

**GUILHERME mIYATA MEIRELLES**

**DESENVOLVIMENTO DE SISTEMA DE CONTROLE DE UMA PLATAFORMA STEWART**

**São José dos Campos, 2025**

**SP – Brasil**

Texto

Descrição gerada automaticamente com confiança média

**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**

**DESENVOLVIMENTO DE SISTEMAS DE CONTROLE DE UMA PLATAFORMA STEWART**

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**

***DESENVOLVIMENTO DE SISTEMAS DE CONTROLE DE UMA PLATAFORMA STEWART***

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**

Elemento opcional no qual o autor presta homenagem ou dedica seu trabalho para uma ou mais pessoas.

Exemplo:

Dedico este trabalho aos meus pais que sempre me incentivaram e apoiaram ...

**Agradecimentos**

Elemento opcional no qual o autor faz agradecimentos dirigidos àqueles que contribuíram de maneira relevante à elaboração do trabalho.

Exemplo:

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais prestamos nossas homenagens:

Aos nossos pais pelo incentivo em todos os momentos de nossas vidas.

Ao nosso orientador, que nos mostrou os caminhos a serem seguidos para a correta elaboração desta monografia.

À empresa TecCamp, que gentilmente permitiu o desenvolvimento da pesquisa em suas instalações.

A todos os professores e colegas do IFSP, que ajudaram de forma direta e indireta na conclusão deste trabalho.

Epígrafe - pensamento, parte de música ... que exprime uma ideia que é simbólica para o autor.

*Espaço destinado a epígrafe – “xxxx.”*

autor/ ano -xxxx

**Resumo (fazer após finalizar)**

SOBRENOME, Nome. **Manual para a Elaboração de Monografias e Trabalhos Acadêmicos**. 2013. 68p. Monografia (Trabalho de Conclusão de Curso). Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São José dos Campos.

Este documento apresenta o modelo de formatação a ser utilizado nas monografias e trabalhos acadêmicos a serem apresentados nos cursos do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus São José dos Campos. Torna-se importante salientar que o resumo se caracteriza como um elemento obrigatório, em que são apresentados os principais pontos relevantes do texto. O resumo deve dar uma visão rápida e clara do conteúdo e das conclusões do trabalho. Constitui-se em uma seqüência de frases concisas e objetivas e não de uma simples enumeração de tópicos, não ultrapassando 500 palavras. Deve ser formatado conforme este texto e, na seqüência, devem ser apresentadas ao menos três palavras-chave que permitam uma rápida identificação dos macro-temas abordados.

**Palavras-chave**: Trabalho de Conclusão de Curso; Monografia; Modelo.

Elemento obrigatório em português contendo no máximo 500 palavras, no qual o autor apresenta de forma concisa os pontos relevantes de seu trabalho.

Seguir a norma ABNT 6028.

Contemplar o objetivo geral, método de pesquisa (classificação e procedimentos), principais resultados e principais conclusões.

**Palavras Chave: sugestão até 5 palavras chave**

***Abstract***

Resumo traduzido para o inglês.

**Key Word:**

**Lista de Figuras**

[Figura 1 – Plataforma de Stewart IFSP 18](#_Toc214231774)

[Figura 2: Estoque Operacional de Robôs Industriais - Mundo 22](#_Toc214231775)

[Figura 3: Instalações Anuais de Robôs Industriais – Mundo 23](#_Toc214231776)

[Figura 4: Robô Serial 23](#_Toc214231777)

[Figura 5: Plataforma de Stewart 24](#_Toc214231778)

[Figura 6: Representação Vetorial da Cinemática da Plataforma de Stewart 26](#_Toc214231779)

[Figura 7: Comparação das curvas com diferentes . 32](#_Toc214231780)

[Figura 8: Aproximação por modelo de 1ª ordem com tempo morto 33](#_Toc214231781)

[Figura 9: Estrutura original da bancada com CLP Siemens e atuadores lineares 37](#_Toc214231782)

[Figura 10: Comando dos Motores 38](#_Toc214231783)

[Figura 11: Arquitetura original com CLP e conversores NPN/PNP 39](#_Toc214231784)

[Figura 12: Funcionamento conversores NPN/PNP 39](#_Toc214231785)

[Figura 13: Arquitetura atual com ESP32-S3 e controle direto 40](#_Toc214231786)

[Figura 14: Atuador Linear 41](#_Toc214231787)

[Figura 15: Conector DB37 com fios numerados 42](#_Toc214231788)

[Figura 16: PCB para ESP32S3 42](#_Toc214231789)

[Figura 17: Aquisição inicial - Ruídos 43](#_Toc214231790)

[Figura 18: Resposta com filtro passa-baixa 44](#_Toc214231791)

[Figura 19: Resposta com filtro média móvel (janela de 8) 44](#_Toc214231792)

[Figura 20: Resposta com Filtro Anti-Picos (mediana + limitador de inclinação) 45](#_Toc214231793)

[Figura 21: Deslocamento dos pistões com aplicação simultânea de PWM 46](#_Toc214231794)

[Figura 22: Correção de Offset 47](#_Toc214231795)

[Figura 23: Interface do Aplicativo de Coleta de Dados 50](#_Toc214231796)

[Figura 24: Dados Obtidos do Pistão 1 51](#_Toc214231797)

[Figura 25: Dados Obtidos do Pistão 2 51](#_Toc214231798)

[Figura 26: Dados Obtidos do Pistão 3 51](#_Toc214231799)

[Figura 27: Dados Obtidos do Pistão 4 52](#_Toc214231800)

[Figura 28: Dados Obtidos do Pistão 5 52](#_Toc214231801)

[Figura 29: Dados Obtidos do Pistão 6 52](#_Toc214231802)

[Figura 30: Coleta simultânea dos 6 pistões com Kp = 5 53](#_Toc214231803)

[Figura 31: Dados Importados para o System Identification 54](#_Toc214231804)

[Figura 32: Configurações do Modelo 55](#_Toc214231805)

[Figura 33: Resultado do Modelo Gerado Pistão 1 e 2 55](#_Toc214231806)

[Figura 34: Resultado do Modelo Gerado Pistão 3 e 4 55](#_Toc214231807)

[Figura 35: Resultado do Modelo Gerado Pistão 5 e 6 56](#_Toc214231808)

[Figura 36: Pistão 1 – Comparação entre as FTs 57](#_Toc214231809)

[Figura 37: Pistão 1 – Erro ao longo do tempo 58](#_Toc214231810)

[Figura 38: Pistão 2 – Comparação entre as FTs 58](#_Toc214231811)

[Figura 39: Pistão 2 – Erro ao longo do tempo 58](#_Toc214231812)

[Figura 40: Pistão 3 – Comparação entre as FTs 59](#_Toc214231813)

[Figura 41: Pistão 3 – Erro ao longo do tempo 59](#_Toc214231814)

[Figura 42: Pistão 4 – Comparação entre as FTs 59](#_Toc214231815)

[Figura 43: Pistão 4 – Erro ao longo do tempo 60](#_Toc214231816)

[Figura 44: Pistão 5 – Comparação entre as FTs 60](#_Toc214231817)

[Figura 45: Pistão 5 – Erro ao longo do tempo 60](#_Toc214231818)

[Figura 46: Pistão 6 – Comparação entre as FTs 61](#_Toc214231819)

[Figura 47: Pistão 6 – Erro ao longo do tempo 61](#_Toc214231820)

[Figura 48: Resultado Obtido – Curva 6 Pistões Sobrepostas para Comparação 63](#_Toc214231821)

[Figura 49: Especificações do Design 64](#_Toc214231822)

[Figura 50: Resultado Desejado 64](#_Toc214231823)

[Figura 51: Resposta do ensaio 65](#_Toc214231824)

[Figura 52: Resposta antes da sincronização – Visão geral 66](#_Toc214231825)

[Figura 53: Resposta antes da sincronização – zoom Tr e Ts 66](#_Toc214231826)

[Figura 54: Resposta após sincronização – Visão geral 67](#_Toc214231827)

[Figura 55: Resposta antes da sincronização – zoom Tr e Ts 67](#_Toc214231828)

**Lista de Quadros**

Sugestão numeração com ordenação por capítulo (1.1, 1.2, 1.3 ... 1.x, por exemplo)

Quadros não possuem estatísticas apenas informações.

**OBS: a chamada abaixo exemplifica como deve ser estruturada a lista de quadros no trabalho de conclusão de curso.**

[**Quadro 2.2** Classificação das MPEs em função do número de empregados. 29](#_Toc199772679)

**Lista de Tabelas**

Sugestão numeração com ordenação por capítulo (1.1, 1.2, 1.3 ... 1.x, por exemplo)

Tabelas possuem estatísticas.

**OBS: a chamada abaixo exemplifica como deve ser estruturada a lista de tabelas no trabalho de conclusão de curso.**

[**Tabela 4.2** Causas das dificuldades e razões de fechamento das empresas 29](#_Toc199772745)

**Lista de Equações**

**OBS: a chamada abaixo exemplifica como deve ser estruturada a lista de equações no trabalho de conclusão de curso.**

[**Equação 1** Cálculo da complexidade 29](#_Toc199772745)

**Lista de Abreviaturas, Nomenclaturas e Siglas**

**FT** Função de Transferência

**Ts** *Settling Time* - Tempo de Acomodação

**Tr** *Rise Time* - Tempo de Subida

**FOPDT** *First Order Plus Dead Time -* Modelo de 1ª ordem com tempo morto

**PID** Proporcional–Integral–Derivativo

**IHM** Interface Homem-Máquina

**HTML** *HyperText Markup Language* - Linguagem de Marcação de Hipertexto

**CSS** *Cascading Style Sheets*

**IMU** *Inertial Measurement Unit*

**Sumário**

[1. INTRODUÇÃO 18](#_Toc214232669)

[1.1. Contexto e Justificativa 18](#_Toc214232670)

[1.2. Problema de Pesquisa 19](#_Toc214232671)

[1.3. Objetivos 20](#_Toc214232672)

[1.4. Objetivo Geral 20](#_Toc214232673)

[1.5. Objetivos Específicos 20](#_Toc214232674)

[1.6. Apresentação da Estrutura do Trabalho 20](#_Toc214232675)

[2. FUNDAMENTAÇÃO TEÓRICA 22](#_Toc214232676)

[2.1. Robótica 22](#_Toc214232677)

[2.2. Cinemática da Plataforma de Stewart 24](#_Toc214232678)

[2.2.1. Cinemática Inversa 25](#_Toc214232679)

[2.2.2. Cinemática Direta 26](#_Toc214232680)

[2.3. Aplicações e Formas de Comando da Plataforma de Stewart 28](#_Toc214232681)

[2.3.1. Rotinas de Movimento 28](#_Toc214232682)

[2.3.2. Controle de Estabilidade 29](#_Toc214232683)

[2.3.3. Controle por Dispositivos Manuais 30](#_Toc214232684)

[2.4. Controle 31](#_Toc214232685)

[2.5. Controlador ESP32-S3 34](#_Toc214232686)

[2.6. Interfaces Gráficas em Sistemas de Controle 34](#_Toc214232687)

[2.7. Conceito de Aplicações Web e Arquitetura Cliente-Servidor 35](#_Toc214232688)

[2.8. Tecnologias Utilizadas na Camada Web 36](#_Toc214232689)

[3. MÉTODO E PROCEDIMENTOS METODOLÓGICOS 36](#_Toc214232690)

[3.1. Análise da Bancada Existente no IFSP 36](#_Toc214232691)

[3.2. Adaptação da Bancada para o ESP32-S3 36](#_Toc214232692)

[3.2.1. Substituição do CLP pelo ESP32-S3 36](#_Toc214232693)

[3.2.2. Retirada dos Optoacopladores e mudança na lógica de controle. 36](#_Toc214232694)

[3.2.3. Reconfiguração dos Drivers de Potência 36](#_Toc214232695)

[3.2.4. Adaptação dos Sensores de Feedback 36](#_Toc214232696)

[3.2.5. Desenvolvimento de Placa de Circuito Impresso (PCB) de Interface 36](#_Toc214232697)

[3.3. testes e aquisições de dados da bancada 36](#_Toc214232698)

[3.3.1. tratamento de ruído e filtragem 37](#_Toc214232699)

[3.3.2. assincronia de deslocamento entre pistões 37](#_Toc214232700)

[3.3.3. MAPEAMENTO TENSÃO → POSIÇÃO (V0–V100) 37](#_Toc214232701)

[3.3.4. COMPENSAÇÃO DE ZONA MORTA/ATRITO 37](#_Toc214232702)

[3.4. IDENTIFICAÇÃO E SINTONIA DE CONTROLE PARA SINCRONIA 37](#_Toc214232703)

[3.4.1. FASE 1 – Aquisição de Dados 37](#_Toc214232704)

[3.4.2. FASE 2 - Identificação de modelo (MATLAB) 37](#_Toc214232705)

[3.4.3. FASE 3 - Abertura de malha e sintonia do controle. 37](#_Toc214232706)

[3.5. DESENVOLVIMENTO DA INTERFACE DE CONTROLE E MODOS DE OPERAÇÃO 43](#_Toc214232707)

[3.5.1. Arquitetura de comunicação 43](#_Toc214232708)

[3.5.2. Monitoramento, Telemetria E Painel Pid 43](#_Toc214232709)

[3.5.3. Implementação Da Cinemática 43](#_Toc214232710)

[3.5.4. Implementação Das Rotinas De Movimento 43](#_Toc214232711)

[3.5.5. Implementação Do Controle Por Imu 43](#_Toc214232712)

[3.5.6. Implementação Do Controle Por Joystick 43](#_Toc214232713)

1. INTRODUÇÃO

As plataformas de Stewart são mecanismos capazes de gerar movimentos precisos em seis graus de liberdade, muito aplicados na área de engenharia para aplicações que exigem alto desempenho dinâmico e repetibilidade. Desde o trabalho de Stewart (1965), que descreveu seu uso em simuladores de voo, essas plataformas passaram a ser aplicadas em testes dinâmicos, compensação de movimentos em ambientes marítimos, robótica, manufatura avançada, reabilitação médica e estabilização de sensores. Sua versatilidade e rigidez estrutural fizeram com que se tornassem referência em sistemas que demandam controle preciso de pose (posição e orientação), servindo como base para pesquisa, desenvolvimento tecnológico e estudos aplicados em controle e automação.

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 São José dos Campos. Diferentemente de projetos que têm como foco apenas a construção de protótipos, esta pesquisa concentra-se na reestruturação e aprimoramento de uma bancada previamente desenvolvida em um Trabalho de Conclusão de Curso, que originalmente utilizava um Controlador Lógico Programável (CLP) para comandar a plataforma.

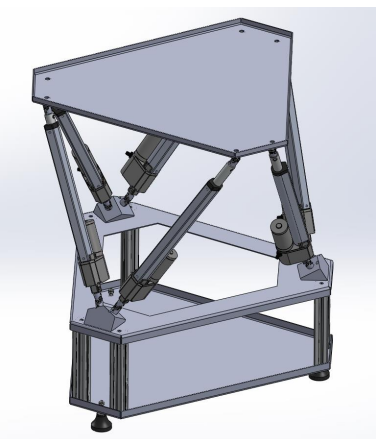


Figura 1 – Plataforma de Stewart IFSP

Fonte: Case Controlador Portátil e Plataforma de Stewart (GONÇALVES, 2023)

Com este estudo, espera-se contribuir para a evolução de soluções de automação e controle no campo da robótica paralela, explorando não apenas as técnicas de controle de movimento, mas também o desenvolvimento de uma interface homem-máquina (IHMy) intuitiva. O objetivo é ampliar a acessibilidade e a aplicabilidade da bancada em atividades acadêmicas, tornando seu uso mais prático e eficiente para alunos e professores em contextos didáticos.

* 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, elétrica, teoria de controle e ciência da 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.

O projeto propõe superar essas restrições por meio do desenvolvimento de um sistema de controle dedicado, composto por hardware compatível com as demandas da aplicação e uma interface gráfica intuitiva para operação e monitoramento. A proposta integra hardware e software em uma arquitetura unificada, oferecendo meios para a realização de experimentos com cinemática direta e inversa, simulação de trajetórias e implementação de técnicas de controle.

Além de restaurar e modernizar a funcionalidade da plataforma, este trabalho busca transformá-la em uma ferramenta didática robusta e acessível, capaz de apoiar práticas de ensino e pesquisa em Engenharia de Controle e Automação. 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). Assim, pretende-se ampliar a usabilidade da bancada para alunos e professores, contribuindo para o fortalecimento da infraestrutura laboratorial do IFSP-SJC e para o desenvolvimento de competências técnicas essenciais à atuação profissional.

* 1. Problema de Pesquisa

Diante 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. Objetivos

O objetivo deste trabalho é desenvolver um sistema de controle dedicado a bancada. O sistema deve contar com o desenvolvimento do hardware controlador, a criação de uma interface gráfica para operação e a implementação de algoritmos de controle. A proposta é de fornecer um ambiente que permita a aplicação em estudos e atividades práticas de ensino.

* 1. Objetivo Geral

Desenvolver e validar um sistema de controle para a plataforma de Stewart, compreendendo e adaptando o hardware controlador existente e desenvolver interface gráfica de operação e monitoramento.

* 1. Objetivos Específicos

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

1. Levantamento e análise da plataforma existente (mecânica, sensores, atuadores, drivers, fonte de potência, fixações, juntas e curso útil), identificando limitações.
2. Implementar uma placa para controle que faça interface com os outros componentes de hardware já existentes.
3. Avaliar e implementar algoritmos de controle para funcionamento dos atuadores do robô.
4. Desenvolver a interface gráfica (IHM) para operação local, incluindo: modos de comando (pistão a pistão e cinemática inversa), visualização do estado (posições).
5. Validar o funcionamento do sistema com testes práticos e simulações comparativas.
   1. Apresentação da Estrutura do Trabalho

Fazer depois

O Capítulo 1 se encerra com uma apresentação sucinta do que o leitor encontrará nos próximos capítulos. Faça uma síntese do conteúdo que o leitor encontrará nas páginas subseqüentes. Não há necessidade de apresentar o conteúdo do capítulo introdutório, pois o leitor acabou de ter contato com o mesmo.

O trabalho está estruturado em 5 capítulos, os quais estão apresentados com os seguintes conteúdos:

**Capítulo 1:** Introdução – ...;

**Capítulo 2:** Fundamentação Teórica – ...;

**Capítulo 3:** Método – ...;

**Capítulo 4:** Apresentação e Análise do Estudo de Caso – ...;

**Capítulo 5:** Conclusões – ....

Ao final são apresentadas as referências utilizadas no desenvolvimento do trabalho e os anexos/apêndices.

ESTRUTURAR SEÇÃO

1. FUNDAMENTAÇÃO TEÓRICA
   1. Robótica

A robótica, nas últimas décadas, 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 4,28 milhões de robôs operando em fábricas no mundo em 2023 e instalações anuais acima de meio milhão de unidades por três anos consecutivos, 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, 2023).

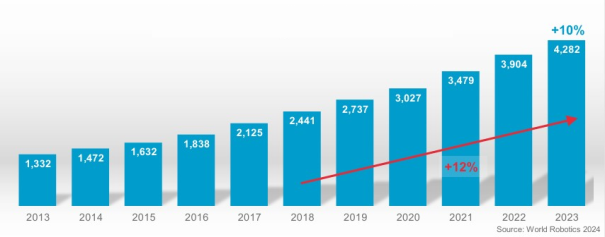


Figura 2: Estoque Operacional de Robôs Industriais - Mundo

Fonte: INTERNATIONAL FEDERATION OF ROBOTICS. World Robotics 2024.

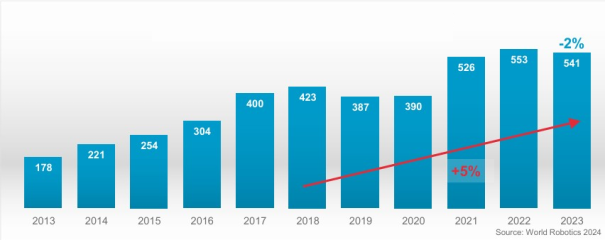


Figura 3: Instalações Anuais de Robôs Industriais – Mundo

Fonte: INTERNATIONAL FEDERATION OF ROBOTICS. World Robotics 2024.

Os sistemas robóticos podem ser classificados de diversas formas, sendo uma das mais comuns a distinção entre robôs seriais e robôs paralelos. Os manipuladores seriais são compostos por elos e juntas dispostos em sequência, amplamente utilizados em linhas de produção em tarefas como soldagem, pintura e montagem. Apresentam como principais vantagens o grande alcance e a flexibilidade, mas sofrem com menor rigidez estrutural. Já os robôs paralelos possuem múltiplas cadeias cinemáticas fechadas que conectam a base à plataforma móvel, o que proporciona maior rigidez, precisão e capacidade de suportar cargas elevadas, ainda que com área de trabalho reduzida e análise cinemática mais complexa (CRAIG, 2012). Essa distinção é fundamental, pois influencia diretamente o desempenho, as estratégias de controle e as possíveis aplicações de cada tipo de robô.

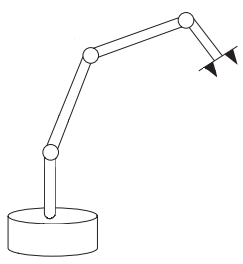


Figura 4: Robô Serial

Fonte: Robótica (CRAIG, 2012)

Entre os robôs paralelos, destaca-se a plataforma de Stewart, proposta originalmente em 1965 para aplicações em simuladores de voo (STEWART, 1965). O mecanismo é composto por uma base fixa e uma plataforma móvel interligadas por seis atuadores prismáticos, configurando um sistema com seis graus de liberdade, três translacionais e três rotacionais. Sua versatilidade foi comprovada ao longo das décadas, expandindo-se para áreas como simulação de veículos e aeronaves, ensaios estruturais, compensação de movimento em ambientes marítimos, dispositivos médicos de alta precisão, micromanipuladores e pesquisas acadêmicas em cinemática e controle.

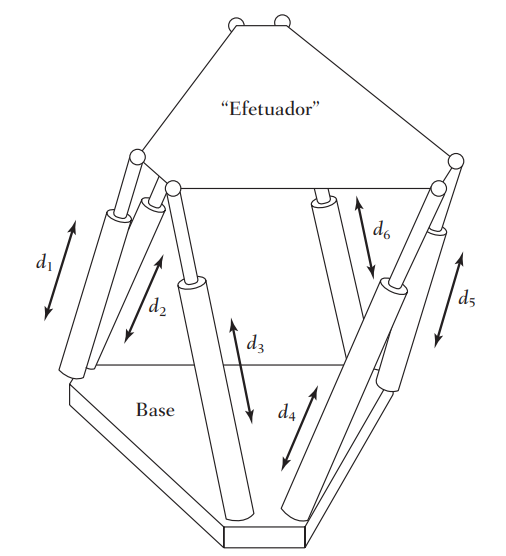


Figura 5: Plataforma de Stewart

Fonte: Robótica (CRAIG, 2012)

Do ponto de vista teórico, a plataforma de Stewart exige domínio dos conceitos de cinemática direta e inversa, que permitem relacionar os deslocamentos das juntas com a posição e a orientação da plataforma.

Além disso, a síntese de controladores deve lidar com o acoplamento dinâmico entre os atuadores, a sensibilidade a incertezas geométricas e a necessidade de sincronismo em tempo real.

Conforme destaca Craig (2012), os métodos de controle linear, como os baseados em controladores PID, devem ser entendidos como aproximações aplicadas a sistemas essencialmente não lineares, como 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.

* 1. Cinemática da Plataforma de Stewart

A análise cinemática é um dos elementos centrais para o estudo e o controle de robôs paralelos, como a plataforma de Stewart. Esse tipo de mecanismo apresenta seis atuadores conectando rigidamente a base à plataforma móvel por meio de juntas esféricas, universais ou no caso desse projeto, juntas de Cardan, formando um sistema completamente fechado. Dessa forma, a relação entre posição/orientação da plataforma e os comprimentos dos atuadores não é trivial e envolve restrições geométricas acopladas.

* + 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. Este é o problema mais direto do ponto de vista computacional: aplicam-se transformações homogêneas (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.

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

* Referencial de base
* Referencial da plataforma móvel

Cada pistão conecta um ponto fixo na base, descrito por a um ponto fixo na plataforma, descrito por .

A pose da plataforma é representada pelo par , em que é a posição do centro da plataforma em relação a e é a matriz de rotação que transforma vetores de para .

O ponto expresso no referencial da base é obtido por:

Assim, o vetor correspondente ao atuador é:

E seu comprimento é:

Essa formulação vetorial constitui a base da cinemática inversa, pois relaciona diretamente a pose desejada com os comprimentos dos atuadores.

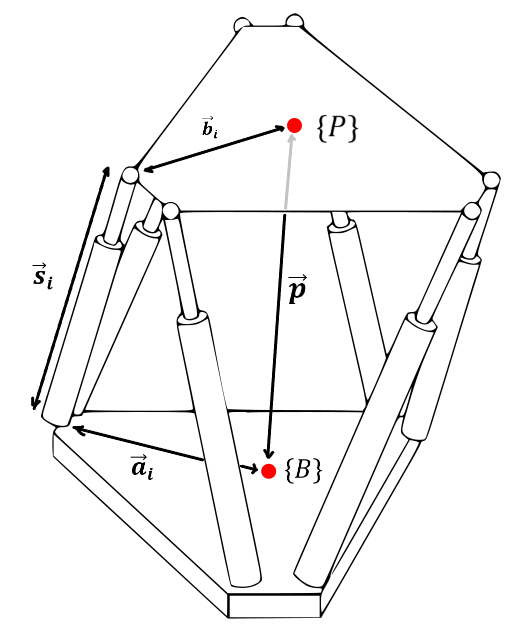


Figura 6: Representação Vetorial da Cinemática da Plataforma de Stewart

Fonte: Adaptado de Robótica (CRAIG, 2012)

* + 1. Cinemática Direta

A cinemática direta da plataforma de Stewart consiste em determinar a pose da plataforma móvel, definida pelo vetor de posição e pela matriz de rotação , a partir dos comprimentos conhecidos dos seis atuadores . 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.

Assim como na cinemática inversa, cada atuador liga um ponto fixo da base a um ponto da plataforma móvel . Quando a plataforma assume a pose , o ponto da plataforma expresso no referencial da base é dado por:

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

E o comprimento medido do atuador é:

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

Este conjunto de seis equações não lineares, envolvendo raízes, produtos cruzados e combinações de seno e cosseno advindos de , 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 são fisicamente impossíveis, espelhadas ou instáveis. 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:

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.

* 1. 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. Pesquisas recentes ampliaram ainda mais o escopo, incluindo manufatura avançada, reabilitação médica, estabilização de sensores e sistemas de energia oceânica.

* + 1. Rotinas de Movimento

As plataformas de Stewart são amplamente aplicadas em geração de movimentos controlados, onde 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. A literatura demonstra que plataformas paralelas 6-DOF são especialmente adequadas para esse tipo de tarefa, sendo utilizadas 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. 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. 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. 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 de aeronaves. Elas constituem, portanto, uma das aplicações mais versáteis da plataforma de Stewart, pois permitem controlar integralmente a pose ao longo do tempo com alto nível de repetibilidade, característica essencial para experimentação científica, modelagem, validação de controladores e simulações complexas.

* + 1. 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 sofrem movimentos de seis graus de liberdade (surge, sway, heave, roll, pitch e yaw) devido à ação de ondas, vento e correntes. Esses movimentos podem comprometer a precisão de operações, a segurança de pessoal e a qualidade de medições e processos.

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.

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.

* + 1. Controle por Dispositivos Manuais

Outra categoria importante de aplicação envolve o controle direto por 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, em sistemas de teleoperação, permitindo que um operador humano controle a orientação e posição de plataformas robóticas, em bancadas didáticas que facilitam a exploração interativa dos seis graus de liberdade e em IHMs voltadas para avaliação subjetiva de movimento. Mecanismos baseados em plataformas de Stewart podem funcionar como joysticks espaciais em 6 DOF, permitindo que o usuário controle sistemas complexos de forma contínua e intuitiva. Yildiz (2016), por exemplo, desenvolveu um joystick 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.

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 Yildiz (2012) reforçam essa abordagem ao demonstrar que uma plataforma Stewart 3×3 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 tanto ao ensino quanto à simulação profissional e à pesquisa, permitindo que operadores explorem o espaço de movimento em tempo real, testem limites do sistema e interajam de forma direta com mecanismos de seis graus de liberdade.

* 1. 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.

Antes de projetar um controlador, é necessário compreender o comportamento dinâmico da planta, geralmente representado por modelos de primeira e segunda ordem, que descrevem a relação entre entrada e saída de um sistema linear no domínio de Laplace.

Os sistemas de primeira ordem são caracterizados por uma única constante de tempo e ganho estático , com resposta ao degrau exponencial, tendendo gradualmente ao valor de regime permanente. Sua função de transferência típica é dada por:

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

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 .

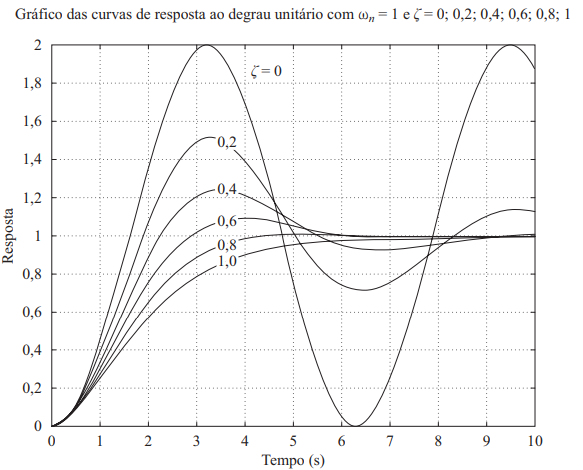


Figura 7: Comparação das curvas com diferentes .

Fonte: Engenharia de Controle Moderno (OGATA, 2011)

Sua função de transferência geral pode ser expressa por:

em que:

* é o ganho estático do sistema;
* é a frequência natural não amortecida;
* é o coeficiente de amortecimento, que determina a forma da resposta transitória.

De acordo com Garcia (2018), quando o sistema é superamortecido (), 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 () na modelagem.



Figura 8: Aproximação por modelo de 1ª ordem com tempo morto

Fonte: Controle de Processos Industriais (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:

em que , e são os ganhos proporcional, integral e derivativo, respectivamente; é o erro instantâneo; e 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.

* 1. CONTROLADOR ESP32-S3

O ESP32-S3 possui um processador dual-core Xtensa® LX7 operando a até 240 MHz, projetado para aplicações que demandam alto desempenho. Cada núcleo dispõe de unidades de ponto flutuante (FPU), que aceleram cálculos com números reais, garantindo maior precisão em algoritmos de controle contínuo, como o PID. Além disso, o chip conta com aceleração para instruções vetoriais (SIMD), permitindo o processamento simultâneo de múltiplos dados um recurso essencial em rotinas matemáticas vetoriais e de cinemática direta e inversa.

Outro diferencial é o suporte a instruções dedicadas de Inteligência Artificial (AI instructions), otimizadas para multiplicações e acumulações vetoriais (MAC), transformadas rápidas e operações matriciais. Essas instruções tornam o ESP32-S3 adequado não apenas para controle clássico, mas também para futuras pesquisas envolvendo aprendizado de máquina embarcado, controle adaptativo e predição de falhas.

Em termos de conectividade e expansão, o microcontrolador oferece 45 GPIOs configuráveis, incluindo até 20 canais PWM, 12 entradas analógicas (ADC de 12 bits), além de interfaces UART, SPI, I²C, I²S, CAN e USB OTG nativo, que permitem a integração com sensores, atuadores e sistemas supervisórios (ESPRESSIF SYSTEMS, 2024).

Assim, o ESP32-S3 combina elevado desempenho computacional, flexibilidade de Entrada/Saída (I/O) e baixo consumo de energia, apresentando-se como uma plataforma moderna, capaz de atender tanto às exigências do controle da plataforma de Stewart quanto a projetos futuros de maior complexidade.

* 1. Interfaces Gráficas em Sistemas de Controle

Nos sistemas de automação e robótica, a interface gráfica exerce o papel de camada de supervisão (IHM - Interface Homem-Máquina), permitindo que o operador visualize o comportamento do sistema em tempo real e interaja com os parâmetros de controle. Essa camada é responsável por transformar dados técnicos provenientes de sensores e atuadores em informações compreensíveis, como gráficos, indicadores e modelos tridimensionais.

Em aplicações como a plataforma de Stewart, a interface gráfica atua como o elo entre o sistema embarcado e o usuário, possibilitando o envio de setpoints, comandos de operação e o acompanhamento das respostas dinâmicas dos atuadores. Dessa forma, a HMI torna-se parte fundamental do ciclo de controle, contribuindo para o diagnóstico, ajuste e validação do desempenho do sistema.

* 1. Conceito de Aplicações Web e Arquitetura Cliente-Servidor

As aplicações web seguem o modelo cliente–servidor, no qual o cliente — geralmente um navegador — é responsável pela interface visual e pela interação com o usuário, enquanto o servidor realiza o processamento das informações e gerencia a comunicação com os dispositivos físicos. Essa arquitetura é amplamente utilizada por possibilitar acesso remoto, escalabilidade e interoperabilidade entre diferentes plataformas conectadas em rede.

De acordo com Jeng e Chieng (2020), as IHMs baseadas em tecnologias web têm se consolidado como uma solução eficaz para supervisão de processos industriais, pois permitem integração direta com controladores, operação multiplataforma e atualização contínua dos dados transmitidos pela planta. A combinação de protocolos como HTTP e WebSocket possibilita que o servidor disponibilize informações de forma assíncrona e bidirecional, garantindo baixa latência e mantendo o usuário constantemente atualizado sobre o estado do sistema. Essa abordagem, aliada à utilização de estruturas modulares no front-end, viabiliza a construção de interfaces responsivas, capazes de representar sinais de sensores e comandos de controle em tempo real, característica essencial em aplicações de robótica e sistemas dinâmicos.

No contexto deste projeto, essa arquitetura de HMI web foi fundamental para tornar possível a operação supervisória da plataforma de Stewart diretamente pelo navegador. O servidor, implementado em Python, atua como intermediário na troca de dados entre o ESP32-S3 e o front-end, processando comandos, organizando telemetria e distribuindo as informações por meio de WebSocket com baixa latência. Com isso, a interface apresenta em tempo real as posições dos atuadores, estados internos e comandos de controle, permitindo a operação segura e responsiva do sistema. A adoção desse modelo elimina a dependência de softwares proprietários utilizados em estudos anteriores, além de garantir portabilidade, escalabilidade e compatibilidade com múltiplos sistemas operacionais.

* 1. Tecnologias Utilizadas na Camada Web

A camada de interface web é composta por tecnologias complementares, responsáveis pela estrutura, estilo e comportamento dinâmico das páginas.

* HTML (*HyperText Markup Language*): define a estrutura e o conteúdo da interface, organizando elementos como títulos, parágrafos, imagens e botões.
* CSS (*Cascading Style Sheets*): responsável pela formatação visual, controlando cores, fontes, espaçamento e disposição dos elementos na tela.
* JavaScript: linguagem de programação que confere interatividade e comportamento dinâmico às páginas. Permite atualizar gráficos, animar objetos 3D e processar dados recebidos do backend sem recarregar a página.
* Python: utilizado no backend por sua simplicidade e ampla gama de bibliotecas para controle e comunicação. Atua como servidor de dados, realizando a ponte entre o microcontrolador (ESP32-S3) e o navegador, por meio de APIs REST ou conexões WebSocket.

Essas tecnologias, quando integradas, permitem o desenvolvimento de interfaces modernas, responsivas e capazes de processar informações em tempo real, características fundamentais para aplicações de controle e monitoramento.

1. MÉTODO E PROCEDIMENTOS METODOLÓGICOS

Este capítulo descreve os procedimentos adotados no desenvolvimento do projeto, desde a análise da bancada existente até a implementação da nova estrutura de controle utilizando o microcontrolador ESP32-S3. A metodologia foi dividida em etapas para facilitar o entendimento do processo de adaptação e reestruturação da plataforma Stewart.

* 1. Análise da Bancada Existente no IFSP

Inicialmente, foi realizado um mapeamento completo da bancada disponível no Laboratório do IFSP, Campus São José dos Campos. 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.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 9: Estrutura original da bancada com CLP Siemens e atuadores lineares

Fonte: Case Controlador Portátil e Plataforma de Stewart (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 e de seu alto custo no mercado, optou-se por buscar uma solução mais acessível, no caso, o ESP32-S3.

* 1. Adaptação da Bancada para o ESP32-S3

A adaptação da bancada envolveu diversas modificações tanto na parte eletrônica quanto na integração lógica e estrutural. As principais alterações realizadas:

* + 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 velocidade dos atuadores;
* Gerar sinais digitais para as entradas de direção (IN1 e IN2).
* Realizar leitura dos sinais de feedback (via ADC);
* Executar o algoritmo de controle PID;
* Comunicar com sistemas externos via serial.
  + 1. 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.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 12: Funcionamento conversores NPN/PNP

Fonte: Case Controlador Portátil e Plataforma de Stewart (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:

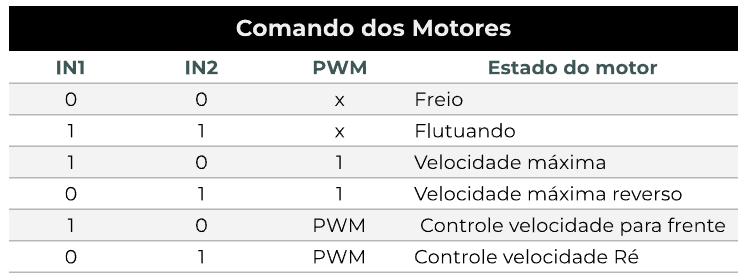


Figura 10: Comando dos Motores

Fonte: Driver Motor Ponte H (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)

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:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 11: Arquitetura original com CLP e conversores NPN/PNP

Fonte: Case Controlador Portátil e Plataforma de Stewart (GONÇALVES, 2023)

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 12: Funcionamento conversores NPN/PNP

Fonte: Case Controlador Portátil e Plataforma de Stewart (GONÇALVES, 2023)

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 13: Arquitetura atual com ESP32-S3 e controle direto

* + 1. **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.

* + 1. 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 14: Atuador Linear

Fonte: Case Controlador Portátil e Plataforma de Stewart (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.

* + 1. 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.

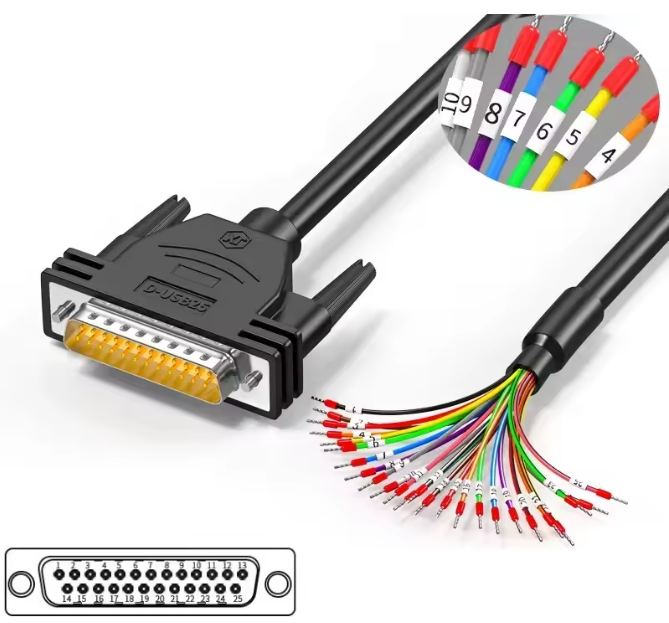
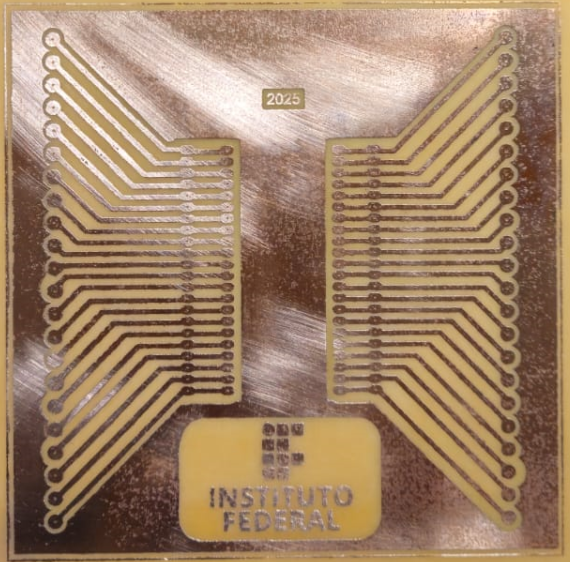


Figura 15: Conector DB37 com fios numerados

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.

 Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Figura 16: PCB para ESP32S3

* + 1. Mapeamento dos Sinais do Conector DB37

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 a seguir 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):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

* 1. 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.

* + 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 X), no qual as curvas de feedback mostram variações bruscas e não coerentes com o comportamento físico esperado.

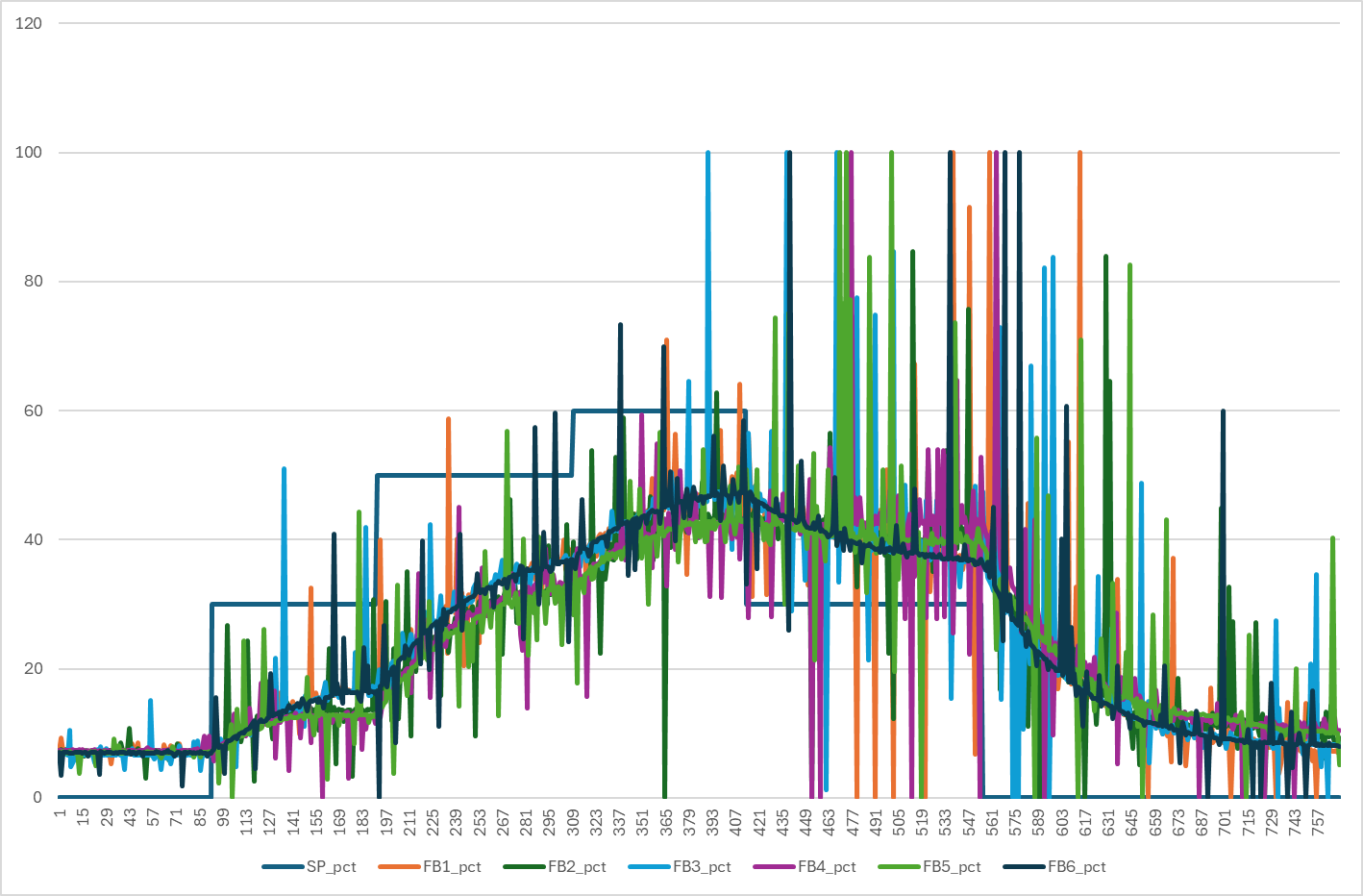


Figura 17: Aquisição inicial - Ruídos

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 (fc = 4 Hz, valor inicial), reduzindo ruídos de medição sem prejudicar a dinâmica de resposta. O resultado do uso do filtro é apresentado na Figura Y, onde se nota a suavização significativa dos sinais e maior coerência entre o setpoint e o retorno de posição de cada pistão.

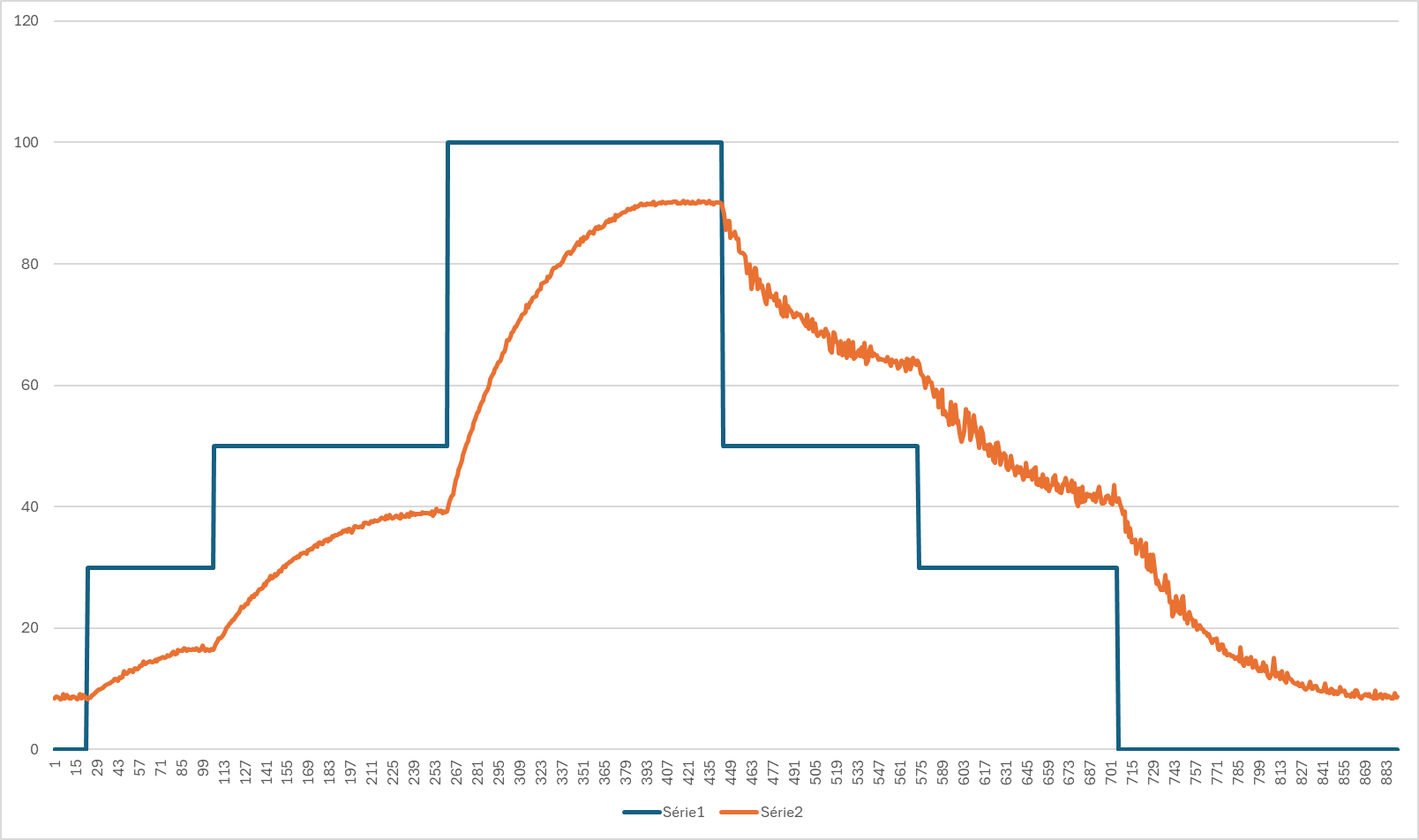


Figura 18: Resposta com filtro passa-baixa

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 Z) 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.

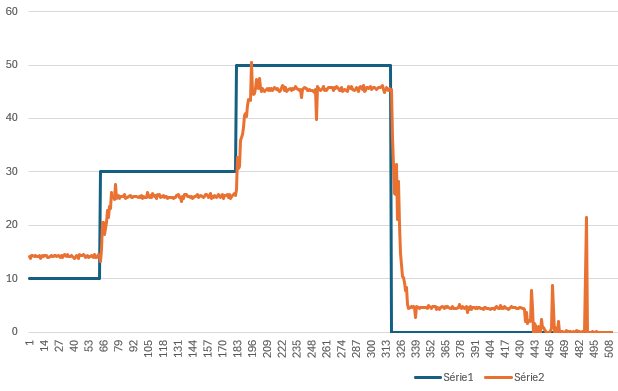


Figura 19: Resposta com filtro média móvel (janela de 8)

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 “glitches” 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, . No firmware, adotou-se , 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 W 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.

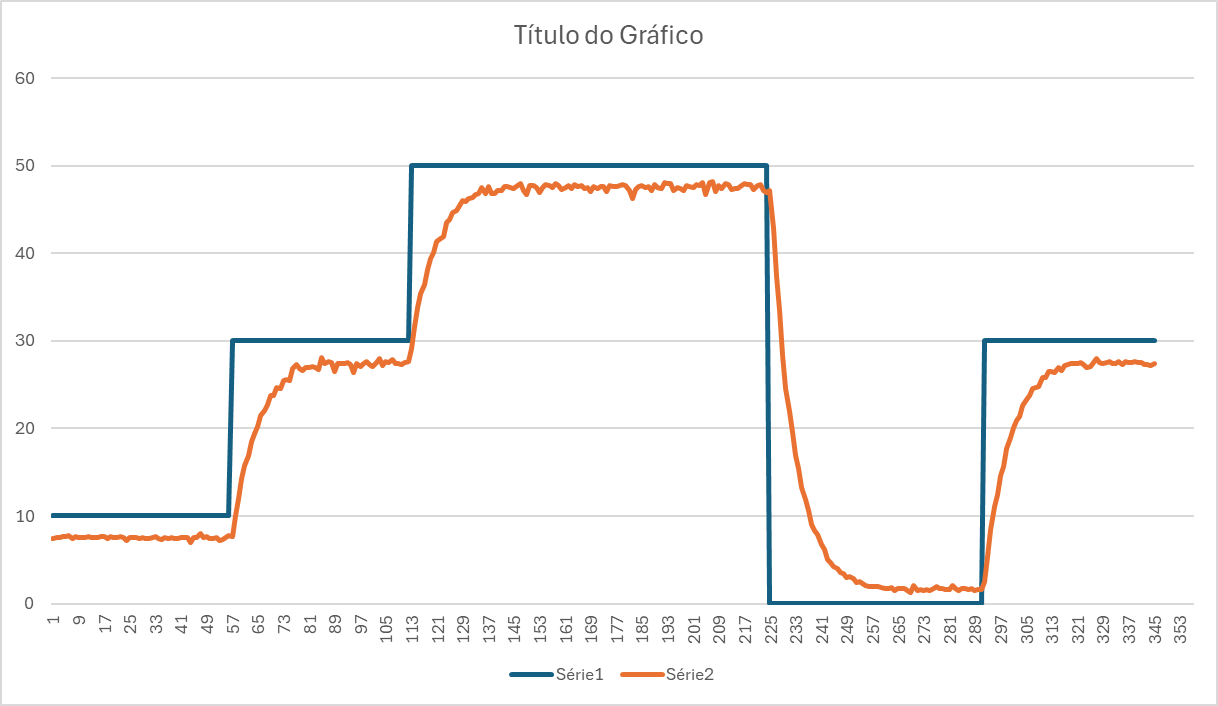


Figura 20: Resposta com Filtro Anti-Picos (mediana + limitador de inclinação)

* + 1. 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 Z) 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.

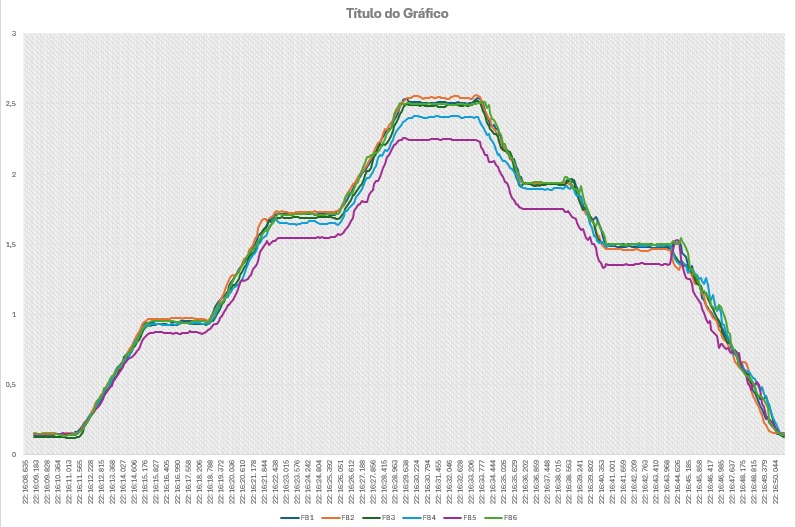


Figura 21: Deslocamento dos pistões com aplicação simultânea de PWM

* + 1. MAPEAMENTO TENSÃO → POSIÇÃO (V0–V100)

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 (V0 e V100), mapeando tensão em posição (mm) ao longo do curso útil. Assim, o sistema interpreta V0 como posição zero e utiliza o *span* para definir o ganho de conversão, assegurando coerência entre a posição física dos atuadores e os valores empregados pelo controle.

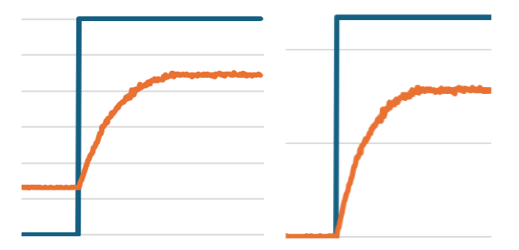


Figura 22: Correção de Offset

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

1. V0: a tensão com o pistão no início do curso (definido como 0 mm).
2. V100: a tensão com o pistão no fim do curso útil (definido como mm).

Assume-se que, dentro desse intervalo, a relação é aproximadamente linear. Assim, qualquer leitura entre V0 e V100 é convertida para posição por uma regra de três:

Em termos práticos, V0 corrige o zero do sensor (elimina o erro de “zeragem”) e V100 fixa o ganho (define “quantos volts valem 1 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 . 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.

* + 1. COMPENSAÇÃO DE ZONA MORTA/ATRITO

A operação do sistema mecânico evidenciou não linearidades, como atrito estático e viscoso, histerese, folgas mecânicas, pequenos desalinhamentos 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 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 “travavam”, 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: (avançar) e (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 e foram somados ao sinal de controle apenas enquanto o erro permanecia fora da faixa morta (*deadband*), fornecendo o impulso inicial de “arranque”. 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 pequenos degraus 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.

* 1. IDENTIFICAÇÃO E SINTONIA DE CONTROLE PARA SINCRONIA

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 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 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-picos, 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, (avançar) e (retornar), em substituição ao 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.

* + 1. FASE 1 – Aquisição de Dados

Optou-se por realizar a identificação do modelo com o sistema operando em malha fechada, por razões práticas, como mitigar não linearidades inerentes ao sistema físico, zona morta, atrito estático e pontos de travamento ao longo do curso.

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 degraus de referência (e sequências de degraus) com , registrando a entrada efetiva aplicada a cada pista, o setpoint em milímetros, a saída em milímetros, convertida pela calibração V0–V100 e filtrada e o tempo de aquisição.

Para simplificar e padronizar as coletas, foi desenvolvido um aplicativo próprio em Python/Tkinter denominado **(Figura X**). 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 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.

Texto

O conteúdo gerado por IA pode estar incorreto.

Figura 23: Interface do Aplicativo de Coleta de Dados

Dados obtidos:

Interface gráfica do usuário, Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 24: Dados Obtidos do Pistão 1

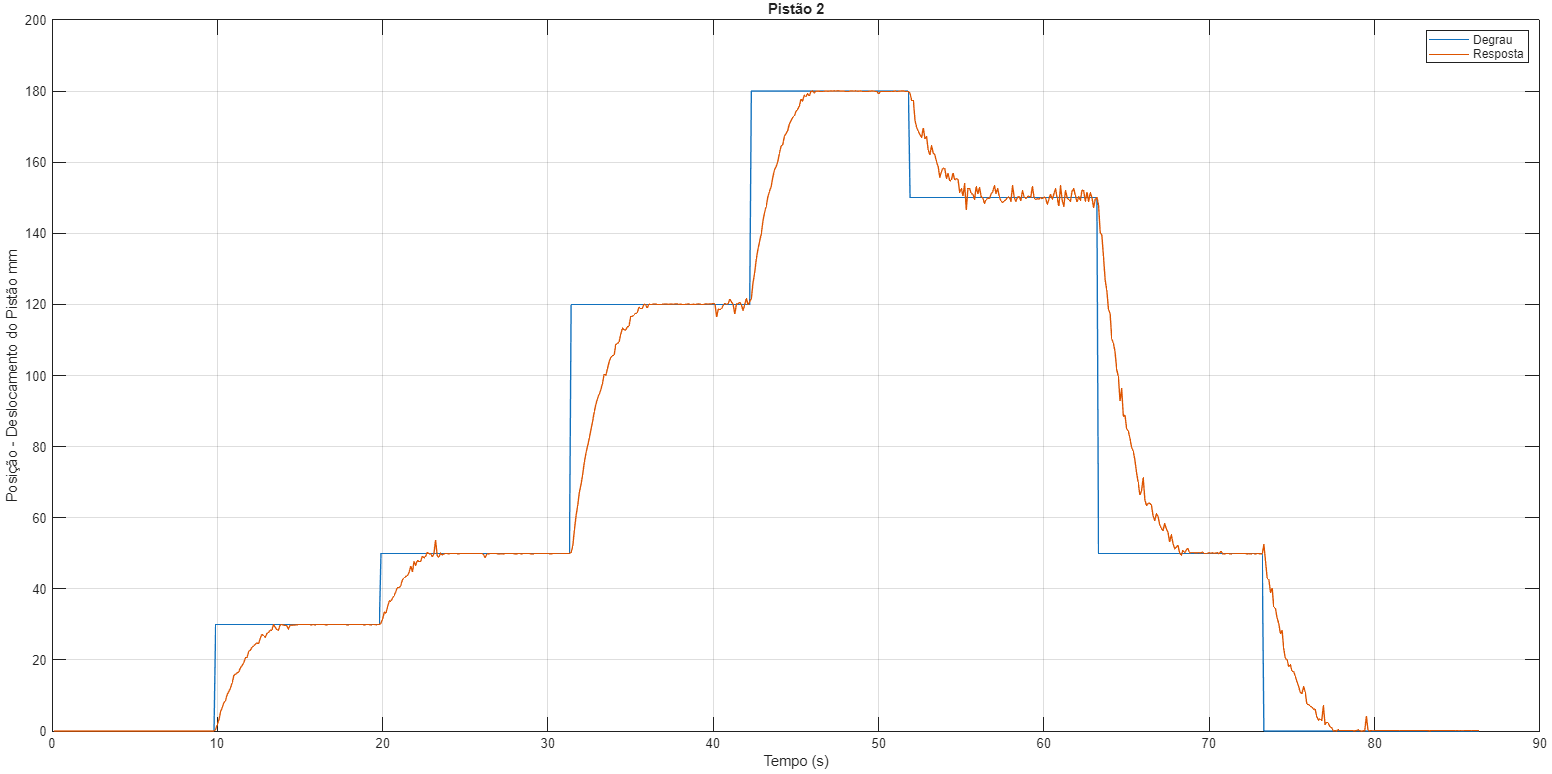


Figura 25: Dados Obtidos do Pistão 2

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 26: Dados Obtidos do Pistão 3

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 27: Dados Obtidos do Pistão 4

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 28: Dados Obtidos do Pistão 5

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 29: Dados Obtidos do Pistão 6

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 Kₚ = 5, os pistões apresentavam um comportamento mais linear e uniforme ao longo do curso.

Nessa etapa, foi implementado um filtro anti-picos 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 (avançar) e (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:

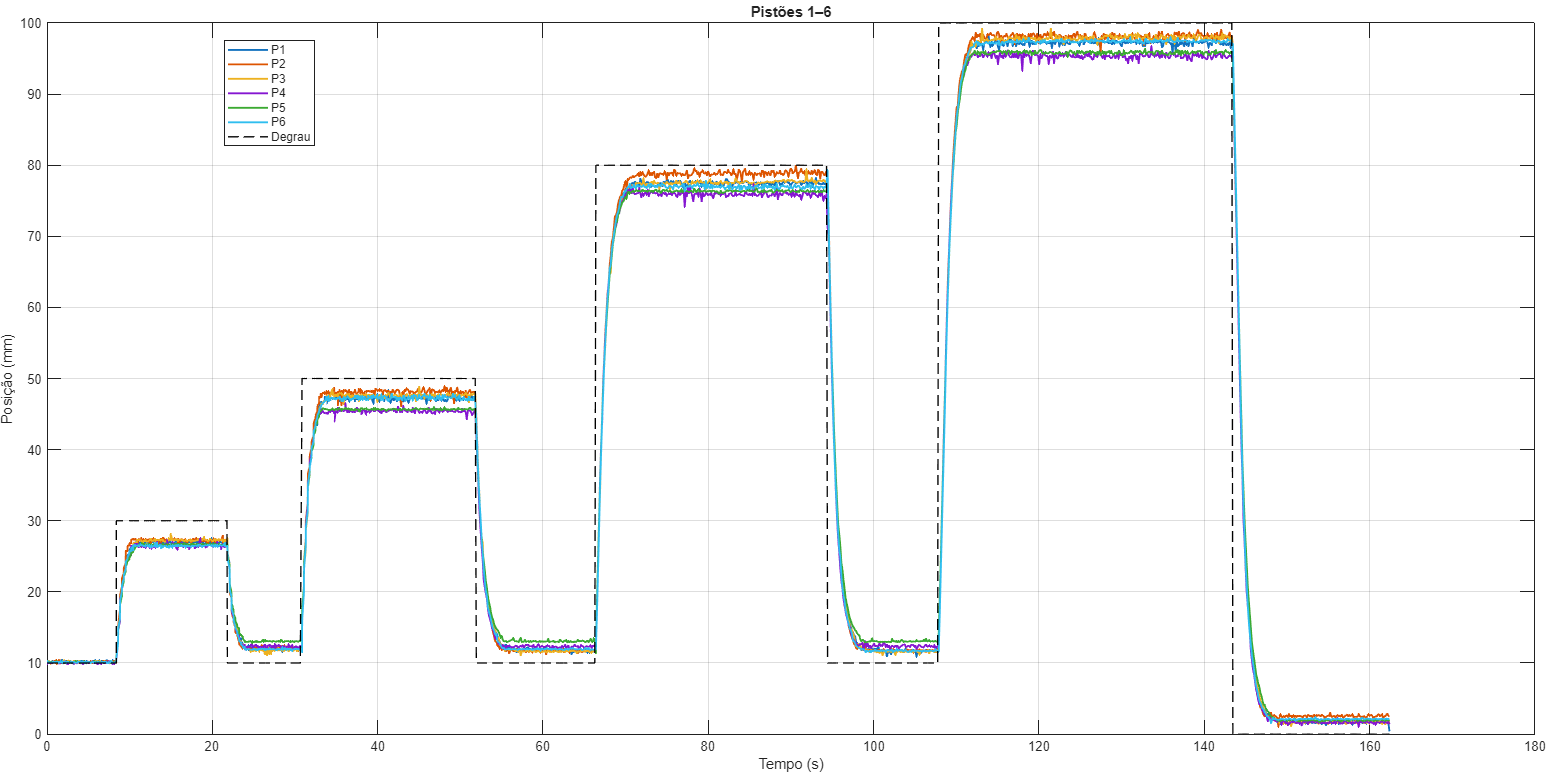


Figura 30: Coleta simultânea dos 6 pistões com Kp = 5

* + 1. FASE 2 - Identificação de modelo (MATLAB)

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:

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Figura 31: Dados Importados para o System Identification

* + - 1. Estimação de um modelo:

Para o caso utilizando uma estimação de segunda ordem.

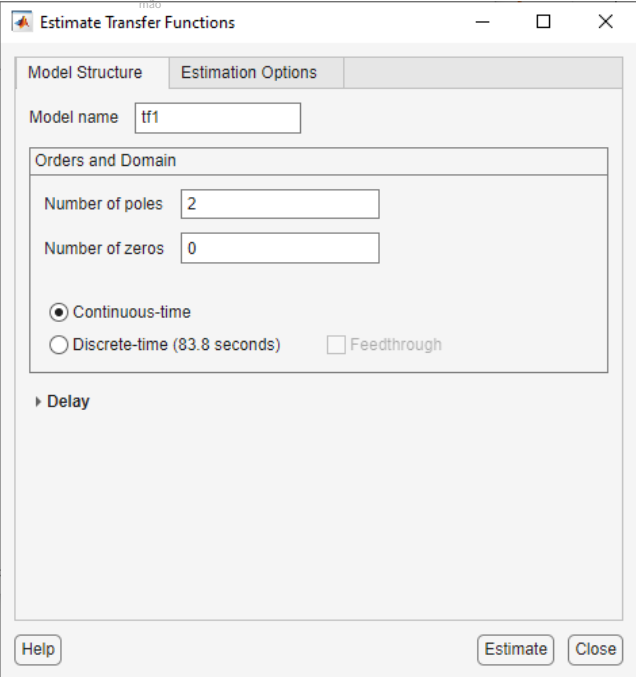


Figura 32: Configurações do Modelo

* + - 1. 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.

Gráfico, Gráfico de linhas

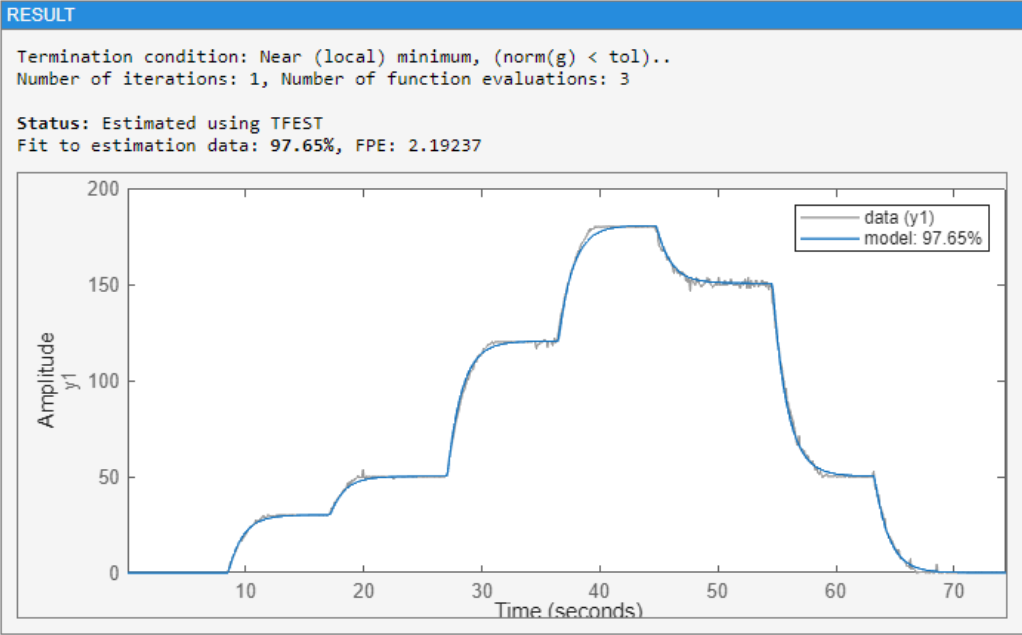
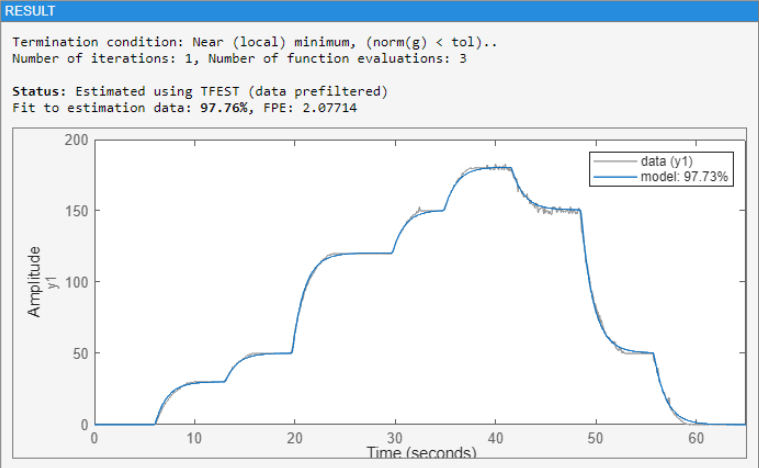
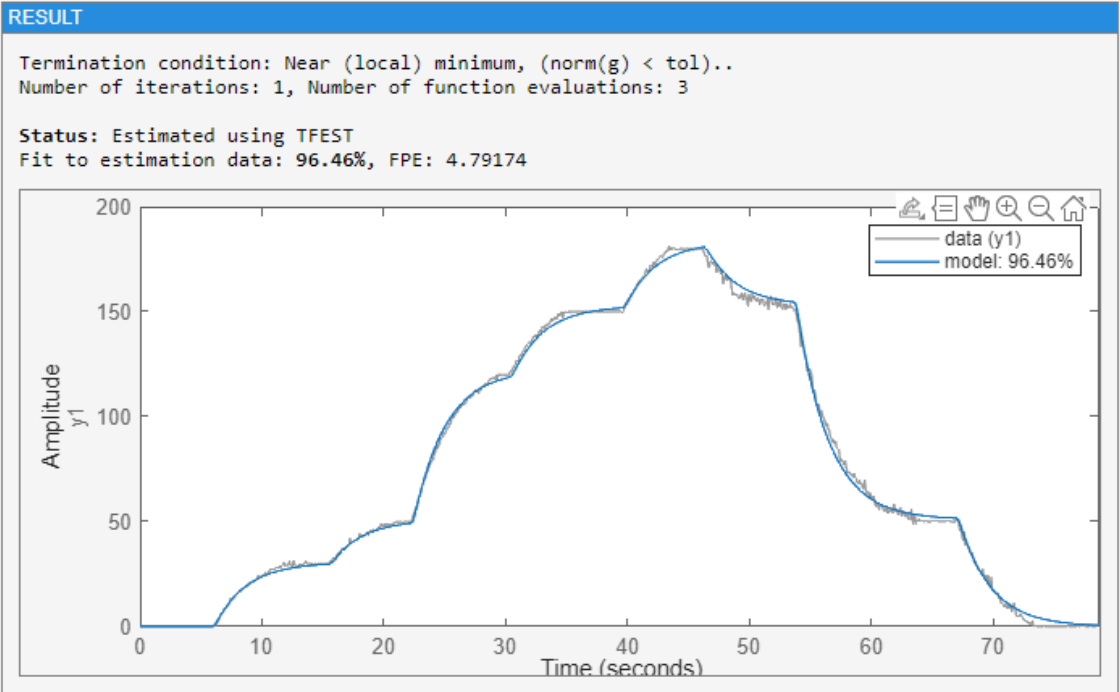
O conteúdo gerado por IA pode estar incorreto. 

Figura 33: Resultado do Modelo Gerado Pistão 1 e 2

 Interface gráfica do usuário, Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 34: Resultado do Modelo Gerado Pistão 3 e 4

 Interface gráfica do usuário, Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 35: Resultado do Modelo Gerado Pistão 5 e 6

Funções de Transferência em malha fechada obtidas:

Apesar dos resultados satisfatórios obtidos em simulação, a aplicação prática desses ganhos 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, 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.

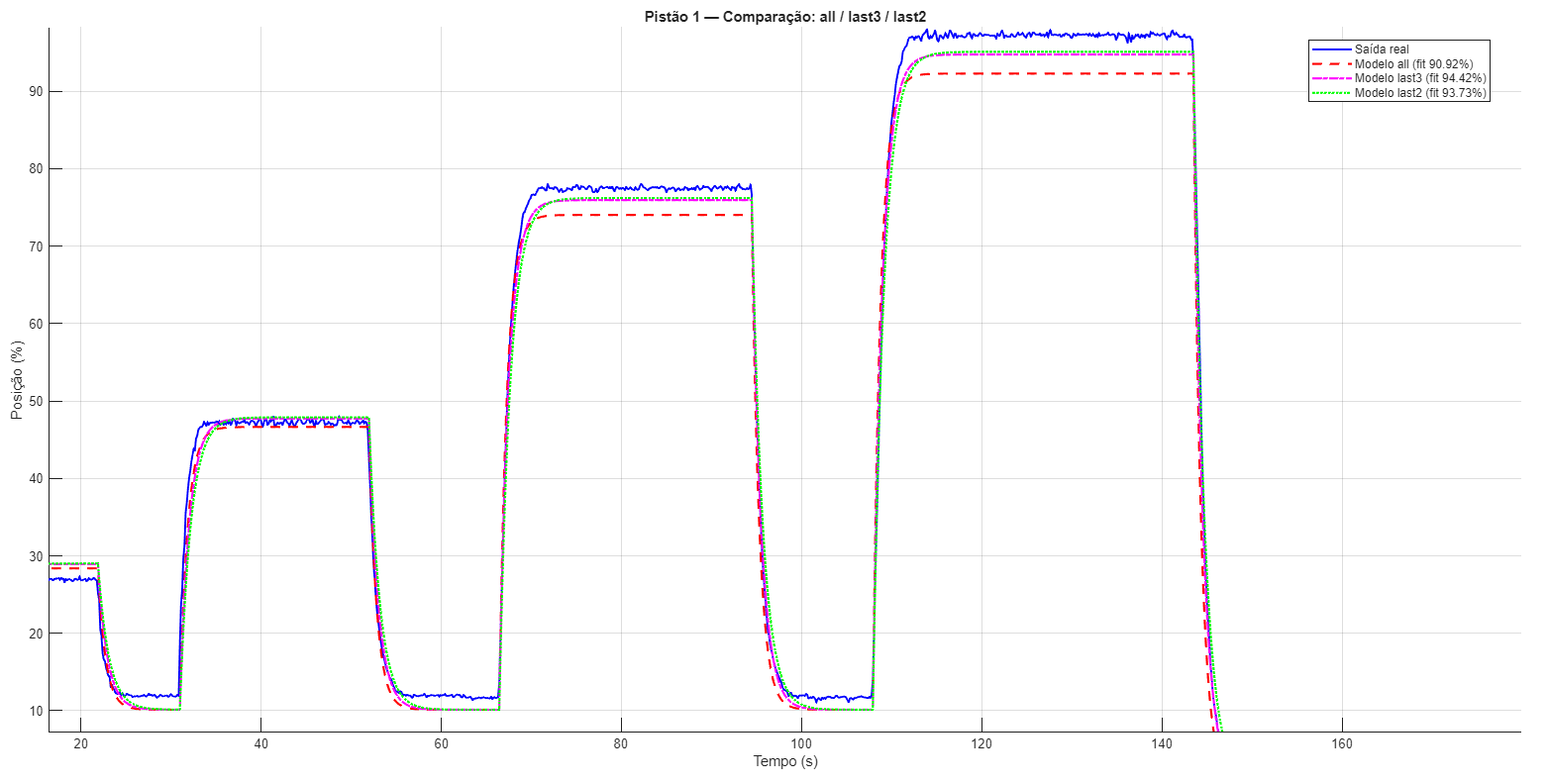


Figura 36: Pistão 1 – Comparação entre as FTs

Gráfico, Histograma

O conteúdo gerado por IA pode estar incorreto.

Figura 37: Pistão 1 – Erro ao longo do tempo

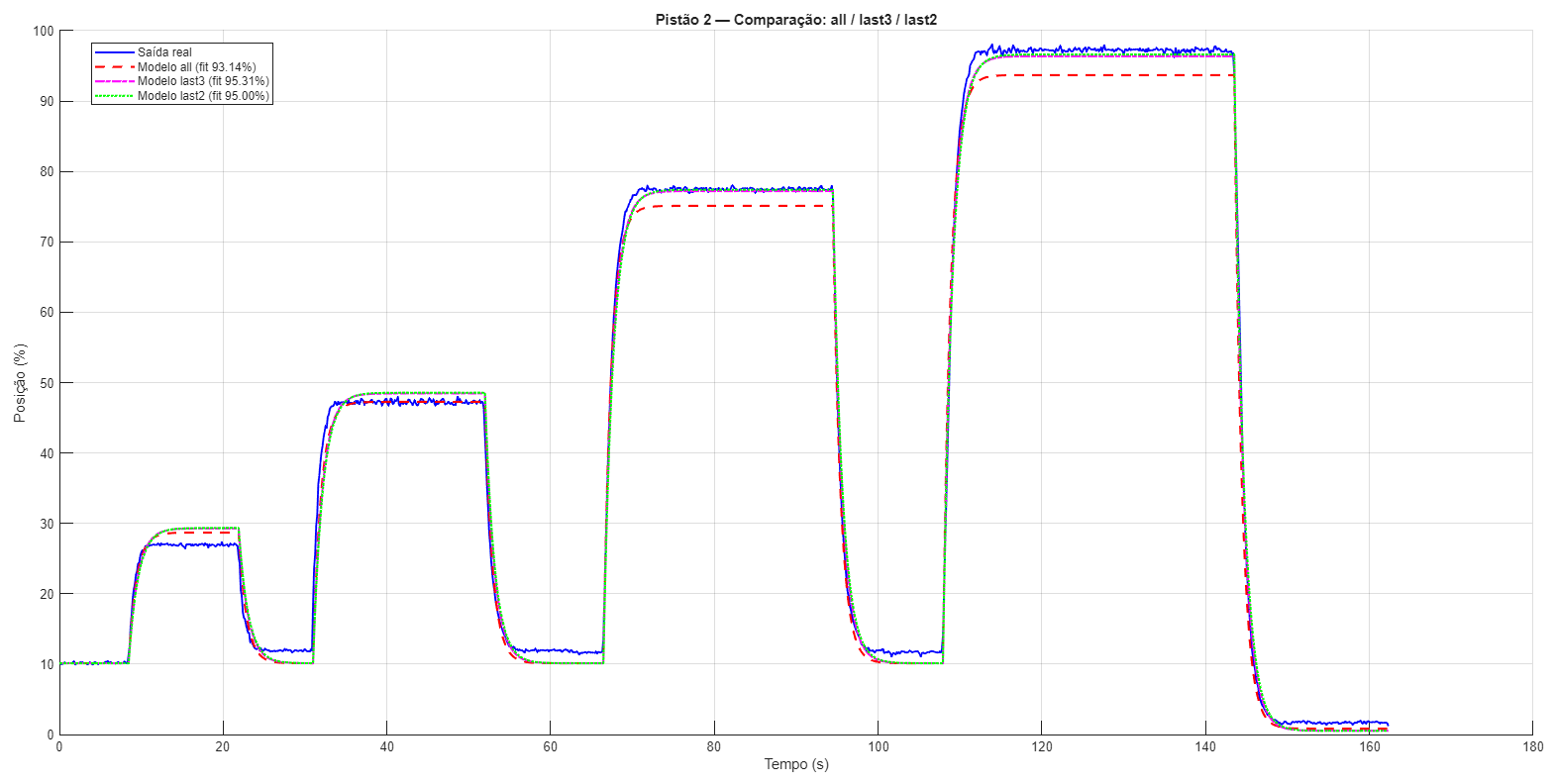


Figura 38: Pistão 2 – Comparação entre as FTs

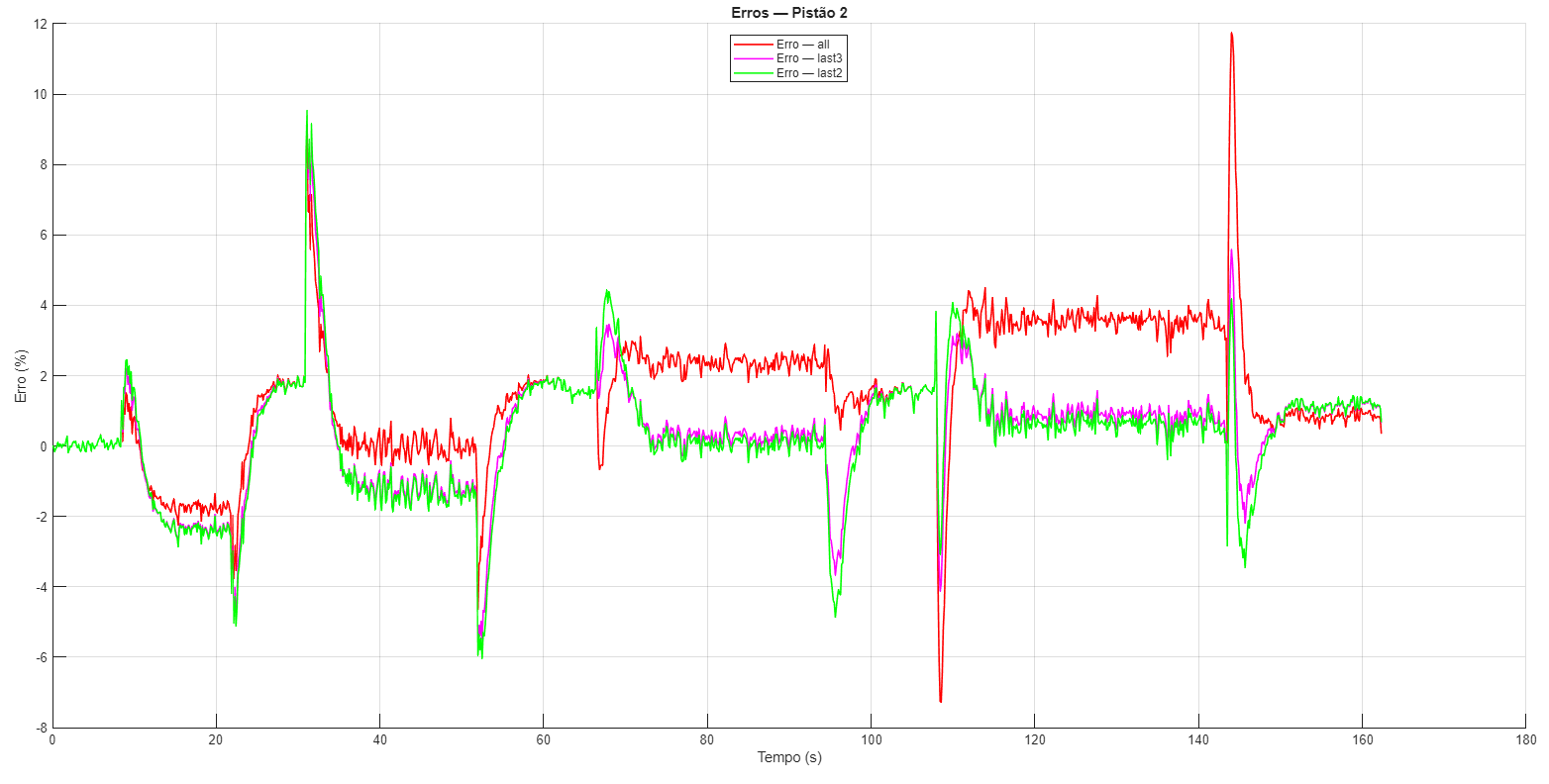


Figura 39: Pistão 2 – Erro ao longo do tempo

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 40: Pistão 3 – Comparação entre as FTs

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 41: Pistão 3 – Erro ao longo do tempo

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 42: Pistão 4 – Comparação entre as FTs

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 43: Pistão 4 – Erro ao longo do tempo

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 44: Pistão 5 – Comparação entre as FTs

Gráfico, Histograma

O conteúdo gerado por IA pode estar incorreto.

Figura 45: Pistão 5 – Erro ao longo do tempo

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 46: Pistão 6 – Comparação entre as FTs

Gráfico, Gráfico de linhas

O conteúdo gerado por IA pode estar incorreto.

Figura 47: Pistão 6 – Erro ao longo do tempo

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.

Funções de transferência obtidas de primeira ordem em malha fechada

* + 1. FASE 3 - Abertura de malha e sintonia do controle.

Com as funções de transferência 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:

onde representa a função de transferência identificada em malha fechada e é o ganho proporcional aplicado durante o ensaio.

Dessa forma, para a primeira estratégia, adotou-se , obtendo-se:

Com as plantas 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) para todos os pistões. Dessa forma, ainda que existissem pequenas diferenças dinâmicas entre as plantas , 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 de cada pistão, fechando-se as respectivas malhas conforme:

As respostas a um degrau de 30 mm foram simuladas para cada controlador, e as curvas foram comparadas no mesmo gráfico para avaliar o alinhamento temporal e garantir sobressinal desprezível.

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 48: Resultado Obtido – Curva 6 Pistões Sobrepostas para Comparação

Quando necessário, realizaram-se ajustes finos na frequência de projeto 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.

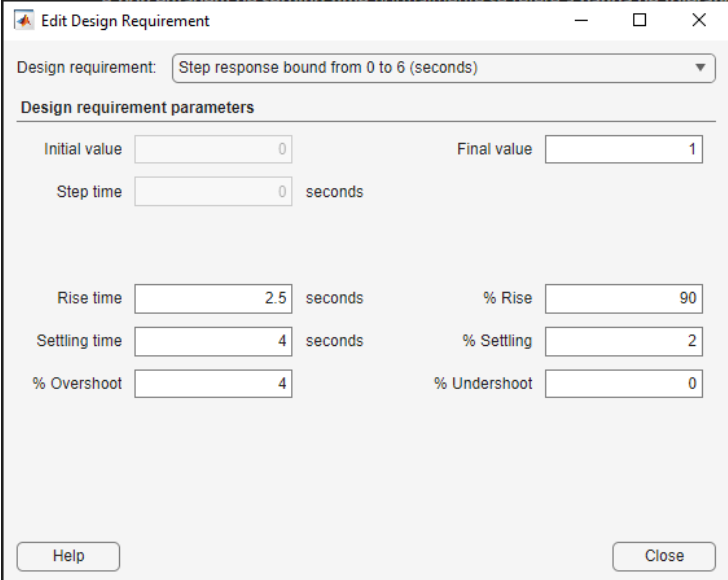


Figura 49: Especificações do Design

Gráfico

O conteúdo gerado por IA pode estar incorreto.Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 50: Resultado Desejado

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

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

Nessa etapa, ao realizar novos testes em bancada, observou-se que, com e , 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 sob o controlador .

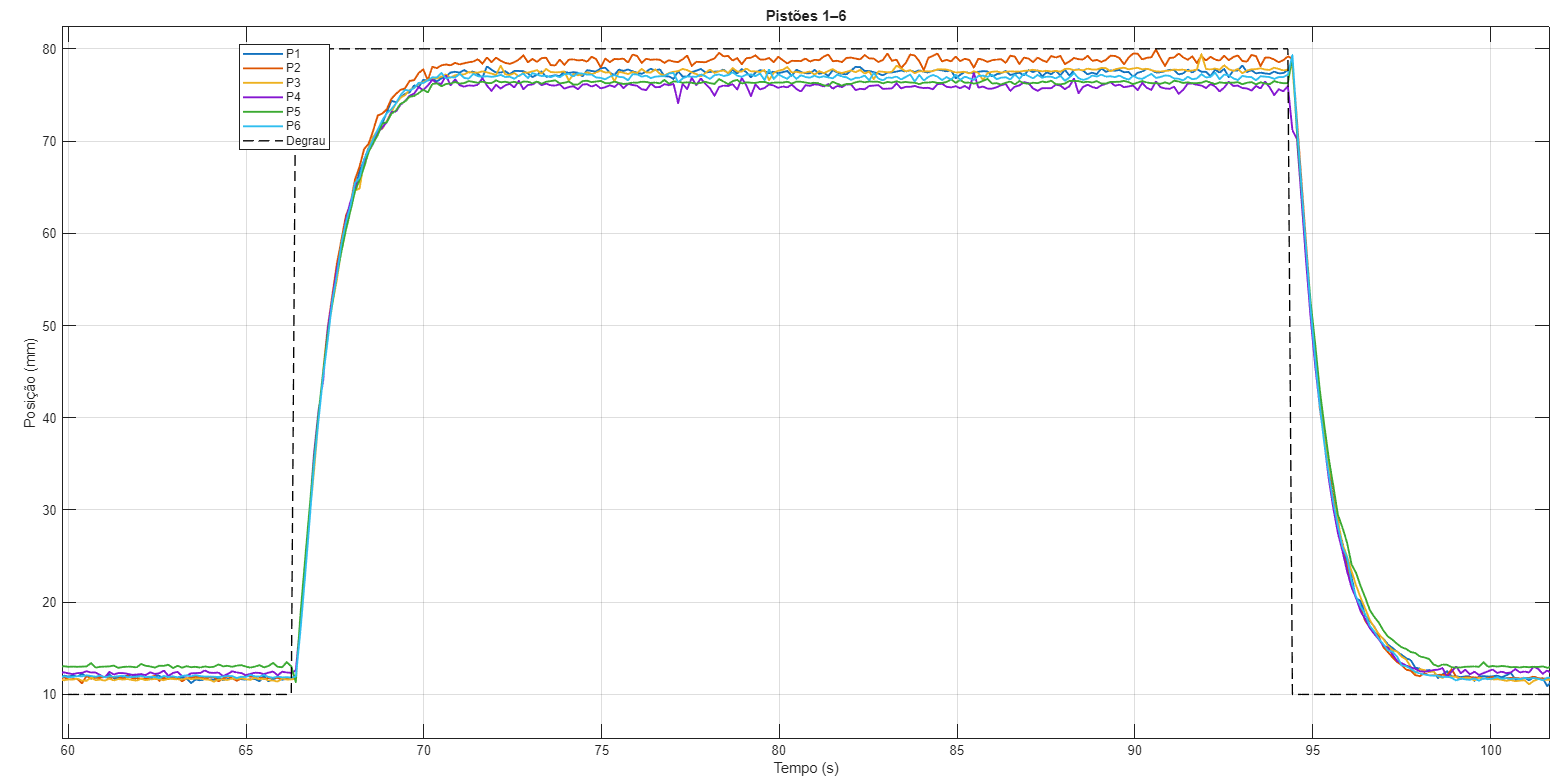


Figura 51: Resposta do ensaio

Com base nesse resultado, definiu-se o controlador de referência como:

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

Essa expressão permite que cada controlador 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 .

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

onde e são, respectivamente, os ganhos proporcional e integral determinados a partir da fatoração de ;

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.

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 52: Resposta antes da sincronização – Visão geral

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 53: Resposta antes da sincronização – zoom Tr e Ts

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.

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 54: Resposta após sincronização – Visão geral

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 55: Resposta antes da sincronização – zoom Tr e Ts

Todos os controladores apresentaram estrutura proporcional-integral (PI), uma vez que o termo derivativo () 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 são apresentados a seguir:

Observa-se que os valores de permanecem próximos de 5 em todos os casos, o que reforça a uniformidade dinâmica entre os atuadores após a aplicação da estratégia de equalização.

Já os valores de 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.

Essa sintonia final consolidou o comportamento sincronizado da plataforma, garantindo tempos de subida semelhantes e erro estacionário próximo de zero para todos os atuadores.

* 1. DESENVOLVIMENTO DA INTERFACE DE CONTROLE E MODOS DE OPERAÇÃO
     1. Arquitetura de comunicação

A solução adotada para o controle da plataforma combina um frontend estático em HTML/JavaScript, um backend desenvolvido em FastAPI e o firmware embarcado no ESP32-S3. As páginas web interagem com o backend por meio de requisições REST, utilizadas para operações síncronas. Para a recepção contínua de dados, um canal WebSocket (/ws/telemetry) foi implementado, garantindo atualização com baixa latência.

|  |  |  |
| --- | --- | --- |
| **ENDPOINTS 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 | Fecha 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. |

O backend mantém um gerenciador serial baseado em pyserial, responsável por traduzir requisições REST em comandos enviados ao ESP32-S3. Esse módulo implementa a supervisão da porta serial, normalização de unidades e validação de limites antes que qualquer comando alcance o microcontrolador. A telemetria enviada pelo firmware segue o formato CSV, incluindo posição atual dos pistões, valores de PWM e leituras de IMU. O backend repassa esses pacotes ao WebSocket, permitindo que o frontend atualize gráficos e indicadores.

* + 1. implementação painel pid

Para testes e execução do controle PID dos pistões foi implementada uma página com um painel PID, que consome o WebSocket para renderizar séries temporais de deslocamento e PWM utilizando uma biblioteca gráfica chamada Chart.js.

Essa página foi criada com o objetivo de permitir testes com setpoints globais e individuais de atuadores, registrando o setpoint e a resposta real, possibilitando análise do comportamento do sistema de forma prática. Além disso, para análises offline dos dados no MATLAB, Python ou ferramentas equivalentes, os dados podem ser exportados em formato CSV.

O painel PID também disponibiliza controles para ajuste dos parâmetros Kp, Ki e Kd, enviados ao backend via /pid/gains.O backend persiste essas configurações, encaminha ao firmware e confirma a aplicação. Parâmetros complementares, como PWM mínimo e deadband, podem ser configurados pelo endpoint /pid/settings.

|  |  |  |
| --- | --- | --- |
| **ENDPOINTS 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). | |

* + 1. implementação da cinemática

A implementação da cinemática inversa no backend foi inspirada no estudo conceitual apresentado por RAW Labs em “Inverse Kinematics of a Stewart Platform” (RAW.org, 2020), que descreve uma formulação vetorial clara para obter os comprimentos dos atuadores a partir da pose desejada. A partir dessa abordagem, o presente trabalho adaptou o modelo matemático para os parâmetros geométricos reais da bancada, considerando vetores de ligação

O algoritmo foi implementado em Python, dentro do backend FastAPI, utilizando NumPy para manipulação matricial em tempo real. A função exposta no endpoint /calculate recebe a pose desejada (x, y, z, roll, pitch, yaw), calcula os comprimentos dos seis atuadores, verifica limites geométricos e retorna ao frontend tanto os valores numéricos quanto os pontos 3D para renderização em Three.js. Essa abordagem permitiu manter a execução leve, consistente com o método apresentado no projeto de referência, porém estendida para comunicação direta com o ESP32-S3 via comandos seriais.

* + 1. implementação das rotinas de movimento

As rotinas de movimento foram estruturadas como geradores parametrizáveis no backend, contemplando trajetórias senoidais, circulares e helicoidais. O frontend routines.html permite configurar amplitude, frequência, deslocamentos, duração e demais parâmetros. Ao iniciar uma rotina, essas configurações são enviadas ao endpoint /motion/start, que inicializa um laço de amostragem periódica.

A cada iteração, o backend calcula a pose alvo correspondente ao instante de tempo, valida por cinemática inversa e agenda o envio ao ESP32-S3. Simultaneamente, um pacote de dados contendo a pose comandada e a telemetria real é transmitido via WebSocket, permitindo que a interface exiba erros de seguimento e registre trajetórias automaticamente em IndexedDB.

* + 1. implementação do controle por imu

O modo de controle por IMU utiliza sensores como BNO085 ou MPU6050 conectados a um módulo transmissor que envia orientações ao ESP32-S3 via ESP-NOW. O firmware recebe quaternions ou ângulos de Euler.

No backend, o modo é habilitado através da página accelerometer.html, que controla a ativação do sistema via REST e acompanha a orientação recebida pelo WebSocket. Para estabilização, a orientação medida é convertida em uma pose alvo inversa, calculada pela cinemática inversa e enviada ao microcontrolador. O firmware incorpora limites de saturação e mecanismos de anti-windup, garantindo segurança mesmo na presença de perturbações rápidas.

* + 1. implementação do controle por joystick

O controle manual é realizado pela página controller.html, que captura eventos de de um controle de videogame padrão. Esses comandos são normalizados em uma estrutura de pose e enviados ao endpoint /joystick/pose.

O backend converte cada comando do usuário em comprimentos de atuadores por meio da cinemática inversa, aplicando limites de curso para poses proibidas. A telemetria retornada pelo WebSocket é utilizada para atualizar a interface em tempo real, destacando saturações ou desvios. Em caso de desconexão do dispositivo, o modo de joystick é automaticamente desabilitado e a plataforma retorna ao setpoint neutro.

* + 1. Implementação da Integração com o FlightGear

A integração com o simulador FlightGear é realizada por meio de um módulo dedicado (fg-bridge.py), que se conecta ao servidor Telnet do simulador para leitura de roll e pitch. Esses valores passam por saturação configurável e são convertidos em uma pose segura, com Z e yaw fixos.

Antes de aplicar o movimento à plataforma, o backend solicita ao endpoint /calculate a verificação da pose. A pré-visualização é enviada a /flight-simulation/preview, permitindo que o frontend exiba a trajetória prevista. A aplicação efetiva ocorre apenas quando a flag de segurança (enabled) retornada por /flight-simulation/status está ativa.

# 4 APRESENTAÇÃO E ANÁLISE DE RESULTADOS

xxxxxxxxxxxxxxxxx Parágrafo introdutório.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

# 5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

xxxxxxxxxxxxxxxxx Parágrafo introdutório.

# 5.1 Conclusões

Fechamento do problema, objetivos - geral e específicos. cumprimento da proposta do trabalho, estabelecido no capítulo 1

# 5.2 Considerações Finais

considerações a cerca do que foi desenvolvido na proposta/implicações/impactos/influências de resultados.

# 5.3 Limitações da Pesquisa

limitações no desenvolvimento do trabalho, condições de contorno do método e implicações

# 5.4 Proposta de Trabalhos Futuros

A partir do desenvolvimento do trabalho e das limitações apresentadas, se estabelece as seguintes propostas de trabalhos futuros:

1. ....;
2. ....;
3. ....;

# REFERÊNCIAS

1. CRAIG, John J. **Robótica**. São Paulo: Pearson Prentice Hall, 2012.
2. ESPRESSIF SYSTEMS. **ESP32-S3 Datasheet**. Version 2.7. Espressif Systems, 2024. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf>. Acesso em: 5 out. 2025.
3. GARCIA, Cláudio. **Controle de Processos**. v. 1. São Paulo: Edgard Blücher, 2018.
4. GONÇALVES, Felipe D. **Case Controlador Portátil e Plataforma de Stewart**. Trabalho de Conclusão de Curso – Instituto Federal de São Paulo – Campus São José dos Campos, 2023.
5. HERNÁNDEZ-GUZMÁN, V. M.; SILVA-ORTIGOZA, R. **Automatic Control with Experiments**. Cham: Springer, 2019. (Advanced Textbooks in Control and Signal Processing).
6. IFR – International Federation of Robotics. **Global Robot Density in Factories Doubled in Seven Years**. 2024. Disponível em: [https://ifr.org/wr-industrial-robots/#](https://ifr.org/wr-industrial-robots/). Acesso em: 20 jan. 2025.
7. IFR – International Federation of Robotics. **World Robotics: Industrial Robots**. 2024. Disponível em: <https://ifr.org/ifr-press-releases/news/global-robot-density-in-factories-doubled-in-seven-years>. Acesso em: 21 jan. 2025.
8. JENG, Shyr-Long; CHIENG, Wei-Hua. **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.
9. MATHWORKS. MATLAB: **The Language of Technical Computing. Version R2025b**. Natick, MA: The MathWorks, Inc., 2025.
10. OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.
11. PINHO, André Gurgel et al. **Desenvolvimento de bancada didática contendo múltiplos sensores e atuadores**. Research, Society and Development, v. 10, n. 13, e222101321165, 2021.
12. ROSSITER, John Anthony et al. **Control education for societal-scale challenges: A community roadmap**. Annual Reviews in Control, v. 55, p. 1–17, 2023.
13. STEWART, D. **A Platform with Six Degrees of Freedom**. Proceedings of the Institution of Mechanical Engineers, v. 180, n. 1, p. 371–386, 1965.
14. MERLET, Jean-Pierre; GOSSELIN, Clément. **Parallel Mechanisms and Robots**. In: SICILIANO, Bruno; KHATIB, Oussama (eds.). *Springer Handbook of Robotics*. Berlin: Springer, 2008. p. 269–285.
15. 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**, v. 147, p. 1–17, 2025.
16. 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*, v. 6, p. 263-281, 1992.
17. 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.
18. 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*, v. 2021, p. 1-15, 2021.
19. 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. Disponível em: <https://www.davidpublisher.com>\\*. Acesso em: 17/11/2025.
20. 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.
21. 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. Disponível em: <https://tsapps.nist.gov/publication/get\_pdf.cfm?pub\_id=820425>\\*. Acesso em: 17/11/2025.