tabela 4: Exemplo prático da estrutura do pacote de dados	52
Tabela 5: Configurações da comunicação serial	53
Tabela 6: Ruído medido do sensor	58
Tabela 7: Erro na leitura	62
Tabela 8: Índices de correlação entre medições de sensores a 30Hz	63
Tabela 9: Índices de correlação entre medições de sensores a 40Hz	63
Tabela 10: Índices de correlação entre medições de sensores a 50Hz	63
Tabela 11: Índices de correlação entre medições de sensores a 60Hz	64
Tabela 12: Propriedades geométricas da seção	64
Tabela 13: Frequências naturais calculadas.....	64
Tabela 14: Caracterização do sinal medido.....	65
Tabela 15: Relação de custo detalhada do sistema de medição	65

SUMÁRIO

1. INTRODUÇÃO.....	14
1.1. JUSTIFICATIVA DO TRABALHO	16
1.2. OBJETIVO GERAL	17
1.2.1. Objetivos Específicos.....	17
1.3. ORGANIZAÇÃO DO TRABALHO.....	17
2. MONITORAMENTO DA INTEGRIDADE ESTRUTURAL.....	19
3. SISTEMA DE MEDIÇÃO	24
3.1. SENSORES.....	25
3.1.1. Acelerômetros	25
3.1.1.1. Técnicas Empregadas	27
3.1.1.2. Sensores MEMS	31
3.1.1.3. Parâmetros de Escolha.....	31
3.2. CONDICIONAMENTO DO SINAL	33
3.3. CONVERSÃO DO SINAL.....	34
3.4. COMUNICAÇÃO DIGITAL	37
3.5. CALIBRAÇÃO.....	40
4. SISTEMA DE MEDIÇÃO PROPOSTO PARA O MONITORAMENTO DINÂMICO DE ESTRUTURAS.....	42
4.1. MATERIAIS EMPREGADOS.....	42
4.1.1. MPU-6500.....	42
4.1.2. ESP-32	43
4.1.3. Raspberry Pi	44
4.1.4. Módulo RS-485.....	45
4.1.5. Módulo USB-TTL.....	46

4.1.6. Cabo de Rede	46
4.1.7. Banco de Baterias.....	47
4.1.8. Regulador de Tensão.....	48
4.2. MÉTODOS	48
4.2.1. Esquema de Ligação	48
4.2.2. Leitura e Envio de Dados do Sensor.....	50
4.2.3. Aquisição e Tratamento de Dados	53
4.2.4. Análise de Performance de Ruído.....	53
4.2.5. Throughput de rede	54
4.2.6. Perdas no envio de dados	55
4.2.7. Validação da Leitura	55
4.2.8. Modelo em dimensões reduzidas	56
4.2.9. Preço do sistema de medição	57
5. RESULTADOS E DISCUSSÕES.....	58
5.1. ANÁLISE DE PERFORMANCE DE RUÍDO.....	58
5.2. THROUGHPUT DE REDE.....	59
5.3. PERDAS NO ENVIO DE DADOS	60
5.4. VALIDAÇÃO DA LEITURA	61
5.5. MODELO EM DIMENSÕES REDUZIDAS	64
5.6. PREÇO TOTAL DO SISTEMA DE MEDIÇÃO.....	65
6. CONCLUSÃO.....	67
6.1. TRABALHOS FUTUROS	68
REFERÊNCIAS BIBLIOGRÁFICAS	69
APÊNDICE 1: ESPECIFICAÇÕES TÉCNICAS DO GIROSCÓPIO	76
APÊNDICE 2: ESPECIFICAÇÕES TÉCNICAS DO ACELERÔMETRO	77
ANEXO 1: CÓDIGO ESP-32.....	78

ANEXO 3: FUNÇÃO ERRORCHECKING.....	82
ANEXO 4: FUNÇÃO READDATA.....	83
ANEXO 5: FUNÇÃO STARTPROCESS	84
ANEXO 6: FUNÇÃO CONVERTSAVEDATA	85
ANEXO 7: FUNÇÃO THROUGHPUT	86
ANEXO 8: FUNÇÃO LOSTPACKETS.....	87
ANEXO 9: FUNÇÃO SHOWRESULTS.....	88
ANEXO 10: LEITURA DOS SENSORES A 0HZ	91
ANEXO 11: LEITURA DOS SENSORES A 30HZ:	92
ANEXO 12: LEITURA DOS SENSORES A 40HZ	94
ANEXO 13: LEITURA DOS SENSORES A 50HZ	96
ANEXO 14: LEITURA DOS SENSORES A 60HZ	98

1. INTRODUÇÃO

A busca do comportamento de uma estrutura na construção civil deve ser feita em todas as suas fases. Na etapa de projeto, utilizam-se modelos de cálculo para prever seu comportamento frente aos mais variados tipos e intensidades das ações do ambiente, observadas tanto nas fases de construção quanto de operação. Em face dessa variabilidade nas ações, utilizam-se fatores de segurança estipulados em normas de projeto. Aliado a isso, tem-se o emprego de modelos de cálculo simplificados e, em diversos casos, a favor da segurança. Como resultado, obtém-se muitas vezes um acréscimo na capacidade portante em comparação àquela definida para um nível de confiabilidade ideal (COSTA; PROVERBIO; SMITH, 2018). Destaca-se que esse excesso é difícil de quantificar uma vez que torna-se suscetível a alterações dos seguintes fatores: divergências entre modelo de cálculo e real; mudanças nas cargas de serviço (no caso de pontes, crescimento no fluxo e carga por eixo de caminhões); deterioração dos materiais utilizados (causada por tempo e ciclos de uso ou fatores ambientais); atendimento a novos requisitos (por exemplo, alargamento de pista) e mudança em fatores climáticos (COSTA; PROVERBIO; SMITH, 2018; SANCHEZ-SILVA; KLUTKE; ROSOWSKY, 2011). Hui et al. (2015) destacam ainda que tais fatores podem ser agravados em função dos seguintes itens: rápido crescimento populacional, falta de uniformidade e consistência nas fases de projeto, construção e operação, além de programas inadequados na investigação de patologias e manutenção.

Tomando-se como exemplo as pontes, sabe-se que elas representam um grande papel na infraestrutura de transporte. Hou e Wu (2019) destacam que nos últimos 50 anos diversas foram construídas na China. Segundo os autores, entre 2001 e 2017, o seu quantitativo teve aumento de aproximadamente 200%, chegando a 832500 unidades. Estes autores afirmam que gradualmente apareceram problemas nessas construções relacionados a alguns dos fatores citados anteriormente. De acordo com o Relatório de Infraestrutura dos Estados Unidos (*America's Infrastructure Report*), no ano de 2016 quatro em cada dez pontes nos Estados Unidos estavam há mais de 50 anos em operação e que 56007 delas (9,1% do total) possuíam alguma deficiência na sua estrutura. Estima-se que o total necessário para reabilitação das mesmas seja de 123 bilhões de dólares (AMERICAN SOCIETY OF CIVIL ENGINEERS, 2017a). Em relação ao Brasil, não há um levantamento abrangente sobre o estado das pontes. Oliveira, Greco e Bittencourt (2019), com base no relatório de inspeção conduzido pelo DNIT em seu inventário de pontes (4725 unidades) em todo o território nacional, apresentam o

seguinte resultado: 0,2% obra crítica (grave insuficiência estrutural); 4,1% obra problemática (significativa insuficiência estrutural); 38,2% obra potencialmente problemática (alguma insuficiência estrutural); e 57,5% obra sem problemas ou sem problemas importantes. Destaca-se que a classificação utilizada remete a procedimentos presentes nas normas DNIT010/2004-PRO (DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES, 2004) e ABNT NBR 9452 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2019).

Este cenário não se restringe apenas às pontes, mas sim a diversos tipos de construções. No caso de barragens, por exemplo, o Relatório de Infraestrutura dos Estados Unidos estima que 15500 dessas construções estavam com potencial de alto risco e 45 bilhões de dólares deveriam ser empregados para repará-las (AMERICAN SOCIETY OF CIVIL ENGINEERS, 2017b). No Brasil, segundo o Relatório de Segurança de Barragens, das 7257 barragens classificadas quanto ao Dano Potencial Associado (DPA) 66% apresentaram DPA alto. Em relação à Categoria de Risco (CRI) reportou-se 5879 barragens das quais 26% CRI alto (AGÊNCIA NACIONAL DE ÁGUAS E SANEAMENTO BÁSICO, 2020). A situação é ainda mais preocupante ao perceber que acidentes com esse tipo de estrutura são recorrentes, onde os últimos de maior impacto foram nas cidades de Mariana (2015) e Brumadinho (2019), Minas Gerais.

World Economic Forum (2014) destaca que uma solução economicamente mais viável para esta situação é melhorar a utilização, eficiência e longevidade do sistema de infraestrutura atual. Para tanto, evidencia-se a importância do uso de métodos para detectar e quantificar danos a estruturas e, assim, poder acionar programas de restauração da confiabilidade a níveis aceitáveis de projeto (KAMINSKI; RIERA, 1997). Com o intuito de avaliar o comportamento da estrutura durante sua implementação e operação, pode-se empregar o Monitoramento da Integridade Estrutural, MIE (*Structural Health Monitoring*, SHM). Esta engloba o processo de programar e desenvolver uma rede de dispositivos capazes de executar as seguintes tarefas: adquirir dados de sensores espalhados ao longo da estrutura; tratar e processar estes dados; e entregar uma resposta acerca do comportamento da estrutura (ISTE, 2017). Isso tudo podendo ser feito de forma remota.

Em contrapartida aos aspectos positivos da implantação do MIE, observa-se que este não é bem difundido em obras de construção civil. Isso se dá pela complexidade e alto custo empregados na instalação de diversos sensores e interpretação de uma grande quantidade de dados (LYNCH; LOH, 2006; TONDOLO et al., 2018). Nesse contexto, o avanço tecnológico observado na produção de novos sensores e microcontroladores/microprocessadores de baixo

custo, pode ampliar a variedade de opções disponíveis ao MIE. Além disso, a fácil interação destas novas tecnologias ao conceito de IoT (*Internet of Things*) garante um melhor compartilhamento de dados e tomadas de decisão via internet (remotamente) (HOU; WU, 2019). Por outro lado, o uso de tais dispositivos deve cumprir o desafio de garantir uma comunicação robusta entre os sensores e a central responsável pela coleta e processamento de dados. Além disso, busca-se um dimensionamento adequado de sua fonte de energia (baterias) a fim de evitar interrupções no serviço. Desta forma, as novas tecnologias devem se mostrar como ferramentas confiáveis frente aos requisitos exigidos ao desempenho do MIE. (ABRUZZESE et al., 2020).

1.1. Justificativa do Trabalho

A redução da capacidade portante aliada a incertezas em relação à integridade das construções civis pode ocasionar, em casos extremos, a inviabilidade de uso ou até mesmo colapso das mesmas. Como resultado, evidenciam-se danos à manutenção da vida, perdas financeiras, econômicas, ambientais ou uma combinação entre todos. Por conta disso, a identificação de danos a estruturas mostra-se de grande relevância para evitar tais empecilhos. Muitas formas de detecção de danos estão baseadas em métodos visuais ou procedimentos experimentais em âmbito local. Tais procedimentos tem como entraves a necessidade de saber o possível posicionamento do dano e também a dificuldade de acesso até este local. Nesse cenário emerge a necessidade de uma técnica de monitoramento que seja remota e atue em âmbito global, como o Monitoramento da Integridade Estrutural (MIE) (CURADELLI et al., 2008). Desta forma, a viabilidade do monitoramento a longo prazo de um fenômeno de degradação na estrutura possibilitaria o estabelecimento de políticas mais eficientes de manutenção preventiva reduzindo, assim, custos e tempo de reparo. Além disso, a observação da resposta estrutural a eventos extremos (terremotos, furacões e incêndios) orientaria para tomadas de decisões mais rápidas e assertivas a respeito da integridade estrutural. (TONDOLO et al., 2018)

A implantação do MIE encontra algumas barreiras na sua difusão tanto em obras existentes quanto naquelas a serem executadas. Dentre elas destacam-se os custos relacionados a implantação e manutenção do sistema. Como resultado, estas podem limitar o número de sensores empregados, ocasionando uma redução na acurácia da informação coletada para estruturas de grandes dimensões (NOTAY; SAFDAR, 2011). Com o intuito de tornar a utilização do MIE mais atrativa, necessita-se desenvolver dispositivos que empreguem

tecnologias atuais, compactas, de baixo custo e fácil instalação. Além disso, deve-se garantir, acima de tudo, qualidade e manutenção dos dados adquiridos (ABRUZZESE et al., 2020; NOEL et al., 2017).

1.2. Objetivo Geral

Desenvolver, programar e implementar um dispositivo de baixo custo baseado em acelerômetro MEMS a ser aplicado como ferramenta no monitoramento de integridade estrutural.

1.2.1. Objetivos Específicos

- a) Entender o comportamento e configuração do acelerômetro MEMS;
- b) Avaliar os diferentes tipos de comunicação a fim de estabelecer uma conexão confiável e eficiente entre os dispositivos;
- c) Detectar e minimizar as limitações dos dispositivos em relação velocidade de aquisição, processamento de dados e sincronismo;
- d) Programar um sistema para aquisição simultânea dos dados adquiridos dos sensores.

1.3. Organização do Trabalho

O trabalho está dividido em seis capítulos. O presente capítulo, encarrega-se de introduzir o tema e evidenciar os principais tópicos tratados no decorrer do trabalho.

No segundo capítulo, aborda-se o assunto de monitoramento da integridade estrutural. Inicialmente, apresentam-se os níveis para detecção de falhas em estruturas e algumas metodologias. Após isso, chega-se ao conceito de monitoramento da integridade estrutural e sua abrangência. Por fim, dá-se enfoque ao monitoramento por vibração.

O terceiro capítulo trata de sistemas de medição. Inicialmente mostra-se o conceito geral de sistema de medição e pontos de atenção na hora de dimensioná-lo. Posteriormente, parte-se para alguns conceitos-chave na temática que se relacionam com o tema do presente trabalho, tais como: sensores, condicionamento de sinal, conversão do sinal, comunicação digital e calibração.

O quarto capítulo engloba a fase experimental no desenvolvimento do sistema de medição proposto. No seu decorrer, constará a metodologia, resultados e discussões referentes às atividades desempenhadas a fim de avaliar o funcionamento do dispositivo.

No quinto capítulo, apresentam-se os resultados e discussões a respeito dos ensaios estabelecidos no capítulo quatro.

No sexto capítulo consta a conclusão do trabalho. Esta embasa-se nos resultados e discussões abordadas no capítulo cinco. Por fim, mostram-se trabalhos futuros que o autor almeja desenvolver dentro do tema proposto.

2. MONITORAMENTO DA INTEGRIDADE ESTRUTURAL

Segundo Rytter (1993), os métodos de detecção de falha podem ser classificados em quatro níveis de acordo com a extensão do diagnóstico:

- a) Nível 1 – Detecção: permite dizer se há danos na estrutura.
- b) Nível 2 – Localização: apresentam áreas onde o dano ocorreu.
- c) Nível 3 – Quantificação: avaliam a intensidade do dano
- d) Nível 4 – Previsão: permite a previsão da vida útil remanescente da estrutura.

A inspeção visual é uma das formas de detecção de falhas em estruturas mais antiga e usual. Esta representa importante papel na identificação de patologias ou anomalias quando se fala em ensaios não destrutivos. Entretanto, ressalta-se que a qualidade da sua aplicação é diretamente ligada à experiência e perspectiva de quem a executa (RYTTER, 1993). Em complemento, há também a avaliação do comportamento da estrutura por meio de instrumentação (MONTEIRO, 2010). Entretanto, destaca-se que apenas o diagnóstico de falhas por meio da análise de dados oriundos de sensores representa uma visão mais pontual, chamada de Avaliação Não-Destrutiva, AND, (do inglês, *Non-Destructive Evaluation, NDE*).

Ao se pensar em um prognóstico de falhas, precisa-se de um monitoramento mais prolongado (maior aquisição de dados), abrangente (maior distribuição de sensores), robusto (maior processamento de dados) e remoto (melhor transmissão de dados). Desta forma, necessita-se de mais ferramentas para alcançar tal objetivo. Como forma de suprir esta demanda deixada pelo AND, o MIE, engloba as atividades relacionadas a aquisição de dados de resposta, extração de informações e avaliação da segurança estrutural do sistema. Para tanto, empregam-se conhecimentos em sensores, desenvolvimento de hardware, programação, análise e validação estrutural e modelagem numérica (SANTOS, 2014). A Figura 1 exemplifica um esquema utilizado para MIE em pontes.

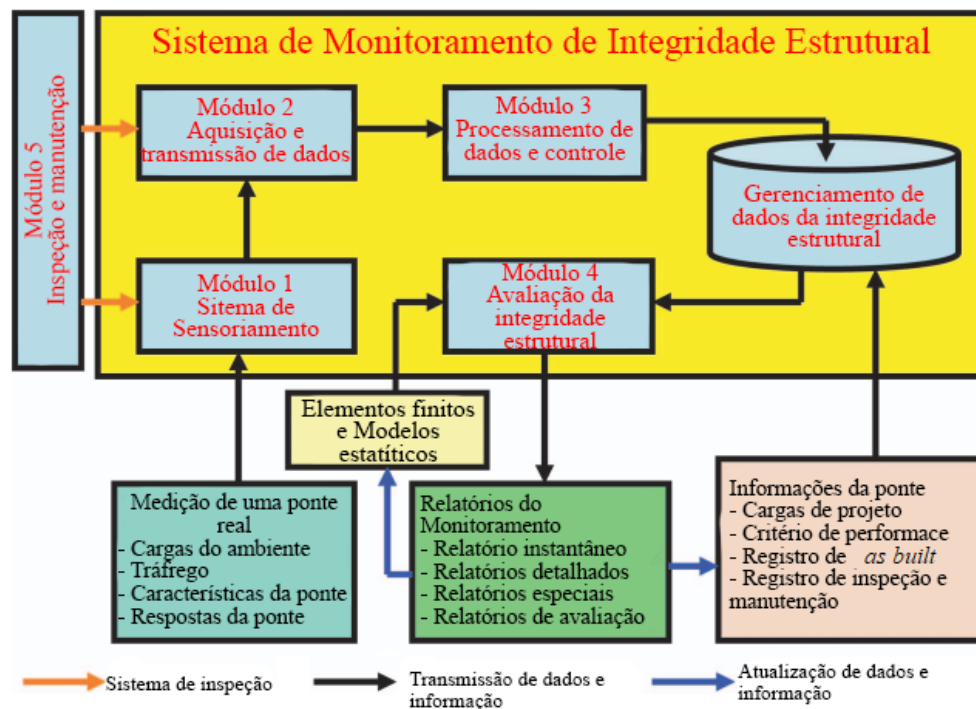


Figura 1: Esquema de um sistema de MIE (baseado em: Wong, 2007)

O MIE pode ser aplicado tanto em âmbito local (elementos individuais e materiais constituintes), quanto global (conjunto de todos os elementos). Além disso, em relação a sua fase de aplicação, pode-se usar tal técnica nas etapas de confecção ou operação dos elementos. No primeiro caso, busca-se verificar e otimizar as propriedades dos materiais constituintes durante processos de preparo e montagem dos elementos (por exemplo, moldagem, transporte e instalação de uma viga pré-moldada de concreto armado). No outro caso, utiliza-se o MIE em estruturas já em operação como forma de assegurar que estas se mantenham no domínio de serviço especificado em projeto. Tal preocupação deve-se a incidência de fatores que modificam a capacidade portante da mesma ao longo do tempo, são eles: divergências entre modelo de cálculo e real; mudanças nas cargas de serviço; deterioração dos materiais utilizados; atendimento a novos requisitos e mudança em fatores climáticos (COSTA; PROVERBIO; SMITH, 2018; SANCHEZ-SILVA; KLUTKE; ROSOWSKY, 2011)

Na literatura, observam-se diversos estudos práticos abordando a área de MIE. Li et al. (2020) monitoraram a deformação de torres eólicas localizadas offshore com o auxílio de sensores FBG (*Fiber Bragg Grating*). Utilizou-se os dados coletados para alimentar um modelo de método dos elementos finitos inverso (*inverse Finite Element Method*, iFEM). O intuito do estudo era ter um controle em tempo real da integridade da estrutura garantindo, assim, maior eficiência na manutenção. Zhou et al. (2020) propuseram um Sistema Detalhado de

Monitoramento da Integridade (*Health State Comprehensive Measurement System, HSCMS*) aplicado a pontes localizadas em uma região de clima subtropical onde se observava variações em temperatura, umidade do ar e também deformações na fundação. Analisaram-se propriedades dos materiais constituintes (resistência do concreto ou potencial de corrosão do aço), dos elementos estruturais (espessura) e da resposta a uma carga estática (deformações). Wang et al. (2020) monitoraram um prédio de 12 pavimentos feito em concreto que utilizava agregado reciclado de construção ou demolição. Mediram-se respostas de aceleração e deformação do edifício, além da temperatura dos ambientes internos.

Tomando como enfoque a engenharia civil, as categorias a serem monitoradas podem ser resumidas por: carga (vento, terremoto e impacto), fatores do meio (temperatura, umidade e acidez), efeitos da carga no âmbito global (aceleração, velocidade, deformação estática e deslocamento dinâmico) e local (fissuras e tensão) e deterioração do material (corrosão, carbonatação e fadiga) (OU; LI, 2010). Percebe-se que o MIE pode trabalhar com diversas grandezas, utilizando-se de variados equipamentos. A escolha dos tipos de sensores (medição) e dispositivos de monitoramento (aquisição e processamento de dados) torna-se uma tarefa de grande importância no dimensionamento de um sistema para MIE (OU; LI, 2010).

Nota-se nos exemplos acima que o MIE pode ser implementado utilizando-se de variados tipos e arranjos de sensores. Quando se trata do dimensionamento de uma rede de sensores, diversos aspectos devem ser levados em conta. O primeiro deles é o tipo de medida, onde atenta-se para seleção do sensor e sua construção. Neste caso, definem-se parâmetros como grandeza a ser medida, alcance da leitura, sensibilidade, resolução, nível de ruído, autonomia, forma dos dados (analógico/digital), entre outros. Dando continuidade, planeja-se o número de sensores e localização. Além disso, define-se a comunicação e transmissão de dados entre as unidades de monitoramento (redes com ou sem fio e protocolos de comunicação). Por fim, tem-se a forma de armazenamento podendo ser local ou remota (MUSTAPHA et al., 2020).

Segundo Ostachowicz, Soman e Malinowski (2019) o MIE deve procurar atender os seguintes requisitos: baixo custo; avaliação contínua da estrutura; sensibilidade a baixos níveis e variados tipos de dano; indiferença a ações de carga do ambiente, ruído e mudanças de condições do meio ambiente. Entretanto, sabe-se que satisfazer todos estes ainda é um desafio para o MIE. Os autores destacam ainda técnicas que são capazes de englobar alguns dos pontos citados acima, são elas: monitoramento baseado em vibração, deformação, ondas elásticas, impedância

eletromecânica e vácuo comparativo (*comparative vacuum monitoring*, CVM, do inglês). O presente trabalho dará destaque à primeira técnica.

A detecção de falhas na estrutura por meio de vibração baseia-se no conceito de que mudanças em propriedades mecânicas da construção (massa, rigidez ou ambos) são refletidas em variações nas características dinâmicas da mesma (KAMINSKI; RIERA, 1997). Rytter (1993) em seu trabalho apresenta o conceito de Indicadores de Dano (*Damage Indicators, DI*) como parâmetros ou representações gráficas que são sensíveis a alterações no comportamento dinâmico. Individualmente, os mesmos atuam como métodos de detecção de falha nível 1. A combinação de alguns *DI*s pode ser utilizada em métodos nível 2 ou 3. O autor exemplifica alguns desses *DI*s como: frequência natural, formas modais, razão de amortecimento, frequências anti-ressonantes, transmissibilidade entre outros. Uma variedade de métodos de análise modal experimental (*Experimental Modal Analysis, EMA*) foram desenvolvidos ao longo dos anos. Estes são usualmente classificados de acordo com os seguintes parâmetros: domínio de aplicação (tempo ou frequência); tipo de formulação (indireto ou modal e direto); número de modos analisados (um ou vários graus de liberdade); e número de entradas e tipo de estimativas (entrada única saída única, entrada única saída múltipla, múltipla entrada múltipla saída, múltipla entrada saída única). Como exemplos de tais métodos pode-se citar Detecção de Picos (*Peak Picking, PP*), Variáveis Instrumentais (*Instrumental Variable, IV*), Método de identificação estocástica de subespaços baseados nas covariâncias (*Covariance-driven stochastic subspace identification, SSI COV*), Método de identificação estocástica de subespaços baseados nas séries temporais (*Data-driven stochastic subspace identification, SSI-DATA*), entre outros (CUNHA et al., 2018).

Destaca-se ainda que a detecção de falhas por meio da análise de vibrações está sujeita a interferências por fatores do meio externo. Xia et al. (2012) e Kaminski e Riera (1997) afirmam que, para maioria dos materiais, um aumento na temperatura reflete uma diminuição do módulo de elasticidade ocasionando, assim, queda na frequência de vibração. Entretanto, o efeito da temperatura pode assumir um comportamento mais complexo, pois afeta fatores como deslocamento em elementos de suporte e variações nas condições de contorno de elemento estruturais. Além disso, o problema pode ser agravado por uma distribuição não uniforme da temperatura ao longo de estruturas com grandes dimensões. Cornwell et al. (1999) estudaram um vão de uma ponte de concreto e detectaram ao longo do dia mudanças na frequência de vibração em torno de 6%. Askegaard e Mossing (1998) estudaram uma ponte em concreto

armado de três vãos durante três anos. Como resultado, constataram variações nas frequências de 10% ao comparar diferentes estações do ano. Por fim, sabe-se que usualmente danos tem início em uma escala local (um elemento). Isto pode representar um empecilho na utilização da técnica baseada em vibração, uma vez que esta é mais sensível à detecção de falhas em nível global (OSTACHOWICZ; SOMAN; MALINOWSKI, 2019).

Observou-se nos últimos anos um notório progresso no MIE devido a avanços tecnológicos ligados a sensores, malha de sensores, aquisição de dados, comunicação, processamento de sinais e gerenciamento de informações. Como exemplo, pode-se citar a tecnologia MEMS (*Micro Eletro Mechanical Systems*). Por meio desta pôde-se desenvolver sensores em escalas reduzidas a baixo preço. Além disso, a difusão de plataformas de código aberto (*open source*) tendo como característica a facilidade de manuseio do *hardware* e *software* proporcionou o desenvolvimento de estudos no campo de sistemas de monitoramento baseados em vibrações. Neste âmbito destacam-se plataformas de desenvolvimento dotadas de microcontroladores, Arduino (ARDUINO, 2018) e ESP32 (ESPRESSIF SYSTEMS, 2021), e microprocessador, Raspberry Pi (RASPERRY PI FOUNDATION, 2018). Na literatura, percebe-se que a utilização de sistemas de monitoramento baseados nesses dispositivos performaram uma aquisição de dados confiável, o que sugere seu emprego no MIE (DUC; ANH; DINH, 2018; HJORT; HOLMBERG, 2015; LO IACONO; NAVARRA; OLIVA, 2017).

3. SISTEMA DE MEDIÇÃO

Um sistema de medição encarrega-se de adquirir informações a respeito de uma grandeza física ou de uma variável a ser medida. Em sua essência, pode ser composto por uma única unidade capaz de fornecer um sinal ou leitura de saída de acordo com a magnitude de uma variável ao qual ele é submetido. Em situações onde a medição se torna mais complexa, utilizam-se variados elementos para desempenhar funções adicionais como pode ser visto na Figura 2. Diferentes tipos de sistemas de medição podem ser encontrados no mercado. Em alguns casos, encontram-se plataformas compostas por conversores analógico-digital (A/D) para aquisição e conversores digital-analógico (D/A) para atuadores. Além disso, estes podem conter armazenamento de dados embutido, unidade de processamento e sistema operacional. Nesses casos, utiliza-se um computador para programar as operações da plataforma e também para enviar e receber informações. A seguir, destacam-se alguns pontos de atenção na escolha de um sistema de aquisição de dados (DERAEMAEKER; WORDEN, 2010; MORRIS, 2001):

- a) nível de ruído do sinal (consideração aos tipos de conversor A/D e D/A);
- b) taxa de aquisição de dados;
- c) presença e tipos de filtros *anti-aliasing*;
- d) a taxa de processamento de dados na unidade de armazenamento;
- e) número de canais de medição;
- f) adequação a condições do meio;
- g) custo de aquisição.

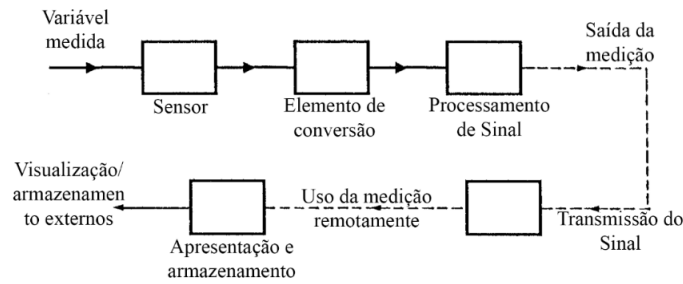


Figura 2: Elementos de um sistema de medição (baseado em: MORRIS, 2001)

3.1. Sensores

De forma ampla, os sensores são responsáveis por mensurar (quantitativamente e qualitativamente) parâmetros físicos, químicos, ambientais e médicos. A rigor, seu funcionamento baseia-se em duas partes: receptor e transdutor. A primeira encarrega-se de reconhecer estímulos por meio das leis da natureza. Ao passo que o transdutor transforma o estímulo captado e o converte em um sinal de saída, Figura 3. Na maioria dos casos o produto final desse processo é um sinal elétrico (HASEEB et al., 2020).

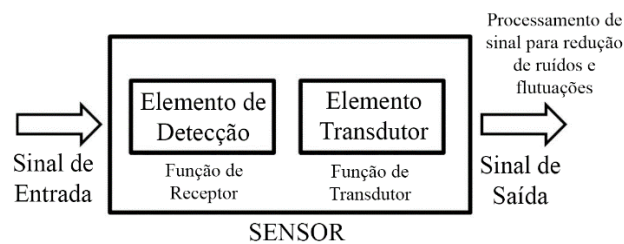


Figura 3: Esquema de funcionamento de um sensor (baseado em: Haseeb et al., 2020)

3.1.1. Acelerômetros

Tomando como foco a análise de vibração, os acelerômetros destacam-se como os sensores mais utilizados. Por existirem variados tipos, o presente trabalho dará enfoque àqueles mais comumente utilizados. O princípio de funcionamento compartilhado por eles, como pode ser visto na Figura 4, baseia-se no conjunto massa-mola-amortecedor fixado a um objeto móvel. Destaca-se que tal objeto deve apresentar massa muitas vezes maior do que a do conjunto evitando-se, assim, interferências em sua resposta dinâmica. As equações (3.1), (3.2) e (3.3) governam o comportamento desse sistema.

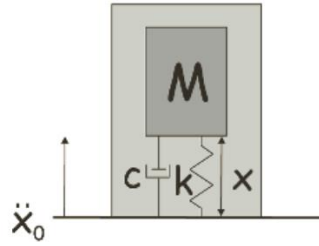


Figura 4: Esquema de funcionamento do acelerômetro (Deraemaeker e Worden, 2010)

No domínio do tempo,

$$M\ddot{x} + c\dot{x} + kx = -M\ddot{x}_0 \quad (3.1)$$

No domínio da frequência, representa-se x/\ddot{x}_0 como,

$$\frac{x}{\ddot{x}_0} = \frac{-1}{-\omega^2 + \omega_n^2 + 2j\xi\omega\omega_n^2} \quad (3.2)$$

Para o caso de $\omega \ll \omega_n$,

$$\frac{x}{\ddot{x}_0} \cong \frac{-1}{\omega_n^2} \quad (3.3)$$

Onde:

M é a massa do conjunto acelerômetro;

x é o deslocamento relativo entre M e a massa do objeto ao qual o acelerômetro foi fixado;

\ddot{x}_0 é a aceleração do objeto monitorado;

ω é a frequência de vibração do sistema;

ω_n é a frequência natural do acelerômetro.

Na Figura 5, percebe-se que a baixas frequências a resposta do acelerômetro segue um comportamento linear em relação ao deslocamento relativo, x . Com o aumento da frequência de excitação do sistema a níveis próximos de ω_n , observa-se uma mudança drástica no comportamento da curva. Dá-se lugar a uma não linearidade que compromete as leituras do sensor. Além disso, pode-se danificá-lo caso a frequência natural seja alcançada sem o emprego de um sistema de amortecimento (MEASUREMENT COMPUTING, 2012).

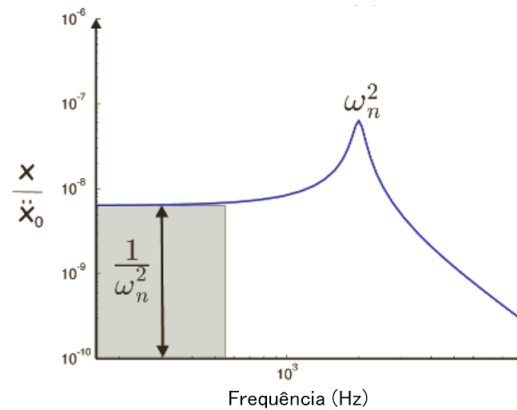


Figura 5: Comportamento em relação à frequência de excitação (baseado em: Deraemaeker e Worden, 2010)

Entendido o modo de operação do acelerômetro, parte-se agora para determinação do valor de x . Diferentes técnicas podem ser utilizadas nesta etapa como piezoresistividade-*Strain gauges*, piezoeletricidade, capacitância, óptica e ressonante. No presente trabalho, dar-se-á destaque às três primeiras.

3.1.1.1. Técnicas Empregadas

Piezoresistividade e *Strain Gauges*

Piezoresistividade é uma característica atribuída aos materiais que mudam sua resistência elétrica quando submetidos a pressão (BEEBY et al., 2004). A sensibilidade (*gauge factor*, GF) desse comportamento é apresentada a seguir na equação (3.4). Nela demonstra-se que dois fatores distintos influenciam no GF. O primeiro, $((d\rho/\rho)/\varepsilon_l)$, refere-se ao efeito piezoelétrico enquanto que o segundo, $(1+2\nu)$, ao efeito geométrico. Sensores que exibem mudança na resistência majoritariamente em função da deformação são chamados de *strain gauges*. Em contrapartida, para aqueles que o efeito piezoelétrico predomina denomina-se piezoresistores (BEEBY et al., 2004). Como exemplo, a Tabela 1 mostra *gauge factors* de alguns materiais, ao passo que na Figura 6 consta um esquema construtivo do *strain gauge*.

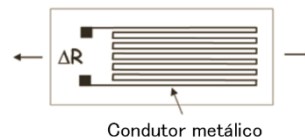
$$GF = \frac{d\rho/\rho}{\varepsilon_l} + (1 + 2\nu) \quad (3.4)$$

Onde ρ é a resistividade elétrica (Ωcm), ε_l a deformação longitudinal do elemento e ν o coeficiente de Poisson.

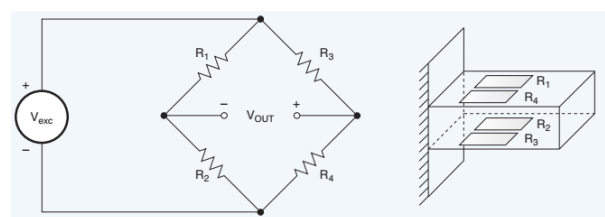
Tabela 1: *Gauge Factor* para diferentes materiais

Material	Gauge Factor
Strain gauge em folha metálica	2 – 5
Folha metálica fina	2
Cristal simples de silício	-125 até +200
Polisilício	± 30
Resistor de filme espesso	10

Fonte: Beeby et al. (2004)

Figura 6: *Strain Gauge* piezoresistivo (baseado em: Deraemaeker e Worden, 2010)

Na construção de alguns acelerômetros, uma massa calibrada é colocada no extremo de uma barra em balanço. Posicionam-se *strain gauges* tanto na face superior quanto inferior da mesma. Desta forma, sob efeito de aceleração perpendicular ao plano dos *strain gauges*, um capta alongamento ao passo que o outro encurtamento (MEASUREMENT COMPUTING, 2012). Quantifica-se esta variação na resistência por meio de uma ponte de Wheatstone. Na Figura 7, ilustra-se a instalação e funcionamento do dispositivo. Deraemaeker e Worden (2010) ressaltam que a posição dos *strain gauges* varia entre os diferentes tipos de sensor, podendo assumir a função de mola do conjunto. Além disso, os autores destacam que uma das vantagens desse tipo de acelerômetro está na possibilidade de medir a 0 Hz.

Figura 7: Acelerômetro *strain gauge* (MEASUREMENT COMPUTING, 2012)

Piezoelasticidade

Materiais piezoelétricos são aqueles capazes de produzir uma carga elétrica superficial em consequência da aplicação de um esforço mecânico. Tal comportamento segue o efeito piezoelétrico direto, onde a quantidade de carga gerada é proporcional ao estresse aplicado no material (UCHINO, 2017). Como forma de intensificar a resposta elétrica, esses materiais são polarizados em uma direção desejada. O sinal gerado a partir de acelerômetros que utilizam

esses materiais é intensificado por meio de amplificadores. Este processo não é ideal para medições estáticas, porém pode trabalhar em altas frequências na ordem de MHz (DERAEMAERKER; WORDEN, 2010). A Figura 8 mostra alguns modos de deslocamento utilizados para gerar um sinal elétrico.

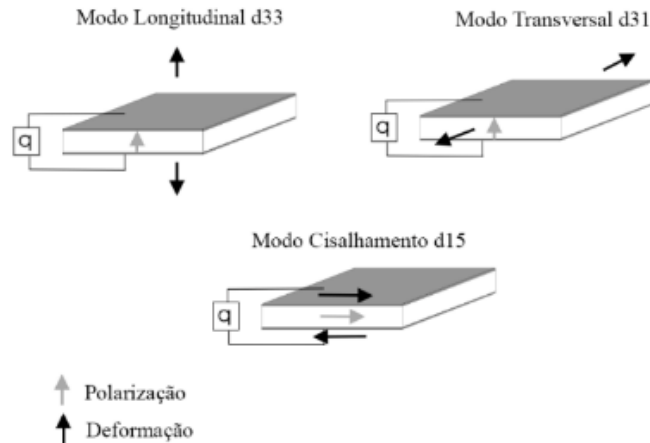


Figura 8: Alguns modos de deformação para gerar carga elétrica (baseado em: Deraemaeker e Worden, 2010)

Capacitância

Capacitância é a uma grandeza física que expressa a quantidade de energia elétrica pode ser armazenada em um capacitor. Sensores que utilizam a técnica da capacitância, trabalham com um ou mais eletrodos fixos e outro em movimento. Beeby et al. (2004) destacam que, apesar do emprego dessa técnica ser relativamente simples, ela apresenta uma medida precisa de movimentação. Por meio da equação (3.5), calcula-se a capacitância para uma um conjunto de placas paralelas, ignorando-se interferências de campo magnético externo.

$$C = \frac{\epsilon_0 \epsilon_r A}{d} \quad (3.5)$$

Onde,

ϵ_0 é a permissividade de um espaço livre;

ϵ_r é a permissividade relativa do material entre as placas;

A é a área de sobreposição das placas; e

d a distância entre os dois elementos.

A Figura 9 demonstra alguns exemplos de deslocamento entre as placas. Nota-se que no primeiro caso (a), altera-se o valor de d ocasionando um comportamento não linear de resposta.

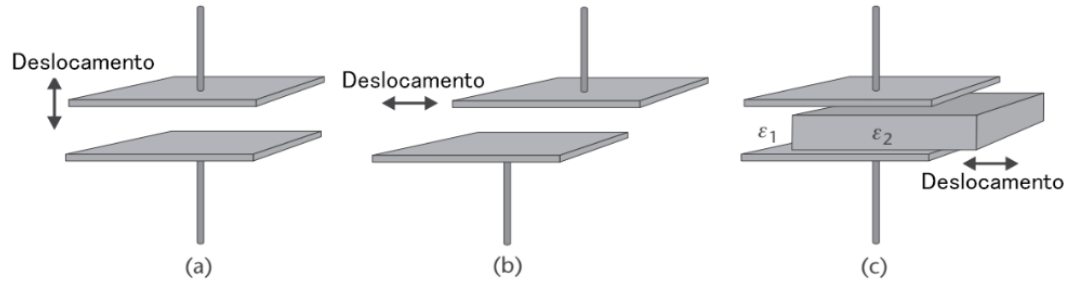


Figura 9: Exemplo de deslocamento de placas em sensores unidirecionais. (a) variação em d ; (b) variação em A ; (c) movimentação de um material dielétrico (baseado em: Beeby et al., 2004).

Um problema comum a essa técnica é sua sensibilidade a mudanças na temperatura, uma vez que isso afeta todos os parâmetros da equação (3.5). Como forma de compensação, usa-se condicionamento do sinal de resposta ou, preferencialmente, adequações à disposição das placas. Neste último caso, pode-se utilizar sensores baseados em capacitância diferencial, como ilustrado na Figura 10. Sabendo-se que todas as placas se submetem a mesma variação em temperatura, este método elimina tal empecilho. Percebe-se na equação (3.6) que esse arranjo se comporta de forma linear enquanto $|x| < d$.

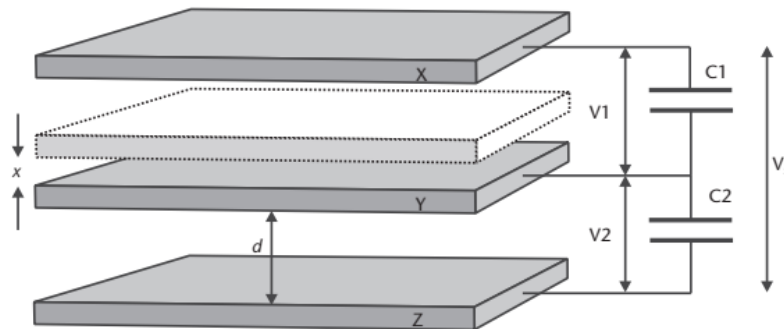


Figura 10: Sensor de capacitância diferencial (Beeby et al., 2004)

$$(V_2 - V_1) = V_s \left(\frac{x}{d} \right) \quad (3.6)$$

Onde V_s é a fonte de alimentação.

Por fim, a técnica baseada na capacitância está sujeita a menos ruído quando comparada com àquela baseada em piezoresistência. Em consequência da ausência de ruído térmico, também chamado de ruído Johnson.

3.1.1.2. Sensores MEMS

Em função da revolução tecnológica dos circuitos integrados (décadas de 60 a 80), observaram-se inovações nas áreas de processos construtivos, equipamento e materiais. Como consequência, o desenvolvimento de dispositivos MEMS (*Micro Electro Mechanical System*) apresentou um expressivo crescimento na década de 90. MEMS são dispositivos resultantes da combinação de componentes mecânicos e elétricos em escala micro, produzidos usando processamento em lote para circuitos integrados (*batch-fabricated*). Este último fator garante uma redução expressiva no seu preço, tornando-se um diferencial em relação a outras tecnologias (MOHAMED, 2006). Acelerômetros MEMS apresentam-se comercialmente em diferentes formatos e características. A forma mais comumente aplicada para determinar o deslocamento diferencial, x , é por meio de mudanças na capacitância. Na Figura 11 pode-se observar um esquema construtivo dessa técnica.

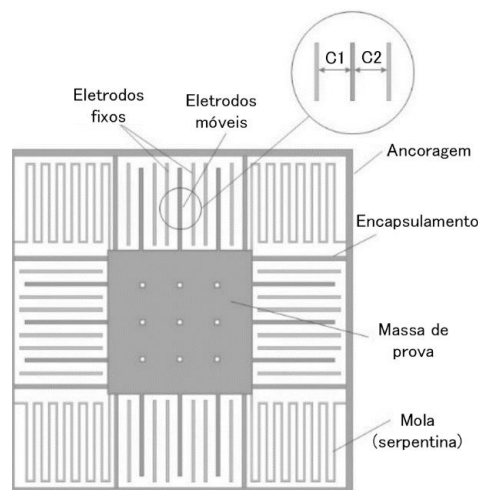


Figura 11: Ilustração de um acelerômetro MEMS triaxial (baseado em: Warsi et al., 2019).

3.1.1.3. Parâmetros de Escolha

A escolha do dispositivo mais adequado ao monitoramento engloba a definição de diversos parâmetros. Mostram-se abaixo alguns deles (ALBARBAR; TEAY, 2017; WARSI et al., 2019):

- a) **Sensibilidade** é a razão entre a resposta elétrica em função da excitação mecânica de entrada. Usualmente, representa-se pela voltagem por unidade de aceleração. Este é um dos fatores principais na escolha de um acelerômetro para MIE. A Figura 12 representa graficamente o conceito de sensibilidade de um sensor;

- b) **Resolução** é a menor variação do sinal de entrada que resultará em uma variação mensurável no sinal de saída;
- c) **Amplitude** representa o alcance máximo de aceleração que pode ser medido;
- d) **Faixa de frequência (Bandwidth)** é o intervalo de operação onde a sensibilidade do sensor matem-se estável. A definição dessa faixa é ligada a características elétricas e mecânicas do transdutor ou dos equipamentos auxiliares do sensor;
- e) **Limite de impacto (shock tolerance)** é o valor máximo ao qual o acelerômetro pode ser submetido sem que ocorra nenhum dano a sua estrutura;
- f) **Linearidade** indica a acurácia da medida de aceleração respeitando a faixa de frequência ideal. A Figura 13 ilustra os comportamentos de um acelerômetro com amplitude $\pm 2g$, onde a linha cheia em preto representa não linearidade na leitura;
- g) **Custo** está diretamente ligado a viabilidade do MIE;
- h) **Conectividade** relaciona-se como a forma que o sensor se comunica com a rede de monitoramento. A classificação principal para esse parâmetro é conectividade via cabo ou wireless;
- i) **Manutenção** implica em um esforço para manter o sensor em operação. Altos níveis de manutenção geram maiores custos ao monitoramento, podendo inviabilizá-lo;
- j) **Consumo de energia** é uma das mais importantes condicionantes para dimensionar a autonomia de sensores wireless.

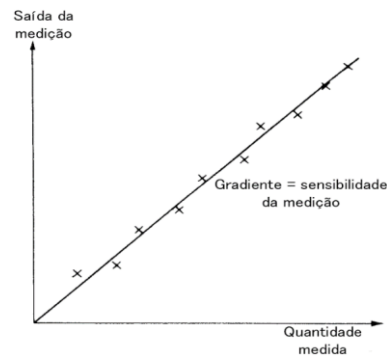


Figura 12: Representação da sensibilidade de um sensor (baseado em: Morris, 2001)

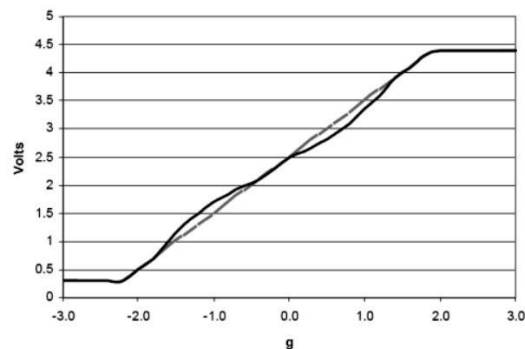


Figura 13: Exemplo de não linearidade em um acelerômetro (baseado em ANALOG DEVICES, s.d.)

3.2. Condicionamento do Sinal

Após a fase de captação e transformação da variável medida, que está a cargo do sensor/transdutor, parte-se para os estágios intermediários de adequação do sinal. Neles há processos que o modificam a fim de facilitar sua interpretação e manuseio. Dentre estes processos, destacam-se a amplificação e a filtragem. Em diversos casos, o sinal originado do transdutor é de pouca intensidade inviabilizando sua leitura em outros dispositivos do sistema de medição. Desta forma, usam-se amplificadores como forma de contornar este empecilho. Holman (2012) destaca ainda que se deve conhecer e respeitar as características operacionais dos amplificadores, caso contrário o sinal pode sofrer distorções na frequência e amplitude. Dando destaque agora à filtragem, este procedimento visa aproveitar no sinal adquirido apenas uma faixa de frequência específica. A parte rejeitada pode ser caracterizada como ruído e sua ausência provoca melhorias na qualidade dos dados.

A curva de resposta de um filtro descreve como ele se comporta. Nela relaciona-se a frequência versus a razão de atenuação (sinal de saída / sinal de entrada), geralmente dada em decibel (dB). Na Figura 14 mostram-se as curvas de resposta para quatro modos de operação dos filtros onde

os limites das faixas são definidos por uma atenuação de -3dB. O primeiro caso (a), chamado de filtro passa-alta, atenua frequências abaixo de f_c ; o segundo (b), filtro passa-baixa, atenua frequências acima de f_c ; em (c), têm-se o caso de um filtro passa-faixa onde frequências fora da faixa f_1 - f_2 são atenuadas; em (d) o filtro rejeita-faixa trabalha de forma oposta àquela vista em (c) (HOLMAN, 2012). Por fim, ressalta-se que diferentes modos de atuação e técnicas podem ser empregados no projeto de um filtro. Um dos pontos de atenção nesse processo é a mudança de fase (*phase shift*) que este induz no sinal adquirido.

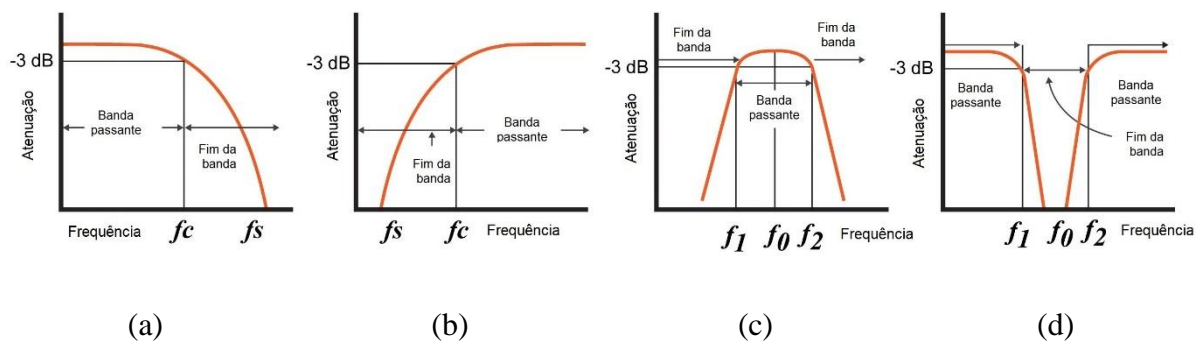


Figura 14: Tipos de filtros em relação ao modo de atenuação (baseado em ALL ABOUT CIRCUITS, 2017)

3.3. Conversão do Sinal

Outra fase do sistema genérico de medição refere-se à leitura e armazenamento dos dados, além do controle da variável medida. Neste estágio, o sinal pode se apresentar na forma analógica ou digital. O primeiro caracteriza-se por ter um comportamento contínuo na representação dos dados. Por outro lado, o sinal digital distribui os dados de forma discreta (por incrementos) dentro de uma faixa finita de valores. A representação de um sinal analógico em digital está a cargo do conversor analógico-digital. Por meio deste pode-se transformar um valor analógico de voltagem em uma sequência binária aproximada. Como forma de alcançar um resultado mais próximo do sinal analógico, deve-se atentar para acurácia e resolução desses dispositivos (MEASUREMENT COMPUTING, 2012). Os sinais digitais podem ser interpretados por computadores onde os dados serão manipulados, analisados e armazenados de forma automática. Na Figura 15, observa-se o comportamento ideal de um conversor analógico-digital.

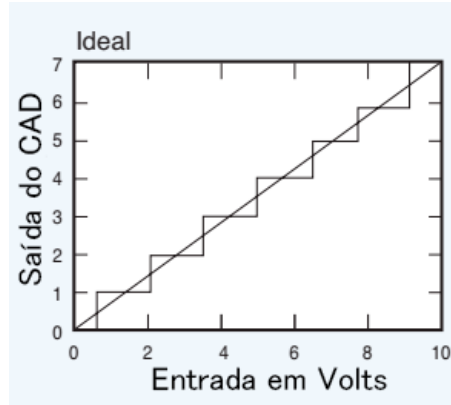


Figura 15: Comportamento ideal de um conversor A/D (baseado em: Measurement Computing, 2012)

A transformação de um sinal digital em analógico é feita pelo conversor digital-analógico. Pode-se empregar tal processo como forma de controle de uma variável física mensurada no sistema. Isto ocorre quando há a implementação de um sinal de *feedback*, como pode ser visto na Figura 25 na seção 3.5.

Durante o processo de conversão analógico-digital amostras do sinal são tomadas diversas vezes por segundo, dando-se o nome de taxa de amostragem. Esta propriedade é de grande relevância na representação de uma onda tanto no domínio do tempo quanto da frequência. Por exemplo, a medição de um sinal de alta frequência requer uma taxa de amostragem proporcionalmente maior. O teorema de Nyquist determina que a aquisição de um sinal a uma frequência máxima (f_{max}) requer uma taxa de amostragem igual ou superior a $2 f_{max}$. Measurement Computing (2012) ressalta que, ao trabalhar no domínio da frequência, é uma prática comum a taxa de aquisição estar entre 5 e 10 vezes o valor de f_{max} .

Quando a aquisição de um sinal é feita a uma taxa de amostragem menor do que aquela definida no teorema de Nyquist, observam-se distorções ao sinal original. Como resultado, aparecem frequências abaixo daquelas realmente medidas. A esse fenômeno dá-se o nome de *aliasing*. A Figura 16 (a) mostra o exemplo de uma onda senoidal de 1 kHz sendo adquirida a uma taxa de amostragem igual a 800 Hz. Como resultado, observa-se a representação da curva a uma frequência inferior ao que esperava. Por outro lado, na Figura 16 (b) a taxa de aquisição é de 5 kHz. Neste caso, nota-se uma representação mais fiel ao sinal medido.

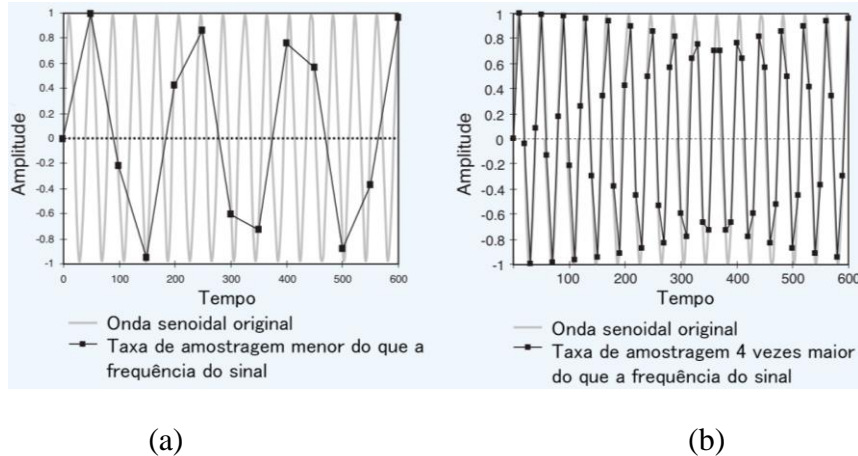


Figura 16: Aquisição de sinal a diferentes taxas de amostragem (baseado em MEASUREMENT COMPUTING, 2012)

É importante notar que frequências acima da metade da taxa de amostragem, também geram *aliasing*, uma vez que não se enquadram naquilo proposto no teorema de Nyquist. Como forma de evitar essa distorção, adota-se um filtro passa-baixa no sinal de entrada com o intuito de atenuar tais frequências.

Na Figura 17, apresenta-se de forma ilustrativa o efeito do *aliasing* no domínio da frequência. No item (a) representa-se a leitura de uma onda quadrada a uma frequência de f igual a 500 Hz e taxa de aquisição máxima igual a 4 kHz (f_{max}). Tal forma de onda é composta por harmônicos nos valores de $1f$, $3f$, $5f$, $7f$, etc. Desta forma, observa-se a aquisição do sinal a frequências acima de $f_{max}/2$ resultando em distorções no gráfico na forma de pequenos picos. A fim de contornar tal situação, emprega-se um filtro passa-baixa configurado a uma frequência de corte (f_c) de 2 kHz, ilustrado no item (b). Nota-se uma melhora na representação do sinal atribuída à remoção dos picos formados pelo *aliasing*. Desta forma, evidenciam-se os picos na frequência fundamental e harmônicas.

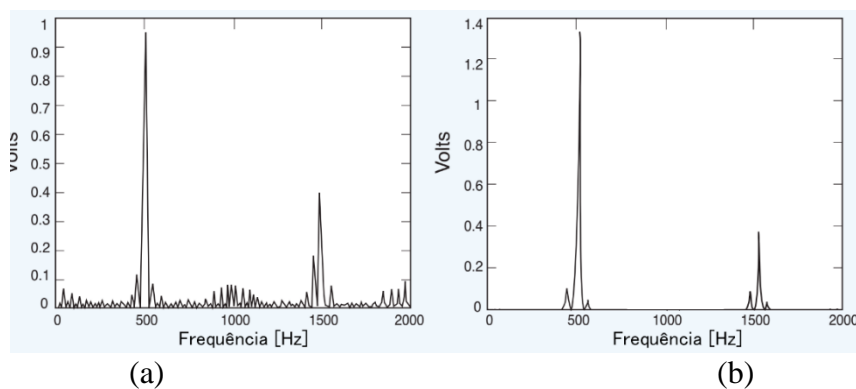


Figura 17: Representação de onda quadrada (baseado em: Measurement Computing, 2012)

3.4. Comunicação Digital

Existem diversas interfaces de comunicação que podem ser empregadas para troca de dados entre dispositivos. A seguir destacam-se aquelas empregadas no desenvolvimento do presente trabalho.

SPI

SPI (*Serial Peripheral Interface*) é uma das mais utilizadas interfaces de comunicação entre microcontroladores/microprocessadores e dispositivos periféricos como sensores, CAD e CDA. Seu funcionamento é baseado em uma comunicação síncrona e *full-duplex* entre o *master* e *slave*. O primeiro termo refere-se à definição de uma frequência de operação (*clock*) fornecida pelo master. A representação dos níveis lógicos (0 ou 1) na transferência de dados está ligada ao estado de subida ou descida no sinal do *clock*. Desta forma, a velocidade dos dados é relacionada ao *clock*, podendo esta assumir alguns MHz. Ao termo *full-duplex*, entende-se como o envio simultâneo de dados ente *master-slave*. Por fim, é relevante notar que o SPI restringe a existência de apenas um *master* ao passo que o mesmo não se aplica ao número de *slaves* (DHAKER, 2018).

Para que as características citadas sejam cumpridas, utilizam-se quatro fios como pode ser visto na Figura 18:

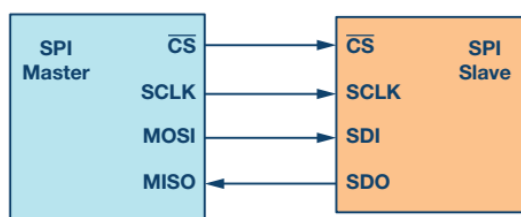


Figura 18: Comunicação SPI (DHAKER, 2018)

SCLK refere-se ao *SPI clock*. Por meio deste, o sinal de *clock* é imposto a todos os *slaves*. CS ou *Chip Select* tem a função de habilitar a comunicação com um determinado *slave* por vez. Geralmente o nível lógico para isto é 0. No caso de múltiplos *slaves*, cada um destes deve ter sua conexão CS_n, Figura 19. MOSI (*Master Out Slave In*) e MISO (*Master In Slave Out*) encarregam-se de transferir dados nos sentidos *master-slave* ou *slave-master*, respectivamente (GUPTA, 2019).

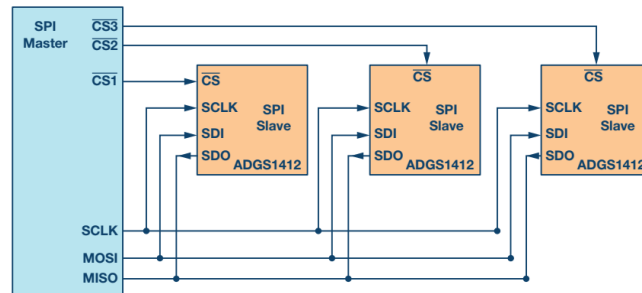


Figura 19: SPI com vários *slaves* (DHAKER, 2018)

UART

UART (*Universal Asynchronous Receiver Transmitter*) é um método de comunicação serial de grande utilização baseada no envio assíncrono de dados. Ou seja, não utiliza o sinal de *clock* compartilhado entre *master* e *slave*. O termo comunicação serial refere-se ao envio de bits em série, como pode ser visto na Figura 20. Este processo destaca-se em relação a comunicação paralelo (Figura 21) por utilizar apenas um canal para transferência de dados, tornando o sistema mais enxuto e eficiente.

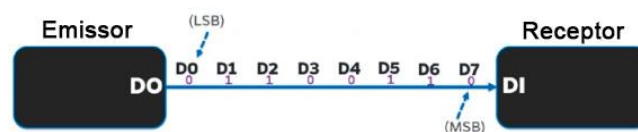


Figura 20: Comunicação serial (baseado em: GUPTA, 2019)

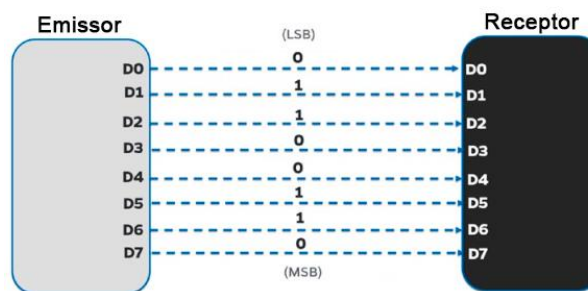


Figura 21: Comunicação paralela (baseado em: GUPTA, 2019)

Por conta das características citadas acima, o envio de um pacote de dados depende da definição de alguns parâmetros, tais como:

- a) bit inicial – define que o início da mensagem é o próximo bit na sequência;
- b) mensagem – é o dado que se deseja transferir;

- c) bit de paridade – é uma técnica para detecção de erro na transmissão dos dados por meio da contagem de dos valores lógicos (1 ou 0);
- d) bit de parada – ao contrário do bit inicial, sua função é sinalizar o fim do pacote de dados;
- e) baudrate – em função da ausência de clock, os dispositivos devem seguir a mesma taxa de transferência de bits por segundo (baudrate). Desta forma, quem recebe a mensagem consegue compreender o que está sendo enviado, sem perda de dados ou má interpretação.

Grande parte dos dispositivos usa a configuração 8N1 onde o primeiro número representa o tamanho da mensagem (8 bits), “N” mostra que não há nenhum bit de paridade e “1” é a quantidade de bits que sinalizam parada. Além disso, também é comum a utilização dos seguintes *baud rates*: 9600, 38400, 19200, 57600 e 115200. Valores diferentes podem ser adotados em função da aplicação ou limitações dos dispositivos. Na Figura 22, representam-se os parâmetros envolvidos no envio de um pacote de dados.

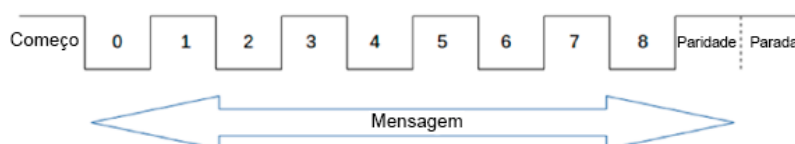


Figura 22: Estrutura do pacote de dados UART (baseado em: GUPTA, 2019).

RS-485

TIA/EIA-485 ou, mais popularmente conhecido, RS-485 é um dos padrões utilizados na interface de comunicação UART (GLORIA; CERCAS; SOUTO, 2017). Desde a sua aprovação em 1993 na EIA (*Electronics Industries Association*), vem sendo amplamente empregado nos ramos da indústria, medicina e aplicações domésticas. Isso deve-se principalmente a alguns fatores como: baixo custo de implementação, boa velocidade de transmissão (10 Mbps a 12m), capacidade de estabelecer uma malha de comunicação entre dispositivos e robustez na transferência de dados a longas distâncias (máximo de 1200m a 100kbps). Somando-se a isso, ainda há a possibilidade de trabalhar em modo *full-duplex* (AXELSON, 2000; KUGELSTADT, 2016).

Diferentemente de uma interface de comunicação, como SPI, que define especificações funcionais, mecânicas e elétricas, o RS-485 refere-se exclusivamente a um padrão elétrico. Em

outras palavras, ele estipula um método onde características elétricas específicas são compartilhadas entre os dispositivos (módulos e receptores) (KUGELSTADT, 2016). Em tal padrão o módulo responsável pelo envio da mensagem gera dois sinais opostos, um em cada canal (A e B). A interpretação fica a cargo do receptor que irá analisar a tensão diferencial entre os dois canais. De acordo com o valor resultante, sabe-se o estado lógico correspondente. Na Figura 23, item (b), exemplifica-se o comportamento dos sinais citados acima. O item (a) mostra a variação no valor diferencial de tensão à medida que o sinal dos canais é invertido. Por fim, a Figura 24 apresenta a análise do sinal diferencial na forma de um pacote de dados no formato serial.

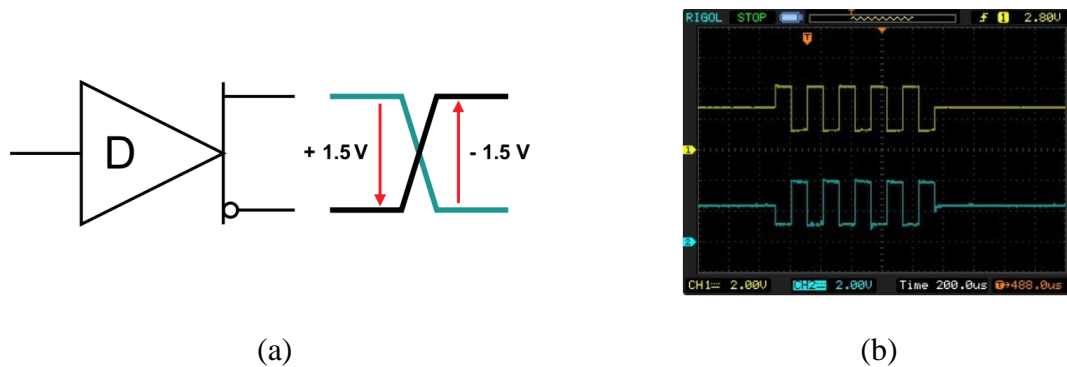


Figura 23: Sinais de saída do módulo RS-485 (a-KUGELSTADT, 2016); b-SEALEVEL, 2012).

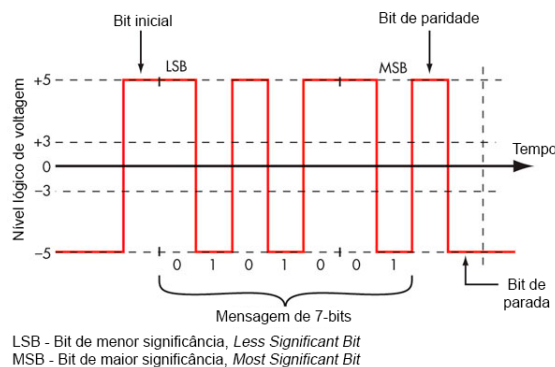


Figura 24: Interpretação do valor diferencial (baseado em: FRENZEL, 2013).

3.5. Calibração

Após a construção do sistema de medição, é de grande importância verificar seu correto funcionamento. Para tanto, utiliza-se uma fonte de sinal pré-definida ligada à entrada do sistema. Para atestar o correto funcionamento do conjunto, o valor apresentado na saída deve ser o mais próximo possível do que foi aplicado à entrada do sistema. A esse processo dá-se o

nome de calibração. Como resultado, todas as fases do sistema de medição mostradas anteriormente serão verificadas e, assim, corrigidas ou aprimoradas. Na Figura 25, têm-se o funcionamento generalizado de um sistema de medição.

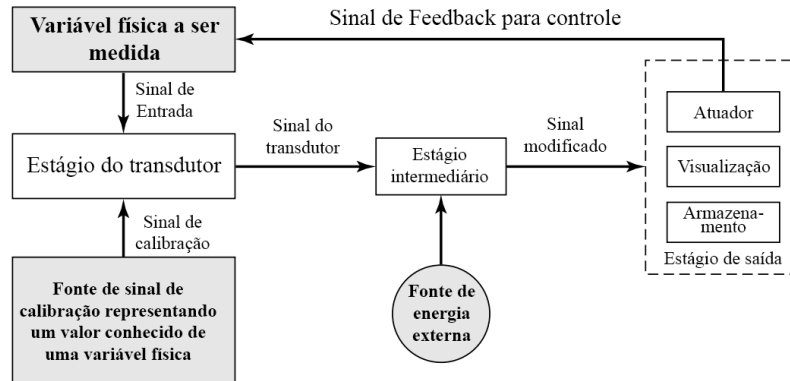


Figura 25: Esquemático de um sistema de medição generalizado (baseado em: HOLMAN, 2012)

4. SISTEMA DE MEDIÇÃO PROPOSTO PARA O MONITORAMENTO DINÂMICO DE ESTRUTURAS

O presente capítulo mostrará a fase experimental do desenvolvimento de um sistema de medição de baixo custo a ser empregado no monitoramento dinâmico de estruturas. No seu decorrer, constarão os materiais e metodologias empregados.

4.1. Materiais Empregados

Para o sistema de monitoramento proposto utilizaram-se os seguintes materiais:

4.1.1. MPU-6500

O MPU-6500 é um módulo de múltiplos *chips* (*Multiple-Chip Module, MCM*) desenvolvido pela TDK *InvenSense* e composto por um bloco de circuito integrado. Neste pacote, encontram-se um acelerômetro e um giroscópio, ambos medindo nos três eixos (X, Y e Z). O MPU-6500 apresenta pequenas dimensões, 3x3x1mm, e baixo custo de aquisição (TDK INVENSENSE, 2020). Utilizou-se no presente trabalho o conjunto sensor mais regulador de tensão (5 – 3,3 V) montados em uma placa de circuito impresso, Figura 26.



Figura 26: MPU-6500 e regulador de tensão (autoria própria)

Na função acelerômetro, suas amplitudes de medida podem ser programadas para $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$. Há ainda as opções de programar um filtro passa baixa e a taxa de aquisição dentro dos intervalos de 5 a 460 Hz e 3,91 a 4000 Hz, respectivamente. Em relação à taxa de ruído, o fabricante explicita uma densidade de ruído de $220 \mu g/\sqrt{Hz}$. No caso do giroscópio, pode-se escolher as seguintes escalas: $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$ e $\pm 2000^\circ/s$. Assim como no acelerômetro, programam-se o filtro passa baixa e a taxa de aquisição dentro dos intervalos de 5 a 3600 Hz e 1 a 32 kHz, respectivamente. De acordo com o fabricante, a 10Hz observou-se uma taxa de ruído equivalente a $0,005 \text{ }^\circ/s/\sqrt{Hz}$ no espectro de potência. Além disso, em cada um dos 6 canais de medição há um conversor analógico-digital de 16 bits. Em relação às interfaces de comunicação, o MPU-6500 oferece a possibilidade de trabalhar com I²C (até 400

kHz) ou SPI (até 20 MHz) (TDK INVENSENSE, 2020). No presente trabalho, utilizou-se a segunda opção por ser mais rápida. Por fim, a sua corrente elétrica de operação para um modo de operação normal (giroscópio e acelerômetros ativados) é de aproximadamente 4,1mA, segundo especificações técnicas do fabricante.

Na Figura 27, mostra-se o digrama de blocos para o MPU-6500. Maiores informações sobre esse sensor encontram-se nos Apêndices APÊNDICE 1 e APÊNDICE 2.

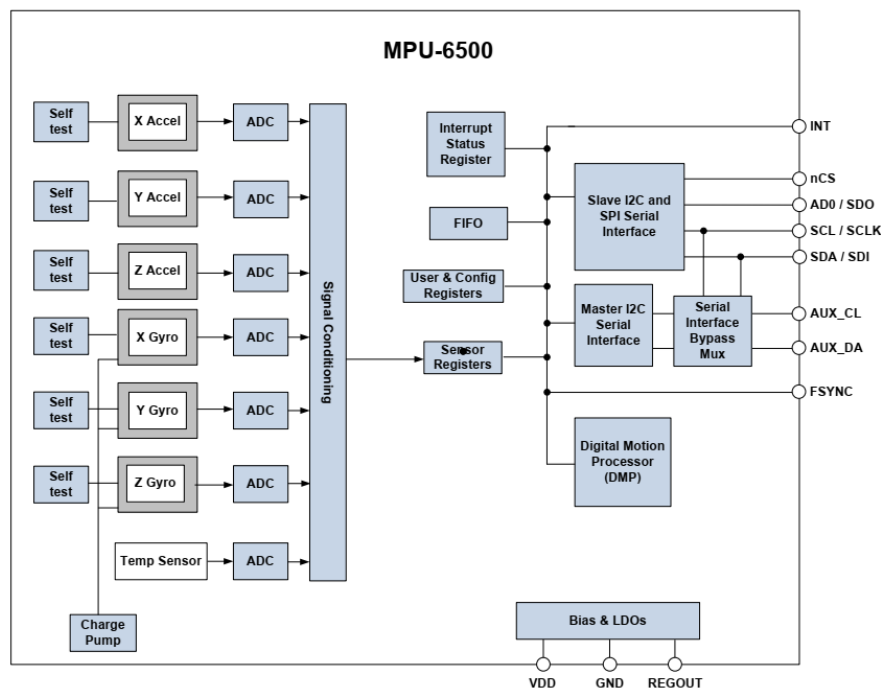


Figura 27: Diagrama de blocos do MPU-6500 (TDK INVENSENSE, 2020).

4.1.2. ESP-32

Consiste em um microcontrolador wireless (*MCU – Micro Controller Unit*) de baixo custo do fabricante Espressif. Ele é composto por um processador duplo Xtensa 32-bit LX6 capaz de alcançar até 240 MHz. Em relação ao consumo de energia, as especificações técnicas do fabricante indicam 500mA. Além disso, este dispositivo possibilita a utilização de diversas interfaces periféricas como: WiFi, Bluetooth, UART, GPIO, I2C, I2S e SPI (ESPRESSIF SYSTEMS, 2021). Por conta dessas características, o mesmo tornou-se bastante utilizado em projetos IoT (*Internet of Things*).

No presente trabalho, utilizou-se o modelo ESP-32-WROOM-32U montado na placa Devkitc_v4. Neste conjunto, consta um regulador de tensão (5 – 3,3 V) e o chip para comunicação USB. Empregou-se este módulo para gerenciar o MPU-6500, definindo sua

configuração inicial de funcionamento e também a taxa de aquisição de dados. Além disso, o mesmo encarregou-se de enviar os dados ao Raspberry Pi. Por fim, a Figura 28 mostra a placa utilizada no presente trabalho.

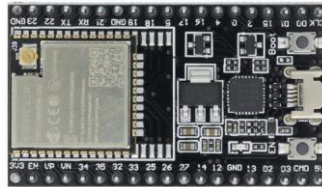


Figura 28: Módulo ESP-32 (autoria própria).

4.1.3. Raspberry Pi

Consiste em uma plataforma de desenvolvimento composta por um microprocessador de baixo custo desenvolvido pela Fundação Raspberry Pi no Reino Unido. De forma análoga ao ESP-32, este possui diversas interfaces de comunicação (Wifi, UART, GPIO, I2C, I2S e SPI). Em contrapartida, o Raspberry Pi funciona por meio de sistemas operacionais baseados em Linux, onde o Raspbian destaca-se como mais utilizado. O modelo empregado no presente trabalho foi o Raspberry Pi 3 B+. Algumas de suas características são: processador 1.2 GHz 64-bit *quad-core* ARMv8, 1GB de memória RAM (RASPBERRY PI FOUNDATION, 2018). Escolheu-se este microcomputador para aquisição e processamento dos dados provenientes do ESP-32. Na Figura 29, observam-se tanto a placa utilizada como sua pinagem.



(a)

3v3 Power	2	5v Power
GPIO 2 (I2C1 SDA)	3	5v Power
GPIO 3 (I2C1 SCL)	5	Ground
GPIO 4 (GPCLK0)	7	GPIO 14 (UART TX)
Ground	9	GPIO 15 (UART RX)
GPIO 17	11	GPIO 18 (PCM CLK)
GPIO 27	13	Ground
GPIO 22	15	GPIO 23
3v3 Power	17	GPIO 24
GPIO 10 (SPI0 MOSI)	19	Ground
GPIO 9 (SPI0 MISO)	21	GPIO 25
GPIO 11 (SPI0 SCLK)	23	GPIO 8 (SPI0 CE0)
Ground	25	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	GPIO 1 (EEPROM SCL)
GPIO 5	29	Ground
GPIO 6	31	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	Ground
GPIO 19 (PCM FS)	35	GPIO 16
GPIO 26	37	GPIO 20 (PCM DIN)
Ground	39	GPIO 21 (PCM DOUT)

(b)

Figura 29: Raspberry Pi 3 B+ (a) modelo utilizado; (b) pinagem do modelo (a - autoria própria; b - Raspberry Pi Foundation, 2018).

4.1.4. Módulo RS-485

Utilizou-se no presente trabalho um módulo de baixo custo capaz de estabelecer uma comunicação diferencial no padrão RS-485, Figura 30. A tensão de operação empregada foi de 3,3V tendo em vista que este é o mesmo valor aplicado a todos os outros dispositivos. Além disso, segundo especificações do fabricante, seu consumo de energia é de 300 μ A.



Figura 30: Módulo RS-485 (autoria própria)

O módulo constitui-se dos seguintes componentes (MAXIM, 2003):

- a) RS485MAX – transceptor de baixo consumo, produzido pela empresa Maxim e aplicado à comunicação RS-485;
- b) R1, R2, R3 e R4 – resistores *pull-up* de 10 k Ω posicionados na linha de dados;
- c) R5 e R6 – resistores de 20 k Ω posicionados um em cada canal (A e B) com a função de forçar um estado de tensão conhecido no período em que a rede não estiver transmitindo dados;
- d) R7 – resistor de 120 Ω colocado entre os canais A e B como forma de impedir reflexão do sinal por diferença de impedância no cabo.

As conexões desse módulo são as seguintes:

- a) RO (*Receiver Output*) – liga-se ao pino serial Rx do microcontrolador/microprocessador;
- b) RE (*Receiver Enable*) – quando em estado lógico 0, faz o módulo trabalhar como receptor;
- c) DE (*Driver Enable*) – quando em estado lógico 1, faz o módulo trabalhar como transmissor;
- d) DI (*Driver Input*) – liga-se ao pino serial Tx do microcontrolador/microprocessador;
- e) VCC e GND – alimentação do módulo (utilizou-se 3,3 V); e

f) A e B – canais de dados opostos.

Tanto os dispositivos quanto os conectores estão representados no esquema de ligação na Figura 31.

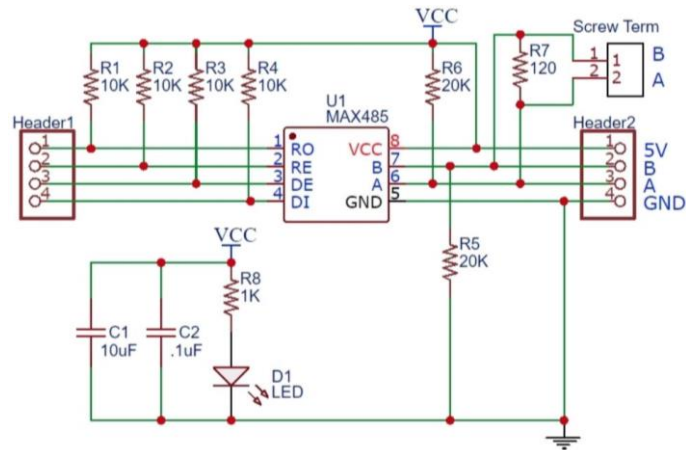


Figura 31: Esquema de ligação (PROTOSUPPLIES, 2020)

4.1.5. Módulo USB-TTL

Emprega-se este módulo, Figura 32, para conversão entre o padrão de comunicação empregado nos microcontroladores (UART ou TTL) e aquele empregado nas portas USB em microprocessadores como o Raspberry Pi (RS-232, *Recommended Standard 232*). Este módulo utiliza o *chip* PL-2303 produzido pela empresa Prolific. Segundo fabricante, seu funcionamento requer 24mA. No presente trabalho empregou-se o módulo USB-TTL como uma forma de ampliar o número de portas seriais (UART) no Raspberry Pi 3B+, uma vez que o mesmo possui apenas uma e o necessário são três, uma para cada ponto de medição.



Figura 32: Módulo USB-TTL (autoria própria)

4.1.6. Cabo de Rede

Utilizou-se para a comunicação de longa distância um cabo de rede blindado no padrão Cat5e, Figura 33. Seus condutores são compostos por 4 pares de fios de cobre entrelaçados, cobertos individualmente por uma proteção termo plástico. O conjunto de fios é blindado por uma fita

metálica e protegido por uma capa externa de PVC. A escolha desse cabo baseou-se na sua fácil aquisição, baixo preço, disposição dos fios e blindagem. Os dois últimos itens favorecem na integridade da mensagem durante o envio de dados.



Figura 33: Cabo de dados Cat5e (autoria própria).

4.1.7. Banco de Baterias

Como forma de viabilizar o uso remoto do sistema de medição, utilizaram-se bancos de bateria. Para as unidades de carga, empregaram-se baterias de íons de lítio no formato 18650, 4,2V de tensão e capacidade de 2200mAh, Figura 34.



Figura 34: Baterias empregadas (autoria própria).

Adotou-se um circuito de proteção para utilização e carregamento das baterias chamado *BMS* (*Battery Management System*), modelo BMS hx-2s-d01, Figura 35. Ele previne sobrecarga, descarga excessiva e curto-circuito.

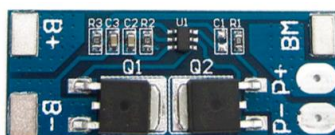


Figura 35: Circuito de proteção BMS (autoria própria).

O esquema elétrico de ligação segue o formato 2S, ou seja, dois conjuntos de bateria ligados em série fornecendo 8,4V, Figura 36.

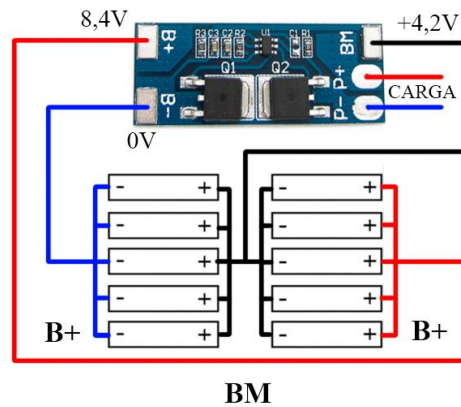


Figura 36: Esquema elétrico de ligação (autoria própria).

4.1.8. Regulador de Tensão

Com o intuito de ajustar a tensão fornecida pelo banco de baterias (8,4V) ao nível de entrada solicitado pelos dispositivos (5V), empregou-se o regulador de tensão *Stepdown* DC-DC L2596 HW-411, Figura 37. A razão da escolha desse tipo de regulador se dá pelo seu alto índice de eficiência operacional (80%) em comparação com outros métodos (TEXAS INSTRUMENTS INCORPORATED, 2020).

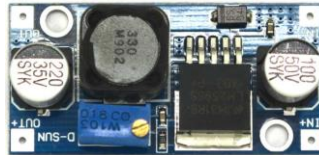


Figura 37: Regulador de tensão (autoria própria).

4.2. Métodos

Na presente seção consta os métodos empregados para elaboração e avaliação de um sistema de monitoramento baseado no funcionamento de acelerômetros e giroscópios. Buscou-se implementar um sistema com três pontos de medição operando de forma síncrona e com a mesma taxa de aquisição de dados. Em cada um desses pontos, monitorou-se aceleração nos eixos Y e Z e giro no eixo X.

4.2.1. Esquema de Ligação

Com o intuito de estabelecer a comunicação entre os dispositivos escolhidos, adotou-se em cada ponto de medição o esquema de ligação apresentado na

Figura 38. Nele o MPU-6500 liga-se ao ESP-32 por meio de uma interface SPI. Este comunica-se com o módulo RS-485 por meio de uma porta serial (UART). Por fim, o Raspberry Pi recebe os dados do módulo RS-485 de forma análoga, via porta serial.

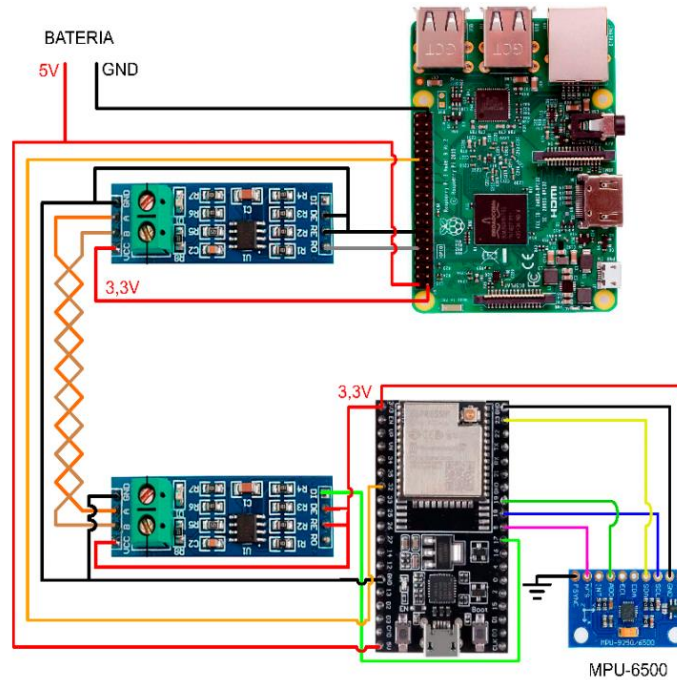


Figura 38: Esquema detalhado de ligação de um ponto de medição (autoria própria)

A Figura 39 mostra o esquema completo de ligação do sistema de medição proposto. Ressalta-se que, como o Raspberry Pi 3B+ apresenta apenas uma porta serial de acesso externo (pinos 8 e 10), utilizaram-se os módulos USB-TTL para conectar os outros dois pontos de medição via USB.

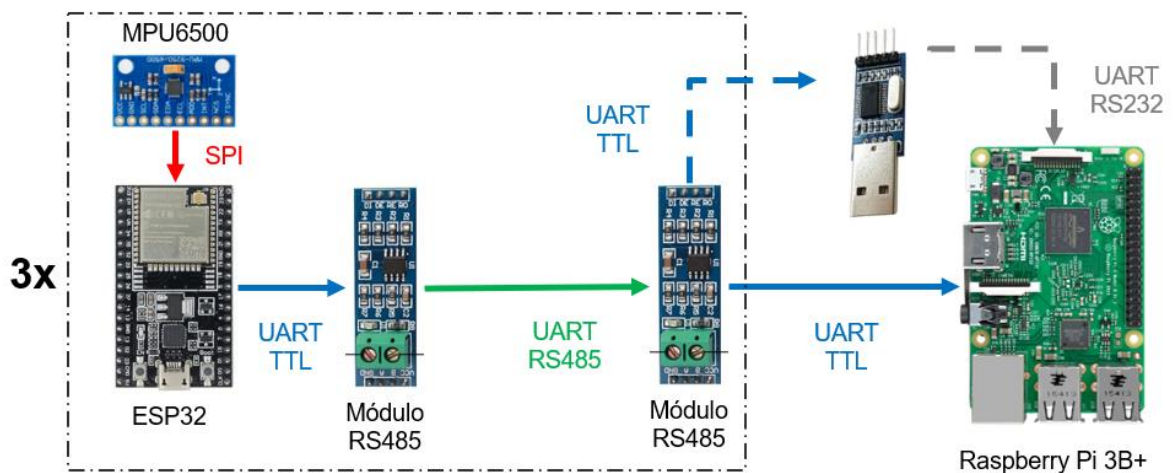


Figura 39: Esquema completo do sistema de medição (autoria própria)

As Figuras 40 e 41 mostram o estágio final da montagem de um dos pontos de medição e do módulo de aquisição de dados, respectivamente.



Figura 40: Ponto de medição (autoria própria).

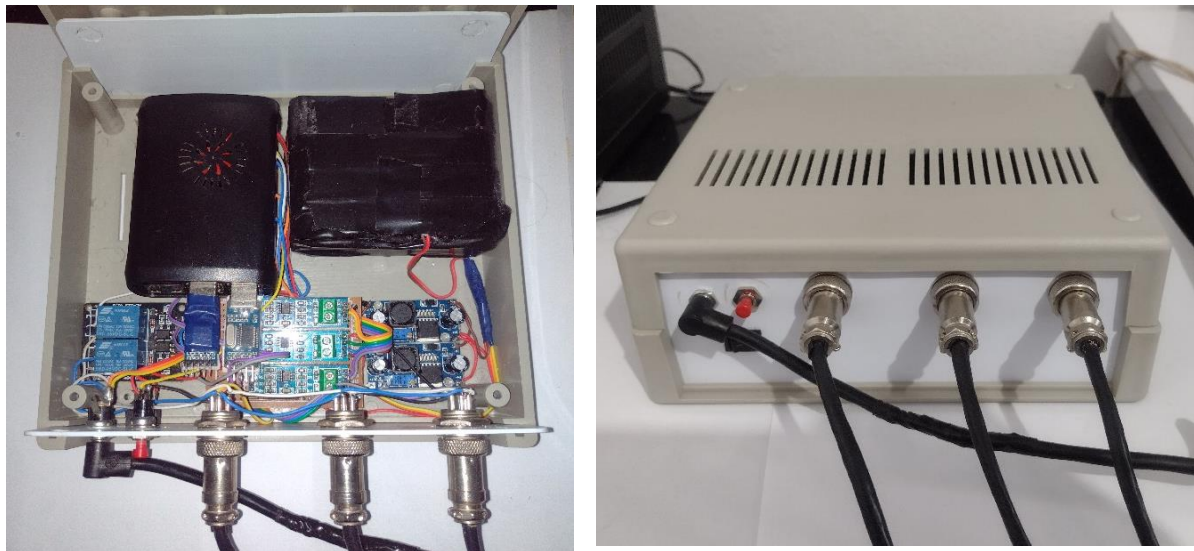


Figura 41: Módulo de aquisição de dados (autoria própria).

4.2.2. Leitura e Envio de Dados do Sensor

Como mencionado anteriormente, ESP-32 responsabilizou-se por controlar o funcionamento do MPU-6500 e enviar os dados coletados por esse sensor. Utilizou-se aqui programação em C++ enviada ao microcontrolador via IDE do Arduino, Figura 42. O código criado pode ser consultado integralmente no ANEXO 1. Nele encontra-se a classe criada no presente trabalho MyMPU como o modelo para criação do objeto MPU.

```

SPL_v3 | Arduino 1.8.11
Arquivo Editar Sketch Ferramentas Ajuda

SPL_v3 | MyMPU.cpp | MyMPU.h
#include "Arduino.h"
#include "SPI.h"
#include "MyMPU.h"

bool ck=1; // Variável auxiliar
bool ck1=1;
MyMPU MPU; // Definição do objeto MPU

void setup(){
  Serial.begin(1000000); // Configura a comunicação serial
  while(!Serial) {}; // Indica se a porta serial está pronta
  MPU.intConfig(); // Configuração inicial do sensor
}

void loop(){
  while(ck1){
    while(ck){
      MPU.tl=micros(); // Salva o valor de tempo inicial
      ck=false;
    }
  }
}

```

Figura 42: IDE do Arduino (autoria própria)

Além disso, desenvolveram-se as funções `intConfig`, `writeRegister`, `readSensor` e `sendDataOverSerial`. A primeira encarrega-se de definir as configurações iniciais do sensor, sendo elas apresentadas na Tabela 2.

Tabela 2: Configurações iniciais MPU-6500

Descrição	Definição
Acelerômetro	Leitura nos eixos Y e Z
Giroscópio	Leitura no eixo X
Escala do acelerômetro	± 2 g
Escala do giroscópio	$\pm 250^\circ/\text{s}$
Taxa de Aquisição	1 kHz
Bandwidth	100 Hz

Fonte: autoria própria

As duas funções subsequentes, estabelecem a comunicação via SPI por meio dos endereços de registro do MPU-6500, consultados no mapa de registros e descrições em Invensense (2013).

Por último, a função `sendDataOverSerial` junta em um mesmo pacote de dados o caractere que sinaliza o início do pacote (“<”), a leitura do sensor e o tempo decorrida em cada ciclo de leitura. Usou-se o artifício do caractere como forma de assegurar que a leitura do pacote começasse de fato pelo início do mesmo evitando, assim, falhas na comunicação. Além disso, esta função encarregou-se de enviar o pacote via porta serial. A Tabela 3 demonstra a construção da mensagem enquanto a Tabela 4 apresenta um exemplo prático da aplicação da estrutura de dados proposta.

Tabela 3: Estrutura do pacote de dados

Caractere 1	Caractere 2	Caractere 3	Acel. Y (MSB)	Acel. Y (LSB)	Acel. Z (MSB)	Acel. Z (LSB)	Giro. X (MSB)	Giro. X (LSB)	Tempo (MSB)	Tempo (LSB)
8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits

Fonte: autoria própria

Tabela 4: Exemplo prático da estrutura do pacote de dados

Formato	Caractere			Acel. Y		Acel. Z		Giro. X		Tempo	
Hexadecimal	0x3C	0x3C	0x3C	0x01	0x23	0x01	0x56	0x02	0x09	0x02	0xE3
Convertido	<	<	<	291		342		521		739	

Fonte: Autoria própria

Onde *MSB* e *LSB* significam *Most Significant Byte* e *Less Significant Byte*, respectivamente. Esta representação em dois bytes dos valores medidos é uma característica do próprio sensor. Tendo em vista que o envio dos dados via porta serial se faz em um byte por vez, decidiu-se manter essa representação. A junção dos bytes e construção da mensagem ficou a cargo do Raspberry Pi.

Destaca-se que, a fim de garantir a sincronicidade de leitura dos três sensores, adotou-se um sistema de monitoramento do estado lógico do pino 33 no ESP-32. Este pino é ligado ao Raspberry Pi e foi definido como estado lógico alto (*INPUT_PULLUP*). Apenas quando seu estado muda para baixo (situação controlada pelo Raspberry Pi), começa-se a adquirir dados do sensor e os enviar via porta serial. Caso contrário, o ESP-32 permanece em um laço de repetição verificando o estado do pino a cada 18 μ s, ou seja, 58,8kHz. Desta forma, todas as unidades de monitoramento possuem um referencial de quando começar a operar. A Figura 43 mostra o comportamento desse mecanismo de sincronização, onde os pontos de medição (MPU-6500+ESP-32) representam-se como N1, N2 e N3, e o Raspberry Pi como RPi.

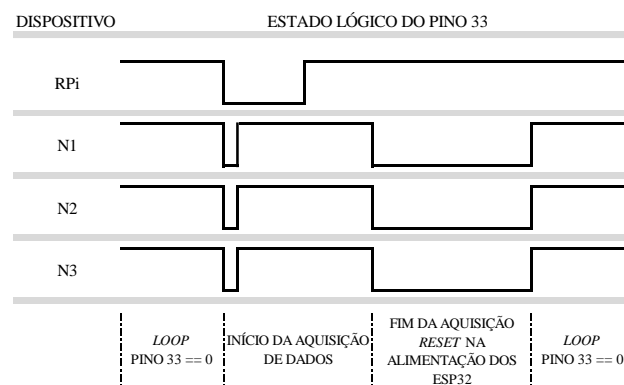


Figura 43: Representação do mecanismo de sincronização da leitura (autoria própria)

Por fim, a Tabela 5 compartilha as configurações adotadas para abertura da comunicação serial no ESP-32.

Tabela 5: Configurações da comunicação serial

Descrição	Valor
Baudrate	230400 bits/s
Quantidade de bits de parada (<i>StopBits</i>)	1
Tamanho do Byte (<i>Bytesize</i>)	8
Bits de paridade	0

Fonte: Autoria própria

4.2.3. Aquisição e Tratamento de Dados

Esta etapa ficou a cargo do Raspberry Pi. Ele recebe os dados enviados do ESP-32 por meio da porta serial, armazena os dados em um vetor de *bytes* e gera a leitura dos dados. Para tanto, contou com programação em Python. Nela desenvolveu-se as funções *intConfig*, *readData*, *startProcess*, *errorChecking*, *convertSaveData* e *showResults*. *intConfig* (ANEXO 2) inicia e configura a comunicação serial de acordo com o estabelecido na Tabela 5. *readData* e *startProcess* trabalham em conjunto. A primeira (ANEXO 4) adquire e salva os dados recebidos via porta serial em um vetor de *bytes*. A segunda (ANEXO 5) controla quando os pontos de medição (ESP32+MPU6500) devem começar a enviar dados. Na função *convertSaveData* (ANEXO 6), aciona-se *errorChecking* (ANEXO 3) a fim de identificar se algum pacote de dados difere da forma apresentada na Tabela 3. Posteriormente, reagrupam-se os *bytes* (*MSB* e *LSB*) com o intuito de gerar um número com 16 bits. Além disso, ajusta-se o tempo (de microssegundos para segundos) e se aplica o fator de sensibilidade nos valores coletados para aceleração e giroscópio. Optou-se por realizar esse procedimento em uma função a parte, fora da *readData*, com intuito de não aumentar o tempo de processamento nas fases de leitura do MPU (a cargo do ESP-32) e aquisição dos pacotes de dados (a cargo do Raspberry Pi). Por fim, criou-se *showResults* (ANEXO 9) como forma de padronizar a apresentação dos dados. Primeiramente, mostra-se a leitura em função do tempo. Logo depois, apresenta-se a densidade espectral de potência calculada por meio da função *periodogram* presente na rotina MRPy (*Multivariate Random Processes with Python*) (ROCHA, 2019).

4.2.4. Análise de Performance de Ruído

Como forma de validar o nível de ruído informado pelo fabricante, executou-se um teste para medição da densidade de ruído nos sensores. Para tanto, posicionou-se cada sensor dentro de um bloco de espuma sob uma base estável, Figura 44. Por meio disso, buscou-se isolar cada

dispositivo de interferências externas. O tempo de aquisição de dados para o ensaio foi de dez minutos. Posteriormente, gerou-se o espectro de potência dos valores adquiridos. Por meio deste pôde-se visualizar melhor o comportamento do ruído ao longo da frequência além de calcular o valor RMS do ruído. Por fim, calculou-se a densidade do ruído (DR) por meio da equação (4.1). Adotou-se o *bandwidth* de 0 a 100 Hz e a média dos ruídos rms obtido em cada sensor. As componentes analisadas foram acelerômetro eixos Y e Z, e giroscópio eixo X.

$$DR = \frac{\sigma_{rms}}{\sqrt{BW}} \quad (4.1)$$

Onde,

σ_{rms} – nível de ruído a uma determinada faixa de banda;

BW – bandwidth adotada.

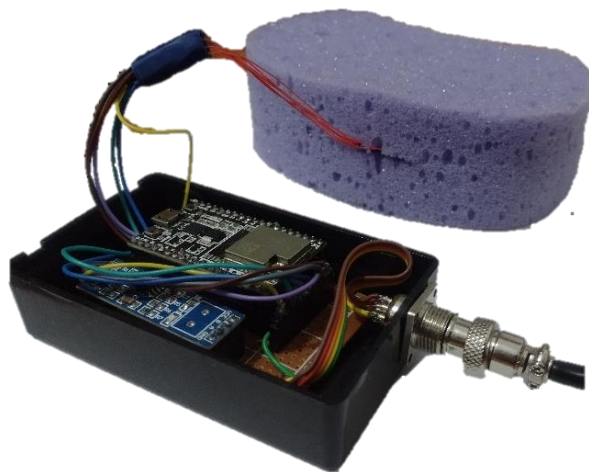


Figura 44: Ensaio de performance de ruído (autoria própria)

4.2.5. Throughput de rede

Nesta etapa, procurou-se determinar a real taxa de aquisição do sistema. Diferentemente do *bandwidth* da transmissão, que define a taxa máxima de aquisição de dados, o *throughput* evidencia limitações dos dispositivos no sistema tais como processamento lento dos dados e pouca memória no *buffer* de entrada. Como consequência, estas acabam por reduzir a taxa de transferência de dados. Destaca-se que o *buffer* se refere a uma região na memória interna destinada ao armazenamento temporário de dados até que um dispositivo (de entrada ou saída) os recupere. Todos os dados recebidos pela porta serial chegam primeiro ao *buffer* de entrada, para depois serem resgatados e utilizados na programação.

O procedimento para determinação do *throughput* começa com o desenvolvimento da função *throughput* (ANEXO 7), que herda o mesmo princípio de funcionamento da *readData*. O que as difere aqui é o monitoramento do tempo necessário para aquisição do pacote de dados pelo Raspberry Pi, o registro de quantos bytes estão no *buffer* de entrada da porta serial a cada ciclo de leitura e a porcentagem de utilização do processador. Durante os ensaios, trabalhou-se com a transferência de mensagens com 12 bytes cada durante o intervalo de 45 minutos. Além disso, compararam-se as seguintes taxas de aquisição para o ensaio: 1000, 500 e 250 Hz. Os comprimentos de cabo utilizados foram os seguintes para cada ponto de medição: N1 - 5m, N2 - 20m e N3 - 20m.

4.2.6. Perdas no envio de dados

Neste ensaio avaliou-se a quantidade de dados perdidos no envio de mensagens com 12 bytes cada. Como forma de verificar cada pacote recebido, enviou-se um pacote de dados (referência) sempre com o mesmo conteúdo. À medida que os dados foram adquiridos, guardavam-se os mesmos em uma matriz. Posteriormente, verificava-se a igualdade de cada um destes com a mensagem de referência. Os três canais de aquisição de dados foram testados de forma síncrona, simulando o funcionamento normal do sistema de medição. O tempo de ensaio foi de 45 minutos e a taxa de envio dos dados foi de 1000 Hz. Em função desses valores, enviou-se o equivalente a 2,7 milhões de mensagens por cada ponto de medição (N1, N2 e N3). Desta forma, verificou-se um total de 8,1 milhões de mensagens. No presente ensaio, utilizou-se o cabo padrão Cat5e nos seguintes comprimentos: 0,10 m (ref), 100 m, 200 m, 300 m e 400 m. A função criada nesta fase (*lostPackets*) encontra-se no ANEXO 8.

4.2.7. Validação da Leitura

Com o intuito de confirmar se a leitura dos sensores era coerente e também síncrona entre eles, submetem-se os mesmos a uma excitação com forma de onda senoidal a frequências pré-definidas. Para tanto, utilizou-se um gerador de onda e um transdutor (alto falante). Como forma de adquirir leituras em ambos as direções do acelerômetro ao mesmo tempo (eixo Y e Z), usou-se uma base inclinada em aproximadamente 24° onde os sensores foram alocados. O conjunto utilizado pode ser visto na Figura 45. Durante o teste, aplicaram-se frequências de 30, 40, 50 e 60 Hz. Como forma de avaliar o comportamento do sistema de forma mais detalhada, empregou-se uma taxa de aquisição de dados relativamente alta para aplicações na engenharia civil, 1000 Hz.

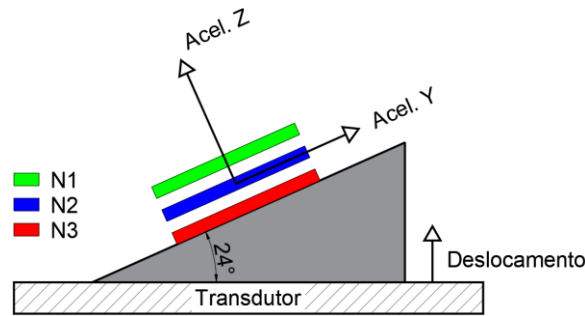


Figura 45: Aparato para teste de validação da leitura (autoria própria)

Após a aquisição dos dados, calculou-se a densidade espectral de potência dos sinais de cada sensor. Nesta fase, buscou-se por erros entre o valor de frequência do sinal interpretado pelo sistema de medição e aquele aplicado como referência. Por fim, calculou-se a correlação cruzada dos sinais de cada sensor com o intuito de verificar a similaridade entre as leituras e, assim, identificar possíveis defasagens (DERRICK; THOMAS, 2004).

4.2.8. Modelo em dimensões reduzidas

Como forma de testar o dispositivo em uma situação mais prática, desenvolveu-se um modelo de dimensões reduzidas. Para tanto, empregou-se uma chapa de aço inox, engastada em ambos os apoios, Figura 46, representando uma viga bi engastada. Por meio das propriedades geométricas e do material, calcularam-se as frequências naturais. Destaca-se que devido a imperfeições nas dimensões da seção transversal, empregou-se análise de propagação de erros no cálculo (ANDERSON, 1976). Neste procedimento, colheu-se 20 amostras tanto da base quanto da altura da chapa utilizada. Calcularam-se a média e desvio padrão da distribuição dessas amostras. Estes valores foram empregados no método de Monte Carlo com 100000 interações.

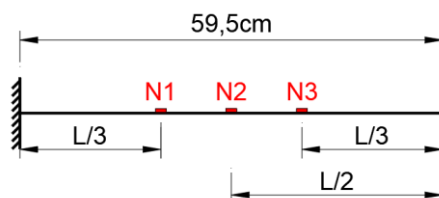


Figura 46; Representação esquemática do modelo ensaiado (autoria própria)

Durante o ensaio, submeteu-se o modelo a uma condição de vibração ambiente caracterizada por um fluxo de ar turbulento, como proposto por Ribeiro e Lameiras (2019). Os sensores captaram a aceleração no sentido transversal nos pontos N1, N2 e N3, como representado na Figura 46. A taxa e tempo de aquisição foram de 1000 Hz e 2 min. Após a aquisição dos dados,

utilizou-se o método de identificação estocástica de subespaços baseado nas covariâncias (SSI-COV). A programação do método foi desenvolvida por CARINI (2021).

Para o cálculo das frequências naturais empregou-se da equação (4.2) (RAO, 2011).

$$f_i = \frac{(\beta_{il})^2}{2\pi} \sqrt{\frac{EI}{\rho A l^4}} \quad (4.2)$$

Onde:

f_i são as frequências naturais em Hz; $\beta_1 l = 4,730041$

E é o módulo de elasticidade (200GPa); $\beta_2 l = 7,853205$

I é o momento de inércia; $\beta_3 l = 10,995608$

A é a área da seção transversal; $\beta_4 l = 14,137165$

l é o comprimento da chapa

Por fim, a Figura 47 representa o registro fotográfico do aparato utilizado no ensaio.

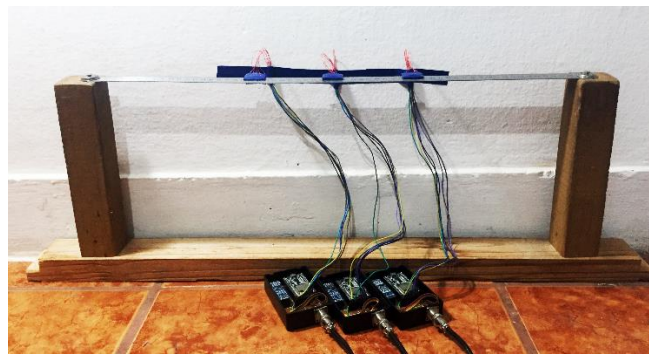


Figura 47: Aparato para ensaio (autoria própria)

4.2.9. Preço do sistema de medição

Nesta fase, levantou-se o valor gasto para aquisição de todos os componentes utilizados no sistema de medição proposto. Apresentaram-se os valores tanto na moeda real quanto dólar. Por fim, compara-se este levantamento com sistemas de medição comercialmente empregados e também com trabalhos que empregaram dispositivos de baixo custo.

5. RESULTADOS E DISCUSSÕES

5.1. Análise de Performance de Ruído

Em função dos dados adquiridos no ensaio, desenvolveu-se o gráfico de densidade espectral de potência, Figura 48. As linhas tracejadas em vermelho, posicionadas a 0 e 100 Hz, indicam a *bandwidth* utilizada para calcular a densidade do ruído. Na Tabela 6, apresentam-se os valores de ruído calculados. Como resultado, observaram-se diferenças entre os valores medidos e aqueles definidos em especificação técnica pelo fabricante. No caso da aceleração no eixo y, constatou-se um valor de ruído 8,1% menor do que a referência. Comportamento contrário ocorreu na aceleração no eixo z, demonstrando aumento de 47,5%. Por fim, o ruído na função giroscópio no eixo X também foi alcançando uma diferença de 33,0 %.

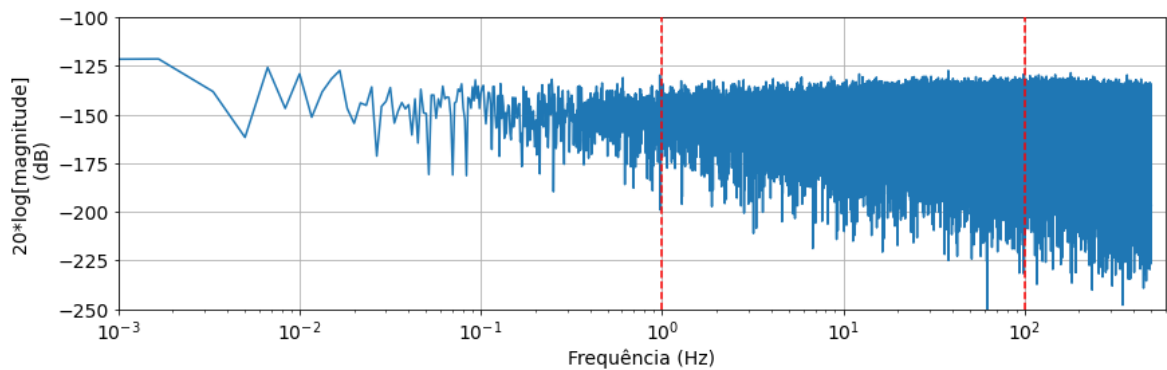


Figura 48: Densidade Espectral de Potência (autoria própria).

Tabela 6: Ruído medido do sensor

Descrição	Espec. técnica	Ruído rms	Densidade de ruído	Erro	Ruído pico a pico (6,6 σ)
Acel. Eixo Y	220 $\mu\text{g}/\sqrt{\text{Hz}}$	2,02 mg	202,24 $\mu\text{g}/\sqrt{\text{Hz}}$	-8,1 %	13,3 mg
Acel. Eixo Z	220 $\mu\text{g}/\sqrt{\text{Hz}}$	3,24 mg	324,46 $\mu\text{g}/\sqrt{\text{Hz}}$	47,5 %	21,4 mg
Giros. Eixo X	0,05 $^{\circ}/\text{s-rms}$	0,067 $^{\circ}/\text{s}$	0,00668 $^{\circ}/\text{s}/\sqrt{\text{Hz}}$	33,0 %	0,442 $^{\circ}/\text{s}$

Fonte: Autoria própria

Os valores de ruído observados foram condizentes com aqueles apresentados na literatura para acelerômetros de baixo custo (RAIL-IP, 2014; RIBEIRO; LAMEIRAS, 2019). Ao analisar a instrumentação tradicional, tais resultados mostraram-se distante do que foi proposto por Cunha et al., (2018) para o monitoramento de estruturas submetidas a vibração ambiente. No seu trabalho, indicaram-se dispositivos com baixo valor de ruído pico a pico, cerca de 2 mg. Por outro lado, CARINI (2021) ao estudar métodos de identificação de sistemas como forma de

conhecer as propriedades dinâmicas de estruturas submetidas a ações ambientais, constatou que o emprego de alguns dos métodos mostrou-se eficaz mesmo em registros com relação ruído/sinal ente 21% e 75%.

5.2. Throughput de rede

Durante os ensaios monitorou-se o número de bytes no *buffer* de entrada da porta serial no Raspberry Pi, a utilização do processador e o tempo de aquisição de um conjunto de dados. O tempo de aquisição de dados foi definido como 45 minutos e as taxas de aquisição ensaiadas foram 1000, 500 e 250 Hz.

Os dados adquiridos pela porta serial são primeiramente armazenados no *buffer* de entrada. Durante os ensaios constatou-se uma maior rapidez no processamento dos dados ao se recuperar uma grande quantidade de bytes de uma só vez acumulados no *buffer* de entrada. Desta forma, desenvolveu-se na programação uma condição em que os dados seriam recuperados apenas quando a lotação do *buffer* chegasse a 300 pacotes de mensagens, equivalente a 3600 *bytes*, levando-se em conta que cada mensagem possuía 12 *bytes*. Destaca-se que esse valor não poderia exceder a capacidade máxima de armazenamento de 4kB estabelecida pelo fabricante, caso contrário acarretaria mal funcionamento da porta serial e perda de dados. A Figura 49 demonstra o funcionamento proposto.

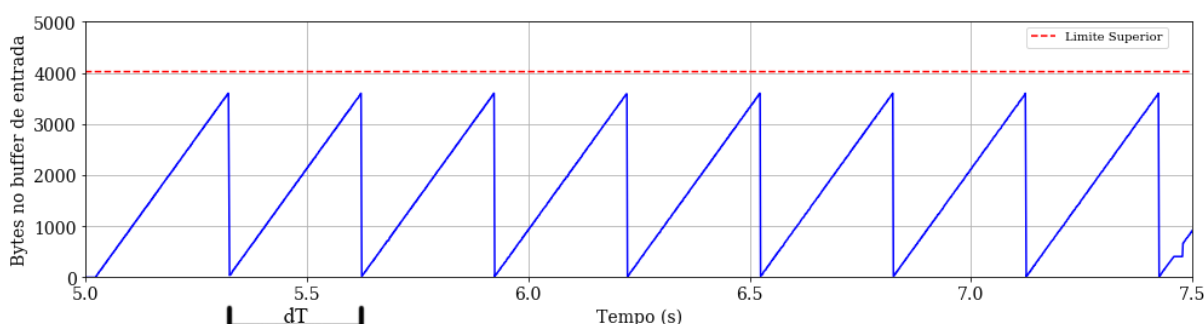


Figura 49: Monitoramento do *buffer* de entrada (autoria própria).

Além disso, monitorou-se o tempo para aquisição desse conjunto de dados por meio da variável dT , Figura 49. Em teoria, dT deveria ser equivalente ao tempo de aquisição de cada pacote de dados multiplicado pelo número de pacotes, 300. Para taxas de aquisição de 250, 500 e 1000 Hz, dT deveria se aproximar de 1200, 600 e 300 ms, respectivamente.

A Figura 50 mostra o comportamento de dT durante o tempo de ensaio. Observou-se que as leituras apresentaram baixa variação quanto aos valores de referência, evidenciando uma

tendência de alta à medida que a taxa de aquisição crescia: 0,14% (250 Hz), 0,16% (500 Hz) e 0,18% (1000 Hz). Tal comportamento repetiu-se no caso do coeficiente de variação (razão entre desvio padrão e média): 0,09% (250 Hz), 0,12% (500 Hz) e 0,17 % (1000 Hz). Isso se deu possivelmente por conta de uma maior solicitação da capacidade de processamento do Raspberry Pi em função do aumento da taxa de aquisição, como pode ser visto na Figura 51. Por fim, destaca-se que a uma taxa de aquisição de 1000 Hz, a utilização do processador cresceu de forma expressiva alcançando 70 % em 45 minutos de ensaio. Isto evidencia que esse tempo de aquisição seja o limite máximo para a citada taxa de aquisição. Valores acima disso poderiam causar mau funcionamento do dispositivo.

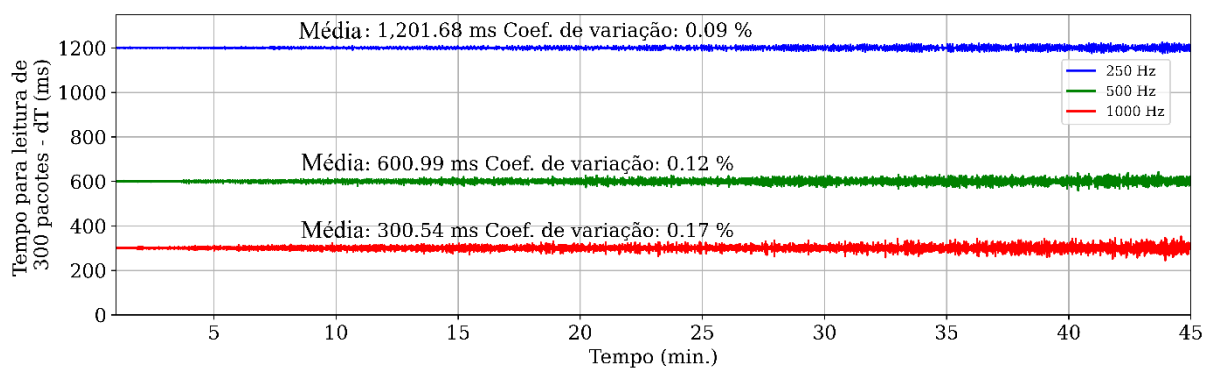


Figura 50: Tempo para aquisição de um conjunto de dados – dT (autoria própria).

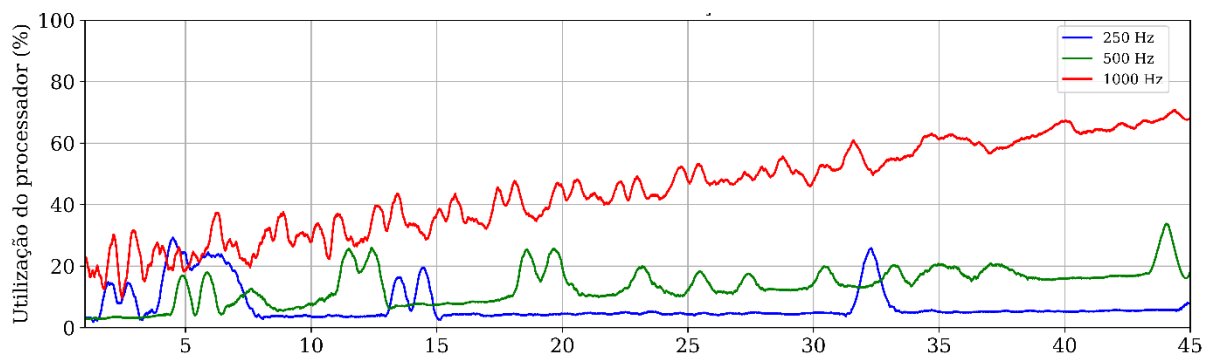


Figura 51: Utilização do processador (autoria própria).

5.3. Perdas no Envio de Dados

Como resultado ao ensaio de perda no envio de dados verificou-se que nenhum pacote de dados ou parte dele foi perdido. Constatou-se tal comportamento para todas as distâncias ensaiadas (0,10 m, 100 m, 200 m e 400 m). Desta forma, observou-se que a interface de comunicação empregada para o dispositivo de monitoramento foi capaz de manter a integridade dos dados na comunicação.

5.4. Validação da Leitura

Na Figura 52, apresenta-se a densidade espectral de potência dos sinais medidos no modo acelerômetro no eixo Z em cada sensor nas frequências de excitação de 30Hz, 40Hz, 50Hz e 60Hz.

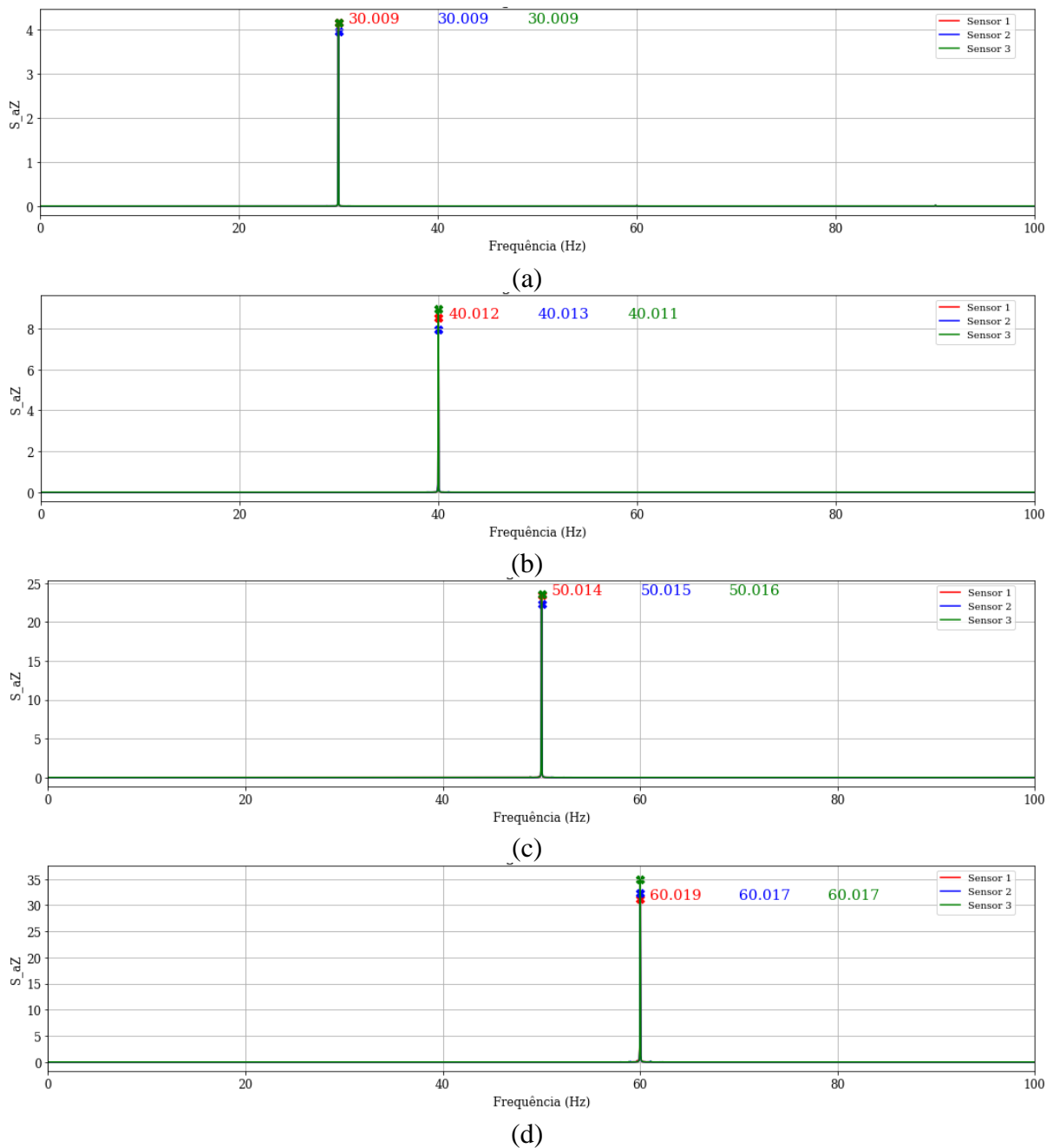


Figura 52: Periodograma dos sinais medidos em cada sensor. (a) 30Hz; (b) 40Hz; (c) 50Hz e (d) 60Hz (autoria própria).

Na Tabela 7, mostra-se o erro referente a diferenças entre o sinal de excitação e aquele adquirido pelo sistema de medição. Os coeficientes de correlação de Pearson entre as medições dos três

sensores constam nas Tabelas 8, 9, 10 e 11. Além disso, as Figuras 53, 54, 55 e 56 demonstram o coeficiente de correlação em função da aplicação de uma defasagem entre os sinais.

Tabela 7: Erro na leitura

Frequência de Excitação (Hz)	Erro (%)
30	0,030
40	0,028 - 0,033
50	0,028 - 0,032
60	0,028 - 0,032

Fonte: Autoria própria

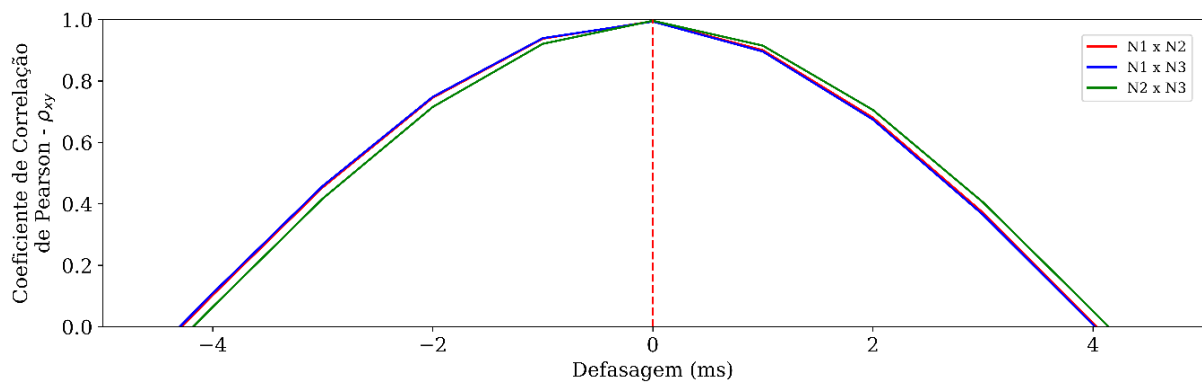


Figura 53: Coeficiente de correlação para 30Hz (autoria própria).

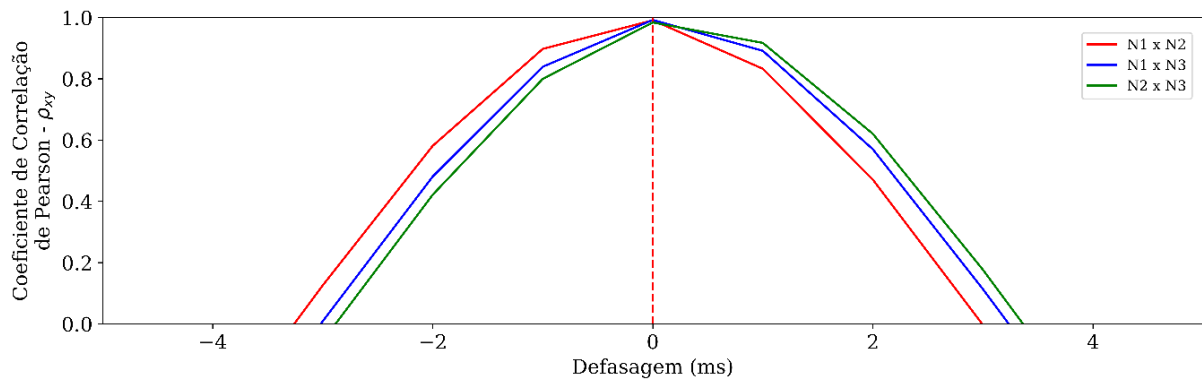


Figura 54: Coeficiente de correlação para 40Hz (autoria própria).

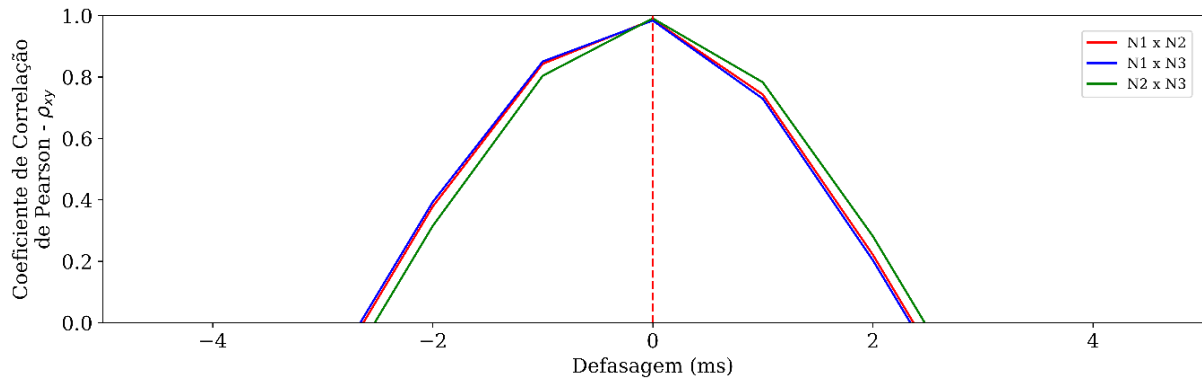


Figura 55: Coeficiente de correlação para 50Hz (autoria própria).

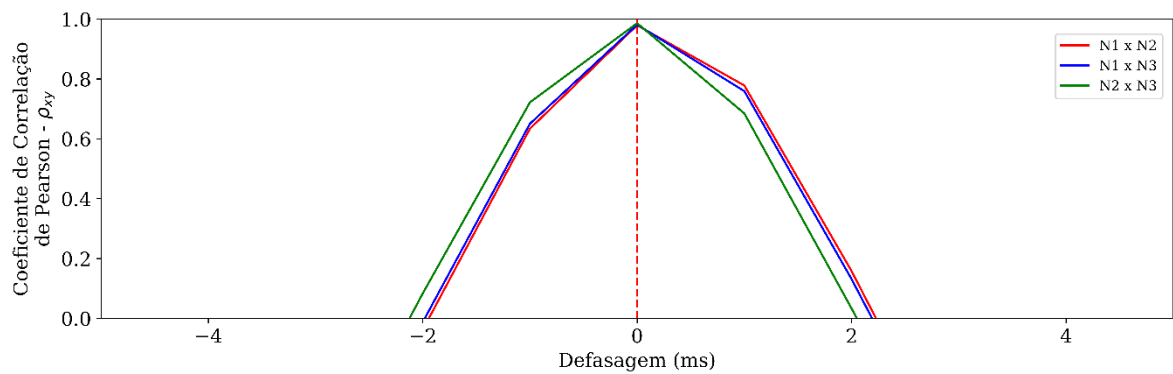


Figura 56: Coeficiente de correlação para 60Hz (autoria própria).

Tabela 8: Índices de correlação entre medições de sensores a 30Hz

Sensor	Acel. eixo Y	Acel. eixo Z	Giros. eixo X
1 – 2	0.995	0.995	0.989
1 – 3	0.994	0.995	0.988
2 – 3	0.996	0.997	0.993

Fonte: Autoria própria

Tabela 9: Índices de correlação entre medições de sensores a 40Hz

Sensor	Acel. eixo Y	Acel. eixo Z	Giros. eixo X
1 – 2	0.991	0.992	0.988
1 – 3	0.992	0.993	0.988
2 – 3	0.984	0.985	0.976

Fonte: Autoria própria

Tabela 10: Índices de correlação entre medições de sensores a 50Hz

Sensor	Acel. eixo Y	Acel. eixo Z	Giros. eixo X
1 – 2	0.987	0.988	0.982
1 – 3	0.984	0.986	0.979
2 – 3	0.991	0.992	0.988

Fonte: Autoria própria

Tabela 11: Índices de correlação entre medições de sensores a 60Hz

Sensor	Acel. eixo Y	Acel. eixo Z	Giros. eixo X
1 – 2	0.980	0.981	0.965
1 – 3	0.982	0.984	0.969
2 – 3	0.987	0.988	0.981

Fonte: Autoria própria

Observou-se por meio dos ensaios que entre todas as frequências de excitação obteve-se um erro máximo de 0,032%. Em relação à análise de correlação cruzada dos sinais, constatou-se que em todos os ensaios os coeficientes de correlação (ρ_{xy}) alcançaram valores próximos de 1, indicando uma alta similaridade entre as leituras. Além disso, o pico com maior valor de ρ_{xy} aconteceu no ponto de defasagem nula entre os sinais. Desta forma, assegurou-se que as leituras estavam em sincronismo (DERRICK; THOMAS, 2004).

5.5. Modelo em Dimensões Reduzidas

Em função das propriedades geométricas da chapa de aço, obtiveram-se os resultados presentes na Tabela 12.

Tabela 12: Propriedades geométricas da seção

	Base (mm)	Altura (mm)
Valor médio	28,833	0,647
Desvio padrão	0,001	0,013

Fonte: Autoria própria

A Tabela 13 apresenta as faixas calculadas para frequências naturais do modelo. Para tanto, adotou-se dois desvios padrões como limites das faixas. Como resultado, observou-se que os valores medidos para todas as frequências se situaram dentro dos intervalos calculados. Desta forma, o sistema de medição proposto se mostrou capaz de extrair as frequências naturais do modelo de dimensões reduzidas.

Tabela 13: Frequências naturais calculadas

fn	1	2	3	4
Faixa calculada - 2σ (Hz)	9,13 - 9,89	25,16 - 27,28	49,33 - 53,49	81,55 - 88,41
Medido (Hz)	9,58	26,21	52,59	87,01

Fonte: Autoria própria

A Tabela 14 traz um comparativo entre as características do sinal medido e o nível de ruído dos sensores.

Tabela 14: Caracterização do sinal medido

Aceleração (g)	N1	N2	N3	Ruído ensaiado
Pico a pico	1,18	1,19	1,28	0,0214
RMS	0,14	0,16	0,14	0,00324

Fonte: Aatoria própria

5.6. Preço Total do Sistema de Medição

O sistema de medição proposto trabalhou com a aquisição de dados de três pontos de medição. Em cada ponto o sensor correspondente encarregou-se de medir aceleração nos eixos Y e Z e velocidade de giro no eixo X. Desta forma, estabeleceu-se o monitoramento de nove canais de medição simultâneos. O preço de cada dispositivo assim como o quantitativo total do sistema de medição consta na Tabela 15.

Tabela 15: Relação de custo detalhada do sistema de medição

N.	Descrição	Qty.	Valor Unit.	Total
1	Kit Raspberry Pi 3B+	1 und.	R\$ 464,98	R\$ 464,98 \$ 85,47
2	ESP32 WROOM 32U	3 und.	R\$ 23,23	R\$ 69,69 \$ 12,81
3	MPU-6000	3 und.	R\$ 9,95	R\$ 29,85 \$ 5,49
4	Módulo RS485-TTL	6 und.	R\$ 9,90	R\$ 59,40 \$ 10,92
5	Módulo USB-TTL	2 und.	R\$ 19,90	R\$ 39,80 \$ 7,32
6	Bateria 18650 Li-ion 4,2V	10 und.	R\$ 9,00	R\$ 90,00 \$ 16,54
7	Módulo BMS 2S	1 und.	R\$ 16,86	R\$ 16,86 \$ 3,10
8	Regulador de tensão - <i>StepDown</i>	2 und.	R\$ 6,71	R\$ 13,42 \$ 2,47
9	Carregador 9V	1 und.	R\$ 50,00	R\$ 50,00 \$ 9,19
10	Módulo Relé	1 und.	R\$ 16,60	R\$ 16,60 \$ 3,05
11	Cabo Cat5e blindado	45 m	R\$ 1,65	R\$ 74,25 \$ 13,65
12	Conectores Mike	6 und.	R\$ 12,70	R\$ 76,20 \$ 14,01
13	Caixa Plástica	1 und.	R\$ 50,00	R\$ 50,00 \$ 9,19
TOTAL			R\$ 1.051,05	\$ 193,21

Fonte: Aatoria própria

O patamar de preços dos itens empregados no dispositivo de monitoramento proposto foi condizente com o praticado na literatura com enfoque em dispositivos de baixo custo, como pode ser visto nos trabalhos de (HOU; WU, 2019; RAIL-IP, 2014). Os equipamentos empregados no SHM em muitos casos representam um alto valor de aquisição. Celebi (2002)

mostra que o preço do sistema de monitoramento pode ser estimado em torno de \$ 2000,00 por canal, considerando sistema de gravação mais o sensor. Desta forma, o dispositivo proposto no presente trabalho enquadrou-se comparativamente como de baixo custo, uma vez que demonstrou um custo de \$ 21,47 por canal monitorado ou \$ 64,41 por ponto de medição.

6. CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um dispositivo de baixo custo, utilizando-se de sensores do tipo MEMS, que pudesse ser aplicado ao monitoramento de integridade estrutural. O dispositivo caracterizou-se por ser um sistema de aquisição de dados composto por três unidades de medição simultânea. Cada uma delas possuía um sensor MPU-6500 programado para monitorar aceleração no eixo Y e Z e velocidade de giro no eixo X. Para tanto, precisou-se entender o comportamento e configuração dos sensores, avaliar o comportamento das interfaces de comunicação entre os dispositivos, detectar e minimizar limitações dos dispositivos empregados e programar um sistema de aquisição simultânea dos dados.

A avaliação do comportamento do dispositivo proposto deu-se por meio de diferentes ensaios. Destes concluiu-se o seguinte:

- a) No ensaio de performance de ruído, pôde-se avaliar o nível de ruído em uma determinada *bandwidth*. Em comparação aos valores informados pelo fabricante, observou-se uma variação considerável. Além disso, tomando-se como base níveis de ruído propostos na literatura para uso em SHM, os resultados obtidos mostram que o sensor utilizado apresenta altos níveis de ruído em sua leitura. Tal empecilho pode ser atenuado empregando-se técnicas de identificação de sistemas como visto em CARINI (2021);
- b) No ensaio de *throughput* de rede avaliou-se a capacidade do dispositivo em adquirir os dados provenientes dos sensores. Os resultados indicaram que os atrasos observados nas leituras não foram suficientes para afetar o funcionamento do sistema. Destaca-se que apenas a uma taxa de aquisição de 1000 Hz e 45 minutos de ensaio o dispositivo chegou a um nível alarmante da utilização do processador. No entanto, tal taxa excede e muito aquelas praticadas no monitoramento de estruturas na construção civil. Desta forma, o dispositivo é apropriado para o monitoramento em ciclos de pelo menos 45 minutos de duração;
- c) No ensaio de perdas no envio de dados, concluiu-se que a comunicação entre os dispositivos foi capaz de transmitir dados sem erros mesmo quando o sistema

se submeteu a um longo comprimento de cabo (400 m) e alta taxa de aquisição (1000 Hz);

- d) Avaliou-se o sinal adquirido por cada sensor comparando-o com um sinal de referência. Por meio disso, concluiu-se que o dispositivo proposto adquiriu os sinais de forma fidedigna ao de referência e síncrona entre os sensores;
- e) No ensaio conduzido no modelo de dimensões reduzidas, viu-se que o dispositivo conseguiu extrair as frequências naturais do modelo proposto, tendo em vista que a relação ruído/sinal foi de 2,3%;
- f) Ao avaliar o valor total gasto para construção do dispositivo proposto comprovou-se que o mesmo se tratava de um dispositivo de baixo custo.

Por fim, pôde-se desenvolver um sistema de aquisição de dados confiável para emprego no monitoramento de integridade estrutural, utilizando equipamentos de baixo custo e grande disponibilidade no mercado. Ressalta-se que cada caso onde se deseja implementar o monitoramento de uma estrutura possui requisitos específicos como: amplitude esperada do sinal, agressividade do meio, nível de interferências ao sinal, fonte de energia, conectividade com a internet, entre outros. Desta forma, destaca-se a importância de estudar tais quesitos em específico como forma de desenvolver um sistema de monitoramento adequado às reais condições a que o mesmo será submetido.

6.1. Trabalhos Futuros

Para trabalhos futuros, pretende-se utilizar o sistema proposto em uma situação prática como uma passarela, ponte ou prédio. Planeja-se também implementar uma comunicação wireless entre os pontos de medição e a unidade de aquisição de dados. Além disso, almeja-se desenvolver um produto final mais enxuto, com tamanho reduzido. Para isto, deve-se trabalhar melhor no dimensionamento das placas de circuito impresso a fim de agrupar de forma mais eficiente os módulos empregados. Por fim, pretende-se avaliar comparativamente diferentes tipos de acelerômetros com o intuito de identificar opções com menor nível de ruído.

REFERÊNCIAS BIBLIOGRÁFICAS

ABRUZZESE, Donato et al. IoT sensors for modern structural health monitoring: a new frontier. **Procedia Structural Integrity**, Roma, v. 25, n. 2019, p. 378–385, 2020. Disponível em: <<https://doi.org/10.1016/j.prostr.2020.04.043>>

AGÊNCIA NACIONAL DE ÁGUAS E SANEAMENTO BÁSICO. **Relatório de segurança de barragens**, Brasília, 2020. Disponível em: <<http://marefateadyan.nashriyat.ir/node/150>>

ALBARBAR, A.; TEAY, S. H. MEMS Accelerometers: Testing and Practical Approach for Smart Sensing and Machinery Diagnostics. In: ZHANG, D.; WEI, B. (Eds.). **Advanced Mechatronics and MEMS Devices II**. 1. ed. Switzerland: Springer International Publishing, 2017. p. 19–40.

AMERICAN SOCIETY OF CIVIL ENGINEERS. **2017 Infrastructure Report Card: Bridges**. New York City: ASCE, 2017. a.

AMERICAN SOCIETY OF CIVIL ENGINEERS. **2017 Infrastructure Report Card: Dams**. New York City: ASCE, 2017. b.

ALL ABOUT CIRCUITS. An Introduction to Filters. 2017. Disponível em: <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/>. Acesso em: 11 set 2021.

ANALOG DEVICES. **Accelerometer Specifications - Quick Definitions**. s.d.. Disponível em: <<https://www.analog.com/en/products/landing-pages/001/accelerometer-specifications-definitions.html>>. Acesso em: 22 jun. 2020.

ANDERSON, G. M. Error propagation by the Monte Carlo method in geochemical calculations. **Geochimica et Cosmochimica Acta**, Toronto, v. 40, n. 12, p. 1533–1538, 1976. Disponível em: <[https://doi.org/10.1016/0016-7037\(76\)90092-2](https://doi.org/10.1016/0016-7037(76)90092-2)>

ARDUINO. **What is Arduino**. 2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 21 abr. 2021.

ASKEGAARD, V.; MOSSING, P. Long term observation of RC bridge using changes in natural frequency. **Nordic Concrete Research**, Oslo, v. 7, p. 20–27, 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 9452: Inspeção de pontes, viadutos e passarelas de concreto - Procedimento**. Rio de Janeiro: ABNT, 2019.

AXELSON, Jan. **Serial Port Complete - Programming and Circuits for RS-232 and RS-485 Links and Networks.pdf**. Madison: Lakeview Research, 2000.

BEEBY, Stephen et al. **MEMS Mechanical Sensors**. 1st. ed. Lodon: Artech House, Inc, 2004.

CELEBI, M. **Seismic instrumentation of buildings (with emphasis on federal buildings) Technical Report No. 0-7460-68170**, Menlo Park, 2002.

CORNWELL, P. et al. Environmental variability of modal properties. **Experimental Techniques**, Los Alamos, v. 23, n. 6, p. 45–48, 1999.

COSTA, Alberto; PROVERBIO, Marco; SMITH, Ian F. C. Cyber civil infrastructure and IoT for cities. In: 2017 IEEE SYMPOSIUM SERIES ON COMPUTATIONAL INTELLIGENCE, SSCI 2017 - PROCEEDINGS 2018, Honolulu. **Anais...** Honolulu

CUNHA, Álvaro et al. Dynamic identification and continuous dynamic monitoring of bridges: different applications along bridges life cycle. **Structure and Infrastructure Engineering**, Porto, v. 14, n. 4, p. 445–467, 2018. Disponível em: <<http://doi.org/10.1080/15732479.2017.1406959>>

CURADELLI, R. O. et al. Damage detection by means of structural damping identification. Porto Alegre, v. 30, n. 12, p. 3497–3504, 2008. Disponível em: <<http://dx.doi.org/10.1016/j.engstruct.2008.05.024>>

DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES. **NORMA DNIT 010/2004 - PRO: Inspeções em pontes e viadutos de concreto armado e protendido - Procedimento**. Rio de Janeiro.

DERAEMAERKER, Arnaud;; WORDEN, Keith. **New Trends in Vibration Based Structural Health Monitoring**. 1. ed. Udine: SpringerWienNewYork, 2010. v. 520

DERRICK, Timothy R.; THOMAS, Joshua M. Time Series Analysis : The Cross-Correlation Function Time Series Analysis. In: STERGIOU, Nicholas (Ed.). **Innovative Analyses of Human Movement**. 1. ed. Champaign. p. 189–205.

DHAKER, Piyu. **Introduction to SPI Interface Analog Dialogue**. Wilmington. Disponível em: <<https://www.analog.com/media/en/analog-dialogue/volume-52/number-3/introduction-to-spi-interface.pdf>>.

DUC, Tuan Ta; ANH, Tuan Le; DINH, Huong Vu. Estimating modal parameters of structures using arduino platform. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE

ON ADVANCES IN COMPUTATIONAL MECHANICS 2018, Vietnam. **Anais...** Vietnam: Springer Nature Singapore, 2018.

ESPRESSIF SYSTEMS. **ESP32: Technical Reference Manual**, Shanghai, 2021. Disponível em: <[http://www.hec.usace.army.mil/software/hec-hms/documentation/HEC-HMS_Technical Reference Manual_\(CPD-74B\).pdf](http://www.hec.usace.army.mil/software/hec-hms/documentation/HEC-HMS_Technical_Reference_Manual_(CPD-74B).pdf)>

FRENZEL, Lou. **Y j c v ø u " V j g " F k h h g t g p 232g And RS-485 g Serial " V j g ' Interfaces?** 2013. Disponível em: <<https://www.electronicdesign.com/technologies/communications/article/21800966/whats-the-difference-between-the-rs232-and-rs485-serial-interfaces>>. Acesso em: 9 set. 2020.

GLORIA, Andre; CERCAS, Francisco; SOUTO, Nuno. Comparison of communication protocols for low cost Internet of Things devices. **2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)**, Lisbon, p. 1–6, 2017.

GUPTA, Aditya. **V j g " K q V " J c e m g t ø u " J c p f d q q m < " C " R t c v k e c of Things**. Walnut: Apress, 2019.

HASEEB, Abdul et al. Microprocessors and Microcontrollers: past, present and future. In: **Functional reverse engineering of machine tools**. Boca Raton: CRC Press, 2020. p. 3–27.

HJORT, Adam; HOLMBERG, Mans. Measuring mechanical vibrations using an Arduino as a slave I/O to an EPICS control system. **Physics Project with a Research Connection**, Uppsala, v. 4, p. 1–25, 2015.

HOLMAN, J. P. **Experimental Methods for Engineers**. 8. ed. New York: McGraw-Hill, 2012.

HOU, Shitong; WU, Gang. A low-cost IoT-based wireless sensor system for bridge displacement monitoring. **Smart Materials and Structures**, Nanjing, v. 28, n. 8, 2019.

HUI, First et al. Detection of Precast Concrete Beam Defects Using Finite-Difference Time-Domain Modelling. In: **IEEE 3RD INTERNATIONAL CONFERENCE ON SMART INSTRUMENTATION 2015**, Putrajaya. **Anais...** Putrajaya: IEEE, 2015.

INVENSENSE. **MPU-6500: Register Map and Descriptions**, Sunnyvale, 2013.

ISTE. **Structural health monitoring**. 1. ed. Newport Beach: ISTE, 2017.

KAMINSKI, J.; RIERA, J. D. Structural damage detection by means vibration tests. In: **14TH**

INTERNATIONAL CONFERENCE ON STRUCTURAL MECHANICS IN REACTOR TECHNOLOGY 1997, Lyon. **Anais...** Lyon

KUGELSTADT, Thomas. **The RS-485 Design Guide**, Dallas, 2016.

LI, Mingyang et al. Structural health monitoring of an offshore wind turbine tower using iFEM methodology. **Ocean Engineering**, Istanbul, v. 204, n. February, p. 107291, 2020. Disponível em: <<https://doi.org/10.1016/j.oceaneng.2020.107291>>

LO IACONO, Francesco; NAVARRA, Giacomo; OLIVA, Maria. Structural monitoring of “Himera” viaduct by low-cost MEMS sensors: characterization and preliminary results. **New Trends in Dynamics and Stability**, Enna, v. 52, n. 13, p. 3221–3236, 2017.

LYNCH, Jerome P.; LOH, Kenneth J. A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring. **The Shock and Vibration Digest**, Ann Arbor, v. 38, n. 2, p. 91–128, 2006.

CARINI, M. R. Identificação das propriedades dinâmicas de estruturas submetidas a ações ambientais. Tese (Doutorado em Engenharia) – Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.

MAXIM. **Low Power, Slew-Rate-Limited RS-485/RS-422 Transceivers** Current, Sunnyvale, 2003.

MEASUREMENT COMPUTING. **Signal and Conditioning PC-Based Data Acquisition Handbook**. 3. ed. Madrid: Measurement Computing Corporation, 2012.

MOHAMED, Gad-el-Hak (ED.). **MEMS Introduction and Fundamentals**. 2. ed. New York: CRC, 2006.

MONTEIRO, Felipe Alexandre Cavaleiro. **Monitorização de Estruturas**. 2010. Universidade da Beira Interior, Covilhã, 2010.

MORRIS, Alan S. **Measurement and instrumentation Principles**. 1. ed. Woburn: BH, 2001.

MUSTAPHA, Samir et al. Sensor networks for structural health monitoring. **Journal of Sensors**, Gdansk, v. 2020, p. 1–2, 2020.

NOEL, Adam B. et al. Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey. **IEEE Communications Surveys and Tutorials**, St. John’s, v. 19, n. 3, p. 1403–1423, 2017.

NOTAY, Jasleen K.; SAFDAR, Ghazanfar A. A wireless sensor network based structural

health monitoring system for an airplane. **Proceedings of 2011 17th International Conference on Automation and Computing, ICAC 2011**, Luton, n. September, p. 240–245, 2011.

OLIVEIRA, C. B. L.; GRECO, M.; BITTENCOURT, T. N. Analysis of the brazilian federal bridge inventory. **Revista IBRACON de Estruturas e Materiais**, Belo Horizonte, v. 12, n. 1, p. 1–3, 2019.

OSTACHOWICZ, Wieslaw; SOMAN, Rohan; MALINOWSKI, Pawel. Optimization of sensor placement for structural health monitoring: a review. **Structural Health Monitoring**, Gdansk, v. 18, n. 3, p. 963–988, 2019.

OU, Jinping; LI, Hui. Structural health monitoring in mainland china: Review and future trends. **Structural Health Monitoring**, Dalian, v. 9, n. 3, p. 219–231, 2010.

PROTOSUPPLIES. **Esquema de Ligação - Módulo RS-485**. 2020. Disponível em: <<https://protosupplies.com/product/max485-ttl-to-rs-485-interface-module/>>. Acesso em: 1 set. 2020.

RAIL-IP, Rene. **Evaluation of low-cost MEMS accelerometers and investigation of inertial algorithms for dead reckoning in railway environment**. 2014. University of Toronto, Toronto, 2014.

RAO, Singiresu S. **Mechanical Vibrations**. 5. ed. Miami: Pearson Education, 2011.

RASPBERRY PI FOUNDATION. **Raspberry Pi 3 Model B +**, Cambridge, 2018.

RASPBERRY PI FOUNDATION. **Raspberry Pi Pinout**. 2020. Disponível em: <<https://pinout.xyz/>>. Acesso em: 10 set. 2020.

RIBEIRO, Renan Rocha; LAMEIRAS, Rodrigo de Melo. Evaluation of low-cost MEMS accelerometers for SHM: Frequency and damping identification of civil structures. **Latin American Journal of Solids and Structures**, Brasília, v. 16, n. 7 CILAMCE 2018, p. 1–24, 2019.

ROCHA, Marcelo Maia. **MRPy: Multivariate Random Processes with Python**, Porto Alegre, 2019. Disponível em: <<https://github.com/mmaiarocha/MRPy>>

RYTTER, Andres. **Vibrational Based Inspection of Civil Engineering Structures**. 1993. Aalborg Universitet, Aalborg, 1993.

SANCHEZ-SILVA, Mauricio; KLUTKE, Georgia-Ann; ROSOWSKY, David V. Life-cycle performance of structures subject to multiple deterioration mechanisms. **Structural Safety**,

Bogota, v. 33, n. 2011, p. 206–2017, 2011.

SANTOS, João Pedro de Oliveira Dias Prudente Dos. **Smart Structural Health Monitoring Techniques for Novelty Identification in Civil Engineering Structures**. 2014. Universidade de Lisboa, Lisboa, 2014.

SEALEVEL. **Serial Electrical Interfaces**. 2012. Disponível em: <<https://www.sealevel.com/support/serial-electrical-interfaces/>>. Acesso em: 10 set. 2020.

TDK INVENSENSE. **MPU-6500: Product specification** **Building Research & Information**, San Jose, 2020.

TEXAS INSTRUMENTS INCORPORATED. **LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator** **Datasheet**, Dallas, 2020.

TONDOLO, F. et al. Smart reinforcement steel bars with low-cost MEMS sensors for the structural health monitoring of RC structures. **Construction and Building Materials**, Torino, v. 173, p. 740–753, 2018. Disponível em: <<https://doi.org/10.1016/j.conbuildmat.2018.04.045>>

UCHINO, Kenji. The Development of Piezoelectric Materials and the New Perspective. In: **Advanced Piezoelectric Materials: Science and Technology**. 1. ed. State College: Elsevier Ltd., 2017. p. 1–696.

WANG, Chunhui et al. Structural health monitoring and performance analysis of a 12-story recycled aggregate concrete structure. **Engineering Structures**, Shanghai, v. 205, n. 110102, p. 1–18, 2020. Disponível em: <<https://doi.org/10.1016/j.engstruct.2019.110102>>

WARSI, Zain Hussain et al. Sensors for structural health monitoring: A review. In: 2019 2ND INTERNATIONAL CONFERENCE ON LATEST TRENDS IN ELECTRICAL ENGINEERING AND COMPUTING TECHNOLOGIES, INTELLECT 2019 2019, Karachi. **Anais...** Karachi: IEEE, 2019.

WONG, Kai-yuen. Design of a structural health monitoring system for long-span bridges Design of a structural health monitoring system for long-span bridges. **Structure and Infrastructure Engineering**, Hong Kong, v. 3, n. 2, p. 169–185, 2007.

WORLD ECONOMIC FORUM. **Strategic Infrastructure Steps to Operate and Maintain Infrastructure Efficiently and Effectively**, Davos, 2014.

XIA, Yong et al. Temperature effect on vibration properties of civil structures: A literature

review and case studies. **Journal of Civil Structural Health Monitoring**, Hong Kong, v. 2, n. 1, p. 29–46, 2012.

ZHOU, Cuiying et al. Comprehensive measurement techniques and multi-index correlative evaluation approach for structural health monitoring of highway bridges. **Measurement: Journal of the International Measurement Confederation**, Guangzhou, v. 152, p. 107360, 2020. Disponível em: <<https://doi.org/10.1016/j.measurement.2019.107360>>

APÊNDICE 1: ESPECIFICAÇÕES TÉCNICAS DO GIROSCÓPIO

3 ELECTRICAL CHARACTERISTICS

3.1 GYROSCOPE SPECIFICATIONS

Typical Operating Circuit of section 4.2, VDD = 1.8V, VDDIO = 1.8V, T_A=25°C, unless otherwise noted.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	3
	FS_SEL=1		±500		°/s	3
	FS_SEL=2		±1000		°/s	3
	FS_SEL=3		±2000		°/s	3
Gyroscope ADC Word Length			16		bits	3
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	3
	FS_SEL=1		65.5		LSB/(°/s)	3
	FS_SEL=2		32.8		LSB/(°/s)	3
	FS_SEL=3		16.4		LSB/(°/s)	3
Sensitivity Scale Factor Tolerance	25°C		±3		%	2
Sensitivity Scale Factor Variation Over Temperature	-40°C to +85°C		±4		%	1
Nonlinearity	Best fit straight line; 25°C		±0.1		%	1
Cross-Axis Sensitivity			±2		%	1
ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±5		°/s	2
ZRO Variation Over Temperature	-40°C to +85°C		±0.24		°/s/°C	1
GYROSCOPE NOISE PERFORMANCE (FS_SEL=0)						
Total RMS Noise	DLPFCFG=2 (92 Hz)		0.1		°/s-rms	2
Rate Noise Spectral Density			0.01		°/s/√Hz	4
GYROSCOPE MECHANICAL FREQUENCIES		25	27	29	KHz	2
LOW PASS FILTER RESPONSE	Programmable Range	5		250	Hz	3
GYROSCOPE START-UP TIME	From Sleep mode		35		ms	1
OUTPUT DATA RATE	Programmable, Normal (Filtered) mode	4		8000	Hz	1

Table 1. Gyroscope Specifications

Fonte: TDK INVENSENSE (2020)

APÊNDICE 2: ESPECIFICAÇÕES TÉCNICAS DO ACELERÔMETRO

3.2 ACCELEROMETER SPECIFICATIONS

Typical Operating Circuit of section 4.2, VDD = 1.8V, VDDIO = 1.8V, T_A=25°C, unless otherwise noted.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		g	3
	AFS_SEL=1		±4		g	3
	AFS_SEL=2		±8		g	3
	AFS_SEL=3		±16		g	3
ADC Word Length	Output in two's complement format		16		bits	3
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	3
	AFS_SEL=1		8,192		LSB/g	3
	AFS_SEL=2		4,096		LSB/g	3
	AFS_SEL=3		2,048		LSB/g	3
Initial Tolerance	Component-level		±3		%	2
Sensitivity Change vs. Temperature	-40°C to +85°C AFS_SEL=0 Component-level		±0.026		%/°C	1
Nonlinearity	Best Fit Straight Line		±0.5		%	1
Cross-Axis Sensitivity			±2		%	1
ZERO-G OUTPUT						
Initial Tolerance	Component-level, all axes		±60		mg	2
Zero-G Level Change vs. Temperature	-40°C to +85°C, Board-level	X and Y axes	±0.64		mg/°C	1
		Z axis	±1		mg/°C	1
NOISE PERFORMANCE						
Power Spectral Density	Low noise mode		300		µg/√Hz	4
LOW PASS FILTER RESPONSE	Programmable Range	5		260	Hz	3
INTELLIGENCE FUNCTION INCREMENT			4		mg/LSB	3
ACCELEROMETER STARTUP TIME	From Sleep mode		20		ms	1
	From Cold Start, 1ms V _{DD} ramp		30		ms	1
OUTPUT DATA RATE	Low power (duty-cycled)	0.24		500	Hz	1
	Duty-cycled, over temp		±15		%	
	Low noise (active)	4		4000	Hz	

Table 2. Accelerometer Specifications

Fonte: TDK INVENSENSE (2020)

ANEXO 1: CÓDIGO ESP-32

Código Principal

Responsável por iniciar a comunicação serial, gerenciar dos dados do sensor por meio da classe MyMPU e enviar os dados via serial.

```
#include "Arduino.h"
#include "SPI.h"
#include "MyMPU.h"

bool ck; // Variável auxiliar
MyMPU MPU; // Definição do objeto MPU

void setup(){
  Serial.begin(1000000,SERIAL_8N2); // Configura a comunicação serial
  while(!Serial) {}; // Indica se a porta serial está pronta
  MPU.intConfig(); // Configuração inicial do sensor
}

void loop(){
  while (ck){
    MPU.t1=micros(); // Salva o valor de tempo inicial
    ck=false;
  }
  MPU.sendDataOverSerial(); // Adquire e envia os dados via serial
}
```


Classe MyMPU

Classe que gerencia o sensor em relação as configurações iniciais, aquisição de dados e o envio dos mesmos via serial.

```
#ifndef MPU9250_h          // Verifica se a classe já foi definida
#define MPU9250_h

#include "Arduíno.h"
#include "SPI.h"          // A comunicação com o sensor utiliza o protocolo SPI

class MyMPU {
private:
    union Data_int{       // Utilizada para representar uma variável int (16bits) na forma de 2 bytes (2 * 8bit)
        uint16_t a;
        byte b[2];
    };

    // Registros para comunicação com o MPU9250 //

    const uint8_t PWR_MGMT_1 = 0x6B; // Gerenciamento de energia 1
    const uint8_t PWR_MGMT_2 = 0x6C; // Gerenciamento de energia 2
    const uint8_t PWR_MGMT_CONFIG = 0x80; // Reseta o MPU (b'10000000')
    const uint8_t HAB_AX = 0x1F; // Habilita apenas o acelerômetro no eixo x (b'00011111')
    const uint8_t ACCEL_CONFIG = 0x1C; // Conf. do acelerômetro (altera a escala)
    const uint8_t ACCEL_FS_SEL_2G = 0x00; // Configura a escala de leitura para +-2g
    const uint8_t ACCEL_FS_SEL_4G = 0x08; // Configura a escala de leitura para +-4g
    const uint8_t ACCEL_FS_SEL_8G = 0x10; // Configura a escala de leitura para +-8g
    const uint8_t ACCEL_FS_SEL_16G = 0x18; // Configura a escala de leitura para +-16g
    const uint8_t ACCEL_CONFIG_2 = 0x1D; // Conf. do acelerômetro (altera bandwidth)
    const uint8_t ACCEL_DLPF_113K = 0x08; // Configura bandwidth para
    const uint8_t SMPDIV = 0x19; // Configura o divisor do internal sample rate
    const uint8_t SPI_READ = 0x80;

    uint8_t val[5]; // Array a ser enviado via serial
    uint8_t tf; // Armazena o intervalo de tempo entre cada leitura do sensor

public:
    uint8_t ACCEL_XOUT_H = 0x3B; // Define qual componente do accel. será lida
    unsigned long t1, t2; // Registra o início e fim de uma leitura
    int csPin=0; // Chip Select (comunicação SPI)
    const uint32_t SPI_LS_CLOCK = 1000000; // Baixa velocidade - 1 MHz (Escrita)
    const uint32_t SPI_HS_CLOCK = 20000000; // Alta velocidade - 20 MHz (Leitura)
    uint8_t _buffer[2]; // Armazena High and Low bytes

    // As funções não retornam nenhum valor
    void writeRegister (uint8_t subAddress, uint8_t data);
    void readSensor(uint8_t subAddress, uint8_t count, uint8_t* dest);
    void intConfig();
    void sendDataOverSerial();
};

#endif
```

Funções

```
# include "Arduino.h"
#include "MyMPU.h"

void MyMPU :: writeRegister (uint8_t subAddress, uint8_t data){
  SPI.beginTransaction(SPISettings(SPI_LS_CLOCK, MSBFIRST, SPI_MODE3)); // Inicia a comunicação
  digitalWrite(_csPin,LOW); // Seleciona o chip
  SPI.transfer(subAddress); // Endereço do registro
  SPI.transfer(data); // Escreve o dado
  digitalWrite(_csPin,HIGH); // Deseleciona o chip
  SPI.endTransaction(); // Finaliza a transferência
}

void MyMPU::readSensor(uint8_t subAddress, uint8_t count, uint8_t* dest) {
  SPI.beginTransaction(SPISettings(SPI_HS_CLOCK, MSBFIRST, SPI_MODE3));
  digitalWrite(_csPin,LOW); // Seleciona o chip
  SPI.transfer(subAddress | SPI_READ); // Especifica o endereço de registro inicial
  for(uint8_t i = 0; i < count; i++){
    dest[i] = SPI.transfer(0x00); // Adquire o dado
  }
  t2=micros();
  tf=t2-t1;
  t1=t2;
  digitalWrite(_csPin,HIGH); // Deseleciona o chip
  SPI.endTransaction(); // Finaliza a transferência
}

void MyMPU :: intConfig(){
  pinMode(_csPin,OUTPUT); // Define o pino CS como output
  digitalWrite(_csPin,HIGH); // Atribui high para o pino CS (modo desselecionado do SPI)
  SPI.begin(); // Inicia a comunicação SPI

  writeRegister(PWR_MGMT_1,PWR_MGMT_CONFIG); // Resetar o MPU9250
  delay(1); // Espera o MPU retornar

  writeRegister(PWR_MGMT_2,HAB_AX); // Habilita leitura do ACCEL X
  writeRegister(ACCEL_CONFIG,ACCEL_FS_SEL_16G); // Estabelece a amplitude para +-16g
  writeRegister(ACCEL_CONFIG_2,ACCEL_DLPF_113K); // Estabelece bandwidth para 1.13KHz
  writeRegister(SMPDIV,0x00); // Define SMPDIV = 0
};

void MyMPU :: sendDataOverSerial(){
  readSensor(ACCEL_XOUT_H, 2, _buffer);

  val[0]=0x3C; // Caracter de início da sequência de dados ("<")
  val[1]=0x3C; // Caracter de início da sequência de dados ("<")
  val[2]=_buffer[0]; //
  val[3]=_buffer[1]; //
  val[4]=tf;
  Serial.write(val, 5); // Envia o array (val) de dados

  Serial.flush(); // Espera o array ser enviado
};
```

ANEXO 2: FUNÇÃO INTCONFIG

```
#-----#
...
Recebe informações preliminares como 'porta' e 'baudrate' a fim de dar início à
comunicação serial
...
def intConfig(port, baudRate):
    global ser
    ser = serial.Serial(port, baudRate, timeout=5)
    ser.stopbits = serial.STOPBITS_ONE           # Um stopbit
    ser.bytesize = 8                             # Tamanho do byte
    ser.parity = serial.PARITY_NONE              # Não há verificação de paridade

    print ("Serial port opened " + port + " over Baudrate " + str(baudRate))

    return ser
#-----#
```

ANEXO 3: FUNÇÃO ERRORCHECKING

```

#-----#
...
Verifica a existência e localização de algum pacote de dados que foge ao padrão
pré-estabelecido
...
def errorChecking4(data, nBytes):
    import ctypes
    position=np.array([[ ]],dtype=int)
    status=np.array([[ ]])
    j=0
    lim=xBuffer*nSamples*nBytes-(nBytes-3)
    while (j<lim):
        if (bytes(ctypes.c_uint8(data[j]))=='<'.encode()) \
            and (bytes(ctypes.c_uint8(data[j+1]))=='<'.encode()) \
            and (bytes(ctypes.c_uint8(data[j+2]))=='<'.encode()):# Procura por (<<<)
            position=np.append(position, j) #Salva a posição de cada pacote
            j=j+nBytes
            p=(j/lim)*100
            sys.stdout.write('\rPROCURANDO POR ERROS. PROGRESSO {0:.0f} %'.format(p))
        else:
            j=j+1
    for i in range (len(position)-1):
        if (position[i+1]-position[i]==nBytes):# Procura por pacotes fora do padrão
            status=np.append(status,[True])
        elif (position[i+1]-position[i]<nBytes):
            status=np.append(status,[False])
        else:
            status=np.append(status,[False])
            status=np.append(status,[False])
    time.sleep(1)
    return position, status # Retorna a posição e status de cada pacote
#-----#

```

ANEXO 4: FUNÇÃO READDATA

```

#-----#
'''
Função que adquire e salva os dados recebidos via serial em um vetor de bytes.
'''

def readData(ser, nBytes, nSamples, xBuffer, i):
    data=b''
    print("Aquisição iniciada")

    ser[i].reset_input_buffer()
    for k in range(nSamples):
        data+=ser[i].read(nBytes*xBuffer)
        print(ser[i].in_waiting," ",k)

    print("Aquisição concluída")
    ser[i].reset_input_buffer()
    ser[i].close
    return data

#-----#

```

ANEXO 5: FUNÇÃO STARTPROCESS

```
#-----#  
...  
Função que sincroniza o envio de dados de cada ponto de medição (ESP32+MPU6500).  
...  
def startProcess():  
    global ESP_START  
    import RPi.GPIO as GPIO  
    GPIO.setmode(GPIO.BOARD)  
  
    GPIO.setup(ESP_START, GPIO.OUT)  
  
    time.sleep(5)  
    GPIO.output(ESP_START, GPIO.LOW)  
    time.sleep(0.5)  
    GPIO.output(ESP_START, GPIO.HIGH)  
#-----#
```

ANEXO 6: FUNÇÃO CONVERTSAVE DATA

```

#-----#
...
A presente função encarregase de reagrupam-se os bytes (MSB e LSB) com o intuito de
gerar um número com 16 bits. Além disso, ajusta o tempo para segundos e aplica o
fator de sensibilidade nos valores coletados para aceleração e giroscópio.
...
def convertSaveData3 (data, add, freq, S_a, S_g, nBytes):
    import struct
    import time
    e=0
    position, status=errorChecking4(data, nBytes)
    newData=np.array([[0,0,0,0]],dtype=np.int16)
    time=np.array([[0]],dtype=np.int16)
    n=str(data[position[2]+3])
    ta=0
    for j in range (len(status)-1):
        p=(j/(len(status)-1)*100)
        sys.stdout.write('\rCONVERTENDO DADOS. PROGRESSO {0:.0f} %'.format(p))
        if status[j]==True:# Aceleração no eixo Y, Z; Giros. eixo X
            ay=struct.unpack_from('>h', data, offset=(position[j]+6))[0]/S_a
            az=struct.unpack_from('>h', data, offset=(position[j]+8))[0]/S_a
            gx=struct.unpack_from('>h', data, offset=(position[j]+12))[0]/S_g
            t=struct.unpack_from('>h', data, offset=(position[j]+18))[0]
            tr=ta+t
            newData=np.append(newData, [[ay, az, gx, tr]], axis = 0)
            ta=tr
        else:
            ay1=struct.unpack_from('>h', data, offset=(position[j-1]+4))[0]
            ay2=struct.unpack_from('>h', data, offset=(position[j+1]+4))[0]
            az1=struct.unpack_from('>h', data, offset=(position[j-1]+6))[0]
            az2=struct.unpack_from('>h', data, offset=(position[j+1]+6))[0]
            gx1=struct.unpack_from('>h', data, offset=(position[j-1]+8))[0]
            gx2=struct.unpack_from('>h', data, offset=(position[j+1]+8))[0]
            t1=struct.unpack_from('>h', data, offset=(position[j]+10))[0]
            t2=struct.unpack_from('>h', data, offset=(position[j]+10))[0]
            tr=ta+(t1+t2)/2
            newData=np.append(newData, [[(ay1+ay2)/(2*S_a),(az1+az2)/(2*S_a),\
                (gx1+gx2)/(2*S_g),tr]], axis = 0)
            ta=tr
            print("ERROR DETECTED IN SENSOR",n,"J:",j)
            e+=1
    data2 = MRPy.resampling(newData[1:,-1]/1000000, newData[1:,-1])
    data2.to_file(add+n, form='mrpy')
#-----#

```

ANEXO 7: FUNÇÃO THROUGHPUT

```

#-----#
...
Monitora o tempo de aquisição de cada pacote de dados assim como o número de bytes
no buffer de entrada.
...

def throughput(ser, nBytes, nSamples, xBuffer, i):
    t=np.array([])
    b=np.array([])

    data=b''
    print("Aquisição iniciada")

    ser[i].reset_input_buffer()          # Limpa o buffer de entrada da porta serial
    for k in range(nSamples):
        while(ser[i].in_waiting<=nBytes*xBuffer):
            t=np.append(t,[[time.time()]])

            b=np.append(b,ser[i].in_waiting)
            data+=ser[i].read(nBytes*xBuffer)

    print("Aquisição concluída")
    ser[i].reset_input_buffer()
    ser[i].close

    t2=np.zeros(len(t))
    for j in range((len(t))-1):
        t2[j+1] = (t[j+1]-t[0])      # Converte o tempo para segundos

    data2 = MRPy.resampling(t2, b)
    data2.to_file("SHM/ENSAIOS/7 - throughput/TESTE{}".format(i), form='mrpy')
#-----#

```


ANEXO 8: FUNÇÃO LOSTPACKETS

```

def lostPackets(smrk, nBytes, nSamples):
    global data
    startMarker=str(smrk).encode() # str que sinaliza o inicio da mensagem (previamente configurado no ESP8266)

    data=np.array([[0,0]]) # Array para armazenamento de dados
    print("Aquisição iniciada")

    ser.reset_input_buffer() # Limpa o buffer de entrada da porta serial
    for k in range(nSamples):
        wait=True
        while(wait):
            if (ser.read()==startMarker and ser.read()==startMarker): # Verifica se há uma sequência sucessiva de startMarker
                a=ser.read(nBytes-2) # Lê os bytes posteriores à sequência de inicialização
                data=np.append(data, [[twos(((a[0])<<8)|(a[1]),2),a[2]]], axis=0) # Recebe os bytes high e low do MPU e transforma em um int16
                wait=False
        ser.close
        n=0
    for j in range(len(data)):
        if and(data[j,0]==0, data[j,1]==1, data[j,2]==2):
            n+=1

    print("Aquisição concluída")
    print("Número de pacotes perdidos: {0} \nRepresentatividade: {1}%".format(n, n/nSamples))

```

ANEXO 9: FUNÇÃO SHOWRESULTS

```

#-----#
...
Apresenta os resultados na forma gráfica (apenas para execução via Jupyter Notebook).
Resultados:
1 - Aceleração x Tempo
...
def showResults_3a(data0, data1, data2, xlabel, ylabel, title,limx1,limx2,limy1,limy2,save):
    ##### OFFSET #####
    ay0 = MRPy(data0[0], data0.fs)-(-0.027357961899093428+0)
    ay1 = MRPy(data1[0], data1.fs)-(-0.03146244470753679+0)
    ay2 = MRPy(data2[0], data2.fs)-(-0.04900648116140767+0)

    az0 = MRPy(data0[1], data0.fs)-(-0.9031430798883686+1)
    az1 = MRPy(data1[1], data1.fs)-(-1.0228305425954398+1)
    az2 = MRPy(data2[1], data2.fs)-(-0.9943399810241024+1)

    gx0 = MRPy(data0[2], data0.fs)-(-1.3692738464838845+0)
    gx1 = MRPy(data1[2], data1.fs)-(-0.07591762508553412+0)
    gx2 = MRPy(data2[2], data2.fs)-(-12.805798962926817+0)

    print("Média Ay sensor 1",np.mean(ay0))
    print("Média Ay sensor 2",np.mean(ay1))
    print("Média Ay sensor 3",np.mean(ay2))

    print("Média Az sensor 1",np.mean(az0))
    print("Média Az sensor 2",np.mean(az1))
    print("Média Az sensor 3",np.mean(az2))

    print("Média Gx sensor 1",np.mean(gx0))
    print("Média Gx sensor 2",np.mean(gx1))
    print("Média Gx sensor 3",np.mean(gx2))

    ##### ACELERAÇÃO EIXO Y #####

    ty0 = data0.t_axis()
    ty1 = data1.t_axis()
    ty2 = data2.t_axis()

    fig1, x0 = plt.subplots(1,figsize=(15, 4), clear=True)

    ## Leitura do Sensor ##
    x0.plot(ty0, ay0[0], color='red', label='Sensor 1')
    x0.plot(ty1, ay1[0], color='blue', label='Sensor 2')
    x0.plot(ty2, ay2[0], color='green', label='Sensor 3')

```

```

x0.set_xlabel("Tempo (s)", fontsize=12)
x0.set_ylabel("Aceleração no eixo Y (g)", fontsize=12)

x0.set_xlim(limx1, limx2)
x0.set_ylim(limy1, limy2)

x0.tick_params(axis='y', labelszize=12)
x0.tick_params(axis='x', labelszize=12)

x0.set_title(title+" - Acel. eixo Y", fontsize=16)

x0.legend(bbox_to_anchor=(.9, .98), loc='upper left', borderaxespad=0.)

x0.grid(True)
fig1.tight_layout()
#plt.savefig(save+'.png'.format(save));

##### ACELERAÇÃO EIXO Z #####

tz0 = data0.t_axis()
tz1 = data1.t_axis()
tz2 = data2.t_axis()

fig, x1 = plt.subplots(1,figsize=(15, 4), clear=True)

## Leitura do Sensor ##
x1.plot(tz0, az0[0], color='red', label='Sensor 1')
x1.plot(tz1, az1[0], color='blue', label='Sensor 2')
x1.plot(tz2, az2[0], color='green', label='Sensor 3')

x1.set_xlabel("Tempo(s)", fontsize=12)
x1.set_ylabel("Aceleração no eixo Z (g)", fontsize=12)

x1.set_xlim(limx1, limx2)
x1.set_ylim(limy1, limy2)

x1.tick_params(axis='y', labelszize=12)
x1.tick_params(axis='x', labelszize=12)

x1.set_title(title+" - Acel. eixo Z", fontsize=16)

x1.legend(bbox_to_anchor=(.9, .98), loc='upper left', borderaxespad=0.)

```

```

x1.grid(True)
fig.tight_layout()
#plt.savefig(save+'.png'.format(save));

##### GIROSCÓPIO EIXO X #####

tx0    = data0.t_axis()
tx1    = data1.t_axis()
tx2    = data2.t_axis()

fig2, x2 = plt.subplots(1,figsize=(15, 4), clear=True)

## Leitura do Sensor ##
x2.plot(tx0, gx0[0], color='red', label='Sensor 1')
x2.plot(tx1, gx1[0], color='blue', label='Sensor 2')
x2.plot(tx2, gx2[0], color='green', label='Sensor 3')

x2.set_xlabel("Tempo(s)", fontsize=12)
x2.set_ylabel("Taxa de Giro (°/s)", fontsize=12)

x2.set_xlim(limx1, limx2)
x2.set_ylim(limy1*70, limy2*70)

x2.tick_params(axis='y', labelsz=12)
x2.tick_params(axis='x', labelsz=12)

x2.set_title(title+" - Giro. eixo X", fontsize=16)

x2.legend(bbox_to_anchor=(.9, .98), loc='upper left', borderaxespad=0.)

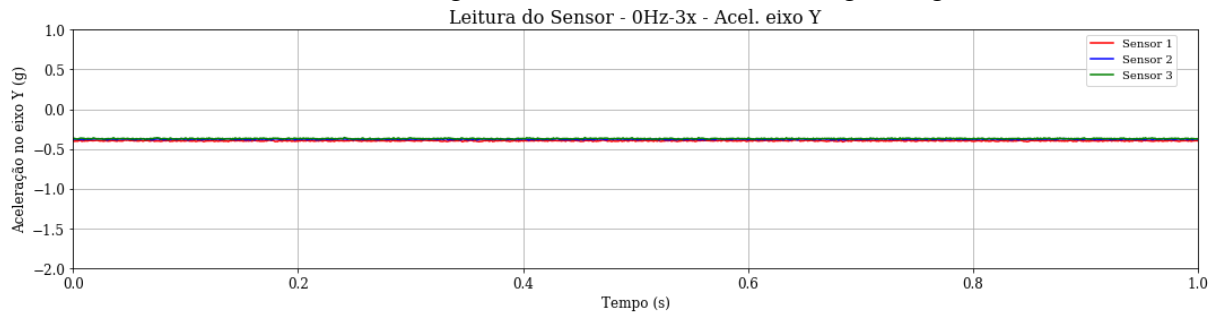
x2.grid(True)
fig2.tight_layout()
#plt.savefig(save+'.png'.format(save));

plt.plot();
plt.show();
#-----#

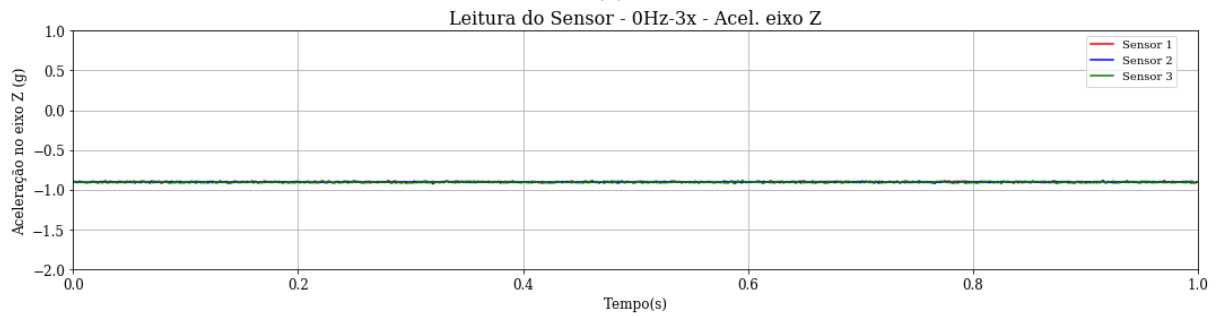
```

ANEXO 10: LEITURA DOS SENSORES A 0HZ

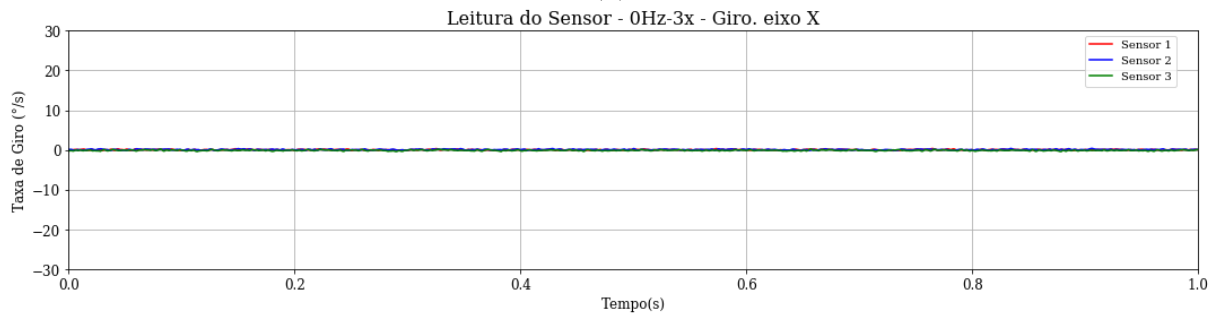
(a) Acelerômetro eixo Y, (b) Acelerômetro eixo Z, (c) Giroscópio eixo X, (d) Periodograma acel. eixo Y, (e) Periodograma acel. eixo Z, (f) Periodograma giro. eixo X



(a)



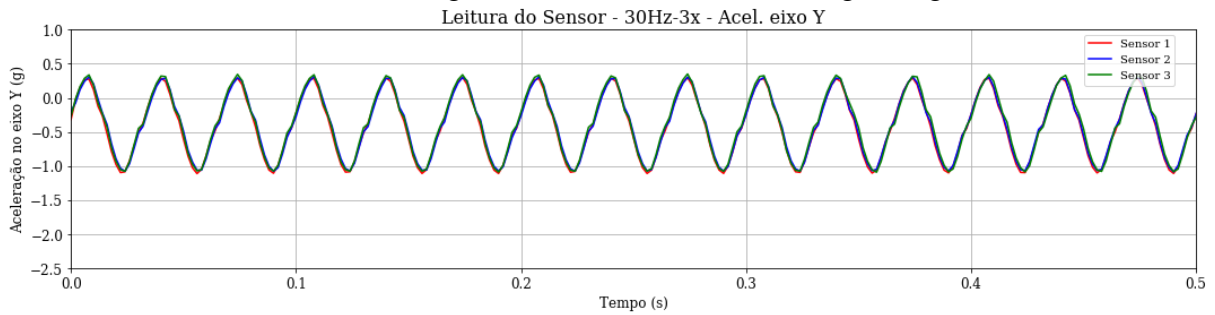
(b)



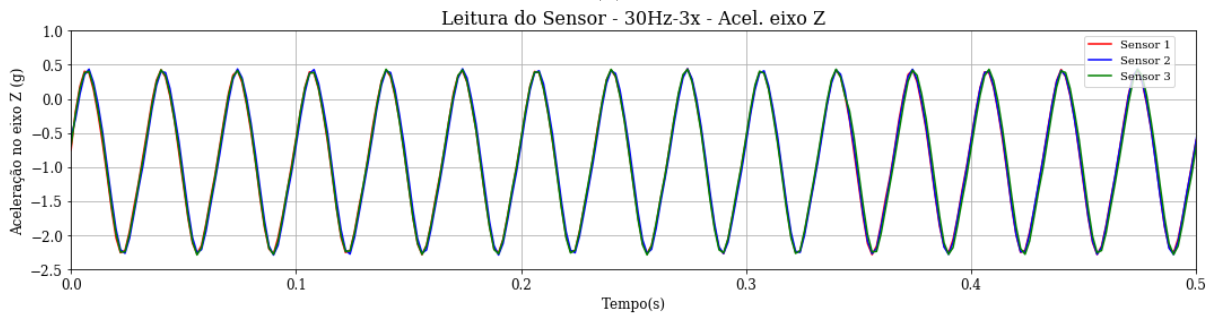
(c)

ANEXO 11: LEITURA DOS SENSORES A 30HZ:

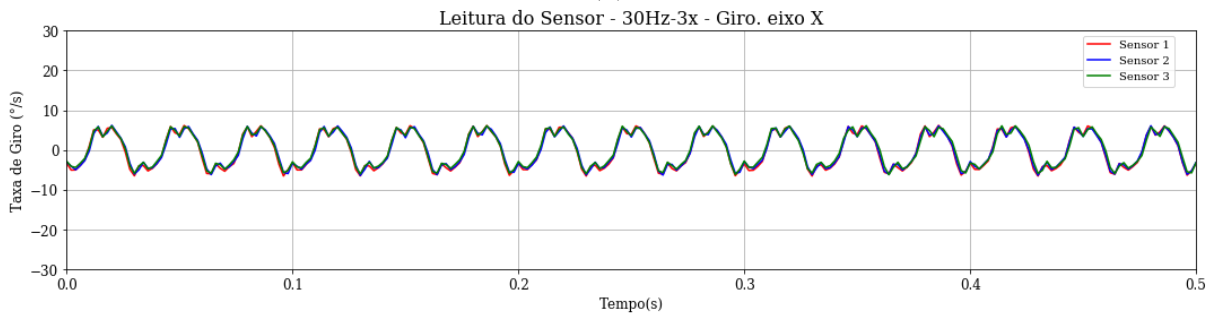
(a) Acelerômetro eixo Y, (b) Acelerômetro eixo Z, (c) Giroscópio eixo X, (d) Periodograma
 acel. eixo Y, (e) Periodograma acel. eixo Z, (f) Periodograma giro. eixo X



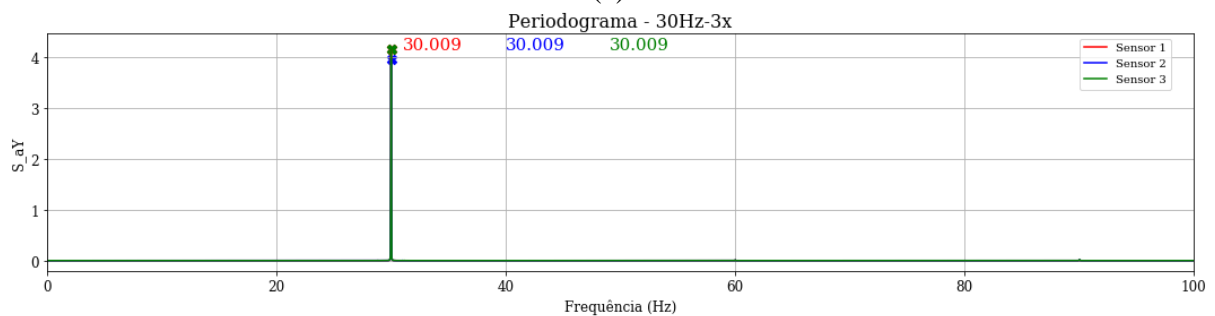
(a)



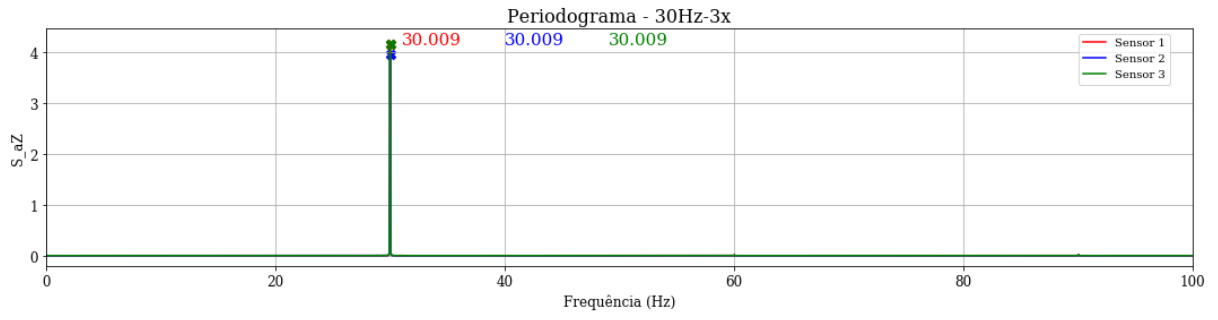
(b)



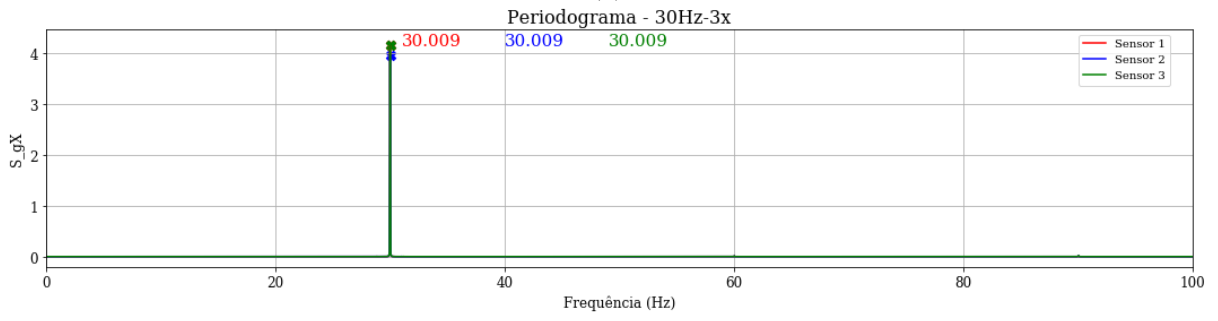
(c)



(d)



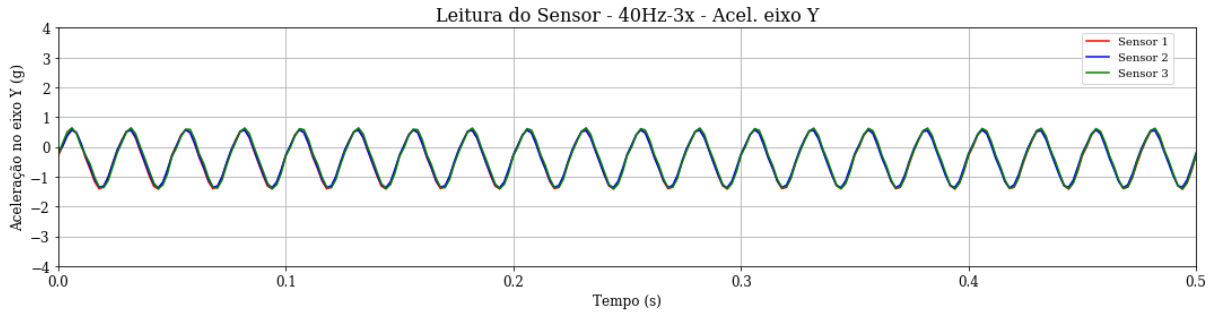
(e)



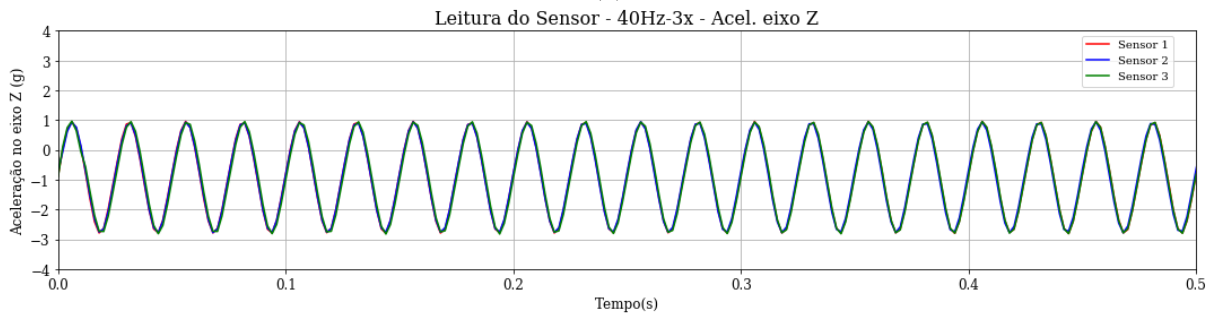
(f)

ANEXO 12: LEITURA DOS SENSORES A 40HZ

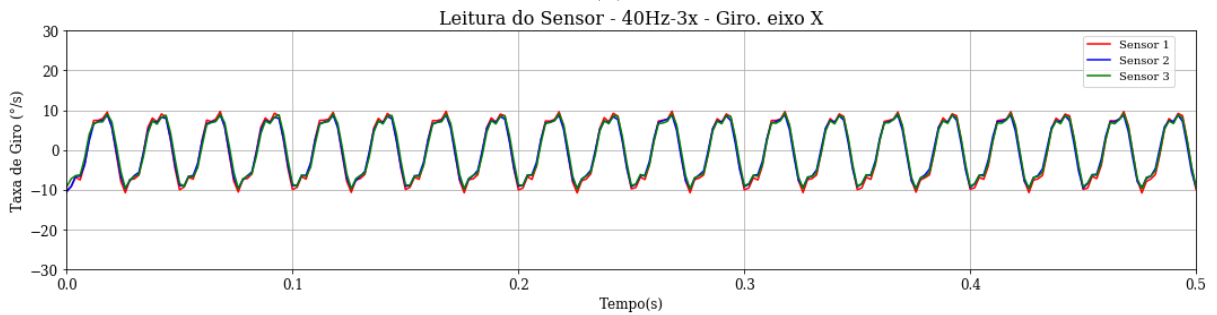
(a) Acelerômetro eixo Y, (b) Acelerômetro eixo Z, (c) Giroscópio eixo X, (d) Periodograma acel. eixo Y, (e) Periodograma acel. eixo Z, (f) Periodograma giro. eixo X



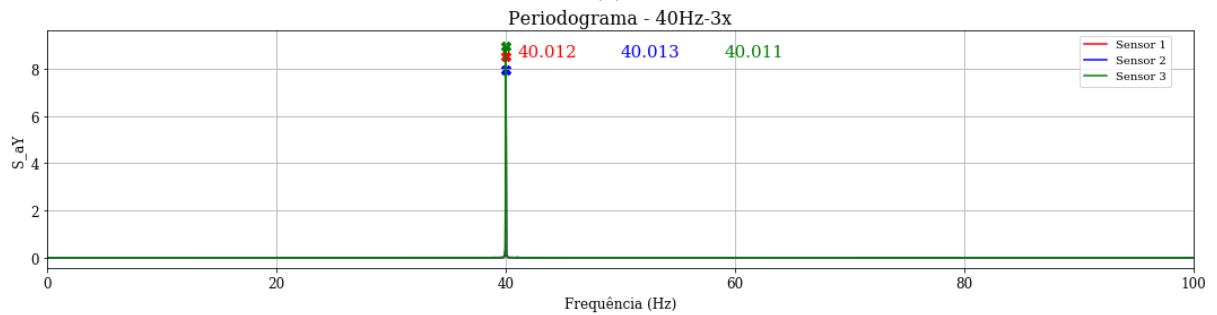
(a)



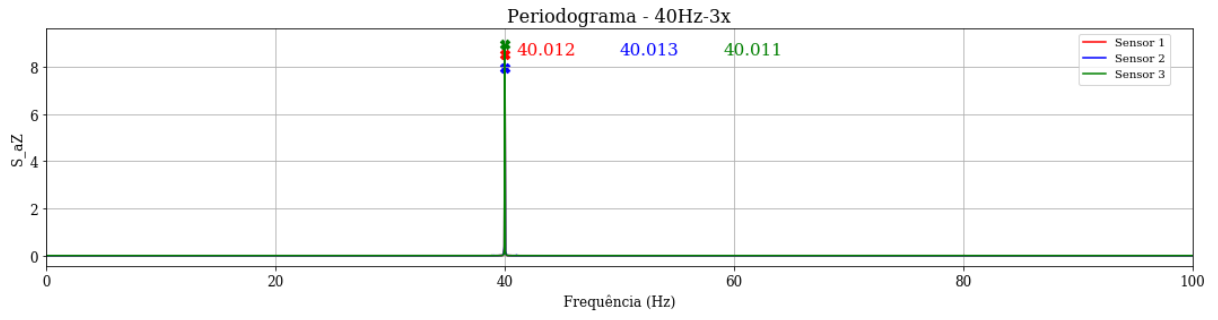
(b)



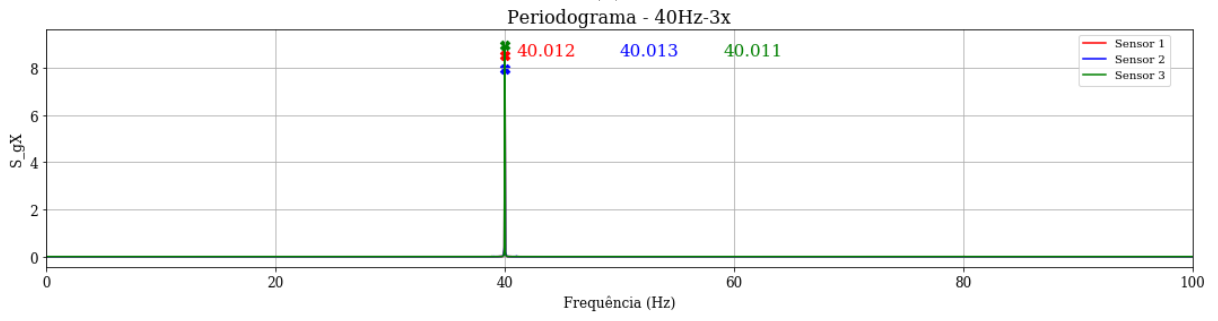
(c)



(d)



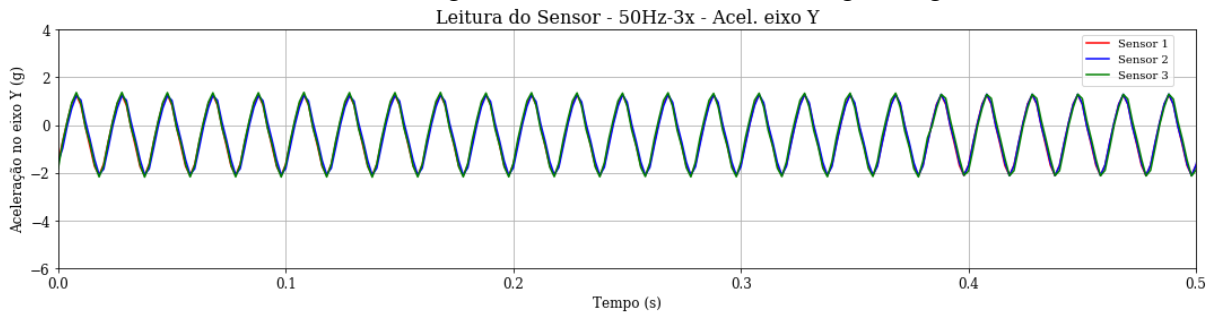
(e)



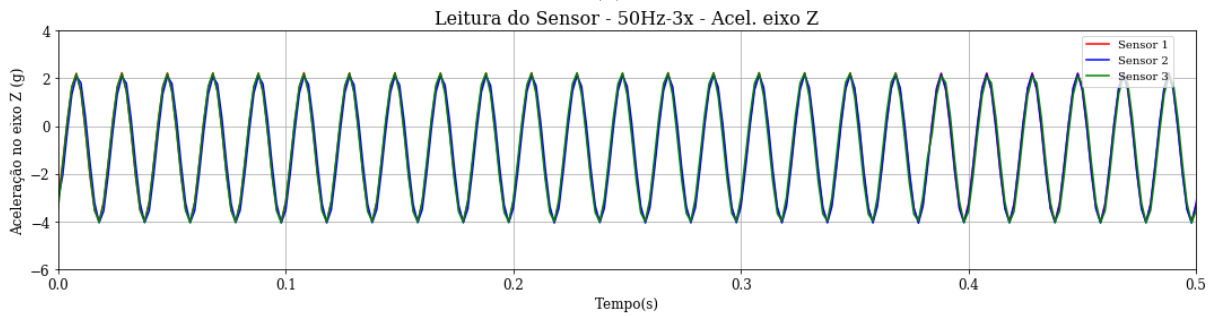
(f)

ANEXO 13: LEITURA DOS SENSORES A 50HZ

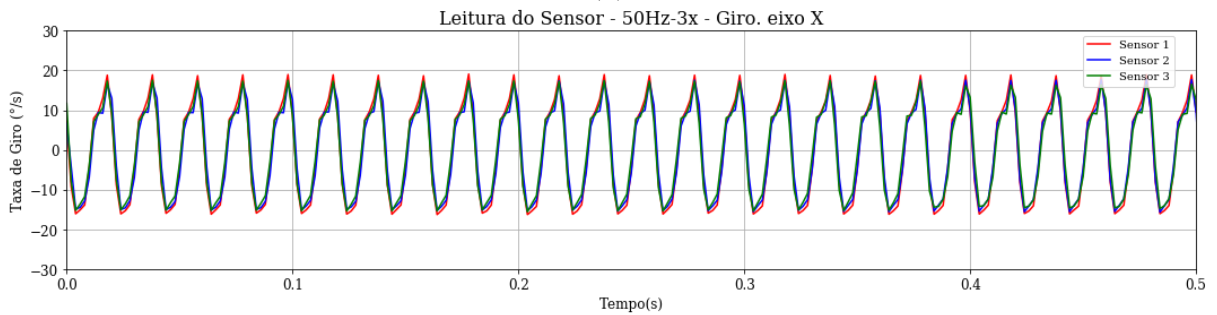
(a) Acelerômetro eixo Y, (b) Acelerômetro eixo Z, (c) Giroscópio eixo X, (d) Periodograma acel. eixo Y, (e) Periodograma acel. eixo Z, (f) Periodograma giro. eixo X



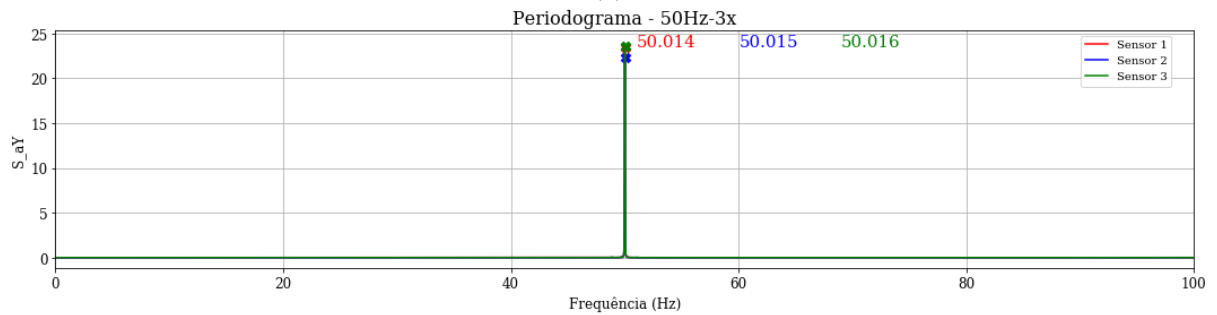
(a)



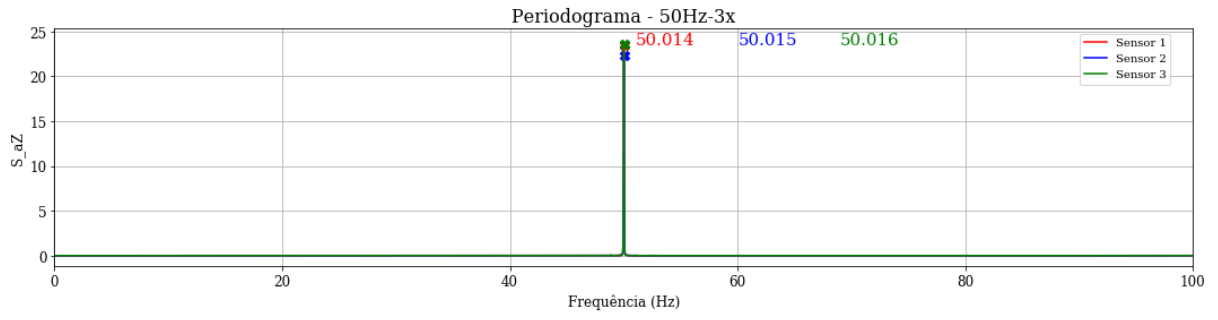
(b)



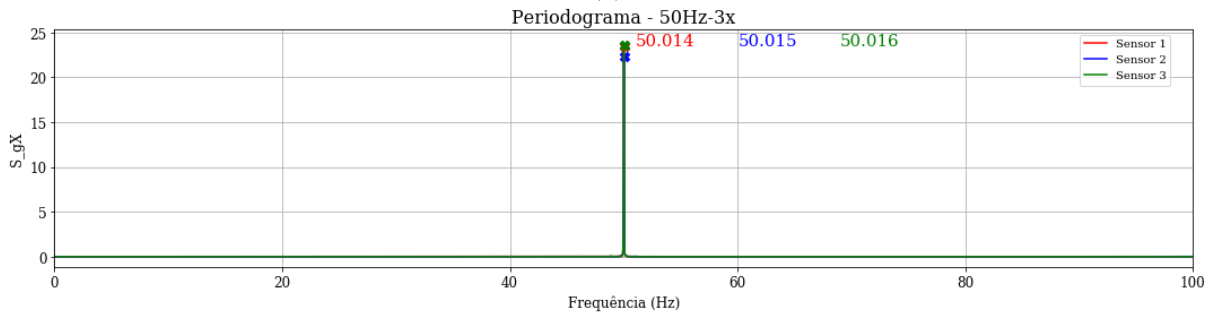
(c)



(d)



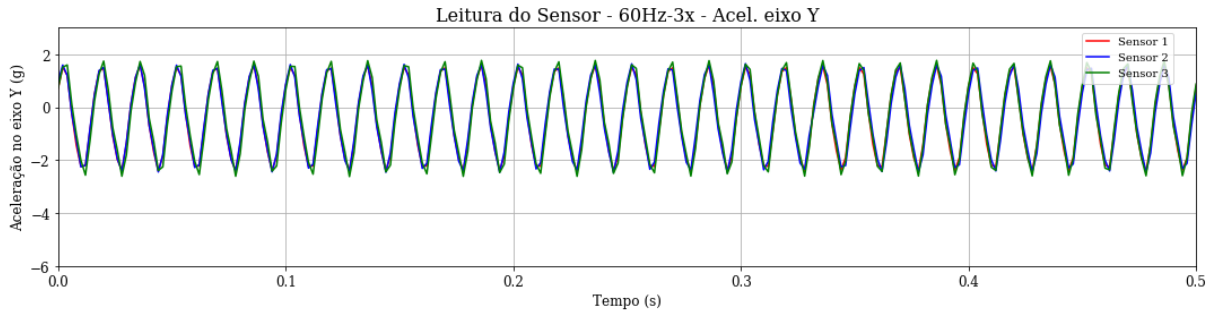
(e)



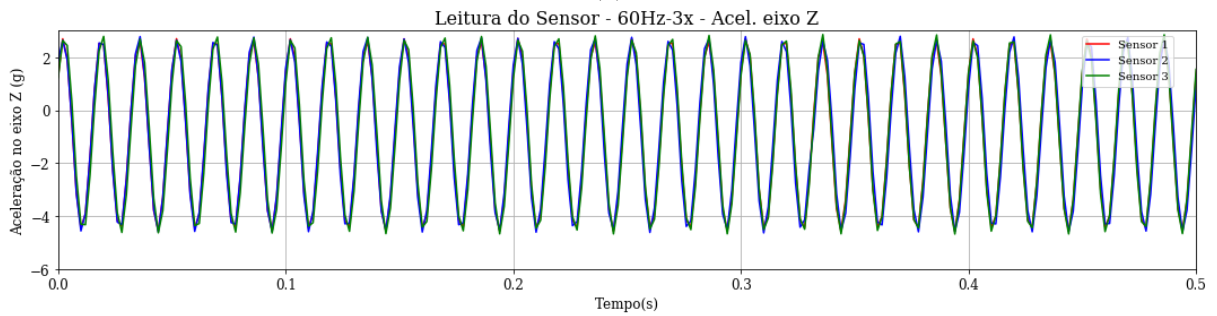
(f)

ANEXO 14: LEITURA DOS SENSORES A 60HZ

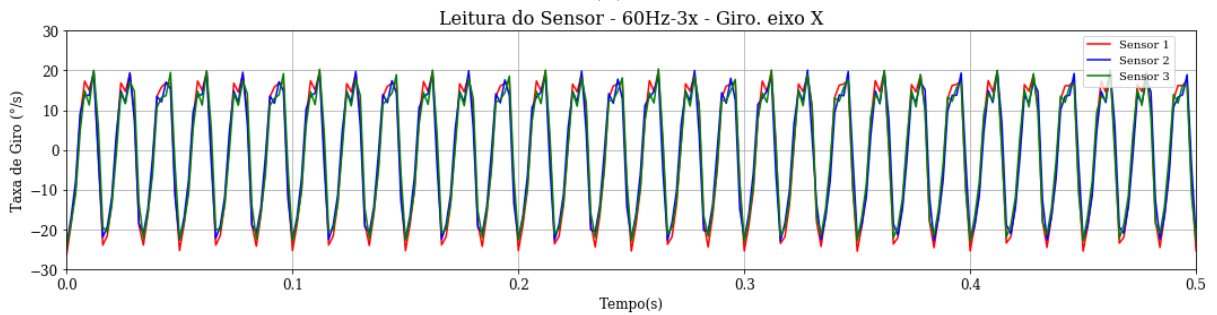
(a) Acelerômetro eixo Y, (b) Acelerômetro eixo Z, (c) Giroscópio eixo X, (d) Periodograma acel. eixo Y, (e) Periodograma acel. eixo Z, (f) Periodograma giro. eixo X



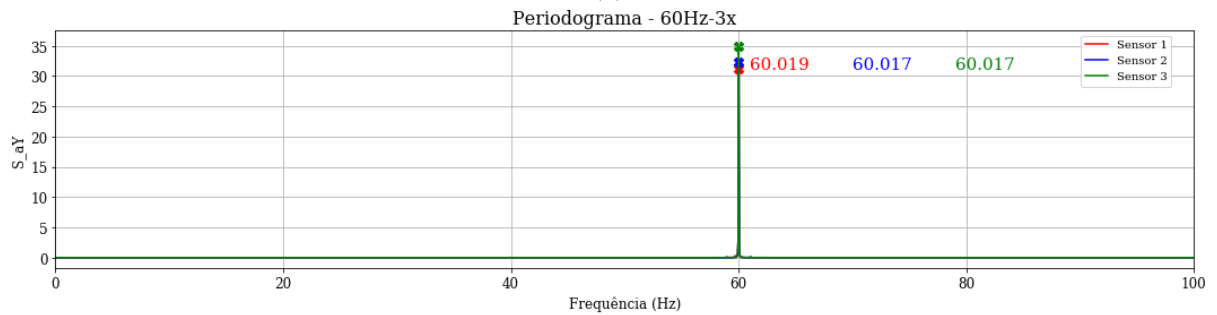
(a)



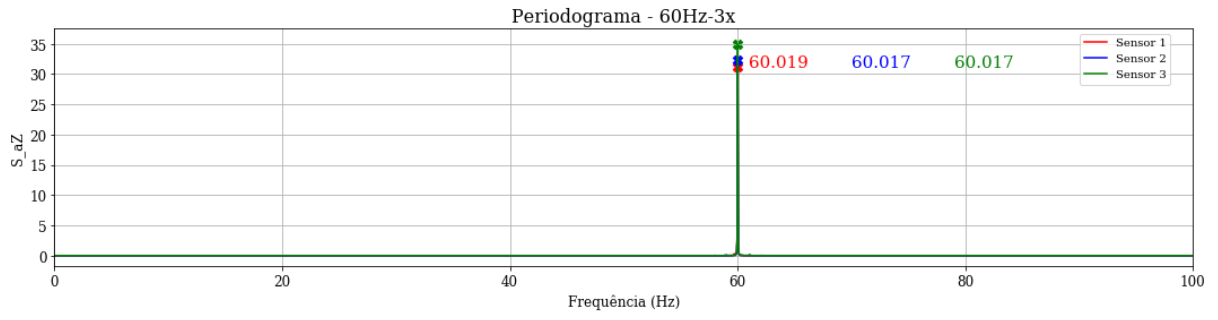
(b)



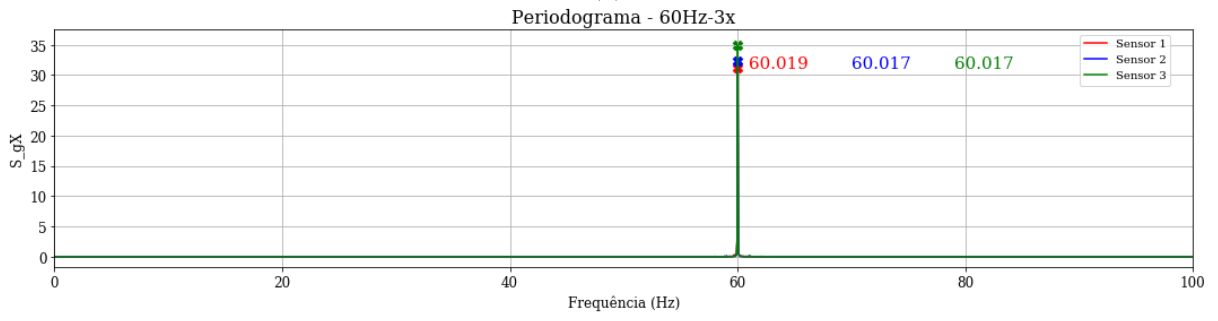
(c)



(d)



(e)



(f)