

**GUILHERME MIYATA MEIRELLES**

**IMPLEMENTAÇÃO DE CINEMÁTICA E  
CONTROLE EM PLATAFORMA DE STEWART  
INTEGRADA A SIMULADOR DE VOO**

**São José dos Campos, 2025  
SP – Brasil**



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO CÂMPUS SÃO JOSÉ DOS CAMPOS**

**BACHARELADO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

**GUILHERME MIYATA MEIRELLES**

**IMPLEMENTAÇÃO DE CINEMÁTICA E  
CONTROLE EM PLATAFORMA DE STEWART  
INTEGRADA A SIMULADOR DE VOO**

Trabalho de Graduação apresentada ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, câmpus São José dos Campos como parte dos requisitos exigidos para conclusão do curso bacharelado em Engenharia de Controle e Automação.

Orientador: Anderson Kenji Hirata  
Coorientador: Carlos Eduardo Oliveira da Silva

**São José dos Campos, 2025  
SP – Brasil**

**Página destinada a ficha catalográfica confeccionada pela biblioteca do  
campus**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO**

**CAMPUS SÃO JOSÉ DOS CAMPOS**

**BACHARELADO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

**TRABALHO DE CONCLUSÃO DE CURSO**

**IMPLEMENTAÇÃO DE CINEMÁTICA E CONTROLE  
EM PLATAFORMA DE STEWART INTEGRADA A  
SIMULADOR DE VOO**

Autor: Guilherme Miyata Meirelles

Orientador: Anderson Kenji Hirata

Coorientador: Carlos Eduardo Oliveira da Silva

A Banca Examinadora composta pelos membros abaixo aprovou este Trabalho de Graduação:

---

**Prof. titulação e nome por extenso do docente orientador**  
**Instituto Federal Educação, Ciência e Tecnologia de São Paulo – Câmpus São José dos Campos**

---

**Prof. titulação e nome por extenso do docente coorientador**  
**Instituto Federal Educação, Ciência e Tecnologia de São Paulo – Câmpus São José dos Campos**

---

**Prof. titulação e nome por extenso do docente**  
**Instituto Federal Educação, Ciência e Tecnologia de São Paulo – Câmpus São José dos Campos**

---

**Prof. titulação e nome por extenso do docente**  
**Instituição/Cidade**

São José dos Campos, xx de xxx de 2025.

## **Dedicatória**

Dedico este trabalho à minha mãe e ao meu irmão, por nunca deixarem faltar apoio, mesmo nos momentos mais difíceis. Cada conquista que alcancei até aqui carrega um pouco de vocês e do que vocês representam na minha vida.

## Agradecimentos

Agradeço ao Instituto Federal de São Paulo – Câmpus São José dos Campos pela formação de excelência e pelo ambiente que ampliou minha visão acadêmica e profissional. Foi no IFSP que encontrei oportunidades reais de crescimento e professores dedicados que contribuíram de maneira decisiva para minha trajetória. Deixo este projeto como forma de retribuição, oferecendo uma bancada que poderá auxiliar futuras turmas, servir como ferramenta de ensino e apoiar novos estudantes que trilharão caminhos semelhantes ao meu.

Agradeço, em especial, ao meu orientador Anderson Kenji Hirata, que acompanhou este projeto de perto em todas as etapas. Sua dedicação ficou evidente em cada encontro semanal, nas orientações detalhadas, nas ideias que aprimoraram o trabalho e no apoio constante ao longo de todo o processo. Sou profundamente grato pela oportunidade de utilizar o laboratório mesmo durante suas aulas, pela disponibilidade em compartilhar artigos, materiais e referências, e pela paciência em esclarecer dúvidas nos momentos mais inesperados — inclusive nos finais de semana, quando não havia qualquer obrigação de fazê-lo. Sua orientação foi fundamental não apenas para este projeto, mas também para o meu desenvolvimento acadêmico e profissional.

Registro também minha sincera gratidão ao meu coorientador Carlos Eduardo Oliveira da Silva e ao professor Valdeci Donizete Gonçalves, que foram essenciais em um dos aspectos mais importantes para o andamento do trabalho: o acesso ao laboratório. Como alunos não podem utilizá-lo sem a presença de um responsável, só pude avançar porque esses professores se dispuseram a abrir o laboratório durante seus horários de aula e a se responsabilizar pela minha permanência no espaço. Esse apoio foi determinante para que eu conseguisse concluir o projeto dentro do prazo.

Agradeço ainda aos meus amigos que compartilharam esta trajetória comigo. Juntos enfrentamos inúmeros projetos, dúvidas, soluções, madrugadas de estudo e momentos de verdadeira parceria. A ajuda mútua, as conversas, os conselhos e o espírito de colaboração foram essenciais para superar desafios e tornar essa jornada mais leve e significativa. Sou grato por cada aprendizado dividido e por todas as conquistas construídas lado a lado.

Por fim, agradeço ao meu amigo Tharick Pontes Ferreira, cuja ajuda foi indispensável em diversas etapas deste trabalho. Suas sugestões, esclarecimentos de dúvidas e contribuições técnicas foram extremamente valiosas, tornando o processo mais fluido e produtivo.

*“A mente que se abre a uma nova ideia jamais volta ao seu tamanho original.”*

Oliver Wendell Holmes Sr., 1872

## Resumo

Este trabalho apresenta o desenvolvimento de um sistema de controle dedicado para uma plataforma de Stewart existente no Instituto Federal de São Paulo, campus São José dos Campos, com o objetivo de superar limitações técnicas da bancada original e ampliar seu potencial didático e experimental. A plataforma encontrada possuía ruído elevado nos sensores de posição, diferenças dinâmicas entre atuadores, restrições de hardware e impossibilidade de operação coordenada dos seis graus de liberdade. Para solucionar essas limitações, foram projetados e implementados um novo hardware controlador baseado em ESP32-S3, uma interface gráfica de operação em ambiente web e algoritmos de controle de posição dos atuadores.

Inicialmente, foi realizado o levantamento da estrutura mecânica e eletrônica da bancada, seguido da identificação das características dinâmicas dos pistões e do desenvolvimento de modelos de primeira ordem utilizados para o projeto de controladores PI individuais. Procedimentos de filtragem, compensação de zona morta e técnicas de *anti-windup* foram incorporados para garantir estabilidade e sincronismo entre os atuadores. Em paralelo, foram implementadas rotinas de cinemática direta e inversa, modos de operação manual e automático, e integração da plataforma com o simulador de voo FlightGear para replicação de trajetórias em seis graus de liberdade.

Os resultados experimentais demonstram que o novo sistema é capaz de operar a plataforma de forma estável, com reduzido erro estacionário e maior uniformidade entre as juntas, além de permitir experimentos avançados em controle, robótica paralela e integração com sistemas embarcados. O trabalho contribui para o uso acadêmico da bancada, oferecendo uma solução acessível, modular e compatível com abordagens modernas de ensino e pesquisa.

**Palavras-chave:** Plataforma de Stewart; Controle; Identificação de Sistemas; Robótica Paralela; ESP32-S3.

## Abstract

This work presents the development of a dedicated control system for an existing Stewart platform at the Federal Institute of São Paulo, São José dos Campos campus, aiming to overcome technical limitations of the original bench and expand its didactic and experimental potential. The initial platform exhibited high noise levels in the position sensors, dynamic discrepancies among actuators, hardware restrictions, and the inability to operate all six degrees of freedom in a coordinated manner. To address these limitations, a new control hardware based on the ESP32-S3 microcontroller was designed and implemented, along with a web-based graphical interface and position control algorithms for the actuators.

The development began with a detailed assessment of the mechanical and electronic structure of the bench, followed by the identification of the dynamic characteristics of the actuators and the creation of first-order models used to design individual PI controllers. Filtering procedures, dead-zone compensation, and anti-windup techniques were incorporated to ensure stability and synchronization among the actuators. In parallel, routines for forward and inverse kinematics, manual and automatic operation modes, and integration with the FlightGear flight simulator were implemented to replicate six-degree-of-freedom trajectories.

Experimental results demonstrate that the new system is capable of operating the platform stably, with reduced steady-state error and improved uniformity among the actuators, while enabling advanced experiments in control, parallel robotics, and embedded systems integration. The work contributes to the academic use of the bench by providing an accessible, modular solution aligned with modern teaching and research methodologies.

**Keywords:** Stewart platform; Control; System identification; Parallel robotics; ESP32-S3.

## Lista de Figuras

Figura 1 – Plataforma de Stewart IFSP .....	20
Figura 2: Estoque Operacional de Robôs Industriais - Mundo .....	24
Figura 3: Instalações Anuais de Robôs Industriais – Mundo .....	24
Figura 4: Robô Serial.....	25
Figura 5: Plataforma de Stewart .....	26
Figura 6: Sistema de Coordenadas de uma Plataforma de Stewart .....	28
Figura 7: Representação Vetorial da Cinemática da Plataforma de Stewart.....	29
Figura 8: Movimentos de Seis Graus de Liberdade em Embarcações .....	34
Figura 9: Exemplo de Plataforma Montada em Embarcação .....	34
Figura 10: Manipulador 6 DOF .....	35
Figura 11: Resposta Típica de um Sistema de Primeira Ordem a um Degrau .....	38
Figura 12: Comparação das curvas com diferentes $\zeta$ .....	39
Figura 13: Aproximação por modelo de 1 <sup>a</sup> ordem com tempo morto .....	40
Figura 14: Ilustração do Integrador de <i>Windup</i> .....	41
Figura 15: Comunicação API .....	45
Figura 16: Modelo de API REST .....	46
Figura 17: Bancada de plataforma de Stewart, disponível no IFSP-SJC .....	47
Figura 18: Estrutura Original da Bancada com CLP Siemens e Atuadores Lineares .....	48
Figura 19: ESP32-S3-DevKitC-1 Pin Layout .....	50
Figura 20: Estrutura Adaptada da Bancada com ESP32-S3 e Atuadores Lineares.....	57
Figura 21: Juntas de Kardan Reconstruídas .....	58
Figura 22: Funcionamento conversores NPN/PNP .....	59
Figura 23: Comando dos Motores .....	60
Figura 24: Arquitetura original com CLP e conversores NPN/PNP .....	60
Figura 25: Funcionamento conversores NPN/PNP .....	61
Figura 26: Arquitetura atual com ESP32-S3 e controle direto .....	61
Figura 27: Atuador Linear .....	62
Figura 28: Conector DB37 com fios numerados .....	63
Figura 29: PCB para Interface com ESP32S3 .....	63
Figura 30: Aquisição inicial - Ruídos.....	65
Figura 31: Resposta com filtro passa-baixa.....	65
Figura 32: Resposta com filtro média móvel (janela de 8).....	66
Figura 33: Resposta com Filtro Anti-Picos (mediana + limitador de inclinação) .....	67
Figura 34: Deslocamento dos pistões (mm) com aplicação simultânea de PWM.....	68
Figura 35: Correção de Offset .....	69
Figura 36: Interface do Aplicativo de Coleta de Dados .....	73
Figura 37: Dados Obtidos da Coleta Pistões 1-6.....	74
Figura 38: Coleta simultânea dos 6 pistões com $K_p = 5$ .....	75
Figura 39: Dados Importados para o System Identification .....	76
Figura 40: Configurações do Modelo .....	77
Figura 41: Resultado do Modelo Gerado Pistão 1 e 2 .....	77
Figura 42: Resultado do Modelo Gerado Pistão 3 e 4 .....	78
Figura 43: Resultado do Modelo Gerado Pistão 5 e 6 .....	78
Figura 44: Pistão 1, 2, 3 e 4 – Comparação entre as FTs e Erro ao longo do tempo.....	80
Figura 45: Pistão 5 e 6 – Comparação entre as FTs e Erro ao longo do tempo.....	81
Figura 46: Resultado Obtido – Curva 6 Pistões Sobrepostas para Comparação .....	84
Figura 47: Especificações do Design.....	84

Figura 48: Resultado Desejado.....	85
Figura 49: Resposta do ensaio .....	86
Figura 50: Resposta antes da sincronização – Visão geral.....	87
Figura 51: Resposta antes da sincronização – zoom Tr e Ts.....	87
Figura 52: Resposta após sincronização – Visão geral.....	88
Figura 53: Resposta ao Degrau Segunda Estratégia.....	90
Figura 54:Página Inicial .....	91
Figura 55: Fluxo de Dados .....	91
Figura 56: Gráfico Controle PID – Chart.js .....	93
Figura 57: Dados do Websocket.....	93
Figura 58: Controle de Setpoints .....	94
Figura 59: Painel de Ajuste de Ganhos PID .....	95
Figura 60: Fluxograma Funcionamento da Cinemática .....	97
Figura 61: Feedback de IMU – BNO085 ou MPU6050.....	97
Figura 62: Controles de Posição.....	98
Figura 63: Modelo 3D – Calculado e Real .....	98
Figura 64: Rotinas Disponíveis .....	100
Figura 65: Rotina Circulo XY .....	101
Figura 66: Rotina Senoide Vertical Z.....	101
Figura 67: Rotina <i>Heave Pitch</i> (Onda) .....	102
Figura 68: Rotina Senoide <i>Pitch</i> (Y) .....	102
Figura 69: Rotina Senoide <i>Roll</i> (X).....	103
Figura 70: Rotina Espiral.....	104
Figura 71: Valores do IMU .....	105
Figura 72: Pré-Visualização dos Estados .....	107
Figura 73: Controle com Posições Válidas .....	107
Figura 74: <i>Feedback</i> de Posição Inválida.....	108
Figura 75: Interface para a Simulação de Voo .....	109
Figura 76: Comprimento de Cada Pistão e Orientação <i>Roll e Pitch</i> .....	110
Figura 77: Modelo Customizado IFSP-SJC .....	111
Figura 78: .png Base para o FlightGear.....	111

## Lista de Tabelas

Tabela 1: Organização Original DB37 .....	64
Tabela 2: Organização DB37 Após Adaptação da Bancada .....	64
Tabela 3: Ganhos do Controlador PID para Cada Pistão .....	83
Tabela 4: Ganhos do Controlador PI para Cada Pistão .....	88
Tabela 5: Ganhos Revisados do Controlador PI para Cada Pistão .....	89
Tabela 6: Endpoints de Comunicação .....	91
Tabela 7: Endpoints de Controle PID .....	95
Tabela 8: Base (Furações F1–F6) .....	96
Tabela 9: Plataforma Superior (FS1–FS6) .....	96
Tabela 10: Endpoints Cinemática .....	99
Tabela 11: Endpoint Controle Joystick .....	106
Tabela 12: Endpoints Integração FlightGear .....	109

## **Lista de Abreviaturas, Nomenclaturas e Siglas**

<b>FT</b>	Função de Transferência
<b>Ts</b>	<i>Settling Time</i> - Tempo de Acomodação
<b>Tr</b>	<i>Rise Time</i> - Tempo de Subida
<b>FOPDT</b>	<i>First Order Plus Dead Time</i> - Modelo de 1 <sup>a</sup> ordem com tempo morto
<b>PID</b>	Proporcional–Integral–Derivativo
<b>IHM</b>	Interface Homem-Máquina
<b>HTML</b>	<i>HyperText Markup Language</i> - Linguagem de Marcação de Hipertexto
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>IMU</b>	<i>Inertial Measurement Unit</i>
<b>CSV</b>	<i>Comma-Separated Values</i>

## Sumário

<b>1. INTRODUÇÃO .....</b>	<b>18</b>
1.1. Contexto e Justificativa .....	19
1.2. Problema de Pesquisa .....	21
1.3. Objetivos .....	21
1.3.1. Objetivo Geral.....	21
1.3.2. Objetivos Específicos .....	21
1.4. Apresentação da Estrutura do Trabalho .....	22
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>24</b>
2.1. Robótica .....	24
2.2. Cinemática da Plataforma de Stewart .....	27
2.2.1. Cinemática Inversa.....	27
2.2.2. Cinemática Direta .....	30
2.3. Aplicações e Formas de Comando da Plataforma de Stewart.....	32
2.3.1. Rotinas de Movimento .....	32
2.3.2. Controle de Estabilidade.....	33
2.3.3. Controle por Dispositivos Manuais.....	35
2.4. Controle .....	36
2.4.1. Fenômeno do Windup em Sistemas com Saturação .....	41
2.5. Conceito de Aplicações Web e Arquitetura Cliente-Servidor.....	42
2.5.1. Arquitetura Cliente-Servidor .....	43
2.5.2. Protocolo HTTP .....	43
2.5.3. WebSocket .....	44
2.5.4. API e API REST .....	45
<b>3. MÉTODO E PROCEDIMENTOS METODOLÓGICOS.....</b>	<b>46</b>
3.1. Análise da Bancada Existente no IFSP .....	47
3.2. Adequação do Controlador da Bancada .....	48
3.2.1. Controlador ESP32-S3.....	49
3.3. Testes e Aquisições de Dados da Bancada .....	50
3.3.1. Procedimentos de Teste e Coleta de Sinais .....	50
3.3.2. Tratamento de Ruído e Filtragem .....	51

3.3.3.	Avaliação do Comportamento de Deslocamento.....	51
3.3.4.	Calibração Tensão–Posição (V0–V100) .....	51
3.3.5.	Identificação de Zona Morta e Não Linearidades.....	51
3.3.6.	Finalidade dos Dados.....	52
3.4.	Identificação de Modelo e projeto do Controlador para a malha fechada .....	52
3.4.1.	Estratégia Geral de Identificação .....	52
3.4.2.	Aquisição de Dados para Identificação.....	53
3.4.3.	Modelagem e Seleção da Estrutura.....	53
3.4.4.	Projeto dos Controladores.....	53
3.5.	Desenvolvimento da Interface de Controle e Modos de Operação .....	54
3.5.1.	Arquitetura de comunicação .....	54
3.5.2.	Integração com os microcontroladores .....	55
3.5.3.	Integração com FlightGear .....	55
3.5.4.	Modos de Operação Planejados.....	56
<b>4.</b>	<b>APRESENTAÇÃO E ANÁLISE DE RESULTADOS .....</b>	<b>57</b>
4.1.	Bancada de plataforma de Stewart após Adequações .....	57
4.2.	Adequação do controlador da bancada .....	58
4.2.1.	Substituição do CLP pelo ESP32-S3.....	58
4.2.2.	Retirada dos Optoacopladores e mudança na lógica de controle.....	59
4.2.3.	Reconfiguração dos Drivers de Potência .....	61
4.2.4.	Adaptação dos Sensores de Feedback .....	62
4.2.5.	Desenvolvimento de Placa de Circuito Impresso (PCB) de Interface .....	63
4.2.6.	Mapeamento dos Sinais do Conector DB37 .....	64
4.3.	Testes e Aquisições de Dados da Bancada .....	64
4.3.1.	Tratamento de Ruído e Filtragem .....	64
4.3.2.	análise de comportamento de deslocamento entre pistões.....	67
4.3.3.	Mapeamento da Tensão → Posição (V0–V100) .....	69
4.3.4.	Compensação de Zona Morta/Atrito .....	69
4.4.	Identificação de Modelo e projeto do Controlador para a malha fechada .....	71
4.4.1.	FASE 1 – Aquisição de Dados .....	71
4.4.2.	FASE 2 - Identificação de modelo .....	76
4.4.3.	FASE 3 - Sintonia do Controle PID.....	82

4.5.	Desenvolvimento da interface de controle e modos de operação .....	90
4.5.1.	Arquitetura de comunicação .....	90
4.5.1.1.	Integração com os microcontroladores.....	92
4.5.1.2.	Comunicação com FlightGear.....	92
4.5.2.	Implementação Painel PID.....	92
4.5.3.	Implementação da Cinemática .....	96
4.5.4.	Implementação das Rotinas de Movimento.....	99
4.5.5.	Implementação do Controle por IMU .....	104
4.5.6.	Implementação do Controle por Joystick .....	106
4.5.7.	Implementação do Simulador de Voo .....	108
<b>5.</b>	<b>CONCLUSÕES E CONSIDERAÇÕES FINAIS .....</b>	<b>112</b>
5.1.	Conclusões.....	112
5.2.	Considerações Finais.....	113
5.3.	Limitações da Pesquisa .....	114
5.3.1.	Limitações Mecânicas da Bancada .....	114
5.3.2.	Juntas de Kardan Impressas em 3D .....	114
5.3.3.	Ausência de Documentação dos Limites Mecânicos .....	115
5.3.3.1.	Impacto dos Limites Desconhecidos nas Funcionalidades do Software	115
5.4.	Limitações dos Sensores de Posição.....	115
5.4.1.	Limitações do Modelo Dinâmico e da Estrutura de Controle .....	115
5.4.1.1.	Modelo Simplificado e Ausência de Dinâmica Acoplada .....	115
5.4.1.2.	Influência da Gravidade e Assimetria entre Subida e Descida .....	116
5.4.1.3.	Controle PI em Sistema Não Linear e Acoplado .....	116
5.4.2.	Limitações nos Testes Experimentais .....	116
5.4.3.	Limitações na Integração com FlightGear .....	116
5.5.	Trabalhos Futuros .....	116
<b>ANEXO I – Arquitetura Geral.....</b>	<b>123</b>	
<b>ANEXO II – ESP32-S3 no Controle dos Pistões.....</b>	<b>123</b>	
<b>ANEXO III – Funcionamento Painel PID .....</b>	<b>123</b>	
<b>ANEXO IV – Configuração de Parâmetros .....</b>	<b>123</b>	
<b>ANEXO V – Funcionamento Cinemática.....</b>	<b>123</b>	
<b>ANEXO VI – Rotinas de Movimento.....</b>	<b>123</b>	

<b>ANEXO VII – Controle por IMU .....</b>	<b>123</b>
<b>ANEXO VIII – Integração com MPU6050 .....</b>	<b>123</b>
<b>ANEXO IX – Integração com BNO085 .....</b>	<b>123</b>
<b>ANEXO X – Controle Via Joystick.....</b>	<b>123</b>
<b>ANEXO XI – Simulação de Voo.....</b>	<b>124</b>

## 1. INTRODUÇÃO

A plataforma de Stewart é um mecanismo robótico paralelo capaz de gerar movimentos precisos em seis graus de liberdade, característica que a torna adequada para aplicações que exigem alta rigidez, precisão e desempenho dinâmico. Desde a proposta original de Stewart 1965, na qual o autor descreveu sua utilização em simuladores de voo, o mecanismo passou a ser amplamente adotado e estudado em diferentes áreas da engenharia. Em navios e embarcações, sistemas de compensação ativa de movimento em ambiente marítimo utilizam plataformas de Stewart para estabilizar estruturas sujeitas a perturbações multieixo causadas por ondas, ventos e correntes do oceano, demonstrando capacidade de rejeição dinâmica de movimentos (WANG et al., 2025). No setor aeroespacial, pesquisas de isolamento ativo de microvibrações para cargas ópticas espaciais evidenciam o uso da plataforma para garantir estabilidade de apontamento em sistemas sensíveis (DU et al., 2025). No campo da dinâmica e controle, estudos de vibrações paramétricas em plataformas eletro-hidráulicas reforçam seu valor como bancada experimental para investigar acoplamentos estruturais (YUAN et al., 2021). Além disso, sua versatilidade permite aplicações avançadas em robôs agrícolas, como no desenvolvimento de manipuladores para corte preciso de raízes de mandioca, onde a plataforma é empregada como unidade de movimentação de alta rigidez (SINGHPOO et al., 2024).

Projetos acadêmicos envolvendo sistemas como a plataforma de Stewart possuem grande relevância na formação em Engenharia de Controle e Automação, pois proporcionam ao aluno o contato direto com sistemas reais, complexos e não lineares. A aprendizagem baseada em experimentação prática melhora significativamente o engajamento e o domínio de conteúdos fundamentais dessa área, tais como modelagem e análise dinâmica de sistemas, e síntese de controladores. De acordo com Taylor, Eastwood e Jones (2013), plataformas didáticas dedicadas oferecem aos estudantes a oportunidade de praticar conceitos de sistemas controle, identificação de modelos matemáticos e *hardware-in-the-loop* de forma muito mais profunda e autônoma, permitindo compreender limitações reais de sensores, atuadores e modelos teóricos. Essa visão é reforçada por Rossiter et al. (2023), que destacam que os desafios contemporâneos da engenharia, exigem que estudantes sejam expostos a problemas reais e multidisciplinares desde o início do curso de graduação, permitindo que os estudantes desenvolvam habilidades práticas e visão sistêmica de engenharia, que são características indispensáveis para a atuação profissional.

Assim, um projeto envolvendo uma plataforma de Stewart permite ao aluno explorar na prática áreas normalmente abordadas de maneira teórica, como sistemas não lineares, robôs

paralelos, acoplamentos dinâmicos e técnicas modernas de controle multivariável, ampliando significativamente o alcance pedagógico em relação ao currículo tradicional, mais focado em manipuladores seriais e sistemas lineares.

### **1.1. CONTEXTO E JUSTIFICATIVA**

No contexto atual da Engenharia de Controle e Automação, observa-se uma demanda crescente por profissionais capazes de projetar, implementar e otimizar sistemas complexos, interconectados e sujeitos a condições dinâmicas de operação (ROSSITER et al., 2023). A robótica, como campo multidisciplinar que integra conhecimentos de engenharia mecânica, eletrônica, teoria de controle e computação, desempenha papel central nesse cenário, viabilizando soluções para desafios diversos.

No IFSP, Campus São José dos Campos, encontra-se uma plataforma de Stewart já construída, porém pouco utilizada em atividades acadêmicas devido às limitações técnicas que reduzem seu potencial didático e operacional. Trabalhos anteriores apontaram problemas como ruído elevado nos sensores de posição, diferenças de desempenho entre os drivers de potência dos atuadores, que resultam em velocidades desiguais, e a restrição no número de canais analógicos de controladores disponíveis no campus, o que inviabiliza o acionamento simultâneo das seis juntas.

Este trabalho apresenta o desenvolvimento de um sistema de controle dedicado para uma plataforma de Stewart já existente no Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus de São José dos Campos (IFSP-SJC). Dessa forma, nesse trabalho, a estrutura de plataforma de Stewart desenvolvida por Gonçalves (2023) será utilizada (Figura 1), no entanto, um novo conjunto de *hardware* controlador é proposto. No trabalho anterior, os comandos e controle do robô foram realizados por meio de um Controlador Lógico Programável (CLP) comercial, bem como a sua interface de operação, desenvolvida em uma Interface Homem-Máquina (IHM). Isso traz limitações no seu uso e modificações, por se tratar de dispositivos que precisam de software específico com licença comercial, além de restringir a exploração de abordagens didáticas e pesquisas sobre a implementação de novos algoritmos ou técnicas de controle à linguagem de programação desse equipamento.

Além disso, em um trabalho anterior desenvolvido no âmbito da disciplina Laboratório de Sistemas de Controle II, no qual foi realizado o controle individual de um pistão da plataforma Stewart (Rabello et al., 2024), foram identificados problemas relevantes, tais como ruído elevado nos sensores internos de posição, diferenças entre os *drivers* de potência dos atuadores, que resultavam em velocidades desiguais entre as juntas, e a limitação de apenas

duas saídas analógicas no módulo NI-USB-6212, o que inviabilizava o acionamento simultâneo das seis juntas do sistema robótico.



Figura 1 – Plataforma de Stewart IFSP  
Fonte: GONÇALVES (2023)

Com o exposto, é esperado que o presente trabalho contribua para a disponibilização da bancada didática de plataforma de Stewart com um novo *hardware* controlador, além de uma interface de operação intuitiva desenvolvida usando linguagens de código aberto, possibilitando a sua aplicabilidade em atividades de ensino e pesquisa no curso de Engenharia de Controle e Automação, para acesso e uso por alunos e professores da comunidade acadêmica do IFSP-SJC.

O projeto propõe superar as restrições atuais da bancada por meio do desenvolvimento de um sistema de controle dedicado, composto por *hardware* aberto e uma interface gráfica intuitiva para operação e monitoramento que sejam compatíveis com demandas modernas, envolvendo a comunicação em rede e uso de protocolos baseados em internet. A proposta integra hardware e software em uma arquitetura unificada para uso de um sistema robótico, oferecendo meios para a realização de experimentos com cinemática direta e inversa, simulação e seguimento de trajetórias. Como exemplo de aplicação, além do controle da plataforma de Stewart, o trabalho propõe seu uso no setor aéreo, no qual ela será usada para posicionar um modelo de aeronave seguindo a trajetória de um simulador de voo que será integrado à aplicação desenvolvida. Dessa forma, a proposta alinha-se à compreensão de que bancadas didáticas constituem recursos fundamentais para a formação de engenheiros, pois permitem aplicar de

forma concreta conceitos de modelagem, controle e instrumentação, ao mesmo tempo em que expõem os estudantes a desafios técnicos reais (PINHO et al., 2021).

## **1.2. PROBLEMA DE PESQUISA**

Dante das limitações identificadas e da necessidade de aprimorar sua capacidade de operação coordenada e precisa, surge a questão central deste trabalho: Como projetar e implementar um sistema de controle, incluindo hardware dedicado e interface gráfica, capaz de operar a plataforma de Stewart do IFSP-SJC com precisão, estabilidade e coordenação dos seis graus de liberdade, superando as limitações técnicas atuais e ampliando seu potencial didático e experimental?

## **1.3. OBJETIVOS**

A seguir, são apresentados os objetivos desse trabalho. Numa visão geral, a principal contribuição pretendida com este trabalho é apresentada em Objetivo Geral. Na sequência, os passos seguidos para o desenvolvimento do trabalho tiveram como métrica os pontos destacados em Objetivos Específicos.

### **1.3.1. OBJETIVO GERAL**

O objetivo deste trabalho foi desenvolver um sistema de controle dedicado para a bancada de plataforma de Stewart, no qual deverá contar com a implementação de um hardware controlador, e uma interface gráfica para operação e monitoramento.

### **1.3.2. OBJETIVOS ESPECÍFICOS**

Em decorrência do objetivo geral proposto foram estabelecidos os seguintes objetivos específicos do trabalho:

- a) Levantamento e análise da plataforma existente (mecânica, sensores, atuadores, drivers, fonte de potência, fixações, juntas e curso útil), identificando suas limitações.
- b) Implementar hardware para controle especificando uma placa eletrônica com controlador embarcado que faça interface com os outros componentes de hardware já existentes, e que permita expansão futura.
- c) Avaliar e implementar algoritmos de controle para funcionamento dos atuadores do robô, baseando-se na sua modelagem e simulação.
- d) Desenvolver a interface gráfica para operação local, incluindo: modos de comando (atuação de cada junta individualmente, e da referência para a plataforma usando cinemática inversa), visualização de posição das juntas, e da bancada para monitoramento.

- e) Validar o funcionamento do sistema com testes práticos e simulações comparativas com o modelo, ou com sensores implementados na bancada.

#### **1.4. APRESENTAÇÃO DA ESTRUTURA DO TRABALHO**

O trabalho está organizado em cinco capítulos, distribuídos de forma a conduzir o leitor desde os fundamentos necessários até o desenvolvimento completo da solução proposta e a análise dos resultados obtidos.

O Capítulo 2 — Fundamentação Teórica apresenta os conceitos essenciais para compreender o projeto, iniciando com uma introdução à robótica e avançando para a descrição da plataforma de Stewart, sua cinemática direta e inversa, e as diferentes formas de comando do manipulador. São discutidos também princípios de controle, incluindo o fenômeno do *windup* em sistemas com saturação e técnicas de anti-*windup*, além de fundamentos sobre aplicações web, arquitetura cliente–servidor, protocolos de comunicação (HTTP e WebSocket) e tecnologias utilizadas no desenvolvimento da interface de operação.

O Capítulo 3 — Método e Procedimentos Metodológicos descreve detalhadamente todas as etapas de desenvolvimento adotadas no trabalho. São apresentados o diagnóstico da bancada experimental existente, a adequação da estrutura mecânica e eletrônica, os procedimentos de identificação de modelo e projeto dos controladores, bem como as etapas de aquisição de dados, filtragem de sinais, identificação de zona morta e não linearidades. Este capítulo também abrange o desenvolvimento da arquitetura de comunicação, da interface de controle e monitoramento, e a integração do sistema com microcontroladores e com o simulador de voo FlightGear.

O Capítulo 4 — Apresentação e Análise de Resultados consolida as melhorias realizadas na bancada e os resultados obtidos com o novo sistema de controle. São apresentados o processo de substituição do CLP pelo ESP32-S3, a reconfiguração dos drivers de potência, a adaptação dos sensores, a calibração completa do sistema e os testes experimentais conduzidos. São avaliadas as respostas dos atuadores, a cinemática implementada, as rotinas de movimento, o controle manual e automático, além da integração com IMU, joystick e com o FlightGear. O capítulo também discute as limitações observadas, o impacto das não linearidades, a influência da zona morta e as diferenças dinâmicas entre atuadores.

O Capítulo 5 — Conclusões e Considerações Finais reúne as principais conclusões do trabalho, destacando as contribuições alcançadas, as limitações identificadas e sugestões para

trabalhos futuros, incluindo melhorias em hardware, algoritmos de controle, integração com simuladores e expansão das aplicações da bancada em atividades didáticas e de pesquisa.

Ao final do documento, são apresentadas as referências utilizadas, seguidas de anexos e apêndices que complementam o material desenvolvido.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1. ROBÓTICA

A robótica tem se consolidado como uma das áreas mais dinâmicas e estratégicas da engenharia, impulsionada pelos avanços em sensores, atuadores, computação embarcada e algoritmos de controle e inteligência artificial. Dados recentes da *International Federation of Robotics* (IFR) indicam um recorde de 4,66 milhões de robôs operando em fábricas no mundo em 2024 e instalações anuais consistentemente acima de meio milhão de unidades desde 2021, sinalizando um cenário de automação em expansão presente em setores tão diversos quanto a indústria, a medicina, a exploração espacial e o entretenimento. Nesse contexto, a robótica desempenha papel essencial na transformação de processos, aumentando a precisão, a eficiência e a segurança das operações (CRAIG, 2012; IFR, 2025).

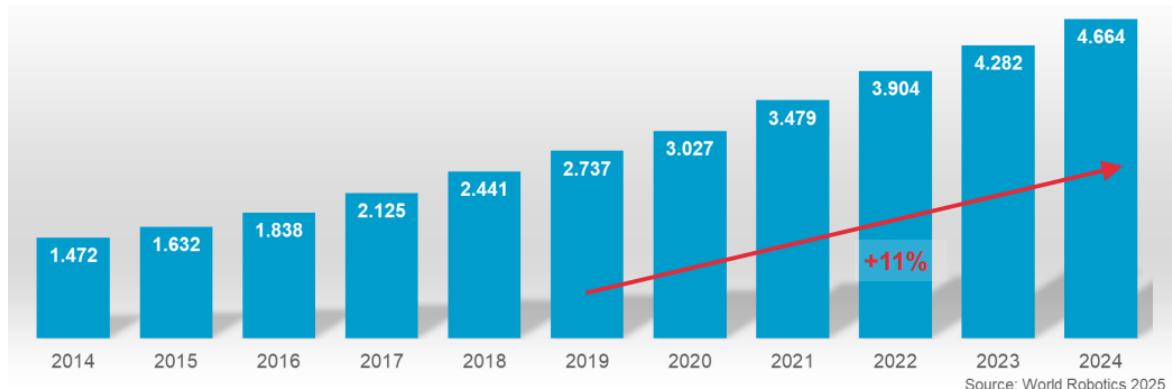


Figura 2: Estoque Operacional de Robôs Industriais - Mundo  
Fonte: INTERNATIONAL FEDERATION OF ROBOTICS (2025)

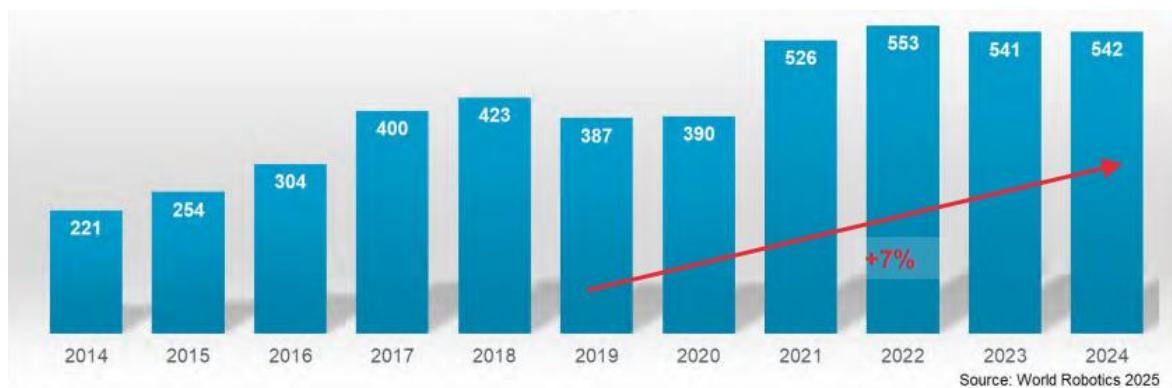


Figura 3: Instalações Anuais de Robôs Industriais – Mundo  
Fonte: INTERNATIONAL FEDERATION OF ROBOTICS (2025)

Os sistemas robóticos podem ser classificados de diversas formas, por exemplo separando-os por campo de aplicação, ambiente de operação, ou características construtivas.

Nesse último caso, é comum a sua distinção entre robôs seriais e robôs paralelos. Os manipuladores seriais (Figura 4), também chamados de cadeia cinemática aberta, são compostos por elos e juntas dispostos em sequência, sendo que os cálculos de sua cinemática envolvem a descrição de transformações de sistemas de referência partindo da base (que é fixa) até a sua ferramenta. Uma das configurações mais comuns encontradas na indústria, é o manipulador robótico articulado, com seis graus de liberdade que se assemelha à movimentação realizada pelo braço humano, assim. Esses robôs são amplamente utilizados em linhas de produção em tarefas como soldagem, pintura e montagem, apresentando como principais vantagens o grande alcance e a flexibilidade, mas sofrem com menor rigidez estrutural.

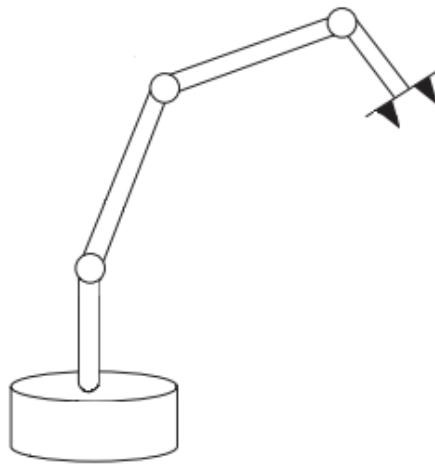


Figura 4: Robô Serial  
Fonte: CRAIG (2012)

Já os robôs paralelos possuem múltiplas cadeias cinemáticas fechadas que conectam a base fixa à plataforma móvel, o que proporciona maior rigidez, precisão e capacidade de suportar cargas elevadas, ainda que com volume de trabalho reduzido e análise cinemática mais complexa. Essa maior complexidade decorre das restrições geométricas não lineares impostas pelas cadeias em laço fechado, o que torna especialmente difícil a solução da cinemática direta, que em muitos casos não admite forma analítica fechada e exige métodos numéricos (CRAIG, 2012; WANG et al., 2017; STEWART, 1965). Essa distinção é fundamental, pois influencia diretamente o desempenho, as estratégias de controle e as possíveis aplicações de cada tipo de robô.

Entre os robôs paralelos, destaca-se a plataforma de Stewart, proposta originalmente para aplicações em simuladores de voo (STEWART, 1965). O mecanismo é composto por uma base fixa e uma plataforma móvel, que são interligadas por seis atuadores prismáticos, configurando um sistema com seis graus de liberdade (6 DOF – *Degree of Freedom*): três translacionais e

três rotacionais (Figura 5). Sua versatilidade foi demonstrada ao longo das décadas, com aplicações em simulação de veículos e aeronaves, ensaios estruturais, compensação de movimento em ambientes marítimos, dispositivos médicos de alta precisão, micromanipulação e pesquisas acadêmicas voltadas à cinemática e ao controle (SILVA; GARRIDO; RIVEIRO, 2022).

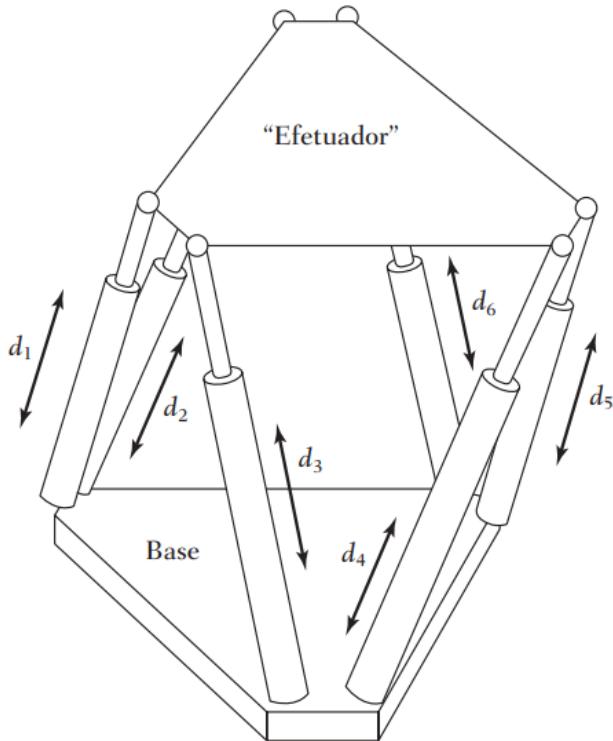


Figura 5: Plataforma de Stewart

Fonte: CRAIG (2012)

O projeto envolvendo a plataforma de Stewart, com o objetivo de descrever como se dá o posicionamento da plataforma móvel, demanda o domínio sobre os conceitos da sua cinemática direta e inversa, que por sua vez envolve a relação entre os deslocamentos das juntas e posição e orientação final atingida pelo robô. Além dos cálculos envolvidos, é necessário realizar o uso de controladores para lidar com o acoplamento dinâmico dos atuadores e a sua estrutura, para atender os requisitos de operação. Fatores como sensibilidade a ruídos dos sinais de sensores empregados, e incertezas geométricas, podem aumentar o nível de complexidade do projeto.

Conforme destaca Craig (2012), os métodos de controle linear, como os baseados em controladores PID (Proporcional, Integral e Derivativo), devem ser entendidos como aproximações aplicadas a sistemas essencialmente não lineares, como os manipuladores robóticos e plataformas paralelas. Apesar dessa limitação teórica, tais métodos continuam sendo amplamente utilizados na prática industrial e em aplicações experimentais, devido à sua

simplicidade de implementação, baixo custo computacional e desempenho satisfatório dentro das faixas operacionais lineares.

## **2.2. CINEMÁTICA DA PLATAFORMA DE STEWART**

A análise cinemática é um dos elementos centrais para o estudo e o controle de robôs, pois relaciona a posição das juntas, como elemento de atuação, para descrever a posição e orientação da ferramenta. A plataforma de Stewart consiste em uma base fixada ao chão, e uma plataforma móvel na qual alguma estrutura ou efetuador é acoplado. As juntas paralelas são atuadores prismáticos, que são conectados em suas extremidades, tanto na base quanto na plataforma móvel, usando-se juntas esféricas (ou tipo Cardan, nesse trabalho), formando um sistema robótico de cadeia fechada. Dessa forma, a relação entre posição/orientação da plataforma e os comprimentos dos atuadores não é trivial e envolve restrições geométricas acopladas. Ainda dentro desse contexto, dois estudos envolvendo a cinemática de sistemas robóticos precisam ser consideradas, a cinemática direta e a cinemática inversa, sendo ambas necessárias para o projeto desse tipo de robô.

### **2.2.1. CINEMÁTICA INVERSA**

A cinemática inversa consiste em determinar os comprimentos dos seis atuadores a partir de uma pose desejada composta por três translações e três rotações, em relação a um sistema de coordenadas de três eixos (X, Y e Z). A Figura 6 ilustra os referenciais adotados para a plataforma de Stewart, evidenciando os sistemas de coordenadas da base  $\{B\}$  e da plataforma móvel  $\{P\}$ . Este é o problema mais direto do ponto de vista computacional: aplicam-se transformações homogêneas (isto é, envolvendo juntas a rotação e translação) aos pontos de fixação da plataforma e, em seguida, calcula-se a distância entre esses pontos transformados e seus correspondentes na base. Assim, cada atuador tem seu comprimento obtido de forma independente, desde que a pose desejada esteja dentro da região de operação e não viole limitações mecânicas do sistema (CRAIG, 2012; DA SILVA, 2025)

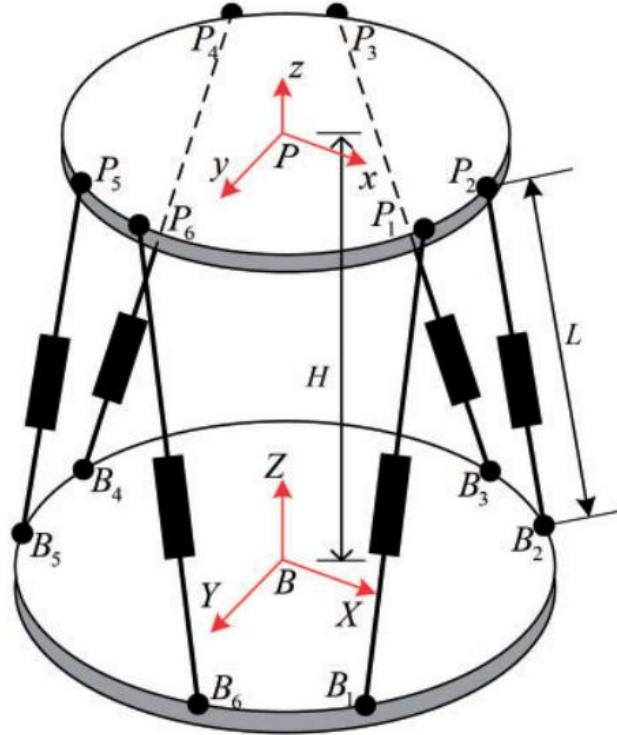


Figura 6: Sistema de Coordenadas de uma Plataforma de Stewart  
Fonte: TANG et al. (2021)

O posicionamento da plataforma móvel é definido pela sua pose, isto é, a combinação entre a posição do ponto central no espaço e a orientação da plataforma em relação ao sistema de coordenadas da base. A posição é representada pelo vetor

$$\mathbf{p} = [x \ y \ z]^T, \quad (1)$$

enquanto a orientação é descrita pelas rotações em torno dos três eixos, conhecidas como *roll* ( $\theta_x$ ), *pitch* ( $\theta_y$ ) e *yaw* ( $\theta_z$ ).

Para determinar a pose final da plataforma, os pontos de fixação superiores  $P_i$ , originalmente definidos no referencial da plataforma  $\{P\}$ , são transformados para o referencial da base  $\{B\}$  utilizando o vetor de posição  $p$  e matriz de rotação composta  $R(\theta_x, \theta_y, \theta_z)$ . Esta matriz é obtida pela composição das rotações elementares em torno dos eixos cartesianos X, Y e Z, dadas por:

$$R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{pmatrix} \quad (2)$$

$$R_y(\theta_y) = \begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{pmatrix} \quad (3)$$

$$R_z(\theta_z) = \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4)$$

A matriz de rotação total, considerando a sequência X–Y–Z, é dada por:

$$R(\theta_x, \theta_y, \theta_z) = R_z(\theta_z) R_y(\theta_y) R_x(\theta_x) \quad (5)$$

Assim, a transformação completa do ponto é dada por:

$$p_i^B = p + R(\theta_x, \theta_y, \theta_z) P_i \quad (6)$$

Esse procedimento torna a cinemática inversa um problema computacionalmente direto: cada atuador tem seu comprimento determinado de forma independente, desde que a pose desejada esteja dentro da região de operação e não viole limitações geométricas ou mecânicas do sistema.

Para descrever matematicamente o comportamento da plataforma de Stewart, utilizam-se dois referenciais principais (Figura 7):

- Referencial de base  $\{B\}$
- Referencial da plataforma móvel  $\{P\}$

Cada pistão  $i$  conecta um ponto fixo na base, descrito por  $\vec{a}_i \in \{B\}$  a um ponto fixo na plataforma, descrito por  $\vec{b}_i \in \{P\}$ .

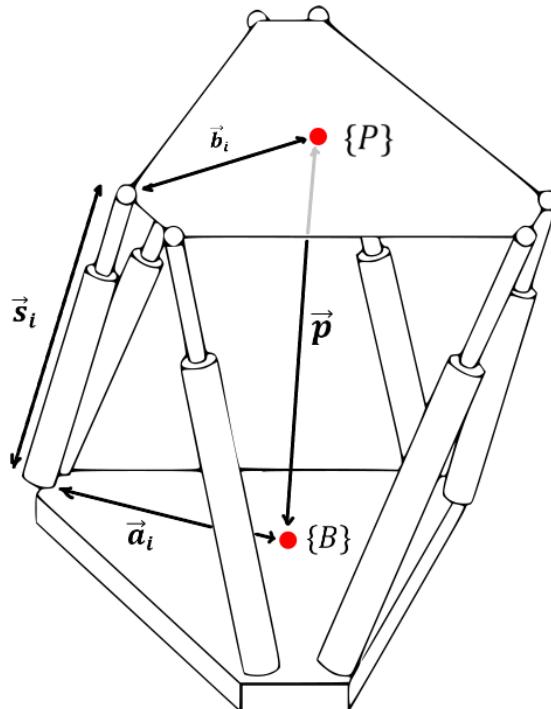


Figura 7: Representação Vetorial da Cinemática da Plataforma de Stewart  
Fonte: CRAIG (2012)

A pose da plataforma é representada pelo par  $(\vec{p}, R)$ , em que  $\vec{p}$  é a posição do centro da plataforma em relação a  $\{B\}$  e  $R$  é a matriz de rotação,  $R(\theta_x, \theta_y, \theta_z)$ , que transforma vetores do referencial  $\{P\}$  para o referencial  $\{B\}$ .

Com essa transformação, cada ponto superior  $\vec{b}_i$  passa a ser expresso no referencial da base da seguinte forma:

$$\vec{P}_i = \vec{p} + R\vec{b}_i \quad (7)$$

Esse ponto  $\vec{p}_i$  representa a coordenada, no sistema  $\{B\}$ , da junta superior do atuador  $i$ . Assim, o vetor que descreve o atuador é:

$$\vec{s}_i = \vec{p} + R\vec{b}_i - \vec{a}_i \quad (8)$$

e seu comprimento é simplesmente a norma desse vetor:

$$L_i = \| \vec{s}_i \| \quad (9)$$

Embora  $L_i$  represente o comprimento geométrico entre os pontos de fixação  $\vec{a}_i$  e  $\vec{p}_i$ , é importante esclarecer seu significado físico no contexto do atuador real. Em um atuador prismático, existe um comprimento mínimo (offset mecânico), um comprimento máximo e um curso útil. Assim, o valor calculado  $L_i$  não corresponde diretamente ao deslocamento do pistão, mas ao comprimento total do elo entre suas juntas.

O deslocamento efetivo do atuador é obtido por:

$$\Delta_i = L_i - L_{i,\min} \quad (10)$$

onde:

- $L_i$  é o comprimento geométrico instantâneo;
- $L_{i,\min}$  é o comprimento do atuador completamente retraído;
- $\Delta_i$  é o comando real aplicado ao atuador.

Essa distinção é fundamental para a implementação prática do controle, pois o modelo matemático usa  $L_i$ , enquanto o controlador do atuador utiliza  $\Delta_i$ .

### 2.2.2. CINEMÁTICA DIRETA

A cinemática direta da plataforma de Stewart consiste em determinar a pose da plataforma móvel, definida pelo vetor de posição  $\mathbf{p} = [x \ y \ z]^T$  e pela matriz de rotação  $R(\phi, \theta, \psi)$ , a partir dos comprimentos conhecidos dos seis atuadores  $L_i$ . Essa formulação é mais complexa que a da cinemática inversa, uma vez que as incógnitas aparecem acopladas dentro de equações não lineares envolvendo normas de vetores e produtos matriciais (SILVA; GARRIDO; RIVEIRO, 2022).

Assim como na cinemática inversa, cada atuador liga um ponto fixo da base  $a_i \in \{B\}$  a um ponto da plataforma móvel  $b_i \in \{P\}$ . Quando a plataforma assume a pose  $(p, R)$ , o ponto da plataforma expresso no referencial da base é dado por:

$$\vec{P}_i = \vec{p} + R\vec{b}_i \quad (11)$$

Assim, o vetor correspondente ao atuador  $i$ , agora conhecido o comprimento medido, é:

$$\vec{s}_i = \vec{P}_i - \vec{a}_i = \vec{p} + R\vec{b}_i - \vec{a}_i \quad (12)$$

E o comprimento medido do atuador é:

$$L_i = \| \vec{s}_i \| \quad (13)$$

Portanto, a cinemática direta impõe que, para a pose correta  $(p, R)$ , cada comprimento previsto pela geometria seja igual ao comprimento medido:

$$\| \vec{p} + R\vec{b}_i - \vec{a}_i \| = L_i, i = 1, \dots, 6 \quad (14)$$

Este conjunto de seis equações não lineares, envolvendo raízes, produtos cruzados e combinações de seno e cosseno advindos de  $R(\phi, \theta, \psi)$ , é altamente acoplado e, conforme demonstrado por Merlet e Gosselin (2008), pode admitir múltiplas soluções. Os autores mostram que mecanismos paralelos planares do tipo 3-RPR podem apresentar até 6 soluções reais, enquanto a plataforma de Stewart-Gough de geometria geral pode apresentar até 40 soluções.

Além da multiplicidade, muitas dessas soluções podem não ser realizáveis fisicamente por limitação de posicionamento dos atuadores, envolver alguma sobreposição ou cruzamento dos elementos estruturais da plataforma, ou mesmo resultar em condições de insta. Isso torna inviável a obtenção de soluções analíticas gerais, que só existem para geometrias muito específicas da plataforma.

Na prática, a solução é obtida reescrevendo o sistema acima como um problema de minimização:

$$\min_{p, \phi, \theta, \psi} \sum_{i=1}^6 (\| p + R(\phi, \theta, \psi)b_i - a_i \| - L_i)^2 \quad (15)$$

Ou seja, busca-se a pose que faz com que os comprimentos previstos coincidam com os medidos, em um sentido de mínimos quadrados. Merlet e Gosselin (2008) indicam que esse problema pode ser resolvido por métodos numéricos iterativos, como Newton–Raphson, que ajustam a pose estimada até que o erro entre comprimentos previstos e medidos seja minimizado.

Outro aspecto crítico da cinemática direta é a presença de singularidades geométricas, regiões em que a plataforma perde rigidez ou graus de liberdade efetivos. Nessas configurações, pequenas variações nos comprimentos dos atuadores podem causar grandes deslocamentos da plataforma, ou o contrário, resultando em amplificação de erro e perda de capacidade de controle. Reconhecer e evitar tais regiões é essencial para garantir segurança, estabilidade e precisão no movimento da plataforma.

Essas características tornam a cinemática direta um dos maiores desafios em manipuladores paralelos. Por isso, este trabalho adota um método numérico robusto baseado em mínimos quadrados para estimar, em tempo real, a pose da plataforma a partir da telemetria dos seis atuadores.

Além das abordagens clássicas descritas por Merlet e Gosselin, métodos mais recentes incluem soluções baseadas em teoria dos parafusos, análises geométricas, coordenadas Soma, sensores adicionais, redes neurais e regressão vetorial, embora métodos iterativos como Newton–Raphson ainda sejam predominantes pela sua eficiência computacional (SILVA; GARRIDO; RIVEIRO, 2022).

### **2.3. APLICAÇÕES E FORMAS DE COMANDO DA PLATAFORMA DE STEWART**

Diversas áreas utilizam plataformas de Stewart devido a essas características. Stewart (1965) destacou aplicações como testes dinâmicos, simulação de voo, compensação de movimentos de navios e automação industrial.

#### **2.3.1. ROTINAS DE MOVIMENTO**

As plataformas de Stewart são amplamente aplicadas em geração de movimentos controlados, no qual o objetivo é reproduzir trajetórias específicas em seis graus de liberdade. Essas rotinas de movimento são utilizadas tanto em ambiente acadêmico quanto industrial, principalmente para simulação de movimentos complexos, como acelerações e vibrações, testes dinâmicos submetendo componentes mecânicos ou eletrônicos a perfis de excitação conhecidos, caracterização e validação de controladores por meio de trajetórias repetitivas e reproduzíveis, e avaliação de desempenho mecânico, observando acoplamentos entre eixos, limites de curso e resposta do sistema (YUAN, 2021). A literatura demonstra que plataformas paralelas são especialmente adequadas para seguir trajetórias definidas com precisão, como mostrado nos experimentos de planejamento e controle de trajetória conduzidos por Nguyen et al. (1992), que evidenciam a capacidade da plataforma em executar movimentos complexos e repetitivos em ambiente de montagem.

Em aplicações industriais e científicas, essas rotinas costumam ser executadas com base em perfis de trajetória pré-definidos, como movimentos senoidais, varreduras harmônicas, trajetórias circulares, rampas, degraus, formas de onda compostas ou sinais pseudoaleatórios (SILVA; GARRIDO; RIVEIRO, 2022). Estudos recentes reforçam esse uso, demonstrando que plataformas de Stewart são submetidas a excitações senoidais e excitações aleatórias para análise dinâmica, identificação modal e avaliação de resposta estrutural. No trabalho realizado por Yuan et al. (2021), por exemplo, investigam o comportamento da plataforma sob excitações periódicas e paramétricas, mostrando como diferentes frequências influenciam a resposta dinâmica e os modos naturais do sistema. Da mesma forma, pesquisas voltadas ao isolamento de vibrações destacam o uso da plataforma como base para aplicar sinais harmônicos e ruído estruturado a fim de avaliar o desempenho em condições de vibração controlada.

Rotinas desse tipo também aparecem em plataformas utilizadas para simulação de voo, direção e navegação marítima, onde o movimento reproduz o comportamento de um modelo matemático do veículo ou o registro real de movimentos medidos (SILVA; GARRIDO; RIVEIRO, 2022). Documentos técnicos da NASA já descreviam, ainda na década de 1990, o uso de plataformas Stewart para simulação de voo e testes de manobrabilidade, destacando a capacidade de reproduzir movimentos suaves, repetíveis e sincronizados com modelos dinâmicos (NGUYEN; ANTRAZI, 1991).

Para esse tipo de aplicação, simuladores profissionais utilizam técnicas de *motion cueing*, que transformam as acelerações e orientações do veículo virtual em comandos fisicamente realizáveis pela plataforma. Dentro desse processo, um conceito fundamental é o *washout*, responsável por trazer a plataforma de volta ao centro do curso de forma lenta e imperceptível, evitando que os atuadores atinjam seus limites durante simulações prolongadas. O relatório de Casas, Olanda e Dey (2017) destaca que esse retorno quase invisível é essencial para manter a sensação de aceleração inicial sem comprometer a segurança mecânica da estrutura. De forma complementar, Chen e Fu (2010) explicam que o *motion cueing* busca fornecer ao usuário sensações coerentes de movimento ao mesmo tempo em que a plataforma opera dentro de sua área física limitada, explorando características do sistema vestibular humano para maximizar orealismo perceptivo.

### **2.3.2. CONTROLE DE ESTABILIDADE**

Além das aplicações de movimento, testes dinâmicos e automação, as plataformas também são amplamente empregadas em sistemas de estabilização, nos quais o objetivo principal não é gerar grandes deslocamentos, mas manter uma posição e orientação

praticamente constantes mesmo quando a base está sujeita a perturbações externas. Esse tipo de aplicação é particularmente relevante em ambientes marítimos, onde navios e estruturas *offshore* (como plataformas de extração de petróleo) sofrem movimentos de seis graus de liberdade (*surge*, *sway*, *heave*, *roll*, *pitch* e *yaw*) devido à ação de ondas, vento e correntes (CHEN et al., 2025). . Esses movimentos podem comprometer a precisão de operações, a segurança de pessoal e a qualidade de medições e processos.

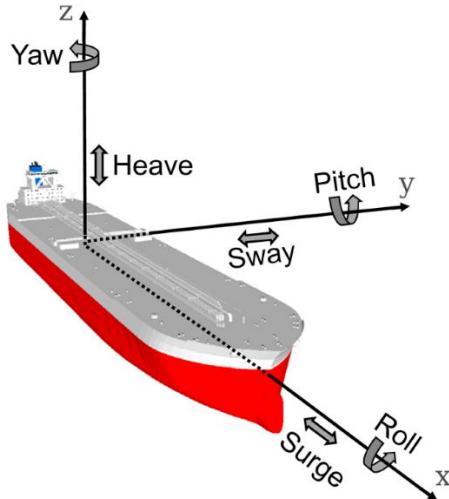


Figura 8: Movimentos de Seis Graus de Liberdade em Embarcações  
Fonte: ALVARELLOS (2021)

Nesses cenários, a plataforma de Stewart pode ser utilizada como uma plataforma de compensação de ondas, ajustando continuamente o comprimento de seus atuadores para gerar um movimento que compensa as variações de posição e atitude da base, criando um ambiente mais estável para equipamentos e operadores. Esse conceito é aplicado, por exemplo, em *helidecks* ativos, sistemas de transferência de carga entre embarcações, bases estabilizadas para sensores e instrumentos de alta precisão, bem como em plataformas de teste montadas sobre navios (WANG et al., 2025).

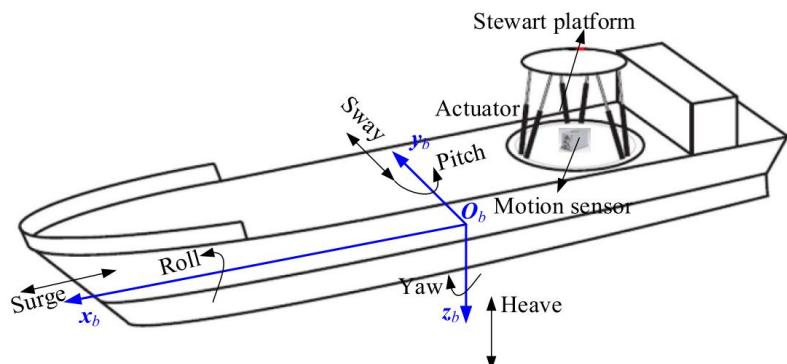


Figura 9: Exemplo de Plataforma Montada em Embarcação  
Fonte: CAI (2020)

Como exemplo desse tipo de aplicação, Chen et al. (2025) apresentam uma plataforma de Stewart eletro-hidráulica empregada especificamente para compensação de ondas em ambiente marítimo. Nesse trabalho, a plataforma é controlada de forma a atenuar em tempo real os movimentos induzidos pelo mar, reduzindo os erros residuais de posição e atitude e garantindo uma região de trabalho mais estável para operações offshore. Os resultados experimentais reportados pelos autores evidenciam que, quando adequadamente controlada, a plataforma de Stewart é capaz de melhorar significativamente a estabilidade do sistema, mesmo sob condições de mar agitadas, o que reforça seu potencial como solução para aplicações que requerem controle de estabilidade em seis graus de liberdade.

### 2.3.3. CONTROLE POR DISPOSITIVOS MANUAIS

Em aplicações de teleoperação, a referência de posição para movimentação da plataforma se dá por meio de dispositivos manuais, no qual o operador interage com a plataforma em tempo real. Esse tipo de controle é amplamente utilizado em simuladores de voo e direção, por meio de manche, volante ou pedais, permitindo que um operador humano controle a orientação e posição de plataformas robóticas, permitindo a exploração interativa com os mecanismos da plataforma, que por sua vez, possibilita a avaliação subjetiva de movimento. Mecanismos baseados em plataformas de Stewart podem funcionar como uma interface de controle em 6 DOF, permitindo que o usuário controle sistemas complexos de forma contínua e intuitiva. Yildiz (2016), por exemplo, desenvolveu um controle baseado em uma plataforma de Stewart para manipular um quadrotor, demonstrando que a plataforma pode atuar como uma interface manual de alta precisão para controlar veículos espaciais ou sistemas dinâmicos multidimensionais.

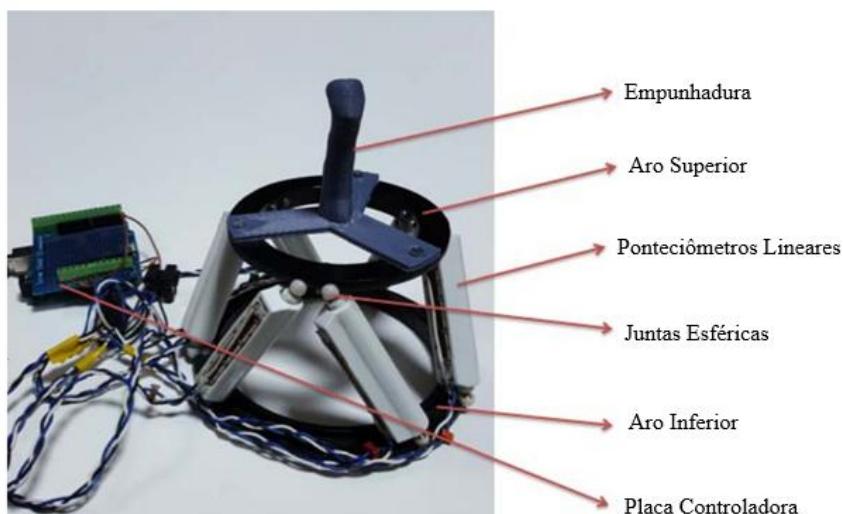


Figura 10: Manipulador 6 DOF  
Fonte: Traduzido de YILDIZ (2016)

Nesses sistemas, dispositivos como joysticks, volantes, manetes ou interfaces gráficas enviam comandos incrementais ou absolutos de posição e orientação. O controlador da plataforma converte esses comandos em poses desejadas e, posteriormente, em comprimentos de atuadores por meio da cinemática inversa. Trabalhos como o de Ömürlü e Yıldız (2012) reforçam essa abordagem ao demonstrar que uma plataforma Stewart  $3 \times 3$ , uma variante reduzida da plataforma paralela do tipo Stewart, formada por três atuadores lineares conectados a três pontos na base e três pontos na plataforma superior, pode ser operada como um joystick espacial com realimentação de força, permitindo manipulação precisa de sistemas de múltiplos graus de liberdade com resposta estável, mesmo em cenários que exigem sensibilidade a forças e torques aplicados pelo usuário.

Embora muitas aplicações industriais priorizem o controle automático, como no caso de plataformas utilizadas para compensação ativa de movimento, o controle manual permanece essencial em sistemas destinados à simulação, treinamento ou inspeção remota. Estudos do NIST (2008) mostram que a operação manual de maquinaria pesada e guindastes robóticos pode ser realizada através de um modo manual da plataforma Stewart, utilizando-a como joystick para controlar manipuladores e sistemas de movimentação de carga com alta precisão.

Assim, o controle por dispositivos manuais se consolida como uma aplicação complementar, mas amplamente difundida, em plataformas de Stewart voltadas ao ensino, simulação profissional e pesquisa. No caso deste trabalho, embora não tenha sido implementado um joystick espacial único como os apresentados em estudos específicos da literatura, nos quais um único manipulador físico comanda a plataforma em seis graus de liberdade, foi desenvolvida uma solução funcionalmente equivalente utilizando um controle de videogame convencional, cujos dois joysticks analógicos permitem comandar as translações e rotações, e os gatilhos a elevação em Z da plataforma em tempo real. Essa implementação possibilita ao operador explorar o espaço de movimento de forma contínua, testar limites do sistema, validar a cinemática aplicada e interagir diretamente com o mecanismo desenvolvido.

## **2.4. CONTROLE**

O controle de sistemas dinâmicos desempenha papel fundamental na robótica e na automação, pois garante estabilidade, precisão e repetibilidade no movimento de atuadores. Entre as diversas estratégias de controle existentes, o controlador Proporcional–Integral–Derivativo (PID) é, segundo Ogata (2011), um dos mais aplicados em sistemas industriais devido à sua simplicidade de implementação, robustez e eficiência em uma ampla gama de processos.

Para o processo de sintonia do controlador PID, pode-se utilizar um modelo matemático que descreve o comportamento dinâmico do sistema em estudo. Tal modelo pode ser obtido de diversas formas, sendo normalmente realizado seguindo duas abordagens. A modelagem que utiliza leis físicas e relações teóricas sobre as variáveis envolvidas, chamada de modelagem caixa-branca, baseia-se em equações diferenciais derivadas de princípios físico-químicos do processo. Já a modelagem caixa-preta utiliza dados experimentais para estimar o comportamento dinâmico sem exigir uma descrição analítica detalhada do sistema. Conforme apresentado por Garcia (2018), esse tipo de modelagem empírica é associado às técnicas de identificação de sistemas, nas quais a resposta obtida experimentalmente é utilizada para construir um modelo aproximado do processo.

No segundo caso, em que é utilizada uma curva de reação do processo para estimar parâmetros, busca-se normalmente empregar modelos mais simples, tipicamente funções de transferência de primeira ou segunda ordem, com ou sem tempo morto. Esse procedimento consiste em submeter o sistema a sinais de teste padronizados, como degrau, impulso ou senoide, e observar a resposta em malha aberta. Como explicado por Garcia, esses ensaios permitem extrair informações essenciais sobre ganho estático, constantes de tempo e atrasos do processo, possibilitando a construção de modelos empíricos de baixa ordem adequados para fins de análise e sintonia de controladores. Esses modelos, ainda que aproximados, geralmente são suficientes para suportar métodos clássicos de sintonia PID, fornecendo uma visão clara do comportamento transitório do processo e permitir a avaliação preliminar de desempenho do sistema de controle.

Os sistemas de primeira ordem são caracterizados por uma única constante de tempo  $\tau$  e um ganho estático  $K$ , conforme apresentado na Equação 16. A resposta desse sistema a uma entrada degrau apresenta comportamento exponencial, aproximando-se gradualmente do valor de regime permanente.

Nessa resposta, um sistema de primeira ordem submetido a um degrau de amplitude  $A$  apresenta uma resposta exponencial que atinge aproximadamente 63,2% do valor final no instante  $t = \tau$ , enquanto cerca de 98% do valor de regime é alcançado em torno de  $4\tau$ . A constante de tempo  $\tau$  está diretamente associada à velocidade de resposta do sistema, enquanto o ganho estático determina a amplitude final atingida. A caracterização desses parâmetros é essencial para a identificação de modelos a partir de curvas de reação e será utilizada posteriormente na interpretação dos resultados experimentais.

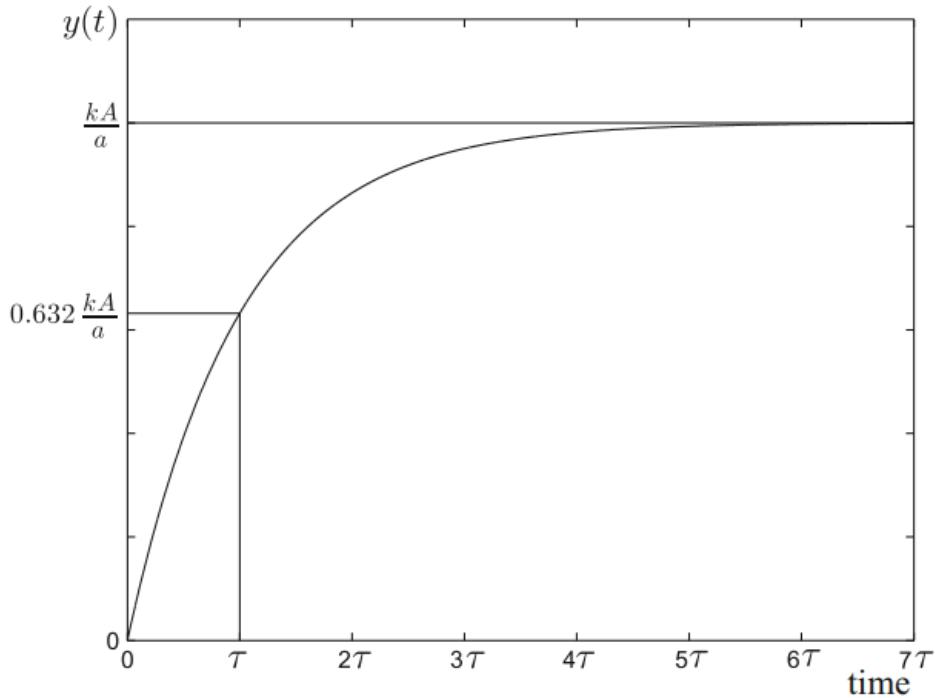


Figura 11: Resposta Típica de um Sistema de Primeira Ordem a um Degrau  
Fonte: GARCIA (2018)

A partir da curva apresentada na Figura 11, observa-se que a saída  $y(t)$  cresce até atingir o valor final determinado pelo ganho estático. No instante  $t = \tau$ , o sistema alcança aproximadamente 63,2% do valor de regime, o que caracteriza a constante de tempo como uma medida direta da rapidez da resposta. Após cerca de  $4\tau$ , a curva já se encontra muito próxima do regime permanente (aproximadamente 98% do valor final). Esses elementos, ganho e constante de tempo, constituem a base da modelagem de primeira (GARCIA, 2018).

Sua função de transferência típica é dada por:

$$G(s) = \frac{K}{\tau s + 1} \quad (16)$$

Quando existe um atraso de transporte (ou tempo morto), o modelo é representado como:

$$G(s) = \frac{K e^{-\theta s}}{\tau s + 1} \quad (17)$$

Esses modelos são amplamente empregados na análise de processos industriais autorregulados, por apresentarem comportamento estável e resposta suave.

Já os sistemas de segunda ordem possuem dois polos e descrevem uma faixa mais ampla de comportamentos dinâmicos, podendo ser subamortecidos, criticamente amortecidos ou superamortecidos, dependendo do coeficiente de amortecimento  $\zeta$ .

Gráfico das curvas de resposta ao degrau unitário com  $\omega_n = 1$  e  $\zeta = 0; 0,2; 0,4; 0,6; 0,8; 1$

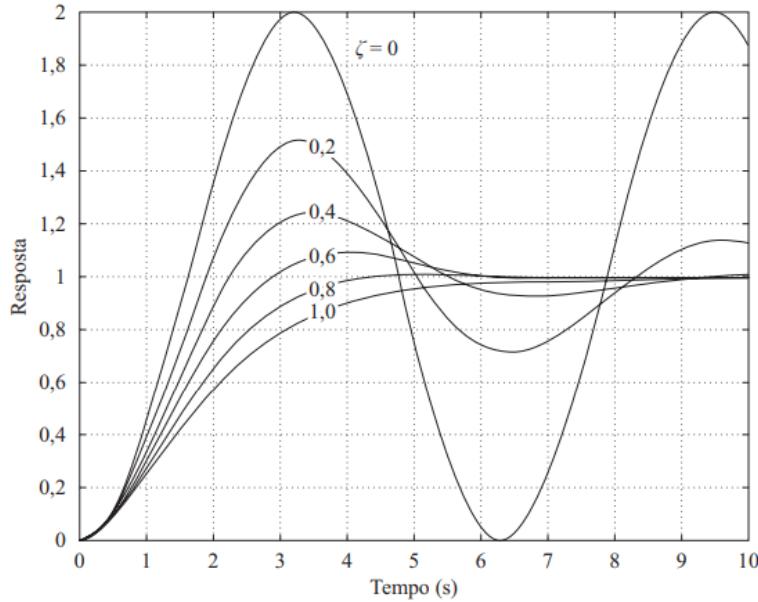


Figura 12: Comparação das curvas com diferentes  $\zeta$ .

Fonte: OGATA (2011)

Sua função de transferência geral pode ser expressa por:

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (18)$$

em que:

- $K$  é o ganho estático do sistema;
- $\omega_n$  é a frequência natural não amortecida;
- $\zeta$  é o coeficiente de amortecimento, que determina a forma da resposta transitória.

De acordo com Garcia (2018), quando o sistema é superamortecido ( $\zeta > 1$ ), os dois polos reais e distintos produzem uma resposta suave e sem sobressinal, que pode ser aproximada por um modelo de primeira ordem com tempo morto (FOPDT) sem perda significativa de precisão.

Essa aproximação é válida porque o polo dominante representa a maior parcela da dinâmica do sistema, enquanto o segundo polo pode ser compensado pelo termo de atraso ( $\theta$ ) na modelagem.

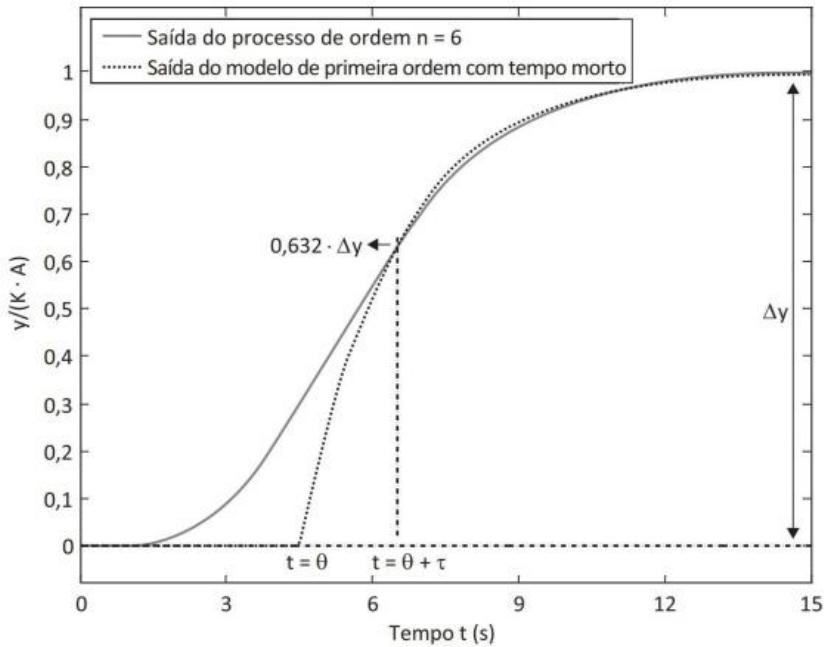


Figura 13: Aproximação por modelo de 1<sup>a</sup> ordem com tempo morto  
Fonte: GARCIA (2018)

Dessa forma, a maioria dos sistemas físicos com essas características podem ser modelados, de forma prática e precisa, por um ganho, uma constante de tempo e um tempo morto, o que simplifica o projeto de controladores e permite a aplicação de métodos clássicos de sintonia, como os de Ziegler–Nichols ou Cohen–Coon. Essa abordagem foi adotada neste trabalho para o controle de posição dos atuadores da plataforma de Stewart, possibilitando uma modelagem mais simples e eficiente para o projeto do controlador PID.

O controlador PID atua sobre o erro do sistema, diferença entre o valor desejado (referência) e o valor medido (saída), aplicando uma ação de controle composta por três termos:

- Proporcional (P): reage diretamente à magnitude do erro, proporcionando resposta rápida, mas não elimina o erro em regime permanente quando usado isoladamente.
- Integral (I): acumula o erro ao longo do tempo, corrigindo desvios residuais e garantindo erro nulo em regime permanente.
- Derivativo (D): antecipa a tendência de variação do erro, atuando como fator de amortecimento que reduz oscilações e melhora a estabilidade transitória.

Matematicamente, a lei de controle PID no domínio do tempo é expressa como:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (19)$$

em que  $K_p$ ,  $K_i$  e  $K_d$  são os ganhos proporcional, integral e derivativo, respectivamente;  $e(t)$  é o erro instantâneo; e  $u(t)$  é o sinal de controle aplicado ao atuador (OGATA, 2011).

No caso da plataforma de Stewart, a escolha do controle PID neste trabalho se alinha ao objetivo de restaurar e modernizar o sistema, fornecendo uma solução prática e robusta para o controle de posição dos atuadores, ao mesmo tempo em que mantém um caráter educacional, permitindo o estudo de diferentes estratégias de modelagem, identificação e sintonia de controladores aplicadas a robôs paralelos.

#### 2.4.1. FENÔMENO DO WINDUP EM SISTEMAS COM SATURAÇÃO

Quando o sistema opera próximo aos limites do atuador, é comum que o controlador gere comandos que excedem a capacidade física do dispositivo. Nessa situação, o sinal de controle é recortado (*clamped*) para o valor máximo permitido, mas o integrador do PID, baseado em um modelo linear independente das limitações reais, continua acumulando erro como se o atuador estivesse respondendo plenamente.

Este comportamento é conhecido como *windup* ou integrator *windup*. Åström e Hägglund (1995) demonstram que este fenômeno pode gerar longos tempos de acomodação, grandes *overshoots*, transientes oscilatórios, perda momentânea de estabilidade e recuperação lenta após a saída da saturação.

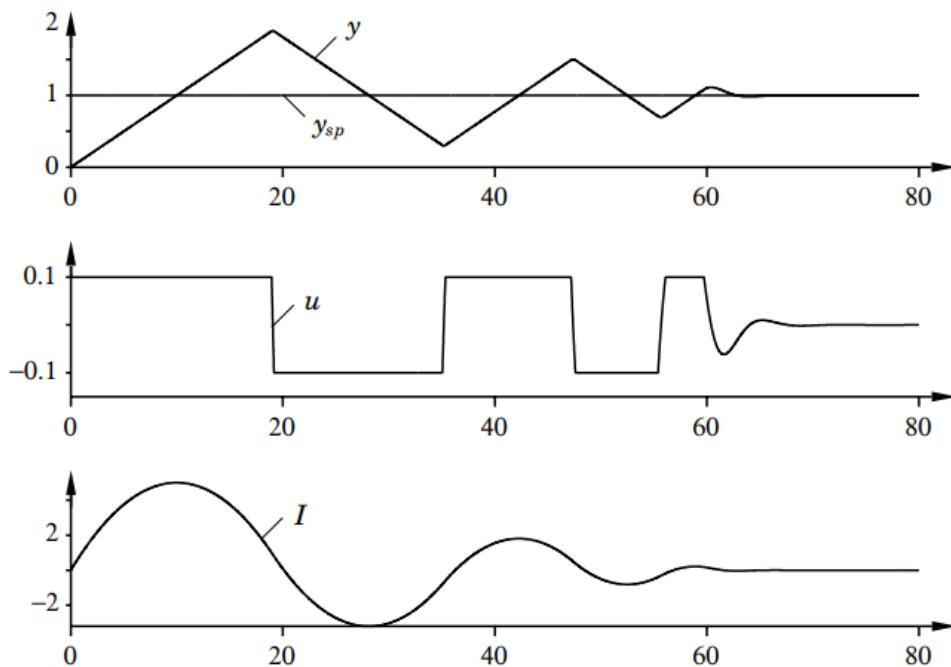


Figura 14: Ilustração do Integrador de *Windup*  
Fonte: ÅSTRÖM E HÄGGLUND (1995)

Em sistemas multivariáveis, como a plataforma de Stewart, o *windup* tende a se intensificar, pois cada atuador pode saturar em instantes diferentes devido ao acoplamento cinemático, fazendo com que a ação integral cresça de forma desigual entre os braços.

Para mitigar os efeitos nocivos do *windup*, diversos métodos de *anti-windup* são utilizados na literatura. Åström e Hägglund (1995) apresentam quatro estratégias amplamente empregadas:

- 1. Back-Calculation (Tracking Anti-Windup)*

Realimenta ao integrador a diferença entre o comando saturado e o comando ideal.

- 2. Conditional Integration*

O integrador só acumula erro quando o atuador está dentro da faixa não saturada.

- 3. Integrator Clamping*

O integrador é congelado quando o erro contribui para aprofundar a saturação.

- 4. Métodos Dinâmicos Baseados em Modelo*

Empregam modelos internos da saturação do atuador para compensar a não linearidade.

Técnicas de *anti-windup* são indispensáveis em sistemas reais, especialmente em robôs paralelos e atuadores eletromecânicos, onde a saturação é frequente.

## **2.5. CONCEITO DE APLICAÇÕES WEB E ARQUITETURA CLIENTE-SERVIDOR**

As aplicações web utilizam, de forma geral, o paradigma cliente–servidor, no qual duas partes cooperam para executar uma tarefa. O cliente, normalmente um navegador ou interface gráfica instalada no dispositivo do operador, é responsável pela apresentação visual das informações e pela interação humana. Já o servidor concentra o processamento lógico, tratando solicitações, gerenciando dados e realizando a comunicação com dispositivos físicos, como sensores, atuadores ou controladores industriais. Essa arquitetura é amplamente adotada por oferecer acesso remoto, independência de plataforma e facilidade de integração entre diferentes sistemas conectados em rede (TANENBAUM; WETHERALL, 2021).

No contexto da automação e da supervisão de processos, tecnologias baseadas em web têm ganhado espaço graças à sua flexibilidade e à capacidade de atualizar informações em tempo real. Conforme evidenciado por Jeng e Chieng (2020), IHMs desenvolvidas sobre plataformas web permitem comunicação direta com controladores industriais, operação

multiplataforma e monitoramento contínuo das variáveis de processo, tornando-se uma alternativa eficaz para supervisão distribuída. Além disso, o uso combinado de protocolos como HTTP e WebSocket possibilita que o servidor envie e receba informações de forma eficiente, permitindo atualizações assíncronas e bidirecionais e garantindo baixa latência, requisito essencial em sistemas dinâmicos e aplicações robóticas.

Para organizar os princípios utilizados neste trabalho, as seções seguintes apresentam os fundamentos técnicos dos principais componentes da arquitetura empregada: o modelo cliente–servidor, os protocolos HTTP e WebSocket, e a estrutura de comunicação através de uma API.

### **2.5.1. ARQUITETURA CLIENTE-SERVIDOR**

A arquitetura cliente–servidor é um modelo amplamente utilizado em sistemas distribuídos, no qual duas entidades desempenham papéis complementares. De acordo com Tanenbaum e Wetherall (2021), o cliente é responsável pela interface de interação, geração de requisições e apresentação das informações ao usuário, enquanto o servidor concentra o processamento lógico, o gerenciamento de dados e os serviços centralizados acessados pelos clientes. Essa separação de responsabilidades permite que múltiplos clientes utilizem simultaneamente os serviços fornecidos pelo servidor, garantindo organização, escalabilidade e padronização no fluxo de dados.

No âmbito de aplicações web e sistemas de supervisão, o cliente geralmente se apresenta como uma interface gráfica executada em navegadores ou dispositivos conectados à rede, enquanto o servidor executa rotinas de comunicação, tratamento de dados e disponibilização de informações. Jeng e Chieng (2020) destacam que tecnologias web têm se consolidado como solução eficaz em ambientes industriais por permitirem acesso remoto, independência de plataforma e atualização contínua das variáveis monitoradas.

A utilização do modelo cliente–servidor também facilita a integração com dispositivos industriais, uma vez que o servidor pode agregar dados provenientes de diferentes fontes, interpretá-los e disponibilizá-los ao cliente de maneira estruturada. Essa característica torna a arquitetura adequada para aplicações que exigem comunicação constante, visualização em tempo real e suporte a múltiplos usuários ou equipamentos, favorecendo sua adoção em sistemas modernos de controle, automação e robótica (TANENBAUM; WETHERALL, 2021).

### **2.5.2. PROTOCOLO HTTP**

O HTTP é o protocolo utilizado para comunicação entre clientes e servidores na Web. Ele estabelece a forma como um navegador, aplicativo ou interface gráfica solicita informações

a um servidor e como esse servidor devolve os dados solicitados. Segundo Kurose e Ross (2021), o HTTP segue um modelo simples de requisição e resposta, no qual o cliente envia um pedido e o servidor retorna o recurso correspondente — como uma página, um arquivo ou um conjunto de dados.

Uma característica importante é que o HTTP é um protocolo sem estado, ou seja, o servidor não guarda informações sobre requisições realizadas anteriormente pelo cliente. Cada interação é tratada de forma independente, o que simplifica o funcionamento da comunicação (KUROSE; ROSS, 2021).

Por ser amplamente difundido, de fácil implementação e compatível com praticamente qualquer dispositivo conectado à Internet, o HTTP tornou-se a base das aplicações web modernas. Interfaces industriais, dashboards de supervisão, sistemas de controle remoto e plataformas de visualização de dados frequentemente utilizam esse protocolo para enviar medições, comandos e configurações ao servidor ou para consultar informações em tempo real (TANENBAUM; WETHERALL, 2021).

Assim, compreender o funcionamento geral do HTTP é essencial para aplicações que, como a desenvolvida neste trabalho, dependem da troca estruturada de informações entre um dispositivo físico e uma interface de usuário acessada via navegador.

### **2.5.3. WEBSOCKET**

É um protocolo de comunicação projetado para permitir troca de dados contínua e bidirecional entre cliente e servidor por meio de uma única conexão persistente. Diferentemente do HTTP, que opera no modelo requisição–resposta, o WebSocket possibilita que ambos os lados enviem mensagens a qualquer momento, sem necessidade de novas solicitações. Isso o torna particularmente adequado para aplicações que exigem atualização em tempo real ou comunicação com baixa latência.

O WebSocket é padronizado pela RFC 6455, documento oficial que especifica todos os detalhes do protocolo. As RFCs (*Request for Comments*) são documentos padronizados publicados pela IETF (*Internet Engineering Task Force*) que descrevem protocolos e mecanismos utilizados na Internet, servindo como referência formal para implementação e interoperabilidade entre sistemas.

A especificação do protocolo descreve o WebSocket como um mecanismo que estabelece um canal full-duplex sobre uma conexão TCP tradicional, permitindo comunicação eficiente e

de longa duração. Segundo a RFC, o objetivo principal do WebSocket é fornecer um meio padronizado para comunicação bidirecional entre cliente e servidor com mínima sobrecarga, possibilitando aplicações interativas em tempo real (FETTE; MELNIKOV, 2011).

Essa característica tem levado o WebSocket a ser amplamente adotado em sistemas que dependem de notificações imediatas ou monitoramento contínuo, como interfaces industriais, plataformas de teleoperação, sistemas de controle remoto, aplicações de robótica e dashboards que exibem valores atualizados de sensores e estados de atuadores. Nesses cenários, sua conexão persistente oferece vantagens significativas, pois evita a necessidade de requisições sucessivas do cliente e reduz a latência associada à reabertura de conexões. Dessa forma, o WebSocket torna-se mais adequado que protocolos baseados exclusivamente no modelo requisição–resposta, como o HTTP, quando o objetivo é manter comunicação contínua e de baixa latência (FETTE; MELNIKOV, 2011).

#### 2.5.4. API E API REST

O termo API (*Application Programming Interface*) refere-se a um conjunto de regras, operações e padrões que permitem que diferentes sistemas de software se comuniquem entre si. Uma API define como um cliente pode solicitar informações ou serviços a um componente de software, especificando os formatos das mensagens, os recursos acessíveis e os comportamentos esperados. Conforme descrevem Richardson e Ruby (2007), uma API funciona como um contrato público que determina a forma correta de interação com um serviço, atuando como uma ponte estruturada que garante integração segura, modularidade e reutilização de funcionalidades.

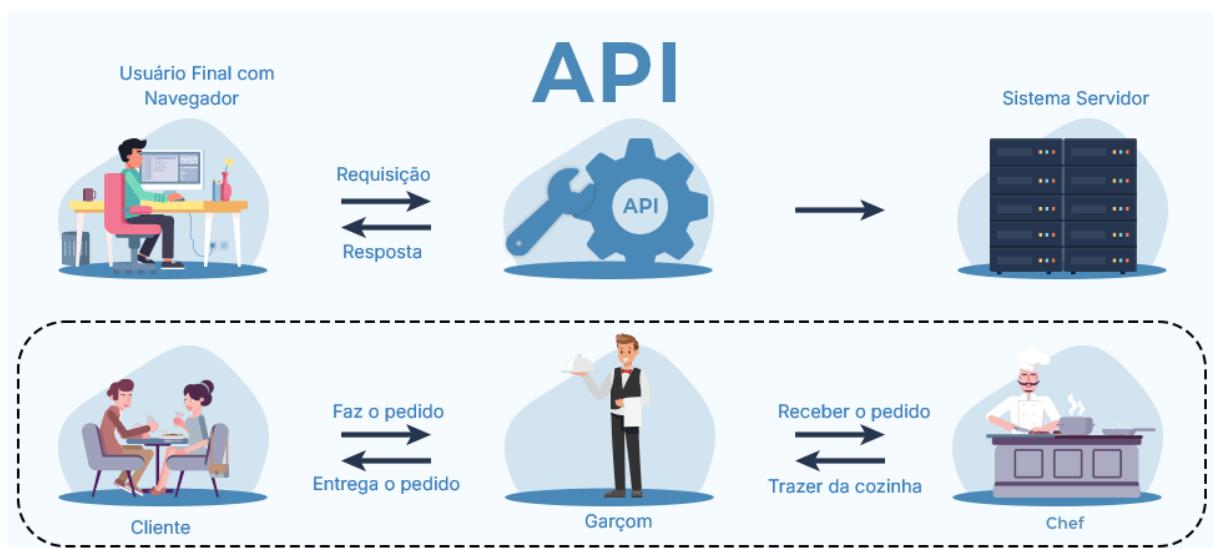


Figura 15: Comunicação API  
Fonte: Traduzido de GEEKSFORGEEKS (2025).

Entre os diferentes modos de estruturar uma API, destaca-se o modelo REST (*Representational State Transfer*), proposto por Roy Fielding em sua tese de doutorado. Segundo o autor, o REST não é um protocolo, mas sim um estilo arquitetural que organiza a comunicação entre cliente e servidor com base em princípios bem definidos, como a comunicação sem estado (*stateless*), a identificação clara de recursos por meio de URIs e a utilização de uma interface uniforme (FIELDING, 2000). Em outras palavras, o REST fornece uma forma padronizada de estruturar as requisições e respostas de uma API, tornando mais previsível a interação entre sistemas distintos.

Em uma API REST, cada recurso é representado por uma URL, e as operações sobre esses recursos são realizadas por métodos padronizados, como GET, POST, PUT e DELETE, geralmente transmitidos sobre o protocolo HTTP. Esse modelo simplifica o desenvolvimento e favorece a interoperabilidade, pois estabelece um formato consistente para acessar e manipular informações (FIELDING, 2000; RICHARDSON; RUBY, 2007).

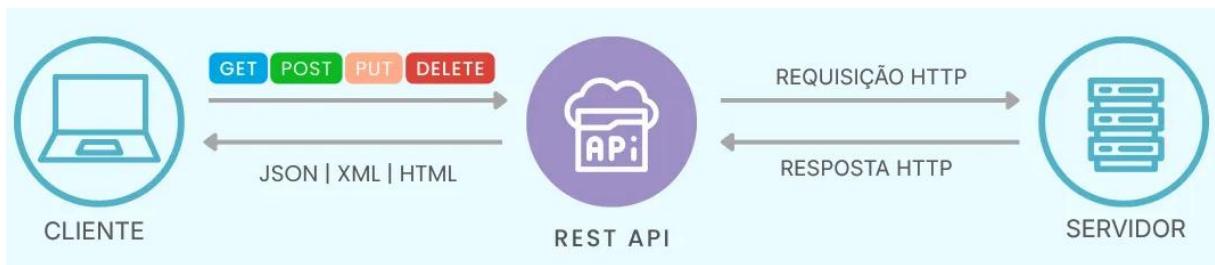


Figura 16: Modelo de API REST  
Fonte: Traduzido de SKIPLEVEL (2025).

Por sua simplicidade, baixa sobrecarga e facilidade de integração, as APIs REST tornaram-se amplamente utilizadas em aplicações web, sistemas distribuídos e plataformas que integram software e hardware. Esses sistemas se beneficiam do REST ao expor comandos, estados ou dados de sensores de forma padronizada, permitindo que diferentes módulos ou interfaces de usuário accessem informações de maneira clara e organizada (RICHARDSON; RUBY, 2007).

### 3. MÉTODO E PROCEDIMENTOS METODOLÓGICOS

Este capítulo descreve os procedimentos adotados neste trabalho para o processo de adaptação e reestruturação da plataforma de Stewart. Sendo assim, partiu-se da análise da bancada existente, na qual foi observado quais componentes foram utilizados para a sua construção, bem como a necessidade de adequações para seu funcionamento completo, conforme proposto neste trabalho. Uma vez que a bancada estivesse funcional, a próxima etapa

do trabalho consistiu em realizar testes experimentais e coleta de dados, usados para realizar o procedimento de modelagem matemática desse sistema usando métodos de identificação. Tal modelo, tem por objetivo reproduzir o comportamento dinâmico dos componentes da bancada para as simulações de projeto do controlador PID. Uma vez implementada as malhas de controle dos atuadores de posicionamento da plataforma, seguiu-se com a implementação da cinemática direta e inversa da plataforma de Stewart, e para demonstração da plataforma de Stewart em um caso real, foi feita a integração da plataforma com um simulador de voo. Durante todas as etapas envolvendo testes e aquisição de sinais da bancada, foi realizada a programação do controlador baseado em microcontrolador e desenvolvimento da interface com o operador, sendo a arquitetura de comunicação descrita também neste capítulo.

### **3.1. ANÁLISE DA BANCADA EXISTENTE NO IFSP**

Inicialmente, foi realizada uma inspeção completa da bancada de plataforma de Stewart (Figura 17) desenvolvida por Gonçalves (2023), disponível no Laboratório de Instrumentação e Controle do IFSP-SJC.



Figura 17: Bancada de plataforma de Stewart, disponível no IFSP-SJC  
Fonte: GONÇALVES (2023)

Esse levantamento englobou os seguintes aspectos:

- Componentes mecânicos (estrutura da base, plataforma móvel, juntas e curso dos atuadores);
- Tipos de sensores embarcados;
- Atuadores lineares utilizados;
- Drivers e fontes de alimentação presentes;

- Fixações mecânicas e pontos de conexão;
- Arquitetura de controle e comunicação adotada anteriormente.

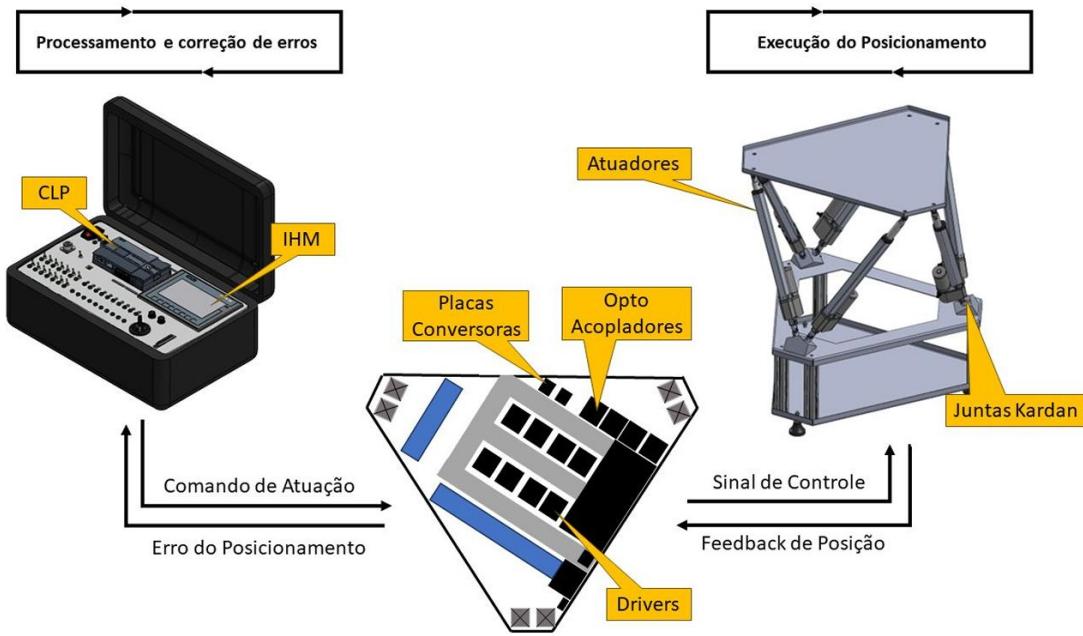


Figura 18: Estrutura Original da Bancada com CLP Siemens e Atuadores Lineares  
Fonte: GONÇALVES (2023)

A bancada foi originalmente projetada para operar com um CLP Siemens S7-1200 CPU 1215C, utilizando módulos de saída digital e analógica para o controle dos pistões elétricos lineares. No entanto, em virtude da indisponibilidade desse equipamento no campus, do seu alto custo no mercado, além da interface desenvolvida também em IHM específica para comunicação com esse CLP, optou-se por buscar uma solução mais acessível, preferencialmente usando placas de desenvolvimento de código aberto, que pudesse ser programada com linguagem e software de desenvolvimento que não necessitasse de licença paga para seu uso.

### 3.2. ADEQUAÇÃO DO CONTROLADOR DA BANCADA

Com o objetivo de desenvolver o controlador para a plataforma usando placas de desenvolvimento de código aberto, e que possibilitasse futuras modificações ou expansões, foi escolhido a placa de desenvolvimento ESP32-S3. Essa placa, além de atender os requisitos mencionados, ela também pode ser programada usando ambientes de desenvolvimento e linguagem de código aberto, baseada na família de placas Arduino. Outro ponto importante, é que a placa ESP32-S3 possui comunicação Wi-Fi, Bluetooth (Classic e BLE), além de protocolos seriais como UART, SPI, I2C, CAN e até o protocolo proprietário ESP-NOW para comunicação direta entre dispositivos sem necessidade de roteador. Assim, ela possibilita o uso

com sistemas modernos de comunicação para sistemas embarcados e possibilitando a integração com redes de computadores, e sistemas Web. Com relação aos sinais de entrada e saída para o controle da plataforma de Stewart, devido ao controlador anterior implementado ser baseado em CLP, cujo padrão de sinal digital é de 24V, e sinal analógico de 10V, foi necessário o uso de placas eletrônicas para conversão de sinal, além de módulos adicionais de expansão devido a quantidade limitada do modelo de CLP usado (GONÇALVES, 2023). Com a adequação do controlador baseado em ESP32-S3, essa configuração pode ser simplificada, devido a quantidade de entradas e saídas suficientes nessa placa de desenvolvimento, além da padronização dos sinais digitais e analógicos (mais detalhes serão discutidos no próximo capítulo).

### **3.2.1. CONTROLADOR ESP32-S3**

O ESP32-S3 foi adotado como unidade central de controle da bancada por reunir características importantes para aplicações didáticas e de pesquisa, como boa capacidade de processamento, facilidade de programação e ampla disponibilidade de pinos de entrada e saída. Embora o microcontrolador ofereça recursos avançados que permitem a implementação futura de algoritmos mais sofisticados, como estratégias de controle adaptativo, otimização ou predição, o foco neste trabalho recaiu principalmente em sua versatilidade e capacidade de integração.

Um dos fatores determinantes na escolha do ESP32-S3 foi o número elevado de GPIOs disponíveis (até 45 pinos configuráveis), que atende confortavelmente às necessidades da plataforma de Stewart. Esses pinos permitem conectar, de forma direta, todos os sinais envolvidos no controle dos atuadores prismáticos: canais PWM para acionamento de velocidade, sinais digitais para definição de direção e entradas analógicas destinadas à leitura dos sensores de feedback. Dessa forma, não se fez necessária a utilização de módulos externos de expansão, o que simplificou o cabeamento e reduziu a complexidade da arquitetura de controle.

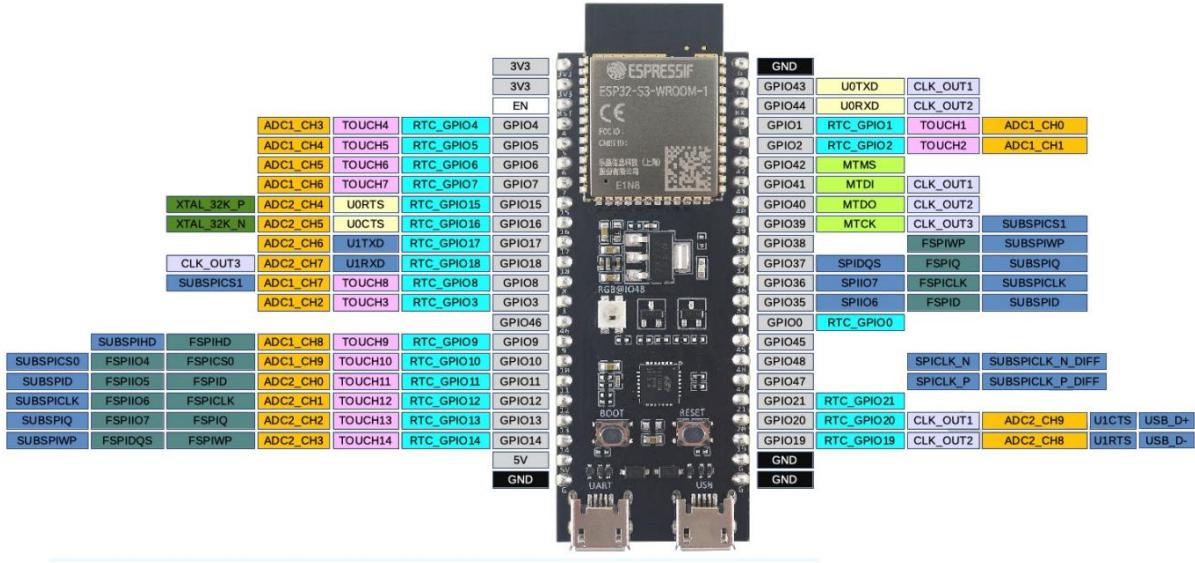


Figura 19: ESP32-S3-DevKitC-1 Pin Layout

Fonte: ESPRESSIF SYSTEMS (2024)

Além disso, o microcontrolador oferece interfaces modernas de comunicação, como Wi-Fi, Bluetooth e protocolos seriais, o que facilita sua integração com sistemas supervisórios, aplicações web e futuras extensões do projeto. Assim, o ESP32-S3 apresenta-se como uma plataforma equilibrada entre desempenho, flexibilidade e acessibilidade, atendendo plenamente às exigências do sistema desenvolvido.

### 3.3. TESTES E AQUISIÇÕES DE DADOS DA BANCADA

Após a conclusão das adequações mecânicas, elétricas e eletrônicas da bancada, foram definidos os procedimentos destinados à aquisição de dados e à avaliação experimental do comportamento dos atuadores lineares e dos sensores de posição. Os testes planejados tiveram como objetivo verificar a qualidade dos sinais de realimentação, caracterizar o desempenho individual de cada pistão e fornecer informações necessárias para as etapas posteriores de identificação de modelo e projeto do controlador.

#### 3.3.1. PROCEDIMENTOS DE TESTE E COLETA DE SINAIS

Para analisar o comportamento dinâmico dos atuadores, foram definidos testes em malha aberta, consistindo na aplicação de degraus de comando e variações controladas no sinal PWM. Durante esses ensaios, os sinais dos sensores de posição seriam coletados pelo ESP32-S3 e transmitidos ao servidor para armazenamento e posterior análise. A estratégia adotada permite comparar as respostas individuais dos seis atuadores e identificar possíveis assimetrias mecânicas ou elétricas, como diferenças de atrito, folgas ou variações de montagem.

### **3.3.2. TRATAMENTO DE RUÍDO E FILTRAGEM**

Como parte da metodologia de aquisição, foram estabelecidos métodos de tratamento de sinais para mitigar o impacto de ruídos inerentes ao processo de medição. Entre as técnicas previstas para avaliação, incluem-se:

- filtro passa-baixa digital de primeira ordem (IIR, *Infinite Impulse Response*), para reduzir componentes de alta frequência;
- filtro de média móvel, utilizando janelas de diferentes tamanhos para suavizar a leitura dos sensores;
- estratégias de supressão de picos, envolvendo pré-amostragem e limitação de variação entre amostras.

A escolha final dos filtros a serem utilizados nas rotinas de controle seria realizada com base nos resultados experimentais, apresentados no Capítulo 4.

### **3.3.3. AVALIAÇÃO DO COMPORTAMENTO DE DESLOCAMENTO**

Como etapa de caracterização da bancada, foram definidos ensaios de comparação entre atuadores, aplicando o mesmo sinal de acionamento simultaneamente aos seis pistões. Essa metodologia permite identificar diferenças de deslocamento, tempo de resposta e velocidade entre atuadores, fornecendo subsídios para ajustes individuais de parâmetros e calibrações específicas em cada canal.

### **3.3.4. CALIBRAÇÃO TENSÃO–POSIÇÃO (V0–V100)**

Para correlacionar a leitura analógica dos sensores com a posição física dos atuadores, foi adotado um procedimento de calibração baseado em dois pontos característicos:

- V0: tensão lida com o pistão totalmente recuado;
- V100: tensão lida com o pistão no limite superior do curso.

Com esses valores, planejou-se estabelecer uma relação linear entre tensão medida e posição em milímetros, permitindo que o controlador interprete adequadamente o deslocamento dos atuadores. Essa calibração seria realizada individualmente em cada pistão para compensar variações inerentes aos sensores, cabos e mecanismos.

### **3.3.5. IDENTIFICAÇÃO DE ZONA MORTA E NÃO LINEARIDADES**

Para detectar não linearidades típicas de sistemas eletromecânicos, como atrito estático, histerese e folgas, foram definidos testes de baixa amplitude, consistindo na aplicação de pequenos degraus de referência próximos ao início de movimento. Esses ensaios permitiriam estimar o nível mínimo de comando necessário para vencer o atrito de cada atuador. Com base

nessa análise, planejou-se implementar estratégias de compensação individualizada de zona morta no controlador.

### **3.3.6. FINALIDADE DOS DADOS**

Todos os dados obtidos durante os ensaios planejados seriam empregados nas etapas de:

- identificação do modelo dinâmico da planta;
- avaliação e escolha das técnicas de filtragem digital;
- implementação da compensação de não linearidades;
- projeto e sintonia do controlador PID.

## **3.4. IDENTIFICAÇÃO DE MODELO E PROJETO DO CONTROLADOR PARA A MALHA FECHADA**

Esta etapa teve como objetivo obter modelos dinâmicos representativos dos atuadores da plataforma e, a partir deles, projetar controladores capazes de garantir o rastreamento de posição e a sincronização entre os seis pistões. Para isso, definiu-se uma metodologia em duas estratégias principais, ambas baseadas em identificação de sistemas em malha fechada e sintonia posterior dos controladores no domínio da função de transferência.

Optou-se pela identificação em malha fechada pois os atuadores apresentam zona morta e atrito estático, o que impediria a obtenção de dados informativos em malha aberta. A operação com realimentação melhora a razão sinal–ruído e restringe o sistema a uma região mais linear do comportamento, favorecendo a obtenção de modelos caixa preta coerentes com a dinâmica real dos pistões.

### **3.4.1. ESTRATÉGIA GERAL DE IDENTIFICAÇÃO**

A identificação do modelo foi estruturada em duas estratégias complementares. Em ambas, o procedimento foi dividido em três fases: aquisição de dados, identificação de modelos e projeto e ajuste dos controladores.

Na primeira estratégia, foram realizados ensaios com cada pistão acionado individualmente, mantendo os demais na posição de referência. Essa abordagem permitiu analisar o comportamento de cada atuador de forma isolada. Inicialmente foi adotada uma estrutura de modelo de segunda ordem, ajustada por meio do *System Identification Toolbox* do MATLAB, a partir de dados obtidos em malha fechada.

Na segunda estratégia, os ensaios foram planejados com acionamento simultâneo dos seis pistões, possibilitando a observação do comportamento dinâmico conjunto da plataforma. Nessa etapa, optou-se por modelos de primeira ordem em malha fechada, mais condizentes com a resposta superamortecida observada em testes preliminares, utilizando novamente o *System*

*Identification Toolbox* como ferramenta de estimação. Os modelos obtidos serviram de base para o projeto de controladores com dinâmica alinhada entre os diferentes atuadores.

### 3.4.2. AQUISIÇÃO DE DADOS PARA IDENTIFICAÇÃO

A aquisição de dados foi realizada com o sistema operando em malha fechada, com ganhos proporcionais pré-definidos, de forma a mitigar não linearidades pronunciadas, como zona morta e atrito estático, e evitar saturações excessivas dos atuadores. Em ambos os conjuntos de ensaios, aplicaram-se degraus de referência em diferentes níveis de posição, registrando-se:

- o sinal de referência  $r(t)$  em milímetros;
- o sinal de controle efetivo  $u(t)$ ;
- a saída  $y(t)$  em milímetros, obtida a partir da calibração tensão–posição;
- o tempo  $t$  de aquisição.

Para padronizar o procedimento de coleta, foi utilizado um aplicativo próprio desenvolvido em Python, responsável por comunicar-se com o ESP32-S3 via porta serial, enviar os perfis de referência aos atuadores e registrar automaticamente os dados de telemetria em arquivos no formato CSV (*comma-separated values*). Esse fluxo de aquisição permitiu repetibilidade dos ensaios e organização dos dados para posterior análise no MATLAB.

### 3.4.3. MODELAGEM E SELEÇÃO DA ESTRUTURA

Com os dados de entrada e saída disponíveis, os sinais foram importados para o MATLAB e organizados no *workspace* para utilização no *System Identification Toolbox*. A metodologia de modelagem consistiu em definir estruturas candidatas de modelo (primeira e segunda ordem em malha fechada), estimar os parâmetros de cada modelo a partir dos dados experimentais e comparar o desempenho das diferentes estruturas com base em métricas de ajuste e análise gráfica da resposta.

A escolha final da estrutura adotada considerou, além dos índices de ajuste numérico, a coerência física com o comportamento observado no sistema.

### 3.4.4. PROJETO DOS CONTROLADORES

Uma vez identificadas as funções de transferência em malha fechada  $T_i(s)$  de cada atuador, procedeu-se à obtenção dos modelos equivalentes em malha aberta  $G_i(s)$ , utilizando relações clássicas entre planta e malha fechada para diferentes valores de ganho proporcional. Esses modelos foram então empregados no projeto dos controladores utilizando ferramentas de síntese do MATLAB, como o comando *pidtune* e o ambiente *Control System Designer*.

O fluxo adotado consistiu em empregar, preferencialmente, controladores de estrutura proporcional–integral (PI), em função da natureza superamortecida da planta e da sensibilidade do termo derivativo a ruídos, definir uma faixa de largura de banda comum para os seis canais, de forma a obter respostas de malha fechada com tempos de subida e acomodação semelhantes e comparar, em simulação, as respostas a degraus de referência para todos os pistões, ajustando iterativamente a banda de projeto até alcançar sincronização satisfatória entre os atuadores.

Adicionalmente, foi implementada a estratégia de controle com mecanismos de *anti-windup* para o termo integral, a fim de lidar com saturações do sinal de comando e restrições impostas pela cinemática da plataforma de Stewart.

### **3.5. DESENVOLVIMENTO DA INTERFACE DE CONTROLE E MODOS DE OPERAÇÃO**

Esta etapa teve como objetivo definir a arquitetura geral do sistema de supervisão e operação da plataforma, bem como os modos de uso previstos para interação com o operador e para integração com sistemas externos. A interface foi realizada com o intuito de permitir o controle manual, visualização em tempo real, execução de rotinas de movimento e integração com sensores inerciais e simuladores de voo, de forma modular e extensível.

#### **3.5.1. ARQUITETURA DE COMUNICAÇÃO**

A solução adotada foi estruturada em uma arquitetura distribuída formada por três elementos principais:

##### **1. Interface Gráfica**

Uma aplicação web estática desenvolvida em HTML, CSS e JavaScript, responsável por exibir controles, visualizações 3D, gráficos e indicadores. O frontend comunica-se com o sistema por meio de requisições REST para operações pontuais e WebSocket para fluxo contínuo de dados.

##### **2. Servidor**

O backend atua como camada intermediária entre o operador e o hardware. Ele recebe comandos do frontend, traduz parâmetros, valida limites de operação e encaminha instruções ao microcontrolador. Também é responsável pelo processamento da telemetria recebida da bancada e pela distribuição dessas informações ao frontend em tempo real.

##### **3. ESP32-S3 e ESP32-IMU**

O firmware do ESP32-S3 executa o controle dos atuadores e transmite continuamente dados de posição, estados internos e níveis de PWM. Um segundo módulo, o ESP32, é

dedicado à leitura de sensores inerciais (IMU), transmitindo quaternions ou ângulos ao controlador principal por meio do protocolo ESP-NOW.

Essa arquitetura foi planejada para permitir divisão clara de responsabilidades, baixo acoplamento e futura expansão do sistema com novos modos de interação, mantendo compatibilidade com protocolos amplamente utilizados em sistemas embarcados e aplicações web.

### **3.5.2. INTEGRAÇÃO COM OS MICROCONTROLADORES**

No planejamento do sistema, o ESP32-S3 foi definido como unidade principal de controle, encarregada de receber comandos do backend, calcular o sinal de controle, acionar os atuadores e enviar telemetria em intervalo regular.

O módulo ESP32 foi planejado como um subsistema independente, dedicado à captura de orientação por sensores inerciais, utilizando comunicação ESP-NOW para permitir transmissão sem depender de infraestrutura de rede externa. Esses dados foram considerados importantes para modos futuros de estabilização e para visualização de orientação em tempo real.

### **3.5.3. INTEGRAÇÃO COM FLIGHTGEAR**

Além da operação local, foi planejada uma integração com o simulador de voo FlightGear, permitindo que variáveis de altitude da aeronave (especialmente Roll e Pitch) fossem utilizadas como referência para a plataforma física.

Essa integração foi concebida por meio de um módulo auxiliar, responsável por:

- estabelecer comunicação com o simulador via Telnet;
- coletar, filtrar e limitar as variáveis recebidas;
- encaminhar as informações ao backend para cálculo da cinemática inversa;
- permitir pré-visualização de movimentos antes da aplicação real.

A integração foi planejada com foco em segurança operacional, garantindo que poses inválidas fossem detectadas antes do envio aos atuadores.

### **3.5.4. MODOS DE OPERAÇÃO PLANEJADOS**

A interface de operação foi projetada para oferecer diferentes modos de interação, adequados às diversas etapas do desenvolvimento e avaliação da plataforma. Os principais modos definidos foram:

#### **a) Painel PID**

Previsto para permitir ajuste dos ganhos de controle, acompanhamento da resposta temporal e execução de testes de referência. O modo seria baseado em recebimento contínuo de telemetria e exibição gráfica para facilitar a análise do comportamento dinâmico dos atuadores.

#### **b) Modo de Cinemática e Visualização 3D**

Planejado para permitir cálculo de cinemática inversa a partir de poses definidas pelo usuário, bem como reconstrução da pose real da plataforma a partir da telemetria enviada pelo ESP32-S3. A visualização seria feita mediante modelo 3D atualizado em tempo real.

#### **c) Rotinas de Movimento**

Previstas como geradores de trajetória parametrizáveis, permitindo movimentos senoidais, circulares, helicoidais e combinações lineares e angulares. O backend ficaria responsável por calcular poses sucessivas e validar o movimento antes de enviar os comandos ao hardware.

#### **d) Controle por IMU**

Planejado para utilizar dados fornecidos pelo módulo ESP32, permitindo que a plataforma seguisse pequenas inclinações detectadas pelos sensores iniciais. Esse modo foi definido como base para futuros estudos de estabilização ativa.

#### **e) Controle por Joystick**

Previsto para permitir controle manual da plataforma utilizando dispositivos compatíveis com a Gamepad API do navegador. Os eixos dos joysticks do controle seriam convertidos em pose alvo e validados pelo módulo de cinemática antes do envio ao controlador.

#### **f) Integração com Simulador de Voo**

Planejada para permitir que o movimento da plataforma fosse guiado por dados provenientes do FlightGear, reproduzindo em escala reduzida manobras simples da aeronave, especialmente em *Roll* e *Pitch*.

## 4. APRESENTAÇÃO E ANÁLISE DE RESULTADOS

### 4.1. BANCADA DE PLATAFORMA DE STEWART APÓS ADEQUAÇÕES

A partir da análise realizada na seção 3.1, a bancada de plataforma de Stewart foi submetida a um conjunto de adequações estruturais, elétricas e de organização com o objetivo de restabelecer seu funcionamento pleno e prepará-la para a nova arquitetura de controle adotada neste trabalho. A Figura 20 apresenta a bancada após o processo de adequação.

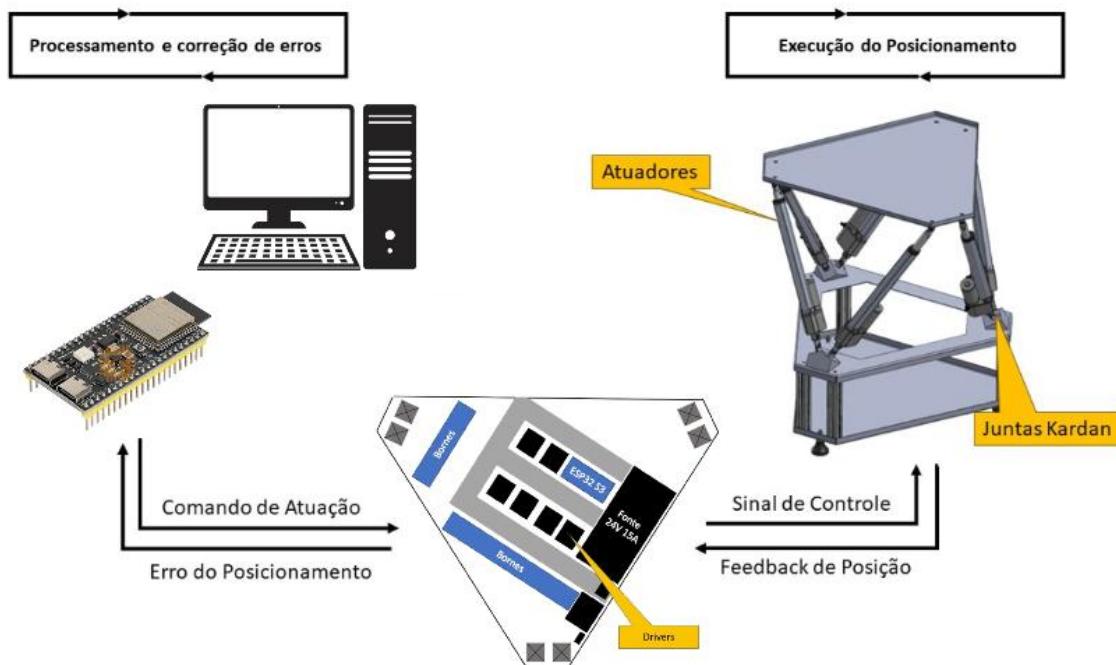


Figura 20: Estrutura Adaptada da Bancada com ESP32-S3 e Atuadores Lineares  
Fonte: Adaptado de GONÇALVES (2023)

O primeiro passo consistiu na verificação das condições mecânicas da estrutura original, incluindo a plataforma superior, a base fixa, as juntas de Kardan e o curso útil dos seis atuadores lineares. Durante essa análise, identificou-se que algumas das juntas de Kardan apresentavam desgaste significativo, podendo comprometer a articulação entre os atuadores e a plataforma móvel. Para garantir segurança mecânica e permitir a reposição futura dessas peças, as juntas foram reprojetadas em ambiente CAD e reconstruídas por meio de uma nova peça desenvolvida no SolidWorks, mantendo dimensões e geometrias compatíveis com a estrutura existente.

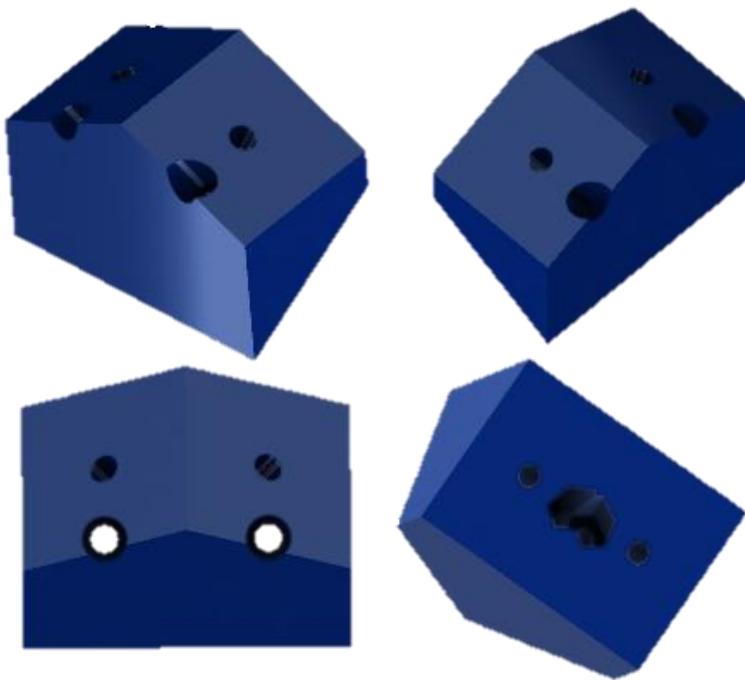


Figura 21: Juntas de Kardan Reconstruídas

Fonte: O autor

Em seguida, procedeu-se à reorganização do cabeamento da bancada. Na configuração original, muitos cabos estavam associados a componentes utilizados pelo CLP Siemens, que deixaram de ser necessários com a nova arquitetura de controle. O cabeamento foi reestruturado para eliminar essas vias redundantes e, ao mesmo tempo, estabelecer uma separação clara entre fios de potência, cabos de controle e linhas de feedback dos sensores, contribuindo para reduzir interferências, facilitar manutenções e melhorar o fluxo de organização interna da bancada.

Na parte eletrônica, foram removidos componentes intermediários que já não fariam parte do novo sistema, como conversores analógico–PWM, placas NPN/PNP e interfaces de 24 V da lógica anterior. Também foram ajustados os circuitos de alimentação dos sensores, adequando níveis de tensão e conexões ao ESP32-S3. Por fim, manteve-se o conector DB37 original, garantindo compatibilidade com a estrutura existente e possibilitando sua integração direta.

## 4.2. ADEQUAÇÃO DO CONTROLADOR DA BANCADA

### 4.2.1. SUBSTITUIÇÃO DO CLP PELO ESP32-S3

O ESP32-S3 passou a assumir o papel de unidade central de controle, sendo responsável por:

- Gerar sinais PWM para controle de posição e velocidade dos atuadores prismáticos;

- Gerar sinais digitais para as entradas de direção de atuação.
- Realizar leitura dos sinais de feedback dos sensores de posição dos atuadores;
- Executar o algoritmo de controle PID;
- Comunicar com sistemas externos via protocolo de comunicação serial.

#### **4.2.2. RETIRADA DOS OPTOACOPLADORES E MUDANÇA NA LÓGICA DE CONTROLE.**

Na arquitetura anterior, eram utilizadas placas NPN e PNP com optoacopladores (modelo C309) para realizar a conversão dos sinais lógicos provenientes do CLP Siemens, que opera com nível lógico de 24 V, para níveis compatíveis com a lógica de 5 V utilizada nos drivers dos motores lineares. Esses optoacopladores eram responsáveis pela comutação das entradas IN1 e IN2, necessárias para o controle de direção dos pistões.

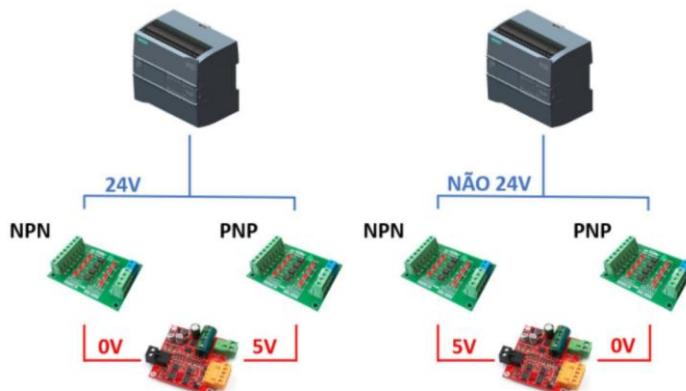


Figura 22: Funcionamento conversores NPN/PNP  
Fonte: GONÇALVES (2023)

Com a substituição do CLP pelo ESP32-S3, a necessidade dessas placas intermediárias foi eliminada. Isso se deve ao fato de que o ESP32-S3 operar com sinais lógicos de 0 a 3,3V e os drivers utilizados (como o JZ-3615 ou similares) operarem com sinais lógicos entre 2 V e 5 V, tornando possível a conexão direta entre o microcontrolador e as entradas de controle do driver, sem risco de incompatibilidade elétrica ou danos aos componentes.

O controle de avanço e recuo dos motores passou a seguir diretamente a lógica de acionamento descrita na tabela do fabricante do driver. De acordo com essa lógica, os estados das entradas IN1 e IN2, combinados com o sinal de PWM, determinam o modo de operação do motor, conforme descrito a seguir:

IN1	IN2	PWM	Estado do motor
0	0	x	Freio
1	1	x	Flutuando
1	0	1	Velocidade máxima
0	1	1	Velocidade máxima reverso
1	0	PWM	Controle velocidade para frente
0	1	PWM	Controle velocidade Ré

Figura 23: Comando dos Motores

Fonte: USINAINFO (2025)

Essa configuração permitiu maior simplicidade no cabeamento e controle, além de reduzir o número de componentes eletrônicos, melhorando a manutenção e expandindo a flexibilidade do sistema de controle da bancada.

As figuras a seguir ilustram como, na arquitetura anterior, as placas NPN e PNP eram utilizadas de forma complementar para adaptar os sinais de 24 V para a lógica de 5 V exigida pelo driver, e como essa adaptação foi eliminada na nova arquitetura, que utiliza controle direto via ESP32-S3:

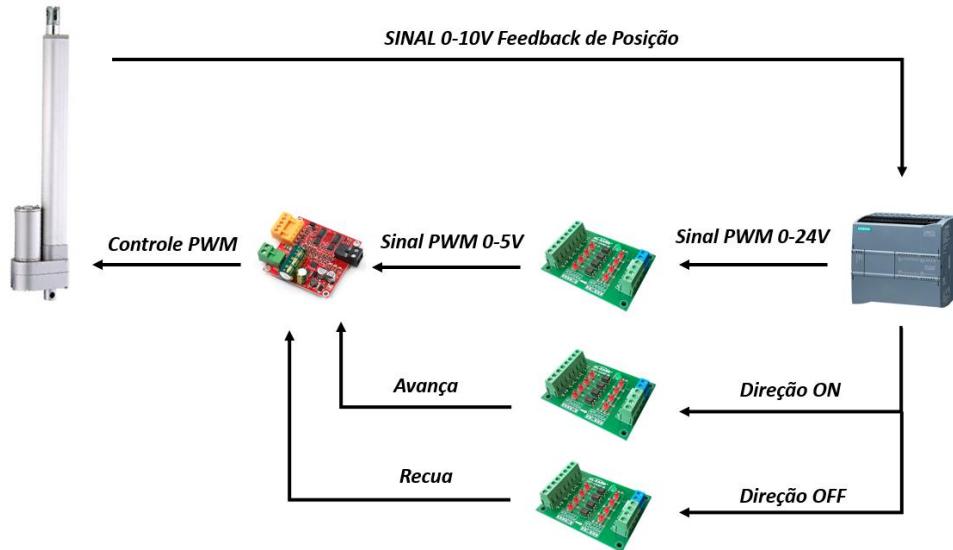


Figura 24: Arquitetura original com CLP e conversores NPN/PNP

Fonte: GONÇALVES (2023)

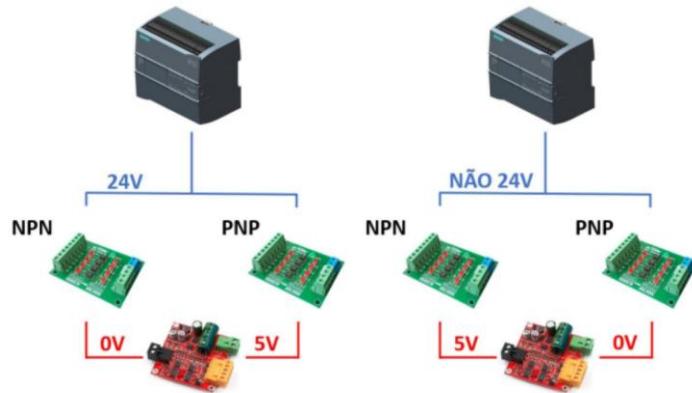


Figura 25: Funcionamento conversores NPN/PNP  
Fonte: GONÇALVES (2023)

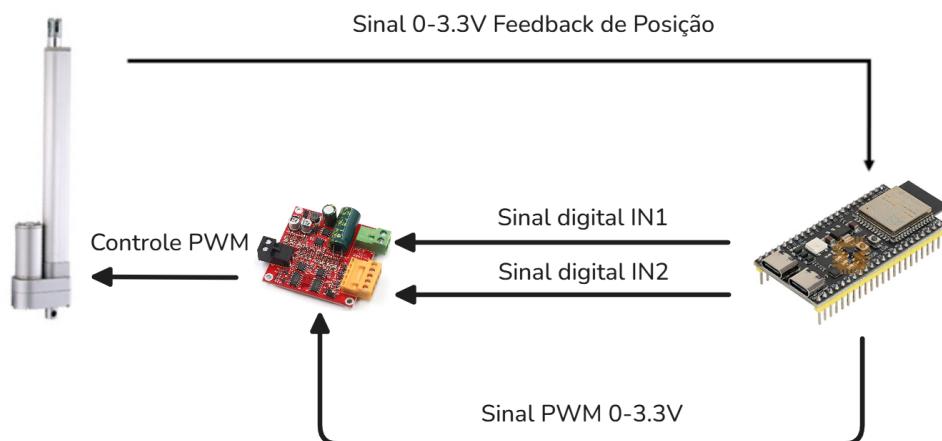


Figura 26: Arquitetura atual com ESP32-S3 e controle direto  
Fonte: Adaptado de GONÇALVES (2023)

#### 4.2.3. RECONFIGURAÇÃO DOS DRIVERS DE POTÊNCIA

Os drivers utilizados (como o modelo RQG ou JZ-3615) foram compatibilizados com o novo microcontrolador ESP32-S3. Para isso, foram realizados os seguintes ajustes:

- A frequência do sinal PWM foi configurada para cerca de 20 kHz, valor recomendado para garantir uma resposta eficiente e silenciosa dos atuadores lineares;
- As entradas de direção dos drivers (IN1 e IN2) passaram a ser controladas diretamente por pinos digitais do ESP32-S3;
- O controle de velocidade dos atuadores passou a ser realizado por meio de canais PWM de hardware do ESP32-S3, utilizando sua estrutura nativa de temporizadores.

Na estrutura original, devido à limitação do CLP Siemens (que possuía apenas quatro saídas PWM), foram utilizadas placas conversoras analógico para PWM, do modelo LC-LM358-PWM2V. Essas placas eram responsáveis por converter sinais analógicos de 0–10 V em sinais PWM de 5 V, com duty cycle proporcional à tensão analógica recebida, permitindo controlar os drivers com base em saídas analógicas do CLP.

Com a adoção do ESP32-S3, essas placas intermediárias tornaram-se desnecessárias. O novo controlador é capaz de gerar múltiplos sinais PWM simultâneos, com alta resolução e controle de *duty cycle* por software, permitindo a substituição direta dos conversores e uma simplificação da eletrônica do sistema.

#### **4.2.4. ADAPTAÇÃO DOS SENSORES DE FEEDBACK**

A bancada utiliza atuadores lineares equipados com feedback resistivo de posição, que funcionam de forma semelhante a potenciômetros. Esses sensores possuem três terminais: dois para os limites de tensão e um terminal de sinal. O funcionamento ocorre com base na variação da resistência elétrica proporcional ao deslocamento do atuador, o que resulta em uma tensão analógica correspondente à posição do atuador.



Figura 27: Atuador Linear  
Fonte: GONÇALVES (2023)

Anteriormente, os sensores eram alimentados com uma faixa de tensão de 0 a 10 V, compatível com módulos analógicos industriais utilizados em CLPs. No entanto, o ESP32-S3 opera com tensão máxima de entrada analógica de 3,3 V, o que tornava inviável a leitura direta desses sinais sem risco de danificar o microcontrolador. Dessa forma, foi realizada uma adequação da alimentação dos sensores de feedback, reduzindo a faixa de 0–10 V para 0–3,3 V.

#### 4.2.5. DESENVOLVIMENTO DE PLACA DE CIRCUITO IMPRESSO (PCB) DE INTERFACE

Com o objetivo de integrar de forma segura e organizada o microcontrolador ESP32-S3 ao hardware da plataforma Stewart, foi desenvolvido um circuito dedicado em placa de circuito impresso (PCB). Essa placa tem a função de acomodar fisicamente o ESP32-S3, disponibilizar suas entradas e saídas em bornes de parafuso organizados e realizar a interface direta com os sinais da plataforma através de um conector DB37.

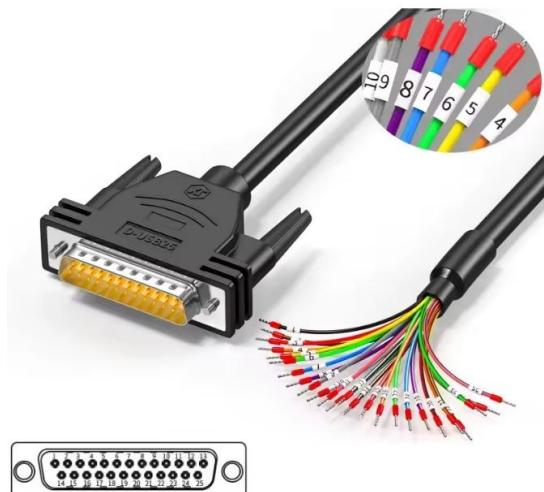


Figura 28: Conector DB37 com fios numerados  
Fonte: ALIEXPRESS (2025)

A necessidade dessa PCB surgiu devido à quantidade significativa de sinais envolvidos no controle da bancada, incluindo os canais de PWM, as entradas de direção (IN1 e IN2) para cada atuador, além das entradas analógicas dos sensores de feedback. De forma a evitar conexões soltas, ruídos elétricos ou falhas por mau contato em protoboards, optou-se por desenvolver uma interface física mais confiável para testes.

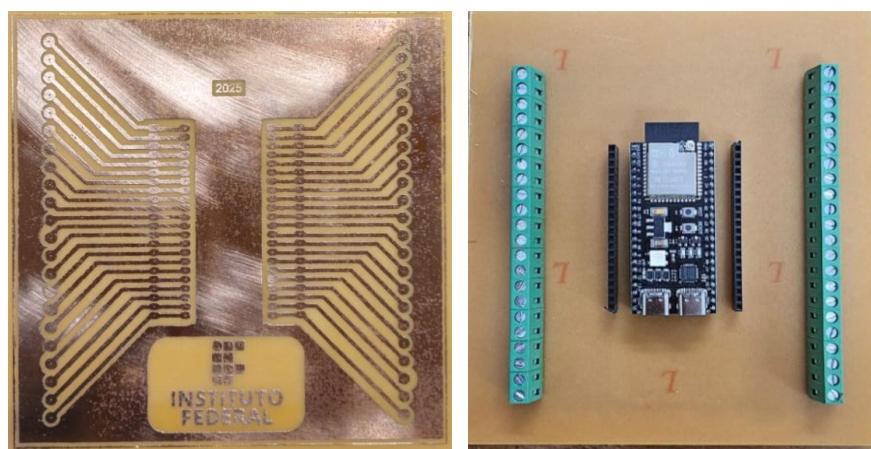


Figura 29: PCB para Interface com ESP32S3  
Fonte: O autor

#### 4.2.6. MAPEAMENTO DOS SINAIS DO CONECTOR DB37

Os sinais provenientes da plataforma estavam organizados da seguinte forma no DB37 original:

Tabela 1: Organização Original DB37

Pistão	Pinos Digitais IN1	Pinos Digitais IN2	Feedback de posição	PWM
1	19	4	25	15
2	20	1	26	16
3	21	11	27	17
4	22	12	28	18
5	23	13	29	33
6	24	14	30	34

Esses sinais foram então reorganizados e conectados aos GPIOs do ESP32-S3, respeitando limitações de conversão analógica, canais de PWM e pinos adequados para uso como saídas digitais.

A Tabela 2 apresenta o mapeamento final dos pinos utilizados no ESP32-S3 para cada um dos seis pistões, incluindo entradas de feedback, canais PWM e sinais digitais de direção (IN1/IN2):

Tabela 2: Organização DB37 Após Adaptação da Bancada

Pistão	Pinos Digitais IN1	Pinos Digitais IN2	Feedback de posição	PWM
1	16	17	1	8
2	13	14	2	18
3	35	36	3	9
4	21	38	4	10
5	39	37	5	11
6	41	42	6	12

#### 4.3. TESTES E AQUISIÇÕES DE DADOS DA BANCADA

Após a finalização das adaptações elétricas e eletrônicas, foram iniciadas as primeiras aquisições de dados da bancada com o ESP32-S3. O objetivo inicial foi verificar o funcionamento da nova estrutura e a fidelidade dos sinais de realimentação obtidos pelos sensores de posição.

##### 4.3.1. TRATAMENTO DE RUÍDO E FILTRAGEM

Durante os primeiros testes, observou-se que os sinais provenientes dos sensores apresentavam ruído de alta frequência, dificultando a análise precisa do deslocamento real dos atuadores. Essa característica é evidenciada no primeiro conjunto de dados (Figura 30), no qual as curvas de feedback mostram variações bruscas e não coerentes com o comportamento físico esperado.

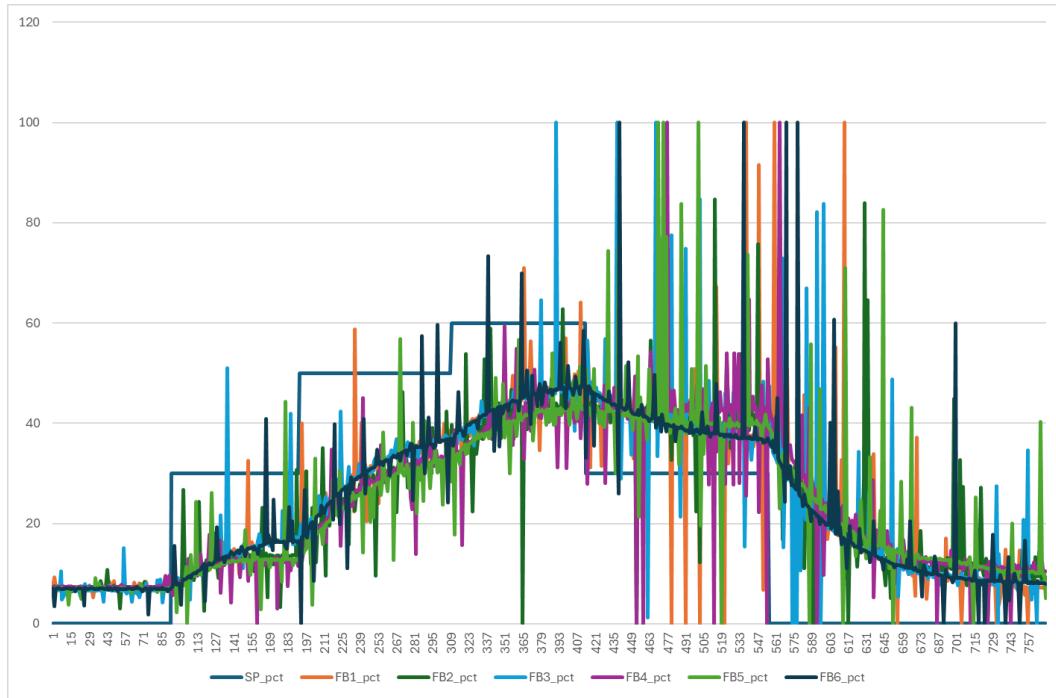


Figura 30: Aquisição inicial - Ruídos

Fonte: O autor

Como primeira medida, implementou-se no firmware do ESP32-S3 um filtro passa-baixa digital de primeira ordem (IIR), configurado com uma frequência de corte ajustável ( $f_c = 4$  Hz, valor inicial), reduzindo ruídos de medição sem prejudicar a dinâmica de resposta. O resultado do uso do filtro é apresentado na Figura 31 onde se nota a suavização significativa dos sinais e maior coerência entre o setpoint e o retorno de posição de cada pistão.

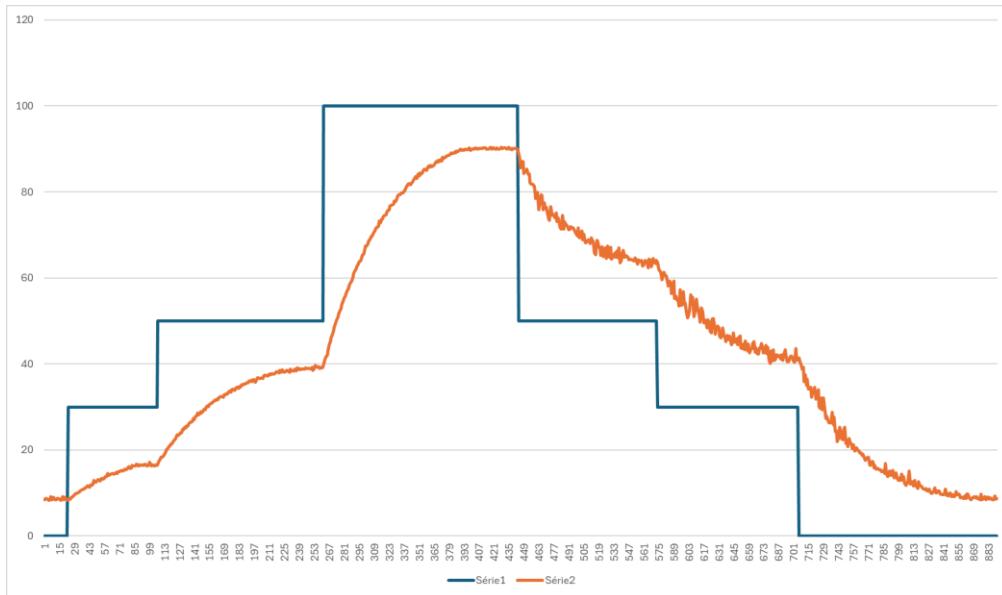


Figura 31: Resposta com filtro passa-baixa

Fonte: O autor

Apesar do ganho obtido, verificou-se que, principalmente no movimento de recuo dos pistões, ainda ocorriam perturbações que degradavam a atuação do controlador. Buscando reduzir esses picos residuais, avaliou-se inicialmente o uso de um filtro de média móvel baseado em janelas deslizantes. No primeiro teste (janela N = 8; Figura 32) observou-se uma melhora considerável no ruído durante o recuo, entretanto, picos isolados ainda surgiam ocasionalmente, além de uma leve defasagem introduzida pela janela de amostragem.

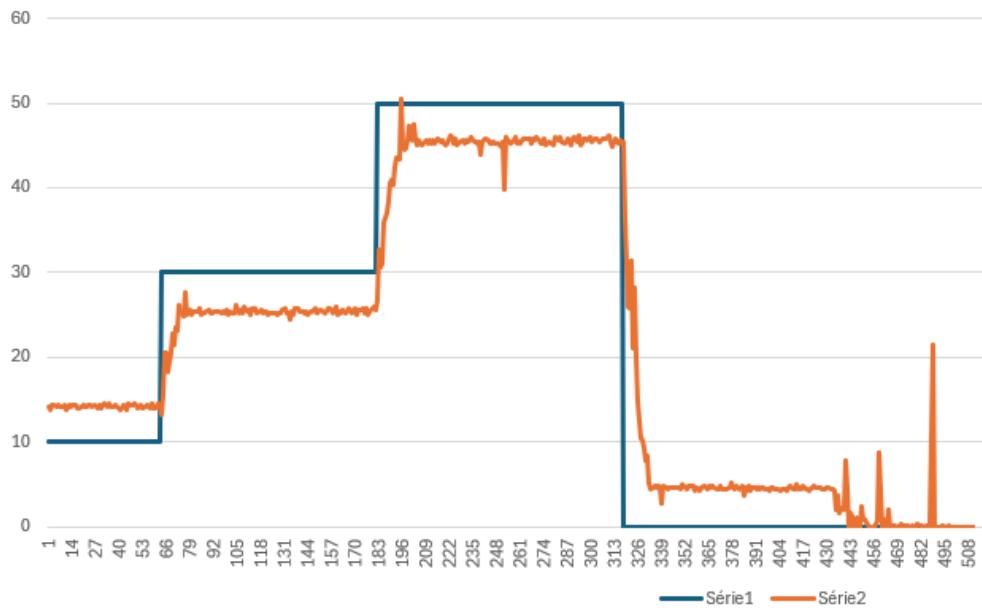


Figura 32: Resposta com filtro média móvel (janela de 8)  
Fonte: O autor

Com o objetivo de eliminar definitivamente esses ruídos sem aumentar a latência, foi implementada uma camada anti-picos de baixo custo computacional, aplicada antes do controlador. Essa abordagem combinou uma pré-amostragem por mediana curta com um limitador de inclinação, dispensando a necessidade de utilizar a média móvel. O funcionamento da camada é descrito a seguir:

1. Pré-amostragem por mediana curta: a cada ciclo são realizadas três leituras rápidas no mesmo canal do ADC e utiliza-se a mediana como amostra. Essa etapa elimina pontos discrepante isolados típicos de comutação e interferência instantânea.
2. Limitador de inclinação: a variação entre a nova amostra e a última amostra “aceita” é limitada a um passo máximo por ciclo,  $|\Delta V| \leq \Delta V_{\max}$ . No firmware, adotou-se  $\Delta V_{\max} \approx 30 \text{ mV/ciclo}$ , valor que suaviza degraus sem mascarar a dinâmica real.

A aplicação isolada dessa estratégia mostrou-se suficiente para eliminar picos e estabilizar a realimentação dos sensores, sem introduzir atraso perceptível na resposta. A Figura

33 apresenta o resultado em que se observa a remoção completa dos ruídos e a manutenção da coerência entre o movimento físico e o sinal medido.

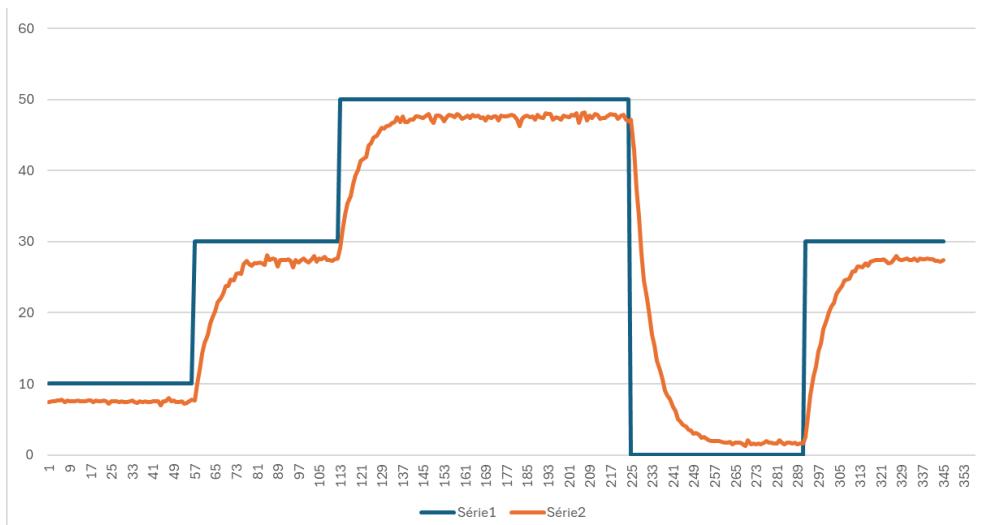


Figura 33: Resposta com Filtro Anti-Picos (mediana + limitador de inclinação)  
Fonte: O autor

#### **4.3.2. ANÁLISE DE COMPORTAMENTO DE DESLOCAMENTO ENTRE PISTÕES**

Os testes subsequentes revelaram outro comportamento, mesmo quando o mesmo sinal PWM era aplicado simultaneamente aos seis atuadores, suas respostas não eram idênticas. As curvas de deslocamento (Figura 34) mostram que há diferenças de tempo e amplitude entre pistões, indicando variações mecânicas e elétricas individuais, como diferenças nos atritos internos, e possíveis discrepâncias nos drivers de potência e variação na calibração dos sensores. Diante disso, definiu-se a necessidade de controle individual por pistão, cada atuador passou a possuir parâmetros próprios de calibração e ajuste, de modo a garantir sincronização real e resposta mais homogênea no deslocamento da plataforma.

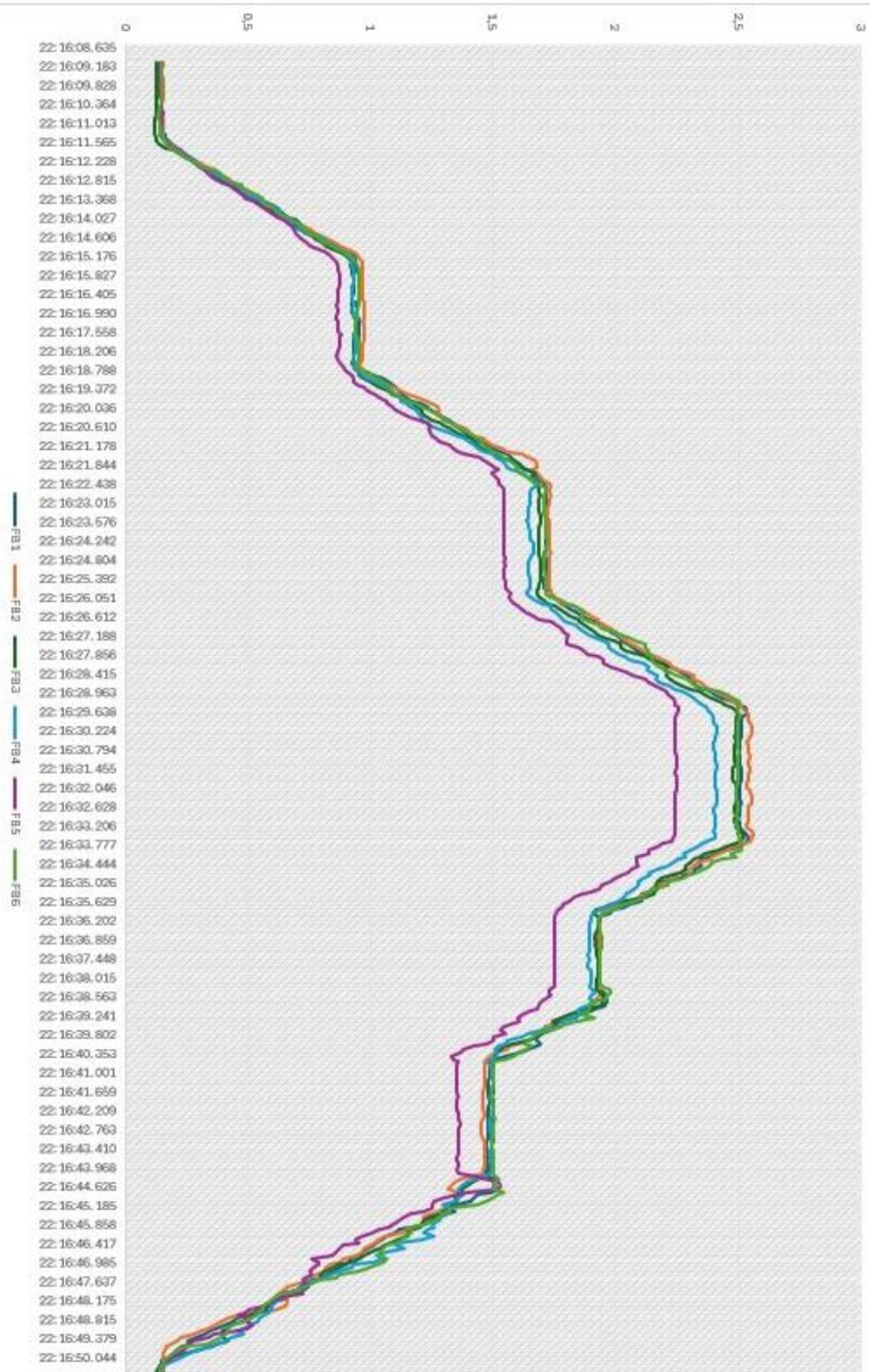


Figura 34: Deslocamento dos pistões (mm) com aplicação simultânea de PWM  
Fonte: O autor

### 4.3.3. MAPEAMENTO DA TENSÃO → POSIÇÃO (V<sub>0</sub>–V<sub>100</sub>)

Durante as calibrações iniciais, constatou-se um offset nas leituras dos sensores de feedback, mesmo com o pistão totalmente recuado, a tensão medida não era zero. Para corrigir esse efeito, adotou-se calibração por dois pontos ( $V_0$  e  $V_{100}$ ), mapeando tensão em posição (mm) ao longo do curso útil. Assim, o sistema interpreta  $V_0$  como posição zero e utiliza o *span* ( $V_{100} - V_0$ ) para definir o ganho de conversão, assegurando coerência entre a posição física dos atuadores e os valores empregados pelo controle.

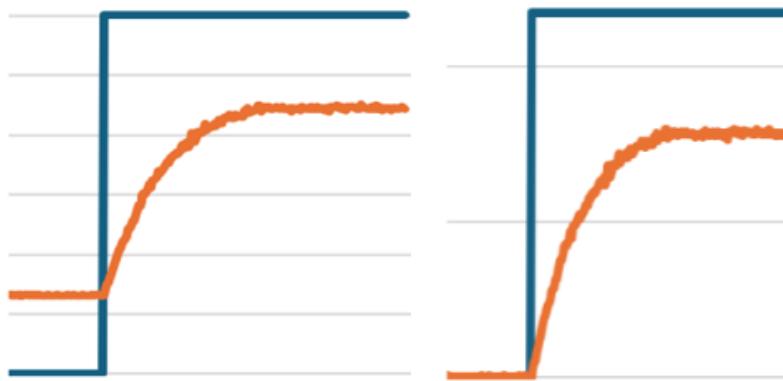


Figura 35: Correção de Offset

Fonte: O autor

Isso porque os sensores fornecem tensão elétrica, mas o controle precisa trabalhar em milímetros. Para converter → posição de forma simples e confiável, mede-se:

1.  $V_0$ : a tensão com o pistão no início do curso (definido como 0 mm).
2.  $V_{100}$ : a tensão com o pistão no fim do curso útil (definido como  $L$ mm).

Assume-se que, dentro desse intervalo, a relação é aproximadamente linear. Assim, qualquer leitura  $V$  entre  $V_0$  e  $V_{100}$  é convertida para posição  $y$  por uma regra de três:

$$y_{mm} = L \cdot \frac{V - V_0}{V_{100} - V_0}, \quad (20)$$

"limitando "y" ao intervalo  $[0, L]$ .

Em termos práticos,  $V_0$  corrige a posição zero do sensor e  $V_{100}$  fixa o ganho (define a relação V/mm). Essa calibração é feita individualmente para cada pistão, pois pequenas diferenças de sensor, cabos, eletrônica e mecânica fazem com que cada atuador tenha seu próprio par ( $V_0, V_{100}$ ). Com isso, a leitura de posição fica alinhada à realidade física, reduz erros sistemáticos e melhora a repetibilidade e a estabilidade do controle próximo ao setpoint.

### 4.3.4. COMPENSAÇÃO DE ZONA MORTA/ATRITO

A operação do sistema mecânico evidenciou não linearidades, que podem ter causas diversas, tais como atrito estático e viscoso, histerese, folgas mecânicas, pequenos

desalinamentos e influência gravitacional, fatores que afetaram o comportamento dinâmico dos atuadores. Tais efeitos são característicos de sistemas de posição: o atrito estático impede o início do movimento até que um esforço mínimo seja vencido, enquanto o atrito de Coulomb se manifesta após o início do deslizamento. Ambos produzem não linearidades descontínuas, resultando em zonas mortas e erros em regime permanente (HERNÁNDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019).

Inicialmente, foi adotado um sinal de PWM mínimo global como forma de compensação, aplicado de maneira uniforme a todos os atuadores. Entretanto, essa abordagem mostrou-se limitada, pois não considerava as diferenças individuais entre pistões, tanto em termos de atrito, folga e geometria, quanto de condição de montagem. Em diversos trechos do curso, observou-se que determinados pistões apresentavam resistência ao movimento, ou seja, um mesmo comando de entrada não gerava deslocamento, evidenciando a presença de zonas mortas distintas para cada atuador.

Para superar essas limitações e tornar a resposta mais previsível, implementou-se uma compensação de zona morta individualizada, baseada em dois níveis fixos de pré-acionamento:  $U_0(\text{adv})$  (avançar) e  $U_0(\text{ret})$  (retornar). Esses valores foram ajustados empiricamente para cada pistão, representando o esforço mínimo necessário para vencer o atrito estático e iniciar o movimento em cada sentido.

Na prática, os valores  $U_0(\text{adv})$  e  $U_0(\text{ret})$  foram somados ao sinal de controle apenas enquanto o erro permanecia fora da faixa morta (*deadband*), fornecendo o impulso inicial de arranque para início do movimento. Ao entrar na região de *deadband*, o pré-acionamento era automaticamente retirado, desenergizando o atuador para evitar oscilações e sobre-esforço.

O procedimento de calibração consistiu em aplicar sinais em degrau de baixa amplitude de referência em cada pistão e ajustar os níveis de avanço e retorno até que o movimento se iniciasse de forma repetível e simétrica em ambos os sentidos, inclusive nos trechos com tendência a travamento.

Essa estratégia resultou em uma redução significativa das não linearidades observadas, garantindo respostas mais consistentes entre os atuadores, maior previsibilidade dinâmica e dados experimentais mais limpos para as etapas posteriores de identificação e sintonia do controle.

#### **4.4. IDENTIFICAÇÃO DE MODELO E PROJETO DO CONTROLADOR PARA A MALHA FECHADA**

Esta etapa teve como objetivo implementar o controle de posição individual dos seis pistões da plataforma e promover a sincronização entre os atuadores, de modo a minimizar as discrepâncias de tempo e amplitude identificadas nos testes experimentais.

O procedimento metodológico foi estruturado em duas estratégias distintas, cada uma composta por três fases sequenciais: aquisição de dados, identificação do modelo e sintonia do controle.

Na primeira estratégia, a coleta de dados foi realizada acionando cada pistão individualmente, mantendo os demais na posição de referência (0 mm). Nessa etapa, foi empregado um sinal de PWM mínimo global, associado a um filtro passa-baixa para suavização do sinal de realimentação. A identificação do sistema foi conduzida por meio do *System Identification Toolbox* do MATLAB, com o objetivo de ajustar um modelo de segunda ordem.

Apesar dos resultados satisfatórios obtidos em simulação, essa abordagem não apresentou desempenho adequado em aplicação prática, em virtude da influência de ruídos nas medições e do comportamento inadequado gerado pelo uso de um sinal de PWM mínimo global. Ainda assim, os dados coletados nessa fase serviram como base para o desenvolvimento da estratégia subsequente.

Na segunda estratégia, a coleta de dados foi realizada de forma simultânea para todos os pistões, possibilitando analisar o comportamento dinâmico conjunto da plataforma. Nessa configuração, foi implementado um filtro anti-picôs, responsável por eliminar ruídos e picos isolados nas medições sem comprometer a resposta temporal do sistema. Além disso, foram adotados sinais individuais de PWM mínimo para cada atuador,  $U_0(\text{adv})$  (avançar) e  $U_0(\text{ret})$  (retornar), em substituição ao sinal de PWM global, medida que reduziu a ocorrência de saturações e aprimorou a linearidade e a sincronização entre os pistões.

Além disso, o sistema foi reaproximado por um modelo de primeira ordem, mais condizente com o comportamento superamortecido observado experimentalmente, servindo como base para a nova etapa de sintonia e controle.

##### **4.4.1. FASE 1 – AQUISIÇÃO DE DADOS**

Optou-se por realizar os testes de identificação com o sistema operando em malha fechada, com o objetivo de modelar diretamente a dinâmica da variável de posição dos atuadores. Nessa configuração, é possível impor degraus de referência (setpoints) com amplitude controlada e registrar a resposta de posição de cada pistão, caracterizando o sistema por meio de curvas de reação adequadas para a obtenção de modelos lineares aproximados.

Essa abordagem segue metodologias experimentais amplamente utilizadas na engenharia de controle, onde a identificação baseada na resposta a degraus da variável controlada é empregada para estimar parâmetros fundamentais, como ganho estático e constante de tempo, diretamente a partir do comportamento observado no sistema real (HERNANDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019).

Na etapa de aquisição de dados, a estratégia inicial consistiu em coletar os dados de cada pistão individualmente, configurar um PWM mínimo global, com o objetivo de reduzir os trechos sem movimento que prejudicariam a estimação dos parâmetros, melhorando a razão sinal-ruído e evitando saturações do atuador.

No procedimento experimental, aplicaram-se sinais em degrau de referência (*setpoint*) com  $K_p = 1$ , registrando a entrada efetiva  $u(t)$  aplicada a cada pista, o setpoint  $r(t)$  em milímetros, a saída  $y(t)$  em milímetros, convertida pela calibração  $V_0 - V_{100}$  filtrada e o tempo de aquisição.

Para simplificar e padronizar as coletas, foi desenvolvido um aplicativo próprio em Python/Tkinter denominado (Figura 36). Esse programa realiza a comunicação com o ESP32-S3 via porta serial (115200 bps) e permite conduzir os ensaios de forma assistida.

A interface do aplicativo lista e gerencia as portas seriais disponíveis, possibilitando a seleção do pistão sob teste e o envio de setpoints de referência em milímetros ou em porcentagem do curso, tanto de forma manual quanto por perfis pré-configurados. Também há um modo personalizado, no qual o próximo setpoint é transmitido somente após a detecção automática de estabilização do sistema (baseada em uma janela deslizante de aproximadamente 8 s e uma tolerância ajustável, com valor padrão de 3 mm).

Durante cada ensaio, o programa registra automaticamente a telemetria recebida em arquivos CSV, permitindo posterior análise dos sinais.

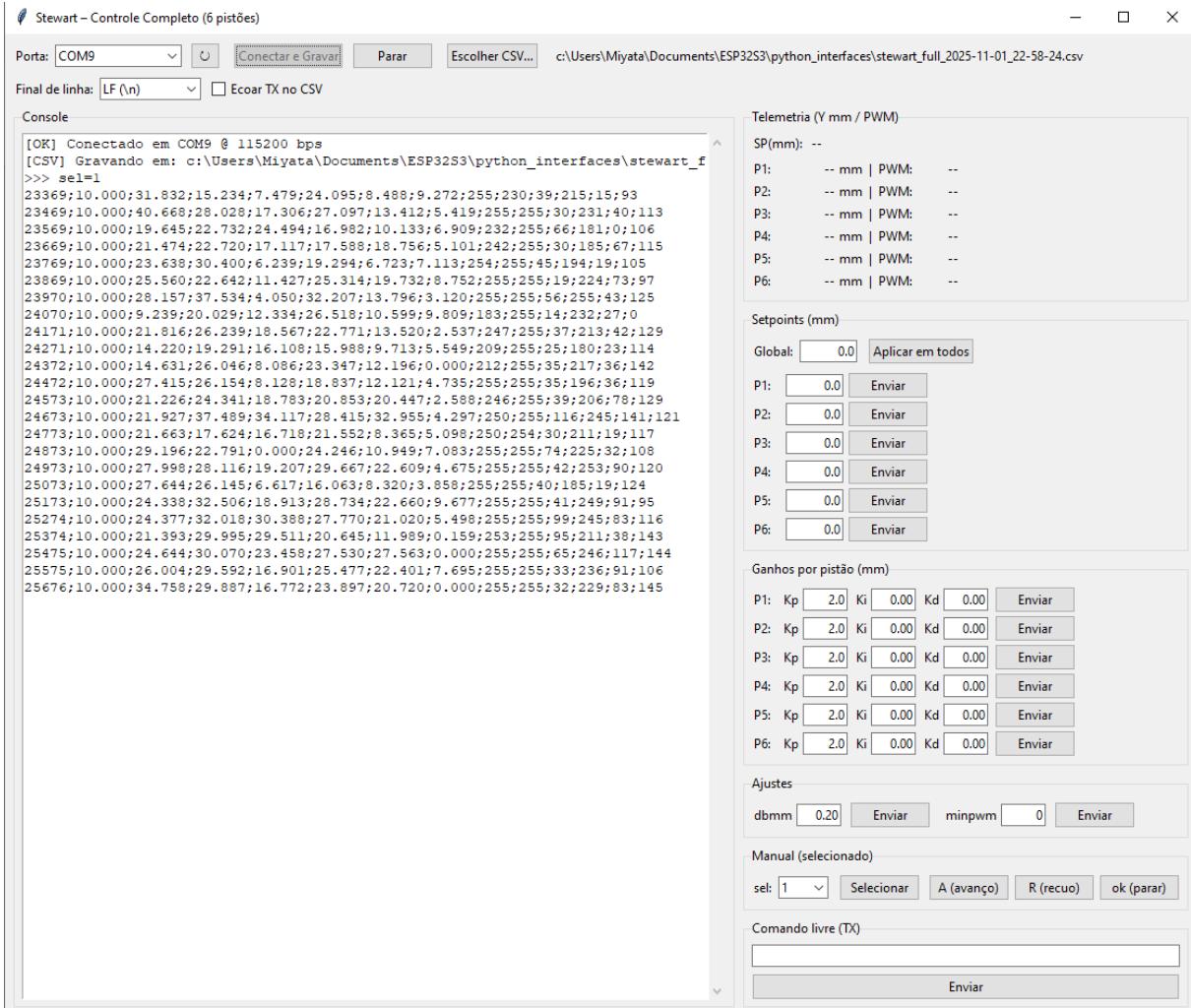


Figura 36: Interface do Aplicativo de Coleta de Dados

Fonte: O autor

Dados obtidos:

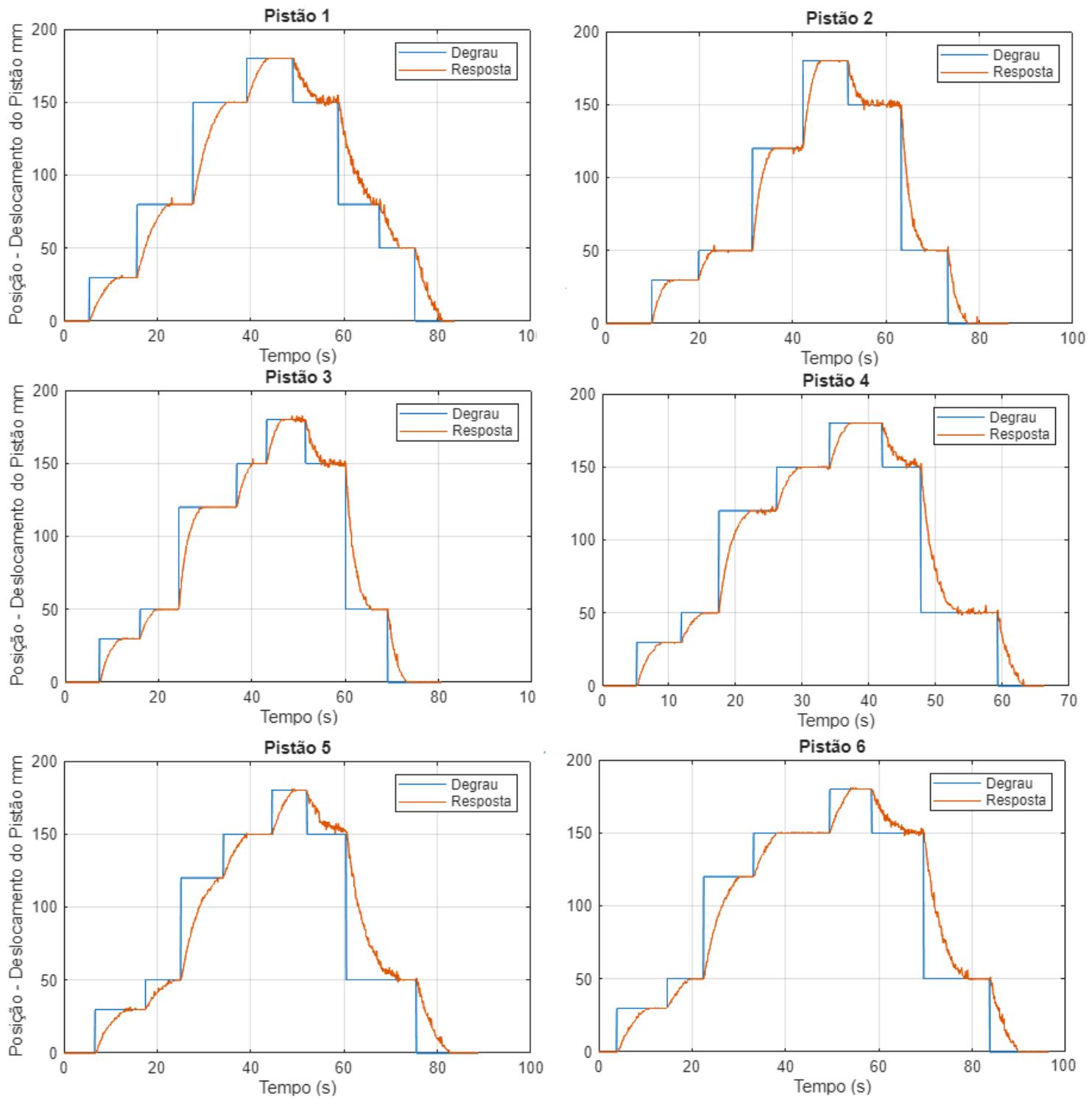


Figura 37: Dados Obtidos da Coleta Pistões 1-6  
Fonte: O autor

Os dados obtidos inicialmente, utilizando o filtro passa-baixa e o PWM mínimo global, apresentaram inconsistências significativas que posteriormente inviabilizaram o controle do sistema, o que motivou a reformulação da estratégia experimental.

Nessa segunda estratégia, as coletas foram realizadas de forma simultânea para todos os pistões, permitindo observar o comportamento conjunto da plataforma. Durante os ensaios, foi verificado que, com um ganho proporcional  $K_p = 5$ , os pistões apresentavam um comportamento mais linear e uniforme ao longo do curso.

Nessa etapa, foi implementado um filtro anti-picôs de baixa complexidade computacional, substituindo o filtro passa-baixa utilizado anteriormente. O método baseia-se na aplicação de uma pré-amostragem por mediana curta e de um limitador de inclinação, capazes de eliminar picos isolados e ruídos transitórios sem introduzir atraso perceptível na resposta.

Além disso, o sistema passou a empregar um PWM mínimo individual por pistão, denominado  $U_0(\text{adv})$  (avançar) e  $U_0(\text{ret})$  (retornar), determinados experimentalmente, de forma a eliminar zonas mortas e garantir uma transição suave entre os sentidos de movimento, evitando a saturação dos atuadores.

O mesmo aplicativo foi utilizado para conduzir os ensaios, mantendo o formato padronizado dos arquivos de telemetria e a comunicação via ESP32-S3 (115200 bps). A principal diferença em relação à estratégia anterior foi a qualidade dos sinais adquiridos, que passaram a apresentar respostas mais limpas e coerentes, possibilitando a identificação mais precisa dos modelos dinâmicos de cada atuador nas etapas subsequentes.

Dados obtidos:

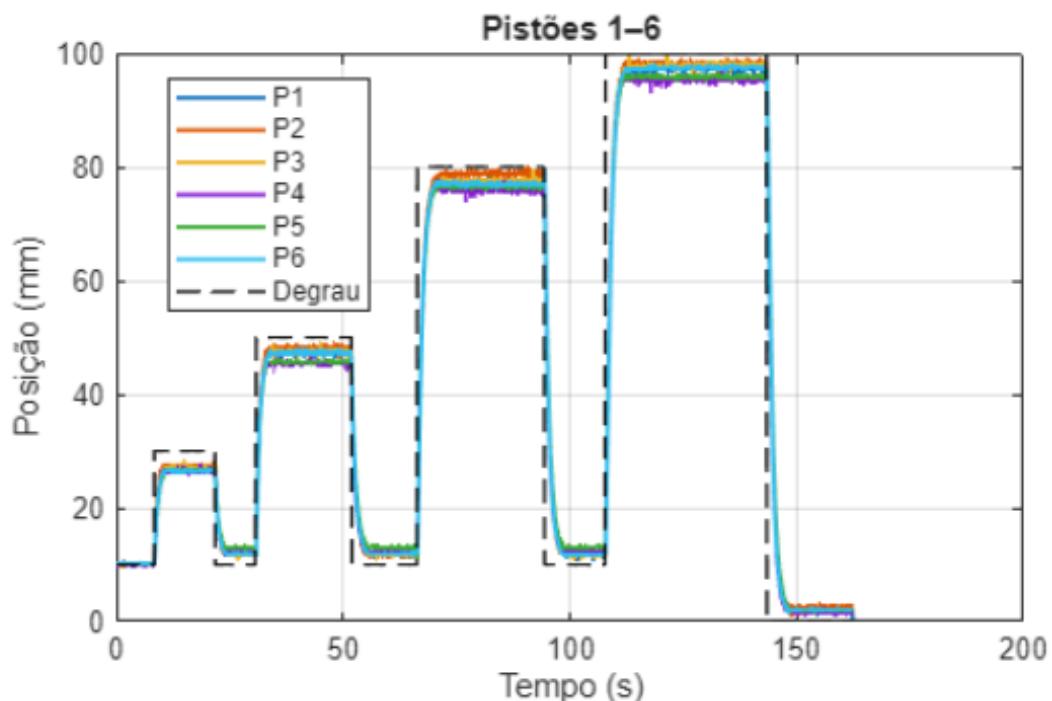


Figura 38: Coleta simultânea dos 6 pistões com  $K_p = 5$   
Fonte: O autor

#### 4.4.2. FASE 2 - IDENTIFICAÇÃO DE MODELO

Os A identificação do sistema foi inicialmente realizada com base nos dados coletados na primeira estratégia de controle, obtidos individualmente para cada pistão com o PWM mínimo global e filtragem passa-baixa. O objetivo dessa etapa foi determinar um modelo de função de transferência que representasse o comportamento dinâmico de cada atuador.

Os dados de referência ( $r$ ), resposta ( $y$ ) e tempo ( $t$ ) foram importados para o MATLAB e armazenados no *workspace* para utilização no *System Identification Toolbox*, com o propósito de estimar os parâmetros do modelo.

O procedimento foi conduzido de forma padronizada para cada pistão, conforme as etapas a seguir:

1. Importação dos dados do workspace para o *System Identification*:

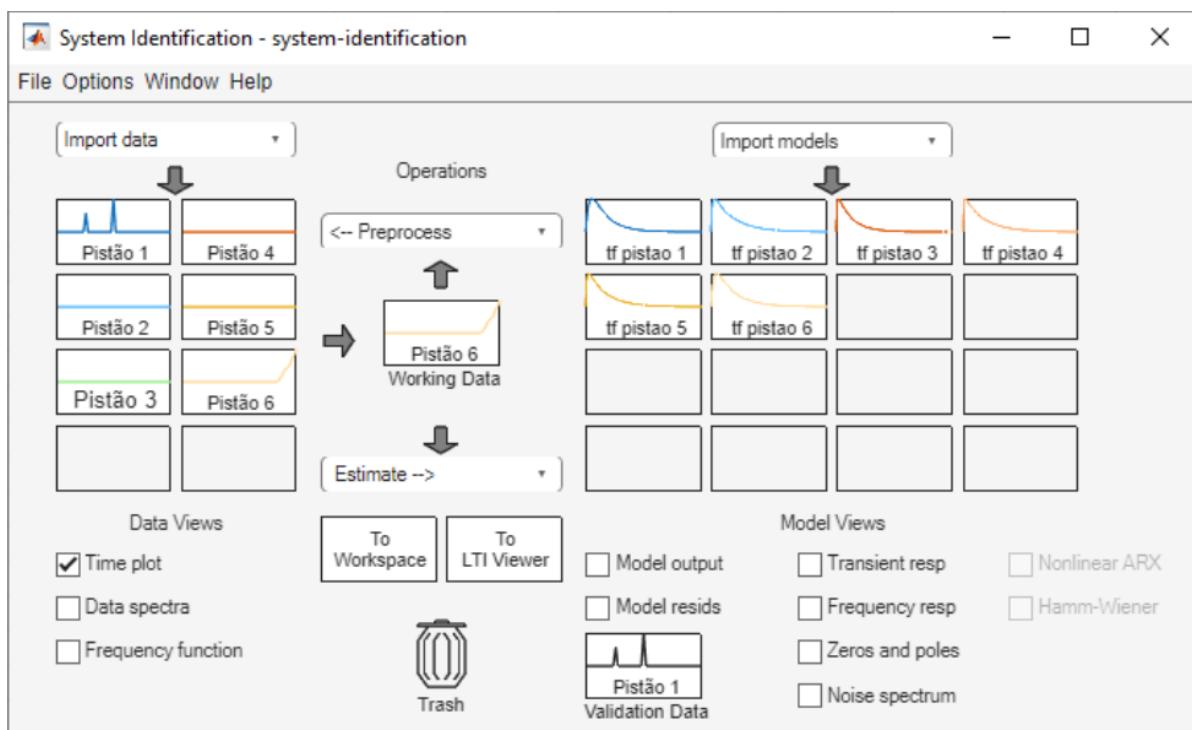


Figura 39: Dados Importados para o System Identification  
Fonte: O autor

2. Estimação de um modelo:

Para o caso utilizando uma estimação de segunda ordem.

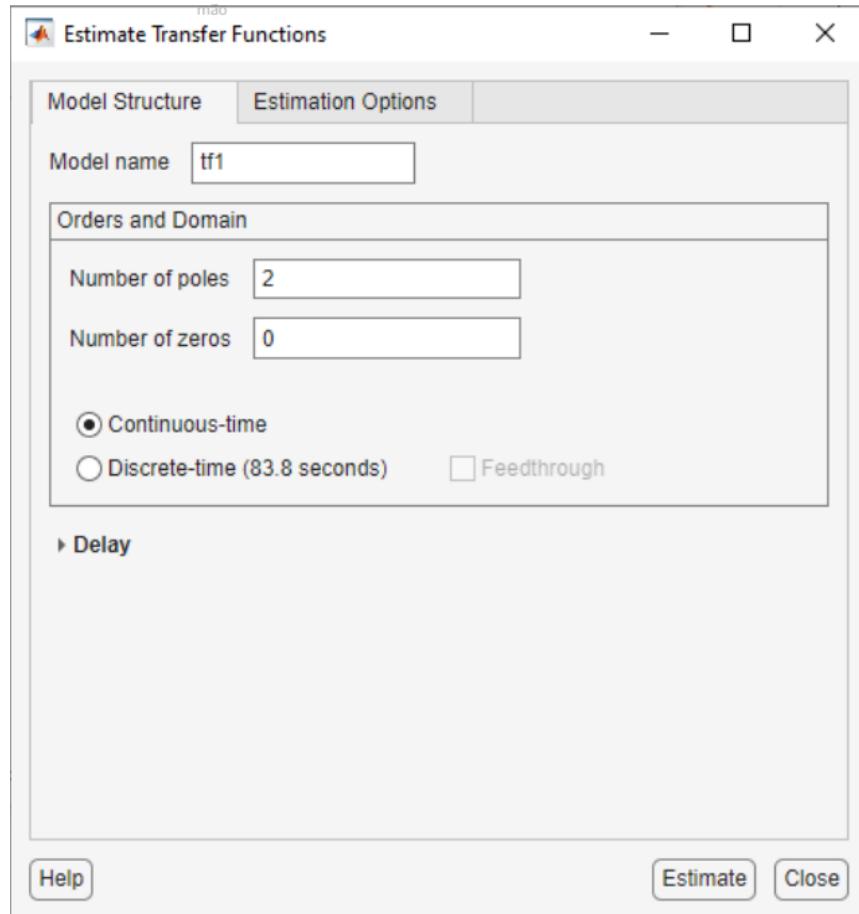


Figura 40: Configurações do Modelo  
Fonte: O autor

### 3. Análise dos resultados:

Os modelos obtiveram índices de ajuste (*fit%*) superiores a 95% para todos os pistões, indicando boa aderência entre a simulação e os dados experimentais.

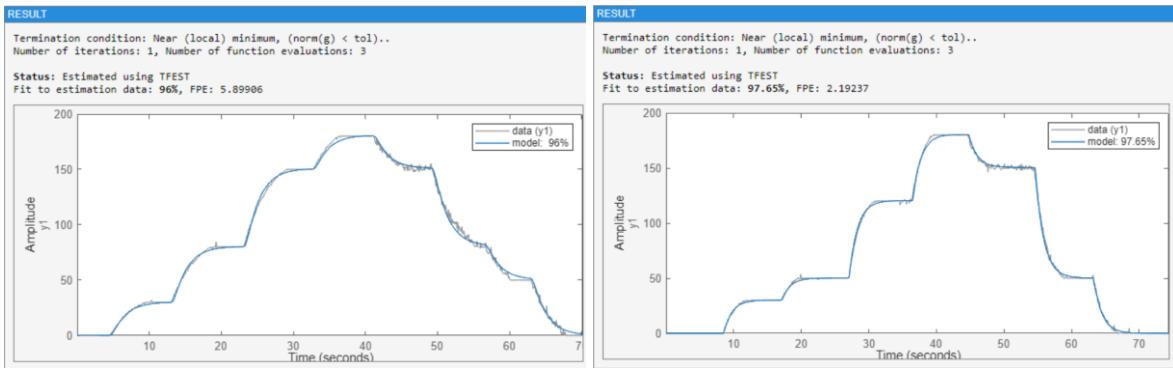


Figura 41: Resultado do Modelo Gerado Pistão 1 e 2  
Fonte: O autor

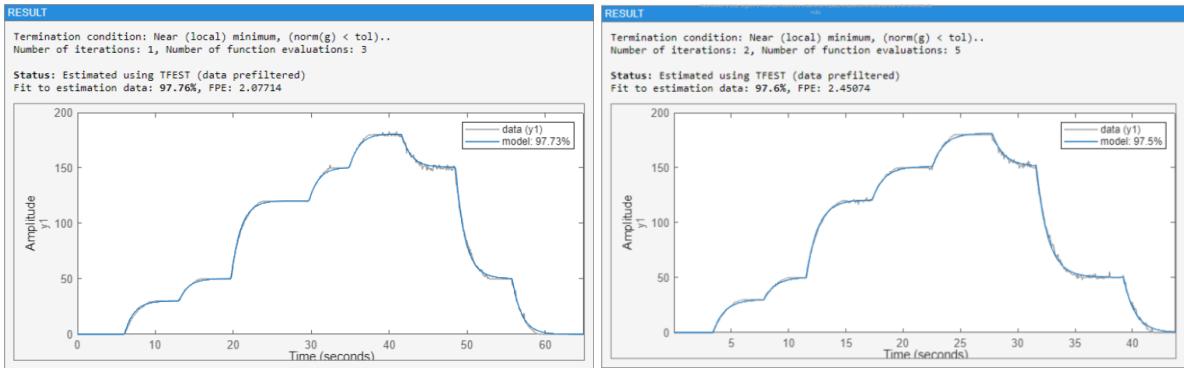


Figura 42: Resultado do Modelo Gerado Pistão 3 e 4  
Fonte: O autor

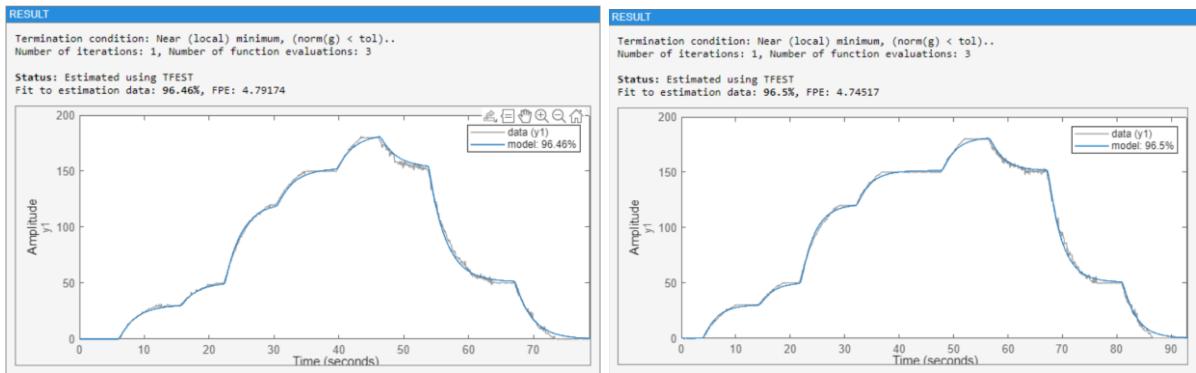


Figura 43: Resultado do Modelo Gerado Pistão 5 e 6  
Fonte: O autor

Funções de Transferência em malha fechada obtidas:

$$G_1(s) = \frac{1,425}{s^2 + 3,146s + 1,417} \quad (21)$$

$$G_2(s) = \frac{11,52}{s^2 + 14,85s + 11,47} \quad (22)$$

$$G_3(s) = \frac{13,36}{s^2 + 16,93s + 13,3} \quad (23)$$

$$G_4(s) = \frac{47,21}{s^2 + 48,5s + 46,85} \quad (24)$$

$$G_5(s) = \frac{2,674}{s^2 + 7,244s + 2,625} \quad (25)$$

$$G_6(s) = \frac{1,311}{s^2 + 3,627s + 1,298} \quad (26)$$

Apesar dos resultados satisfatórios obtidos em simulação, a aplicação prática dos ganhos obtidos a partir dessa FTs (Tabela 3) na plataforma não apresentou o desempenho esperado.

Mesmo com o uso do filtro passa-baixa, o nível de ruído nas medições inviabilizou o controle em malha fechada utilizando o PID, comprometendo a estabilidade do sistema.

Diante dessa limitação, na nova estratégia experimental, observou-se que a resposta da planta apresentava comportamento superamortecido, isto é, sem sobressinal e com dinâmica dominada por um único polo. Essa característica permitiu aproximar o sistema por um modelo de primeira ordem, prática comum quando um polo dominante descreve a maior parte da resposta e os demais efeitos podem ser representados por atrasos efetivos ou perturbações (GARCIA, 2018).

A partir dessa reformulação do modelo, tornou-se possível aplicar técnicas clássicas de identificação e sintonia, como o método de Ziegler–Nichols por curva de reação, em conjunto com o *System Identification Toolbox* do MATLAB, obtendo parâmetros mais aderentes ao comportamento dinâmico real da plataforma.

Simultaneamente, para cada pistão, foram aplicados quatro degraus de entrada em diferentes níveis, registrando-se as respostas individuais. Com base nessas respostas, empregou-se o método de Ziegler–Nichols para determinar os parâmetros característicos da curva (constante de tempo e atraso efetivo) e, a partir deles, os ganhos iniciais de controle.

Como parte da metodologia, foi adotada uma estratégia comparativa, buscando identificar a melhor representação dinâmica do sistema. Assim, foram avaliadas quatro formas de obtenção da função de transferência média: considerando todos os degraus aplicados, considerando apenas os três últimos degraus, considerando apenas os dois últimos degraus e comparando individualmente cada resposta.

O objetivo dessa abordagem foi verificar qual conjunto de dados resultaria no maior percentual de ajuste ( $fit\%$  em relação à resposta experimental.

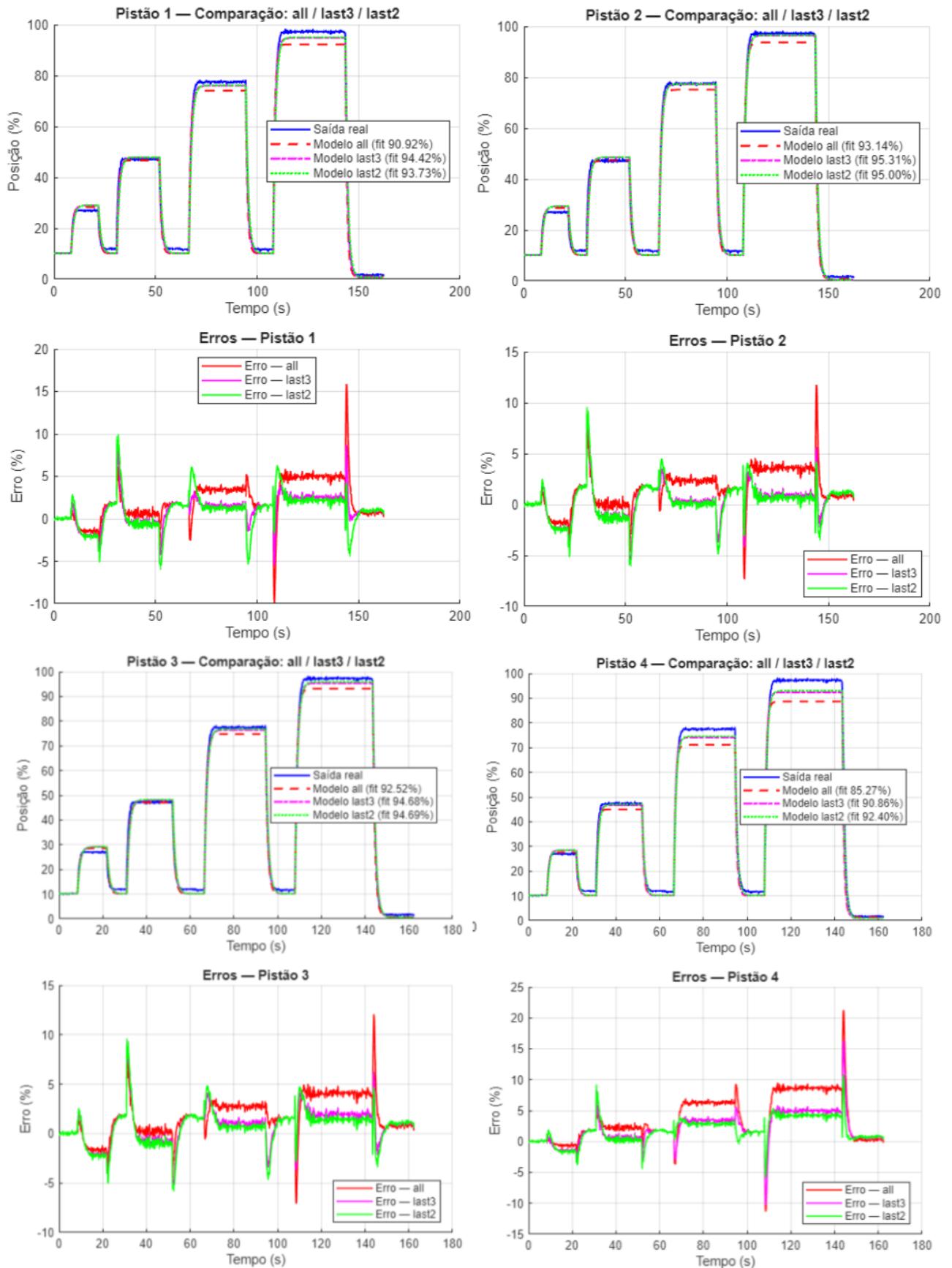


Figura 44: Pistão 1, 2, 3 e 4 – Comparaçao entre as FTs e Erro ao longo do tempo  
Fonte: O autor

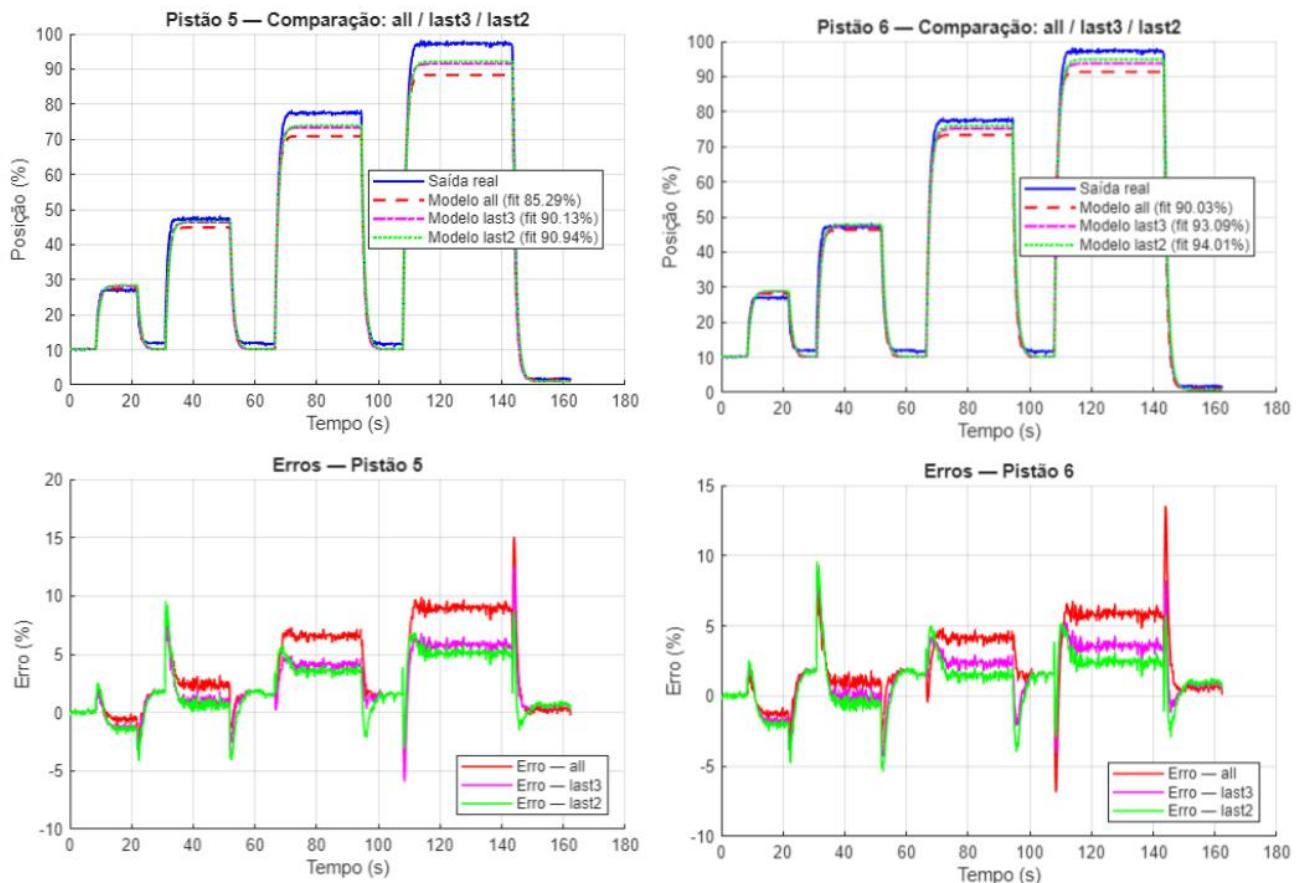


Figura 45: Pistão 5 e 6 – Comparação entre as FTs e Erro ao longo do tempo

Fonte: O autor

Dessa forma, a combinação entre o método clássico de Ziegler–Nichols e a identificação numérica pelo *System Identification Toolbox* permitiu obter modelos representativos e coerentes com o comportamento real dos atuadores, possibilitando a definição precisa dos ganhos de controle aplicados nos ensaios em bancada.

A escolha por um modelo de primeira ordem, mesmo apresentando um *fit* inferior ao modelo de segunda ordem identificado inicialmente, fundamenta-se em princípios consolidados de engenharia de controle. Em síntese, modelos não devem ser avaliados apenas pelo índice de ajuste, mas pela sua capacidade de representar a dinâmica dominante da planta de maneira robusta e implementável. O objetivo da modelagem, nesse contexto, não é a precisão matemática absoluta, mas sim obter um modelo que seja simples o suficiente para permitir análise e projeto, e complexo o suficiente para descrever as propriedades essenciais do sistema (HERNÁNDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019).

O modelo de segunda ordem, apesar do *fit* elevado, apresentou dinâmicas adicionais que não correspondiam ao comportamento físico dominante do atuador, resultando em uma estrutura matemática mais complexa, mas sem benefícios práticos para o projeto de controle.

Em sistemas reais, componentes matemáticos associados a polos não dominantes podem ser descartados em favor de modelos reduzidos que capturam apenas a dinâmica essencial (GARCIA, 2018; OGATA, 2010; HERNÁNDEZ-GUZMÁN; SILVA-ORTIGOZA, 2019).

Além disso, a escolha de modelos reduzidos está diretamente associada à robustez. Modelos de ordem elevada tendem a introduzir polos e zeros adicionais que, embora melhorem o ajuste numérico, ampliam a sensibilidade do controlador a ruído e variações operacionais, fatores que estão presentes nos atuadores e estrutura que estão sendo utilizados. Anteriormente, na identificação de segunda ordem, foi constatado que, por conta de ruídos, não foi possível realizar o controle.

Por fim, o critério decisivo foi a validação prática: o controlador projetado a partir do modelo de primeira ordem foi o que efetivamente funcionou em bancada, apresentando resposta mais estável e previsível. Assim, embora o modelo de segunda ordem tivesse maior ajuste numérico, o modelo de primeira ordem mostrou-se superior para fins de controle, por representar adequadamente a dinâmica dominante e proporcionar desempenho robusto na implementação real.

Funções de transferência obtidas de primeira ordem em malha fechada

$$G_1(s) = \frac{6,968}{s + 7,182} \quad (27)$$

$$G_2(s) = \frac{7,174}{s + 7,308} \quad (28)$$

$$G_3(s) = \frac{6,826}{s + 7,001} \quad (29)$$

$$G_4(s) = \frac{7,038}{s + 7,401} \quad (30)$$

$$G_5(s) = \frac{6,599}{s + 6,892} \quad (31)$$

$$G_6(s) = \frac{6,935}{s + 7,155} \quad (32)$$

#### **4.4.3. FASE 3 - SINTONIA DO CONTROLE PID.**

Com as funções de transferência  $T_1(s), T_2(s), \dots, T_6(s)$  identificadas, a malha foi aberta em simulação para possibilitar a sintonia individual dos controladores.

Considerando que as respostas experimentais haviam sido obtidas em malha fechada com realimentação unitária, a relação utilizada para determinar o modelo de malha aberta foi dada por:

$$G_i(s) = \frac{T_i(s)}{K_p [1 - T_i(s)]} \quad (33)$$

onde  $T(s)$  representa a função de transferência identificada em malha fechada e  $K_p$  é o ganho proporcional aplicado durante o ensaio.

Dessa forma, para a primeira estratégia, adotou-se  $K_p = 1$ , obtendo-se:

$$G_i(s) = \frac{T_i(s)}{1 - T_i(s)} \quad (34)$$

Com as plantas  $G_i(s)$  obtidas, os controladores foram sintonizados individualmente utilizando o comando pidtune do MATLAB.

Para promover a sincronização entre os atuadores, foi fixada a mesma frequência de projeto (largura de banda desejada)  $\omega_b$  para todos os pistões. Dessa forma, ainda que existissem pequenas diferenças dinâmicas entre as plantas  $G_i(s)$ , os controladores PID resultantes apresentaram constantes de tempo de malha fechada semelhantes, aproximando os tempos de subida e de acomodação.

Em seguida, foram obtidos os ganhos ( $K_p, K_i, K_d$ ) de cada pistão, fechando-se as respectivas malhas conforme:

$$T_i(s) = \frac{C_i(s)G_i(s)}{1 + C_i(s)G_i(s)} \quad (35)$$

Tabela 3: Ganhos do Controlador PID para Cada Pistão

Pistão	$K_p$	$K_i$	$Kd$
1	1.7829	0,0249	0,5175
2	1.0333	0,0145	0,0223
3	1.0152	0,0142	0,0165
4	0.8221	0,0143	0,0000
5	2.1773	0,0304	0,2310
6	2.2380	0,0312	0,5083

Fonte: O autor

As respostas a um degrau de 30 mm foram simuladas para cada controlador, e as curvas  $T_1(s), T_2(s), \dots, T_6(s)$  foram comparadas no mesmo gráfico para avaliar o alinhamento temporal e garantir sobressinal desprezível.

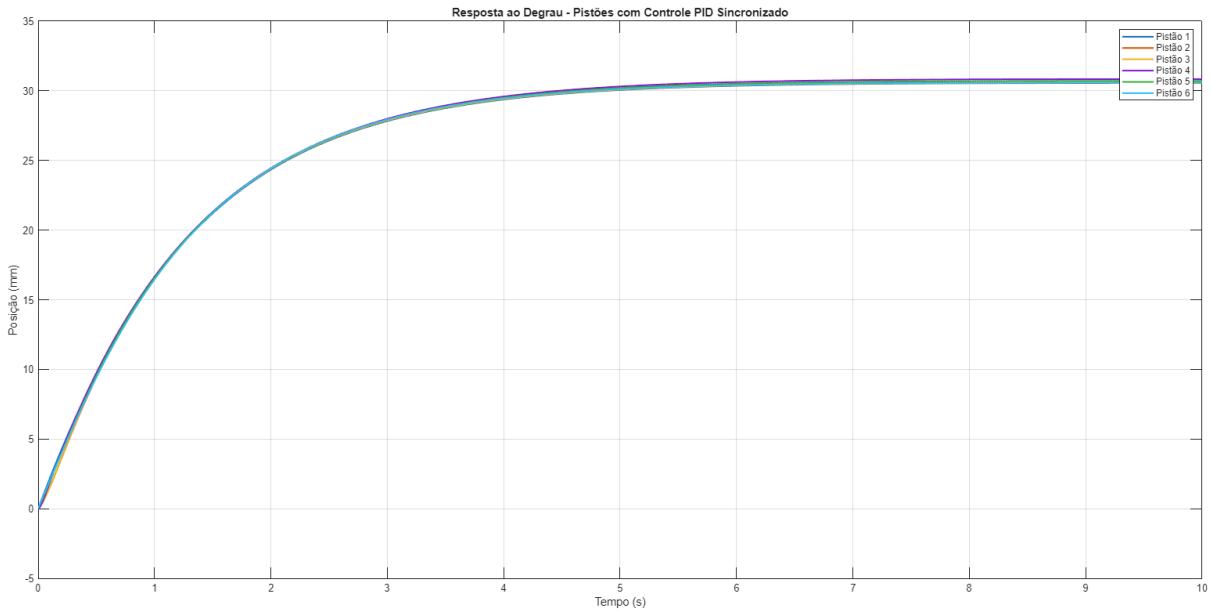


Figura 46: Resultado Obtido – Curva 6 Pistões Sobrepostas para Comparação  
Fonte: O autor

Quando necessário, realizaram-se ajustes finos na frequência de projeto  $\omega_b$  para reduzir diferenças residuais de atraso e tempo de acomodação entre os canais.

Por fim, a validação dos controladores foi realizada no *Control System Designer*, confirmando o atendimento aos critérios de rastreamento de referência e estabilidade definidos no projeto.

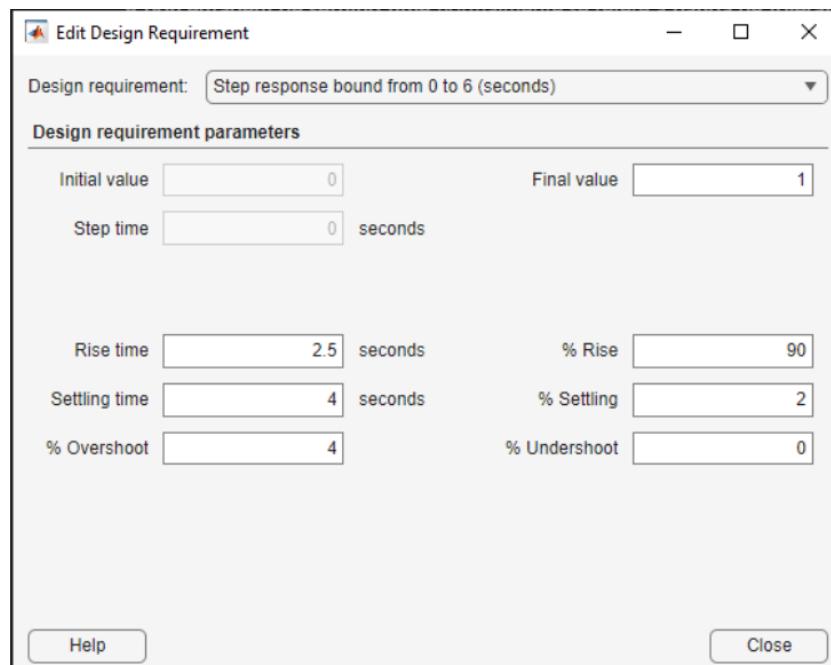


Figura 47: Especificações do Design  
Fonte: O autor

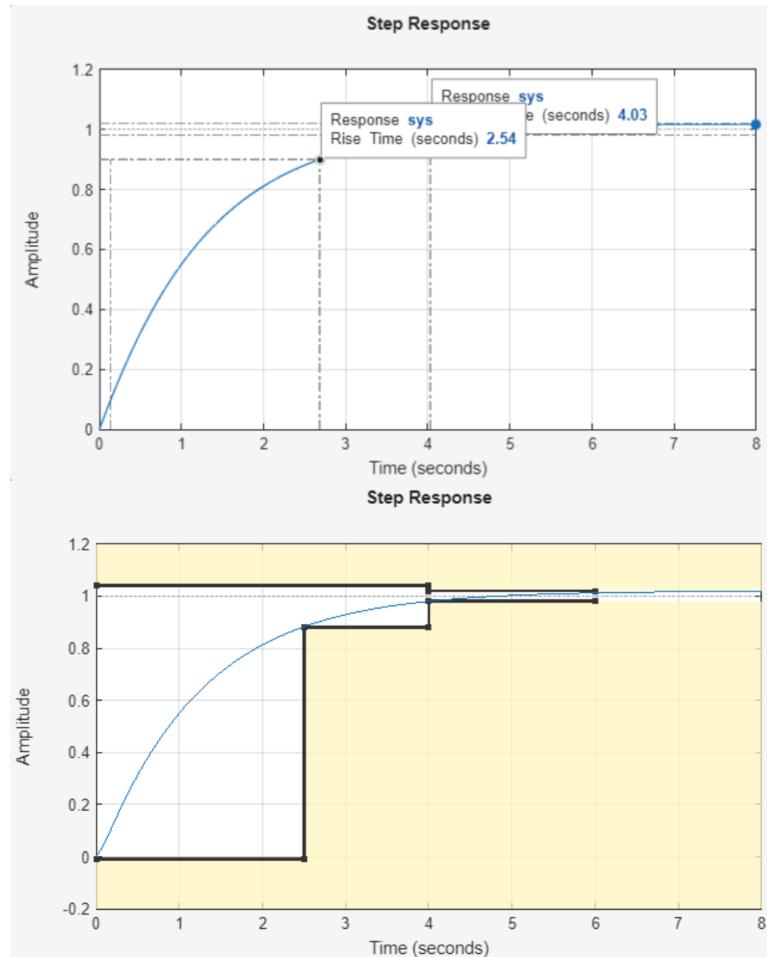


Figura 48: Resultado Desejado  
Fonte: O autor

Na segunda estratégia, o procedimento de abertura de malha foi mantido, porém com ganho proporcional  $K_p = 5$ , que durante a fase de aquisição, como citado anteriormente, resultou em respostas mais lineares.

Assim, a relação para abertura de malha passou a ser:

$$G_i(s) = \frac{T_i(s)}{5 [1 - T_i(s)]} \quad (36)$$

Nessa etapa, ao realizar novos testes em bancada, observou-se que, com  $K_p = 5$  e  $K_i = 0,4$ , obtiveram-se respostas satisfatórias em todos os pistões, os quais atingiram o setpoint de forma praticamente sincronizada.

Também foi verificado que o Pistão 2 apresentou a melhor resposta experimental, sendo, portanto, definido como referência de desempenho. Dessa forma, todos os demais pistões foram sintonizados para seguir o comportamento dinâmico de  $G_2(s)$  sob o controlador  $C_2(s)$ .

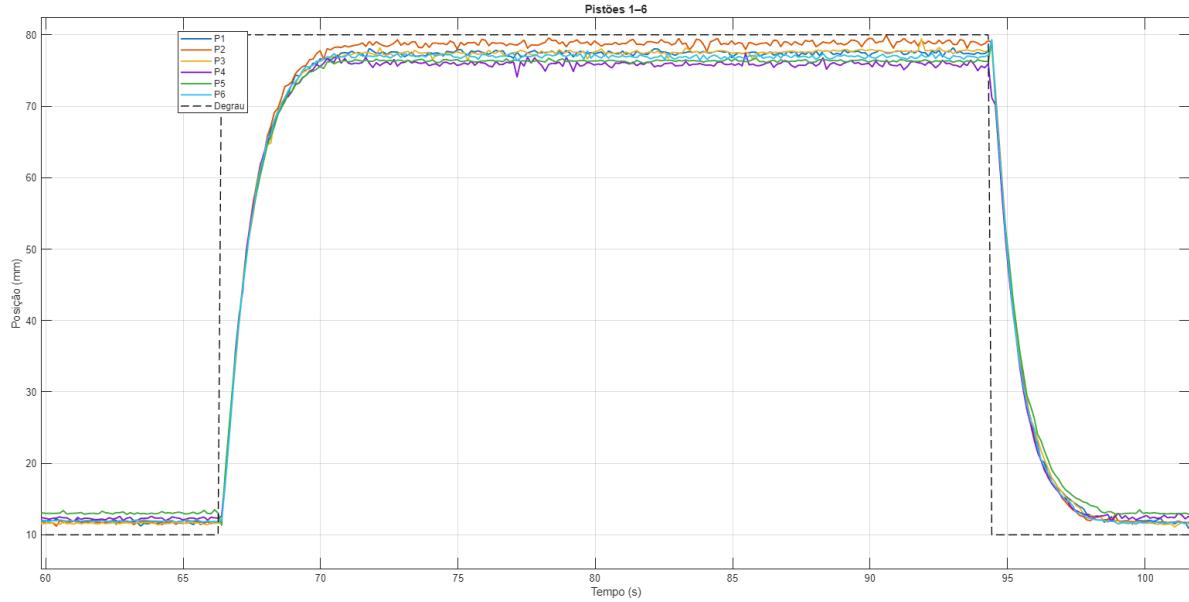


Figura 49: Resposta do ensaio  
Fonte: O autor

Com base nesse resultado, definiu-se o controlador de referência  $C_2(s)$  como:

$$C_2(s) = \frac{5s + 0,4}{s} \quad (37)$$

A partir dessa configuração, todos os demais pistões foram ajustados de modo a reproduzir o comportamento dinâmico do Pistão 2, utilizando a relação:

$$C_i(s) = \frac{G_2(s) C_2(s)}{G_i(s)} \text{ para } i \in \{1,3,4,5,6\} \quad (38)$$

Essa expressão permite que cada controlador  $C_i(s)$  seja ajustado de forma a compensar as diferenças dinâmicas entre as plantas, fazendo com que todas as malhas de controle apresentem respostas temporais equivalentes àquela do pistão de referência  $G_2(s)$ .

Em termos de implementação, cada controlador foi expresso na forma proporcional-integral (PI), obtida pela decomposição do numerador e denominador de  $C_i(s)$ :

$$C_{PI}(s) = \frac{K_p s + K_i}{s} \quad (39)$$

onde  $K_p$  e  $K_i$  são, respectivamente, os ganhos proporcional e integral determinados a partir da fatoração de  $C_i(s)$ ;

Essa metodologia garantiu que todos os atuadores seguissem o mesmo perfil de resposta de referência, promovendo sincronização entre os pistões.

Antes da sincronização, era possível observar pequenas diferenças entre o tempo de subida (Tr) e o tempo de acomodação (Ts) de cada pistão, além de variações de erro em regime permanente.

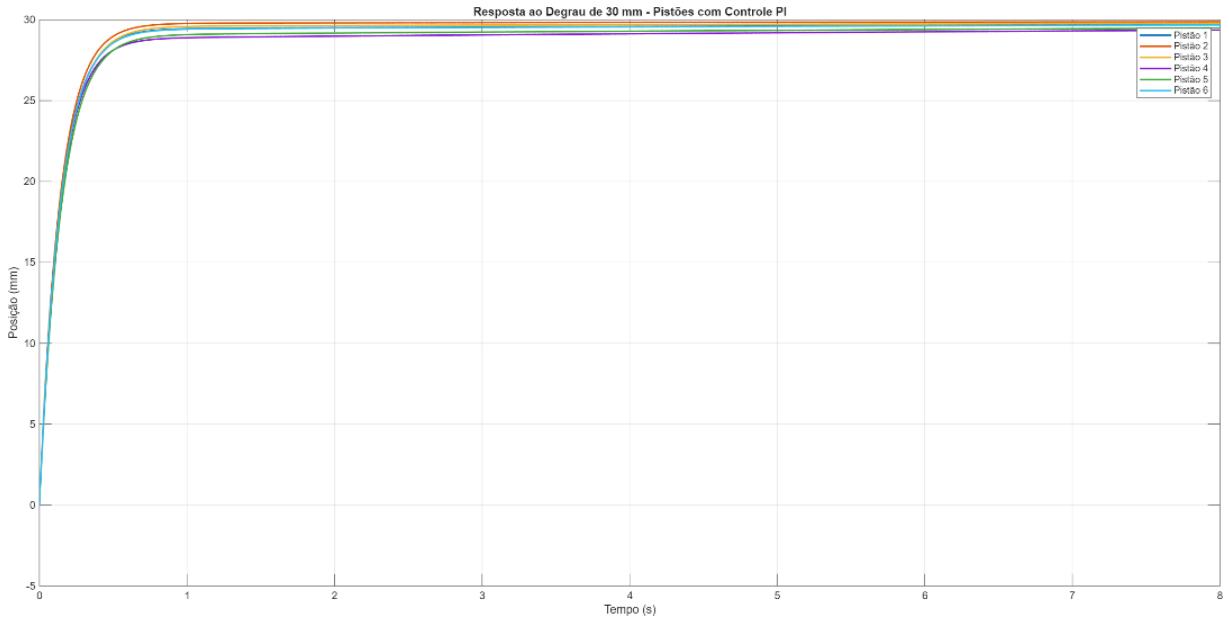


Figura 50: Resposta antes da sincronização – Visão geral  
Fonte: O autor

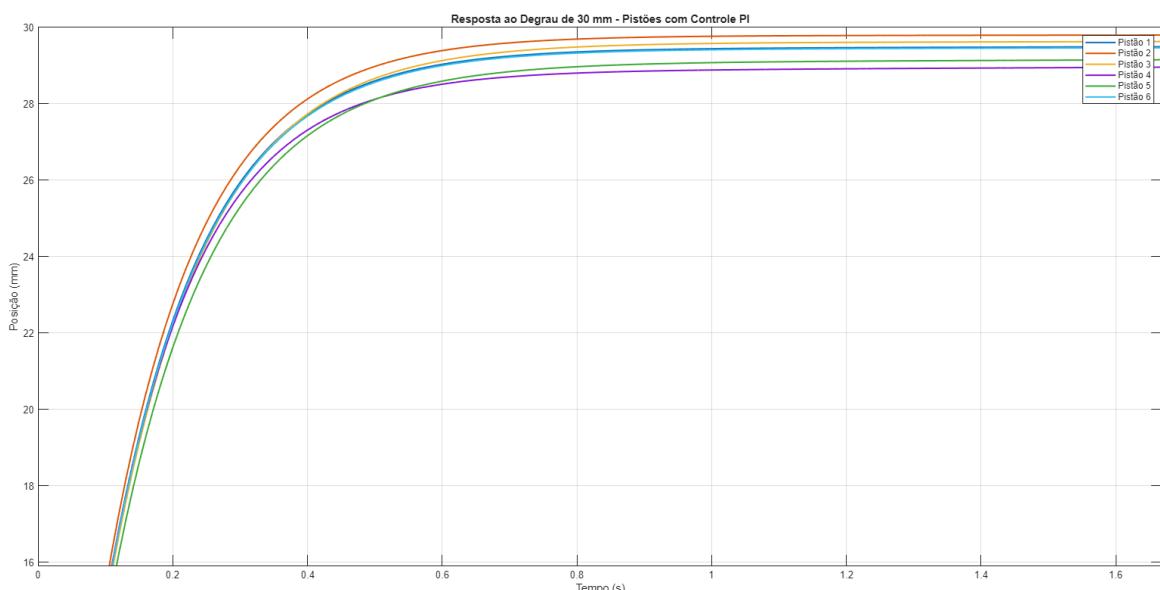


Figura 51: Resposta antes da sincronização – zoom Tr e Ts  
Fonte: O autor

Após a aplicação da estratégia de sincronização, o sistema passou a apresentar movimentos mais suaves, estáveis e coerentes entre os seis atuadores, consolidando essa abordagem como a mais adequada para o controle coordenado da plataforma de Stewart.

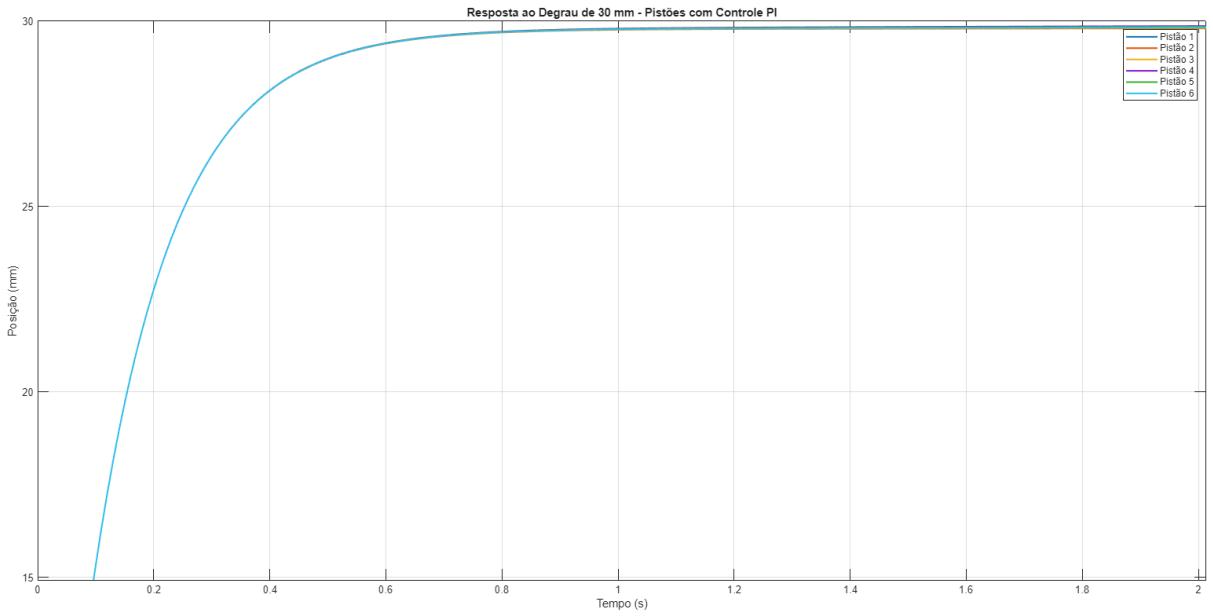


Figura 52: Resposta após sincronização – Visão geral  
Fonte: O autor

Todos os controladores apresentaram estrutura proporcional-integral (PI), uma vez que o termo derivativo ( $K_d$ ) foi mantido nulo em todas as sintonias, atendendo ao comportamento superamortecido da planta e garantindo robustez frente aos ruídos de medição.

Os ganhos obtidos para cada controlador  $C_i(s)$  são apresentados a seguir:

Tabela 4: Ganhos do Controlador PI para Cada Pistão

Pistão	$K_p$	$K_i$
1	5,1478	0,8226
2	5,0000	0,4000
3	5,2552	0,6391
4	5,0969	1,5752
5	5,4362	1,3039
6	5,1724	0,8593

Fonte: O autor

Observa-se que os valores de  $K_p$  permanecem próximos de 5 em todos os atuadores, o que reforça a uniformidade dinâmica entre os pistões após a aplicação da estratégia de equalização. Já os valores de  $K_i$  apresentaram pequenas variações decorrentes das diferenças de polos identificadas entre os pistões, permitindo ajustes finos no rastreamento de referência e na compensação de erro em regime permanente.

Durante a validação em bancada, contudo, verificou-se que a presença de saturação no sinal de comando (limitado pelo PWM e pelas zonas mortas dos atuadores) associada ao

acoplamento cinemático da plataforma de Stewart tornava o sistema suscetível ao fenômeno de *integral windup*. Em determinadas poses, alguns atuadores não conseguiam eliminar completamente o erro devido às restrições geométricas e à interação com os demais pistões, o que levava o termo integral a acumular erro mesmo com o comando já saturado. Para mitigar esse efeito, foi implementada uma estratégia de *anti-windup* do tipo *back-calculation*, na qual a diferença entre o sinal de controle antes e depois da saturação é realimentada ao integrador, forçando-o a acompanhar a saída saturada e evitando o crescimento não controlado do termo integral. Adicionalmente, na região de banda morta em torno do setpoint, adotou-se um “vazamento” controlado do integrador, promovendo o decaimento gradual do termo integral quando o erro permanece pequeno.

Após a aplicação destes ganhos na bancada e a execução de uma nova rodada de testes com ajustes empíricos para refinar a resposta transitória e eliminar assimetrias, obteve-se a versão final da sintonia, definida por:

Tabela 5: Ganhos Revisados do Controlador PI para Cada Pistão

Pistão	$K_p$	$K_i$
1	5,1478	0,8226
2	5,2000	0,7000
3	5,2552	0,6391
4	5,0969	0,8000
5	5,4362	1,1240
6	5,1724	0,8593

Fonte: O Autor

Essa sintonia final, em conjunto com a estratégia de *anti-windup*, consolidou o comportamento sincronizado da plataforma, garantindo tempos de subida semelhantes e erro estacionário próximo de zero para todos os atuadores, conforme ilustrado na Figura 69.

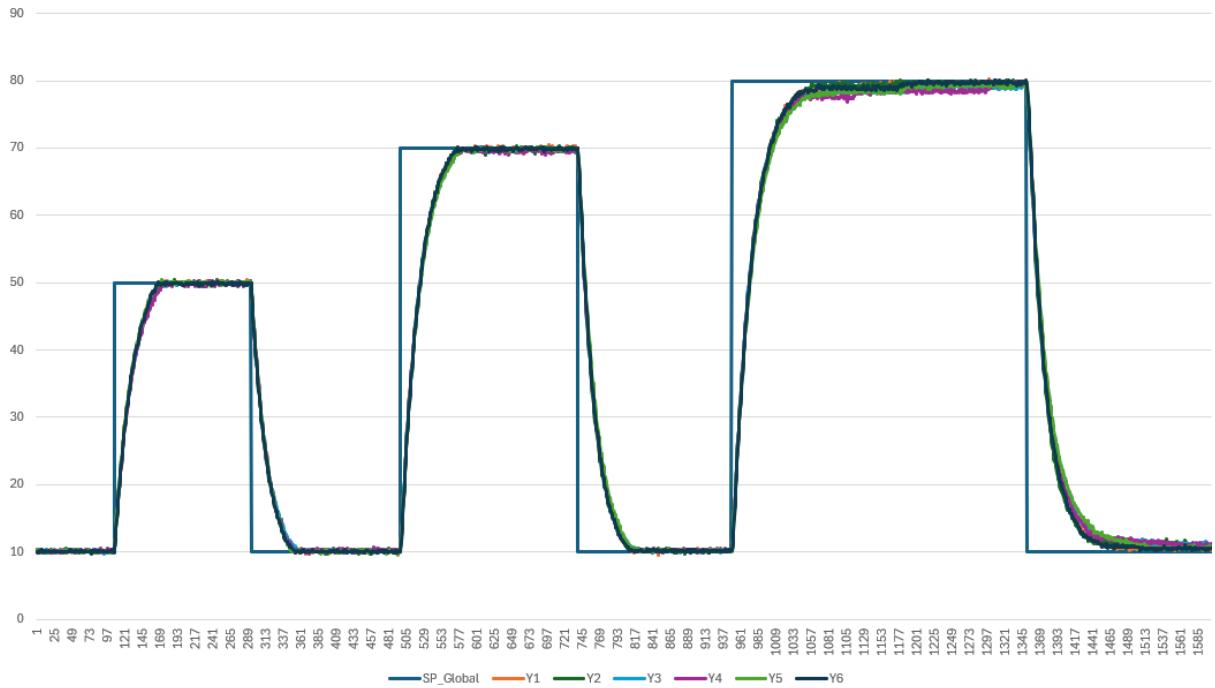


Figura 53: Resposta ao Degrau Segunda Estratégia  
Fonte: O autor

## 4.5. DESENVOLVIMENTO DA INTERFACE DE CONTROLE E MODOS DE OPERAÇÃO

### 4.5.1. ARQUITETURA DE COMUNICAÇÃO

A solução adotada para o controle da plataforma utiliza uma arquitetura distribuída composta por três principais elementos (conforme Anexo I): um frontend estático em HTML/JavaScript, um backend desenvolvido em FastAPI e os firmwares embarcados nos módulos ESP32-S3 e ESP32-IMU.

O frontend interage com o backend por meio de requisições REST, utilizadas para operações síncronas, tais como iniciar rotinas, ajustar parâmetros de PID, definir setpoints e gerenciar a interface serial. Para transmissão contínua de dados, foi implementado o canal WebSocket /ws/telemetry, responsável por entregar pacotes de telemetria ao navegador com baixa latência, permitindo a atualização imediata de gráficos, indicadores e painéis de diagnóstico.

O backend mantém um gerenciador serial baseado na biblioteca pyserial, encarregado de traduzir requisições REST em comandos enviados ao ESP32-S3. Esse módulo realiza a supervisão da porta serial, valida limites operacionais e normaliza unidades antes que qualquer comando seja enviado ao firmware, prevenindo erros de operação. A telemetria enviada pelo ESP32-S3 segue o formato CSV, contendo valores de posição dos pistões, níveis de PWM, e

medidas dos sensores de feedback de cada pistão. Esses pacotes são processados e retransmitidos pelo WebSocket, garantindo que o frontend mantenha uma visão atualizada do comportamento da plataforma.

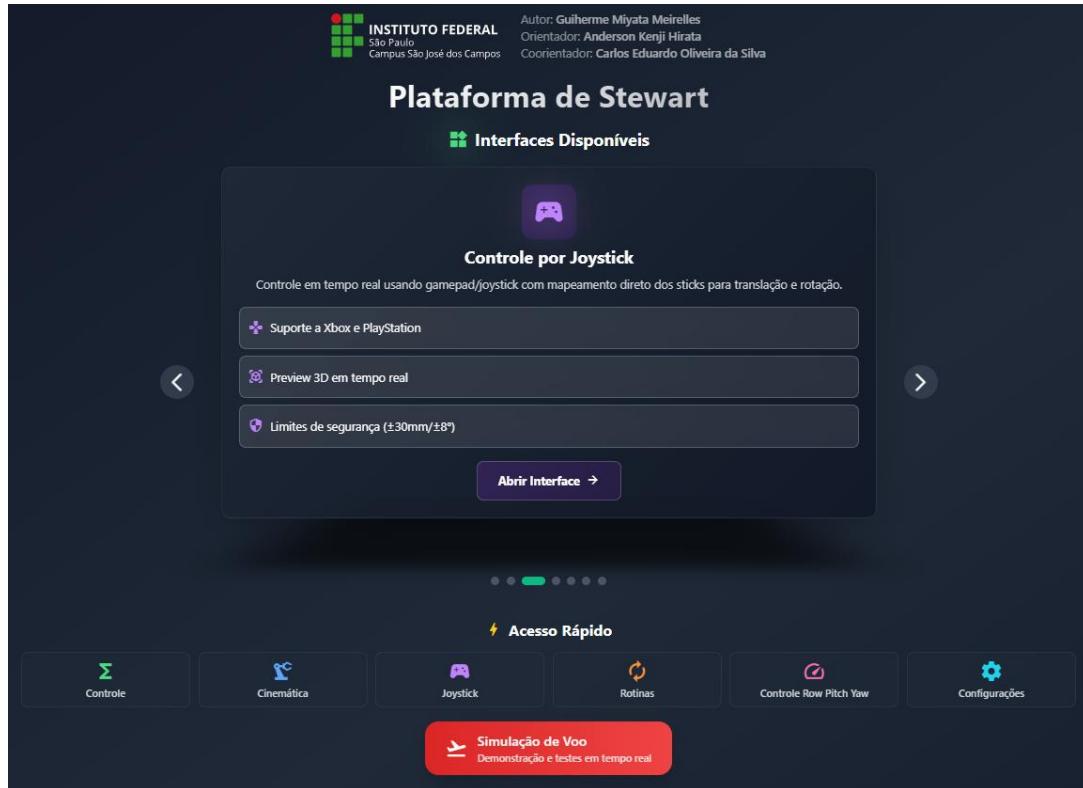


Figura 54:Página Inicial  
Fonte: O autor

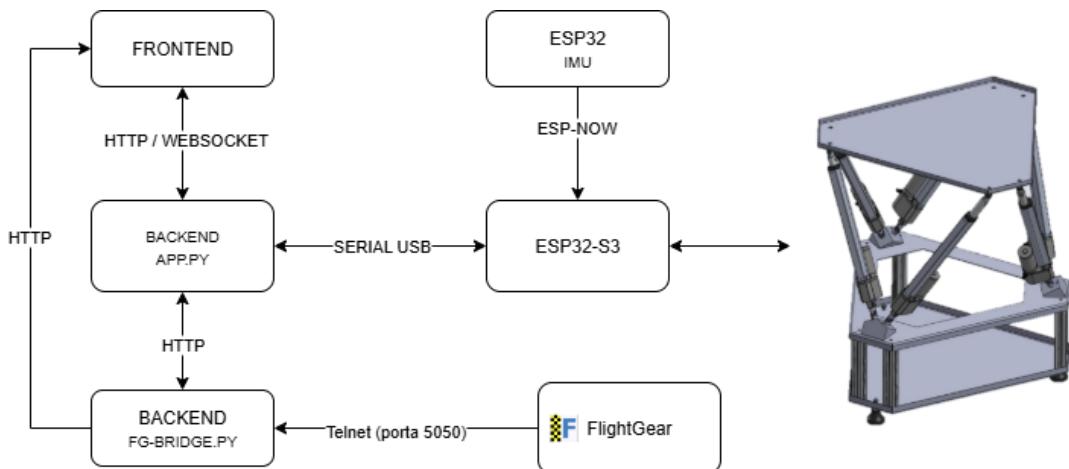


Figura 55: Fluxo de Dados  
Fonte: O autor

Tabela 6: Endpoints de Comunicação

Método	Endpoint	Descrição
GET	/serial/ports	Lista as portas seriais disponíveis no sistema hospedeiro.

POST	/serial/open	Abre a conexão serial com o ESP32-S3 (porta e baud rate).
POST	/serial/close	Fechá a conexão serial atualmente aberta.
GET	/serial/status	Retorna o status da conexão serial (aberta, porta, baud, erros).
POST	/serial/send	Envia um comando livre em ASCII pela serial (debug/baixa nível).
GET	/telemetry	Retorna o último pacote de telemetria recebido do ESP32-S3 em formato JSON.

Fonte: O autor

#### **4.5.1.1. INTEGRAÇÃO COM OS MICROCONTROLADORES**

O ESP32-S3 é o módulo principal de controle (ver Anexo II), responsável por acionar fisicamente os atuadores lineares da plataforma. Ele recebe comandos via serial USB enviados pelo backend (app.py), processa os setpoints recebidos, aplica o controle PID embarcado e retorna telemetria contendo o estado instantâneo de cada pistão, níveis de PWM e demais medições internas relevantes para supervisão.

Em paralelo, um segundo módulo, o ESP32-IMU, coleta dados de sensores inerciais (como BNO085 ou MPU6050) e transmite essas informações utilizando ESP-NOW, permitindo comunicação direta com o ESP32-S3 sem depender de roteadores ou infraestrutura Wi-Fi. Esse módulo envia quaternions ou ângulos de Euler, possibilitando que o sistema opere em modo IMU para fins de visualização, testes funcionais e futuras funcionalidades relacionadas à estabilização da plataforma.

#### **4.5.1.2. COMUNICAÇÃO COM FLIGHTGEAR**

Além do controle local, o backend executa um segundo processo, denominado fg-bridge.py, responsável pela integração com o simulador de voo FlightGear. Essa comunicação é estabelecida via Telnet (porta 5050), por meio da qual o sistema lê continuamente as variáveis de orientação disponibilizadas pelo simulador, como Roll e Pitch. Esses valores passam por um processo de limitação e filtragem e, em seguida, são encaminhados ao backend principal, que alimenta o gerador de poses da plataforma.

Dessa forma, a plataforma física é capaz de reproduzir em escala reduzida os movimentos provenientes do simulador, permitindo testar cenários de voo, dinâmica de aeronaves e comportamentos oscilatórios diretamente no hardware.

#### **4.5.2. IMPLEMENTAÇÃO PAINEL PID**

A fim de realizar testes e validar o desempenho do controle PID aplicado aos pistões da plataforma, foi desenvolvida uma página dedicada com um painel de monitoramento em tempo real (ver Anexo III). Essa interface consome dados via WebSocket e utiliza a biblioteca gráfica

Chart.js para renderizar séries temporais de deslocamento e PWM, possibilitando a visualização contínua da resposta dos atuadores.



Figura 56: Gráfico Controle PID – Chart.js

Fonte: O autor

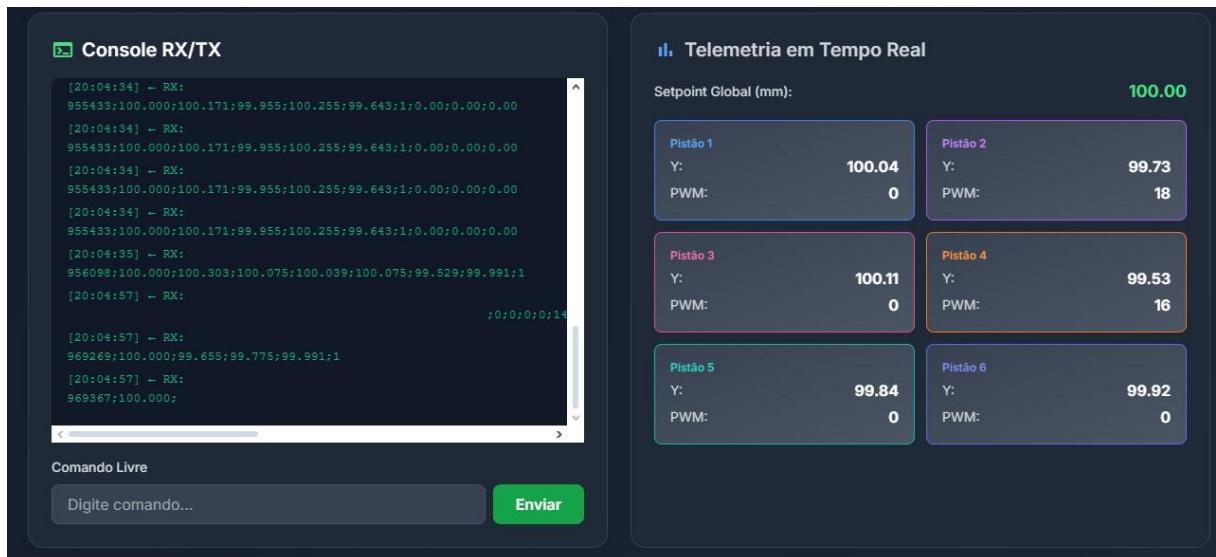


Figura 57: Dados do Websocket

Fonte: O autor

A página permite a execução de testes com setpoints globais ou individuais, registrando simultaneamente o valor de referência e a resposta real do sistema. Dessa forma, torna-se possível analisar o comportamento dinâmico da malha de controle de maneira prática e intuitiva. Para investigações mais detalhadas, os dados exibidos podem ser exportados em

formato CSV, facilitando análises offline em softwares como MATLAB, Python ou ferramentas equivalentes.

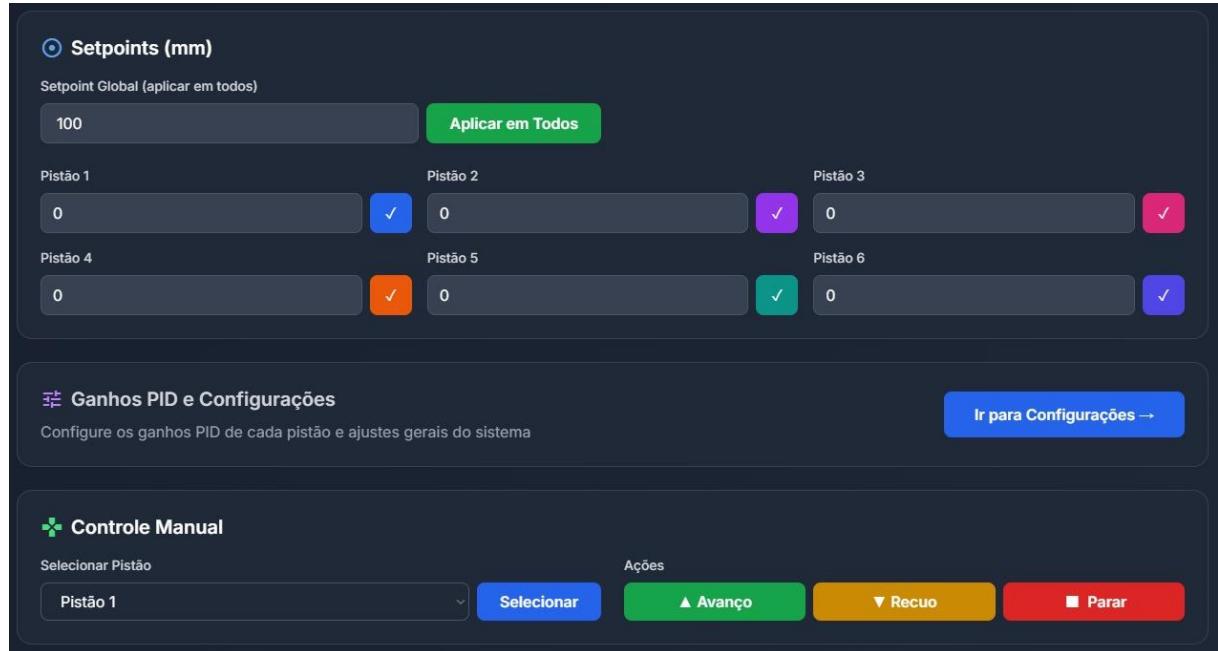


Figura 58: Controle de Setpoints

Fonte: O autor

Além do acompanhamento gráfico, o painel oferece acesso rápido aos ajustes de  $K_p$ ,  $K_i$  e  $K_d$ , por meio da função “Ir para Configurações” (ver Anexo IV). Os valores configurados são enviados ao backend pelo endpoint /pid/gains. O backend, por sua vez, persiste esses parâmetros, encaminha-os ao firmware e confirma sua aplicação. Parâmetros adicionais — como PWM mínimo, deadband e demais ajustes gerais — podem ser configurados pelo endpoint /pid/settings.

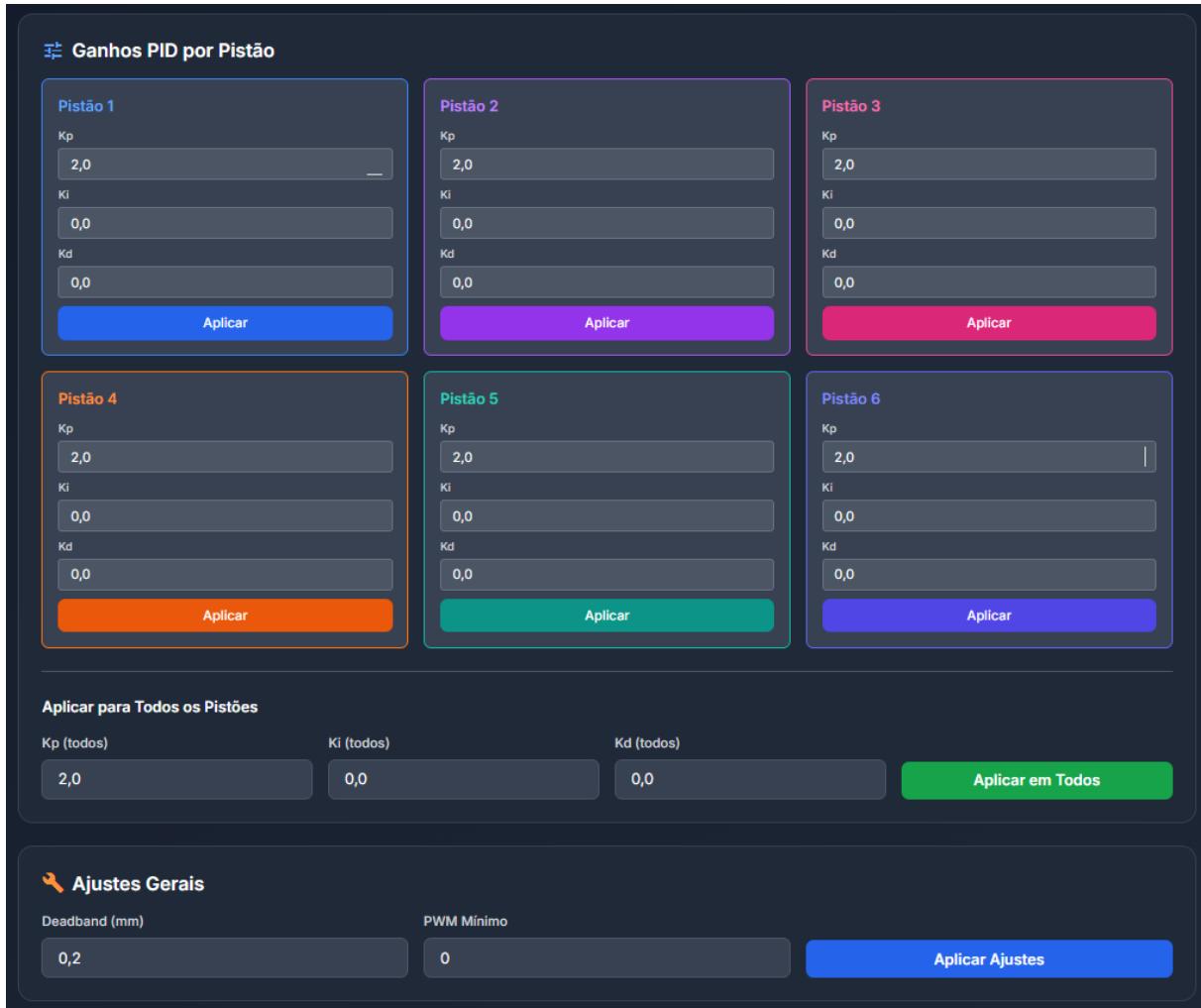


Figura 59: Painel de Ajuste de Ganhos PID  
Fonte: O autor

Tabela 7: Endpoints de Controle PID

Método	Endpoint	Descrição
POST	/pid/setpoint	Define setpoint de posição (mm) global ou para um pistão específico.
GET	/pid/gains	Retorna os ganhos PID de todos os pistões armazenados em cache.
POST	/pid/gains	Define ganhos PID (Kp, Ki, Kd) para um pistão específico.
GET	/pid/gains/{piston}	Retorna os ganhos PID de um pistão específico.
POST	/pid/gains/all	Define simultaneamente Kp, Ki e/ou Kd para todos os pistões.
POST	/pid/feedforward	Define termos de feedforward (u0_adv, u0_ret) para um pistão específico.
POST	/pid/feedforward/all	Define termos de feedforward para todos os pistões.
GET	/pid/settings	Retorna configurações gerais do controlador (deadband, PWM mínimo, etc.).
POST	/pid/settings	Ajusta configurações gerais do PID (deadband, PWM mínimo, etc.).

POST	/pid/manual/{action}	Envia comando manual direto: avanço, recuo ou parada do pistão selecionado.
POST	/pid/select/{piston}	Seleciona qual pistão será afetado pelos comandos manuais.
POST	/pid/offset	Define offset de calibração para um pistão (correção de erro sistemático).

Fonte: O autor

#### 4.5.3. IMPLEMENTAÇÃO DA CINEMÁTICA

A implementação da cinemática inversa no backend foi baseada na formulação vetorial apresentada por Eisele (2019). Essa referência descreve um método direto para calcular os comprimentos dos atuadores a partir de uma pose arbitrária da plataforma, relacionando translação, rotação e vetores de ligação. A partir dessa abordagem, o presente trabalho adaptou o modelo matemático aos parâmetros geométricos reais da plataforma existente no IFSP-SJC, incorporando diretamente os vetores que conectam a base aos pontos de fixação na plataforma móvel (ver Anexo V).

Os parâmetros geométricos utilizados foram obtidos por medições diretas na plataforma original, incluindo as posições dos furos tanto na base quanto na plataforma superior, conforme documentado por Gonçalves (2023) na descrição técnica do protótipo. A tabela a seguir apresenta os pontos medidos (em milímetros), nos referenciais da base (furações F1–F6) e da plataforma superior (FS1–FS6):

Tabela 8: Base (Furações F1–F6)

F	X	Y	Z
F1	305.5	-17	0
F2	305.5	17	0
F3	-137.7	273.23	0
F4	-168	255.7	0
F5	-167.2	-256.2	0
F6	-136.8	-273.6	0

Fonte: O autor

Tabela 9: Plataforma Superior (FS1–FS6)

FS	X	Y	Z
FS1	191.1	-241.5	432
FS2	191.1	241.5	432
FS3	113.6	286.2	432

FS4	-304.7	44.8	432
FS5	-304.7	-44.8	432
FS6	113.1	-286.4	432

Fonte: O autor

As medições dos atuadores indicaram um curso máximo de aproximadamente 250 mm, somado ao comprimento inicial de 500 mm, resultando em um deslocamento total possível de 750 mm. Entretanto, visando segurança operacional e prevenção de colisões mecânicas, o limite superior de trabalho foi restringido a 680 mm no software de controle.

A implementação computacional foi desenvolvida em Python, integrada ao backend FastAPI, empregando NumPy para álgebra linear em tempo real. O endpoint /calculate recebe a pose desejada, posição (x, y, z) e orientação (roll, pitch, yaw), transforma as coordenadas da plataforma móvel, calcula os comprimentos dos seis atuadores, verifica automaticamente violações de curso e retorna ao frontend tanto os valores numéricos quanto os pontos 3D para renderização em Three.js.

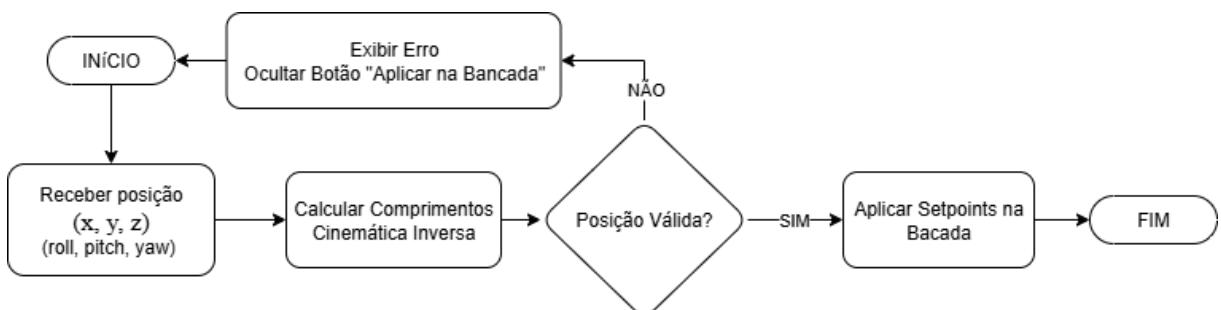


Figura 60: Fluxograma Funcionamento da Cinemática  
Fonte: O autor

```

q = 0.1986 + 0.0298i -0.5391j + 0.8179k   Roll: 0.00 °   Pitch: 0.01 °   Yaw: 0.00 °
  
```

Figura 61: Feedback de IMU – BNO085 ou MPU6050

Fonte: O autor



Figura 62: Controles de Posição  
Fonte: O autor

Além desse cálculo pontual de cinemática inversa, o sistema também executa continuamente a cinemática direta, utilizando os comprimentos reais recebidos em tempo real via telemetria do ESP32-S3. A cada atualização, a postura atual da plataforma é reconstruída e enviada ao frontend, permitindo que o modelo 3D da visualização “Telemetria ao Vivo” represente fielmente o estado físico da bancada. Essa abordagem possibilita a comparação direta entre a pose desejada e a pose realmente executada, integrando o cálculo matemático à operação prática da plataforma.

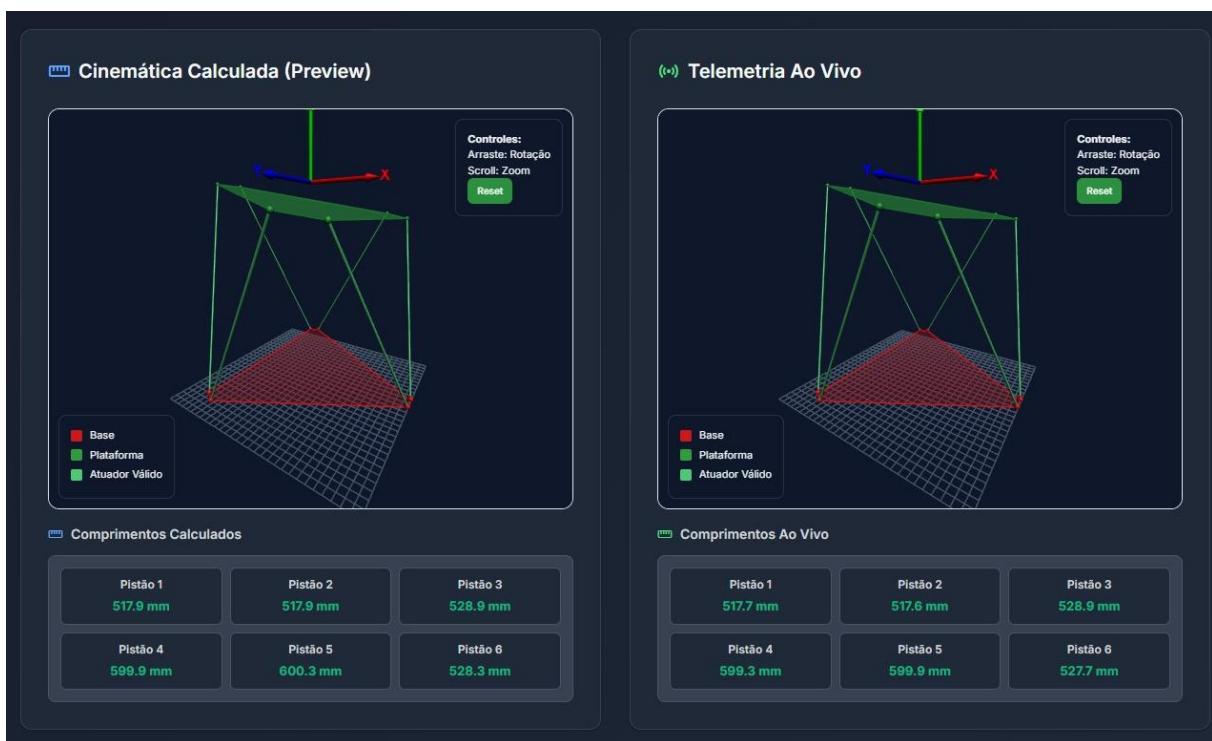


Figura 63: Modelo 3D – Calculado e Real

Fonte: O autor

Essa solução mantém a simplicidade e a eficiência da formulação apresentada por Eisele (2019), porém expandida para operar em conjunto com o sistema embarcado via comunicação serial com o ESP32-S3, suportando tanto pré-visualização quanto reconstrução em tempo real.

Tabela 10: Endpoints Cinemática

Método	Endpoint	Descrição
POST	/calculate	Calcula cinemática inversa: dado uma pose, retorna comprimentos e curso dos atuadores.
POST	/apply_pose	Aplica uma pose: calcula comprimentos e envia comandos para o ESP32-S3.

Fonte: O autor

#### 4.5.4. IMPLEMENTAÇÃO DAS ROTINAS DE MOVIMENTO

As rotinas de movimento da plataforma foram implementadas como geradores parametrizáveis no backend (ver Anexo VI), permitindo a execução de trajetórias senoidais, circulares, helicoidais e combinações lineares e angulares. A interface *routines.html* possibilita ao operador configurar amplitude, frequência, deslocamentos, raios, ciclos verticais, offsets e duração total de cada rotina, de acordo com o tipo de movimento desejado. Uma vez ajustados os parâmetros, a rotina é enviada ao endpoint /motion/start, que inicializa um laço de amostragem periódica responsável por calcular, a cada iteração, a pose alvo correspondente ao instante de tempo. Esse cálculo é processado pelo gerador da rotina e imediatamente validado pelo módulo de cinemática inversa, que verifica limites físicos e restrições geométricas. Com a pose validada, o backend agenda o envio dos comprimentos-alvo ao ESP32-S3 com temporização e, em paralelo, transmite um pacote de dados via WebSocket contendo a pose comandada e a pose real medida pelos sensores. Esses dados permitem que a interface visualize não apenas o comportamento angular e linear da plataforma, mas também o desempenho da malha de controle durante a execução das trajetórias.

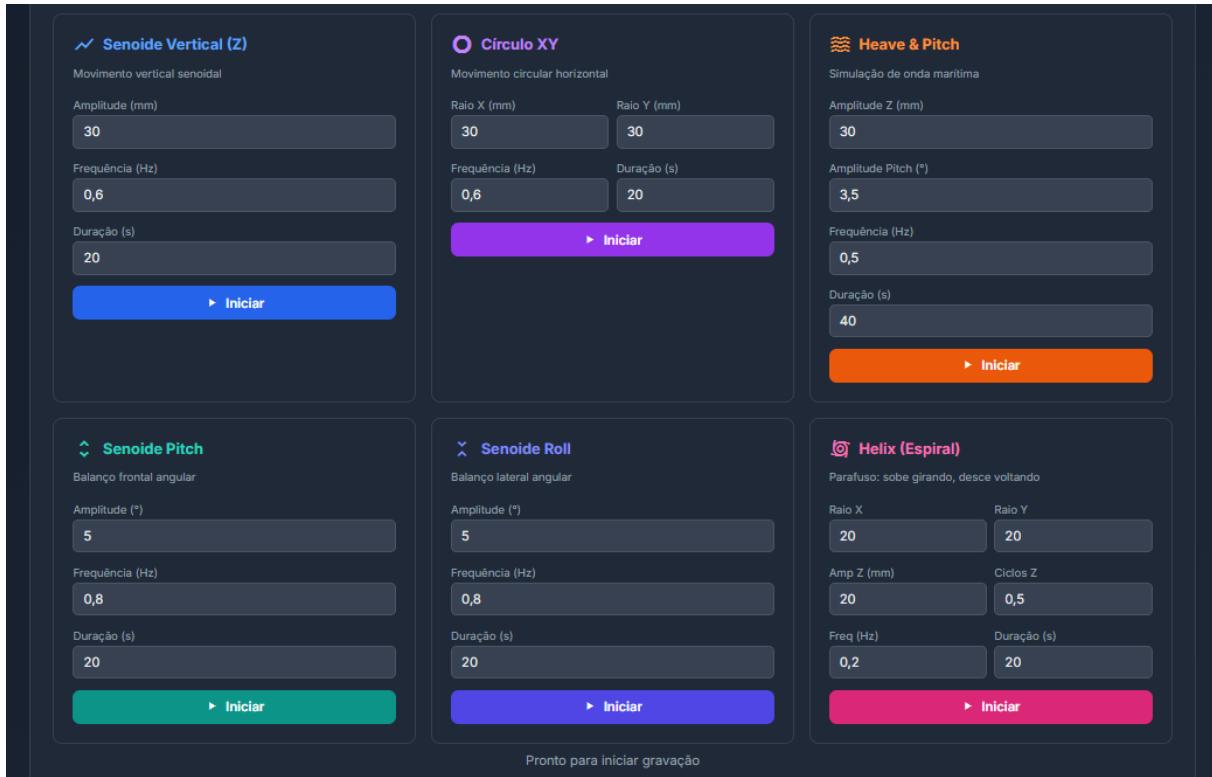


Figura 64: Rotinas Disponíveis  
Fonte: O autor

Além da execução das rotinas em si, o sistema gera gráficos em tempo real com os setpoints calculados para cada um dos seis pistões, possibilitando ao usuário analisar a coerência geométrica da trajetória gerada e o comportamento individual de cada atuador. Os valores exibidos são exatamente aqueles produzidos pelo backend após a aplicação da cinemática inversa, de modo que os gráficos representam a evolução temporal prevista para cada pistão durante a rotina.

A Figura 81 apresenta o comportamento da rotina Circular XY, utilizada para deslocamentos horizontais uniformes da plataforma. Cada pistão descreve uma trajetória senoidal com fases deslocadas entre si, o que é característico de movimentos circulares no plano XY. Essa forma de excitação é típica em aplicações de simulação de voo e navegação, nas quais a plataforma precisa reproduzir translações laterais suaves e contínuas.

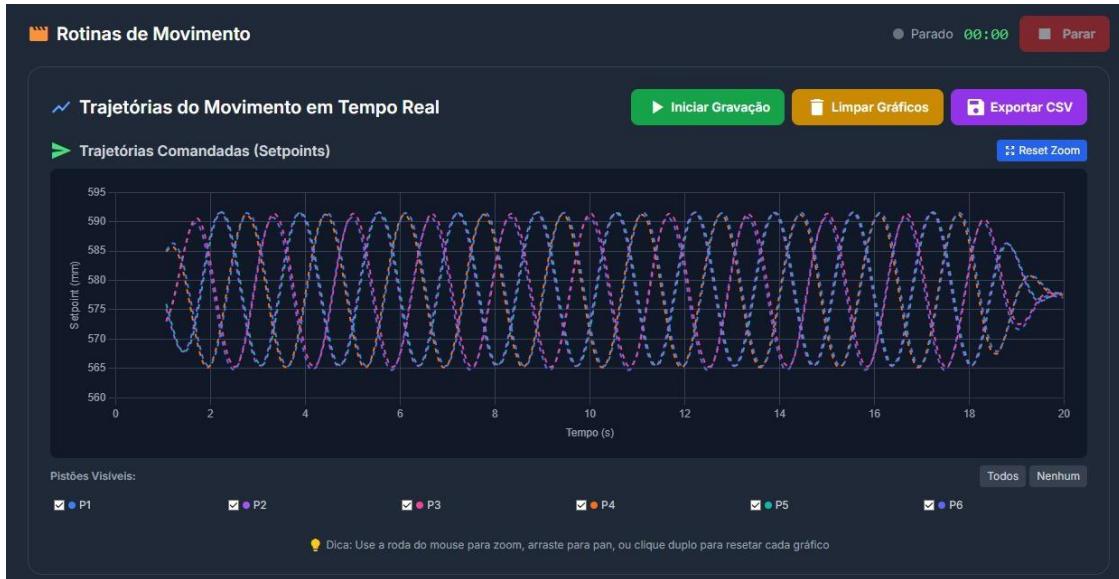


Figura 65: Rotina Circulo XY

Fonte: O autor

A Figura 82 mostra a rotina Senoide Vertical Z, na qual todos os pistões variam de forma sincronizada ao longo do eixo vertical. Esse movimento corresponde ao grau de liberdade conhecido como *heave*, amplamente usado em simuladores de aeronaves e plataformas marítimas para reproduzir subida e descida controladas, vibrações, turbulência vertical ou correções de altitude simulada.



Figura 66: Rotina Senoide Vertical Z

Fonte: O autor

A Figura 83 apresenta a rotina *Heave Pitch* (Onda), que combina elevação vertical com rotação em torno do eixo Y (*pitch*). O resultado é um padrão semelhante a uma onda, essencial

em aplicações de compensação ativa de marés e movimentos de embarcações, onde a plataforma precisa acompanhar ou neutralizar oscilações causadas por ondas incidentes.



Figura 67: Rotina Heave Pitch (Onda)

Fonte: O autor

A Figura 84 exibe a rotina Senoide Pitch (Y), na qual apenas os pistões responsáveis pela orientação no eixo Y variam de forma senoidal, produzindo uma rotação oscilatória longitudinal. Esse tipo de movimento está presente tanto em simuladores de voo, para representar manobras de subida e descida (*pitching*), quanto em sistemas de estabilização embarcados, onde é necessário contrabalançar a proa levantando e abaixando periodicamente.



Figura 68: Rotina Senoide Pitch (Y)

Fonte: O autor

A Figura 85 apresenta a rotina Senoide *Roll* (X), correspondente à oscilação em torno do eixo X. Essa movimentação é fundamental em cenários de simulação aeronáutica (inclinação lateral em curvas) e em plataformas marítimas, nas quais o roll é um dos principais movimentos causados pelas ondas laterais.



Figura 69: Rotina Senoide *Roll* (X)

Fonte: O autor

Por fim, a Figura 86 mostra a rotina Espiral, que combina translação circular no plano XY com variação vertical gradual. Esse movimento mistura componentes de heave, yaw e deslocamento lateral, sendo útil em aplicações avançadas de simulação de navegação, onde o veículo executa trajetórias curvas ascendentes ou descendentes, e em testes de controle multieixos, nos quais é desejável excitar simultaneamente vários graus de liberdade da plataforma.



Figura 70: Rotina Espiral

Fonte: O autor

O sistema também permite iniciar gravações, limpar gráficos e exportar os dados completos em formato CSV, possibilitando análises posteriores em outras ferramentas de processamento.

#### 4.5.5. IMPLEMENTAÇÃO DO CONTROLE POR IMU

O modo de controle baseado em IMU foi desenvolvido para permitir que a plataforma responda diretamente à orientação medida por sensores iniciais (ver Anexo VII), utilizando dispositivos como o BNO085 (ver Anexo VIII) ou o MPU6050 (ver Anexo IX) instalados em um módulo transmissor dedicado. Esses sensores enviam continuamente sua orientação ao ESP32-S3 por meio do protocolo ESP-NOW, garantindo comunicação de baixa latência. O firmware é capaz de interpretar os dados tanto em quaternions quanto em ângulos de Euler (*Roll*, *Pitch* e *Yaw*), permitindo flexibilidade quanto ao tipo de sensor utilizado.

No backend, o modo IMU é habilitado e monitorado pela página *accelerometer.html*, que controla a ativação via comandos REST e exibe, em tempo real, a orientação recebida através do WebSocket. A interface apresenta os valores de Roll, Pitch e Yaw, os quaternions brutos do sensor, a escala de atuação do sistema e uma função de recalibração de referência, que redefine a orientação atual como ponto neutro para compensar pequenos desvios ou inclinações iniciais.

Os ângulos medidos pelo sensor são limitados a uma faixa de  $\pm 5^\circ$  em cada eixo de rotação, evitando solicitações excessivas aos atuadores e garantindo que a plataforma opere dentro de uma região segura.

É importante destacar que, ao utilizar um sensor MPU6050, o ângulo de rotação em Z (*Yaw*) apresenta menor confiabilidade quando comparado aos sensores da família BNO085. Isso ocorre porque o MPU6050 não possui magnetômetro, operando apenas com acelerômetro e giroscópio. Dessa forma, o *Yaw* é estimado exclusivamente pela integração da velocidade angular, processo que acumula erro numérico ao longo do tempo na ausência de uma referência absoluta ao campo magnético terrestre. Esse acúmulo progressivo de erro é conhecido como *drift*, um desvio lento e contínuo do valor real, que faz com que o ângulo estimado se afaste gradualmente da orientação verdadeira mesmo quando o sensor está parado. Como consequência, o eixo Z tende a apresentar erros acumulados e maior instabilidade, especialmente em movimentos lentos ou durante períodos prolongados de operação.

Embora exista a intenção de futuramente utilizar esses dados para gerar uma pose alvo inversa, permitindo que a plataforma compense automaticamente a inclinação detectada pela IMU (uma forma de estabilização ativa), essa funcionalidade ainda não foi implementada, pois requer refinamentos adicionais tanto no controle quanto no tratamento dos sinais. O desenvolvimento completo desse modo se enquadra nos itens previstos para trabalhos futuros.

Atualmente, a página *accelerometer.html* funciona como uma ferramenta de visualização e ensaio do modo IMU, servindo também como base conceitual para a futura implementação de um sistema de estabilização da bancada.

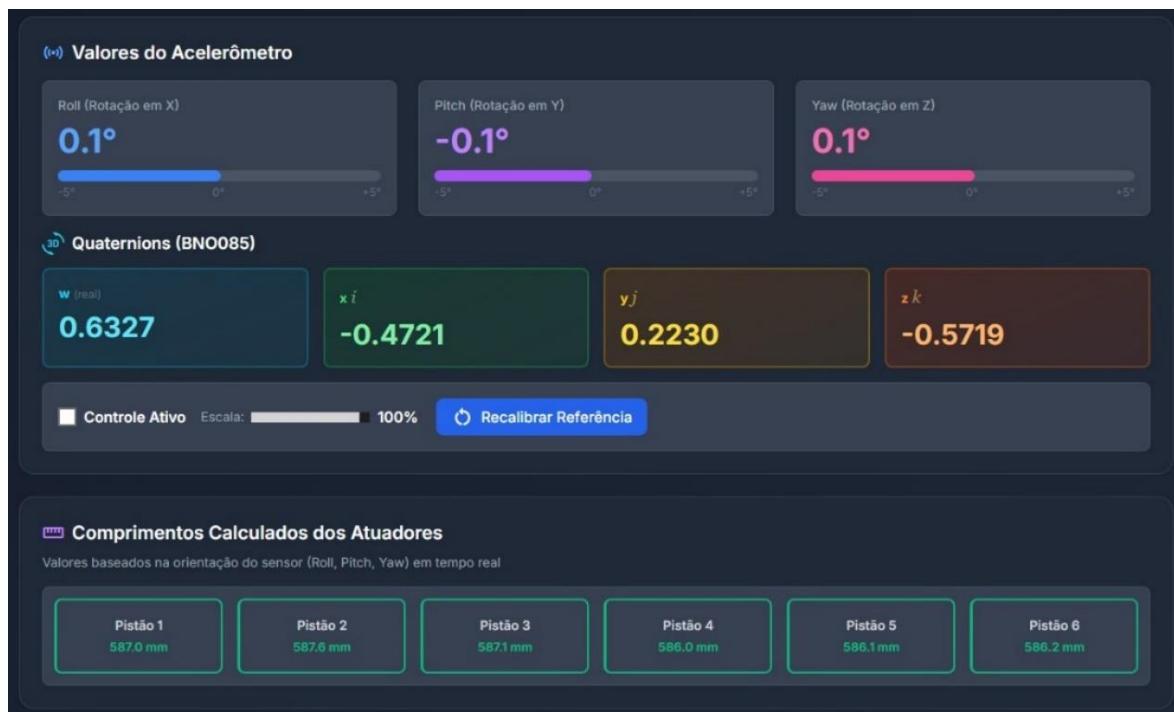


Figura 71: Valores do IMU  
Fonte: O autor

#### 4.5.6. IMPLEMENTAÇÃO DO CONTROLE POR JOYSTICK

O controle manual da plataforma foi implementado na página *controller.html*, utilizando a *Gamepad API* nativa dos navegadores modernos (ver Anexo X). Dessa forma, qualquer controle de videogame compatível com essa API (por exemplo, Xbox 360/One, controles genéricos XInput ou dispositivos equivalentes) pode ser utilizado, desde que seja corretamente reconhecido pelo navegador. Quando um gamepad é conectado, a aplicação registra o dispositivo e passa a ler continuamente os valores dos eixos analógicos e gatilhos, bem como o estado dos botões.

Os sinais lidos são normalizados em uma estrutura de pose que representa as translações (X, Y, Z) e rotações (Roll e Pitch) da plataforma. O mapeamento é definido da seguinte forma: o *stick* esquerdo controla a translação no plano XY ( $\pm 30$  mm), o stick direito controla as rotações *Roll* e *Pitch* (aproximadamente  $\pm 8^\circ$ ) e os gatilhos (LT e RT) são usados para ajustar a altura no eixo Z ( $\pm 20$  mm), permitindo elevação e descida controladas. Uma zona morta (*deadzone*) é aplicada aos eixos para ignorar pequenas variações próximas de zero, reduzindo ruídos e movimentos involuntários.

As poses calculadas são enviadas periodicamente ao backend por meio do endpoint POST */joystick/pose*, juntamente com os valores normalizados dos eixos e gatilhos, além de uma indicação se os comandos devem ser apenas visualizados (modo de pré-visualização) ou efetivamente aplicados ao hardware.

Tabela 11: Endpoint Controle Joystick

Método	Endpoint	Descrição
POST	<i>/joystick/pose</i>	Recebe eixos normalizados do joystick e converte em pose da plataforma (com opção de aplicar).

Fonte: O autor

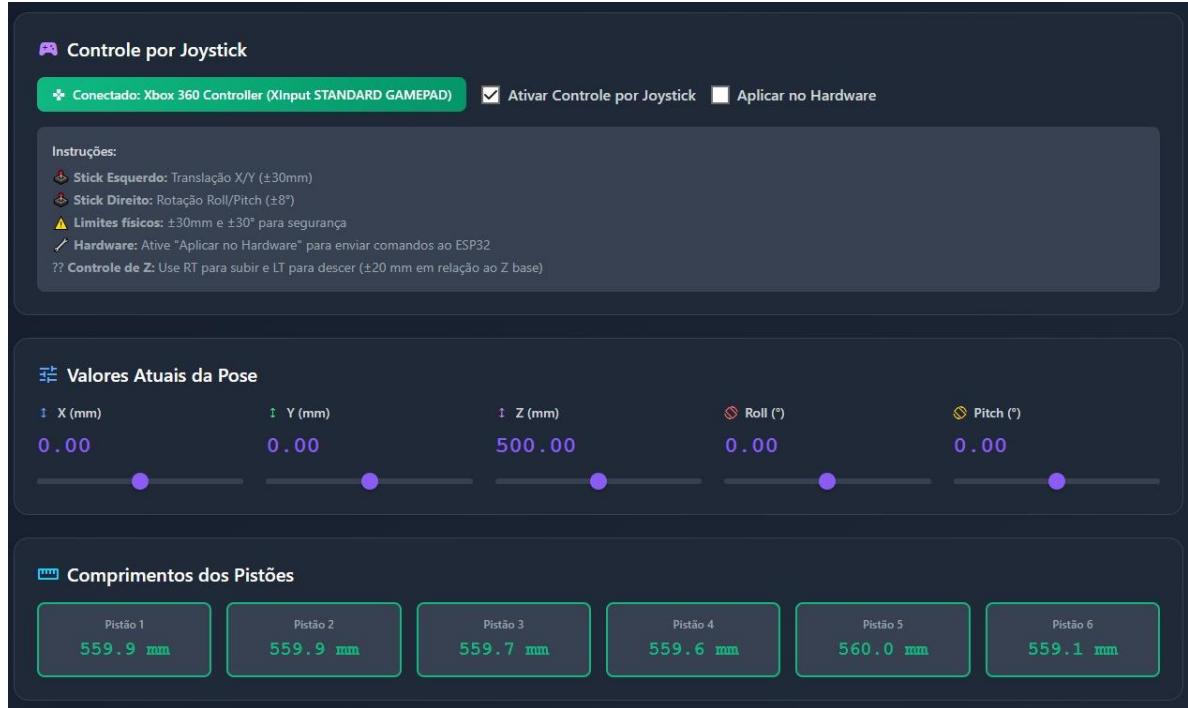


Figura 72: Pré-Visualização dos Estados  
Fonte: O autor

No backend, esses dados são processados pela cinemática inversa, que converte a pose em comprimentos individuais dos seis atuadores e valida os resultados de acordo com os limites geométricos e de curso da plataforma. Quando todos os comprimentos estão dentro da faixa permitida, a interface exibe os cartões dos pistões em verde; se alguma solução ultrapassa os limites físicos ou é considerada inválida, os pistões afetados são destacados em vermelho e o comando não é enviado ao ESP32-S3.

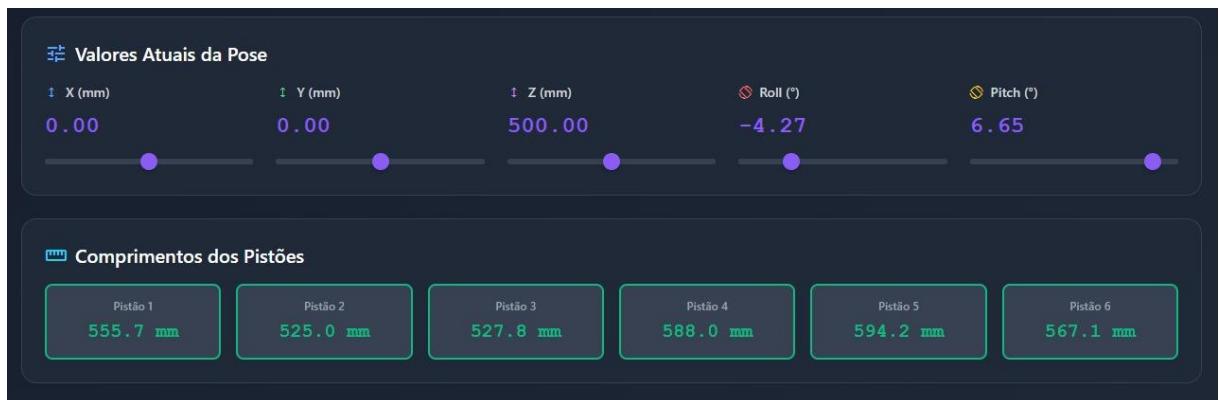


Figura 73: Controle com Posições Válidas  
Fonte: O autor



Figura 74: *Feedback de Posição Inválida*  
Fonte: O autor

A página atualiza, via WebSocket, tanto a pose corrente quanto os comprimentos de cada pistão, permitindo que o usuário observe em tempo real a resposta da plataforma aos comandos do joystick. Caso o gamepad seja desconectado ou o modo de controle manual seja desativado, o controlador interrompe o envio de comandos, desabilita o modo joystick e a plataforma é conduzida de volta ao setpoint neutro, garantindo uma condição segura de repouso.

#### 4.5.7. IMPLEMENTAÇÃO DO SIMULADOR DE VOO

Uma vez conectado ao simulador, o *fg-bridge* lê periodicamente os valores de *Roll* e *Pitch*, que são disponibilizados pelo FlightGear como propriedades internas da aeronave (ver Anexo XI). Esses valores são então submetidos a uma etapa de saturação configurável, garantindo que a plataforma física nunca receba ângulos superiores aos limites projetados. No protótipo desenvolvido, os ângulos são limitados aproximadamente a  $\pm 12^\circ$ , e a componente vertical Z permanece fixa em uma altura segura durante a simulação, evitando colisões mecânicas entre os elos ou aproximações excessivas entre a base e a plataforma superior. O eixo *Yaw* é mantido constante, uma vez que a versão atual da plataforma não possui acomodação estrutural para rotações horizontais.

Depois de filtrados e limitados, os valores coletados pelo *fg-bridge* são enviados ao backend FastAPI através de um endpoint interno. No backend, cada par (*Roll*, *Pitch*) é convertido em uma pose tridimensional da plataforma, que passa pela rotina de cinemática inversa para determinar os comprimentos individuais dos seis atuadores. Antes que qualquer movimento seja aplicado ao hardware, o backend solicita ao endpoint */calculate* a validação completa da pose, garantindo que todos os comprimentos calculados estejam dentro das faixas mecânicas permitidas e que não haja risco ultrapassagem de curso.

O resultado dessa verificação é encaminhado ao endpoint */flight-simulation/preview*, que armazena a pose prevista e permite ao frontend exibir uma pré-visualização 3D da postura da plataforma, baseada nos últimos comprimentos válidos recebidos. O painel inclui também indicadores de segurança, como estado da telemetria, última atualização recebida e permissões de seguimento. Somente quando a flag de segurança retornada por */flight-simulation/status*

estiver ativa, ou seja, quando a plataforma estiver em uma condição mecânica segura, o sistema permite ao operador clicar em “Iniciar seguimento” para aplicar as poses recebidas do FlightGear ao hardware real.

Tabela 12: Endpoints Integração FlightGear

Método	Endpoint	Descrição
POST	/flight-simulation/start	Habilita o modo de simulação de voo com dados recebidos do FlightGear.
POST	/flight-simulation/stop	Desabilita o modo de simulação de voo.
GET	/flight-simulation/preview	Retorna a última pose de pré-visualização recebida para simulação de voo.
POST	/flight-simulation/preview	Armazena uma pose de pré-visualização calculada para simulação de voo.
GET	/flight-simulation/status	Retorna o status atual da integração com o FlightGear (ativo, seguro etc.).

Fonte: O autor

O frontend apresenta, em tempo real, cartões com os comprimentos dos pistões, que mudam para vermelho caso o backend sinalize que algum atuador está fora da faixa segura. Além disso, uma interface 3D renderiza a postura da plataforma usando Three.js, permitindo ao operador visualizar de maneira clara o efeito dos dados provenientes do simulador antes de aplicá-los à estrutura física.

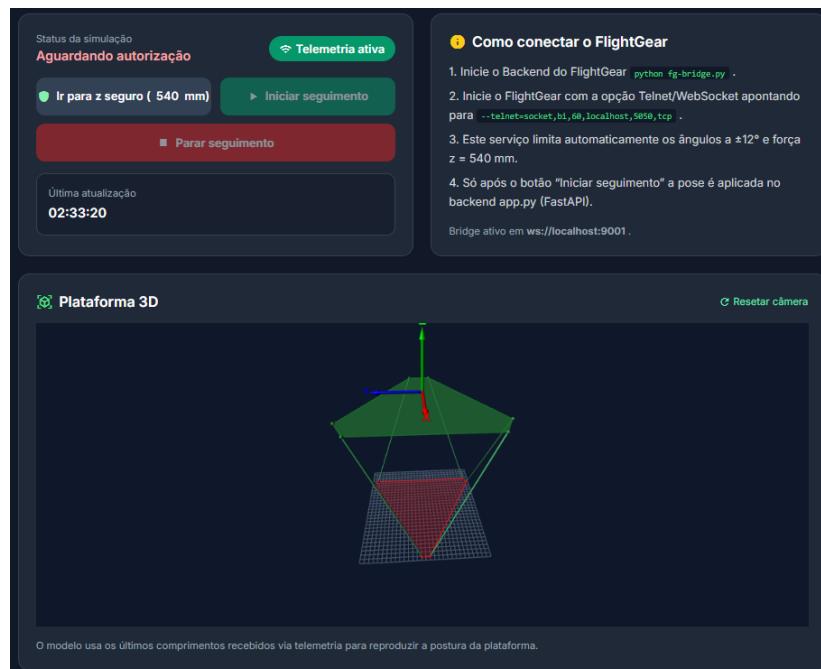


Figura 75: Interface para a Simulação de Voo

Fonte: O autor



Figura 76: Comprimento de Cada Pistão e Orientação *Roll* e *Pitch*  
Fonte: O autor

Para fins de apresentação, imersão e validação visual, foi criado um modelo customizado de aeronave no FlightGear, baseado no Embraer ERJ-145 (YASim), utilizando as cores e identidades visuais do Instituto Federal de São Paulo – Campus São José dos Campos. O modelo inclui inscrições personalizadas (“IFSP-SJC”, menções ao TCC e logotipos) sendo utilizado como aeronave de referência durante a simulação. Embora essa customização não interfira diretamente na integração técnica, ela proporciona um ambiente visual coerente com o contexto acadêmico.



Figura 77: Modelo Customizado IFSP-SJC  
Fonte: O autor



Figura 78: .png Base para o FlightGear  
Fonte: O autor

Essa integração transforma a plataforma em um dispositivo capaz de reproduzir movimentos provenientes de um ambiente de simulação de voo, aproximando o sistema de conceitos utilizados em plataformas de “*motion cueing*”. Embora a versão atual implemente apenas *Roll* e *Pitch* simplificados, o sistema foi projetado para permitir expansões futuras, incluindo controle de translado, filtros de suavização e algoritmos de *washout*, temas reservados para trabalhos futuros.

## 5. CONCLUSÕES E CONSIDERAÇÕES FINAIS

### 5.1. CONCLUSÕES

O objetivo geral deste trabalho foi desenvolver um sistema de controle dedicado para a bancada de plataforma de Stewart do IFSP–São José dos Campos, contemplando a implementação de um novo hardware controlador e de uma interface gráfica completa para operação e monitoramento. Com base nos resultados apresentados no Capítulo 4, conclui-se que esse objetivo foi atingido: a bancada foi restabelecida, modernizada e ampliada em suas funcionalidades, tornando-se adequada para atividades acadêmicas e experimentais.

Quanto aos objetivos específicos, todos foram atendidos de forma estruturada ao longo do desenvolvimento:

a) Realizou-se o levantamento detalhado da plataforma existente, incluindo estrutura mecânica, juntas, atuadores, sensores, drivers e cabeamento. Esse diagnóstico permitiu identificar limitações, como desgaste em juntas de Kardan, ruídos nos sensores resistivos e redundância de cabos remanescentes da arquitetura baseada em CLP.

b) Foi implementado um novo hardware de controle, utilizando o microcontrolador ESP32-S3 e uma placa de interface dedicada. Essa solução substituiu integralmente o CLP Siemens anteriormente utilizado, simplificando a eletrônica, reduzindo o número de módulos intermediários e garantindo capacidade de expansão futura. A integração direta com os drivers, sensores e fonte de potência restabeleceu a funcionalidade elétrica da bancada.

c) Avaliaram-se e implementaram-se algoritmos de controle de posição para os atuadores, baseados em modelagem experimental e identificação de sistemas. A obtenção de modelos de primeira ordem e a sintonia dos controladores PI permitiram sincronizar os seis pistões com boa repetibilidade e erro estacionário reduzido, validando experimentalmente o desempenho da malha fechada.

d) Foi desenvolvida uma interface gráfica abrangente, incluindo:

- controle individual das juntas;
- atuação global por cinemática inversa;
- visualização 3D da plataforma e dos atuadores;
- painel de monitoramento PID;
- execução de rotinas de movimento;
- integração com sensores IMU e com o simulador FlightGear.

Essa interface consolidou todos os subsistemas em um ambiente unificado, responsivo e eficiente.

e) O sistema foi validado por meio de testes práticos, ensaios comparativos e simulações, confirmando a aderência entre o comportamento real da bancada e os modelos utilizados no projeto. Os dados coletados demonstram que a nova arquitetura proporciona estabilidade, responsividade e previsibilidade na operação dos atuadores.

Dessa forma, o conjunto de etapas realizadas evidencia que o trabalho alcançou sua proposta inicial: recuperar, aprimorar e expandir a plataforma de Stewart, devolvendo ao laboratório um equipamento funcional, capaz de apoiar atividades de ensino, pesquisa e desenvolvimento na área de controle e automação.

## **5.2. CONSIDERAÇÕES FINAIS**

O desenvolvimento deste trabalho evidenciou a complexidade e o caráter multidisciplinar envolvidos na reestruturação de uma plataforma de Stewart, especialmente quando aplicada em um ambiente acadêmico. A reconstrução da bancada exigiu integração entre conhecimentos de mecânica, eletrônica, sistemas embarcados, controle, processamento de sinais, microcontroladores e desenvolvimento de software, reforçando a natureza multidisciplinar da Engenharia de Controle e Automação.

A transição de uma arquitetura industrial baseada em CLP para uma solução moderna com microcontrolador não apenas viabilizou o restabelecimento do sistema, como também ampliou sua flexibilidade, acessibilidade e potencial de expansão. O uso do ESP32-S3, aliado à placa de interface projetada e ao modelo de comunicação serial/web, resultou em um sistema de menor custo e com maior capacidade de experimentação. Essa mudança permitiu que a bancada se tornasse novamente utilizável em um contexto didático, ao mesmo tempo em que abriu espaço para estudos avançados em controle.

A implementação da interface gráfica desempenhou papel central nesse processo, unificando todos os subsistemas em uma plataforma de operação. A capacidade de visualizar a

plataforma em 3D, acompanhar o comportamento dos atuadores em tempo real, testar controladores, executar rotinas e integrar sensores iniciais e simuladores de voo acrescenta valor pedagógico significativo ao projeto. Tais recursos transformam a bancada em um ambiente de aprendizado mais completo, permitindo que estudantes compreendam, na prática, conceitos essenciais de cinemática, realimentação, controle PID, lidar com não linearidades, comunicação entre sistemas e instrumentação.

Por fim, os resultados obtidos demonstram que a plataforma possui potencial para aplicações futuras ainda mais amplas, como simulação de voo, estabilização ativa, experimentos de controle multivariável e estudos de sensoriamento avançado. O sistema desenvolvido estabelece uma base sólida para que novos trabalhos possam explorar técnicas modernas de controle, algoritmos adaptativos, filtragem avançada, aprendizado de máquina embarcado e integração com ambientes industriais ou educacionais interativos.

As contribuições alcançadas evidenciam que o trabalho não apenas cumpriu os objetivos iniciais, mas também deixou uma estrutura preparada para crescimento contínuo, consolidando-se como um recurso para o IFSP-SJC e para futuras linhas de pesquisa.

### **5.3. LIMITAÇÕES DA PESQUISA**

Embora o trabalho tenha atingido seus objetivos, algumas limitações foram identificadas ao longo do desenvolvimento, tanto no nível mecânico quanto no nível de controle, as quais influenciam o desempenho e a precisão da plataforma.

#### **5.3.1. LIMITAÇÕES MECÂNICAS DA BANCADA**

Durante os testes, verificou-se que os atuadores apresentavam níveis distintos de atrito interno. Alguns pistões precisavam de valores de PWM significativamente maiores do que outros para iniciar o movimento, o que pode ter sido causado pela falta de lubrificação ou desgaste natural das superfícies internas. Esse comportamento gerou zonas mortas amplas e respostas assimétricas entre avanço e recuo, impactando diretamente o controle.

#### **5.3.2. JUNTAS DE KARDAN IMPRESSAS EM 3D**

As juntas utilizadas são de plástico PLA feitos em impressora 3D, solução prática mas menos robusta que peças usinadas. Esse tipo de junta tem menor resistência mecânica, maior deformação sob carga e menor precisão dimensional, o que influencia a consistência dos movimentos e reduz a vida útil em aplicações mais exigentes.

### **5.3.3. AUSÊNCIA DE DOCUMENTAÇÃO DOS LIMITES MECÂNICOS**

A plataforma original não possuía documentação clara sobre os limites físicos de operação, como curso máximo seguro, ângulos admissíveis e regiões de risco de colisão. Assim, a definição desses limites ocorreu de forma experimental e conservadora, o que introduz incertezas sobre a faixa de operação ideal e restringe o uso pleno da capacidade mecânica da bancada.

#### **5.3.3.1. IMPACTO DOS LIMITES DESCONHECIDOS NAS FUNCIONALIDADES DO SOFTWARE**

A falta de especificação precisa obrigou a adoção de restrições empíricas em todas as funcionalidades do software. Entre elas:

- O curso dos pistões foi limitado a 180 mm, embora o curso mecânico total fosse próximo a 250 mm.
- Em funções baseadas em orientação (IMU, joystick e FlightGear), os ângulos foram limitados a valores diferentes nas páginas do sistema ( $\pm 15^\circ$ ,  $\pm 10^\circ$  ou  $\pm 8^\circ$ ), dependendo do risco percebido de colisão.
- A cinemática inversa utiliza *clamps* (Cortes) de segurança definidos empiricamente para impedir poses potencialmente perigosas.

Essas restrições, embora necessárias para proteger o equipamento, reduzem a amplitude de movimento da plataforma e impedem o uso pleno do espaço operacional do robô.

## **5.4. LIMITAÇÕES DOS SENSORES DE POSIÇÃO**

Os sensores resistivos apresentam ruído elevado, offset e sensibilidade a pequenas variações de alimentação. Mesmo com filtragem digital e calibração V0–V100, a precisão permanece limitada, impactando a resposta do controle e a fidelidade da cinemática direta.

#### **5.4.1. LIMITAÇÕES DO MODELO DINÂMICO E DA ESTRUTURA DE CONTROLE**

##### **5.4.1.1. MODELO SIMPLIFICADO E AUSÊNCIA DE DINÂMICA ACOPLADA**

A modelagem se baseou em funções de transferência individuais de primeira ordem, representando apenas a dinâmica predominante de cada atuador. Entretanto, o sistema real é mecanicamente acoplado, e o esforço necessário para mover um pistão depende da posição e do esforço dos demais. Essa interação não é capturada no modelo.

#### **5.4.1.2. INFLUÊNCIA DA GRAVIDADE E ASSIMETRIA ENTRE SUBIDA E DESCIDA**

Foi observado que subir e descer o pistão requerem esforços distintos, devido à influência da gravidade e das inclinações da plataforma. O ideal seria projetar duas malhas distintas (subida e descida), com ganhos específicos para compensar essas diferenças. Como este trabalho utilizou apenas um conjunto de ganhos PI, o controle operou com compensações simplificadas.

#### **5.4.1.3. CONTROLE PI EM SISTEMA NÃO LINEAR E ACOPLADO**

O controlador PI se mostrou funcional e garantiu sincronização aceitável entre os pistões na maior parte dos casos, porém sua estrutura linear não compensa satisfatoriamente:

- atrito estático e histerese,
- zonas mortas variáveis,
- saturações assimétricas,
- esforços dependentes da pose,
- acoplamento mecânico entre atuadores.

Métodos mais robustos, como controle multivariável, MPC (*Model Predictive Control*), estratégias não lineares e controle por inteligência artificial seriam mais adequados, mas estão além do escopo deste trabalho.

#### **5.4.2. LIMITAÇÕES NOS TESTES EXPERIMENTAIS**

O sistema não foi testado sob carga adicionada, ciclos longos de operação, perturbações externas ou condições adversas. Dessa forma, a robustez e o comportamento de longo prazo permanecem desconhecidos.

#### **5.4.3. LIMITAÇÕES NA INTEGRAÇÃO COM FLIGHTGEAR**

A integração cobre apenas Roll e Pitch, com movimentos reduzidos, sem algoritmos de washout, sem reprodução de movimentos 6-DOF e sem retroalimentação de forças. Assim, o sistema opera apenas como demonstrador de conceitos, e não como plataforma completa de simulação.

### **5.5. TRABALHOS FUTUROS**

Com base nos resultados obtidos e nas limitações identificadas ao longo deste trabalho, diversas oportunidades de continuidade e aprimoramento surgem para pesquisas futuras envolvendo a plataforma de Stewart.

No âmbito mecânico, uma das melhorias relevantes consiste na substituição das juntas de Kardan impressas em 3D por peças usinadas em alumínio ou polímeros técnicos. A impressão 3D foi adequada para prototipagem e reposição emergencial, porém apresenta limitações estruturais que impactam a rigidez e a vida útil das articulações. Juntas usinadas forneceriam maior resistência mecânica, menor folga e melhor repetibilidade do movimento, contribuindo diretamente para uma operação mais estável e previsível. Além disso, recomenda-se a realização de um estudo sobre os limites geométricos da Plataforma, como ângulos máximos, deslocamentos admissíveis e zonas de colisão, uma vez que, devido à ausência desses dados no início do projeto, todas as funcionalidades do software foram limitadas empiricamente. Os pistões, por exemplo, foram restringidos a um curso de 180 mm, embora possam atingir até 250 mm, e os ângulos de operação foram limitados de forma conservadora entre  $\pm 8^\circ$ ,  $\pm 10^\circ$  e  $\pm 15^\circ$ , dependendo do modo de operação. A determinação formal desses limites permitiria expandir o espaço de trabalho de forma segura e bem fundamentada. Também é relevante a adoção de um plano de manutenção mecânica, em especial no que se refere à lubrificação dos pistões, visto que diferenças de atrito resultaram em comportamentos assimétricos, obrigando alguns atuadores a exigirem níveis muito maiores de PWM para iniciar o movimento.

Do ponto de vista da eletrônica e da sensorização, há espaço para aperfeiçoamentos importantes, como a substituição dos sensores resistivos de posição por sensores absolutos, encoders magnéticos ou outras tecnologias que ofereçam maior linearidade e repetibilidade. Além disso, a adição de sensores de força nos elos, permitiria a implementação de controle baseado em forças, impedância ou admitância, o que aproximaria a plataforma de aplicações mais avançadas em simulação e interação física.

Em relação ao controle, o controlador PI implementado demonstrou boa estabilidade e sincronização, mas é limitado diante das não linearidades inerentes ao sistema, incluindo atrito estático, zonas mortas, histerese e principalmente o acoplamento cinemático entre os atuadores, que faz com que o esforço necessário para mover um pistão dependa da posição dos demais. Dessa forma, técnicas mais robustas poderiam ser investigadas, como controle multivariável (MIMO), controle preditivo por modelo (MPC), controle não linear ou estratégias fuzzy. Essas abordagens oferecem maior aderência ao comportamento real de plataformas paralelas e possuem mecanismos internos para lidar com saturações, restrições, acoplamentos e perturbações. Além disso, há espaço para métodos baseados em inteligência artificial, como redes neurais aplicadas à compensação de atrito, algoritmos neuro-fuzzy ou aprendizado por

reforço, permitindo que o controlador aprenda padrões de comportamento e otimize o desempenho em tempo real.

Em termos de interface e arquitetura de software, a plataforma também pode ser expandida para realizar controle baseado em IMU de forma completa, permitindo estabilização ativa da estrutura. Podem ser adicionadas novos padrões de rotinas de movimentos e implementado um gráfico otimizado para comparar as curvas de setpoint dos movimentos como os dados reais da plataforma. Além disso, futuramente, a comunicação entre o backend e o ESP32-S3 pode ser migrada para Wi-Fi ou Ethernet, possibilitando operação remota.

Finalmente, destaca-se o potencial da integração com simuladores de voo. A implementação atual já utiliza dados de *Roll* e *Pitch* provenientes do FlightGear, mas a plataforma pode ser expandida para comportar translados completos, simulação de turbulências, movimentos coordenados e implementação de algoritmos de *washout*, técnica essencial em plataformas de *motion cueing* que cria a sensação contínua de aceleração ao mesmo tempo em que mantém a estrutura dentro de seus limites mecânicos. A integração com outros simuladores, como X-Plane ou Microsoft Flight Simulator, também constitui um caminho de evolução.

Em síntese, a plataforma construída demonstra potencial para estudos acadêmicos e aplicações práticas, e os trabalhos futuros aqui apresentados representam oportunidades de evolução para transformar o protótipo atual em um sistema ainda mais completo, preciso e robusto.

## REFERÊNCIAS

- ALIEXPRESS. Módulo/Produto — imagem utilizada para ilustração.** Disponível em: <https://pt.aliexpress.com/item/1005009041165548.html>. Acesso em: 10/2025.
- ALVARELLOS, A.; FIGUERO, A.; CARRO, H.; COSTAS, R.; SANDE, J.; GUERRA, A.; PEÑA, E.; RABUÑAL, J. Machine Learning Based Moored Ship Movement Prediction.** *J. Mar. Sci. Eng.* 2021, 9, 800. <https://doi.org/10.3390/jmse9080800>
- ÅSTRÖM, K.; HÄGGLUND, T. PID Controllers.** 2nd ed. Research Triangle Park: Instrument Society of America, 1995.
- CAI, Y.; ZHENG, S.; LIU, W.; QU, Z.; HAN, J. Model analysis and modified control method of ship-mounted Stewart platforms for wave compensation.** *IEEE Access*, 2021.

- CASAS, S; OLANDA, R; DEY, N. **Motion cueing algorithms: a review – algorithms, evaluation and tuning.** International Journal of Virtual and Augmented Reality, 2017.
- CHEN, S; FU, L. **An optimal washout filter design for a motion platform with senseless and angular scaling maneuvers.** IEEE, 2010.
- CHEN, Y.; XU, Q.; LI, Z.; ZHANG, J.; WANG, J. **An online dual-loop AMPC strategy for wave compensation of an electro-hydraulic servo Stewart platform.** ISA Transactions, 2025.
- CRAIG, J. J. **Robótica.** São Paulo: Pearson Prentice Hall, 2012.
- DA SILVA, A. F. N. **Desenvolvimento de um protótipo de robô cilíndrico de configuração RPP e estudo da sua cinemática.** 2025. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São José dos Campos, 2025.
- DU, L.; LUO, Y.; JI, L.; YANG, F.; ZHANG, Y.; XIE, S. **Comprehensive parametric model and decoupling design of a Stewart platform for a large spaceborne optical load.** Acta Astronautica, 2025.
- EISELE, Robert. *Inverse kinematics of a Stewart platform.* 2019. Disponível em: <https://raw.org/paper/inverse-kinematics-of-a-stewart-platform/>. Acesso em: 08/2025.
- ESPRESSIF SYSTEMS. **ESP32-S3 Datasheet.** Version 2.7. 2024. Disponível em: [https://www.espressif.com/sites/default/files/documentation/esp32-s3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf). Acesso em: 5 out. 2025.
- FETTE, I.; MELNIKOV, A. **The WebSocket Protocol.** RFC 6455, Internet Engineering Task Force, 2011. Disponível em: <https://www.ietf.org/rfc/rfc6455.txt>. Acesso em: 29 de novembro de 2025.
- FIELDING, R. **Architectural Styles and the Design of Network-based Software Architectures.** 2000. 172 f. Tese (Doutorado em Ciência da Computação) — University of California, Irvine, 2000.
- GARCIA, C. **Controle de Processos.** v. 1. São Paulo: Edgard Blücher, 2018.
- GEEKSFORGEEKS. **What is an API?** Disponível em: <https://www.geeksforgeeks.org/software-testing/what-is-an-api/>. Acesso em: 11/2025.
- GONÇALVES, F. D. **Case Controlador Portátil e Plataforma de Stewart.** Trabalho de Conclusão de Curso – Instituto Federal de São Paulo — Campus São José dos Campos, 2023.
- HERNÁNDEZ-GUZMÁN, V. M.; SILVA-ORTIGOZA, R. **Automatic Control with Experiments.** Cham: Springer, 2019.

- IFR – INTERNATIONAL FEDERATION OF ROBOTICS. **Global Robot Density in Factories Doubled in Seven Years.** 2024. Disponível em: <https://ifr.org/wr-industrial-robots/#>. Acesso em: 20 jan. 2025.
- IFR – INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics: Industrial Robots.** 2025. Disponível em: [https://ifr.org/downloads/press\\_docs/PressConference2025\\_presentation.pdf](https://ifr.org/downloads/press_docs/PressConference2025_presentation.pdf). Acesso em: 21 jan. 2025.
- JENG, S.-L.; CHIENG, W.-H. **Web-based HMI of Industrial Controllers for General Purpose.** In: 3rd IEEE International Conference on Knowledge Innovation and Invention, 2020. DOI: 10.1109/ICKII50300.2020.9318905.
- LI, Z.; DU, H.; SI, J.; WANG, Z.; XIONG, W. **Rigid-flexible coupling dynamic modeling and validation of a helicopter rescue simulator based on an inverted Stewart platform.** Ocean Engineering, 2025.
- MATHWORKS. **MATLAB: The Language of Technical Computing.** Version R2025b. Natick, MA: The MathWorks, Inc., 2025.
- MERLET, J.; GOSSELIN, C. **Parallel Mechanisms and Robots.** In: SICILIANO, B.; KHATIB, O. (eds.). **Springer Handbook of Robotics.** Berlin: Springer, 2008.
- KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: uma abordagem top-down.** 6. ed. Porto Alegre: Pearson, 2021.
- NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). **A robotic crane system utilizing the Stewart platform manual mode with a Stewart platform joystick.** Gaithersburg, MD: NIST, 2008.
- NGUYEN, C. C.; ANTRAZI, S. S. **Experimental Study of Trajectory Planning and Control of a High Precision Robot Manipulator.** Semiannual Progress Report. NASA Goddard Space Flight Center, Grant NAG 5-780, 1991.
- NGUYEN, C. C.; ANTRAZI, S. S.; PARK, J.-Y.; ZHOU, Z.-L. **Trajectory planning and control of a Stewart platform-based end-effector with passive compliance for part assembly.** Journal of Intelligent and Robotic Systems, 1992.
- OGATA, K. **Engenharia de Controle Moderno.** 5. ed. São Paulo: Pearson Prentice Hall, 2011.
- ÖMÜRLÜ, V. E.; YILDIZ, İ. **Parallel Self-Tuning Fuzzy PD + PD Controller for a Stewart–Gough Platform-Based Spatial Joystick.** Arabian Journal for Science and Engineering, v. 37, p. 2089–2102, 2012. DOI: 10.1007/s13369-012-0308-0.

- PINHO, A. G.; OLÍMPIO, J. S. E.; CABRAL, L. M.; FILHO, R. M. O.; SILVA B. C. R.; FURRIEL, G. P.; JUNIOR, G. M. **Desenvolvimento de bancada didática contendo múltiplos sensores e atuadores.** Research, Society and Development, 2021.
- RABELLO, G.; MEIRELLES, G. M.; FERREIRA, T. P.; HIRATA, A. K. **Projeto de Controle de Plataforma Stewart.** Relatório técnico — Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus São José dos Campos, 2024.
- RICHARDSON, L; RUBY, S. **RESTful Web Services.** Sebastopol: O'Reilly Media, 2007.
- ROSSITER, J. A.; CASSANDRAS, C. G.; HESPANHA, J.; DORMIDO, S.; DE LA TORRE, L.; RANADE, G.; VISIOLI, A.; HEDENGREN, J.; MURRAY, R. M.; ANTSALKIS, P.; LAMNABHI-LAGARRIGUE, F.; PARISINI, T. **Control education for societal-scale challenges: A community roadmap.** *Annual Reviews in Control*, 2023.
- SINGHPOO, T.; WONGPICHET, S.; POSOM, J.; RUNAPONGSA SAIKAEW, K.; PHUPHAPHUD, A.; BANTERNG, P.; WONGPHATTI, M.; SAENGPRACHATANARUG, K. **Design, development, and testing of a cassava storage root-cutting robot utilizing a Stewart platform and mask R-CNN for precision agriculture.** *Smart Agricultural Technology*, 2024.
- SKIPLEVEL. **REST API components: how to read them.** Disponível em: <https://www.skiplevel.co/blog/part-2-rest-api-components-how-to-read-them>. Acesso em: 11/2025.
- STEWART, D. **A Platform with Six Degrees of Freedom.** Proceedings of the Institution of Mechanical Engineers, 1965.
- TANENBAUM, A. S.; WETHERALL, D. J. **Redes de Computadores.** 6. ed. Tradução: Daniel Vieira. Revisão técnica: Isaías Lima. Porto Alegre: Bookman, 2021.
- TANG, J.; YANG, Y.; LI, Y.; CAO, D. **A 6-DOF micro-vibration isolation platform based on the quasi-zero-stiffness isolator.** Journal of Mechanical Engineering Science, 2021.
- TAYLOR, B.; EASTWOOD, P.; JONES, B. L. **Development of a low-cost, portable hardware platform for teaching control and systems theory.** In: 10th IFAC Symposium Advances in Control Education, 2013.
- USINAINFO. **Driver motor ponte H para motor DC 9–36 V 12 A JZ-3615A para Arduino/ESP32.** Disponível em: <https://www.usinainfo.com.br/driver-para-motor/driver-motor-ponte-h-para-motor-dc-9-36v-12a-jz-3615-a-para-arduino-esp32-8983.html>. Acesso em: 09/2025.

WANG, W.; NING, Y.; ZHANG, Y.; XU, P.; LI, B. **Linear active disturbance rejection control with linear quadratic regulator for Stewart platform in active wave compensation system.** Applied Ocean Research, 2025.

YILDIZ, İ.; ÖMÜRLÜ, V. E. **A passive Stewart platform based joystick to control spatially moving objects.** International Journal of Materials Science and Engineering, v. 4, n. 2, p. 135–142, 2016.

YUAN, H.; SU, Y.; LI, J.; ZHANG, Y.; LI, Z. **Parametric vibration analysis of a six-degree-of-freedom electro-hydraulic Stewart platform.** Shock and Vibration, 2021.

YUAN, J.; TANG, Y.; WANG, W.; ZHANG, L. **Parametric vibration analysis of a six-degree-of-freedom electro-hydraulic Stewart platform.** Shock and Vibration, 2021.

## **ANEXO I – Arquitetura Geral**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/README.md)

## **ANEXO II – ESP32-S3 no Controle dos Pistões**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/PID-CONTROL-FILTER-SPIKE-BNO.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/PID-CONTROL-FILTER-SPIKE-BNO.md)

## **ANEXO III – Funcionamento Painel PID**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/ACTUATORS-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/ACTUATORS-README.md)

## **ANEXO IV – Configuração de Parâmetros**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/SETTINGS-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/SETTINGS-README.md)

## **ANEXO V – Funcionamento Cinemática**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/KINEMATICS-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/KINEMATICS-README.md)

## **ANEXO VI – Rotinas de Movimento**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/ROUTINES-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/ROUTINES-README.md)

## **ANEXO VII – Controle por IMU**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/ACCELEROMETER-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/ACCELEROMETER-README.md)

## **ANEXO VIII – Integração com MPU6050**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/MPU6050-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/MPU6050-README.md)

## **ANEXO IX – Integração com BNO085**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/BNO085-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/BNO085-README.md)

## **ANEXO X – Controle Via Joystick**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/JOYSTICK-CONTROL-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/JOYSTICK-CONTROL-README.md)

## **ANEXO XI – Simulação de Voo**

Disponível em:

[https://github.com/g-Miyata/PLATAFORMA\\_DE\\_STEWART/blob/main/FLIGHTGEAR-README.md](https://github.com/g-Miyata/PLATAFORMA_DE_STEWART/blob/main/FLIGHTGEAR-README.md)