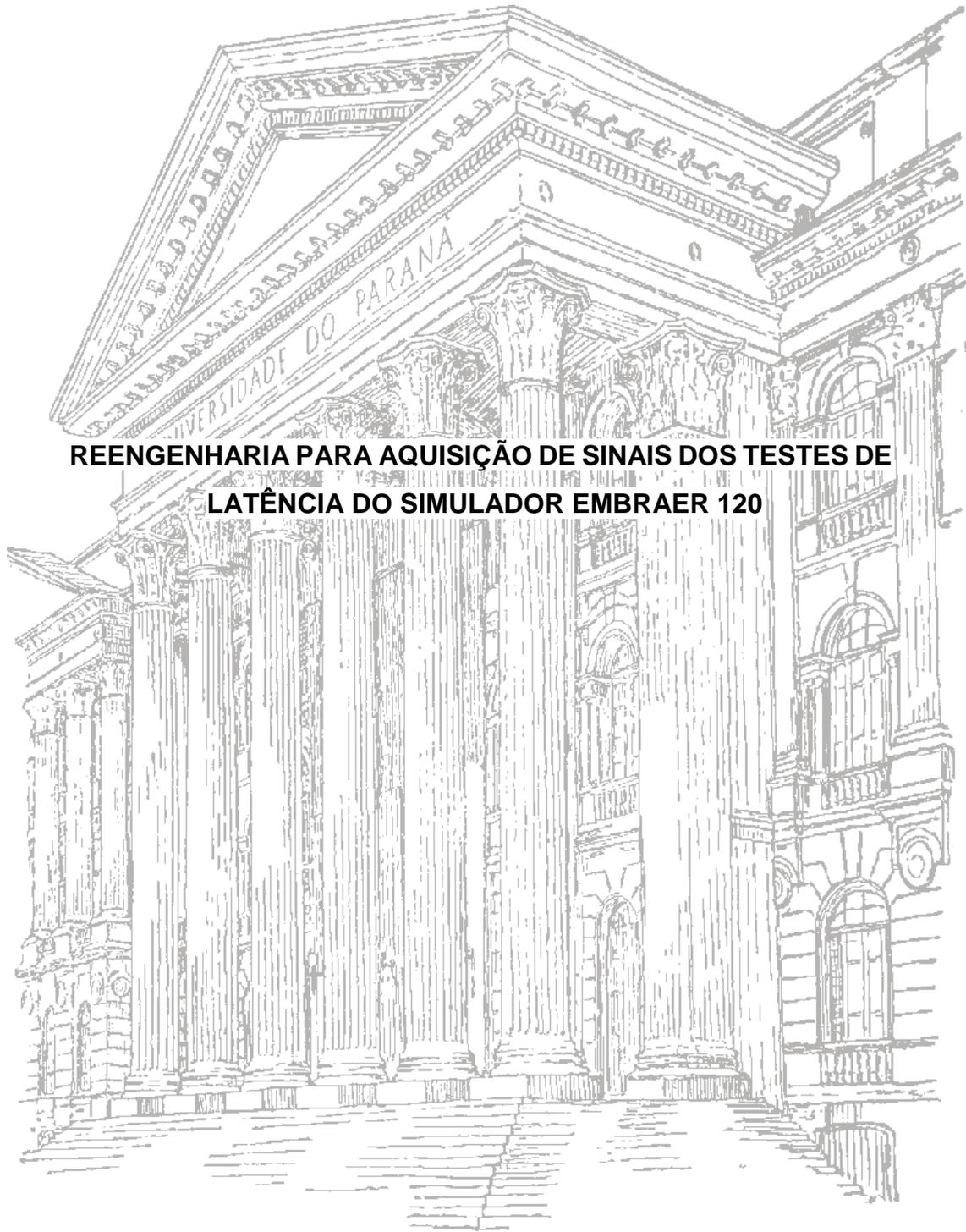


**UNIVERSIDADE FEDERAL DO PARANÁ – UFPR  
ENGENHARIA ELÉTRICA COM ÊNFASE EM SISTEMAS ELETRÔNICOS  
EMBARCADOS**



**REENGENHARIA PARA AQUISIÇÃO DE SINAIS DOS TESTES DE  
LATÊNCIA DO SIMULADOR EMBRAER 120**

**CURITIBA**

**2024**

**Felipe de Oliveira do Carmo**  
**Felipe de Medeiros dos Santos**

**TCC 2**  
**TE347 - TRABALHO DE CONCLUSÃO DE CURSO 2**  
**REENGENHARIA PARA AQUISIÇÃO DE SINAIS DOS TESTES DE LATÊNCIA DO**  
**SIMULADOR EMBRAER 120**

MONOGRAFIA APRESENTADA COMO REQUISITO PARCIAL  
À OBTENÇÃO DO TÍTULO DE ENGENHEIRO ELETRICISTA  
COM ÊNFASE EM SISTEMAS ELETRÔNICOS EMBARCADOS,  
SETOR DE TECNOLOGIA, DEPARTAMENTO DE  
ENGENHARIA ELÉTRICA, UNIVERSIDADE FEDERAL DO  
PARANÁ

ORIENTADOR: Prof. Dr. José Carlos da Cunha  
Coorientador: Rodrigo Gonçalves - EPA

**CURITIBA**  
**2024**

## RESUMO

A aviação moderna exige a contínua evolução dos métodos de treinamento de pilotos, sendo os simuladores *Full Flight Simulator* (FFS) elementos-chave nesse processo. Este trabalho, intitulado "Reengenharia para Aquisição de Sinais dos Testes de Latência do Simulador Embraer 120", concentra-se na otimização da avaliação objetiva de simuladores, com destaque para os testes de latência, essenciais para a validação anual.

A EPA *Training Center*, detentora de um FFS dedicado a aeronave Embraer 120, enfrenta desafios relacionados à aquisição e interpretação dos resultados dos testes de latência. Atualmente, a utilização de um equipamento analógico, o *chart holder*, é o método predominante, porém, este trabalho propõe uma transição para métodos digitalizados mais avançados.

A avaliação dos testes de latência envolve a medição do tempo de resposta dos subsistemas do simulador a partir de movimentos nas superfícies da aeronave. O foco está na otimização desse processo, visando não apenas eficiência, mas também análises mais detalhadas e robustas.

Ao explorar uma solução tecnológica inovadora, este TCC busca ir além da resolução de um desafio técnico específico, aspirando contribuir para a constante evolução no campo da simulação de voo e suas práticas de validação. A implementação dessa proposta promete não apenas aprimorar os padrões da EPA *Training Center*, mas também impactar positivamente o cenário mais amplo da aviação e simulação de voo.

Palavras-chaves: Simulador de voo, aviação, EMB120, testes de latência, treinamento de pilotos, ANAC, *Full Flight Simulator*, *chart holder*, Brasília, *Qualification Test Guide*, QTG, *Pitch, Roll, Yaw*.

## **ABSTRACT**

*Modern aviation requires the continuous evolution of pilot training methods, and simulators Full Flight Simulator (FFS) are key elements in this process. This research, entitled "Reengineering for Signal Acquisition from Embraer 120 Simulator Latency Tests", focuses on the optimization of the objective evaluation of simulators, with emphasis on latency tests, essential for annual validation.*

*The EPA Training Center, holder of an FFS dedicated to the Embraer 120 aircraft, faces challenges related to the acquisition and interpretation of the results of the latency tests. Currently, the use of an analog equipment, the chart holder, is the predominant method, although, this work proposes a transition to more advanced digitized methods.*

*The evaluation of latency tests involves measuring the response time of the simulator subsystems from movements on the surfaces of the aircraft. The focus is on the optimization of this process, aiming not only at efficiency, but also more detailed and solid analyses.*

*In advance of exploring an innovative technological solution, this CBT seeks to go beyond the resolution of a specific technical challenge, aspiring to contribute to the continuous evolution in the field of flight simulation and its validation practices. The implementation of this proposal promises both improvement of the standards of EPA Training Center and positive impact to the broader scenario of aviation and flight simulation.*

*Keywords: Flight simulator, aviation, EMB120, latency tests, pilot training, ANAC, Full Flight Simulator, chart holder, Brasília, Qualification Test Guide, QTG.*

## LISTA DE FIGURAS

FIGURA 1 – Cabine do Simulador Embraer 120 .....	17
FIGURA 2 – Simulador Embraer 120 .....	18
FIGURA 3 – Simulador Embraer 120 visão geral .....	18
FIGURA 4 – CPD visão geral .....	19
FIGURA 5 – CPD computadores do EMB-120 .....	20
FIGURA 6 – Embraer 120 Brasília .....	22
FIGURA 7 – Embraer 121 Xingu .....	22
FIGURA 8 – Detalhe do Elevator na Cauda da Aeronave .....	24
FIGURA 9 – Coluna do Simulador EMB120 .....	25
FIGURA 10 – Coluna do Simulador EMB120 visão frontal.....	25
FIGURA 11 – Detalhe do Airelon nas asas da aeronave.....	26
FIGURA 12 – Manche virado para esquerda.....	26
FIGURA 13 – Detalhe do Rudder na cauda da aeronave.....	27
FIGURA 14 – Detalhe dos Pedais dentro da Cabine do Simulador EMB-120.....	28
FIGURA 15 – Resumo das superfícies de uma aeronave .....	29
FIGURA 16 – Chart Holder.....	31
FIGURA 17 – Panorama do Funcionamento de Testes no Simulador EMB-120... 31	31
FIGURA 18 – EHSI (Electronic Horizontal Situation Indicator) .....	33
FIGURA 19 – Computador central Fokker.....	34
FIGURA 20 – Placa de aquisição do sinal de C/L (FLM).....	34
FIGURA 21 – MEC (Motion Electronic Controller) .....	34
FIGURA 22 – HPU (Hydraulic Power Unit).....	35
FIGURA 23 – Acelerômetros eixo X e Y.....	35
FIGURA 24 – Acelerômetros (vista geral) .....	36
FIGURA 25 – Vital IV .....	37
FIGURA 26 – Placa de aquisição do sinal do visual.....	38
FIGURA 27 – Resultados dos testes de latência apresentados por meio do Chart Holder .....	39
FIGURA 28 – Teste de latência apresentado por meio do simulador ATR 42-500 (YAW Latency) .....	39
FIGURA 29 – Diagrama em Blocos Macro da Solução Proposta .....	41
FIGURA 30 – Diagrama em Blocos Detalhado da Solução Proposta.....	43
FIGURA 31 – Arduino Mega 2560 CH340 com Cabo USB .....	44
FIGURA 32 – LATENCY TEST PANEL.....	46
FIGURA 33 – Placa do Conversor ADS1115 .....	48
FIGURA 34 – Multiplexador duplo CD4052 .....	49

	6
FIGURA 35 – Amplificador Operacional LM124 .....	50
FIGURA 36 – Diodo Zener 5.1V .....	50
FIGURA 37 – Módulo 4MD36 Cartão SD .....	50
FIGURA 38 – Arduino Uno .....	52
FIGURA 39 – Display 1.9 pol IPS Colorido.....	53
FIGURA 40 – Visão Geral (Conversão AD).....	57
FIGURA 41 – Cabo Latency Test Panel.....	58
FIGURA 42 – Circuito Somador AmpOp .....	58
FIGURA 43 – Circuito Somador Usado .....	60
FIGURA 44 – Circuito Ajuste de Faixa com Proteção contra Picos de Tensão Negativo.....	61
FIGURA 45 – Código Seleção MUX.....	63
FIGURA 46 – Código Função readChannel .....	65
FIGURA 47 – Código Inicialização Cartão SD.....	66
FIGURA 48 – Código Formato Cabeçalho .....	66
FIGURA 49 – Código Função startAcquisition.....	67
FIGURA 50 – Código Cabeçalho por Tipo de Teste.....	67
FIGURA 51 – Código Função stopAcquisition.....	68
FIGURA 52 – Display Estado de Espera.....	70
FIGURA 53 – Display Estado de Seleção .....	70
FIGURA 54 – Display Estado de Gravação dos Dados.....	70
FIGURA 55 – Display Estado Salvamento dos Dados .....	71
FIGURA 56 – Comunicação lado CPD.....	71
FIGURA 57 – Comunicação lado Simulador .....	72
FIGURA 58 – Dicionário de Cabeçalhos .....	73
FIGURA 59 – Código Função read_data.....	73
FIGURA 60 – Código Função recover_signals.....	73
FIGURA 61 – Definição de Sinal para Teste de Latência.....	75
FIGURA 62 – Código Associação Cabeçalho e Sinal de Referência.....	75
FIGURA 63 – Código Função Onclick .....	76
FIGURA 64 – Código Cálculo de Latência .....	76
FIGURA 65 – Código Resultado Final Cálculo de Latência.....	76
FIGURA 66 – Comando para Gerar Aplicação Executável .....	77
FIGURA 67 – Hardware CPD.....	78
FIGURA 68 – Hardware Simulador .....	78
FIGURA 69 – Arquivo Pitch.txt .....	79
FIGURA 70 – Arquivo Roll.txt .....	79
FIGURA 71 – Arquivo Yaw.txt .....	79

FIGURA 72 – Dimensões Chart Holder x Protótipo .....	80
FIGURA 73 – Gráfico Teste Pitch e Cálculo de Latência .....	82
FIGURA 74 – Gráfico Teste Roll e Cálculo de Latência .....	82
FIGURA 75 – Gráfico Teste Yaw e Cálculo de Latência .....	83
FIGURA 76 – Arquivo .PDF por Tipo de Teste (Pitch, Roll e Yaw).....	83
FIGURA 77 – Arquivo Executável da Aplicação .....	84

**LISTA DE TABELAS**

TABELA 1 – TERMOS PRIMORDIAIS PARA OPERAÇÃO DE AERONAVES.....	28
TABELA 2 – SINAIS TESTE DE LATÊNCIA POR SUPERFÍCIE .....	47
TABELA 3 – CHAVEAMENTO CD4052 .....	64
TABELA 4 – CHAVEAMENTO MULTIPLEXADOR POR TIPO DE TESTE.....	64
TABELA 5 – ESTADOS DO SISTEMA.....	69
TABELA 6 – MENSAGENS NO DISPLAY POR ESTADO DO SISTEMA .....	69

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	CONTEXTO DO PROBLEMA.....	11
1.2	OBJETIVOS .....	12
1.2.1	Objetivo geral.....	12
1.2.2	Objetivos específicos.....	12
1.3	JUSTIFICATIVA.....	13
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>15</b>
2.1	SIMULADORES NO ÂMBITO NACIONAL E INTERNACIONAL .....	15
2.2	SIMULADOR EMB120.....	16
2.2.1	EMB-120 Aeronave.....	21
2.2.2	Superfícies da aeronave.....	24
2.2.2.1	<i>Elevator</i> .....	24
2.2.2.2	<i>Aileron</i> .....	25
2.2.2.3	<i>Rudder</i> .....	27
2.3	<i>QUALIFICATION TEST GUIDE (QTG)</i> .....	29
2.3.1	TESTE DE LATÊNCIA .....	30
2.3.2	Subsistemas do Simulador.....	31
2.3.2.1	Subsistemas - <i>I/O</i> .....	32
2.3.2.2	Subsistemas - <i>Control Loading</i> .....	33
2.3.2.3	Subsistemas - <i>Motion</i> .....	35
2.3.2.4	Subsistemas - <i>Visual</i> .....	36
2.3.3	Comparativo entre Testes de Latência - EMB120 x Simulador mais atual .....	38
<b>3</b>	<b>METERIAIS E MÉTODOS .....</b>	<b>40</b>
3.1	CENÁRIO ATUAL.....	40
3.2	SOLUÇÃO PROPOSTA .....	40
3.3	<i>HARDWARE CPD</i> .....	43
3.3.1	Unidade microprocessada e Comunicação com Hardware Simulador .....	44
3.3.2	Conversão AD dos sinais .....	45
3.3.3	Saída dos dados para plotagem gráfica (cartão SD).....	50
3.4	<i>HARDWARE SIMULADOR</i> .....	52
3.4.1	Unidade microprocessada .....	52
3.4.2	Interface com aplicador de teste.....	52
3.4.3	Comunicação com <i>Hardware CPD</i> .....	53
3.4.4	Periféricos.....	53
3.5	PLOTAGEM GRÁFICA.....	54
3.5.1	Linguagem/bibliotecas.....	54
3.5.2	Leitura cartão SD .....	55

	10
3.5.3 Funcionamento (.EXE).....	55
3.5.4 Resultados .....	56
<b>4 DESENVOLVIMENTO .....</b>	<b>57</b>
4.1 CONVERSÃO DOS SINAIS ANALÓGICOS PARA DIGITAIS.....	57
4.1.1 Entrada dos Sinais no <i>Hardware</i> CPD.....	57
4.1.2 Ajuste da Faixa de Tensão dos Sinais.....	58
4.1.3 Seleção de Sinais por Tipo de Teste.....	61
4.1.4 Leitura do Conversor ADS1115.....	65
4.2 GRAVAÇÃO DOS DADOS NO CARTÃO SD.....	66
4.3 INTERFACE COM PILOTO .....	68
4.4 COMUNICAÇÃO ENTRE <i>HARDWARE</i> CPD E <i>HARDWARE</i> SIMULADOR.....	71
4.5 PLOTAGEM GRÁFICA.....	72
4.5.1 Leitura dos Dados do Cartão SD.....	72
4.5.2 Reconversão dos Dados.....	73
4.5.3 Definição do Tipo de Teste.....	75
4.5.4 Cálculo de Latência .....	75
4.5.5 Aplicação executável.....	77
<b>5 RESULTADOS.....</b>	<b>78</b>
5.1 <i>HARDWARE E SOFTWARE</i> .....	78
5.1.1 Comparativo entre novo <i>Hardware</i> x <i>Chart Holder</i> .....	80
5.1.2 Comparativo entre Software Embarcado x <i>Chart Holder</i> .....	80
5.2 PLOTAGEM GRÁFICA.....	81
<b>6 CONCLUSÃO .....</b>	<b>85</b>
<b>REFERÊNCIAS.....</b>	<b>87</b>

# 1 INTRODUÇÃO

## 1.1 CONTEXTO DO PROBLEMA

No vasto universo da aviação, a primazia da segurança e a constante busca pela excelência no treinamento de pilotos convergem para a essencialidade dos simuladores *Full Flight Simulator* (FFS). Desempenhando um papel crucial na formação e certificação, esses simuladores, como o dedicado ao avião Embraer 120 na EPA *Training Center*, são submetidos a uma rigorosa validação anual, requisito indispensável estabelecido por órgãos reguladores, como a Agência Nacional de Aviação Civil (ANAC). Este trabalho, intitulado "Digitalização e Análise de Dados dos Testes de Latência para Simuladores de Voo", concentra-se na resolução de um desafio específico enfrentado por esse centro de treinamento, envolvendo a avaliação objetiva do simulador em questão.

A avaliação objetiva, segmentada em testes prescritos pelo *Qualification Test Guide* (QTG), é uma etapa crítica nesse processo. Dentre esses testes, destacam-se os de latência, cujo propósito é medir o tempo de resposta dos subsistemas do simulador, compreendendo o computador do *host*, o sistema de *motion* e a renderização visual. Essa análise minuciosa é conduzida a partir de movimentos iniciais, gerados nas três principais superfícies do avião simulado: *elevator*, *rudder* e *aileron*.

Atualmente, a metodologia adotada para a aquisição e interpretação dos resultados desses testes é baseada em um método analógico, empregando um dispositivo conhecido como *chart holder*. Este equipamento, análogo a um osciloscópio de múltiplas vias, realiza a impressão em tempo real dos sinais provenientes do simulador, utilizando papel térmico. Contudo, diante da dinâmica evolução tecnológica, surge a imperativa necessidade de transição para métodos mais avançados e eficientes.

O *chart holder*, instrumento crucial nesse procedimento, torna-se objeto de aprimoramento neste trabalho. A proposta de mudança visa otimizar não apenas o processo de aquisição e análise de dados dos testes de latência, mas também fomentar a exploração de análises mais detalhadas e robustas. Nesse contexto, o TCC emerge como uma iniciativa dedicada a explorar e implementar uma solução tecnológica que não apenas atenda às demandas imediatas da EPA *Training Center*, mas também ofereça uma abordagem inovadora para a aquisição, digitalização e análise dos dados cruciais provenientes dos testes de latência em simuladores de voo.

Ao longo deste trabalho, será lançado um olhar atento sobre os desafios inerentes a essa transição, detalharemos a metodologia adotada para a digitalização e análise de dados, e serão apontados potenciais impactos e benefícios dessa abordagem inovadora no

contexto da validação de simuladores de voo. Mais que a solução do desafio técnico, o projeto contribui para a evolução constante no campo da simulação de voo e suas práticas de validação.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

O objetivo deste trabalho é desenvolver um sistema de aquisição e tratamentos dos sinais analógicos específicos do simulador EMB-120 da EPA *Training Center*, com foco nos testes de latência conforme estabelecido no *Qualification Test Guide* (QTG) do simulador. Este sistema terá como finalidade corrigir o problema da falta de avanço de tecnologia do equipamento atualmente utilizado para a aquisição e análise dos resultados desses testes na EPA *Training Center*.

### 1.2.2 Objetivos específicos

1. Digitalização dos Sinais Analógicos: Desenvolver um método para capturar e converter os sinais analógicos provenientes dos testes de latência em formato digital, permitindo uma aquisição mais precisa e eficiente dos dados.
2. Tratamento e Análise dos Dados: Implementar algoritmos de tratamento de sinal para processar e analisar os dados digitalizados, permitindo uma interface intuitiva para o técnico que irá avaliar os sinais, possibilitando a identificação de tendências e anomalias nos resultados dos testes de latência com maior assertividade.
3. Validação do Sistema: Realizar testes e validações do sistema desenvolvido para garantir sua eficácia e confiabilidade na aquisição e análise dos resultados dos testes de latência em simuladores de voo, comparando-os com os métodos tradicionais atualmente utilizados.

### 1.3 JUSTIFICATIVA

A necessidade de desenvolver um sistema de aquisição e tratamentos dos sinais analógicos específicos de simuladores de voo, com ênfase nos testes de latência conforme estabelecido no *Qualification Test Guide (QTG)*, surge da urgência em resolver o problema de obsolescência do equipamento atualmente utilizado para a aquisição e análise desses resultados na EPA *Training Center*. A justificativa para este trabalho é fundamentada nos seguintes pontos:

1. **Manutenção da Conformidade Regulatória:** A validação anual dos simuladores de voo é um requisito regulatório essencial estabelecido pela Agência Nacional de Aviação Civil (ANAC) e outras autoridades de aviação civil. A garantia da conformidade com os padrões estabelecidos no QTG do simulador é fundamental para manter a certificação e a operação desses dispositivos, garantindo a segurança e a eficácia dos treinamentos de pilotos.
2. **Necessidade de Atualização Tecnológica:** O equipamento atualmente utilizado para a aquisição e análise dos resultados dos testes de latência na EPA *Training Center* está enfrentando problemas de obsolescência, apresentando limitações em termos de precisão, eficiência e capacidade de processamento de dados. Diante da rápida evolução tecnológica, torna-se imperativo buscar soluções mais avançadas e sustentáveis para atender às demandas atuais e futuras da indústria da aviação.
3. **Melhoria da Eficiência Operacional:** A implementação de um sistema de aquisição e tratamentos dos sinais analógicos digitalizados e a análise automatizada dos resultados dos testes de latência oferecerá benefícios significativos em termos de eficiência operacional na EPA *Training Center*. Essa abordagem permitirá uma aquisição mais rápida e precisa de dados, facilitando a detecção precoce de quaisquer anomalias ou desvios nos testes de latência, o que, por sua vez, possibilitará uma tomada de decisão mais ágil e eficaz.
4. **Contribuição para a Inovação Tecnológica:** Além de atender às necessidades específicas da EPA *Training Center*, este trabalho contribuirá para o avanço da tecnologia na área de simulação de voo. A implementação de soluções mais avançadas e inteligentes para a aquisição e análise de dados dos testes de latência pode servir como um exemplo de boas práticas para outras instituições de treinamento de pilotos e fabricantes de simuladores de voo.

Em resumo, a justificativa para este trabalho está fundamentada na importância da manutenção da conformidade regulatória, na necessidade de atualização tecnológica, na melhoria da eficiência operacional e na contribuição para a inovação tecnológica na área de simulação de voo. Ao abordar esses aspectos, busca-se não apenas resolver um problema imediato enfrentado pela EPA *Training Center*, mas também promover o avanço e a excelência na formação e certificação de pilotos.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão abordados os simuladores de voo com suas características e funcionalidades, incluindo o modelo específico para o qual este trabalho está sendo desenvolvido, o Simulador Embraer 120. O entendimento abrangente dos simuladores de voo, tanto em nível nacional quanto internacional, é crucial para uma análise aprofundada das práticas e tecnologias empregadas na simulação de voo.

Inicialmente será explorado o panorama dos simuladores de voo em uma escala global, destacando suas diferentes aplicações e avanços tecnológicos. Em seguida, o foco passa a ser o Simulador EMB120, examinando tanto a aeronave EMB-120 quanto as especificidades do simulador que replica suas características e comportamento em voo.

Além disso, será discutida a importância do *Qualification Test Guide* (QTG) na avaliação e certificação de simuladores de voo, delineando seu papel na garantia da precisão e confiabilidade dos resultados obtidos durante o treinamento.

Por fim, aspectos técnicos essenciais serão detalhados, como os testes de latência e os subsistemas do simulador, destacando suas funções e influências no desempenho e realismo da simulação. Este capítulo proporcionará uma base sólida para compreender o contexto e os elementos-chave relacionados aos simuladores de voo, tornando possível o desenvolvimento do projeto proposto.

### 2.1 SIMULADORES NO ÂMBITO NACIONAL E INTERNACIONAL

A simulação de voo, atualmente, desempenha um papel fundamental no treinamento de pilotos de aeronaves, sendo reconhecida como uma das ferramentas mais valiosas para garantir a segurança e a eficácia no contexto da aviação.

Em junho de 2018, a quantidade de simuladores de linhas aéreas comerciais em serviço totalizava 1.270 unidades. Dessa base instalada, a empresa CAE contribuiu com 56%, enquanto a L3 CTS representou 20% e a *Flight Safety International*, 10%. Regionalmente, a América do Norte detinha 38% dos simuladores de voo do mundo, seguida pela região Ásia-Pacífico com 25%, e a Europa com 24%. Em termos de fabricantes de aeronaves simuladas, os modelos Boeing lideravam com 45% de participação, seguidos pela Airbus com 35%, Embraer com 7%, *Bombardier* com 6%, e ATR com 3% (QUARTZO, 2024).

Para o treinamento e capacitação de pilotos, os simuladores de voo *Full Flight Simulator* (FFS) desempenham um papel essencial. Esses equipamentos são reproduções detalhadas dos instrumentos, equipamentos, painéis e controles de aeronaves específicas,

proporcionando uma representação precisa das operações em solo e em voo. Os *FFS* podem ser classificados em diferentes níveis de qualificação, variando de *FFS A* a *FFS D*, sendo este último o mais abrangente e representativo em termos de simulação de aeronaves e sistemas (GOV, 2021).

Os simuladores de voo permitem realizar treinamentos completos ou quase completos de uma aeronave, oferecendo uma experiência imersiva e realista para os pilotos. Para serem utilizados em conformidade com os requisitos de treinamento, exames de proficiência ou experiências de voo exigidas pelas regulamentações brasileiras de aviação civil (RBAC), os dispositivos de treinamento devem ser qualificados e aprovados pela Agência Nacional de Aviação Civil (ANAC).

A qualificação do dispositivo envolve uma avaliação rigorosa de acordo com os níveis e requisitos estabelecidos no RBAC 60, que define os padrões para simuladores de voo e dispositivos de treinamento. Já a aprovação do dispositivo inclui sua inclusão em cursos ou programas de treinamento conforme os requisitos estabelecidos em regulamentações específicas, como o RBAC 61, RBAC 142, RBAC 121, entre outros (ANAC, 2020).

Na América do Sul, a *EPA Training Center* se destaca como referência no segmento de treinamento de aviação civil. É o único centro de treinamento na América Latina a abrigar o simulador *Full Flight Simulator (FFS)* do EMB-120, qualificado como nível B, além de ser o único centro de formação e capacitação de pilotos para a aeronave da linha *ATR 42-500* qualificado como nível C (EPA, 2021).

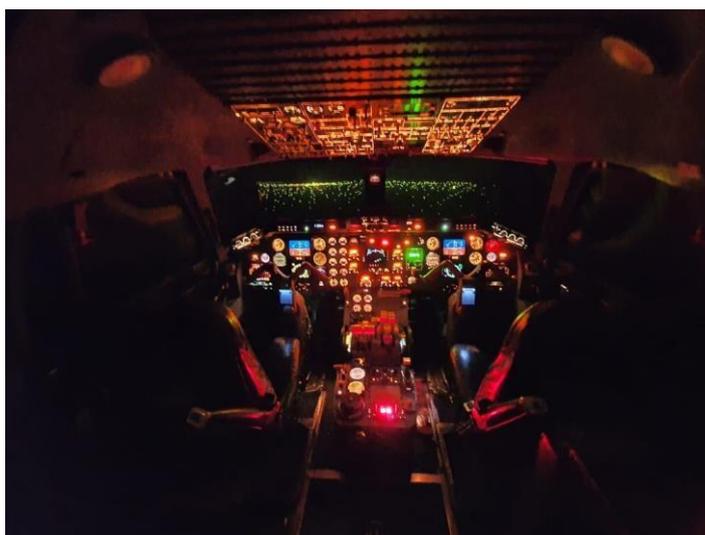
Atualmente, a *EPA Training Center* conta com uma impressionante infraestrutura composta por três simuladores *Full Flight Simulator (FFS)*: o EMB-120, o *ATR42/72* e o EMB-145 (Nível D). Além disso, possui dois dispositivos de treinamento de voo (*FTD - Flight Training Device*): o EMB-110 e o EMB-190. Essa variedade de simuladores permite à *EPA Training Center* oferecer um amplo espectro de treinamentos e capacitações para pilotos, abrangendo diferentes modelos de aeronaves e níveis de qualificação (EPA, 2021).

## 2.2 SIMULADOR EMB120

O simulador *FFS (Full Flight Simulator)* do Embraer EMB-120 Brasília representa um componente vital no treinamento de pilotos para operar esta aeronave regional turboélice. Desenvolvido por diversos fabricantes de simulação de voo, tais como a *FlightSafety International*, *CAE Inc*, Embraer e outras empresas especializadas, este dispositivo oferece uma representação altamente realista das características de voo, sistemas e procedimentos associados ao Embraer 120 (EPA, 2021).

Com uma cabine de voo meticulosamente projetada e controles totalmente funcionais, o simulador do Embraer 120 proporciona uma experiência de treinamento imersiva e segura. Os pilotos têm a capacidade de praticar uma ampla variedade de procedimentos, desde operações normais até situações de emergência, em condições simuladas de voo que replicam com precisão o ambiente real, conforme FIGURA 1 (EPA, 2021).

FIGURA 1 – Cabine do Simulador Embraer 120



FONTE: Os autores (2024).

Além disso, o simulador do Embraer 120 é certificado pelas autoridades de aviação civil, tais como a *Federal Aviation Administration (FAA)* nos Estados Unidos e a Agência Nacional de Aviação Civil (ANAC) no Brasil, garantindo que atenda aos padrões exigidos para o treinamento de pilotos (FAA, 2004).

Por meio da utilização do simulador do Embraer 120, os pilotos têm a oportunidade de aprimorar suas habilidades, adquirir familiaridade com os sistemas da aeronave e praticar procedimentos críticos sem expor passageiros ou tripulação a qualquer risco. Isso torna o simulador uma ferramenta indispensável para garantir a segurança e a eficácia das operações aéreas comerciais (EPA, 2021).

O simulador abordado neste trabalho é um EMB 120 fabricado pela Embraer na década de 1980, projeto concebido em São José dos Campos, SP. Sua equipe de engenheiros dedicou-se ao desenvolvimento de todos os sistemas para assegurar que o simulador atenda aos mais altos padrões de realismo dentro de sua qualificação, atualmente classificado como nível B (AC-120-40A) (EPA, 2021).

O referido simulador (FIGURA 2 e FIGURA 3), atualmente localizado no EPA Training Center, chegou em 2006 a Curitiba, tornando-se o primeiro simulador adquirido

pela Escola Paranaense de Aviação. Desde então, tem operado continuamente, sendo disponibilizado para os treinamentos necessários. Destaca-se que é o único centro de treinamento da América Latina a operar este modelo específico de simulador (EPA, 2021).

FIGURA 2 – Simulador Embraer 120



FONTE: Os autores (2024).

FIGURA 3 – Simulador Embraer 120 visão geral



FONTE: Os autores (2024).

Analisando o equipamento, o simulador é composto por uma variedade de computadores e subsistemas, cada um desempenhando um papel específico para o funcionamento adequado do simulador. Os subsistemas em questão são: *Host*, *Control Loading*, *Motion* e *Visual*. Cada um desses subsistemas desempenha uma função fundamental no processo de simulação e será detalhado na sequência desta monografia.

Este arranjo de subsistemas demonstra a complexidade e a integração necessárias para criar uma experiência de simulação realista e precisa. Cada componente desempenha um papel crucial na replicação das condições de voo e na resposta às entradas do piloto, proporcionando um ambiente de treinamento imersivo e eficaz.

Na sequência deste trabalho, cada subsistema será examinado em detalhes, destacando sua função, importância e contribuição para o funcionamento global do simulador. Isso permitirá uma compreensão abrangente da arquitetura e operação do equipamento, proporcionando uma base sólida para o desenvolvimento de soluções e melhorias propostas.

Para garantir o funcionamento correto desses computadores e subsistemas, é essencial que eles sejam instalados em uma sala especialmente designada para este fim, conhecida como Centro de Processamento de Dados (CPD). Esta sala é projetada para fornecer condições ideais para a operação dos equipamentos, incluindo controle preciso de temperatura e acesso restrito. A FIGURA 4 apresenta uma visão geral desta sala.

FIGURA 4 – CPD visão geral



FONTE: Os autores (2024)

A temperatura dentro do CPD é mantida em um nível constante de 19°C, o que ajuda a garantir que os componentes eletrônicos não superaqueçam e funcionem de maneira eficiente e confiável. O controle rigoroso da temperatura é essencial para evitar danos aos equipamentos e garantir a estabilidade operacional do simulador.

Além disso, o acesso ao CPD é estritamente controlado e limitado apenas ao pessoal autorizado. Isso é feito para proteger os equipamentos contra danos acidentais, interferências externas ou acesso não autorizado, garantindo assim a segurança e a integridade dos sistemas críticos do simulador.

Em resumo, o CPD desempenha um papel crucial no suporte ao funcionamento adequado dos computadores e subsistemas do simulador, fornecendo um ambiente controlado e seguro para garantir sua operação confiável e eficiente. A FIGURA 5 apresenta a repartição do CPD onde fica alocados os computadores do EMB-120.

FIGURA 5 – CPD computadores do EMB-120



FONTE: Os autores (2024)

Como mencionado anteriormente, o processo de qualificação junto à Agência Nacional de Aviação Civil (ANAC) é um procedimento anual que determina a certificação do simulador. Esta avaliação anual é essencial para garantir que o simulador atenda aos padrões e regulamentos estabelecidos pela ANAC, assegurando sua adequação para fornecer treinamento de pilotos.

Durante esse processo de qualificação, o simulador é submetido a uma série de testes rigorosos que avaliam sua precisão, confiabilidade e conformidade com os requisitos regulatórios. Esses testes abrangem uma variedade de aspectos, incluindo desempenho de sistemas, fidelidade de simulação, resposta às entradas do piloto e procedimentos de emergência.

O resultado desse processo de qualificação é a certificação do simulador pela ANAC, com uma validade de um ano. Essa certificação atesta que o simulador cumpre todos os requisitos estabelecidos pelas autoridades de aviação civil e está apto para fornecer treinamento de alta qualidade e seguro para os pilotos.

Portanto, a certificação anual concedida pela ANAC é um selo de aprovação crucial que valida a capacidade do simulador de cumprir os mais altos padrões de segurança e desempenho exigidos pela indústria da aviação. Essa certificação reafirma o compromisso da EPA Training Center com a excelência no treinamento de pilotos e garante a confiança

dos alunos e das companhias aéreas na qualidade de seus programas de treinamento (EPA, 2021).

O processo de qualificação junto à Agência Nacional de Aviação Civil (ANAC) é dividido em duas etapas distintas: avaliação objetiva e avaliação subjetiva (ANAC, 2020).

A avaliação objetiva é conduzida por meio de um conjunto de testes delineados no *Qualification Test Guide* (QTG). Este guia estabelece os critérios e procedimentos para avaliar o desempenho do simulador em diversos aspectos, como fidelidade de simulação, resposta a entradas do piloto, procedimentos de emergência e precisão dos sistemas. Essa etapa objetiva é essencial para garantir que o simulador atenda aos padrões técnicos exigidos pela ANAC (ANAC, 2020).

Por outro lado, a avaliação subjetiva é realizada por um avaliador e um piloto credenciado pela ANAC, que possua a habilitação válida para operar a aeronave em questão. Durante essa etapa, eles visitam o centro de treinamento e conduzem voos e testes práticos no simulador. Essa avaliação subjetiva permite uma análise mais qualitativa do desempenho do simulador, avaliando sua adequação em relação à experiência de voo real e sua capacidade de replicar com precisão as características da aeronave.

A combinação dessas duas etapas de avaliação, objetiva e subjetiva, é fundamental para garantir uma avaliação abrangente e completa do simulador. Através desse processo meticuloso, a ANAC pode determinar se o simulador atende aos requisitos regulatórios e está apto para fornecer treinamento de alta qualidade e seguro para os pilotos.

### 2.2.1 EMB-120 Aeronave

O Embraer EMB 120 Brasília representa um marco significativo na aviação regional, caracterizado por sua eficiência, velocidade e comodidades a bordo. Iniciando-se como um projeto baseado no conhecimento adquirido com o Bandeirante, a Embraer empreendeu o desenvolvimento do turboélice pressurizado EMB 120, visando atender às necessidades das companhias aéreas regionais (EMBRAER, 2024).

O processo de concepção dessa aeronave remonta a 1974, com seu nome inicial de Araguaia, posteriormente alterado para Brasília em 1979, coincidindo com o lançamento oficial do projeto. O EMB 120 foi projetado com uma capacidade para 30 passageiros e apresentava características como cauda e perfil de asa em T, semelhantes às do Xingu. Na FIGURA 6 é apresentada uma fotografia do EMB120-Brasília e na FIGURA 7 o modelo sucessor EMB 121 Xingu.

FIGURA 6 – Embraer 120 Brasília



FONTE: Embraer (2024).

FIGURA 7 – Embraer 121 Xingu



FONTE: Embraer (2024)

Uma das inovações mais notáveis do Brasília foi a utilização das turbinas PW115 de 1.500 HP, fabricadas pela Pratt & Whitney, em conjunto com hélices Hamilton Standard, permitindo que o bimotor atingisse velocidades superiores a 500 km/h. Além disso, o avião oferecia comodidades aos passageiros, como ar-condicionado durante o voo, e garantia de baixo ruído e vibração (EMBRAER, 2024).

O lançamento oficial do EMB 120, em 29 de julho de 1983, foi um evento de destaque internacional, com a presença da imprensa global e representantes de várias companhias aéreas e fornecedores. A atenção nacional e internacional gerada pelo lançamento do Brasília foi reconhecida com o prêmio "Marketing Destaque" da Associação Brasileira de Marketing (ABM), na área de Desenvolvimento Tecnológico (EMBRAER, 2024).

O sucesso do 120 foi evidenciado pela sua rápida entrada em produção no final de 1984, seguida pela certificação do Centro Técnico Aeronáutico (CTA) em maio de 1985.

Sua popularidade internacional foi instantânea, tornando-se a aeronave mais vendida em sua categoria antes mesmo da primeira entrega (EMBRAER, 2024).

A *Atlantic Southeast Airlines*, sediada em *Atlanta, Geórgia*, foi a primeira empresa a receber o Brasília, marcando seu primeiro voo comercial em setembro de 1985, ligando *Gainesville, na Flórida, a Atlanta*. O reconhecimento internacional do Brasília foi ainda mais fortalecido quando se tornou o primeiro avião brasileiro a ser certificado na Alemanha (FAB, 2024).

Em janeiro de 1988, o Brasília entrou em serviço no Brasil, pela Rio-Sul, e em 1994, já era considerado a aeronave regional mais utilizada no mundo, com 26 empresas de 14 países formando uma frota que acumulava mais de três milhões de horas de voo. Então, Brasília, deixou um legado duradouro na aviação regional, consolidando a posição da Embraer como líder na fabricação de aeronaves regionais de sucesso (EMBRAER, 2024).

O Embraer 120, desde seu lançamento oficial em 29 de julho de 1983, marcou uma era na aviação regional, sendo amplamente reconhecido por sua eficiência e desempenho notáveis. Com uma rápida entrada em produção no final de 1984 e certificação pelo CTA, FAA (*Federal Aviation Administration*) e CAA (*Civil Aviation Administration*) em 1985, o Brasília conquistou não apenas o mercado internacional, mas também uma posição de destaque na aviação regional brasileira (IBGE, 2016).

Ao longo dos anos, o EMB 120 Brasília passou por aprimoramentos significativos, culminando na versão *Advanced ER*, que oferecia maior conforto e conveniência aos passageiros, incluindo ar-condicionado tanto em voo quanto em solo. Além disso, os esforços contínuos da Embraer em otimizar os intervalos de manutenção resultaram em uma redução significativa nos custos operacionais, tornando-o ainda mais competitivo no mercado (EMBRAER, 2024).

A versatilidade do Brasília foi demonstrada por sua ampla gama de operadores, tanto no Brasil quanto internacionalmente, e sua capacidade de adaptação a diversas condições de operação, incluindo pistas curtas na região amazônica. Mesmo após décadas de serviço, muitas unidades continuam em operação, testemunhando a durabilidade e confiabilidade deste notável avião regional brasileiro (EMBRAER, 2024).

Assim, o Embraer EMB 120 Brasília não apenas deixou um legado duradouro na história da aviação regional, mas também solidificou a reputação da Embraer como uma das principais fabricantes de aeronaves do mundo, elevando o padrão de excelência na indústria aeroespacial brasileira (EMBRAER, 2024).

## 2.2.2 Superfícies da aeronave

Entender as superfícies de controle de uma aeronave (*Elevator*, *Rudder* e *Aileron*) é essencial para garantir não apenas a segurança, mas também o desempenho otimizado durante o voo. Esses componentes são muito mais do que simples peças mecânicas; são os elementos-chave que permitem aos pilotos manobrar e controlar a aeronave de forma eficaz. Neste contexto, explorar detalhadamente cada uma dessas superfícies torna-se fundamental para uma compreensão completa da operação e funcionamento de uma aeronave moderna.

### 2.2.2.1 *Elevator*

O *Elevator* é uma parte vital da estrutura de cauda de uma aeronave, conhecida como empenagem. Localizado no estabilizador horizontal, esse componente é responsável por controlar a altitude e o ângulo de ataque da aeronave (TOYWING, 2024).

Na FIGURA 8, em marrom, na cauda da aeronave, destaca-se o componente Elevator. Para auxiliar a visualização, o componente está circulado em vermelho.

FIGURA 8 – Detalhe do Elevator na Cauda da Aeronave



FONTE: Adaptado de ABEAR (2019).

O *elevator*, uma superfície aerodinâmica semelhante a uma asa horizontal localizada na cauda da aeronave, desempenha uma função vital no controle da altitude durante o voo. A manipulação do elevador pelo piloto resulta em alterações no ângulo de ataque da aeronave, exercendo um impacto direto na posição do nariz em relação ao

horizonte. Para efetuar essas modificações, o piloto utiliza o comando de *Pitch*, FIGURA 9 e FIGURA 10, que consiste na movimentação da coluna de controle dentro da cabine. Quando se busca aumentar a altitude da aeronave, o piloto realiza um movimento de *Pitch* puxando a coluna em direção ao peito. Por outro lado, para reduzir a altitude ou abaixar o nariz da aeronave, o movimento de *Pitch* é executado empurrando a coluna em relação ao corpo do piloto (TOYWING, 2024).

FIGURA 9 – Coluna do Simulador EMB120



FONTE: Os autores (2024).

FIGURA 10 – Coluna do Simulador EMB120 visão frontal



FONTE: Os autores (2024).

Essa manobra influencia diretamente a estabilidade da aeronave, permitindo que ela retorne ao seu estado original após perturbações, como rajadas de vento ou turbulência. Além disso, o ajuste do *elevator* afeta as forças de sustentação e arrasto na asa, impactando na velocidade e eficiência no consumo de combustível da aeronave (TOYWING, 2024).

#### 2.2.2.2 *Aileron*

Os *Ailerons* são encontrados nas extremidades das asas de uma aeronave, conforme FIGURA 11 em destaque marrom e circulos em vermelho. Estes são responsáveis por controlar o movimento de rolamento da aeronave, permitindo inclinar lateralmente a aeronave para a direita ou esquerda (EBAER, 2019).

FIGURA 11 – Detalhe do Airlon nas asas da aeronave



FONTE: Adaptado de ABEAR (2019).

Quando um *aileron* é abaixado, ocorre um aumento da sustentação na asa correspondente, levando-a a subir, enquanto a outra asa, com o *aileron* levantado, desce. Essa ação resulta em um movimento de rolagem da aeronave, conhecido como *Roll*. Para executar o movimento de rolagem para a esquerda, o piloto efetua um movimento de virar o manche para a esquerda, semelhante à operação de um volante em automóveis, exemplificado na FIGURA 12. Por outro lado, para realizar o movimento de rolagem para a direita, o piloto gira o manche para a direita. Esses comandos são essenciais para controlar a orientação lateral da aeronave durante o voo (TOYWING, 2024).

FIGURA 12 – Manche virado para esquerda



FONTE: Os autores (2024).

Esse movimento cria um diferencial de sustentação entre as asas, resultando em um movimento de rolamento ao longo do eixo longitudinal da aeronave. Durante uma curva para a direita, por exemplo, o aileron da asa esquerda é abaixado enquanto o da asa direita é levantado, promovendo o movimento desejado (TOYWING, 2024).

### 2.2.2.3 *Rudder*

Por fim, o *Rudder*, também conhecido como leme de direção, está localizado no estabilizador vertical da aeronave, conforme FIGURA 13. Ele controla o movimento de guinada, conhecido como *yaw*.

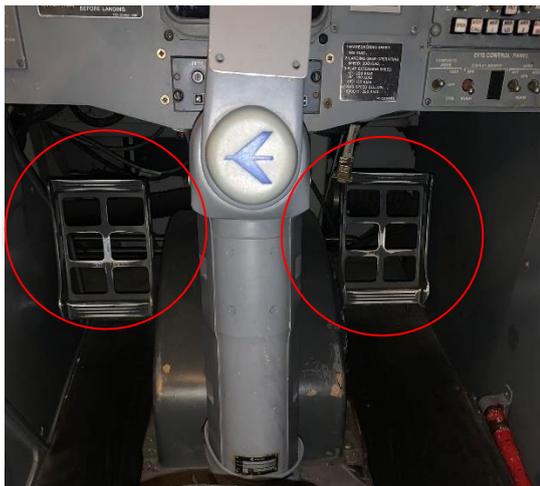
FIGURA 13 – Detalhe do Rudder na cauda da aeronave



FONTE: Adaptado de ABEAR (2019).

Durante o voo, o *rudder* atua no eixo vertical, induzindo o movimento de guinada ao alterar a distribuição de sustentação entre as asas. Essa ação é crucial para manter a aeronave direccionalmente estável e coordenada, especialmente durante curvas. O *rudder* é acionado pelo piloto através dos pedais de controle, permitindo corrigir movimentos indesejados do nariz da aeronave e evitar situações perigosas. Na FIGURA 14 é possível ver com detalhes os pedais que atuam no componente *rudder*.

FIGURA 14 – Detalhe dos Pedais dentro da Cabine do Simulador EMB-120



FONTE: Os autores (2024).

Após a exposição das superfícies relevantes para a condução do presente trabalho, a TABELA 1 compila os termos primordiais empregados na operação de aeronaves.

TABELA 1 – TERMOS PRIMORDIAIS PARA OPERAÇÃO DE AERONAVES

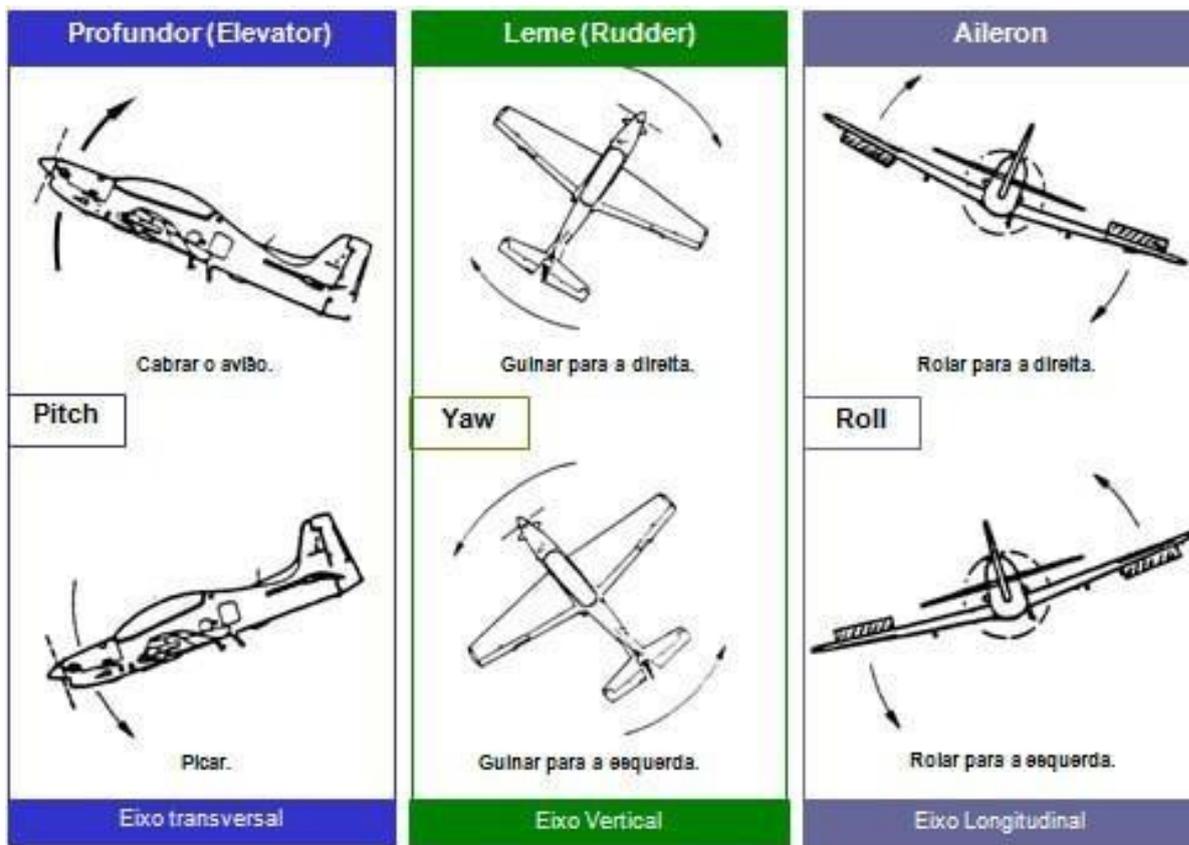
<b>Componente da Aeronave</b>	<i>Elevator</i>	<i>Airelon</i>	<i>Rudder</i>
<b>Manobras na Aeronave</b>	<i>Pitch/Arfagem</i>	<i>Roll/Rolagem</i>	<i>Yaw/Guinada</i>
<b>Componente da Cabine</b>	Coluna	Manche	Pedal
<b>Eixo de Movimentação</b>	Transversal	Longitudinal	Vertical

FONTE: Os autores (2024).

Portanto, esses componentes, juntamente com outros sistemas de controle, desempenham papéis essenciais na operação segura e eficiente de uma aeronave. Compreender como cada um funciona e como são utilizados pelo piloto é fundamental para uma pilotagem responsável e uma experiência de voo tranquila e segura para todos a bordo.

A FIGURA 15 reúne um breve resumo dos componentes da aeronave, nomes das manobras e os componentes que executam esses movimentos.

FIGURA 15 – Resumo das superfícies de uma aeronave



FONTE: TOYWING (2024).

### 2.3 QUALIFICATION TEST GUIDE (QTG)

O *Qualification Test Guide (QTG)*, ou Guia de Testes de Qualificação, desempenha um papel fundamental no processo de avaliação objetiva do simulador de voo (FSTD - *Flight Simulation Training Device*). Esse guia é uma ferramenta essencial que permite avaliar, do ponto de vista da engenharia, se o comportamento do FSTD é comparável ao da aeronave real (GOV, 2021).

Cada simulador de voo, com exceção do *FTD 4* (pelo nível de qualificação do simulador, é previsto que para a sua certificação não são necessários os testes objetivos) devem possuir um QTG específico, que contém uma variedade de informações essenciais. Isso inclui os resultados de testes realizados, declarações de conformidade e capacidade, a configuração da aeronave simulada e outros dados relevantes (ANAC, 2020).

O QTG possibilita a comparação direta do desempenho do FSTD com o desempenho da aeronave real. Ele se baseia em dados objetivos, que podem incluir dados de ensaios em voo da aeronave real, informações constantes de documentos como o *Aircraft Flight Manual (AFM)*, dados fornecidos pelos fabricantes dos sistemas do FSTD (como visual, motion e som), entre outros recursos disponíveis (GOV, 2021).

Essa abordagem objetiva, apoiada pelo QTG, permite uma avaliação precisa e detalhada do simulador de voo, garantindo que ele atenda aos padrões de desempenho e qualidade estabelecidos pelas autoridades de aviação civil (GOV, 2021).

Com base na sua classificação, o EMB 120 é dotado de 62 testes de QTG, sendo todos esses testes necessários para que o simulador seja aprovado pela ANAC. Entre os 62 testes, classificam-se nas seguintes categorias: Testes de força em solo, testes de parâmetros de decolagem e pouso, testes de desempenho em voo, forças de comando em voo, testes de *motion*, visual e latência (ANAC, 2020).

### 2.3.1 TESTE DE LATÊNCIA

O teste de latência em um *Full Flight Simulator (FFS)* tem como principal objetivo mensurar o tempo de resposta entre os diferentes subsistemas do equipamento. A partir de um *input* físico aplicado em uma das três superfícies da aeronave (*elevator*, *aileron* ou *rudder*), espera-se que os subsistemas, incluindo *motion*, *control loading*, *visual* e sistema de I/O, respondam dentro de um intervalo de tempo máximo de 300 ms (milissegundos), conforme estabelecido na regulamentação RBAC 60 (ANAC, 2020).

Para ilustrar de maneira mais detalhada, o teste de latência no simulador EMB-120 apresenta a seguinte dinâmica: inicialmente, o *input* de movimento realizado na superfície de teste é transmitido ao *Host*, o computador principal responsável pela gestão do sistema. Ao detectar o *input*, o *Host* envia sinais para os subsistemas I/O, *control loading*, *motion* e *visual*. O subsistema I/O, por sua vez, detecta o sinal enviado pelo *Host* e aciona um instrumento na cabine, conhecido como HSDI. O subsistema de *motion* interpreta o sinal de entrada e executa um movimento correspondente em seu sistema hidráulico, reproduzindo o movimento da superfície testada. Por fim, o subsistema *visual* recebe o *input* e gera um movimento na tela, representado por um ponto vermelho, indicando que o sistema visual recebeu e interpretou o input, realizando o movimento necessário.

Após a aquisição dos sinais por parte dos subsistemas, os dados são enviados para o *Chart Holder* (FIGURA 16), um dispositivo semelhante a um osciloscópio analógico com 8 entradas e 8 saídas, que registra as respostas em papel térmico. Esse registro permite uma análise posterior dos tempos de resposta de cada subsistema durante o teste de latência.

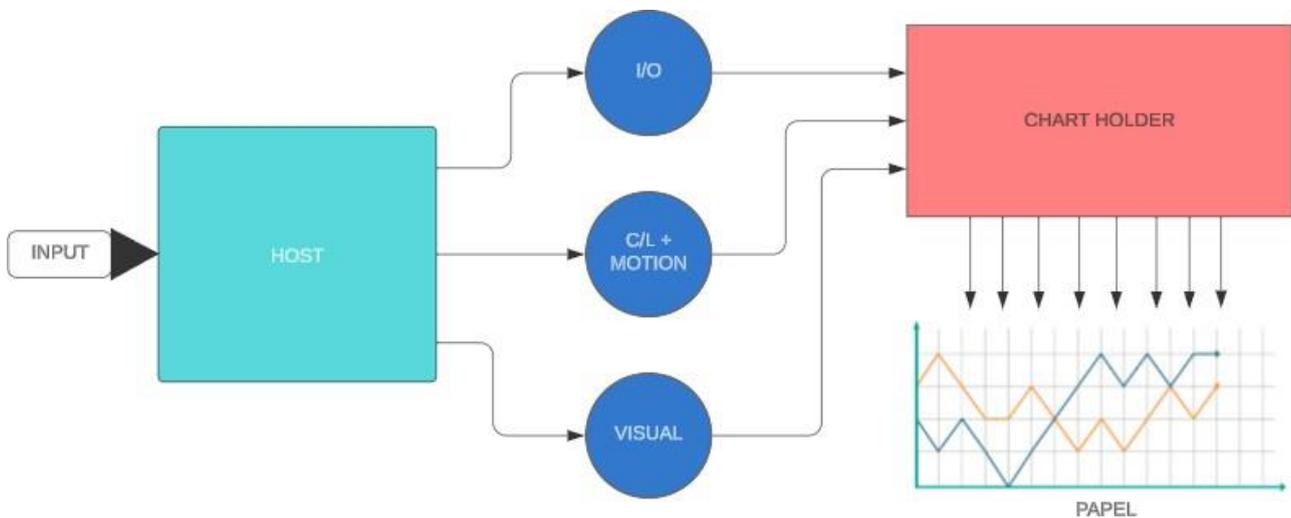
FIGURA 16 – Chart Holder



FONTE: Os autores (2024).

Em suma, essa descrição oferece um panorama do funcionamento do teste de latência no simulador EMB-120, sendo exemplificada pela FIGURA 17 para facilitar o entendimento do processo.

FIGURA 17 – Panorama do Funcionamento de Testes no Simulador EMB-120



FONTE: Os autores (2024).

### 2.3.2 Subsistemas do Simulador

Os subsistemas em um simulador de voo desempenham papéis fundamentais na replicação precisa da experiência de voo e na interação entre o piloto e o ambiente simulado. Cada subsistema é responsável por uma função específica, contribuindo para a

integridade e realismo do treinamento de voo. Abaixo, são descritos alguns dos principais subsistemas encontrados em simuladores de voo:

- *Motion*: O subsistema de *motion* é encarregado de simular os movimentos físicos da aeronave durante o voo. Utilizando sistemas hidráulicos avançados, o *motion* replica com precisão as sensações de aceleração, inclinação, turbulência e outros movimentos que um piloto experimentaria durante o voo real.
- *Visual*: O subsistema *visual* é responsável por criar o ambiente visual ao redor da aeronave simulada. Isso inclui a geração de paisagens, aeroportos, pistas, tráfego aéreo e outras características visuais do ambiente de voo. Sistemas avançados de gráficos e projeção são utilizados para garantir uma representação precisa e realista do ambiente virtual.
- *Control Loading*: O subsistema de *control loading* reproduz as forças físicas que um piloto sentiria ao manipular os controles de voo da aeronave. Essas forças, conhecidas como feedback tátil, ajudam a proporcionar uma sensação realista de controle e resposta da aeronave durante o treinamento.
- *I/O*: O subsistema de *I/O* gerencia todas as entradas e saídas de sinais analógicos do simulador. Ele coordena a comunicação entre o simulador e dispositivos externos, como instrumentos da cabine, controles de voo e sistemas de *feedback*.

Esses subsistemas trabalham em conjunto para criar uma experiência de treinamento de voo altamente imersiva e realista, fornecendo aos pilotos uma plataforma eficaz para desenvolver habilidades, praticar procedimentos e lidar com uma variedade de cenários operacionais.

### 2.3.2.1 Subsistemas – I/O

No contexto geral do simulador, o subsistema de *I/O* (*input/output*) desempenha um papel crucial na gestão de todas as entradas e saídas de sinais analógicos do sistema. Composto por placas conhecidas como *RTP* (*Real Time Products*), esse subsistema é responsável por coordenar e controlar as interações entre o simulador e seus dispositivos externos (RTP, 2024).

No que diz respeito ao teste de latência, o sistema de *I/O* assume a responsabilidade de realizar o movimento no instrumento *EHSI* (*Electronic Horizontal Situation Indicator*) na cabine (FIGURA 18). Esse movimento é realizado para indicar no

instrumento que a ação correspondente ao teste de latência, como *pitch*, *roll* ou *yaw*, foi executada com sucesso. O sistema de I/O atua como uma ponte entre o *input* fornecido durante o teste e a resposta gerada no instrumento, garantindo uma comunicação eficaz entre os subsistemas do simulador e os dispositivos de saída utilizados para *feedback* visual ou tátil.

FIGURA 18 – EHSI (Electronic Horizontal Situation Indicator)



FONTE: Os autores (2024).

### 2.3.2.2 Subsistemas – *Control Loading*

O subsistema de *Control Loading* desempenha um papel essencial na reprodução dos movimentos das superfícies de controle da aeronave, como *pitch*, *roll* e *yaw*, além de controlar o sistema de direção do simulador. Este sistema é responsável por gerenciar atuadores hidráulicos através de um computador central denominado *Fokker* (FIGURA 19).

FIGURA 19 – Computador central Fokker



FONTE: Os autores (2024).

Durante a execução dos testes de latência, o sistema de *Control Loading* é responsável por adquirir os sinais relevantes por meio de placas conhecidas como *FLM* (*Flight Loading Modules*), apresentadas na FIGURA 20, localizadas dentro do sistema *Fokker*. Essas placas desempenham um papel crucial na transmissão de informações sobre os movimentos das superfícies de controle para o simulador, garantindo uma resposta precisa e sincronizada às entradas do piloto ou aos comandos de teste.

FIGURA 20 – Placa de aquisição do sinal de C/L (FLM)



FONTE: Os autores (2024).

Essencialmente, o subsistema de *Control Loading* assegura que os movimentos das superfícies de controle sejam reproduzidos de forma autêntica e realista durante o treinamento de voo simulado. Isso proporciona aos pilotos uma experiência imersiva e prática, permitindo-lhes desenvolver habilidades de pilotagem e familiarizar-se com o comportamento da aeronave em diversas situações operacionais.

### 2.3.2.3 Subsistemas – *Motion*

O subsistema de *Motion* desempenha um papel crucial ao proporcionar a sensação de movimento e realismo ao simulador de voo. Composto por seis cilindros hidráulicos, este sistema é responsável por simular os movimentos da aeronave durante o treinamento, fornecendo aos pilotos uma experiência imersiva e autêntica.

Controlado eletronicamente pelo computador denominado *MEC (Motion Electrical Controller)*, o sistema de *Motion* opera em conjunto com uma unidade de potência hidráulica (*HPU*), que fornece pressão aos motores responsáveis pelo movimento dos cilindros. Estes motores podem atingir uma pressão de até 2000 PSI, garantindo um funcionamento eficiente e preciso do sistema. A FIGURA 21 e FIGURA 22 apresentam o computador *MEC* e o *HPU*, respectivamente.

FIGURA 21 – MEC (Motion Electronic Controller)



FONTE: Os autores (2024).

FIGURA 22 – HPU (Hydraulic Power Unit)



FONTE: Os autores (2024).

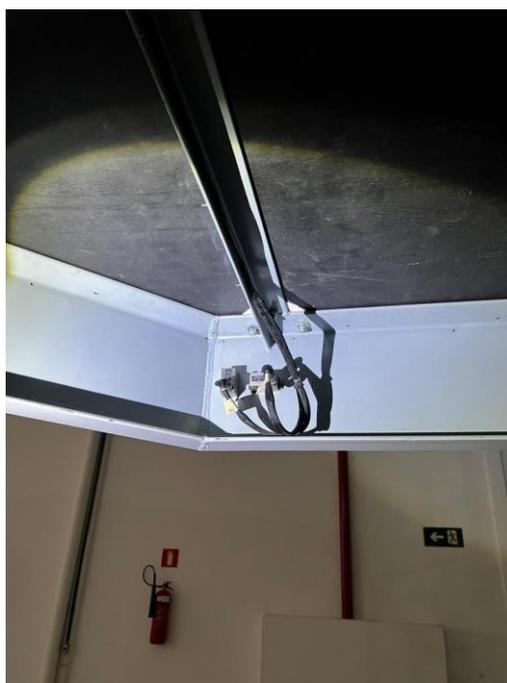
No contexto dos testes de latência, a aquisição de sinais do sistema de *Motion* é realizada através de dois acelerômetros (FIGURA 23 E FIGURA 24) posicionados no eixo X e Y, respectivamente. A combinação desses dois acelerômetros forma o eixo Z, fornecendo referências elétricas essenciais para a avaliação da latência do sistema. Esses acelerômetros são estrategicamente instalados na parte frontal do simulador, permitindo uma análise precisa e confiável do desempenho do subsistema de *Motion* durante os testes.

FIGURA 23 – Acelerômetros eixo X e Y



FONTE: Os autores (2024).

FIGURA 24 – Acelerômetros (vista geral)



FONTE: Os autores (2024).

#### 2.3.2.4 Subsistemas – *Visual*

O subsistema de *Visual* desempenha um papel fundamental ao proporcionar uma experiência imersiva e realista aos pilotos durante o treinamento. Responsável por simular a paisagem e as características externas do cenário, o sistema de *Visual* contribui significativamente para a sensação de estar operando uma aeronave real.

No caso específico do simulador EMB-120, devido às suas características tecnológicas e nível de qualificação B, o sistema de *Visual* possui algumas limitações em comparação com outros simuladores mais avançados. Em vez de oferecer uma gama completa de recursos visuais, o EMB-120 conta principalmente com uma visão noturna, utilizando pontos em diferentes formatos para criar a imagem. Esse processo é conduzido pelo computador *Vital IV* (FIGURA 25), desenvolvido pela *McDonnell Douglas* na década

de 80 e posteriormente adquirido pela *FlightSafety*, uma das principais empresas do setor de simuladores de voo.

FIGURA 25 – Vital IV



FONTE: Os autores (2024).

Com apenas dois projetores para a formação da imagem, o sistema de Visual do EMB-120 adapta-se às necessidades de treinamento e qualificação do simulador. Durante os testes de latência, o *Vital IV* gera um ponto vermelho no canto inferior central da projeção, que se move conforme o teste é executado. Esse sinal é eletronicamente gerado pelo próprio *Vital IV* e processado pelo computador Host, permitindo a aquisição precisa do sinal de Visual para a avaliação da latência do sistema. A FIGURA 26 apresenta a placa de aquisição do visual.

FIGURA 26 – Placa de aquisição do sinal do visual



FONTE: Os autores (2024).

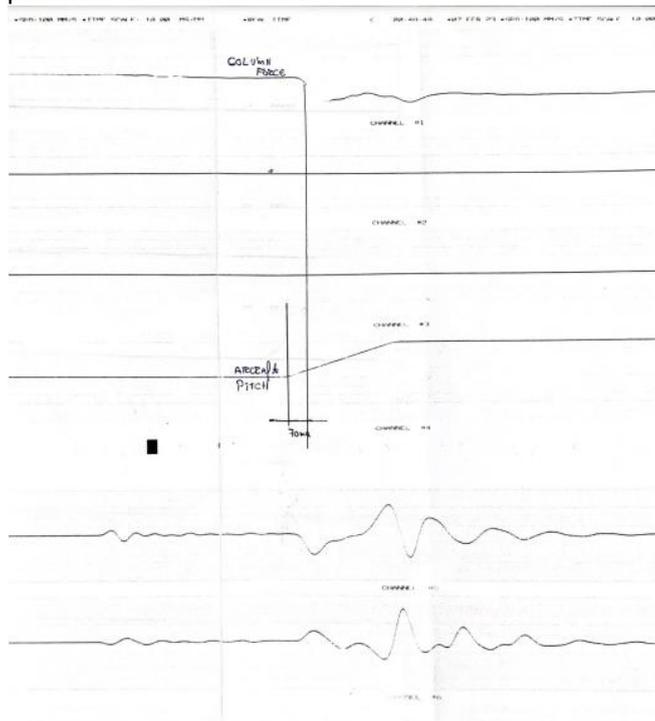
### 2.3.3 Comparativo entre Testes de Latência – EMB120 x Simulador mais atual

Ao compararmos o teste de latência do simulador EMB-120 com o teste realizado no simulador ATR42/72, observamos uma diferença significativa no tempo máximo de resposta permitido. Enquanto no EMB-120 esse tempo é de 300 ms desde o input inicial do movimento até a resposta dos outros subsistemas, no ATR42/72 esse tempo é reduzido para 150 ms. Essa discrepância pode ser atribuída ao nível de qualificação de cada simulador.

A relação entre o nível de qualificação e a precisão do simulador é crucial. Quanto maior o nível de qualificação, maior a exigência de precisão no funcionamento do equipamento. Isso implica em tempos de resposta mais curtos para garantir uma simulação mais realista e eficaz.

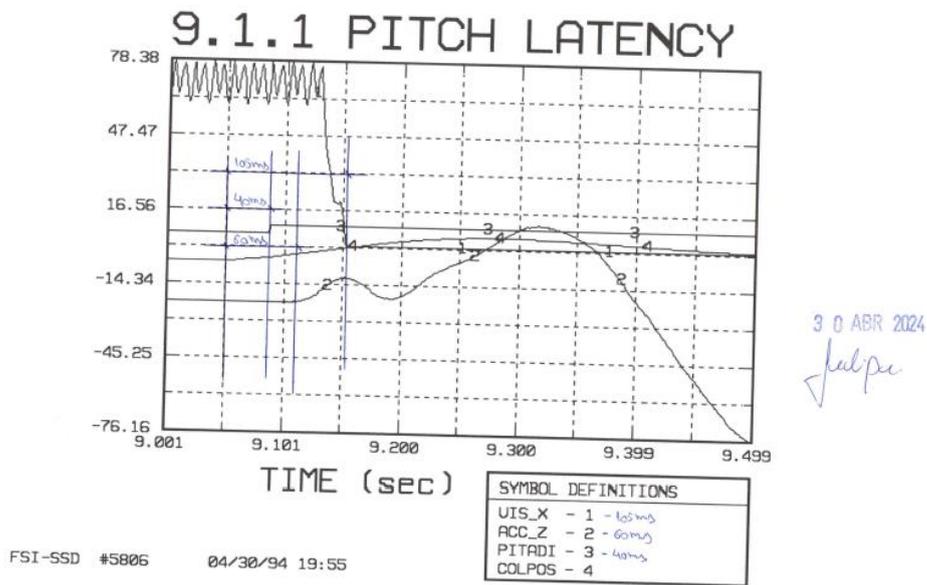
No caso específico do EMB-120, o teste de latência é realizado de forma manual, exigindo a intervenção humana para a execução do movimento inicial e a coleta dos resultados. Esse processo é realizado em um equipamento relativamente antigo, que ocupa muito espaço e utiliza grande quantidade de papel térmico para impressão dos resultados. Em contraste, o teste de latência no simulador ATR42/72 é automatizado, eliminando a variável humana e gerando os resultados em formato PDF, facilitando sua análise e interpretação. O comparativo é evidenciado pelas FIGURAS 27 e 28.

FIGURA 27 – Resultados dos testes de latência apresentados por meio do Chart Holder



FONTE: Os autores (2024).

FIGURA 28 – Teste de latência apresentado por meio do simulador ATR 42-500 (YAW Latency)



FONTE: Os autores (2024).

### 3 MATERIAIS E MÉTODOS

#### 3.1 CENÁRIO ATUAL

No atual cenário dos testes de latência, a execução segue um procedimento meticuloso e coordenado. Inicialmente, são necessários três técnicos para realizar o teste: um posicionado no *IOS* do simulador, outro na posição do piloto para executar o movimento inicial e o terceiro dentro do CPD para iniciar o *chart holder*. Cada técnico é responsável por uma etapa específica do processo.

O procedimento inicia com a posição de cada técnico em suas respectivas áreas designadas. Em seguida, o teste é carregado no *IOS* do simulador de acordo com as instruções detalhadas no manual de testes fornecido pela fabricante. Após o correto carregamento do teste, os três técnicos devem sincronizar suas ações. O técnico no *IOS* dá o comando para iniciar o teste e simultaneamente comunica o técnico no CPD para acionar o *chart holder*.

Com o início do teste, o técnico na posição do piloto executa o input de movimento na superfície correspondente conforme indicado no teste. Uma vez finalizada a execução do teste, o técnico no *IOS* notifica o técnico no CPD para interromper a impressão no *chart holder*.

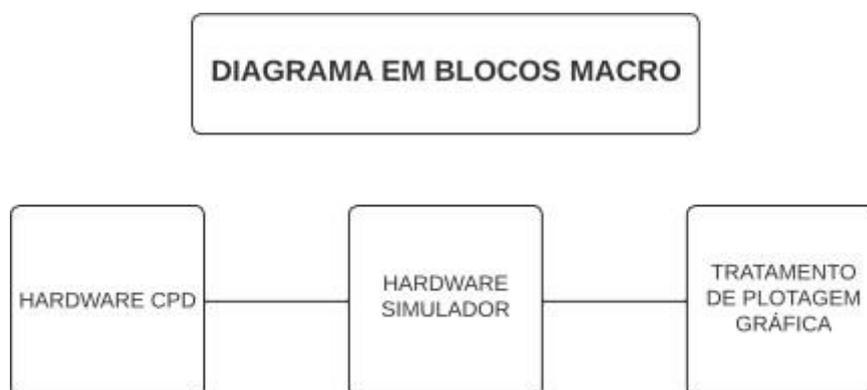
Após a conclusão do teste e a retirada adequada da impressão, é realizada uma análise minuciosa dos resultados. Os valores obtidos são comparados com as tolerâncias estabelecidas pela ANAC para garantir a conformidade do simulador com os padrões de segurança e desempenho exigidos pela regulamentação. Este processo garante que o simulador atenda aos requisitos de qualificação necessários para o treinamento de pilotos e a certificação da ANAC.

#### 3.2 SOLUÇÃO PROPOSTA

Após compreendido o cenário atual, é notável a necessidade de melhorias em diversos aspectos, como por exemplo, a urgência de potencializar a hora trabalhada dos técnicos envolvidos na atividade, a demanda técnica na aquisição desses sinais, a carência por um sistema mais sustentável, além da relevância em tornar o sistema um pouco mais atual às tecnologias disponíveis em conectividade e instrumentação eletrônica. Será garantido a eficiência técnica do processo, mantendo a veracidade dos sinais manipulados, em prol ao objetivo principal dos testes, que é a validação por meio dos reais sinais obtidos pela resposta do simulador aos *inputs* gerados nas superfícies aéreas.

Para um melhor entendimento da solução proposta, na FIGURA 29 é apresentado um diagrama macro dos três grandes blocos a serem implementados como solução ao problema proposto neste TCC.

FIGURA 29 – Diagrama em Blocos Macro da Solução Proposta



FONTE: Os autores (2024).

A solução proposta contempla três grandes blocos principais como pilares, sendo o bloco de *Hardware CPD*, o *Hardware Simulador* e o Tratamento de Plotagem Gráfica.

O bloco *Hardware CPD* será uma placa prototipada com a finalidade de adquirir os sinais que no cenário atual são processados pelo *Chart Holder*, realizando a conversão AD destes sinais. O *Chart Holder* não possui um início definido para a aquisição dos sinais do simulador e inicializa a plotagem dos sinais a partir do momento em que é energizado. Por este motivo, o *Hardware CPD* também será capaz de se comunicar com o *Hardware Simulador* por cabos, recebendo um comando de início ou de término para gravação dos sinais do simulador. Além da otimização dos sinais gravados, afinal será gravado apenas o que é de interesse para a avaliação dos testes, não será mais necessário um terceiro indivíduo no CPD para iniciar a aquisição dos sinais do simulador. Portanto, este bloco será responsável por interpretar um comando de início e tipo de teste para gravação dos sinais vindo do *Hardware Simulador*, realizar a conversão AD dos mesmos, armazená-los em um cartão SD, para futuramente serem acessados, e por fim receber um comando para concluir a gravação. Contará também com outra funcionalidade intrínseca como um *LED* sinalizando a energização da placa e *LEDs* indicadores de *status* do sistema, para auxiliar no desenvolvimento.

O bloco *Hardware Simulador*, como citado anteriormente, terá comunicação por cabos com o bloco *Hardware CPD*. Este sinal será enviado a partir de um *input* dado por

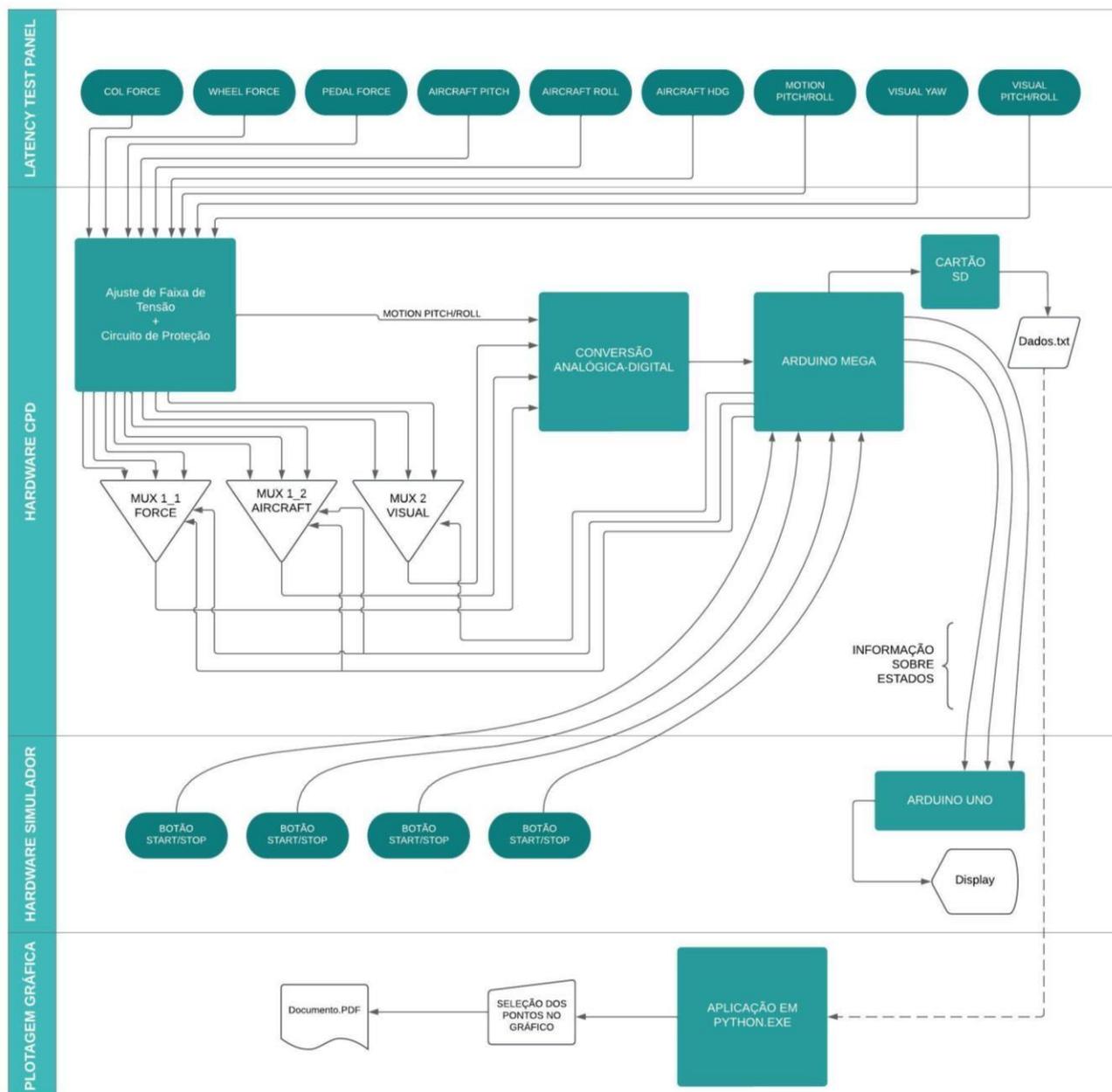
um profissional da cabine do simulador. Para que isso seja possível, o *Hardware* Simulador contará com uma interface com o usuário por meio de *display* e botões para configuração. O operador desta placa poderá selecionar o tipo de teste realizado e iniciar ou parar a gravação dos sinais.

O bloco de Tratamento e Plotagem Gráfica será responsável em gerar os gráficos a partir dos dados salvos no cartão SD pelo bloco *Hardware* CPD. A ideia consiste em gerar um arquivo executável que capture os dados salvos no cartão SD, gere os gráficos para cada tipo de teste em .PDF, fornecendo os tempos de latência entre os sinais conforme especificado no QTG, prontos para serem analisados pelo técnico responsável.

Em suma, o objetivo principal é realizar uma comunicação entre o *Hardware* CPD e o *Hardware* Simulador, possibilitando a troca de informações de forma eficiente e automatizada. Com essa solução, espera-se otimizar o processo de aquisição e tratamento dos sinais, eliminando a necessidade de intervenção manual para iniciar ou parar as gravações e garantindo que apenas os dados relevantes sejam coletados. Além disso, o bloco de Tratamento e Plotagem Gráfica proporcionará uma análise visual rápida e precisa dos resultados, facilitando a interpretação dos dados pelo técnico responsável. Com esses aprimoramentos, a eficiência e a precisão dos testes serão significativamente aumentadas, melhorando o desempenho geral do sistema tornando-o mais tecnológico e sustentável.

Para uma melhor compreensão da solução proposta, na FIGURA 30 é apresentado um diagrama em blocos detalhado da solução proposta para este TCC, a fim de atender a demanda existente na EPA.

FIGURA 30 – Diagrama em Blocos Detalhado da Solução Proposta



FONTE: Os autores (2024).

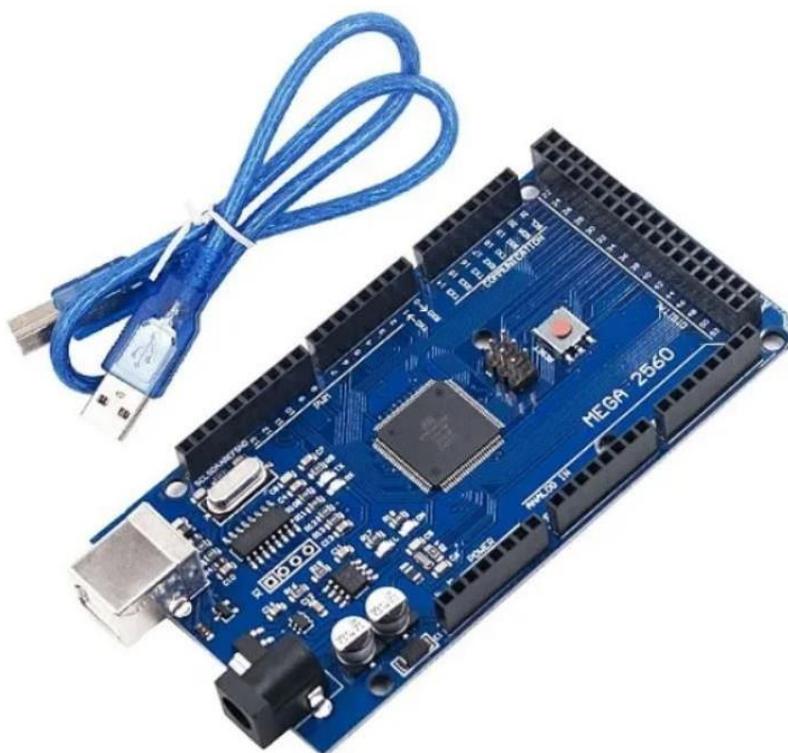
### 3.3 HARDWARE CPD

O bloco *Hardware CPD* é um componente relevante do projeto, concebido para substituir o *Chart Holder* na aquisição e conversão dos sinais provenientes do simulador. Este *Hardware* será responsável pela conversão analógica-digital (AD) dos sinais, comunicação *wireless* com o *Hardware Simulador*, armazenamento dos dados adquiridos e diversas outras funcionalidades que otimizam o processo de aquisição de dados. O desenvolvimento do *Hardware CPD* será estruturado em quatro etapas principais, detalhadas a seguir.

### 3.3.1 Unidade microprocessada e Comunicação com Hardware Simulador

Para a unidade microprocessada e a comunicação com o hardware do simulador, foi selecionada a placa Arduino Mega 2560 (FIGURA 31). Esta escolha foi motivada pela necessidade de integrar as funcionalidades de controle e comunicação em um único módulo compacto e eficiente, garantindo uma operação mais simplificada e coesa para o *Hardware CPD*.

FIGURA 31 – Arduino Mega 2560 CH340 com Cabo USB



FONTE: A2ROBOTICS (2024).

A placa Arduino Mega 2560 é amplamente reconhecida por sua versatilidade e alta capacidade de processamento, sendo uma escolha ideal para projetos que demandam maior número de conexões e processamento de dados mais robusto. Equipada com o microcontrolador ATmega2560, a placa oferece 54 pinos digitais (sendo 15 configuráveis como saídas PWM), 16 entradas analógicas e 4 portas seriais de hardware, permitindo um nível elevado de controle e comunicação para desenvolvimento com cartão SD e conversores analógicos-digitais trabalhando paralelamente (STORE ARDUINO, 2024).

Com 256 KB de memória flash para armazenamento de código, 8 KB de SRAM e 4 KB de EEPROM, o Arduino Mega 2560 possibilita o desenvolvimento do trabalho com

grande volume de dados, sendo especialmente útil para realização de coleta e gravação de informações simultaneamente (STORE ARDUINO, 2024).

O microcontrolador ATmega2560 opera em uma frequência de 16 MHz, garantindo um desempenho adequado para sistemas de aquisição de dados em tempo real. A placa é compatível com uma ampla variedade de bibliotecas e *shields*, mantendo a flexibilidade no desenvolvimento e integração de funcionalidades adicionais no trabalho (STORE ARDUINO, 2024).

O Arduino Mega 2560 é programado via USB, utilizando o chip ATmega16U2 para comunicação serial, e suporta tensões de operação de 5V, simplificando sua interface com outros dispositivos eletrônicos. A robustez do hardware, aliada à extensa documentação e suporte fornecidos pela comunidade Arduino, faz com que esta placa seja uma escolha confiável e eficiente para o desenvolvimento deste trabalho (STORE ARDUINO, 2024).

A unidade microprocessada, baseada no ATmega328P, apresenta características técnicas robustas para o controle e processamento dos sinais convertidos pelo Hardware CPD. Com um *clock* de 16 MHz e memória flash de 32 kB, o ATmega328P é capaz de executar tarefas de controle em tempo real, armazenar o firmware necessário e manipular dados temporários e persistentes de forma eficiente. A familiaridade com o ambiente de desenvolvimento Arduino IDE, conhecido por sua simplicidade e facilidade de uso, permite um desenvolvimento rápido e depuração eficaz do código, utilizando bibliotecas extensas e suporte comunitário (STORE ARDUINO, 2024).

Em resumo, a Arduino Mega 2560 é uma solução robusta, flexível e econômica para a unidade microprocessada e comunicação do Hardware CPD. Ela assegura um desempenho adequado para o processamento de sinais e comunicação eficiente com o hardware do simulador, facilitando a implementação e operação do sistema de maneira coesa e integrada (STORE ARDUINO, 2024).

### 3.3.2 Conversão AD dos sinais

A conversão AD é a primeira e crucial etapa do *Hardware CPD*. Esta placa deve ser capaz de captar e converter os sinais analógicos do simulador em dados digitais para processamento posterior. Para isso, é necessário entender quais sinais serão aquisitados.

Dentro do CPD, no gabinete do simulador EMB-120, é aquisitado os sinais para os testes de latência pelo painel *LATENCY TEST PANEL* (FIGURA 32).

FIGURA 32 – LATENCY TEST PANEL



FONTE: Os autores (2024).

O sistema contempla três tipos diferentes de testes de latência:

- *Column Force Test – Elevator;*
- *Wheel Force Test – Aileron;*
- *Pedal Force Test – Rudder.*

Dos 15 sinais disponíveis no *LATENCY TEST PANEL*, seis são comuns aos três tipos de testes de latência. Esses sete sinais são:

- *AIRCRAFT HDG;*
- *AIRCRAFT ROLL;*
- *AIRCRAFT PITCH;*
- *MOTION ROLL / PITCH;*
- *VISUAL ROLL / PITCH;*
- *VISUAL YAW.*

Além disso, é necessário um sinal de força específico para cada tipo de teste:

- *COL FORCE* para o *Column Force Elevator Test;*
- *WHEEL FORCE* para o *Wheel Force Aileron Test;*
- *PEDAL FORCE* para o *Pedal Force Rudder Test.*

Portanto, dos 15 sinais disponíveis, 9 são utilizados nos três testes de latência: seis sinais comuns e três específicos para cada teste. Os seis sinais restantes não são utilizados. A TABELA 2 fornece uma visão geral da utilização dos sinais:

TABELA 2 – SINAIS TESTE DE LATÊNCIA POR SUPERFÍCIE

COLUMN FORCE TEST ELEVATOR	WHEEL FORCE TEST AILERON	RUDDER PEDAL FORCE TEST RUDDER
COL FORCE	WHELL FORCE	PEDAL FORCE
AIRCRAFT HDG	AIRCRAFT HDG	AIRCRAFT HDG
AIRCRAFT ROLL	AIRCRAFT ROLL	AIRCRAFT ROLL
AIRCRAFT PITCH	AIRCRAFT PITCH	AIRCRAFT PITCH
MOTION ROLL / PITCH	MOTION ROLL / PITCH	MOTION ROLL / PITCH
MOTION YAW	MOTION YAW	MOTION YAW
VISUAL ROLL / PITCH	VISUAL ROLL / PITCH	VISUAL ROLL / PITCH
VISUAL YAW	VISUAL YAW	VISUAL YAW

FONTE: Os autores (2024).

Em verde estão os três sinais que variam dependente da superfície testada. Em comparativo com o *Latency Test Panel*, percebe-se que os sinais *FORCE STEP*, *COL POS*, *WHEEL POS*, *PEDAL POS*, *CHART RECORD REMOTE* e *MOTION YAW* não são utilizados para os testes de latência.

Analisando testes de latência executados no passado pelo *Chart Holder*, nota-se que é possível realizar o teste em um novo formato, visto que nem todos os sinais possuem variação após manobras realizadas dentro da cabine de voo, nem tão pouco tem efeito prático para a análise de latência. Portando, uma nova proposta de aquisição foi discutida dentro da empresa *EPA Training Center*, otimizando o processamento de sinais uma vez que esta mudança diminuirá o número de sinais lidos simultaneamente, respeitando as diretrizes da ANAC e conseqüentemente o QTG.

Nessa nova configuração, para cada tipo de teste, serão lidos apenas os sinais necessários para o cálculo de latência, distribuídos da seguinte forma:

- Teste de Latência para *Pitch* (*COLUMN FORCE TEST ELEVATOR*):
  - COL FORCE;
  - AIRCRAFT PITCH;
  - MOTION ROLL / PITCH;
  - VISUAL YAW.
  
- Teste de Latência para *Roll* (*WHEEL FORCE TEST AILERON*):
  - WHELL FORCE;
  - AIRCRAFT ROLL;
  - MOTION ROLL / PITCH;
  - VISUAL YAW.

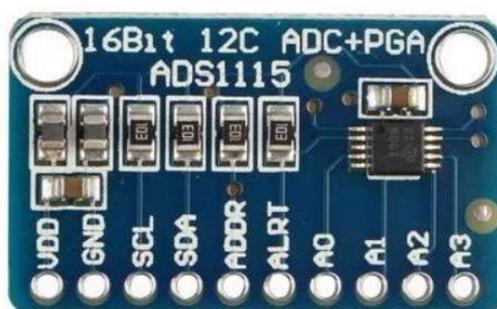
- Teste de Latência para *Yaw (RUDDER PEDAL FORCE TEST RUDDER)*:
  - PEDAL FORCE;
  - AIRCRAFT HDG;
  - MOTION ROLL / PITCH;
  - VISUAL ROLL / PITCH.

Assim, o sistema desenvolvido ficou otimizado, de tal forma que a unidade microprocessada pode executar o dobro de medidas quando comparadas com a aquisição de oito sinais para cada tipo de teste.

Após a definição dos sinais medidos, começou-se a análise final para determinar os componentes necessários para correta conversão analógica-digital dos sinais lidos pelo *Latency Test Panel*. Para a substituição do *Chart Holder* pela placa *Hardware CPD*, será necessário utilizar um conversor analógico digital que atende às demandas do projeto em resolução e taxa de aquisição. Os sinais aquisitados do simulador possuem uma *range* de tensão que varia de -5V à 5V e é necessário uma respostas de pelo menos 50 ms.

Foi utilizado o módulo conversor analógico digital ADS1115 (FIGURA 33), que possui quatro canais de entrada simples ou dois canais diferenciais. Como todos os sinais possuem referência ao sinal de *ground* do circuito, foram utilizados os quatro canais do módulo na configuração *singles-ended*. O projeto necessitou da conversão de quatro sinais. Para viabilizar isso, o conversor ADS1115 possui quatro endereços I<sup>2</sup>C configurados via hardware, possui resolução de 16 bits, 860 amostras/segundo, opera entre 0V e 5V e baixo consumo de corrente (TEXAS INSTRUMENTS, 2009).

FIGURA 33 – Placa do Conversor ADS1115



FONTE: ELETROGATE (2024).

A escolha desse conversor permitiu uma ótima resposta em resolução, porém foi necessário a implementação de um circuito capaz selecionar os sinais que serão lidos, a

partir do tipo de teste, além de converter o *range* de tensão de -5V a 5V para a faixa de 0V a 5V.

Para a seleção dos sinais foi utilizado o multiplexador duplo CD4052. Esse multiplexador é um circuito integrado analógico e digital 4x1, ou seja, a partir das entradas binários, uma das quatro entradas é selecionada para a saída. Analisando novamente a distribuição dos sinais por tipo de teste, percebeu-se que cada teste possui apenas um sinal de força (*COL FORCE*, *WHEEL FORCE* ou *PEDAL FORCE*), um sinal da aeronave (*AIRCRAFT PITCH*, *AIRCRAFT ROLL* ou *AIRCRAFT HDG*) e um sinal de visual (*VISUAL YAW* ou *VISUAL PITCH/ROLL*). E para todos os testes é avaliado o mesmo sinal de movimento, o *MOTION PITCH/ROLL*. Portanto, serão necessários dois circuitos integrados CD4052 (FIGURA 34).

FIGURA 34 – Multiplexador duplo CD4052



FONTE: MAKERHERO (2024).

Resolvido o problema de seleção dos sinais, ainda era necessário adicionar ao sistema um circuito capaz de ler os sinais na faixa de -5V a 5V do *Latency Test Panel* e convertê-los para 0V a 5V, possibilitando a leitura dos conversores ADS1115. A solução basicamente foi a inclusão de um circuito somador com ganho unitário, com amplificadores operacionais LM124 (FIGURA 35) e proteção para picos de então negativa e positiva, com Diodos Zener 5.1V (FIGURA 36), protegendo assim as entradas do multiplexador CD4052 e do conversor ADS1115 no *Hardware CPD*.

FIGURA 35 – Amplificador Operacional LM124



FONTE: MOUSER (2024).

FIGURA 36 – Diodo Zener 5.1V

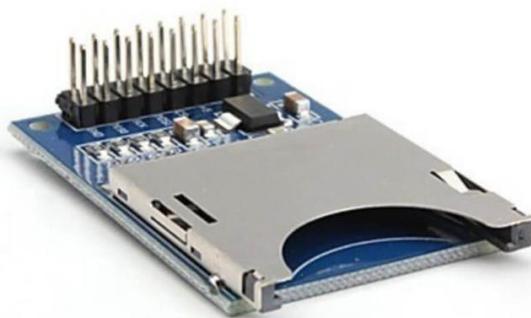


FONTE: MAKERHERO (2024).

### 3.3.3 Saída dos dados para plotagem gráfica (cartão SD)

Para a etapa de saída dos dados para plotagem gráfica, foi selecionado o módulo 4MD36 (FIGURA 37), para salvar os dados adquiridos pelo conversor AD no *Hardware* CPD. Esta escolha foi motivada por diversos fatores que tornam o módulo 4MD36 uma opção ideal para o armazenamento seguro e eficiente dos dados para posterior análise e plotagem gráfica. Esses fatores serão listados na sequência.

FIGURA 37 – Módulo 4MD36 Cartão SD



FONTE: INSTITUTO DIGITAL (2024).

O módulo 4MD36 é um módulo de cartão SD que permite a leitura e escrita de dados em cartões *microSD*. Suas principais especificações incluem compatibilidade com cartões *microSD* e *microSDHC*, suporte a sistemas de arquivos FAT16 e FAT32, e uma interface SPI que facilita a comunicação com microcontroladores. Essas características garantem a capacidade de armazenar grandes volumes de dados de forma eficiente e acessível (INSTITUTO DIGITAL, 2024).

Uma das principais vantagens do módulo 4MD36 é a facilidade de integração com o ambiente Arduino. A biblioteca SD do Arduino fornece uma interface simples para a leitura e escrita de dados no cartão SD, permitindo que os desenvolvedores rapidamente configurem e utilizem o módulo em seus projetos. Este suporte nativo no Arduino IDE simplifica o desenvolvimento e a implementação de funções de armazenamento de dados.

A capacidade de armazenar grandes volumes de dados é crucial para este trabalho, pois permite o registro contínuo e detalhado dos quatro sinais adquiridos pelo conversor AD, para cada tipo de teste. Esses dados podem posteriormente ser utilizados para análises detalhadas e plotagens gráficas, facilitando a interpretação e visualização dos resultados dos testes de latência.

Outro benefício significativo é a portabilidade e acessibilidade dos dados armazenados. Os cartões SD são amplamente utilizados e facilmente removíveis, permitindo que os dados sejam transferidos do *Hardware* CPD para um computador para análise e plotagem gráfica em linguagem *python*. Além disso, a durabilidade e confiabilidade dos cartões SD asseguram que os dados sejam armazenados de forma segura e estejam prontamente disponíveis quando necessário.

O custo-benefício do módulo 4MD36 também é um fator importante. Ele oferece uma solução econômica para o armazenamento de dados, tornando-o uma excelente escolha para projetos de prototipagem e desenvolvimento inicial. A relação custo-benefício é aprimorada pela facilidade de uso e integração com o ambiente Arduino, reduzindo o tempo e esforço necessários para a implementação.

A escolha do módulo 4MD36 para este trabalho é justificada por sua capacidade de atender aos requisitos técnicos de armazenamento de dados do sistema. Sua compatibilidade com o ambiente Arduino, suporte para grandes volumes de dados, portabilidade e excelente custo-benefício tornam o módulo 4MD36 uma escolha prática e eficiente.

Em resumo, o módulo 4MD36 oferece uma solução robusta, flexível e econômica para a saída de dados para plotagem gráfica, assegurando que os dados adquiridos pelos conversores AD no hardware CPD sejam armazenados de forma segura e eficiente para posterior análise e visualização.

### 3.4 HARDWARE SIMULADOR

O *Hardware* Simulador desempenha um papel importante no sistema desenvolvido, servindo como a interface com o aplicador de teste dentro da cabine de simulação, auxiliando no controle e monitoramento dos testes de latência das superfícies da aeronave. Este componente garante a precisão e a eficiência na coleta de dados, permitindo intervir na coleta de dados direto do simulador, definindo início, término e seleção de teste. A seguir, serão descritos os principais aspectos do *Hardware* Simulador, incluindo sua unidade microprocessada, interface com o aplicador de teste, comunicação com o *Hardware* CPD e periféricos.

#### 3.4.1 Unidade microprocessada

A unidade microprocessada utilizada no *Hardware* Simulador foi o microcontrolador Arduino UNO (FIGURA 38). Esta escolha padroniza o desenvolvimento e a manutenção do sistema, garantindo consistência e compatibilidade entre os componentes.

FIGURA 38 – Arduino Uno



FONTE: CASA DA ROBÓTICA (2024).

#### 3.4.2 Interface com aplicador de teste

A interface com o aplicador de teste é composta por um display 1.9 pol IPS Colorido (FIGURA 39), que fornece uma visualização clara e imediata do status do teste dentro da cabine de teste. Este display é utilizado para indicar qual superfície da aeronave está sendo testada (*PITCH*, *ROLL*, *YAW*) e o estado atual do processo de coleta de dados.

FIGURA 39 – Display 1.9 pol IPS Colorido



FONTE: LCDWIKI (2024).

Por exemplo, durante um teste de latência, o display pode exibir mensagens como as mencionadas abaixo no exemplo para o teste da superfície *PITCH*:

- "Teste PITCH: Aguardando Start"
- "Teste PITCH: Gravando Dados"
- "Teste PITCH: Dados Salvos"

A utilização desse display oferece diversas vantagens, incluindo facilidade de leitura, pois proporciona uma leitura rápida e clara das mensagens, mesmo em condições de baixa luminosidade. Este tipo de display é eficiente em termos de consumo de energia, o que é benéfico para a operação contínua dentro da cabine de teste, além de ser simples e confiável, pois os LCDs são componentes simples e robustos, conhecidos por sua durabilidade e baixa taxa de falhas, garantindo uma operação confiável ao longo do tempo.

### 3.4.3 Comunicação com *Hardware* CPD

A comunicação entre o *Hardware* Simulador e o *Hardware* CPD é realizada utilizando comunicação via cabo flat e basicamente trafegarão sinais elétricos que indicam todos os estados da agravação dos dados do *Hardware* CPD para visualização do piloto de dentro da cabine de simulação.

Os dados são enviados do *Hardware* Simulador para o *Hardware* CPD por meio botões de seleção, início e término dos testes.

### 3.4.4 Periféricos

Os periféricos do *Hardware* Simulador foram projetados para facilitar a operação e controle dos testes de latência. Estes incluem:

- **Botões Seletores de Teste:** Existem três botões dedicados para iniciar testes específicos de *PITCH*, *ROLL* e *YAW*. Cada botão, quando pressionado, configura o sistema para o tipo de teste correspondente e atualiza a interface do display LCD.
- **Botão de *Start/Stop* do Teste:** Um botão adicional é utilizado para sinalizar ao *Hardware* CPD o início e término do teste. Ao pressionar este botão, o usuário pode iniciar a coleta de dados ou interrompê-la, enviando comandos apropriados ao *Hardware* CPD para controlar o processo de gravação dos dados.

Estes periféricos garantem uma operação intuitiva e eficiente do *Hardware* Simulador, proporcionando ao usuário um controle direto e preciso sobre os testes de latência realizados nas superfícies da aeronave.

## 3.5 PLOTAGEM GRÁFICA

O tópico de Plotagem Gráfica trata do processamento e visualização dos dados adquiridos e armazenados pelo *Hardware* CPD. Esta seção detalha a utilização de linguagem de programação, bibliotecas, métodos de leitura dos dados do cartão SD, funcionamento do aplicativo executável (.EXE) e os resultados esperados.

### 3.5.1 Linguagem/bibliotecas

Para a realização deste trabalho, foi utilizada a linguagem de programação *Python* devido à sua simplicidade, flexibilidade e ampla gama de bibliotecas projetadas para manipulação, análise e visualização de dados. As principais bibliotecas que compõem o trabalho são:

- **Pandas:** Facilita a leitura e manipulação de dados, permitindo converter informações em formato de texto para estruturas organizadas e adequadas à análise.
- **Matplotlib:** Essencial para a criação de gráficos interativos e estáticos, sendo versátil na apresentação visual dos dados processados.
- **NumPy:** Utilizada para realizar operações matemáticas avançadas e manipular arrays de forma eficiente, complementando o uso do Pandas.

- **PyInstaller:** Responsável por converter o script Python em um executável (.exe), permitindo que o programa seja utilizado em qualquer computador sem necessidade de instalar o Python.
- **Tkinter:** Permite a criação de interfaces gráficas simples e intuitivas para facilitar a interação do usuário com a aplicação, como na seleção de arquivos e na configuração inicial de parâmetros.
- **PyPDF2:** Gera relatórios finais em formato PDF, incluindo gráficos e resultados obtidos durante a análise.
- **Datetime:** Fornece suporte para manipulação de datas e horários, registrando momentos importantes e organizando informações temporais no relatório.
- **Os:** Gerencia operações do sistema, como navegação em diretórios e salvamento de arquivos gerados.

Essas bibliotecas foram escolhidas por sua robustez, popularidade e amplo suporte comunitário, garantindo uma solução confiável, eficiente e de fácil manutenção para a análise e visualização gráfica dos dados.

### 3.5.2 Leitura do cartão SD

A leitura dos dados salvos no cartão SD será realizada pelo módulo 4MD36, que armazena os dados adquiridos pelos conversores AD. O processo de leitura dos dados no Python envolve os seguintes passos:

**Inserção do Cartão SD:** O cartão *microSD* contendo os dados será inserido em um leitor de cartões compatível com o computador.

**Leitura dos Arquivos:** Utilizando a biblioteca Pandas, o *script* Python é capaz de ler os arquivos de texto armazenados no cartão SD. Os arquivos serão carregados em *DataFrames* para facilitar a manipulação e análise dos dados.

**Processamento dos Dados:** Os dados serão processados para verificar sua integridade e formatados conforme necessário para a plotagem gráfica. Isso incluir a conversão de amplitude do sinal, definição de qual tipo de teste será analisado e a determinação de pontos de resposta do simulador para cálculo de latência.

### 3.5.3 Funcionamento (.EXE)

Para garantir que o aplicativo de plotagem gráfica possa ser executado em qualquer computador, o script Python será convertido em um arquivo executável (.exe) usando a biblioteca PyInstaller. Este processo envolve:

**Criação do Script Python:** O script Python deve ser completo e funcional, incluindo todas as etapas de leitura, processamento e plotagem dos dados.

**Configuração do PyInstaller:** O PyInstaller será configurado para incluir todas as dependências necessárias e gerar o arquivo executável.

**Distribuição do Executável:** O arquivo .exe gerado pode ser distribuído e executado em qualquer computador com o sistema operacional compatível, sem a necessidade de uma instalação prévia do Python ou das bibliotecas utilizadas.

### 3.5.4 Resultados

Os resultados proporcionados pela aplicação de plotagem gráfica incluem visualizações claras e detalhadas dos dados adquiridos pelo Hardware CPD, facilitando a análise e interpretação dos testes realizados. Utilizando Python e suas bibliotecas especializadas, a aplicação oferece:

**Gráficos Interativos:** Exibição dinâmica dos sinais adquiridos, permitindo ao usuário selecionar pontos de interesse diretamente no gráfico, com recursos de zoom para maior precisão na análise.

**Análise de Latência:** Cálculo automatizado da diferença de tempo entre o sinal de referência e os demais canais, destacando se os resultados estão dentro do limite de tolerância estabelecido.

**Comparação Visual:** Exibição de múltiplos gráficos lado a lado, possibilitando a comparação eficiente dos resultados obtidos em diferentes testes de latência.

**Geração de Relatórios:** Compilação de gráficos e resultados em um relatório PDF, que inclui mensagens automáticas indicando o status do desempenho analisado. Esse documento pode ser utilizado para registros e revisões futuras.

**Acessibilidade do Software:** Com a conversão do script Python em um executável (.exe), a aplicação pode ser utilizada em qualquer computador, sem necessidade de configuração prévia.

Essa abordagem não apenas automatiza e simplifica a análise de dados, mas também assegura a precisão dos resultados apresentados, tornando-os adequados para uso em processos de validação de desempenho de sistemas.

## 4 DESENVOLVIMENTO

Neste capítulo, serão apresentadas as etapas detalhadas de desenvolvimento do trabalho, assim como as estratégias utilizadas para torná-lo funcional, atendendo as especificações dos testes de latência para o simulador EMB-120. O diagrama geral do funcionamento do sistema desenvolvido é apresentado na FIGURA 30 para auxiliar a compressão de cada etapa do circuito.

### 4.1 CONVERSÃO DOS SINAIS ANALÓGICOS PARA DIGITAIS

O circuito de conversão dos sinais analógicos para digitais foi uma etapa crucial para o desenvolvimento do trabalho. Para melhor entendimento, é apresentado na FIGURA 40 uma visão geral desse funcionamento, abrangendo o desenvolvimento das entradas dos sinais no *Hardware* CPD, do circuito somador para conversão da escala de tensão lida, da proteção contra picos de tensões positivos e negativos, da lógica para seleção dos sinais necessário para cada categoria de teste e finalmente a leitura e conversão pelo ADS1115.

FIGURA 40 – Visão Geral (Conversão AD)



FONTE: Os autores (2024).

#### 4.1.1 Entrada dos Sinais no *Hardware* CPD

Foi definida uma entrada padrão para os sinais vindos do *Latency Test Panel*, e confeccionado um cabo tipo flat com conector tipo jacaré devidamente identificados, para o painel de teste de latência, e na outra ponta um conector polarizado, evitando erros de conexão na placa. Esse cabo (FIGURA 41) permite que os sinais cheguem na entrada do *Hardware* CPD de forma padronizada e confiável.



$$V_{out} = G * \left( \frac{V_{r1}}{N} + \frac{V_{r2}}{N} \right)$$

Para:

**$V_{r1}$** : Tensão fixa em 5V para *offset* do sinal.

**$V_{r2}$** : Sinal

**$G$**  : Ganho

**$N$** : Número de entradas no circuito somador

**$V_{out}$**  : Tensão de Saída

Sendo  $G$  o ganho definido como:

$$G = \frac{R3}{R4} + 1$$

Como o circuito não possui resistores de alimentação, o ganho será unitário:

$$G = \frac{R3}{R4} + 1$$

$$R3 = 0 \Omega \text{ e } R4 = 0 \Omega$$

$$\mathbf{G = 1}$$

Sabendo que o sinal proveniente do *Latency Test Panel* seja na faixa de -5V a 5V, pode-se exemplificar alguns níveis de tensão esperados:

Para  $V_{r2} = -5V$  (valor mínimo da faixa):

$$V_{out} = G * \left( \frac{V_{r1}}{N} + \frac{V_{r2}}{N} \right)$$

$$V_{out} = 1 * \left( \frac{5V}{2} + \frac{-5V}{2} \right)$$

$$V_{out} = 1 * [(2,5V + (-2,5V))]$$

$$V_{out} = 1 * 0V$$

$$\mathbf{V_{out} = 0V}$$

Para  $V_{r2} = 0V$  (valor médio da faixa):

$$V_{out} = G * \left( \frac{V_{r1}}{N} + \frac{V_{r2}}{N} \right)$$

$$5V \quad 0V$$

$$V_{out} = 1 * \left( \frac{\quad}{2} + \frac{\quad}{2} \right)$$

$$V_{out} = 1 * (2,5V + 0)$$

$$V_{out} = 1 * 2,5V$$

$$\mathbf{V_{out} = 2,5V}$$

Para  $V_{r2} = 5V$  (valor máximo da faixa):

$$V_{out} = G * \left( \frac{V_{r1}}{N} + \frac{V_{r2}}{N} \right)$$

$$5V \quad 5V$$

$$V_{out} = 1 * \left( \frac{\quad}{2} + \frac{\quad}{2} \right)$$

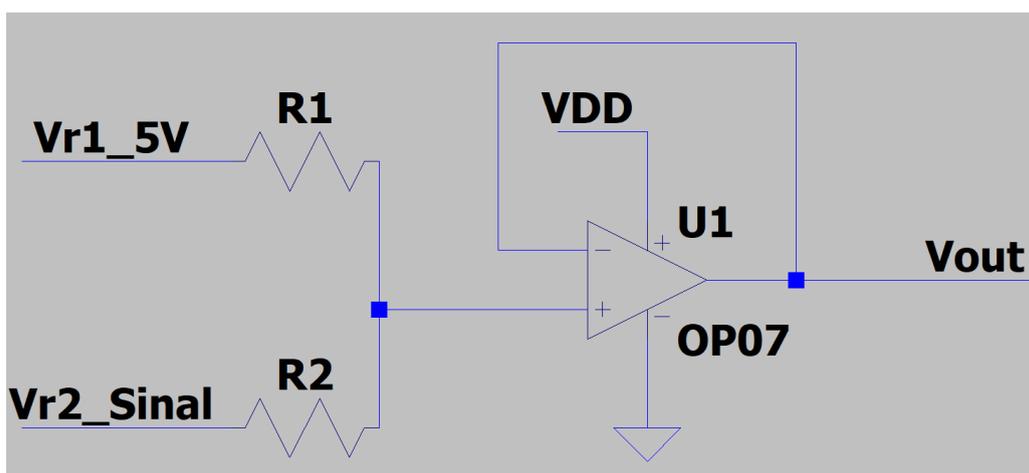
$$V_{out} = 1 * (2,5V + 2,5V)$$

$$V_{out} = 1 * 5V$$

$$\mathbf{V_{out} = 5V}$$

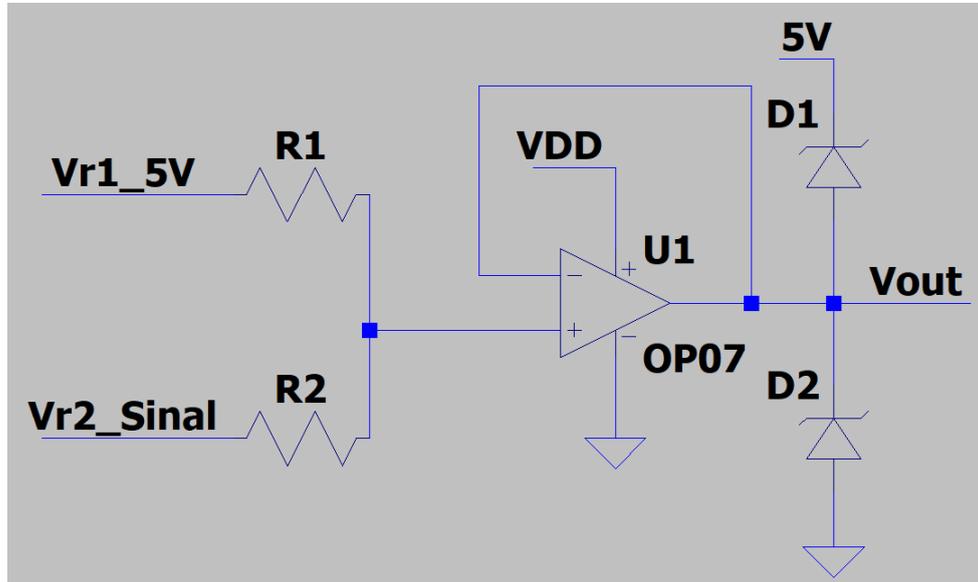
Com essa configuração é possível realizar o correto ajuste da faixa de tensão, possibilitando ler apenas valores positivos, o que irá contribuir para a sequência do circuito e seu correto funcionamento. A utilização de amplificadores operacional garante uma leitura linear para o ajuste na faixa de tensão, pois possui uma alta impedância de entrada e uma impedância de saída baixa. O circuito de ajuste de faixa de tensão utilizado é definido na FIGURA 43:

FIGURA 43 – Circuito Somador Usado



Pensando em evitar possíveis surtos de tensão negativos, foi implementado no circuito anterior um arranjo de Diodos Zeners de 5.1V na saída do sinal  $V_{out}$ . O circuito final é o apresentado na FIGURA 44:

FIGURA 44 – Circuito Ajuste de Faixa com Proteção contra Picos de Tensão Negativo



FONTE: Os autores (2024).

#### 4.1.3 Seleção de Sinais por Tipo de Teste

Relembrando a distribuição dos sinais por tipo de teste, chegamos no seguinte arranjo:

- Teste de Latência para *Pitch* (*COLUMN FORCE TEST ELEVATOR*):
  - COL FORCE;
  - AIRCRAFT PITCH;
  - MOTION ROLL / PITCH;
  - VISUAL YAW.
  
- Teste de Latência para *Roll* (*WHEEL FORCE TEST AILERON*):
  - WHELL FORCE;
  - AIRCRAFT ROLL;
  - MOTION ROLL / PITCH;
  - VISUAL YAW.

- Teste de Latência para *Yaw* (*RUDDER PEDAL FORCE TEST RUDDER*):
  - PEDAL FORCE;
  - AIRCRAFT HDG;
  - MOTION ROLL / PITCH;
  - VISUAL ROLL / PITCH.

Notou-se que para cada tipo de teste, é necessário um sinal de força, um sinal de *aircraft* e um sinal de visual, sendo constante para todas as superfícies o sinal de *Motion Pitch/Roll*.

O multiplexador CD4052 foi implementado para permitir a leitura de múltiplos sinais usando um número reduzido de entradas no ADS1115. Este componente é um multiplexador analógico e digital 4x1, capaz de selecionar uma entre quatro entradas com base em dois pinos de controle (S0 e S1). O projeto utiliza três multiplexadores, controlados pelos pinos digitais D2, D3 e D4 do Arduino Mega do *Hardware CPD* para selecionar qual sinal será lido.

Cada multiplexador recebe diferentes entradas, organizadas conforme o tipo de teste a ser executado. Abaixo estão definidas as ligações usadas, separando sinais de força, aeronave e visual por multiplexadores:

- **Multiplexador 1 (Força):**
  - X0: COL FORCE
  - X1: WHEEL FORCE
  - X2: PEDAL FORCE
  - X3: GND
  
- **Multiplexador 2 (Aeronave):**
  - Y0: AIRCRAFT PITCH
  - Y1: AIRCRAFT ROLL
  - Y2: AIRCRAFT HDG
  - Y3: GND
  
- **Multiplexador 3 (Visual):**
  - X0: VISUAL YAW
  - X1: VISUAL ROLL/PITCH
  - X2: GND
  - X3: GND

A seleção do canal em cada multiplexador é feita de acordo com o teste ativo (*Pitch*, *Roll* ou *Yaw*), através dos sinais S0 e S1 de cada multiplexador (FIGURA 45).

FIGURA 45 – Código Seleção MUX

```
void setTestChannels(char test) {
    // Resetar os sinais de controle
    digitalWrite(S0_MUX_1_e_2, LOW);
    digitalWrite(S1_MUX_1_e_2, LOW);
    digitalWrite(S0_MUX_3, LOW);

    switch (test) {
        case 'P': // Teste Pitch
            digitalWrite(S0_MUX_1_e_2, LOW);
            digitalWrite(S1_MUX_1_e_2, LOW);
            digitalWrite(S0_MUX_3, LOW);
            break;
        case 'R': // Teste Roll
            digitalWrite(S0_MUX_1_e_2, LOW);
            digitalWrite(S1_MUX_1_e_2, HIGH);
            digitalWrite(S0_MUX_3, LOW);
            break;
        case 'Y': // Teste Yaw
            digitalWrite(S0_MUX_1_e_2, HIGH);
            digitalWrite(S1_MUX_1_e_2, LOW);
            digitalWrite(S0_MUX_3, HIGH);
            break;
    }
}
```

FONTE: Os autores (2024).

A função `setTestChannels(char test)` é utilizada para configurar dois chips multiplexadores CD4052, com o objetivo de selecionar os sinais corretos para cada teste (*Pitch*, *Roll*, *Yaw*). Nesse projeto, dois circuitos integrados são usados: o primeiro circuito integrado controla o Multiplexador 1 (Força) e o Multiplexador 2 (Aeronave), enquanto o segundo circuito integrado controla apenas o Multiplexador 3 (Visual). Os multiplexadores selecionam os sinais de entrada que serão enviados para o sistema de aquisição de dados com base no tipo de teste.

Logo no início da função no *software*, todos os sinais de controle dos pinos que configuram os multiplexadores são resetados para *LOW*, tanto para o primeiro quanto para o segundo circuito integrado CD4052. Isso garante que qualquer configuração anterior seja desfeita, permitindo iniciar a configuração do teste de maneira limpa.

Os pinos de controle `S0_MUX_1_e_2` (D2) e `S1_MUX_1_e_2` (D3) controlam o primeiro circuito integrado, que abriga o Multiplexador 1 e o Multiplexador 2. O pino `S0_MUX_3` (D4) controla o segundo circuito integrado, onde está o Multiplexador 3. O sistema utiliza esses pinos para selecionar diferentes entradas analógicas de cada multiplexador, conforme o teste selecionado.

No caso do teste de *Pitch*, os pinos de controle S0 e S1 do primeiro circuito permanecem ambos em *LOW*, o que configura o Multiplexador 1 e o Multiplexador 2 para

selecionar a primeira entrada de cada um. No segundo circuito, o pino *S0\_MUX\_3* também permanece em *LOW*, selecionando a primeira entrada do Multiplexador 3. Isso garante que os sinais apropriados para o teste de *Pitch* sejam capturados.

Para o teste de *Roll*, o pino *S1\_MUX\_1\_e\_2* do primeiro circuito é configurado como *HIGH*, enquanto *S0\_MUX\_1\_e\_2* continua em *LOW*. Essa combinação faz com que o Multiplexador 1 e o Multiplexador 2 selecionem a segunda entrada de cada um. No segundo circuito, *S0\_MUX\_3* permanece em *LOW*, mantendo a primeira entrada do Multiplexador 3 selecionada. Essa configuração direciona os sinais corretos para o teste de *Roll*.

Quando o teste de *Yaw* é selecionado, o pino *S0\_MUX\_1\_e\_2* do primeiro circuito é ajustado para *HIGH* e *S1\_MUX\_1\_e\_2* é mantido em *LOW*, o que faz com que o Multiplexador 1 e o Multiplexador 2 selecionem a terceira entrada de cada um. No segundo circuito, o pino *S0\_MUX\_3* também é ajustado para *HIGH*, selecionando a segunda entrada do Multiplexador 3. Essa combinação garante que os sinais corretos sejam capturados para o teste de *Yaw*.

A TABELA 3 demonstra o chaveamento do CD4052 com base nas entradas *S0* e *S1*. A TABELA 4 define todas as possibilidades de seleção de teste, a partir dos inputs recebidos nas entradas *D2*, *D3* e *D4* do Arduino Mega do *Hardware CPD*.

TABELA 3 – CHAVEAMENTO CD4052

S0	S1	Saída
0	0	X0
0	1	X1
1	0	X2
1	1	X3

FONTE: Os autores (2024).

TABELA 4 – CHAVEAMENTO MULTIPLEXADOR POR TIPO DE TESTE

Teste Selecionado	D2	D3	D4	Saída MUX 1	Saída MUX 2	Saída MUX 3
	S1_MUX_1_e_2	S1_MUX_1_e_2	S0_MUX_3			
Teste de Pitch	0	0	0	COL FORCE	AIRCRAFT PITCH	VISUAL YAW
Teste de Roll	0	1	0	WHEEL FORCE	AIRCRAFT ROLL	VISUAL YAW
Teste de Yaw	1	0	1	PEDAL FORCE	AIRCRAFT HDG	VISUAL PITCH/ROLL

FONTE: Os autores (2024).

Portanto, a função controla dois *chips* multiplexadores CD4052 e, com base no tipo de teste selecionado, configura os pinos de controle para selecionar diferentes sinais de

entrada de cada multiplexador. O primeiro *chip* é responsável pelos sinais dos Multiplexadores 1 e 2, enquanto o segundo *chip* lida com os sinais do Multiplexador 3. A função coordena as entradas para garantir que os sinais corretos sejam enviados para o sistema de aquisição de dados conforme o teste (*Pitch, Roll ou Yaw*).

Os pinos S0 e S1 do CD4052 controlam a entrada que será direcionada para os canais A0, A1 e A3 do ADS1115, enquanto o sinal MOTION ROLL/PITCH é ligado diretamente no A2 do ADS1115 após tratamento pelo circuito de ajuste de faixa, já que ele é comum a todos os testes. A saída do Multiplexador 1 é conectada à A0 do ADS1115, a saída do Multiplexador 2 é conectada à A1 do ADS1115 e a saída do Multiplexador 3 é conectada à A3 do ADS1115.

#### 4.1.4 Leitura do Conversor ADS1115

Para concluir o bloco de conversão de sinais analógicos para digitais, o último processo consiste na leitura do conversor ADS1115, com os sinais dentro da faixa de tensão correta e selecionados conforme tipo de teste. Do ponto de vista do *software*, foi utilizado uma biblioteca específica para utilização do conversor ADS1115. Essa biblioteca, <ADS1115\_WE.h>, fornece as funções essenciais para interagir com o dispositivo.

Buscando a melhor performance possível do ADS1115, o componente foi configurado para trabalhar em 860 amostras por segundo e comunicação I2C em 400KHz. Verificações foram realizadas na inicialização do componente, para que o piloto possa realizar o teste apenas se o ADS1115 tenha iniciado de forma satisfatória. A leitura do sinal é realizada pela função *readChannel()*, que consiste em converter o sinal analógico lido em digital, utilizando 16bits de resolução e retornando o valor em volts, conforme FIGURA 46.

FIGURA 46 – Código Função *readChannel*

```
float readChannel(ADS1115_WE &adc, ADS1115_MUX channel) {  
    adc.setCompareChannels(channel);  
    //delay(1); // Aguarda a conversão  
    return adc.getResult_V(); // Retorna o valor em volts  
}
```

FONTE: Os autores (2024).

## 4.2 GRAVAÇÃO DOS DADOS NO CARTÃO SD

A gravação no cartão SD é crucial para armazenar os dados adquiridos pelo ADS1115. O funcionamento dessa parte é abordado em várias seções do código, que incluem a configuração inicial do cartão, a abertura de arquivos, a gravação de dados e o encerramento da aquisição.

Do ponto de vista do *software*, foi utilizada uma biblioteca específica para utilização cartão SD. Essa biblioteca, <SD.h>, fornece as funções essenciais para interagir com o dispositivo.

Na função *setup()*, após inicializar o ADS1115 e configurar a comunicação I2C, o código tenta inicializar o cartão SD usando (FIGURA 47).

FIGURA 47 – Código Inicialização Cartão SD

```
if (!SD.begin(SD_CS_PIN)) {
    Serial.println("Erro ao iniciar o SD!");
}
```

FONTE: Os autores (2024).

A gravação dos dados no cartão SD é realizada quando o pino de controle *START/STOP* está em nível alto (*HIGH*). A gravação é feita em um arquivo de texto .txt, com os dados dos testes sendo separados por ponto e vírgula para facilitar a análise posterior em Python (FIGURA 48).

FIGURA 48 – Código Formato Cabeçalho

```
// Calcula o tempo corrido desde o início
unsigned long elapsedMillis = currentMillis - startMillis;

// Constrói a linha de dados
String dataString = String(signal1, 4) + ";" + String(signal2, 4) + ";" +
    String(motionData, 4) + ";" + String(signal3, 4) + ";" +
    String(elapsedMillis);

// Imprime os dados no Serial e escreve no arquivo no cartão SD
Serial.println(dataString);
dataFile.println(dataString);

// Garante que os dados sejam gravados no arquivo
dataFile.flush();
```

FONTE: Os autores (2024).

A função *flush()* garante que todos os dados gravados em buffer sejam escritos no cartão, evitando a perda de dados em caso de falha de energia ou desconexão.

A função *startAcquisition()* (FIGURA 49) é chamada para iniciar a gravação. Nesta função, um arquivo é aberto com base no tipo de teste selecionado (*Pitch*, *Roll*, ou *Yaw*):

FIGURA 49 – Código Função *startAcquisition*

```
void startAcquisition() {
    String fileName;
    if (currentTest == 'P') {
        fileName = "Pitch.txt";
        Serial.println("Pitch.txt");
    } else if (currentTest == 'R') {
        fileName = "Roll.txt";
        Serial.println("Roll.txt");
    } else if (currentTest == 'Y') {
        fileName = "Yaw.txt";
        Serial.println("Yaw.txt");
    }

    dataFile = SD.open(fileName, FILE_WRITE);
    if (!dataFile) {
        Serial.println("Erro ao abrir o arquivo.");
        return;
    }
}
```

FONTE: Os autores (2024).

Se a abertura do arquivo falhar, uma mensagem de erro é exibida. Caso contrário, um cabeçalho apropriado é gravado no arquivo, de acordo com o tipo de teste. A implementação desse alerta foi essencial para a etapa de testes em bancada. O cabeçalho fornece contexto sobre os dados que serão gravados a seguir.

FIGURA 50 – Código Cabeçalho por Tipo de Teste

```
// Escrever cabeçalho de acordo com o teste
switch (currentTest) {
    case 'P':
        //dataFile.println("Column Force; Aircraft Pitch; Motion Pitch/Roll; Visual Yaw; Tempo (ms)");
        dataFile.println("Visual Yaw; Column Force; Aircraft Pitch; Motion Pitch/Roll; Tempo (ms)");
        Serial.println("Visual Yaw; Column Force; Aircraft Pitch; Motion Pitch/Roll; Tempo (ms)");
        break;
    case 'R':
        //dataFile.println("Wheel Force; Aircraft Roll; Motion Pitch/Roll; Visual Yaw; Tempo (ms)");
        dataFile.println("Visual Yaw; Wheel Force; Aircraft Roll; Motion Pitch/Roll; Tempo (ms)");
        Serial.println("Visual Yaw; Wheel Force; Aircraft Roll; Motion Pitch/Roll; Tempo (ms)");
        break;
    case 'Y':
        //dataFile.println("Pedal Force; Aircraft HDG; Motion Pitch/Roll; Visual Pitch/Roll; Tempo (ms)");
        dataFile.println("Visual Pitch/Roll; Pedal Force; Aircraft HDG; Motion Pitch/Roll; Tempo (ms)");
        Serial.println("Visual Pitch/Roll; Pedal Force; Aircraft HDG; Motion Pitch/Roll; Tempo (ms)");
        break;
}
```

FONTE: Os autores (2024).

Por fim, na função `stopAcquisition()`, na FIGURA 51, o arquivo é fechado e as variáveis de estado são atualizadas:

FIGURA 51 – Código Função `stopAcquisition`

```
void stopAcquisition() {
  if (isAcquiring) {
    dataFile.close();
    isAcquiring = false;
    digitalWrite(LED_RECORDING_PIN, LOW);
    digitalWrite(LED_STANDBY_PIN, HIGH);
    Serial.println("Aquisição finalizada e arquivo salvo.");
  }
}
```

FONTE: Os autores (2024).

### 4.3 INTERFACE COM PILOTO

O Hardware Simulador é responsável pela interface com o piloto. Esse Hardware auxiliará a visualização de todos os estados do sistema, possibilitando a configuração de tipo de teste, início e término da medição dos sinais. Foi desenvolvido uma placa com quatro botões, sendo eles:

**Botão *Start/Stop*:** Determina início e fim da medição;

**Botão *Pitch*:** Seleciona tipo de teste *Pitch*;

**Botão *Roll*:** Seleciona tipo de teste *Roll*;

**Botão *Yaw*:** Seleciona tipo de teste *Yaw*.

O funcionamento é simples e intuitivo. Após configurado o teste dentro da cabine de simulação pelo *IOS*, o piloto deverá selecionar um tipo de teste pelo *Hardware Simulador* apertando um dos botões entre *Pitch*, *Roll* e *Yaw*. Após selecionado uma das três superfícies para teste, o piloto poderá apertar o botão de *Start/Stop* para iniciar e concluir a medição.

Além dos botões, o *Hardware Simulador* possui um display que notificará todas as etapas do processo, como superfície selecionada, status de gravação dos sinais, status do cartão SD, entre outros.

Se tratando do software, o processamento dos botões da placa *Hardware Simulador* é feito pelo Arduino Mega do Hardware CPD. Essa estratégia foi adotada como alternativa

à implementação de um sinal de *feedback*. Quando apertado um botão de seleção de teste no simulador, o sinal elétrico chega ao *Hardware* CPD via cabo e é realizado o processamento de seleção de tipo de teste nos Multiplexadores.

Como não foi utilizado um sinal de feedback, a partir dos inputs dos botões e processamento do *Hardware* CPD, o próprio envia três bits através de três portas digitais (D31, D33 e D35) para o *Hardware* Simulador, sinalizando as etapas do processo. Ao todo são seis estados e estão discriminados pela TABELA 5.

TABELA 5 – ESTADOS DO SISTEMA

Estado	Arduino MEGA D31	Arduino MEGA D33	Arduino MEGA D35
<i>Standby</i>	0	0	0
Teste Pitch	0	0	1
Teste Roll	0	1	0
Teste Yaw	0	1	1
Gravando	1	0	0
Arquivo Salvo	1	0	1

FONTE: Os autores (2024).

Esses três sinais digitais são recebidos pelas portas digitais (D2, D3 e D4) e interpretados no *Hardware* Simulador. O Arduino Uno do *Hardware* Simulador utiliza esses sinais para informar ao piloto por meio do *display* qual o estado atual. A combinação da seleção de um teste mais o início da gravação pelo botão *Start/Stop*, proporciona mensagens ao piloto, conforme TABELA 6:

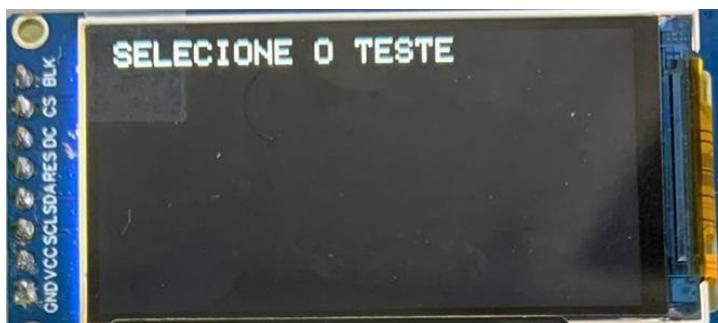
TABELA 6 – MENSAGENS NO *DISPLAY* POR ESTADO DO SISTEMA

Estado	<i>Display</i>
Modo de espera	Selecione o Teste
Teste Pitch	Teste Pitch Selecionado
Teste Roll	Teste Roll Selecionado
Teste Yaw	Teste Yaw Selecionado
Gravando	Gravando teste "Tipo de Teste"
Arquivo Salvo	"Tipo de Teste" Salvo

FONTE: Os autores (2024).

A visualização final da interface com piloto para modo de espera está conforme FIGURA 52.

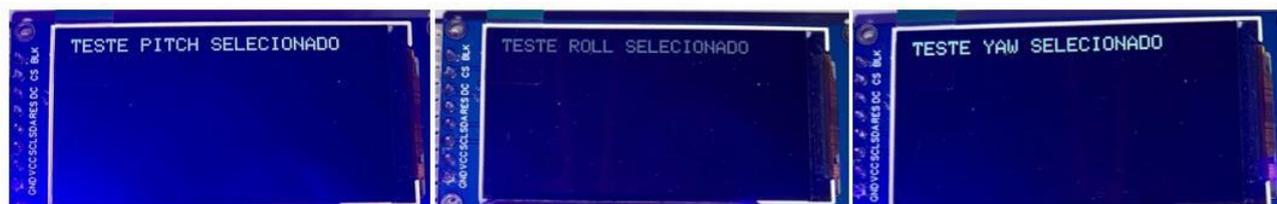
FIGURA 52 – *Display* Estado de Espera



FONTE: Os autores (2024).

A visualização final da interface com piloto após seleção dos dados está conforme FIGURA 53.

FIGURA 53 – *Display* Estado de Seleção



FONTE: Os autores (2024).

A visualização final da interface com piloto durante gravação dos dados está conforme FIGURA 54.

FIGURA 54 – *Display* Estado de Gravação dos Dados



FONTE: Os autores (2024).

A visualização final da interface com piloto após a finalização dos testes está conforme FIGURA 55.

FIGURA 55 – *Display* Estado Salvamento dos Dados



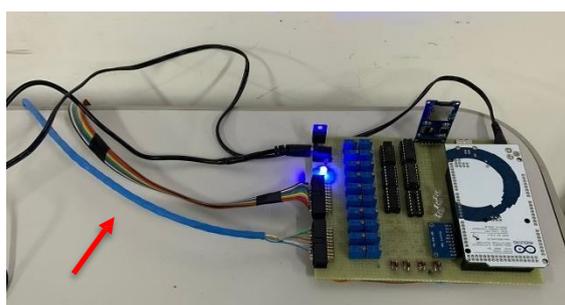
FONTE: Os autores (2024).

O aviso de teste salvo permanece no display por dois segundos e o sistema volta para o Modo de Espera, aguardando um novo teste ser selecionado e iniciado pelo piloto de dentro da cabine de simulação.

#### 4.4 COMUNICAÇÃO ENTRE *HARDWARE* CPD E *HARDWARE* SIMULADOR

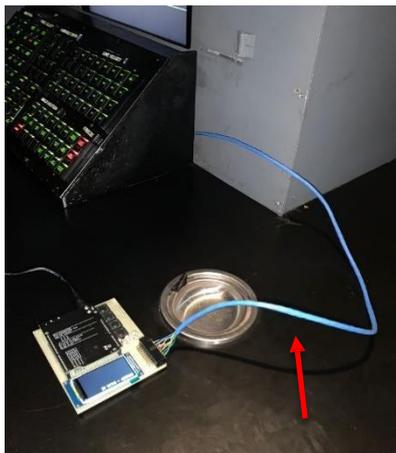
A comunicação entre o *Hardware* CPD e *Hardware* Simulador é por um cabo de rede de 8 vias destacado nas FIGURAS 56 e 57. Foi desenvolvido um cabo tipo de rede com oito vias e conectores polarizados para evitar inversão de conexão.

FIGURA 56 – Comunicação lado CPD



FONTE: Os autores (2024).

FIGURA 57 – Comunicação lado Simulador



FONTE: Os autores (2024).

Quatro vias dessa conexão foram destinadas para o sinal dos botões de seleção de tipo de teste e o sinal do botão *Start/Stop*. Três vias foram destinadas para a comunicação entre *Hardware CPD* e *Hardware Simulador* e uma via de referência para *ground* entra as placas.

#### 4.5 PLOTAGEM GRÁFICA

O sistema desenvolvido inclui uma aplicação robusta para leitura, processamento e análise dos dados coletados pelo *Hardware CPD*. A aplicação não apenas calcula a latência entre sinais, mas também fornece recursos interativos, como a seleção manual de pontos em gráficos para cálculos precisos, zoom para inspeção detalhada e geração de relatórios em PDF com todas as análises consolidadas. Com o objetivo de proporcionar acessibilidade e facilidade de uso, a aplicação foi convertida em um arquivo executável *.exe*, permitindo que usuários rodem o programa em qualquer computador sem a necessidade de instalar *Python* ou bibliotecas adicionais.

##### 4.5.1 Leitura dos Dados do Cartão SD

Os dados coletados pelo *Hardware CPD* são armazenados em arquivos *.txt*, onde cada linha corresponde a uma medição realizada para os quatro canais. Para garantir que a análise seja precisa, os dados são carregados e estruturados conforme o tipo de teste (*PITCH*, *ROLL* ou *YAW*). Essa diferenciação é realizada com base nos cabeçalhos apropriados, associados a cada teste por meio de um dicionário. O seguinte trecho de código na FIGURA 58 exemplifica como os cabeçalhos são definidos:

FIGURA 58 – Dicionário de Cabeçalhos

```

5# Dicionário de cabeçalhos para os diferentes arquivos
6headers = {
7    'PITCH': ['Column Force', 'Aircraft Pitch', 'Motion Pitch/Roll', 'Visual Yaw', 'Tempo (ms)'],
8    'ROLL': ['Wheel Force', 'Aircraft Roll', 'Motion Pitch/Roll', 'Visual Yaw', 'Tempo (ms)'],
9    'YAW': ['Pedal Force', 'Aircraft HDG', 'Motion Pitch/Roll', 'Visual Pitch/Roll', 'Tempo (ms)']

```

FONTE: Os autores (2024).

Com o cabeçalho apropriado definido, a função `read_data` lê o arquivo e retorna um `DataFrame` estruturado (FIGURA 59):

FIGURA 59 – Código Função `read_data`

```

26# Função para ler os dados do arquivo .txt com base no cabeçalho específico
27def read_data(file_path, header):
28    try:
29        data = pd.read_csv(file_path, sep=';', skiprows=1, header=None, names=header)
30        print(f"Dados lidos com sucesso do arquivo: {file_path}")
31        return data
32    except Exception as e:
33        print(f"Erro ao Ler o arquivo: {e}")
34        return None

```

FONTE: Os autores (2024).

Esse processamento inicial é vital para garantir que as etapas subsequentes de reconversão, análise e plotagem sejam realizadas corretamente.

#### 4.5.2 Reconversão dos Dados

Os sinais adquiridos pelo ADS1115 foram normalizados para o intervalo de 0V a 5V, mas os valores originais precisam ser recuperados para a análise. A reconversão, de 0V a 5V para -5V a 5V, é feita por uma transformação linear aplicada a cada coluna do `DataFrame`, exceto à de tempo (FIGURA 60).

FIGURA 60 – Código Função `reconvert_signals`

```

36# Função para reverter os sinais de 0-5V para -5V a 5V (se necessário)
37def reconvert_signals(data):
38    for column in data.columns[:-1]:
39        data[column] = data[column] * 2 - 5
40    return data

```

FONTE: Os autores (2024).

A reconversão dos dados é feita pela seguinte transformação linear:

$$Vr = Vp * 2 - 5$$

Sendo:

$Vr$ : Valor Real na faixa de -5V a 5V

$Vp$ : Valor positivo passado pelo bloco de ajuste de faixa de tensão (0V a 5V)

Pode-se exemplificar alguns valores obtidos com a transformação linear realizada pela aplicação em *Python*.

Para dados em 0V:

$$Vr = Vp * 2 - 5$$

$$Vr = 0V * 2 - 5$$

$$\mathbf{Vr = -5V}$$

Para dados em 2,5V:

$$Vr = Vp * 2 - 5$$

$$Vr = 2,5 * 2 - 5$$

$$Vr = 5 - 5$$

$$\mathbf{Vr = 0V}$$

Para dados em 5V:

$$Vr = Vp * 2 - 5$$

$$Vr = 5 * 2 - 5$$

$$Vr = 10 - 5$$

$$\mathbf{Vr = 5V}$$

Esse ajuste é imprescindível para que os valores reflitam o comportamento real dos sinais coletados. Essa função é aplicada imediatamente após a leitura dos dados.

### 4.5.3 Definição do Tipo de Teste

A aplicação interativa solicita ao usuário que escolha o tipo de teste a ser analisado. Essa escolha define não apenas os cabeçalhos, mas também o sinal de referência para o cálculo de latência (FIGURA 61).

FIGURA 61 – Definição de Sinal para Teste de Latência

```
168# Escolha do arquivo para análise
169 file_choice = input("Escolha o arquivo para análise (PITCH, ROLL, YAW): ").strip().upper()
170 file_path = f'{file_choice}.txt'
```

FONTE: Os autores (2024).

Após a escolha, a aplicação verifica a validade da entrada e associa o cabeçalho e o sinal de referência (FIGURA 62).

FIGURA 62 – Código Associação Cabeçalho e Sinal de Referência

```
172# Verifica se o cabeçalho correspondente existe no dicionário
173 if file_choice in headers and file_choice in reference_signals:
174     header = headers[file_choice]
175     reference_signal = reference_signals[file_choice]
176 else:
177     print("Escolha inválida. O programa será encerrado.")
178     exit()
```

FONTE: Os autores (2024).

Esse mecanismo evita erros de análise e garante que os dados carregados sejam tratados adequadamente conforme o contexto do teste.

### 4.5.4 Cálculo de Latência

O cálculo de latência no programa é realizado a partir da interação do usuário com os gráficos gerados. Primeiramente, o programa apresenta os sinais adquiridos de cada canal em gráficos individuais. O usuário pode clicar diretamente no gráfico para selecionar os pontos de interesse, sendo um no canal de referência e os demais nos canais alvo. A posição do clique é mapeada para o eixo do tempo, registrando o momento exato associado ao ponto selecionado.

Por exemplo, o evento de clique no gráfico é tratado com a função onclick, que adiciona uma linha vertical no local clicado e salva o tempo correspondente (FIGURA 63).

FIGURA 63 – Código Função *OnClick*

```
def onclick(event):
    if event.inaxes in axes:
        ax = event.inaxes
        x = event.xdata # Captura o tempo do clique
        ax.axvline(x, color='red', linestyle='--') # Adiciona a linha vertical
        times_selected[axes.index(ax)] = x # Armazena o tempo selecionado
        plt.draw()
```

FONTE: Os autores (2024).

Após a seleção dos pontos de referência e dos canais alvo, o programa calcula a latência como a diferença entre o tempo do ponto de referência e os tempos selecionados nos canais alvo (FIGURA 64).

FIGURA 64 – Código Cálculo de Latência

```
latencies = [abs(ref_time - t) for t in times_selected[1:]]
```

FONTE: Os autores (2024).

O valor calculado é então comparado com um limite de tolerância (300 ms). Caso alguma latência ultrapasse esse valor, uma mensagem de alerta é exibida no console (FIGURA 65).

FIGURA 65 – Código Resultado Final Cálculo de Latência

```
for i, latency in enumerate(latencies):
    if latency > 300:
        print(f"Alerta: Canal {i+1} com latência {latency} ms fora do limite!")
    else:
        print(f"Canal {i+1}: Latência {latency} ms dentro do limite.")
```

FONTE: Os autores (2024).

Finalmente, todos os resultados são salvos em um relatório gerado automaticamente, incluindo os valores de latência e as observações de conformidade ou não com os limites predefinidos. Essa abordagem interativa e automatizada garante precisão e praticidade na análise de latência dos sinais.

#### 4.5.5 Aplicação executável

Para garantir que a aplicação seja acessível em qualquer ambiente, foi criado um arquivo executável .exe utilizando a biblioteca *PyInstaller*. Esse executável encapsula o código, as bibliotecas e todos os recursos necessários para rodar o programa.

O comando utilizado para gerar o executável foi é apresentado na FIGURA 66.

FIGURA 66 – Comando para Gerar Aplicação Executável

```
pyinstaller --onefile --add-data "data_folder;data" app.py
```

FONTE: Os autores (2024).

A criação do executável elimina a necessidade de instalações adicionais, permitindo que qualquer usuário execute o programa com apenas um clique. Esses recursos tornam a aplicação altamente funcional e acessível, combinando processamento eficiente, análise visual interativa e distribuição prática.

## 5 RESULTADOS

O desenvolvimento do trabalho visou a criação de um sistema digital para a substituição do *chart holder* em simuladores de voo EMB-120, envolvendo a aquisição e tratamento de sinais para testes de latência. A proposta consistia em desenvolver um *hardware* capaz de realizar a coleta dos sinais de maneira fidedigna e desenvolver uma aplicação capaz de apresentar esses sinais no formato digital, eliminando a necessidade de análises manuais que demandavam traçar retas no papel impresso pelo *chart holder* e documentar esses resultados em livros dentro do acervo de documentos da empresa.

### 5.1 HARDWARE E SOFTWARE

O *Hardware* CPD (FIGURA 67) e *Hardware* Simulador (FIGURA 68) foram desenvolvidos em duas placas perfuradas e seus desempenhos foram satisfatórios com o esperado em relação à teoria eletrônica estudada para esse trabalho de conclusão de curso.

FIGURA 67 – *Hardware* CPD



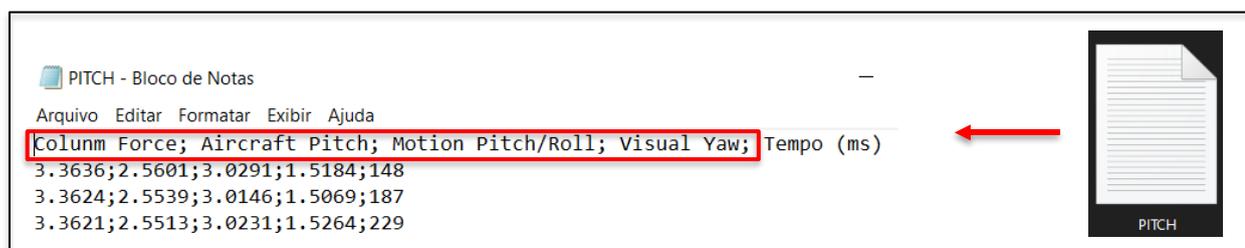
FONTE: Os autores (2024).

FIGURA 68 – *Hardware* Simulador

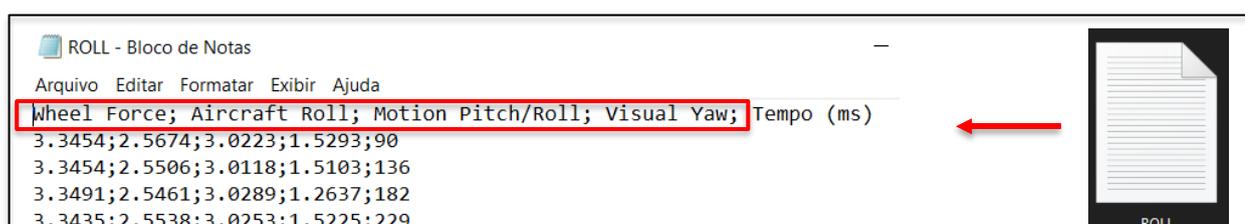


FONTE: Os autores (2024).

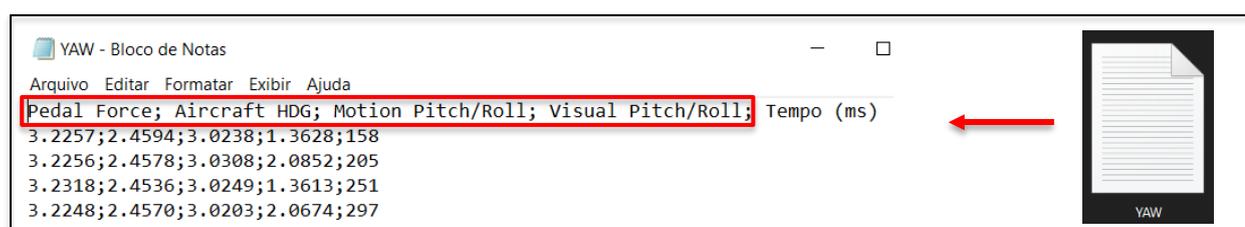
As FIGURAS 69, 70 e 71 ilustram a estrutura de salvamento dos dados nos arquivos gerados pelo *Hardware* CPD. Os arquivos são gravados no formato de texto (.TXT) e nomeados de acordo com a superfície testada. É importante destacar que, para cada tipo de teste, o arquivo gerado contém exclusivamente os dados necessários para a avaliação posterior pelo técnico, conforme o tipo de teste realizado.

FIGURA 69 – Arquivo *Pitch.txt*

FONTE: Os autores (2024).

FIGURA 70 – Arquivo *Roll.txt*

FONTE: Os autores (2024).

FIGURA 71 – Arquivo *Yaw.txt*

FONTE: Os autores (2024).

Foi possível realizar a aquisição dos dados em uma taxa abaixo de 50 milissegundos com amplitude de -5V a 5V, conforme o que foi proposto para o trabalho. Segue o resultado em relação à taxa de transmissão e amplitude do sinal por tipo de teste:

**Teste de Pitch:** Amplitude de -5V a 5V e 39 milissegundos;

**Teste de Roll:** Amplitude de -5V a 5V e 46 milissegundos;

**Teste de Yaw:** Amplitude de -5V a 5V e 47 milissegundos.

### 5.1.1 Comparativo entre novo *Hardware* x *Chart Holder*

No comparativo entre o *Hardware* desenvolvido e o *Chart Holder* é notável a evolução em relação às dimensões, impacto financeiro e ecológico. O sistema ficou mais compacto, barato e sustentável. Sem a necessidade de utilização de papel, o *Hardware* CPD e *Hardware* Simulador possuem dimensões muito inferiores ao *Chart Holder*, demonstrados pela FIGURA 72.

FIGURA 72 – Dimensões *Chart Holder* x Protótipo



FONTE: Os autores (2024).

### 5.1.2 Comparativo entre Software Embarcado x *Chart Holder*

Foi possível desenvolver um software em linguagem C para o *Hardware* CPD e o *Hardware* do simulador, permitindo a comunicação eficiente entre os dois dispositivos. Essa implementação automatizou as tarefas relacionadas à leitura de sinais e ao cálculo de latência para o simulador EMB-120, otimizando significativamente o processo.

A substituição do *chart holder* tradicional por uma aplicação embarcada trouxe diversas vantagens. Uma das principais é a praticidade de edição do código, possibilitando ajustes rápidos e personalizados conforme as necessidades do sistema. Além disso, a manutenção se torna mais simples, uma vez que problemas podem ser identificados e corrigidos diretamente no software, sem a necessidade de intervenções mecânicas ou físicas complexas em um sistema já obsoleto como o *chart Holder*.

A calibração do sistema também foi facilitada, já que o uso de um ambiente digital permite ajustes precisos e repetíveis, reduzindo erros e melhorando a confiabilidade das medições. A flexibilidade do software embarcado ainda permite a expansão futura de funcionalidades e integração com novas tecnologias, aumentando a longevidade e a eficiência do sistema.

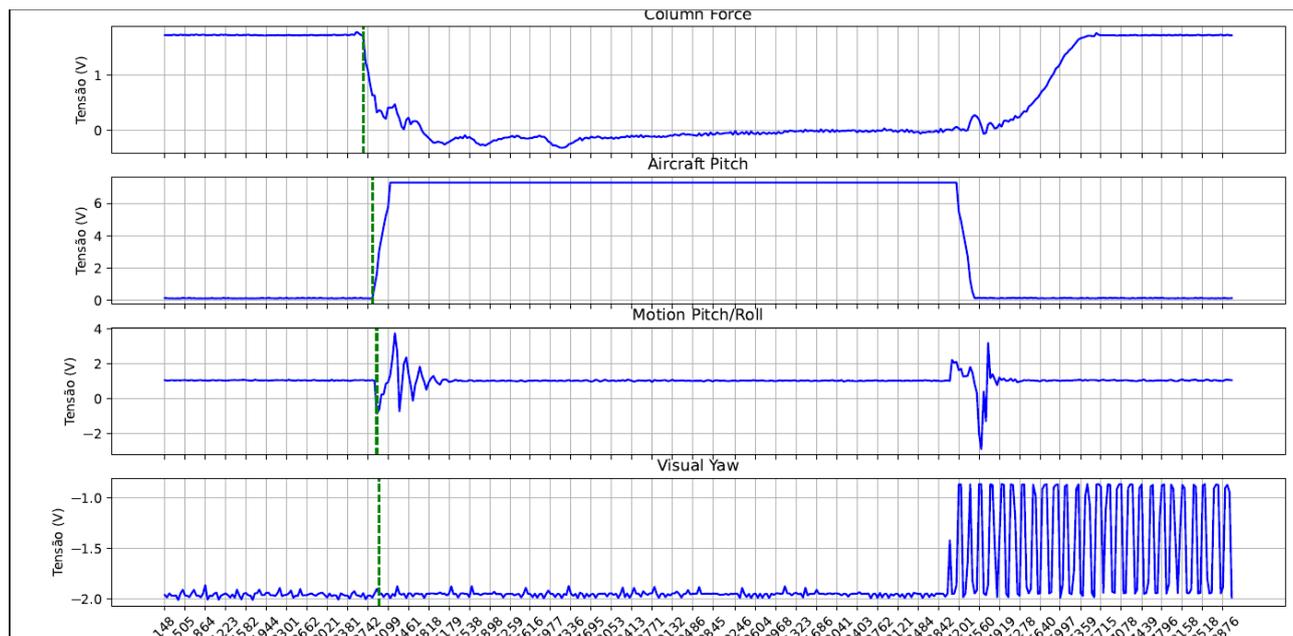
Por fim, a adoção de um software embarcado no lugar de um *chart holder* físico representa um avanço significativo em termos de praticidade, precisão e adaptabilidade, atendendo de forma robusta às demandas do simulador EMB-120 para os testes de latência.

## 5.2 PLOTAGEM GRÁFICA

Com a implementação do código em Python, foi possível modernizar e automatizar o processo de leitura, análise e documentação de sinais do *Latency Test Panel*, eliminando a dependência do *chart holder* físico. O software permite a leitura direta de arquivos de dados gerados pelo *Hardware* CPD em formato .TXT, facilitando o processamento automatizado. Ele organiza as informações em colunas com cabeçalhos padronizados, dependendo do tipo de teste (*Pitch*, *Roll* ou *Yaw*), garantindo clareza e consistência na análise. Além disso, o código inclui a funcionalidade de reconversão de tensões de 0–5V para -5V a 5V, adequando os valores para análises mais precisas e eliminando a necessidade de cálculos manuais adicionais.

Outro destaque importante é o cálculo automático da latência entre os sinais de referência e os outros canais, com indicação clara se os valores estão dentro ou fora dos limites de tolerância (300 ms). Essa automação reduz significativamente o tempo e os erros associados à análise manual. A interface interativa implementada permite que o usuário selecione pontos nos gráficos de maneira precisa, com a possibilidade de ampliar a visualização para garantir maior exatidão nos tempos de latência calculados, aumentando a confiabilidade dos resultados.

Os gráficos gerados para cada teste (FIGURAS 73, 74 e 75), juntamente com os cálculos de latência e mensagens de tolerância são salvos automaticamente em um relatório PDF (FIGURA 76). Esse relatório recebe o nome do teste correspondente, substituindo o registro manual em papel e tornando o processo mais ágil, organizado e sustentável. Além disso, a centralização e digitalização dos dados eliminam a necessidade de armazenamento físico de documentos, reduzindo o risco de perda de informações e facilitando auditorias e revisões futuras.

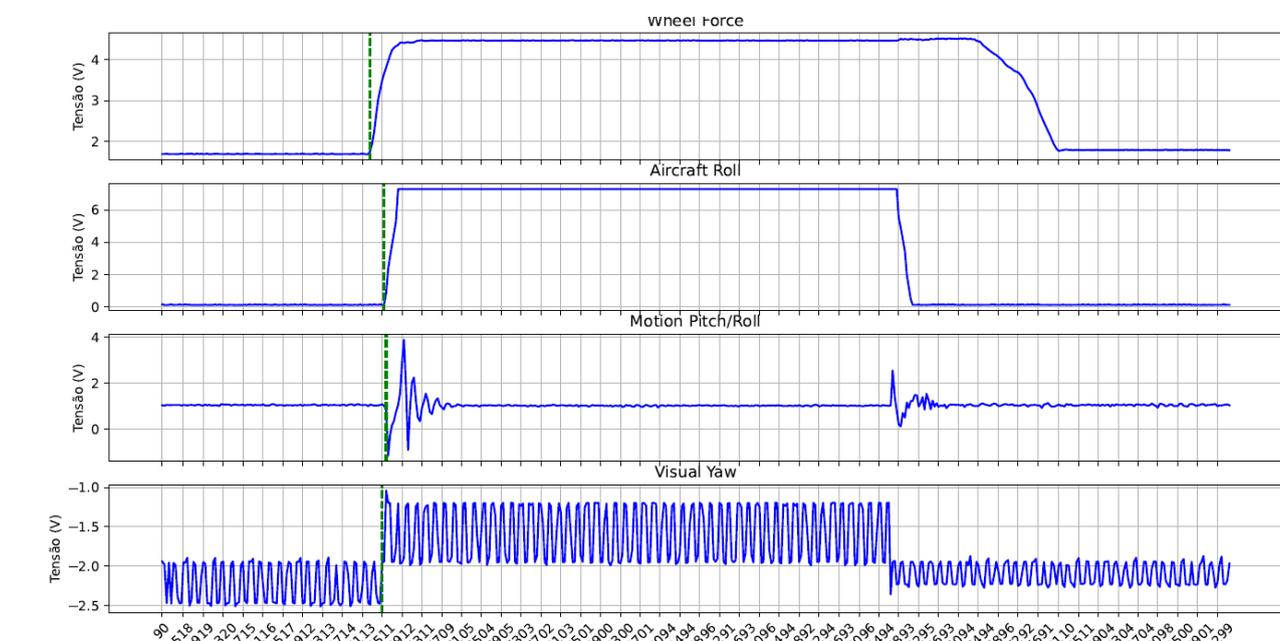
FIGURA 73 – Gráfico Teste *Pitch* e Cálculo de Latência

Latência entre Column Force e Aircraft Pitch: 161.00 ms - Sinal dentro da tolerância

Latência entre Column Force e Visual Yaw: 282.00 ms - Sinal dentro da tolerância

Latência entre Column Force e Motion Pitch/Roll: 239.00 ms - Sinal dentro da tolerância

FONTE: Os autores (2024).

FIGURA 74 – Gráfico Teste *Roll* e Cálculo de Latência

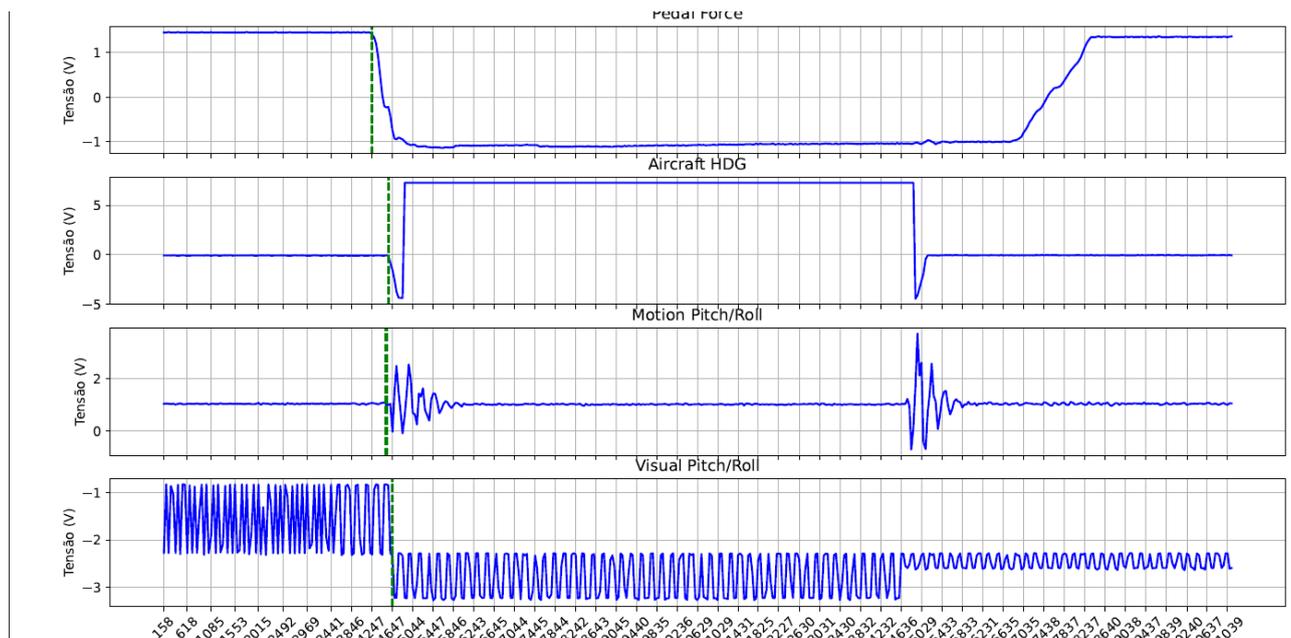
Latência entre Wheel Force e Aircraft Roll: 278.00 ms - Sinal dentro da tolerância

Latência entre Wheel Force e Visual Yaw: 239.00 ms - Sinal dentro da tolerância

Latência entre Wheel Force e Motion Pitch/Roll: 321.00 ms - Sinal fora da tolerância

FONTE: Os autores (2024).

FIGURA 75 – Gráfico Teste Yaw e Cálculo de Latência



Latência entre Pedal Force e Aircraft HDG: 317.00 ms - Sinal fora da tolerância  
 Latência entre Pedal Force e Visual Pitch/Roll: 400.00 ms - Sinal fora da tolerância  
 Latência entre Pedal Force e Motion Pitch/Roll: 278.00 ms - Sinal dentro da tolerância

FONTE: Os autores (2024).

FIGURA 76 – Arquivo .PDF por Tipo de Teste (*Pitch*, *Roll* e *Yaw*)

FONTE: Os autores (2024).

Um dos principais resultados dessa implementação foi a criação de um arquivo executável (.EXE), que consolidou todas as funcionalidades em uma aplicação prática e acessível (FIGURA 77). Com apenas alguns cliques, o técnico responsável pela análise consegue carregar os arquivos de teste, plotar os gráficos, realizar a seleção dos pontos de medição com precisão, incluindo a função de zoom, e salvar o relatório completo em formato PDF. Esse arquivo PDF, por sua vez, pode ser facilmente adicionado aos relatórios técnicos finais, que posteriormente serão enviados para validação pela ANAC.

FIGURA 77 – Arquivo Executável da Aplicação



FONTE: Os autores (2024).

A flexibilidade do software também permite ajustes e expansões, possibilitando a incorporação de novas funcionalidades ou adaptações específicas às necessidades futuras. Essa modernização trouxe ganhos expressivos em eficiência, com a redução do tempo de análise e documentação, maior confiabilidade devido à minimização de erros humanos, sustentabilidade por meio da diminuição do uso de papel e armazenamento físico e acessibilidade com a criação de relatórios digitais fáceis de compartilhar e armazenar. Dessa forma, a adoção desse software e a criação do executável representam um avanço significativo no processo de teste e análise de sinais, facilitando a integração das análises em processos regulatórios e certificações.

## 6 CONCLUSÃO

Este capítulo reafirma o sucesso do desenvolvimento de um sistema de aquisição e tratamento de sinais analógicos específicos do simulador EMB-120 da EPA *Training Center*. Este projeto surgiu para solucionar um cenário crítico enfrentado pela empresa, que utilizava um sistema obsoleto baseado em um *chart holder*. Esse equipamento, além de depender de um papel específico que não está mais disponível no mercado e deixou a empresa sem fornecedores, exigia uma metodologia de trabalho manual e ineficiente. A análise dos gráficos era realizada pelo técnico responsável com régua e canetas, um processo que consumia tempo e estava sujeito a erros humanos.

Além disso, o sistema antigo não contava com automação para o início e fim dos testes, o que exigia a presença de três profissionais: um piloto, um técnico para configurar o IOS e acionar o teste dentro da cabine de simulação, e outro para ligar e desligar o *chart holder*, com a comunicação geralmente por telefone. Essa metodologia aumentava a complexidade operacional e os custos de mão de obra para os testes de latência.

O novo sistema implementado trouxe uma transformação completa nesse processo. Ele automatiza o início e o término da gravação diretamente do simulador, eliminando a necessidade de um profissional exclusivamente para operar o *chart holder*. Essa melhoria não apenas reduziu a necessidade de mão de obra, mas também trouxe maior precisão e confiabilidade para os testes. O sistema é capaz de realizar medições com amplitude de -5V a 5V e uma taxa de transmissão abaixo de 50 ms, atendendo os mesmos padrões técnicos do sistema anterior, mas com eficiência significativamente maior.

Outro avanço importante foi a digitalização dos sinais analógicos e a automação da análise dos dados. O técnico responsável agora pode, plotar gráficos, aplicar zoom, determinar pontos de medição para o cálculo da latência e salvar os resultados em arquivos PDF nomeados automaticamente conforme o teste realizado. Esses relatórios prontos para uso podem ser adicionados diretamente aos documentos que serão enviados à ANAC, eliminando completamente a necessidade de análise manual e aumentando a padronização e a credibilidade dos resultados.

A substituição do *chart holder* por um sistema digital robusto não apenas modernizou o processo de aquisição e análise, mas também trouxe vantagens operacionais e estratégicas para a EPA Training Center. A eliminação do papel como insumo crítico e a automação das etapas mais trabalhosas tornaram os testes mais rápidos, precisos e economicamente viáveis. Além disso, o executável (.exe) desenvolvido para a interface do sistema facilita a adoção e operação por qualquer técnico da equipe, reduzindo a curva de aprendizado e otimizando a rotina de trabalho.

Em resumo, o projeto alcançou todos os objetivos propostos e foi além, ao agregar melhorias significativas ao fluxo operacional da EPA *Training Center*. Ele não só resolve os problemas de tecnologia defasada como também eleva o padrão de qualidade e confiabilidade dos testes de latência, posicionando a empresa de forma competitiva no mercado e em total conformidade com os regulamentos da ANAC. Este trabalho exemplifica como sistemas embarcados podem ser aplicados de forma inovadora para superar desafios reais, trazendo ganhos tecnológicos e operacionais que impactam positivamente toda uma cadeia de processos.

## REFERÊNCIAS

ABEAR, Associação Brasileira das Empresas Aéreas. **Qual a funcionalidade dos flaps de uma aeronave?**. 2019. Disponível em: <<https://www.abear.com.br/imprensa/agencia-abear/noticias/qual-a-funcionalidade-dos-flaps-de-uma-aeronave/>>. Acesso em: 2 abr. 2024.

ANAC. Regulamento Brasileiro da Aviação Civil. **Requisitos para Qualificação e Uso de Dispositivos de Treinamento para Simulação de Voo**. 2020. Disponível em: <[https://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-60/@@display-file/arquivo\\_norma/RBAC60EMD00.pdf](https://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-60/@@display-file/arquivo_norma/RBAC60EMD00.pdf)>. Acesso em: 1 abr. 2024.

EMBRAER, Historical Center. **EMB 120 Brasilia**. 2024. Disponível em: <<https://historical-center.embraer.com/global/en/emb-120-brasilia>>. Acesso em: 1 abr. 2024.

EPA, Training Center. **EMB-120**. 2021. Disponível em: <<https://epa.aero/training-center/emb-120/>>. Acesso em: 3 abr. 2024.

FAA. **Advisory Circular**. 2004. Disponível em: <[https://www.faa.gov/documentLibrary/media/Advisory\\_Circular/AC\\_120-40A.pdf](https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-40A.pdf)>. Acesso em: 1 abr. 2024.

FAB. Museu Aeroespacial. **Primeira Linha Aérea Transatlântica**. 2024. Disponível em: <<https://www2.fab.mil.br/musal/index.php/curiosidades-historicas-item-de-menu/917-primeira-linha-aerea-transatlantica-e-inaugurada>>. Acesso em: 1 abr. 2024.

GOV. Agência Nacional de Aviação Civil (ANAC). **Definições e Normas**. 2021. Disponível em: <<https://www.gov.br/anac/pt-br/assuntos/regulados/empresas-aereas/simuladores-de-voo-fstd/definicoes-e-normas>>. Acesso em: 2 abr. 2024.

IBGE. Biblioteca. **[Aeronave EMB 120 Brasilia] : São José dos Campos, SP**. 2016. Disponível em: <<https://biblioteca.ibge.gov.br/index.php/bibliotecacatalogo?view=detalhes-&id=446142>>. Acesso em: 5 abr. 2024.

QUARTZO. **Simuladores de Voo e Operacionais**. 2024. Disponível em:

<<https://quartzodefense.com/detalhamento-simuladores-de-voo-e-operacionais/#:~:text=Em%20junho%20de%202018%2C%20havia,linhas%20a%C3%A9reas%20comerciais%20em%20servi%C3%A7o>>. Acesso em: 10 abr. 2024.

RTP, Corp. 2024. Disponível em: <<https://rtpcorp.com/about-rtp/>>. Acesso em: 11 abr. 2024.

TOYWING. Tudo sobre aeromodelo. **Superfície de Controle**. 2024. Disponível em:

<<https://toywing.com.br/superficie-de-controle/>>. Acesso em: 3 abr. 2024.

USINAINFO. **Arduino UNO Wifi Atmega328P com ESP8266 e CH340**. 2024. Disponível

em: <<https://www.usinainfo.com.br/placas-arduino/arduino-uno-wifi-atmega328p-com-esp8266-e-ch340-8301.html>>. Acesso em: 5 maio. 2024.

ATMEL. ATMEGA328P datasheet: 8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash. ATMEL, 2008. 26 p.

ESPRESSIF. ESP8266 datasheet: Technical Reference. ESPRESSIF, 2008. 117 p.

TEXAS INSTRUMENTS. ADS1115 datasheet: Ultrapequeno, baixo consume de energia, 16bits Conversor Analógico-Digital com Referência Interna. TEXAS INSTRUMENTS, 2009. 37 p.

ELETROGATE. **Módulo Conversor Analógico Digital ADC ADS1115**. 2024. Disponível

em: <<https://www.eletrogate.com/modulo-conversor-analogico-digital-adc-ads1115>>.

Acesso em: 10 maio. 2024.

MAKERHERO. **Conversor analógico digital ADS1115 e sensor LM35**. 2024. Disponível

em: <<https://www.makehero.com/blog/conversor-analogico-digital-ads1115-sensor-lm35/>>. Acesso em: 12 maio. 2024.

INSTITUTO DIGITAL. **Módulo Cartão SD Card – Para Arduino**. 2024. Disponível em: <

<https://www.institutodigital.com.br/produto/modulo-cartao-sd-card/>>. Acesso em: 17 maio. 2024.

AUTOCORE. **Display Lcd 16x2 Backlight Azul**. 2024. Disponível em:< <https://www.autocorerobotica.com.br/display-lcd-16x2-hd44780>>. Acesso em: 22 maio. 2024

A2ROBOTCS. **Arduino Mega R3 CH340 com Cabo USB**. 2024. Disponível em:< [https://www.a2robotics.com.br/arduino-mega-r3-ch340-com-cabo-usb?utm\\_source=Site&utm\\_medium=GoogleShopping&utm\\_campaign=IntegracaoGoogle&gad\\_source=1&gclid=CjwKCAiA3ZC6BhBaEiwAeqfvyvn0eqTmUHMxhGXu9Gm5PpRgY3lsK9BvV4qLGHoCy22o7Vf9kjjp5xoCv3UQAvD\\_BwE](https://www.a2robotics.com.br/arduino-mega-r3-ch340-com-cabo-usb?utm_source=Site&utm_medium=GoogleShopping&utm_campaign=IntegracaoGoogle&gad_source=1&gclid=CjwKCAiA3ZC6BhBaEiwAeqfvyvn0eqTmUHMxhGXu9Gm5PpRgY3lsK9BvV4qLGHoCy22o7Vf9kjjp5xoCv3UQAvD_BwE)>. Acesso em: 26 julho. 2024

STORE ARDUINO. **Arduino Mega 2560 REV3**. 2024. Disponível em:< <https://store.arduino.cc/products/arduino-mega-2560-rev3>>. Acesso em: 10 agosto. 2024

MAKERHERO. **CD4052 CI Multiplexador-Demultiplexador**. 2024. Disponível em:< <https://www.makehero.com/produto/cd4052-ci-multiplexador-demultiplexador/>>. Acesso em: 12 agosto. 2024

CASA DA ROBÓTICA. **Placa UNO R3 smd ATMEGA328 sem cabo**. 2024. Disponível em:< <https://www.casadarobotica.com/placas-embarcadas/arduino/placas/placa-uno-r3-smd-atmega328-sem-cabo>>. Acesso em: 12 agosto. 2024

MOUSER. **LM124**. 2024. Disponível em:< <https://br.mouser.com/ProductDetail/Texas-Instruments/LM124-MDR?qs=MXTMmq2PTRsc5Nr5oAmqPA%3D%3D>>. Acesso em: 22 agosto. 2024

MAKERHERO. **Diodo Zener 5.1V**. 2024. Disponível em:< <https://www.makehero.com/produto/diodo-zener-5-1v/>>. Acesso em: 12 setembro. 2024

LCDWIKI. **1.9 INCH IPS MODULE**. 2024. Disponível em:< [http://lcdwiki.com/1.9inch\\_IPS-Module](http://lcdwiki.com/1.9inch_IPS-Module)>. Acesso em: 1 outubro. 2024