



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO  
UNIDADE ACADÊMICA DO CABO DE SANTO AGOSTINHO  
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

Thiago Elias dos Santos

Projeto e controle de um sistema do tipo barra-bola.

Thiago Elias dos Santos

Projeto e controle de um sistema do tipo barra-bola.

Monografia apresentada ao curso de Graduação em Engenharia Eletrônica da Unidade Acadêmica do Cabo de Santo Agostinho, Universidade Federal Rural de Pernambuco, como requisito para a obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Dr. Reinel Beltrán Aguedo

Cabo de Santo Agostinho – PE

2025

Dados Internacionais de Catalogação na Publicação  
Sistema Integrado de Bibliotecas da UFRPE  
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

S237p Santos, Thiago Elias dos.  
Projeto e controle de um sistema do tipo barra-  
bola / Thiago Elias dos Santos. - Cabo de Santo  
Agostinho, 2025.  
88 f.; il.

Orientador(a): Reinel Beltrán Aguedo.

Trabalho de Conclusão de Curso (Graduação) –  
Universidade Federal Rural de Pernambuco,  
Unidade Acadêmica Cabo de Santo Agostinho -  
UACSA, Bacharelado em Engenharia Eletrônica,  
Cabo de Santo Agostinho, BR-PE, 2025.

Inclui referências.

1. sistema barra-bola. 2. Sistemas dinâmicos  
(Física matemática). 3. Fuzzy (Inteligência  
artificial). 4. *Controladores PID* 5. Interação  
humano-máquina. I. Aguedo, Reinel Beltrán, orient.  
II. Título

CDD 621.3

Projeto e controle de um sistema do tipo barra-bola.

Monografia apresentada ao Curso de Graduação em Engenharia Eletrônica da Universidade Federal Rural de Pernambuco, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovada em: 20/03/2025.

**BANCA EXAMINADORA**

---

Dr. Reinel Beltrán Aguedo  
Orientador, UACSA-UFRPE

---

Dra. Ania Lussón Cervantes  
Examinadora, UACSA-UFRPE

---

Dr. Erb Ferreira Lins  
Examinador, UACSA-UFRPE

## RESUMO

Este trabalho aborda o desenvolvimento de um protótipo do sistema barra-bola, incluindo sua modelagem matemática e estratégias de controle, com o objetivo de comparar técnicas de controle como PID, I-PD, lógica fuzzy, realimentação de estados e controladores híbridos. A modelagem dinâmica do sistema foi derivada das leis de Newton, e o sistema foi construído usando materiais e componentes de baixo custo: um microcontrolador ESP32, um sensor VL53L0X (para medição da posição da bola) e um sensor ultrassônico HC-SR04 (para definição da referência desejada), acoplados a um servomotor TowerPro MG996R que regula a inclinação da barra. O projeto incorpora duas Interfaces Homem-Máquina: uma embarcada no protótipo para o ajuste de parâmetros e visualização de dados em tempo real, e outra externa para monitoramento remoto. A implementação utilizou ferramentas como MATLAB/Simulink (para simulação e projeto de controladores), Arduino IDE (programação do ESP32) e Visual Studio Code (desenvolvimento da IHM externa). Os testes de desempenho, realizados com entradas do tipo degrau e rampa, avaliaram os controladores por meio de critérios integrais e da análise das características da resposta transitória. Os resultados demonstram que cada técnica apresenta vantagens específicas, dependendo do critério analisado, destacando a importância da seleção contextual do controlador para otimizar o desempenho do sistema.

**Palavras-chave:** sistema barra-bola; controle de sistemas dinâmicos; controlador *fuzzy*; *fuzzy*-PID; interface Homem-Máquina; controlador PID.

## ABSTRACT

This work addresses the development of a ball-and-beam system prototype, including its mathematical modeling and control strategies, aiming to compare control techniques such as PID, I-PD, fuzzy logic, state feedback, and hybrid controllers. The system's dynamic modeling was derived from Newton's laws, and the prototype was constructed using low-cost materials and components: an ESP32 microcontroller, a VL53L0X sensor (for measuring the ball's position), and an HC-SR04 ultrasonic sensor (for defining the desired reference), coupled with a TowerPro MG996R servo motor that regulates the beam's tilt. The project integrates two Human-Machine Interfaces: an embedded interface within the prototype for parameter adjustment and real-time data visualization, and an external interface designed for remote monitoring. The implementation leverages tools such as MATLAB/Simulink (for system simulation and controller design), Arduino IDE (for programming the ESP32 microcontroller), and Visual Studio Code (for developing the external HMI). Performance tests were carried out using step and ramp inputs, evaluating the controllers through integral performance criteria and an analysis of the system's transient response characteristics. The results demonstrate that each technique has specific advantages depending on the analyzed criterion, highlighting the importance of contextual controller selection to optimize system performance.

**Keywords:** ball-and-beam system; dynamic control systems; fuzzy control; fuzzy-PID; human-machine interface; modified PID controller.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo do protótipo com forças agindo sobre a bola.....	18
Figura 2 - Sistema com controle PID.....	21
Figura 3 - Diagrama em blocos do controlador I-PD.....	23
Figura 4 - Diagrama em blocos do controlador <i>Fuzzy</i> .....	25
Figura 5 - Exemplo de função de pertinência.....	25
Figura 6 - Controlador <i>fuzzy</i> -PID de Ação Direta.....	27
Figura 7 - Controlador <i>fuzzy</i> -PID com topologia FGS.....	27
Figura 8 - Controlador <i>fuzzy</i> -PID com topologia Híbrida.....	27
Figura 9 - Servossistema com controle por realimentação de estados.....	28
Figura 10 - Servossistema com controle por realimentação de estados discreto.....	29
Figura 11 - Modelo 3D do protótipo.....	32
Figura 12 - Dimensões do protótipo em milímetros.....	33
Figura 13 - Protótipo do sistema barra-bola.....	34
Figura 14 - Esquema elétrico.....	34
Figura 15 - ESP32.....	35
Figura 16 - <i>Display</i> LCD 16x2 I2C.....	36
Figura 17 - Sensor de distância VL5310x.....	37
Figura 18 - Servomotor MG996R.....	38
Figura 19 - Sensor Ultrassônico HC-SR04.....	39
Figura 20 - <i>Joystick</i> de 3 Eixos.....	40
Figura 21 - Diagrama em blocos do sistema de controle.....	42
Figura 22 - Menu Principal.....	43
Figura 23 - Submenus do Menu PID.....	43
Figura 24 - Submenus dos menus $K_d$ e $\Delta T$ PID.....	44
Figura 25 - Tela do menu com a leitura dos sensores.....	44
Figura 26 - Tela “Conectar”.....	45
Figura 27 - Tela “Controle PID”.....	46
Figura 28 - Tela “Controle I-PD”.....	46
Figura 29 - Tela “Controle <i>fuzzy</i> ”.....	47
Figura 30 - Tela “Regras do controlador <i>fuzzy</i> ”.....	47

Figura 31 - Tela “Parâmetros de erro de posição do controlador <i>fuzzy</i> ” .....	48
Figura 32 - Tela “Controle <i>fuzzy</i> -PID FGS” .....	48
Figura 33 - Tela “Controle <i>fuzzy</i> -PID Híbrido” .....	49
Figura 34 - Tela “Controle por realimentação de estados” .....	49
Figura 35 - Tela “Ajuda - parte 1” .....	50
Figura 36 - Tela “Ajuda - parte 2” .....	51
Figura 37 - Tela “Configurar” .....	51
Figura 38 - Sistema com controlador PID discreto .....	52
Figura 39 - Saída usando PID discreto .....	53
Figura 40 - Comparação entre os resultados ante uma rampa usando controlador PID .....	54
Figura 41 - Controlador I-PD discreto aplicados ao sistema .....	55
Figura 42 - Comparação entre os resultados ante um degrau usando controlador I-PD.....	56
Figura 43 - Comparação entre os resultados ante uma rampa usando controlador I-PD .....	57
Figura 44 - Controlador <i>fuzzy</i> aplicado ao sistema.....	58
Figura 45 - Função de pertinência do erro de posição .....	58
Figura 46 - Função de pertinência da velocidade da bola .....	59
Figura 47 - Função de pertinência do ângulo do servomotor .....	59
Figura 48 - Superfície <i>fuzzy</i> .....	60
Figura 49 - Comparação entre os resultados ante um degrau usando controlador <i>fuzzy</i> .....	61
Figura 50 - Comparação entre os resultados ante uma rampa usando controlador <i>fuzzy</i> .....	62
Figura 51 - Diagrama em blocos do controle <i>Fuzzy</i> -PID FGS .....	63
Figura 52 - Comparação entre os resultados ante um degrau usando controle <i>Fuzzy</i> -PID FGS	63
Figura 53 - Comparação entre os resultados ante uma rampa usando controlador <i>Fuzzy</i> -PID FGS.....	64
Figura 54 - Diagrama em blocos do controle <i>Fuzzy</i> -PID Híbrido.....	65
Figura 55 - Comparação entre os resultados ante um degrau usando controle <i>Fuzzy</i> -PID Hí- brido .....	66
Figura 56 - Comparação entre os resultados ante uma rampa usando controlador <i>Fuzzy</i> -PID Híbrido .....	67
Figura 57 - Diagrama em blocos do controle por realimentação de estados .....	68
Figura 58 - Comparação entre os resultados ante um degrau usando controle por realimen- tação de estados.....	68

Figura 59 - Comparação entre os resultados ante uma rampa usando controle por realimentação de estados .....	69
Figura 60 - Teste complexo com controle PID .....	70
Figura 61 - Ações de controle para o teste com controle PID .....	71
Figura 62 - Teste complexo com controle I-PD .....	71
Figura 63 - Ações de controle para o teste com controle I-PD .....	72
Figura 64 - Teste complexo com controle <i>fuzzy</i> .....	72
Figura 65 - Ações de controle para o teste com controle <i>fuzzy</i> .....	73
Figura 66 - Teste complexo com controle <i>Fuzzy</i> -PID (FGS) .....	73
Figura 67 - Ações de controle para o teste com controle <i>fuzzy</i> -PID (FGS) .....	74
Figura 68 - Teste complexo com controle <i>fuzzy</i> -PID Híbrido .....	74
Figura 69 - Ações de controle para o teste com controle <i>fuzzy</i> -PID Híbrido .....	75
Figura 70 - Teste complexo com controle por realimentação de estados .....	75
Figura 71 - Ações de controle para o teste com controle por realimentação de estados .....	76

## LISTA DE TABELAS

Tabela 1 - Definição dos parâmetros .....	18
Tabela 2 - Medidas do protótipo .....	33
Tabela 3 - Principais características do ESP32-WROOM-32U .....	35
Tabela 4 - Principais características do <i>Display</i> LCD 16x2 I2C .....	36
Tabela 5 - Características do Sensor VL53L0X .....	37
Tabela 6 - Características do Servomotor TowerPro MG996R.....	38
Tabela 7 - Principais características do Sensor Ultrassônico HC-SR04 .....	39
Tabela 8 - Especificações da resposta transitória para uma referência degrau de 20 cm.....	53
Tabela 9 - Critérios integrais para um degrau usando controle PID .....	53
Tabela 10 - Critérios integrais para uma rampa usando controle PID .....	54
Tabela 11 - Especificações da resposta transitória para uma referência degrau de 20 cm usando controle I-PD .....	55
Tabela 12 - Critérios integrais para um degrau usando controle I-PD.....	56
Tabela 13 - Critérios integrais para uma rampa usando controle I-PD .....	57
Tabela 14 - Comparação entre os resultados simulados e práticos para uma referência degrau de 20 cm usando controle <i>fuzzy</i> .....	60
Tabela 15 - Critérios integrais para um degrau usando controle <i>fuzzy</i> .....	61
Tabela 16 - Critérios integrais para uma rampa usando controle <i>fuzzy</i> .....	62
Tabela 17 - Especificações da resposta transitória para uma referência degrau de 20 cm usando controle <i>Fuzzy</i> -PID FGS.....	63
Tabela 18 - Critérios integrais para um degrau usando controle <i>Fuzzy</i> -PID FGS .....	64
Tabela 19 - Critérios integrais para uma rampa usando controle <i>Fuzzy</i> -PID FGS.....	64
Tabela 20 - Especificações da resposta transitória para uma referência degrau de 20 usando controle <i>Fuzzy</i> -PID Híbrido .....	65
Tabela 21 - Critérios integrais para um degrau usando controle <i>Fuzzy</i> PID Híbrido .....	66
Tabela 22 - Critérios integrais para uma rampa usando controle <i>Fuzzy</i> -PID Híbrido .....	67
Tabela 23 - Especificações da resposta transitória para uma referência degrau de 20 cm usando controle por realimentação de estados .....	68
Tabela 24 - Critérios integrais para um degrau usando controle por realimentação de estados .	69
Tabela 25 - Critérios integrais para uma rampa usando controle por realimentação de estados	70

Tabela 26 - Máximo Sobressinal - ensaio simulado com entrada degrau .....	76
Tabela 27 - Critérios integrais - ensaio simulado com entrada degrau .....	77
Tabela 28 - Máximo Sobressinal - ensaio prático com entrada degrau.....	77
Tabela 29 - Critérios integrais - ensaio prático com entrada degrau.....	78
Tabela 30 - Critérios integrais obtidos a partir de simulações .....	79
Tabela 31 - Critérios integrais obtidos a partir de experimentos .....	79

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Apresentação Do Problema .....	13
1.2	Motivação .....	14
1.3	Objetivos .....	15
1.3.1	Objetivo Geral .....	15
1.3.2	Objetivos Específicos.....	15
<b>2</b>	<b>REVISÃO TEÓRICA</b>	<b>16</b>
2.1	Estado da arte .....	16
2.2	Sistema barra-bola.....	17
2.3	Controle PID .....	20
2.4	Controlador I-PD .....	23
2.5	Controlador <i>fuzzy</i> .....	24
2.6	Controlador <i>fuzzy</i> -PID .....	26
2.7	Controle Por Realimentação De Estados .....	28
2.8	Métricas Utilizadas .....	29
<b>3</b>	<b>PROTÓTIPO DO SISTEMA BARRA-BOLA</b>	<b>32</b>
3.1	Ferramentas Computacionais Utilizadas.....	41
3.2	Sistema Controlado .....	41
3.2.1	IHM Integrada.....	43
3.2.2	IHM Externa.....	44
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>52</b>
4.1	Controle PID .....	52
4.2	Controle I-PD .....	54
4.3	Controle <i>fuzzy</i> .....	57
4.4	Controle <i>fuzzy</i> -PID (FGS).....	62
4.5	Controle <i>fuzzy</i> -PID (híbrido) .....	65
4.6	Controle Por Realimentação De Estados .....	67
4.7	Comportamento Do Sistema Em Testes Complexos .....	70
4.8	Comparação Dos Métodos De Controle .....	76

**5 CONCLUSÃO**

**80**

**REFERÊNCIAS**

**81**

## 1 INTRODUÇÃO

Segundo Ogata (2011) sistemas de controle existem desde a revolução industrial (final do século XVIII) sendo o primeiro trabalho significativo de controle a invenção do governador centrífugo por James Watt. Esse era um controlador bastante rudimentar que regulava a vazão de combustível em uma máquina com base na rotação do mesmo. Deste então, os sistemas de controle vêm passando por diversos aperfeiçoamentos, onde os sistemas mecânicos foram substituídos por relés, e posteriormente por transistores até os tempos atuais com circuitos integrados.

Assim, como os sistemas de controle, os métodos de controle também evoluíram com o tempo, iniciando com métodos rudimentares e, com o tempo, evoluindo para formas cada vez mais robustas como: controlador proporcional integral derivativo (PID), Lógica *fuzzy*, Redes Neurais Artificiais, controle Robusto e controle por Algoritmos Genéticos. Com cada um deles tendo seus prós e contras, onde a eficácia no controle depende principalmente do que se deseja controlar e qual a sua complexidade.

Um sistema barra-bola é um sistema de controle de equilíbrio onde uma bola deve permanecer na posição desejada sobre uma barra com guias laterais. O desafio para o sistema de controle é ajustar a inclinação da barra para que a bola fique na posição desejada em um intervalo de tempo satisfatório e com poucas oscilações no entorno desta posição, e também corrigir as mudanças de posição causadas por perturbações externas.

Neste Trabalho de Conclusão de Curso, serão abordadas todas as etapas envolvidas no desenvolvimento do sistema, desde a modelagem matemática até a implementação e validação de diversas estratégias de controle. Com este estudo, espera-se contribuir para o aprofundamento do conhecimento sobre sistemas dinâmicos subatuados e o uso de diversas técnicas de controle, desde as clássicas até as modernas, fornecendo uma base sólida para futuras aplicações e pesquisas na área.

### 1.1 Apresentação Do Problema

O sistema barra-bola é um exemplo clássico de sistema dinâmico subatuado e um dos problemas clássicos na área do controle de sistemas, onde a bola deve ser estabilizada em uma posição desejada sobre uma barra inclinável. A complexidade desse sistema decorre do fato de que a única variável de controle disponível é o ângulo de inclinação da barra, enquanto a posição da bola é a variável de interesse. Isso torna o problema desafiador, pois o sistema como tal é não linear e instável, onde pequenas perturbações deslocam a bola rapidamente, como vibrações ou mudanças bruscas no ângulo da barra.

O principal desafio do controle desse sistema é equilibrar a bola em uma posição desejada através do ajuste do ângulo de inclinação da barra, garantindo uma resposta rápida e precisa. Um sistema de controle eficaz deve ser capaz de reduzir oscilações, minimizar o tempo de estabilização

e lidar com imprecisões e perturbações externas. Além disso, é essencial a robustez do controlador para manter a estabilidade do sistema mesmo diante de variações ou imprecisões nos sensores e atuadores.

## 1.2 Motivação

O controle de sistemas dinâmicos tem papel fundamental em diversas aplicações da engenharia, desde processos industriais até sistemas autônomos. Entre os desafios encontrados, o controle de sistemas subatuados, como o sistema barra-bola, é um problema clássico de modelagem e controle.

Modelar e controlar este tipo de sistema é importante pois simula problemas reais onde pequenos erros levam a grandes desequilíbrios (ex.: equilíbrio de robôs bípedes, drones, ou até foguetes durante pousos). Além disso, os seus princípios são aplicáveis em sistemas de estabilização industrial como esteiras transportadoras, plataformas giratórias, e na robótica, como braços mecânicos e exoesqueletos.

Por outro lado, este sistema permite o projeto de controladores para estabilizar sistemas naturalmente instáveis, e validar algoritmos clássicos e modernos de controle em um ambiente controlado e de baixo custo.

Desde um ponto de vista educacional, combina conceitos de física (dinâmica rotacional), matemática (equações diferenciais não lineares) e engenharia (sistemas embarcados e sensoriamento), onde a sua simplicidade contrasta com sua complexidade dinâmica, estimulando a criatividade para resolver problemas com restrições práticas (limites de torque, ruído em sensores, entre outras).

Finalmente, dominar um sistema do tipo barra-bola não só solidifica fundamentos de controle, mas também prepara profissionais para enfrentar problemas reais de automação e estabilização, tornando-o uma ferramenta pedagógica e técnica indispensável.

### 1.3 Objetivos

#### 1.3.1 Objetivo Geral

Comparar o desempenho das diferentes técnicas de controle aplicadas em um sistema do tipo barra-bola.

#### 1.3.2 Objetivos Específicos

- Adicionar novos métodos de controle ao protótipo de um sistema do tipo barra-bola existente na UACSA.
- Realizar a modelagem da planta usando descrição fenomenológica e/ou via identificação experimental.
- Aprimorar a interface do usuário para uso da planta, permitindo o uso de diferentes estratégias para o controle de posição.
- Realizar simulações do sistema controlado e avaliar o desempenho dos controladores em diferentes cenários.
- Comparar os resultados simulados com experimentais e determinar qual técnica de controle oferece o melhor desempenho no controle de posição.

## 2 REVISÃO TEÓRICA

Neste capítulo são apresentados os fundamentos de um sistema barra-bola, assim como as bases teóricas das estratégias de controle que serão utilizadas para estabilizar a bola numa posição determinada.

### 2.1 Estado da arte

Existe uma quantidade significativa de literaturas que abordam sistemas de controle do tipo barra-bola, variando desde a modelagem matemática do sistema, simulações, até prototipagem, como que Rahmat, Wahid e Wahab (2010) modelou, simulou e comparou o desempenho de três estratégias de controle usando o *MATLAB*, além de criar uma Interface Homem-Máquina (IHM) com o mesmo, para exibir as simulações do sistema e alternar entre estratégias de controle, sendo elas: proporcional (P), PID, Regulador Linear Quadrático (LQR) e RNAs.

Em sua publicação Moezzi, Vu e Tamre (2018) simulam o controle do sistema barra-bola utilizando o *software MATLAB/Simulink*, porém em sua obra as métricas de desempenho dos controladores são analisadas para entradas de diversos formatos com rampas e senóides; quanto aos métodos de controle, foram simuladas duas variantes de controladores *fuzzy*, Mamdani e Sugeno, além de um controlador PID.

Já Camargo (2018) modelou e construiu um protótipo funcional desse tipo de sistema usando a placa de prototipagem Arduino UNO, um servomotor e um sensor de posição *SoftPot* juntamente com um *software* de controle desenvolvido pelo próprio autor usando a IDE *Microsoft Visual Studio* 2013, para que possa ser usado por estudantes da disciplina de controle de sistemas para treinar e aprofundar os conhecimentos sobre controladores PID de forma prática e intuitiva, com o auxílio do *software* que integra um computador ao protótipo.

Usando como base o projeto de Camargo (2018), Prunzel (2019) aprimorou o trabalho nos quesitos físicos como a troca da placa de desenvolvimento por uma com mais memória (Arduino MEGA 2560) além de fazer melhorias na estrutura do protótipo. Quanto aos elementos de *software*, o projeto ganhou mais dois métodos de controle modernos, além do conhecido PID, sendo eles o controle baseado em lógica nebulosa (também conhecida como lógica *fuzzy*) e o controle baseado em Redes Neurais Artificiais (RNAs), além de alterações na interface do *software* usado no computador para que o mesmo oferecesse suporte aos novos métodos de controle adicionados.

No trabalho de Latif, Muhammad e Naeem (2019) é criado um sistema barra-bola para fins educacionais, no projeto utiliza um Arduino Mega que se comunica com o *MATLAB* utilizando uma conexão RS232, o sistema também possui um motor acoplado a uma caixa de engrenagens com um controlador PID para corrigir e controlar o ângulo do motor. Quanto ao controle de posição da bola, foram implementados dois controladores que podem ser escolhidos, sendo o primeiro um

controlador *fuzzy*, e o segundo método um controlador PID clássico.

O projeto de iniciação científica escrito por Azevedo (2020) trata da modelagem matemática de um sistema barra-bola e o uso de um controlador proporcional derivativo (PD) com realimentação por velocidade para ser aplicado em um protótipo.

O trabalho de Alves *et al.* (2020) aborda a modelagem matemática do sistema barra-bola e faz um estudo comparativo entre três formas de sintonizar um controlador PID, sendo a primeira com auxílio computacional usando a ferramenta Sisotool, a segunda usando o método de Ziegler-Nichols e a terceira usando o método empírico-analítico. Todos esses métodos foram testados em um protótipo de sistema barra-bola construído com partes impressas em 3D que estão disponíveis de forma livre na internet.

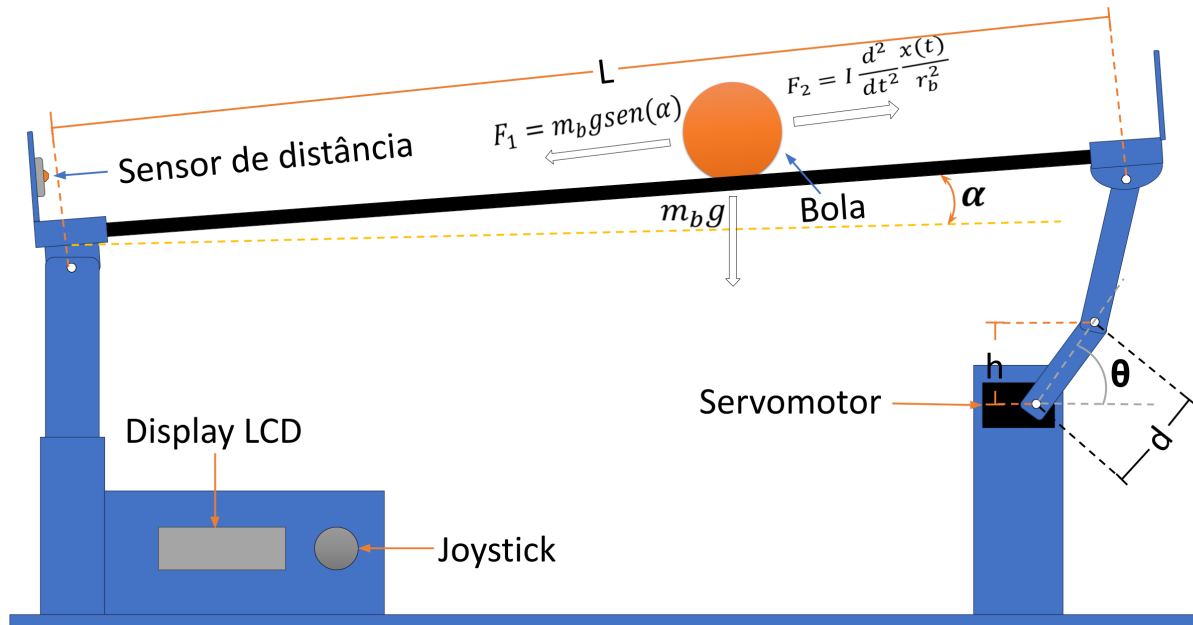
Em sua publicação, Muftah *et al.* (2022) tratam de um sistema de controle utilizando um atuador pneumático não linear para alterar o ângulo da barra, usando dois controladores PID, sendo o primeiro deles um controlador FOPID (PID de ordem fracionária) responsável pelo controle da posição do atuador pneumático e o segundo um controlador PID responsável pelo controle de posição da bola.

O artigo de Saleem *et al.* (2022) utiliza processamento digital de imagens com *Simulink* para interpretar os dados recebidos a partir de uma *webcam* que monitora o sistema. Os dados são processados e convertidos em posição e velocidade para que possam ser usados em um controlador ANFIS (*Adaptive Neuro Fuzzy Inference System*) ou em um controlador PID.

## 2.2 Sistema barra-bola

O sistema barra-bola consiste em uma barra inclinável e uma bola que se move livremente sobre sua superfície, sendo o objetivo estabilizar a bola em uma posição desejada. Neste trabalho, foi adotado o modelo de haste móvel como mostrado na Figura 1 e as definições dos parâmetros e variáveis usadas se mostram na Tabela 1. No modelo mostrado, a inclinação da barra é controlada por um servomotor. Esse arranjo permite maior precisão no controle da posição da bola.

Na dedução do modelo matemático do sistema, são consideradas algumas simplificações, como por exemplo, a bola sempre estará em contato com a barra, sem deslizamento, e sempre girará sobre dois trilhos. Existirá movimento da bola sempre que a barra for inclinada e a resistência do ar será desprezível.

**Figura 1** – Modelo do protótipo com forças agindo sobre a bola

Fonte: elaborado pelo próprio autor.

**Tabela 1** – Definição dos parâmetros

Parâmetro	Definição	Unidade
$m_b$	Massa da bola	[Kg]
$R_b$	Raio da bola	[m]
$\tau_b$	Torque da bola	$[\frac{N}{m}]$
$a$	Aceleração da bola	$[\frac{m}{s^2}]$
$g$	Aceleração da gravidade	$[\frac{m}{s^2}]$
$\alpha$	Ângulo de inclinação da barra em relação ao eixo horizontal	[rad]
$\gamma_b$	Ângulo de rotação da bola	[rad]
$F_1$	Força em x devido à gravidade	[N]
$F_2$	Força de inércia de rotação	[N]
$I$	Momento de inércia da bola	$[kg.m^2]$
$h$	Altura da haste do servomotor em relação ao ponto de fixação	[m]
$L$	Comprimento da barra	[m]
$\theta$	Ângulo de inclinação da haste do servomotor	[rad]
$d$	Comprimento da haste do servomotor	[m]

Fonte: Elaborado pelo próprio autor.

Utilizado como base a segunda lei de Newton para a bola:

$$\sum F = m_b a \quad (1)$$

A aceleração,  $a$ , pode ser representada em função da posição,  $x$ , assim:

$$\sum F = m_b \frac{d^2x}{dt^2} = F_1 - F_2 \quad (2)$$

A força  $F_1$  é a componente horizontal da força gravitacional que atua sobre a bola, como mostrado na Equação (3).

$$F_1 = m_b g \text{sen}(\alpha) \quad (3)$$

A segunda força que atua sobre a bola acontece devido à inércia rotacional, representada pela Equação (4).

$$F_2 = \frac{\tau_b}{r_b} \quad (4)$$

Onde o torque na bola,  $\tau_b$ , é o produto do momento de inércia  $I$  pela aceleração angular da bola  $\alpha_a$ , assim:

$$\tau_b = I\alpha_a = I \frac{d^2}{dt^2} \gamma_b \quad (5)$$

Por outro lado, a posição da bola ao longo da barra é dada por:

$$x = r_b \gamma_b \quad (6)$$

Isolando  $\gamma_b$  da Equação (6) e substituindo na Equação (5):

$$\tau_b = \frac{I}{r_b} \frac{d^2x}{dt^2} \quad (7)$$

Por fim, substituindo a Equação (7) na (4), chega-se à equação da força de inércia de rotação:

$$F_2 = \frac{I}{r_b^2} \frac{d^2x}{dt^2} \quad (8)$$

Desta forma, a Equação (2) pode ser representada como:

$$\frac{d^2x}{dt^2} = \frac{m_b g \text{sen}(\alpha) r_b^2}{I + m_b r_b^2} \quad (9)$$

Analisando novamente a Figura 1, chega-se à relação:

$$h = \text{sen}(\alpha)L = \text{sen}(\theta)d \quad (10)$$

Assumindo uma restrição de ângulo de 0,698 radianos (40 graus) no braço de servomotor para

evitar que seu parafuso de fixação colida com a base onde o servomotor é fixado, pode-se usar a aproximação  $\text{sen}(\theta) = \theta$ , dessa forma a Equação (10) pode ser reescrita como:

$$\text{sen}(\alpha) = \frac{\text{sen}(\theta)d}{L} = \frac{\theta d}{L} \quad (11)$$

Finalmente, substituindo a Equação (11) na (9),

$$\frac{d^2x}{dt^2} = \frac{m_b g d r_b^2 \theta}{(I + m_b r_b^2)L} \quad (12)$$

Fazendo  $C = \frac{m_b g d r_b^2}{(I + m_b r_b^2)L}$ , a Equação (12) pode ser reescrita como:

$$\frac{d^2x(t)}{dt^2} = C\theta(t) \quad (13)$$

Finalmente, aplicando a transformada de Laplace considerando as condições iniciais nulas na Equação (13), a dinâmica do sistema barra-bola pode ser dada pela seguinte função de transferência:

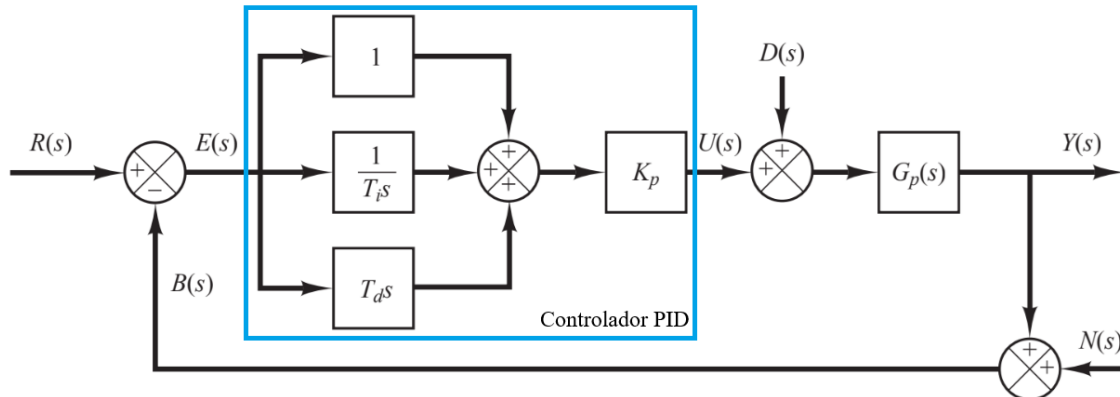
$$\frac{X(s)}{\theta(s)} = \frac{C}{s^2} \quad (14)$$

Note-se que a função de transferência dada na Equação (14) possui um integrador duplo, caracterizando um sistema marginalmente estável. Ou seja, na ausência de controle externo, o sistema tenderá a divergir (crescer indefinidamente) quando sujeito a uma entrada constante, caracterizando um problema de controle desafiador.

### 2.3 Controle PID

Segundo Ogata (2011) mais da metade dos controladores usados na indústria são PID ou alguma versão modificada de PID; então é fundamental tratar esse tipo de controlador tão versátil e que pode ser utilizado em praticamente qualquer sistema, que seja possível ou não obter a sua função de transferência.

**Controlador PID analógico:** Surgindo no início do século XX, o PID e suas variantes ficaram populares por sua robustez e versatilidade, podendo ser usados para solucionar diversos problemas. Esses controladores operam em tempo contínuo e podem ser sintonizados alterando suas resistências ou capacitâncias de modo que alterem os parâmetros  $K_p$ ,  $K_i = \frac{K_p}{T_i}$  e  $K_d = K_p T_D$ , constantes proporcional, integral e derivativa, respectivamente. O sistema com controlador PID é mostrado na Figura 2.

**Figura 2** – Sistema com controle PID

Fonte: Ogata, 2011.

Esse controlador adiciona zeros e polos no sistema para que o mesmo tenha uma resposta satisfatória, e sua função de transferência é dada pela Equação (15).

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{T_i}{s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s \quad (15)$$

**Controlador PID digital:** Com o avanço tecnológico dos microcontroladores passou a ser possível implementar controladores digitais que ficaram populares devido à sua adaptabilidade e praticidade, pois para alterar um parâmetro é necessário alterar somente uma variável no código (muitas vezes feita de forma remota) enquanto nos controladores analógicos precisavam que os componentes fossem substituídos fisicamente ou reajustados manualmente. Diferente de seu antecessor analógico que trabalha em tempo contínuo, o controlador PID digital opera em tempo discreto, dessa forma a função de transferência analógica deve ser discretizada para que possa ser implementada em um controlador digital. Usando a transformação Bilinear (Equação (16)) para a parte integral, chega-se a seu equivalente em tempo discreto, mostrado na Equação (17).

$$H_a(s) = H(z) \Big|_{s=\frac{2}{\Delta T} \frac{z-1}{z+1}} \quad (16)$$

onde  $\Delta T$  é o intervalo de amostragem do sinal contínuo para a obtenção do sinal discreto, e  $z$  é a variável no domínio  $Z$  (discreto).

$$H(z) = K_i \frac{\Delta T (z+1)}{2(z-1)} \quad (17)$$

A parte derivativa pode ser aproximada por:

$$\frac{dx(t)}{dt} \approx \frac{x[n] - x[n-1]}{\Delta T} \quad (18)$$

sendo  $x[n]$  e  $x[n-1]$ , as amostras atual e anterior da posição, respectivamente.

Usando a transformada Z na Equação (18).

$$sx(z) = \frac{x(z) - x(z)z^{-1}}{\Delta T} \quad (19)$$

Reorganizando a Equação (19), chega-se à Equação (20), a aproximação de  $s$  para a parte derivativa.

$$s = \frac{1 - z^{-1}}{\Delta T} \quad (20)$$

Substituindo as Equações 17 e 20 na Equação (15), se obtém a Equação (21), a função de transferência discreta do PID.

$$\frac{U(z)}{E(z)} = K_p + K_i \frac{\Delta T(z+1)}{2(z-1)} + \frac{K_d(1-z^{-1})}{\Delta T} \quad (21)$$

Fazendo algumas manipulações matemáticas na Equação (21), chega-se a:

$$U(z)(z-1) = E(z)[K_p(z-1) + \frac{K_i \Delta T}{2}(z+1) + \frac{K_d}{\Delta T}(z-2+z^{-1})] \quad (22)$$

Para que o sistema seja causal, o maior expoente de  $z$  deve ser zero. Portanto, a Equação (22) deve ser multiplicada por  $z^{-1}$  para satisfazer essa condição, assim:

$$U(z)(1-z^{-1}) = E(z)(K_p(1-z^{-1}) + \frac{K_i \Delta T}{2}(1+z^{-1}) + \frac{K_d}{\Delta T}(1-2z^{-1}+z^{-2})) \quad (23)$$

Para que o controlador PID possa finalmente ser implementado em um sistema digital é necessário aplicar a transformada inversa e realizar algumas manipulações matemáticas na Equação (23) para convertê-la em uma equação de diferenças, assim:

$$u(n) = u(n-1) + (K_p + \frac{K_i \Delta T}{2} + \frac{K_d}{\Delta T})e(n) + (-K_p + \frac{K_i \Delta T}{2} - 2\frac{K_d}{\Delta T})e(n-1) + \frac{K_d}{\Delta T}e(n-2) \quad (24)$$

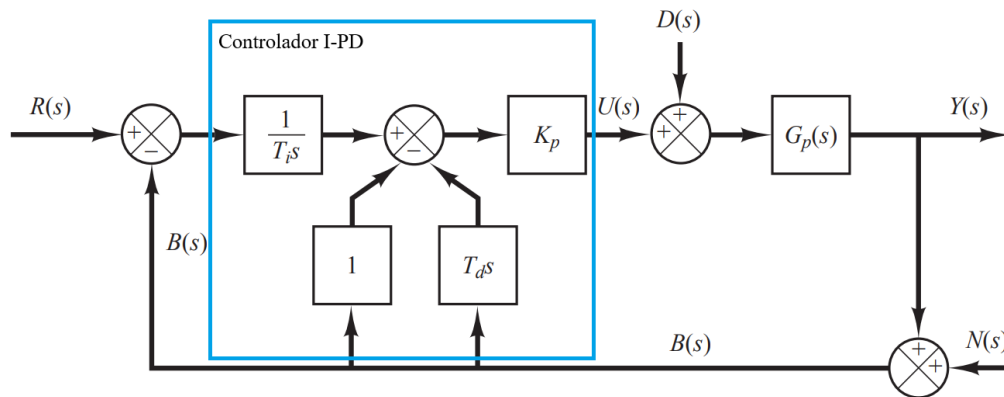
sendo  $u(n)$  e  $u(n-1)$  a saída atual e anterior do controlador, assim como também  $e(n)$ ,  $e(n-1)$  e  $e(n-2)$ , o erro atual, na amostra anterior e duas amostras atrás do sistema, respectivamente.

Desta forma a Equação (24) representa a ação de controle PID a ser embarcada no protótipo.

## 2.4 Controlador I-PD

Segundo OGATA (2011), o controlador I-PD (Figura 3) é uma versão modificada do já conhecido controlador PID, no qual as ações, derivativa e proporcional, foram removidas do ramo de avanço e colocadas no ramo de realimentação. Desta forma, as parcelas derivativa e proporcional agem com base no sinal de saída e não com base no sinal de erro. Isto é feito para eliminar o efeito proporcional no atuador, pois com um degrau na entrada de referência, haverá um degrau no atuador, o que em algumas vezes pode ser prejudicial ao atuador.

**Figura 3** – Diagrama em blocos do controlador I-PD



Fonte: Ogata, 2011.

**Controlador I-PD analógico:** Por ser uma versão modificada do controlador PID, a implementação analógica é bem semelhante, possuindo os mesmos parâmetros  $K_p$ ,  $K_i$  e  $K_d$ . A função de transferência do controlador é dada pela Equação (25).

$$U(s) = K_p \frac{1}{T_i s} R(s) - K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) B(s) \quad (25)$$

sendo  $R(s)$  o sinal de referência e  $B(s)$  o sinal de saída do sistema  $Y(s)$  somado a um ruído  $N(s)$  na saída.

**Controlador I-PD digital:** Para inserir o controlador em um sistema digital, é necessário discretizar a sua função de transferência. Nesse caso, foi utilizada a forma mais genérica possível da função de transferência para facilitar a modificação dos parâmetros do controlador durante o seu funcionamento sem a necessidade de parar o sistema para enviar um novo *firmware* ao controlador.

Substituindo as partes integral e derivativa na Equação (25) por suas equivalentes em tempo discreto, chega-se à Equação (26).

$$U(z) = K_p \frac{\Delta T}{T_i(z-1)} R(z) - K_p \left[ 1 + \frac{\Delta T}{T_i(z-1)} + \frac{T_d(z-1)}{(\Delta T)z} \right] B(z) \quad (26)$$

Sabendo que o sinal de erro é representado pela Equação (27).

$$E(z) = R(z) - B(z) \quad (27)$$

Dessa forma a Equação (26) pode ser expressada em função do erro  $E$ :

$$U(z) = K_p \frac{\Delta T}{T_i(z-1)} E(z) - K_p \left( 1 + \frac{T_d(z-1)}{(\Delta T)z} \right) B(z) \quad (28)$$

Para uma implementação da Equação (28) em um sistema digital, a mesma deve ser escrita no formato de Equação de diferenças, chegando assim à Equação (29).

$$u(n) = u(n-1) + K_i \Delta T e(n) + K_p [B(n) - B(n-1)] + \frac{K_d}{\Delta T} [B(n) - 2B(n-1) + B(n-2)] \quad (29)$$

sendo  $B(n)$ ,  $B(n-1)$  e  $B(n-2)$  o sinal de saída na amostra atual, uma amostra atrasada e duas amostras atrasadas, respectivamente.

## 2.5 Controlador *fuzzy*

Segundo Marro *et al.* (2010), a lógica *fuzzy*, também conhecida como lógica nebulosa ou difusa, foi proposta em 1965 no artigo *Fuzzy Sets* publicado por Lotfi Zadeh como uma intenção de ajudar a classificar coisas no mundo físico que não podem ser normalmente classificadas com precisão.

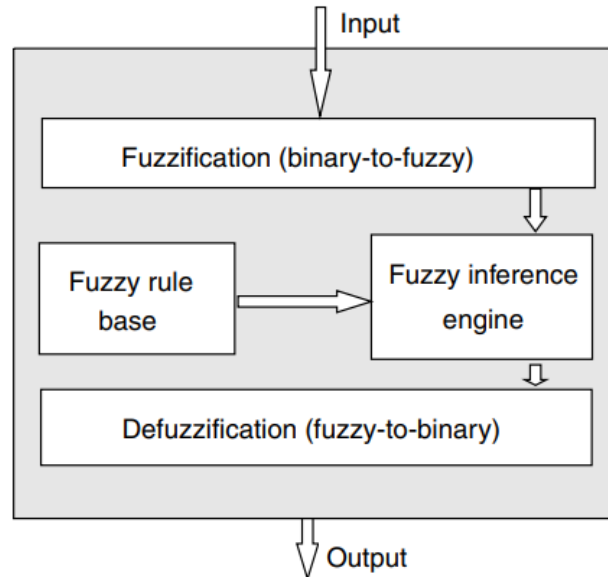
Um exemplo da aplicação da lógica *fuzzy* é determinar se um copo está cheio; a partir de uma abordagem utilizando lógica booleana seria possível somente obter duas informações sobre o copo, ou ele está cheio ou não está cheio. Usando a lógica *fuzzy* pode-se obter muito mais informações sobre a quantidade de líquido no copo, por exemplo com a lógica booleana não era possível saber se o copo está vazio porque a única informação disponível é se o mesmo está cheio, logo qualquer informação que não é essa é descartada. Voltando à lógica *fuzzy*, o nível de líquido no copo pode assumir qualquer valor real entre 0 e 1, que é obtido a partir de uma função de pertinência, usada para calcular o grau de pertinência (certeza) que o copo está cheio, simulando o critério de classificação humano.

Como mostrado na Figura 4, um sistema baseado em lógica *fuzzy* possui quatro etapas principais:

**1 - Fuzzificação:** Nessa etapa, as variáveis de entrada são convertidas em variáveis linguísticas, onde é determinado o grau de pertinência de cada entrada usando uma função de pertinência, como mostrado na Figura 5. Essas funções podem ser triangulares, trapezoidais, gaussianas ou sigmóides,

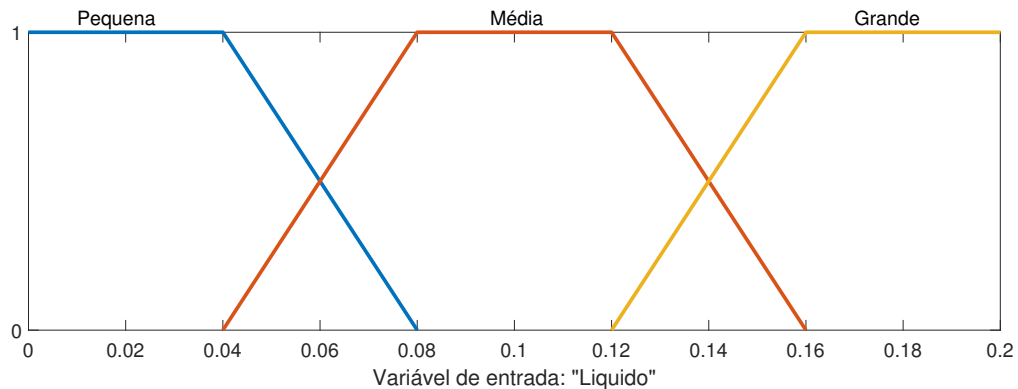
dependendo da aplicação. No exemplo do copo, a função de pertinência quantidade de líquido é trapezoidal e pode ser classificada como pequena, média ou grande.

**Figura 4** – Diagrama em blocos do controlador *Fuzzy*



**Fonte:** Kovacic e Bogdan, 2010,

**Figura 5** – Exemplo de função de pertinência



**Fonte:** elaborado pelo próprio autor.

**2 - Mecanismo de Inferência Fuzzy:** é o “cérebro” do controlador *Fuzzy*. Esta etapa é o momento em que serão analisadas as regras ou proposições através da associação entre as entradas após a fuzzificação e aplicá-las no conjunto de regras usando operadores “e” ou “ou” para obter o valor do conseqüente para cada regra. Exemplo: Copo com  $\frac{3}{4}$  da capacidade: se a quantidade de líquido é grande, então a chance de o copo estar cheio é alta.

**3 - Agregação:** é uma etapa específica dentro do Mecanismo de Inferência Fuzzy, nela são agregadas todas as funções dos conseqüentes de cada regra em um único conjunto *fuzzy*. Exemplo:

Considerando o copo com  $\frac{3}{4}$  da capacidade, tem as seguintes regras: se a quantidade de líquido é média, então a chance do copo estar cheio é moderada. Se a quantidade de líquido é grande, então a chance do copo estar cheio é alta. Ambas as regras contribuem para o resultado final, formando uma saída fuzzy composta pelos diferentes graus de pertencimento ao conjunto “cheio”.

**4 - Defuzzificação:** é a quarta e última etapa e é onde os resultados obtidos são convertidos em um valor real entre zero e um. Exemplo: Se o resultado da agregação indica que a chance do copo estar cheio é 60% média e 80% alta, a defuzzificação pode resultar em um valor entre 0 e 1, como 0,75, indicando que o copo está 75% cheio. Com isso, o sistema pode tomar uma decisão precisa, como reduzir o fluxo de água para evitar o transbordamento. Os métodos mais comuns de defuzzificação são: centróide, média dos máximos, critério máximo (ou mínimo) e média ponderada dos máximos.

## 2.6 Controlador *fuzzy*-PID

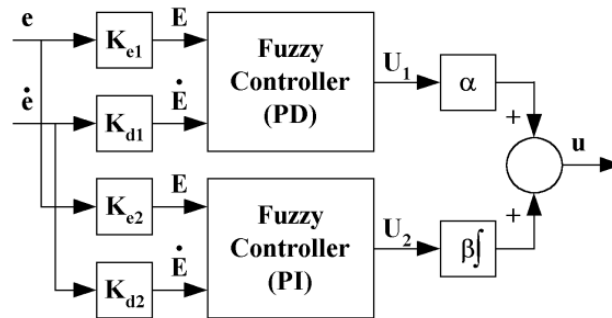
O controlador *Fuzzy*-PID, como seu nome sugere, combina as propriedades do clássico controlador PID com a flexibilidade do controle *fuzzy*. Essa junção permite lidar com sistemas lineares e não lineares de maneira mais eficaz, adicionando novas estratégias e a capacidade de adaptação que podem ser usadas para superar algumas limitações dos controladores PID tradicionais.

Existem diferentes formas de implementar um controlador *Fuzzy*-PID, sendo as mais conhecidas:

**1 - Controladores *Fuzzy*-PID de Ação Direta:** podem ser de uma, duas ou até três entradas. Para a variante de uma entrada, a sintonia do controlador depende apenas do erro do sistema. Essa é uma escolha simples e eficaz para sistemas de menor complexidade. Já a variante de duas entradas (Ver Figura 6), além do erro, considera também a derivada do erro (ou taxa de variação do erro). Isso permite melhorar a resposta transitória, reduzindo oscilações e acelerando o tempo de estabilização. E por fim, a variante com três entradas que considera o erro, a derivada do erro e a integral do erro. Essa abordagem fornece maior controle sobre o comportamento do sistema, mas tem como desvantagem o aumento da complexidade na definição das regras *fuzzy* e no ajuste de parâmetros.

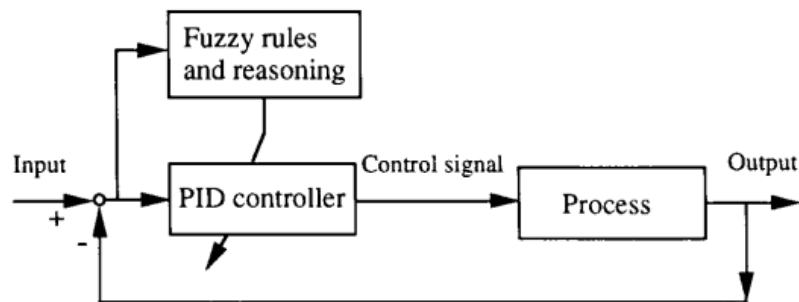
**2 - *Fuzzy Gain Scheduling* (FGS):** nesta topologia, um sistema *fuzzy* é utilizado para ajustar dinamicamente os ganhos do controlador PID em tempo real, como mostrado na Figura 7. Essa abordagem é especialmente útil em sistemas sujeitos a mudanças de regime de operação ou não linearidades, garantindo transições suaves e desempenho consistente em diferentes condições operacionais.

**Figura 6** – Controlador *fuzzy*-PID de Ação Direta



Fonte: Yesil *et al.*, 2003.

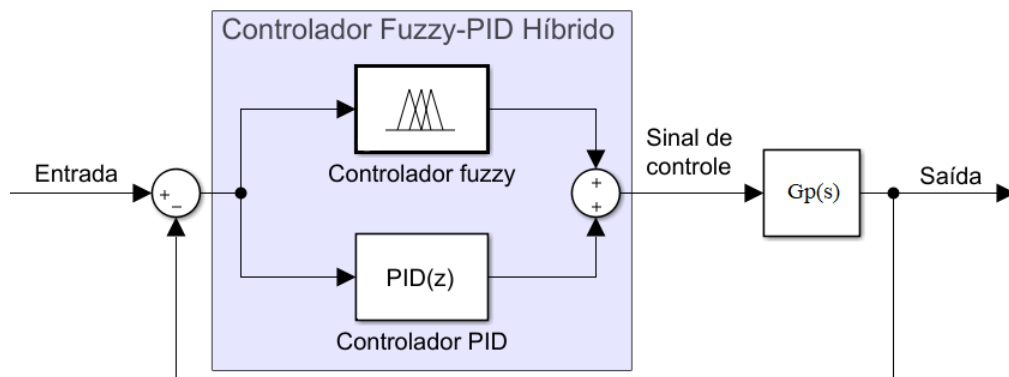
**Figura 7** – Controlador *fuzzy*-PID com topologia FGS



Fonte: Yesil *et al.*, 2003.

**3 - Controladores *Fuzzy*-PID Híbridos:** esses controladores combinam um PID convencional com um controlador *fuzzy*. Essa estrutura híbrida pode ser implementada com alternância entre os dois controladores ou com a fusão de seus sinais de controle (Ver Figura 8). É uma escolha interessante para sistemas onde se deseja aliar o desempenho robusto do PID tradicional com as características do controle *fuzzy*.

**Figura 8** – Controlador *fuzzy*-PID com topologia Híbrida

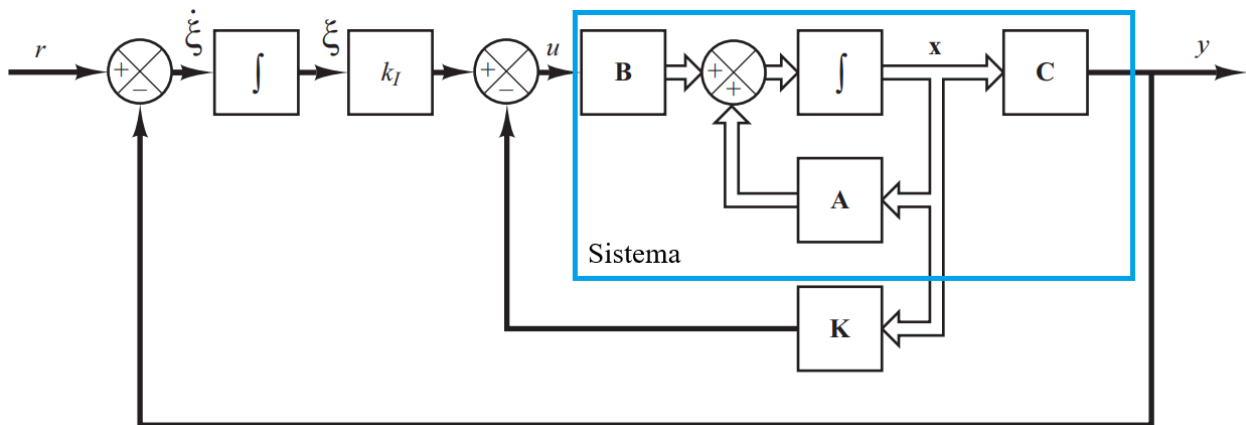


Fonte: elaborado pelo próprio autor.

## 2.7 Controle Por Realimentação De Estados

Um servossistema com controle por realimentação de estados (Figura 9) é um sistema de controle avançado que utiliza informações do estado do sistema para determinar as ações de controle, visando um controle preciso e robusto. Ao contrário dos controladores PID tradicionais, que se baseiam principalmente no erro, este sistema considera todas as variáveis de estado para otimizar o desempenho.

**Figura 9** – Servossistema com controle por realimentação de estados



Fonte: Ogata, 2011.

Como mostrado na Figura 9, o sistema dinâmico é representado por um modelo no espaço de estados, que descreve sua dinâmica com duas equações principais:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (30)$$

$$y = \mathbf{C}\mathbf{x} \quad (31)$$

sendo  $\mathbf{x}$  o vetor de estados,  $u$  o sinal de controle (escalar),  $y$  a saída (escalar) e  $\mathbf{A}$ ,  $\mathbf{B}$  e  $\mathbf{C}$  as matrizes que definem as relações entre estados, entradas e saídas, respectivamente.

A ação de controle usada na realimentação de estados é calculada usando a Equação (32)

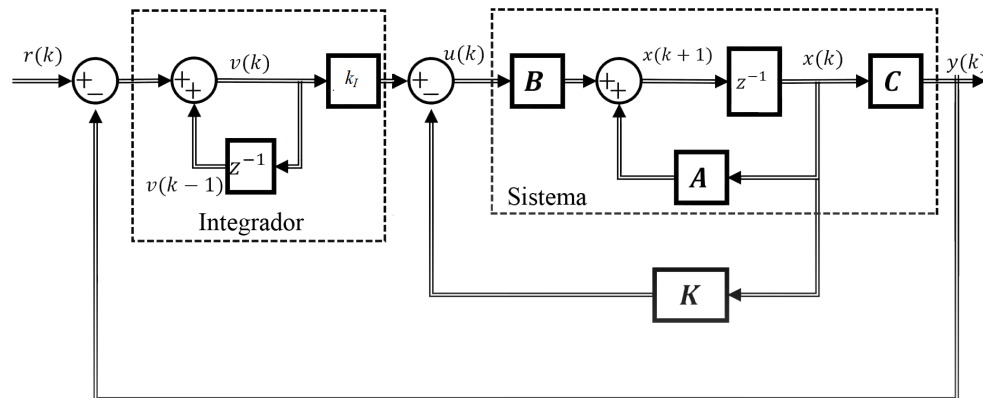
$$u = -\mathbf{K}\mathbf{x} + k_I\xi \quad (32)$$

sendo  $\mathbf{K}$  a matriz de estados do sistema,  $k_I$  o ganho integral e  $\xi$  o estado obtido através da integral do erro.

O projeto do controlador envolve a determinação da matriz  $\mathbf{K}$  e do ganho  $k_I$ , de forma que os polos do sistema em malha fechada estejam localizados em posições desejadas para garantir o desempenho desejado.

Para um sistema discreto, o controlador pode ser representado como na Figura 10.

**Figura 10** – Servossistema com controle por realimentação de estados discreto



**Fonte:** Control Automático Educación, 2025.

O sistema agora é representado por duas novas equações principais:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (33)$$

$$y(k) = \mathbf{C}\mathbf{x}(k) \quad (34)$$

E o sinal de controle é dado por:

$$u(k) = -\mathbf{K}\mathbf{x}(k) + k_I v(k) \quad (35)$$

onde  $v(k)$  é o ganho do integrador.

## 2.8 Métricas Utilizadas

O desempenho de um sistema de controle diz da habilidade que tem o controlador em lidar com as divergências entre a variável controlada e o valor desejado. Tal divergência pode ser quantificada por um número, chamado de índice de desempenho. A literatura apresenta os índices de desempenho baseados na integral do erro como uns dos mais utilizados para avaliar o desempenho de um controlador. Assim, um controlador é considerado como ótimo quando seus parâmetros são ajustados para que o índice em questão alcance um valor mínimo ou máximo. Desta forma, quando maior é o índice baseado na integral do erro, pior será o desempenho do controlador.

A ideia de utilizar critérios baseados na integral do erro surgiu com Ziegler e Nichols em 1942 no artigo “*Optimum Settings for Automatic Controllers*”, onde os autores propuseram métricas para ajustar controladores PID de forma empírica. Posteriormente, Graham Dunstan e Lathrop (1953), no trabalho “*The Synthesis of Optimum Response: Criteria and Standard Form*”, expandiu essa abordagem ao introduzir o critério ITAE (Integral do produto do tempo pelo valor absoluto do erro

entre o SP e a PV em um horizonte de tempo/análise), que penaliza erros de longo prazo para melhorar a resposta transitória dos sistemas.

O Índice de desempenho Integral do Erro ao Quadrado (ISE) acumula o quadrado do erro do sistema (saída - referência) ao longo do tempo. O mesmo possui uma sensibilidade maior aos erros grandes do que aos pequenos, pois é importante ressaltar que sistemas de controle especificados para minimizar o índice ISE tendem a eliminar os grandes erros rapidamente; no entanto, constata-se que pequenos erros persistem na resposta do sistema por um tempo maior, oscilações longas e de pequena amplitude na mesma. Ideal para quando é necessário penalizar grandes erros e é dado pela Equação (36) :

$$ISE = \int_0^T (e(t))^2 dt \quad (36)$$

Semelhante ao ISE, o Índice de Desempenho Integral do Erro ao Quadrado vezes o Tempo (ITSE), como mostrado na Equação (37), acumula o quadrado do erro ao longo do tempo, mas com o valor do erro ponderado pelo tempo. É ideal para quando é necessário penalizar grandes erros, mas com um menor peso durante o início do controle.

$$ITSE = \int_0^T t(e(t))^2 dt \quad (37)$$

Já o critério Integral do Erro Absoluto (IAE), mostrado pela Equação (38), é calculado pela integral do módulo do erro ao longo do tempo, sendo ideal para analisar sistemas com amortecimento mediano. Espera-se que o sistema de controle apresente menos oscilações na saída do que o ISE.

$$IAE = \int_0^T |e(t)| dt \quad (38)$$

Semelhante ao IAE, o índice (ITAE), mostrado pela Equação (39), acumula o módulo do erro ao longo do tempo, com o valor do módulo do erro ponderado pelo tempo sendo ideal para analisar sistemas com amortecimento mediano, mas com menor peso para os instantes iniciais.

$$ITAE = \int_0^T t|e(t)| dt \quad (39)$$

Como todos os índices dependem da soma do erro ao longo do tempo, menores índices indicam um melhor desempenho para um determinado sistema de controle, mas não garantem que condições específicas do projeto sejam satisfeitas.

Além dos índices de desempenho baseados na integral do erro, as especificações da resposta transitória do sistema, conforme definido por Ogata (2011), incluem métricas fundamentais como:

Tempo de subida ( $t_r$ ): tempo necessário para a resposta alcançar pela primeira vez um valor próximo ao estado estacionário;

Tempo de pico ( $t_p$ ): instante em que a resposta atinge o primeiro pico de sobressinal;

Máximo sobressinal ( $M_p$ ): porcentagem do valor máximo ultrapassado em relação ao valor final;

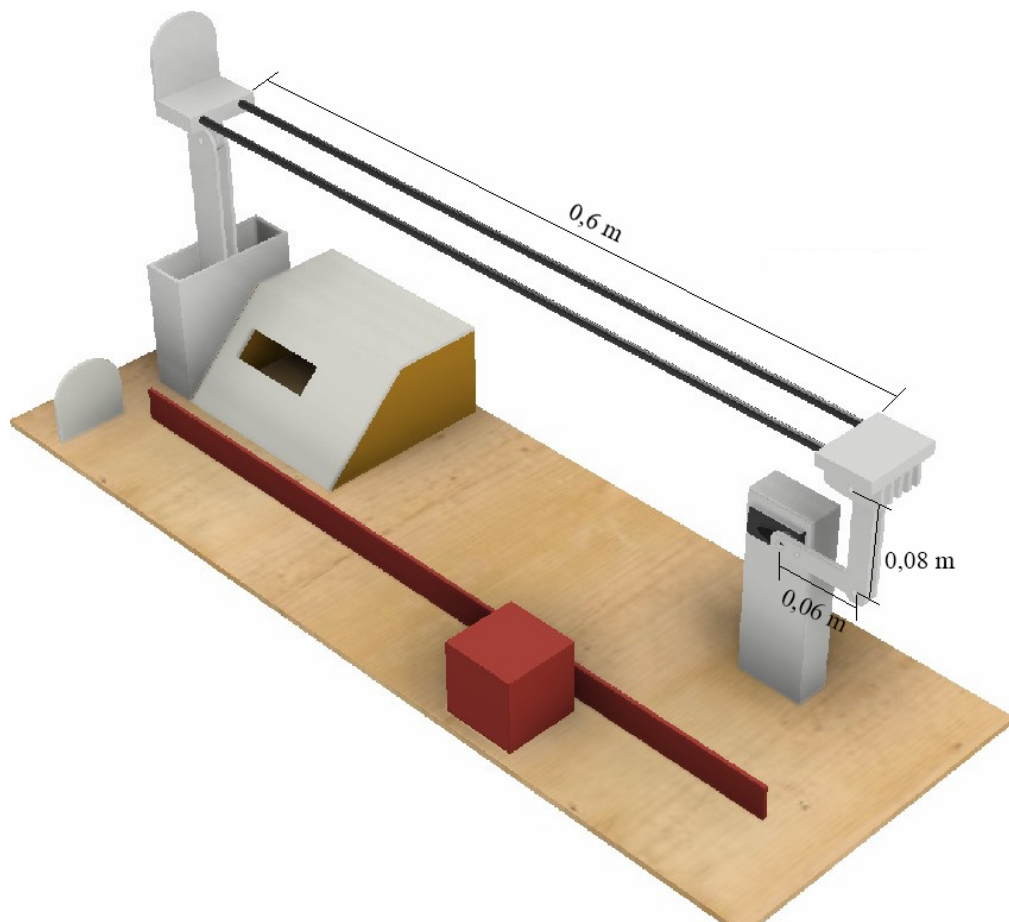
Tempo de acomodação ( $t_s$ ): tempo necessário para a resposta permanecer dentro de uma faixa específica (geralmente de 2% ou 5%) em torno do valor final.

Essas métricas permitem uma avaliação mais abrangente do desempenho do sistema, complementando os índices baseados na integral do erro.

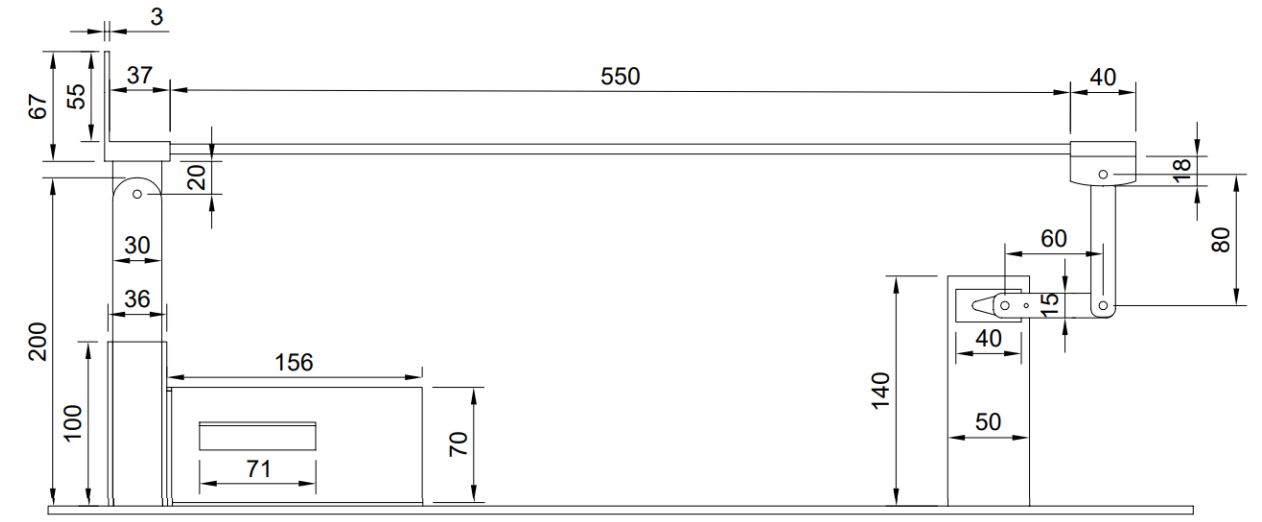
### 3 PROTÓTIPO DO SISTEMA BARRA-BOLA

Antes de iniciar a construção do protótipo, é fundamental desenvolver um modelo 3D do sistema barra-bola, para poupar esforços e retrabalhos desnecessários, além de ajudar de forma significativa o processo de desenho e dimensionamento dos componentes mecânicos. Os desenhos 2D e 3D do protótipo foram inteiramente criados usando as ferramentas do software de modelagem 3D “Autodesk Fusion 360”. A Figura 11 mostra em visão isométrica, do modelo 3D do protótipo, enquanto a Figura 12 destaca suas medidas e por último a Tabela 2 mostra suas principais medidas.

**Figura 11** – Modelo 3D do protótipo



**Fonte:** elaborado pelo próprio autor.

**Figura 12** – Dimensões do protótipo em milímetros

**Fonte:** elaborado pelo próprio autor.

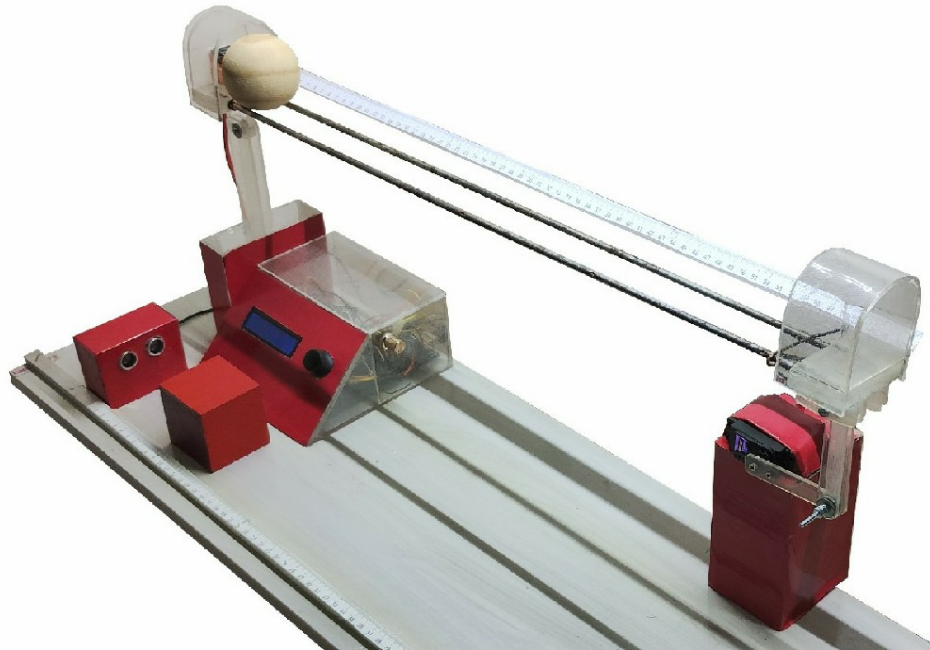
**Tabela 2** – Medidas do protótipo

<b>Componente</b>	<b>Unidade</b>
Barra de apoio	0,6 m
Haste fixa ao servomotor	0,06 m
Haste fixa a barra	0,08 m
Raio da bola	0,031 m
Peso da bola	0,044 Kg

Fonte: Elaborado pelo próprio autor.

O modelo final do protótipo foi construído com chapas de material acrílico, hastes de metal e sua base é feita de MDF. A Figura 13 mostra a versão final do protótipo.

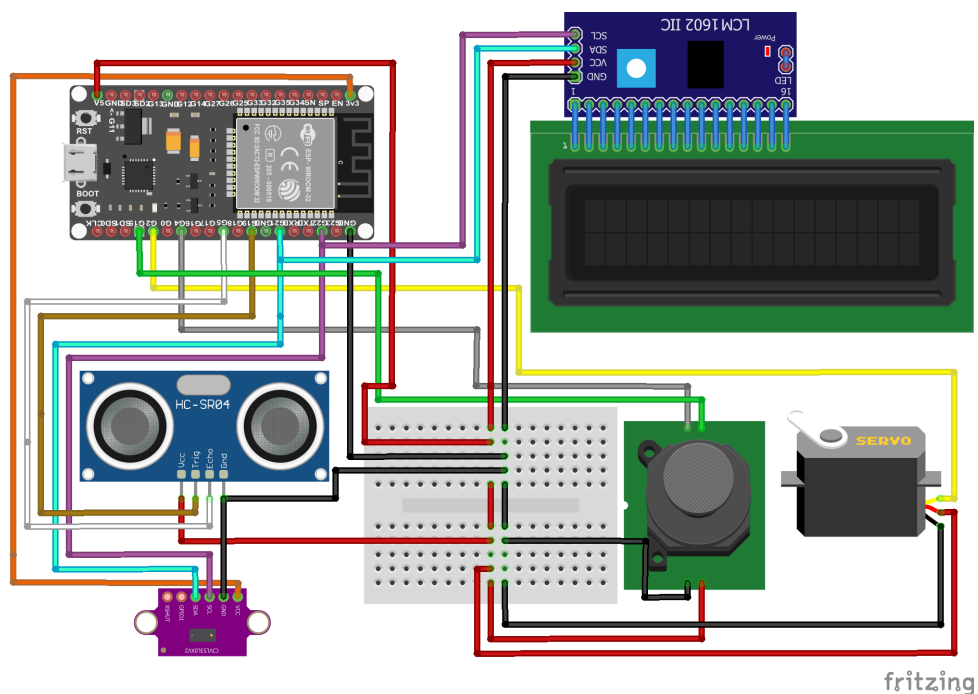
**Figura 13** – Protótipo do sistema barra-bola



**Fonte:** elaborado pelo próprio autor.

O diagrama esquemático para a ligação dos componentes elétricos é mostrado na Figura 14, e é formado pelos seguintes componentes:

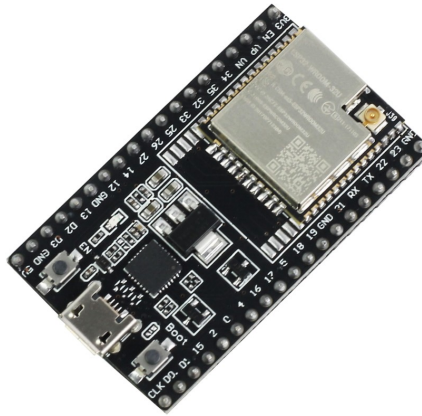
**Figura 14** – Esquema elétrico



**Fonte:** elaborado pelo próprio autor.

**ESP32:** como mostrado na Figura 15, é um microcontrolador de alto desempenho, desenvolvido pela *Espressif Systems*. É amplamente utilizado em aplicações de Internet das Coisas (IoT) e sistemas embarcados que exigem conectividade sem fio, pois integra Wi-Fi em seu chip. Sua arquitetura dual-core, associada a recursos como alta velocidade de clock e múltiplas interfaces de comunicação, torna-o uma excelente escolha para projetos mais avançados.

**Figura 15** – ESP32



**Fonte:** Usinainfo, 2025.

O ESP32 se destaca por sua versatilidade, com um grande número de pinos digitais e analógicos, suportando entradas e saídas de alta velocidade. Além disso, o microcontrolador possui diversas interfaces de comunicação, como SPI, I2C, UART, e CAN, o que o torna ideal para uma ampla gama de aplicações. A Tabela 3 apresenta as principais características técnicas do ESP32.

**Tabela 3** – Principais características do ESP32-WROOM-32U

<b>Característica</b>	<b>Descrição</b>
Microcontrolador	ESP32-D0WD
Tensão de Operação	3.3V
Pinos Digitais	34
Pinos Analógicos	18
Corrente por Pino I/O	Até 12mA
Memória Flash	4MB
SRAM	520KB
Velocidade de Clock	Até 240MHz
Conectividade	Wi-Fi 802.11 b/g/n
Dimensões	55mm x 28mm
Peso	32g

**Fonte:** Espressif, 2025.

**Display LCD 16x2 I<sup>2</sup>C:** mostrado na Figura 16, é uma opção popular para projetos que exigem uma interface de usuário simples e legível. Este tipo de *display* utiliza uma interface I<sup>2</sup>C para se comunicar com o Arduino, o que simplifica a conexão e reduz a quantidade de fios necessários.

**Figura 16** – *Display* LCD 16x2 I<sup>2</sup>C



**Fonte:** Usinainfo, 2024.

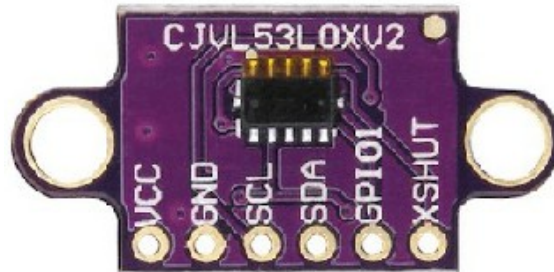
Este *display* é capaz de exibir até 16 caracteres em cada uma de suas linhas. Ele possui um controlador embutido que simplifica a interface com o ESP32 e permite ser controlado com apenas algumas linhas de código. A Tabela 4 apresenta algumas das principais características deste *display*.

**Tabela 4** – Principais características do *Display* LCD 16x2 I<sup>2</sup>C

Característica	Descrição
Interface	I <sup>2</sup> C
Tamanho do <i>Display</i>	16x2
Tensão de Operação	5V
Corrente de Operação	<2mA
Corrente do LED (backlight)	75mA a 200mA
Contraste Ajustável	Sim
Retroiluminação LED	Sim
Dimensões	80 x 36 x 12mm

**Fonte:** Handsontec, 2024.

**Sensor de distância VL5310x:** é um sensor de distância a laser com comunicação I<sup>2</sup>C mostrado na Figura 17, no projeto ele pode ser visto na malha de realimentação do sistema; em outras palavras, é o sensor responsável por monitorar a saída do sistema e converter para o formato digital. O tempo entre a leitura e a conversão do sinal é conhecido como tempo de amostragem e é de extrema importância para sistemas digitais, porque uma taxa de amostragem muito alta pode fazer com que um sistema estável em tempo contínuo torne-se instável em tempo discreto. A Tabela 5 mostra as características do sensor.

**Figura 17** – Sensor de distância VL53L0X

Fonte: ESPHome, 2024.

**Tabela 5** – Características do Sensor VL53L0X

Característica	Descrição
Tipo de sensor	Sensor de distância a laser
Taxa de atualização máxima (Baixa precisão)	50 Hz
Taxa de atualização	30 Hz
Faixa de medição	Até 2 metros
Precisão	± 5 %
Tensão de alimentação	3,3V a 5V
Interface	I <sup>2</sup> C
Consumo de corrente	10 mA (máximo)
Dimensões	25 mm x 12,7 mm x 3,5mm
Temperatura de operação	-20°C a 70°C
Campo de visão	25°
Emissor infravermelho	940 nm

Fonte: STMicroelectronics, 2024.

**Servomotor TowerPro MG996R:** é o elemento atuador responsável por alterar a inclinação da barra, por isso foi escolhido um modelo com alto torque, o TowerPro MG996R mostrado na Figura 18 que é capaz de suportar o sistema sem esforço. Informações adicionais do servomotor são mostradas na Tabela 6.

**Figura 18** – Servomotor MG996R

**Fonte:** Ferarmac, 2024.

**Tabela 6** – Características do Servomotor TowerPro MG996R

<b>Característica</b>	<b>Descrição</b>
Modelo	TowerPro MG996R
Tipo	Servomotor de alto torque
Tensão de operação	4,8V - 6,6V
Torque	9,4 kg/cm (4.8V), 11 kg/cm (6V)
Velocidade de operação	0,19 seg/60° (4,8V), 0,15 seg/60° (6V)
Engrenagens	Metal
Faixa de rotação	180°
Dimensões	40,7mm x 19,7mm x 42,9mm
Peso	55g

**Fonte:** Digikey, 2024.

A escolha de um servomotor para controlar a inclinação da barra foi motivada principalmente por conta de seu controlador interno que faz com que o erro no eixo de saída tenda a zero.

**Sensor Ultrassônico HC-SR04:** mostrado na Figura 19, o sensor é um dispositivo amplamente utilizado em projetos de robótica e automação para medir a distância até um objeto. Este sensor opera emitindo pulsos ultrassônicos e medindo o tempo que esses pulsos levam para serem refletidos de volta para o sensor. Com base nessa medida de tempo, é possível calcular a distância até o objeto usando o tempo medido e a velocidade do som.

**Figura 19** – Sensor Ultrassônico HC-SR04

Fonte: Eletrogate, 2024.

O HC-SR04 é conhecido por sua precisão e facilidade de uso. Ele possui uma interface simples de quatro pinos (VCC, GND, Trigger e Echo) e pode ser facilmente integrado ao Arduino. Na Tabela 7, são apresentadas as principais características deste sensor.

**Tabela 7** – Principais características do Sensor Ultrassônico HC-SR04

Característica	Descrição
Tensão de Alimentação	5V
Corrente de Operação	<15mA
Faixa de Medição	2cm a 400cm
Precisão	<3mm
Ângulo de Detecção	15°
Frequência de Operação	40kHz
Dimensões	45mm x 20mm x 15mm

Fonte: Sparkfun, 2024.

**Joystick de 3 Eixos:** representado na Figura 20, é um dispositivo de entrada comumente usado em projetos de controle manual. Este *joystick* permite que o usuário controle a posição em duas direções lidas por potenciômetros e pode ser pressionado, funcionando como um botão. Essas características tornam-no um componente versátil para interação com dispositivos eletrônicos.

**Figura 20** – Joystick de 3 Eixos



**Fonte:** Casa da robótica, 2024.

**Fonte de Alimentação 5V 2,2A:** para garantir o funcionamento adequado de todos os componentes do sistema, foi realizada uma estimativa do consumo máximo de corrente. A escolha de uma fonte de 2,2 A foi baseada na necessidade de fornecer energia suficiente para todos os dispositivos simultaneamente, considerando suas correntes de operação em condições normais e de pico.

A seguir, é apresentado o consumo individual de cada componente:

- ESP32: 3 a 20 mA (em modo Modem Sleep)
- Display LCD 16x2 I2C: 75 a 200 mA (com backlight no máximo)
- Sensor de distância VL53L0X: 10 mA
- Servomotor TowerPro MG996R: 500 a 900 mA
- Sensor Ultrassônico HC-SR04: 15 mA
- Joystick de 3 eixos: até 1 mA

Ao considerar o pior caso, onde todos os dispositivos estão operando no máximo consumo simultaneamente, a corrente total pode ser estimada da seguinte forma:

$$I_{total} = 20 + 200 + 10 + 900 + 15 + 1 = 1146mA = 1,146A \quad (40)$$

Embora o consumo máximo estimado seja de 1,146 A, é essencial dimensionar a fonte com uma margem de segurança para evitar o seu sobrecarregamento. Portanto, foi escolhida uma fonte de 2,2 A que vai operar com uma carga de 52,1% no pior dos casos.

### 3.1 Ferramentas Computacionais Utilizadas

A utilização de ferramentas computacionais desempenha um papel fundamental na simulação e obtenção de resultados em diversos campos da ciência e engenharia. No contexto deste trabalho, as ferramentas *MATLAB*, *Simulink* e *Visual Studio Code* foram utilizadas para a realização de simulações e criação de interfaces para o usuário, proporcionando uma ampla gama de recursos que possibilitam análises detalhadas e a obtenção de resultados do sistema.

O *MATLAB* é amplamente reconhecido por sua versatilidade e capacidade de realizar cálculos e análise de dados. Sua utilização em modelagem matemática, análise de sistemas dinâmicos e resolução de equações diferenciais é fundamental para a obtenção de resultados em diversas áreas, como engenharia, computação e matemática aplicada. Segundo Valentine e Hahn (2022), o *MATLAB* possui técnicas computacionais robustas de cálculo úteis para cientistas e engenheiros.

*Simulink* é uma ferramenta gráfica integrada ao *MATLAB* que facilita a modelagem e simulação de sistemas dinâmicos multidisciplinares. Essa ferramenta permite a criação de modelos por meio de blocos que representam componentes e interações de sistemas, sendo especialmente útil para simulações de sistemas de controle, sistemas mecatrônicos, entre outros. De acordo com Valentine e Hahn (2022), *Simulink* é uma ferramenta essencial para simular e analisar sistemas dinâmicos que permite a simulação em blocos de sistemas contínuos, discretos ou um híbrido entre os dois.

O Arduino IDE, por sua vez, é essencial para o desenvolvimento e programação de sistemas embarcados. Com a popularização do Arduino, essa plataforma tornou-se uma escolha viável para protótipos de baixo custo e de fácil implementação, especialmente em projetos que envolvem sensores e atuadores. Com o passar do tempo e com o apoio da comunidade, a IDE vem recebendo suporte a novas placas de desenvolvimento. De acordo com Banzi e Safary (2011), o Arduino IDE é uma ferramenta poderosa para engenheiros e entusiastas de tecnologia, permitindo a criação de soluções inovadoras em projetos. Sua simplicidade e a grande comunidade de desenvolvedores contribuem para a velocidade no desenvolvimento de protótipos.

O *Visual Studio Community* é uma poderosa IDE (*Integrated Development Environment*) amplamente utilizada para o desenvolvimento de software, incluindo interfaces gráficas para o usuário. Embora seja mais comumente associada ao desenvolvimento de aplicações de software tradicionais, a mesma é capaz de receber dados de um dispositivo externo ao computador para criar gráficos e analisar os dados recebidos.

### 3.2 Sistema Controlado

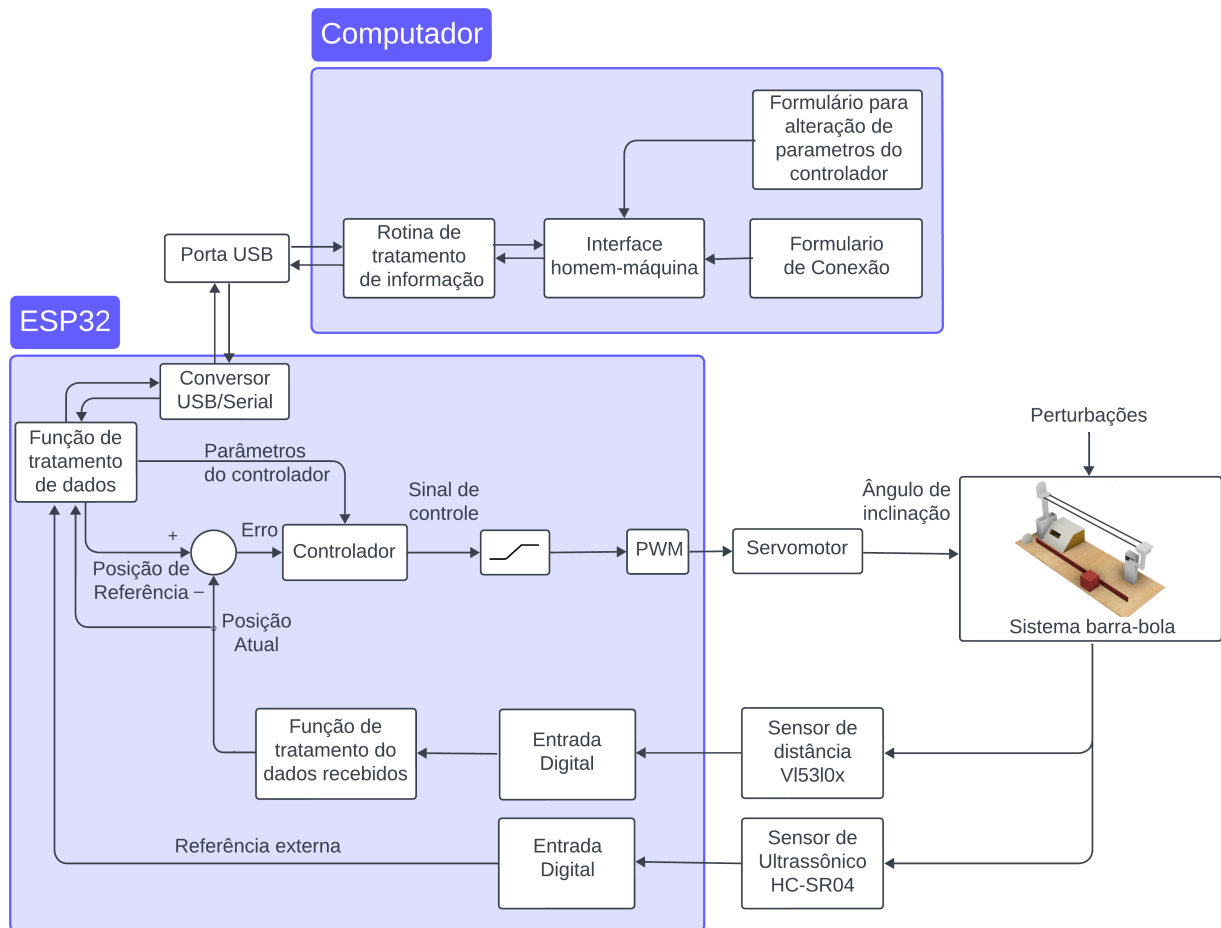
Na Figura 21, é apresentado o diagrama de blocos do sistema, no qual a estratégia de controle pode ser escolhida livremente para manter a bola equilibrada em qualquer posição desejada.

O diagrama da Figura 21 está separado entre o computador responsável pela IHM externa com

funções como a conexão entre os dispositivos, tratamento de dados recebidos e envio de comandos ao sistema; já o microcontrolador ESP32 é responsável pela aquisição dos dados dos sensores, cálculo das ações de controle, geração do sinal PWM responsável pela movimentação do servomotor e tratamento de dados recebidos do sensor de distância ou da comunicação serial.

O sistema conta com dois sensores: o sensor de distância VI5310x é usado para medir a posição da bola, e o sensor ultrassônico HC-SR04 pode ser usado como uma fonte externa de referência, substituindo a referência interna recebida de uma das IHMs disponíveis. Ou seja, a bola vai seguir a posição do segundo objeto posicionado na parte frontal do sensor ultrassônico. A mudança entre referência externa ou interna pode ser feita pelo usuário usando os menus das IHMs.

**Figura 21** – Diagrama em blocos do sistema de controle



**Fonte:** elaborado pelo próprio autor.

O sistema de controle conta com duas IHMs (interfaces homem-máquina), sendo uma integrada, composta por um *display* LCD 16x2 e um *joystick*, com a função de alterar parâmetros dos controladores e métodos de controle ou até mesmo servir para aferição dos sensores; já a IHM

externa é um software desenvolvido para funcionar no sistema operacional *Microsoft Windows*, permitindo alterar parâmetros dos controladores e métodos de controle, além de ser capaz de exibir gráficos da resposta do sistema.

### 3.2.1 IHM Integrada

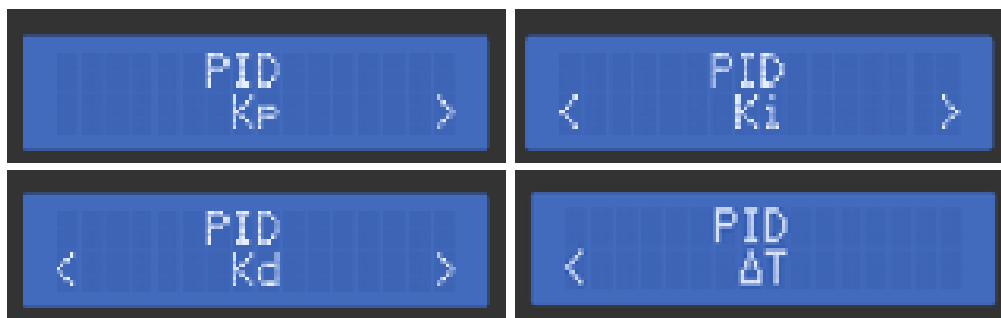
A IHM Integrada é uma interface ideal para pequenas mudanças como alteração dos parâmetros  $K_p$ ,  $K_i$  e  $K_d$  dos controladores PID, I-PD e *fuzzy*-PID Híbrido, ou até mudanças rápidas na referência (*setpoint*) e aferição das medições dos sensores V15310x e HC-SR04 para verificar que os dados obtidos pelo sistema sejam condizentes com a realidade. Os dados são exibidos em um *display* LCD e a navegação entre os menus é feita usando um *joystick*. As principais telas da IHM Integrada são: o menu principal (ver Figura 22), contendo as opções para alterar o tipo de controle do sistema, modificar o valor de referência, controlar manualmente o sistema e realizar aferição das medições dos sensores; os submenus do controlador PID onde são mostradas as opções para alterar os valores dos parâmetros  $K_p$ ,  $K_i$ ,  $K_d$  e  $\Delta T$  mostrados nas Figuras 23 e 24; e por último, na Figura 25, são mostradas as leituras dos sensores.

**Figura 22** – Menu Principal

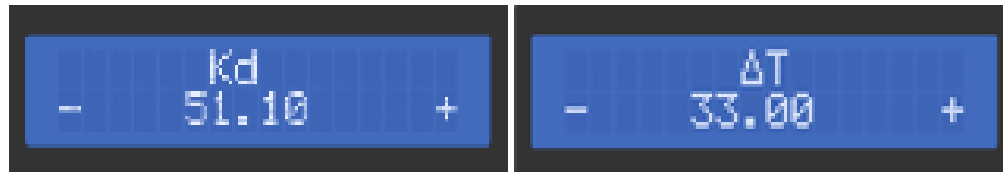


**Fonte:** elaborado pelo próprio autor.

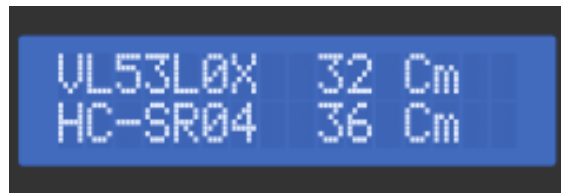
**Figura 23** – Submenus do Menu PID



**Fonte:** elaborado pelo próprio autor.

**Figura 24** – Submenus dos menus  $K_d$  e  $\Delta T$  PID

**Fonte:** elaborado pelo próprio autor.

**Figura 25** – Tela do menu com a leitura dos sensores

**Fonte:** elaborado pelo próprio autor.

### 3.2.2 IHM Externa

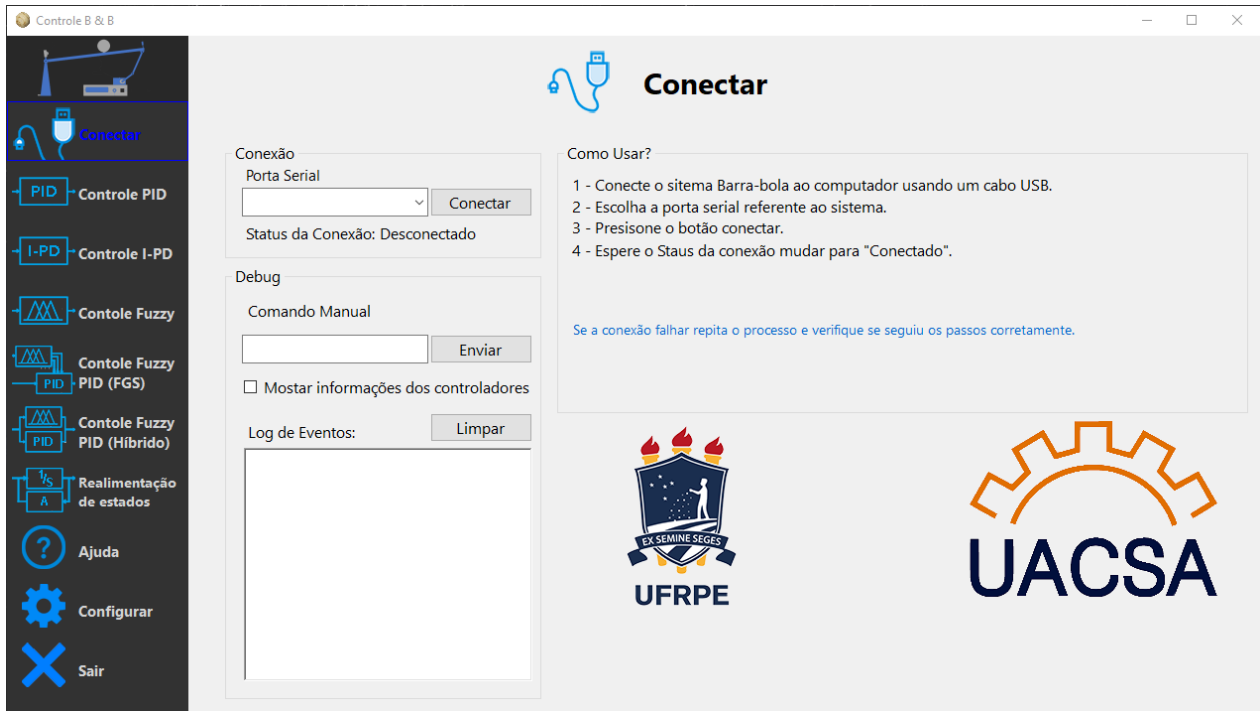
A IHM Externa é uma interface ideal para fazer uma grande quantidade de testes com maior agilidade, capaz de fazer tudo que a IHM Integrada faz, além de ser essencial na criação de gráficos para comparar as respostas do sistema para diferentes parâmetros. As principais telas da IHM Externa são mostradas nas Figuras 26, 27 e 28.

A IHM Externa foi desenvolvida na linguagem *C Sharp* com auxílio do *software Visual Studio 2022 Community Edition* com base na plataforma de aplicativos *.NET* na versão 4.7.2, tornando-a compatível com as versões mais recentes do sistema operacional *Microsoft Windows*.

Todas as telas da IHM Externa podem ser acessadas usando o menu lateral onde é possível chegar até as telas “Conecta”, “Controle PID”, “Controle I-PD”, “Controle *Fuzzy*”, “Controle *Fuzzy*-PID FGS”, “Controle *Fuzzy*-PID Híbrido”, “Controle por realimentação de estados”, “Ajuda” e “Configurar”.

A tela “Conectar” mostrada na Figura 26 é a tela principal e onde a IHM inicia. Na mesma estão descritos os passos necessários para conectar o computador ao sistema barra-bola, o formulário de conexão, onde é possível selecionar a porta COM em que o sistema foi conectado e também uma seção reservada para testes chamada de “*Debug*”.

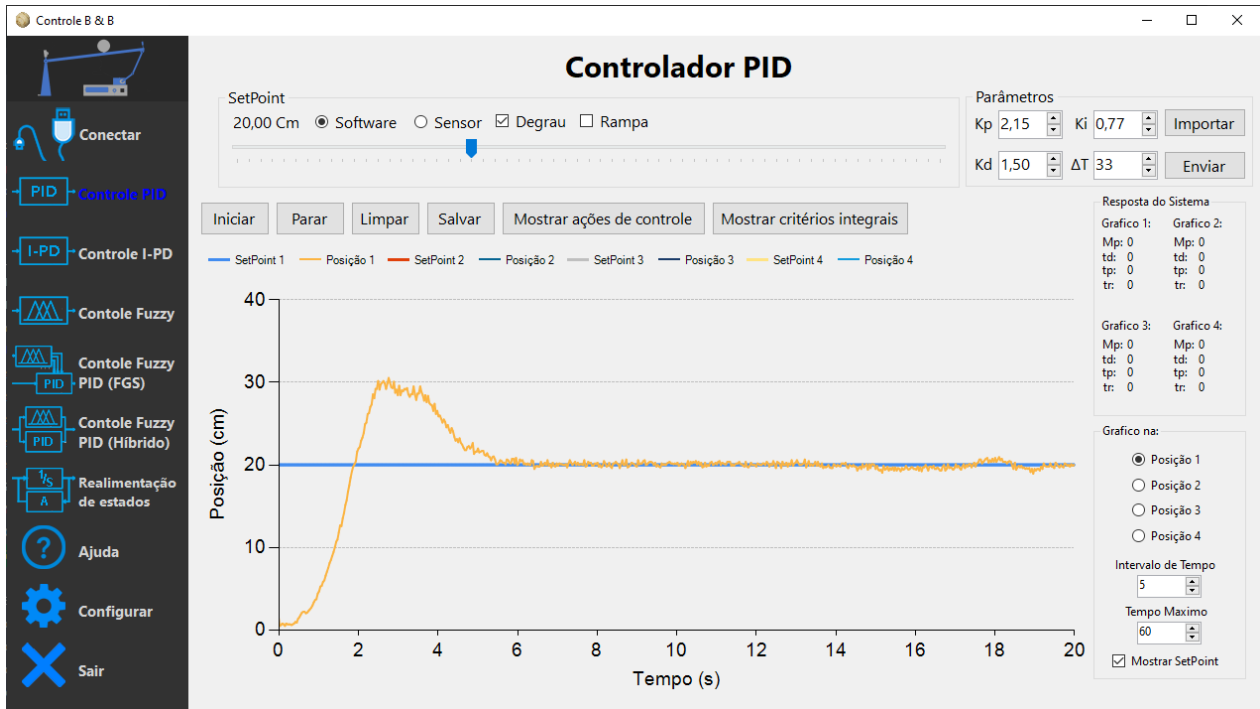
**Figura 26** – Tela “Conectar”



**Fonte:** elaborado pelo próprio autor.

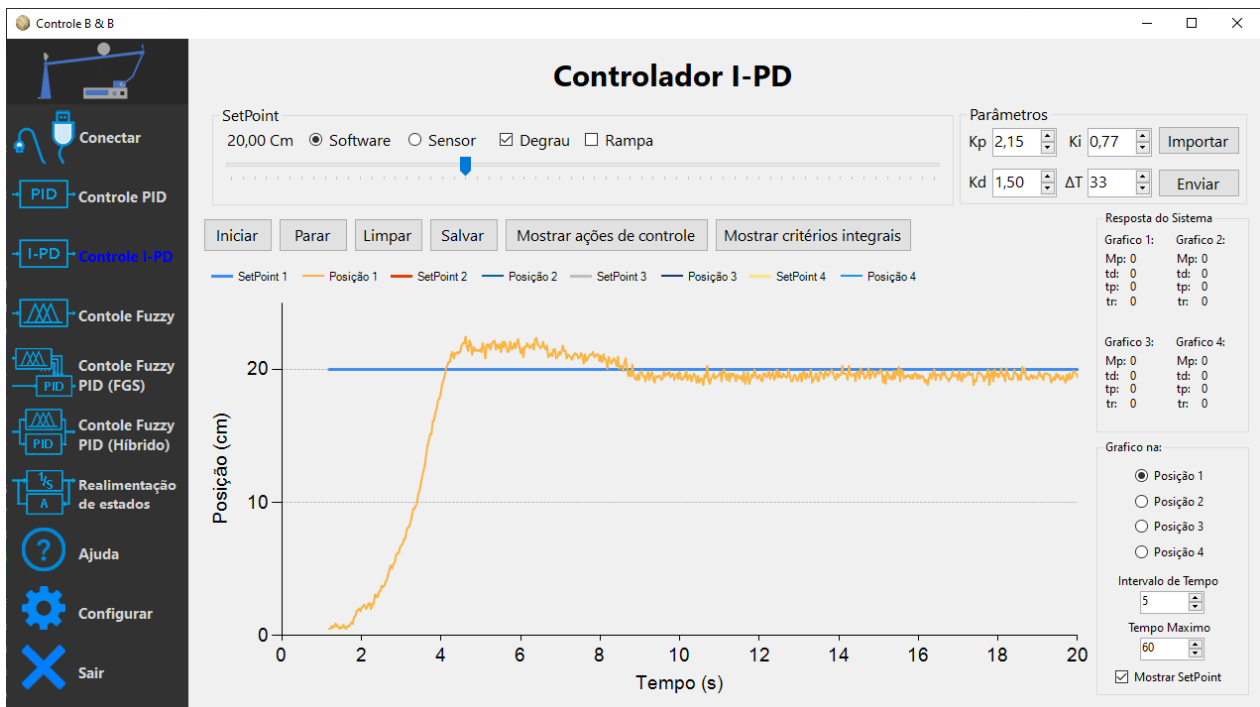
As telas “Controle PID”, “Controle I-PD”, “Controle *Fuzzy*”, “*Fuzzy*-PID FGS”, “Controle *Fuzzy*-PID Híbrido” e “Controle por realimentação de estados” são responsáveis por: alterações nos parâmetros de seus respectivos controladores, cálculo de critérios integrais e criação de gráficos de posição e das ações de controle do sistema em tempo real. Nessas telas é possível alterar a referência do sistema para um valor fixo, uma rampa ou alterá-la para um valor externo recebido do sensor HC-SR04, plotar e sobrepôr até quatro respostas do sistema escolhendo o tempo de teste, salvar os gráficos gerados como imagens e alterar em tempo real os parâmetros  $K_p$ ,  $K_i$ ,  $K_d$  e  $\Delta T$  nos controladores PID e I-PD ou modificar as funções de pertinência dos controladores *fuzzy*. Os controladores *fuzzy* são casos especiais, por conta de sua maior quantidade de parâmetros, foram necessários menus extras para configurá-los. A Figura 30 mostra o menu extra para seleção individual das regras do controlador que, ao selecionar cada uma delas, será exibido o menu referente a essa regra, como mostrado na Figura 31.

**Figura 27 – Tela “Controlador PID”**



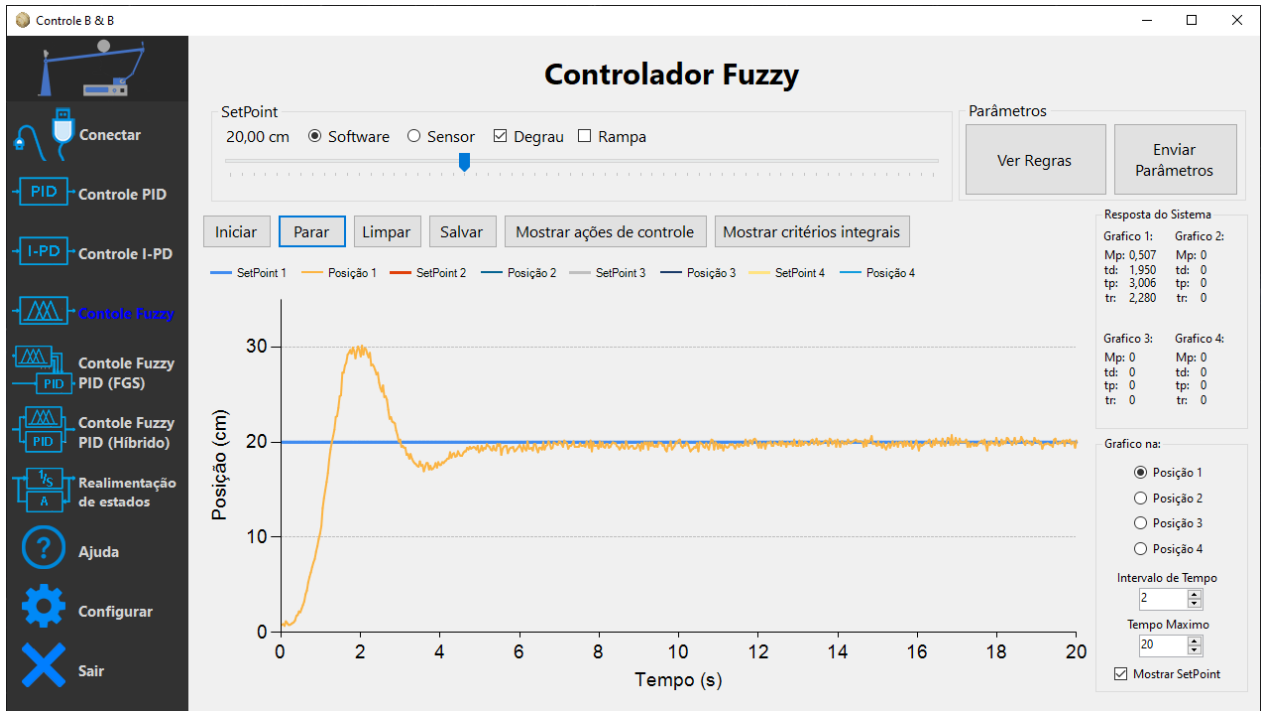
Fonte: elaborado pelo próprio autor.

**Figura 28 – Tela “Controlador I-PD”**



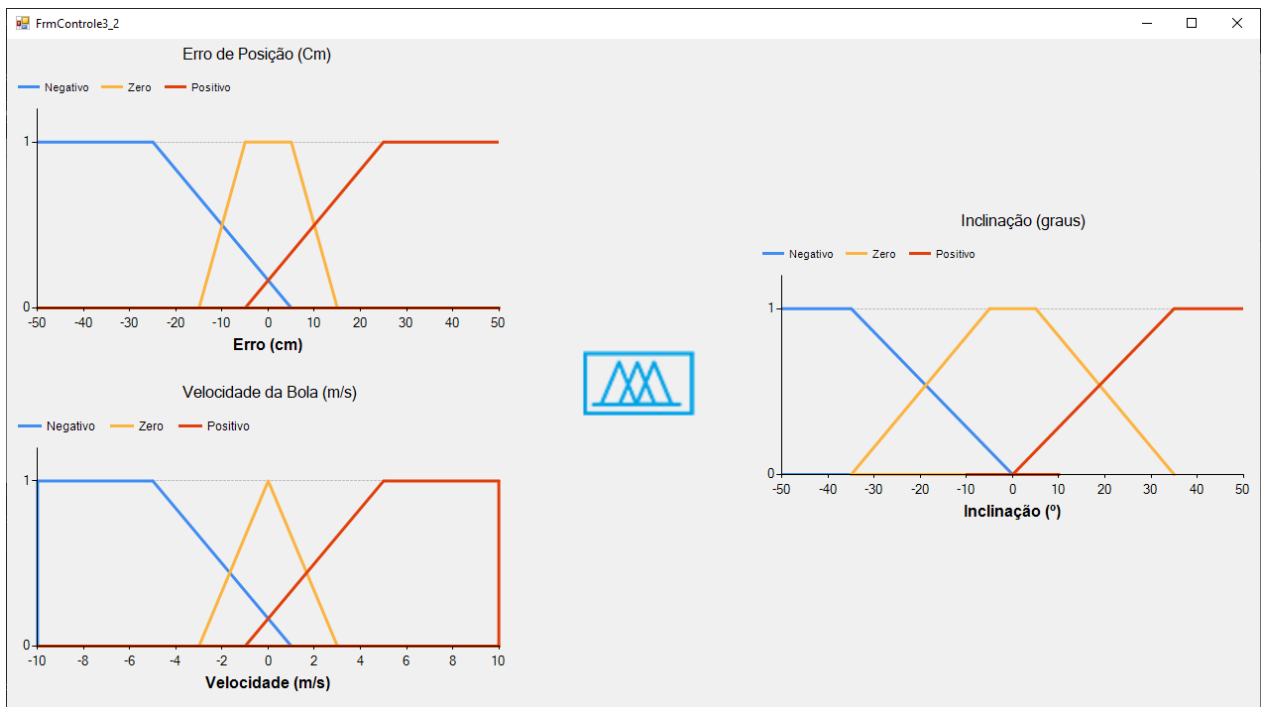
Fonte: elaborado pelo próprio autor.

**Figura 29** – Tela “Controle *fuzzy*”



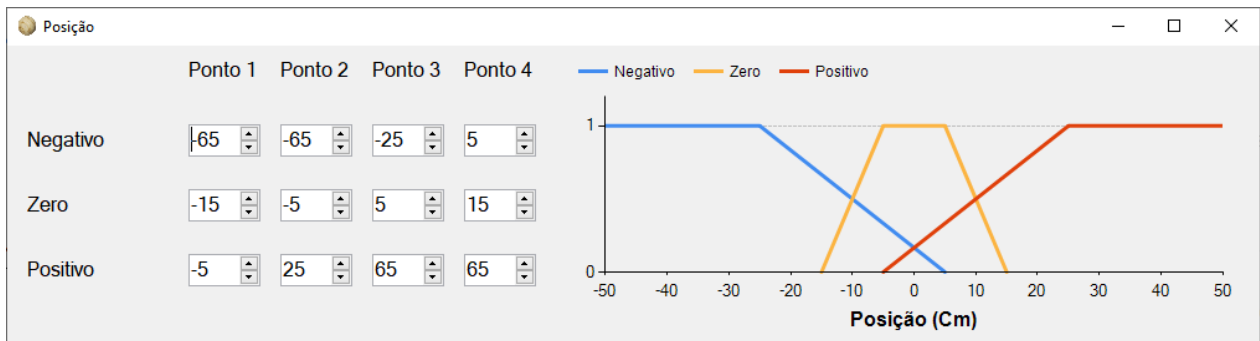
Fonte: elaborado pelo próprio autor.

**Figura 30** – Tela “Regras do controlador *fuzzy*”



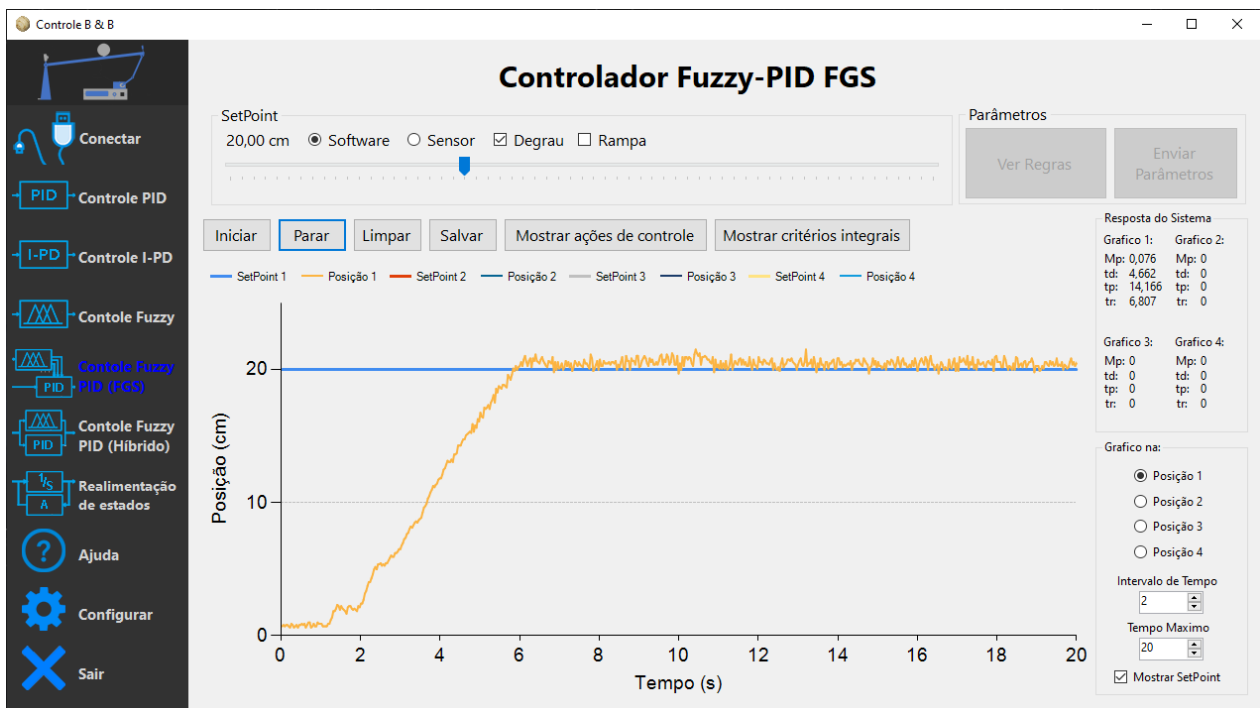
Fonte: elaborado pelo próprio autor.

**Figura 31** – Tela “Parâmetros de erro de posição do controlador *fuzzy*”



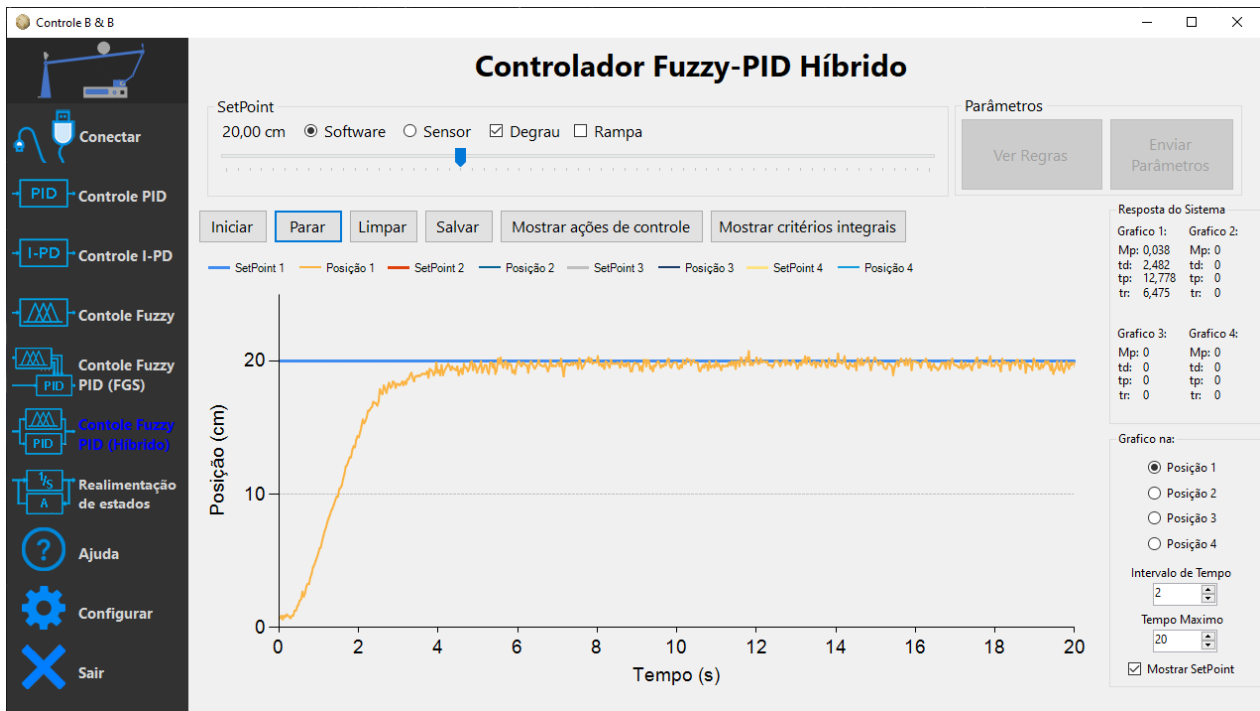
Fonte: elaborado pelo próprio autor.

**Figura 32** – Tela “Controlador *fuzzy*-PID FGS”



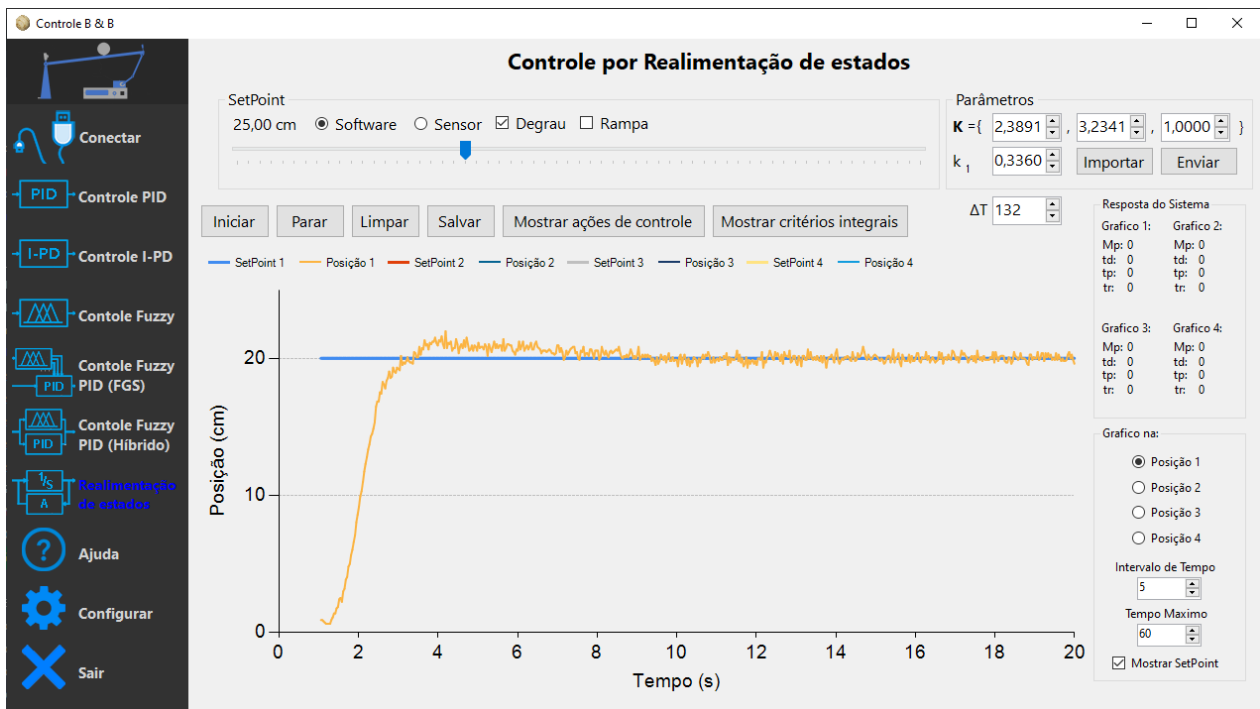
Fonte: elaborado pelo próprio autor.

Figura 33 – Tela “Controlador *fuzzy*-PID Híbrido”



Fonte: elaborado pelo próprio autor.

Figura 34 – Tela “Controlador por realimentação de estados”



Fonte: elaborado pelo próprio autor.

A tela de Ajuda mostrada nas Figuras 35 e 36 contém instruções passo a passo de como usar a IHM Externa. E, por último, o menu configurar, onde é possível ajustar os valores lidos pelos sensores e definir os valores padrão dos parâmetros de cada um dos controladores.

**Figura 35 – Tela “Ajuda - parte 1”**

**Ajuda**

**Como alterar parâmetros dos controladores PID e I-PD ?**

- 1 - Selecione o controlador desejado usando o menu lateral;
- 2 - Procure o grupo parâmetros na parte direita da tela;
- 3 - Altere os parâmetros desejados usando as setas ou escreva o valor desejado;
- 4 - Pressione o botão “Enviar” junto aos parâmetros.

**Como alterar o setpoint (Referência) do sistema?**

- 1 - Selecione o controlador desejado usando o menu lateral;
- 2 - Procure o grupo “setpoint”;
- 3 - Escolha uma referência: software ou sensor;
  - Caso escolha software a referência deve ser escolhida usando o controle rolável
  - Caso escolha Externo a referência será definida por qualquer objeto que seja colocado em frente ao sensor HC-SR04;
- 4 - Pressione o botão “Enviar” junto aos parâmetros.

**Qual é a utilidade dos botões junto ao gráfico?**

- 1 - Botão “Iniciar” ao selecionar o botão o sistema passa a funcionar com os últimos parâmetros enviados, gráficos com o valor de posição e de referência, também serão criados em tempo real;
- 2 - Botão “Parar” ao selecionar o botão o sistema manter a bola para na posição inicial (Próxima ao sensor V15310x);
- 3 - Botão “Limpar” ao selecionar o botão os gráficos criados em tempo real serão apagados;

**Fonte:** elaborado pelo próprio autor.

Figura 36 – Tela “Ajuda - parte 2”

**Qual é a utilidade dos botões junto ao gráfico?**

- 1 - Botão "Iniciar" ao selecionar o botão o sistema passa a funcionar com os últimos parâmetros enviados, gráficos com o valor de posição e de referência, também serão criados em tempo real;
- 2 - Botão "Parar" ao selecionar o botão o sistema manter a bola para na posição inicial (Próxima ao sensor VL53L0x);
- 3 - Botão "Limpar" ao selecionar o botão os gráficos criados em tempo real serão apagados;
- 4 - Botão "Salvar" ao selecionar o botão os gráficos contidos na tela serão salvos na pasta em que esse software está instalado.

**Qual é a utilidade dos botões junto ao gráfico?**

- 1 - Ao terminar de desenhar o primeiro gráfico selecione o botão "Parar";
- 2 - Procure o grupo gráficos e escolha a "posição 2" ou outra que não tenha sido usada;
- 3 - Caso ache necessário repita os passos 1 e 2 para criar um terceiro e um quarto gráfico;
- 4 - Se o setpoint for o mesmo é recomendado desativar a opção "mostrar setpoint" para evitar sobreposições desnecessárias.

Fonte: elaborado pelo próprio autor.

Figura 37 – Tela “Configurar”

**Configurações**

Sensores

VL53L0x: 0,00 cm      Fator de correção VL53L0X: 2,007

HC-SR04: 0,00 cm      Fator de correção HC-SR04: 0,110

Alterar

Valores padrão do controlador PID

Kp 2,15    Ki 0,77    Kd 1,50    ΔT (ms) 33    Salvar

Valores padrão para gráfico

Intervalo de Tempo 5    Tempo Maximo 60    Salvar

Valores padrão do controlador I-PD

Kp 2,15    Ki 0,77    Kd 1,50    ΔT (ms) 33    Salvar

Valores padrão Realimentação de estados

$K = \{ 2,3891, 3,2341, 1,0000 \}$      $k_1 = 0,3360$     Salvar

Fonte: elaborado pelo próprio autor.

#### 4 RESULTADOS E DISCUSSÕES

Voltado à Equação (14), agora com todas as medidas necessárias (Ver Tabela 2) para calcular a constante  $C$  e substituindo na Equação (41), chega-se à função de transferência do sistema.

$$\frac{X(s)}{\theta(s)} = \frac{0,7}{s^2} \quad (41)$$

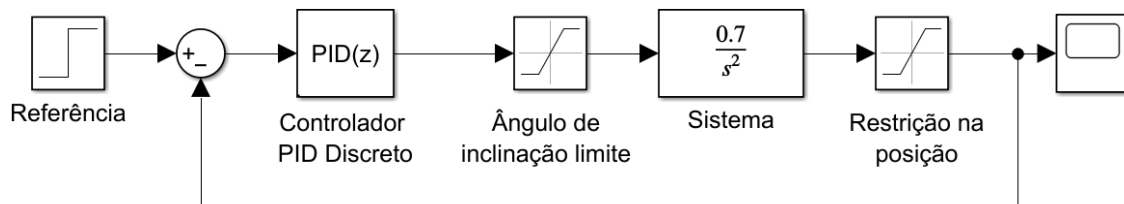
Para controlar a posição da bola, o sistema utiliza o ângulo da barra como variável de controle. Ao inclinar a barra, a bola se desloca devido à força da gravidade. Pequenas variações no ângulo alteram a inclinação das hastes de apoio da bola, gerando uma aceleração que a move na direção desejada.

O controlador ajusta dinamicamente essa inclinação para compensar desvios da posição desejada. Quando a bola está à esquerda da posição de referência, a barra é inclinada ligeiramente de modo que o ângulo  $\theta$  seja negativo, fazendo com que a bola role na direção correta. Da mesma forma, se a bola ultrapassar a posição desejada, a inclinação é invertida para desacelerar e estabilizar sua posição. Esse processo ocorre de forma contínua e suave, garantindo que a bola permaneça próxima ao ponto de equilíbrio. Logo, o ajuste fino do controlador é essencial para evitar oscilações excessivas e garantir uma resposta rápida e precisa.

##### 4.1 Controle PID

Na Figura 38 é mostrado o diagrama de blocos do sistema utilizado para simulação; a mesma possui a função de transferência do sistema; o ângulo de inclinação limite de 40 graus, que evita que o parafuso do braço do servomotor colida com sua base; a restrição na posição é simplesmente o limite físico onde a bola pode se mover, indo do início ao fim da barra (0 - 0,6 m); e por fim, o controlador PID discreto dado pela Equação (24) e utiliza os parâmetros  $K_p = 2,15$ ,  $K_i = 0,77$  e  $K_d = 1,5$ , obtidos de forma empírica.

**Figura 38** – Sistema com controlador PID discreto



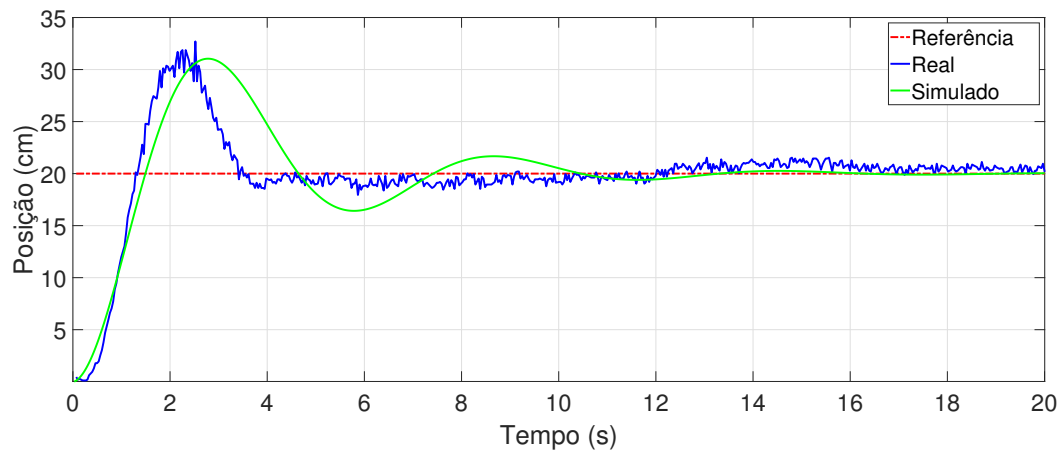
**Fonte:** elaborado pelo próprio autor.

A Figura 39 mostra a comparação entre os resultados simulados e práticos para uma referência do tipo degrau de 20 cm. O sistema simulado obteve um desempenho levemente pior que na prática, como mostrado na Tabela 8.

**Tabela 8** – Especificações da resposta transitória para uma referência degrau de 20 cm

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	1,49	1,35	-0,14
Máximo sobressinal (%)	55,3	63,4	+8,1
Tempo de pico (s)	2,78	2,4	-0,38
Tempo de assentamento(s)	10,5	4,0	-6,5

Fonte: Elaborado pelo próprio autor.

**Figura 39** – Saída usando PID discreto

Fonte: elaborado pelo próprio autor.

Já se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados na Tabela 9, onde, nos critérios ISE, ITSE e IAE, o sistema real se saiu melhor que a simulação, sendo inferior somente no critério ITAE. Os motivos mais prováveis do pior desempenho obtido no critério ITAE são o desvio padrão na leitura do sensor de distância e as pequenas oscilações na posição da bola após alcançar o estado estacionário, que fazem com que o índice aumente conforme o tempo avança, em especial neste caso, uma vez que esse critério penaliza erros mais tardios.

**Tabela 9** – Critérios integrais para um degrau usando controle PID

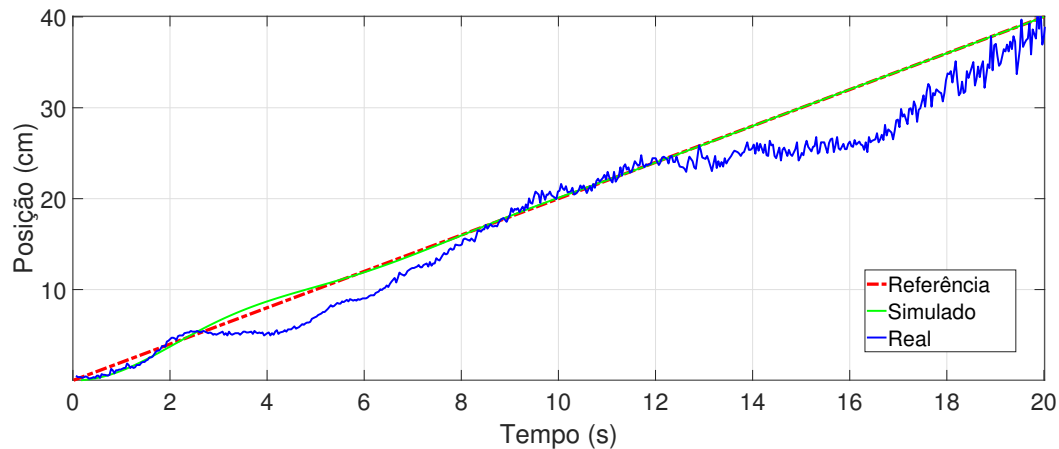
Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0475	0,0787	0,5028	1,598
Real	0,2 m	0,0432	0,0523	0,4356	1,697

Fonte: Elaborado pelo próprio autor.

A Figura 40 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição foi de 0,15 centímetros; enquanto nos

ensaios práticos foi de 1,14 centímetros, sendo 7,6 vezes maior que nas simulações. Aumento causado principalmente pela imprecisão na leitura do sensor que faz com que o sistema não corrija a posição de maneira ideal.

**Figura 40** – Comparação entre os resultados ante uma rampa usando controlador PID



**Fonte:** elaborado pelo próprio autor.

Como o sistema real apresentou mais oscilações e um maior erro de estado estacionário em comparação com as simulações, era esperado que os critérios de avaliação de desempenho integrais (Tabela 10) refletissem essa diferença. De fato, nos testes, as respostas obtidas apresentaram índices de desempenho significativamente distintos, com a simulação superando o sistema real em todos os critérios.

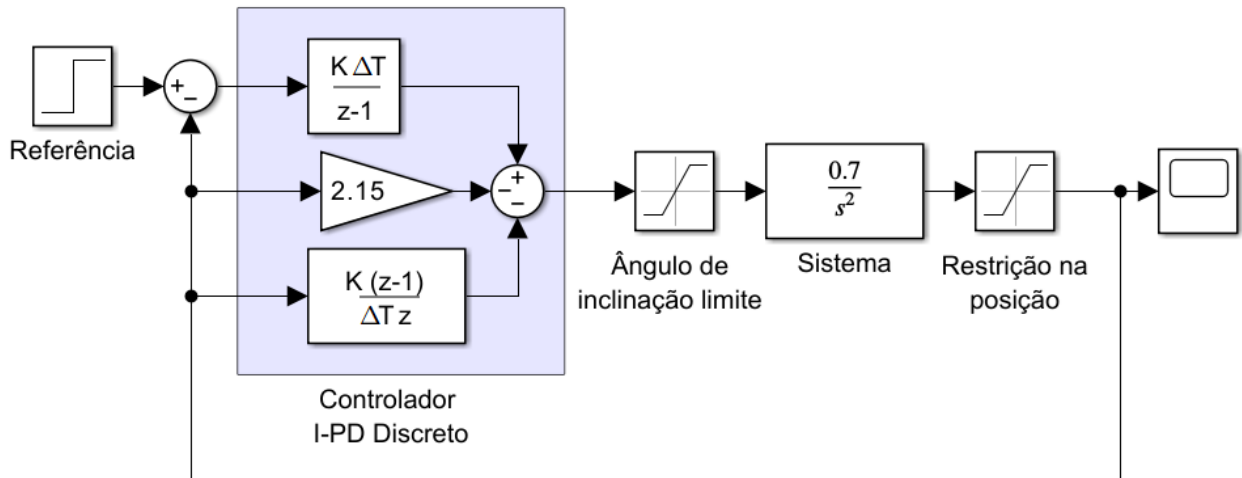
**Tabela 10** – Critérios integrais para uma rampa usando controle PID

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0002	0,0005	0,0356	0,1306
Real	0 a 0,4 m	0,0114	0,1576	0,3754	4,5970

Fonte: Elaborado pelo próprio autor.

## 4.2 Controle I-PD

Na Figura 41 se mostra o diagrama de blocos simplificado na simulação usando o controlador I-PD discreto dado pela Equação (29). Foram utilizados os mesmos parâmetros de sintonia que o controlador PID para permitir avaliar as diferenças de desempenho entre as duas estratégias de controle.

**Figura 41** – Controlador I-PD discreto aplicados ao sistema

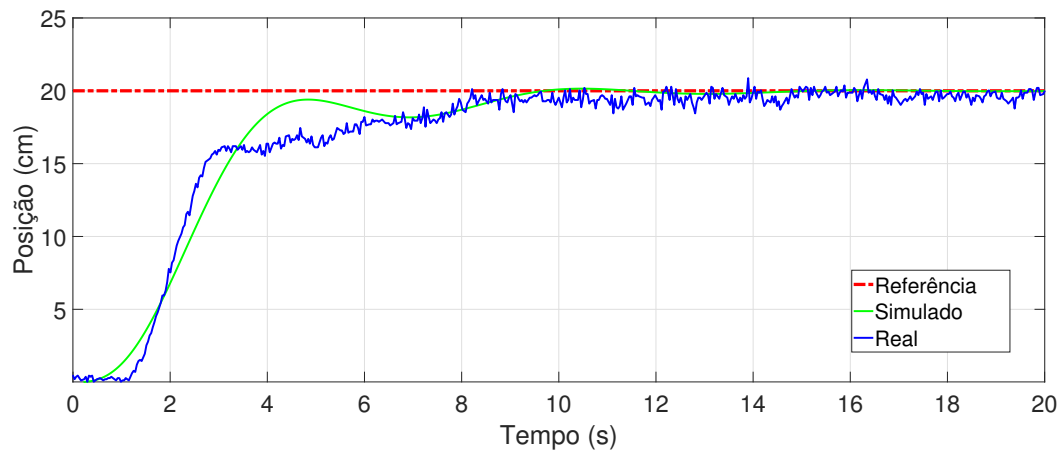
**Fonte:** elaborado pelo próprio autor.

A Figura 42 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo degrau. O sistema simulado obteve um desempenho levemente melhor que na prática; a comparação entre os resultados pode ser vista na Tabela 11.

**Tabela 11** – Especificações da resposta transitória para uma referência degrau de 20 cm usando controle I-PD

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	9,65	8,32	-1,32
Máximo sobressinal (%)	0,75	4,00	+3,25
Tempo de pico (s)	10,5	15,0	+4,50
Tempo de assentamento(s)	9	8,2	-0,8

Fonte: Elaborado pelo próprio autor.

**Figura 42** – Comparação entre os resultados ante um degrau usando controlador I-PD

**Fonte:** elaborado pelo próprio autor.

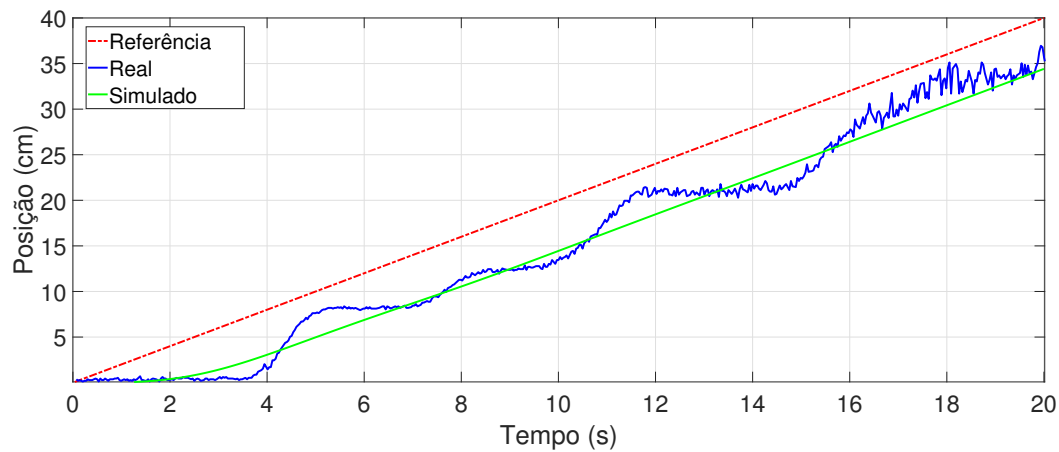
Se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados na Tabela 12, onde, em todos os critérios, o sistema real se saiu levemente inferior que na simulação. Essa diferença se deve, principalmente, à presença de efeitos como não linearidade e atrito nos testes práticos, os quais influenciaram a dinâmica do sistema, tornando seu comportamento diferente do previsto nas simulações.

**Tabela 12** – Critérios integrais para um degrau usando controle I-PD

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0774	0,0909	0,5599	1,201
Real	0,2 m	0,0979	0,1502	0,7389	2,366

Fonte: Elaborado pelo próprio autor.

A Figura 43 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição médio foi de 5,56 centímetros; enquanto nos ensaios práticos o erro médio foi de 5,4 centímetros, sendo 0,16 centímetros menor do que o esperado, o que é aceitável levando em conta que o tamanho do erro.

**Figura 43** – Comparação entre os resultados ante uma rampa usando controlador I-PD

**Fonte:** elaborado pelo próprio autor.

Embora o sistema real tenha apresentado oscilações que não acontecem nas simulações, os resultados da avaliação de desempenho por critérios integrais, apresentados na Tabela 13, indicam que os índices de desempenho foram próximos em ambos os casos. No entanto, a simulação obteve um desempenho ligeiramente superior, especialmente nos critérios ITSE e ITAE, que consideram o tempo como fator determinante.

**Tabela 13** – Critérios integrais para uma rampa usando controle I-PD

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0528	0,5994	1,002	10,92
Real	0 a 0,4 m	0,0756	0,8334	1,174	12,53

Fonte: Elaborado pelo próprio autor.

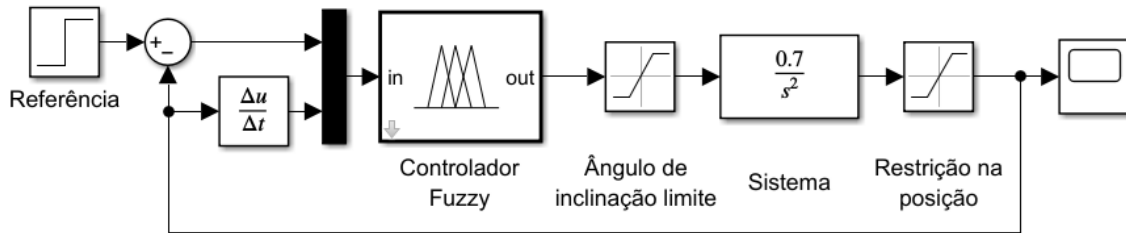
### 4.3 Controle *fuzzy*

O controlador *fuzzy* mostrado no diagrama de blocos da Figura 44 calcula a inclinação da barra com base no erro de posição e também na variação da posição da bola, que é calculada de acordo com a Equação (42).

$$v = \frac{x[n] - x[n-1]}{\Delta t} \quad (42)$$

sendo:  $v$  velocidade da bola,  $x[n]$  e  $x[n-1]$ , a posição atual e anterior da bola, respectivamente.

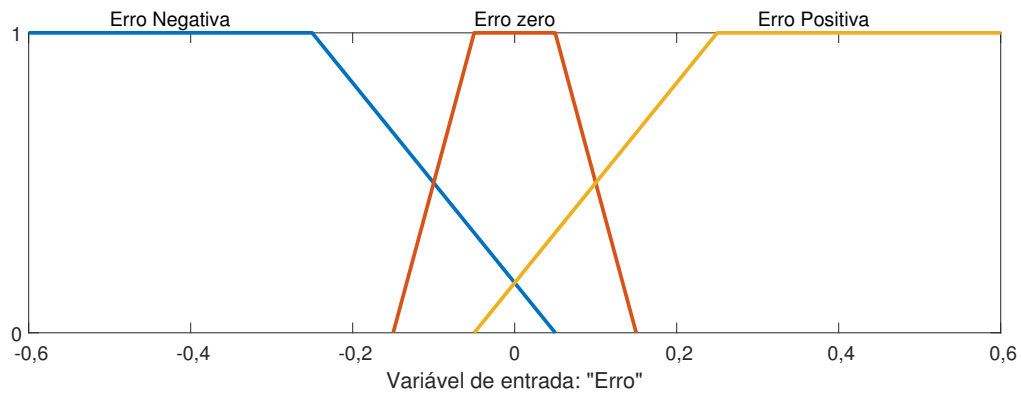
**Figura 44** – Controlador *fuzzy* aplicado ao sistema



**Fonte:** elaborado pelo próprio autor.

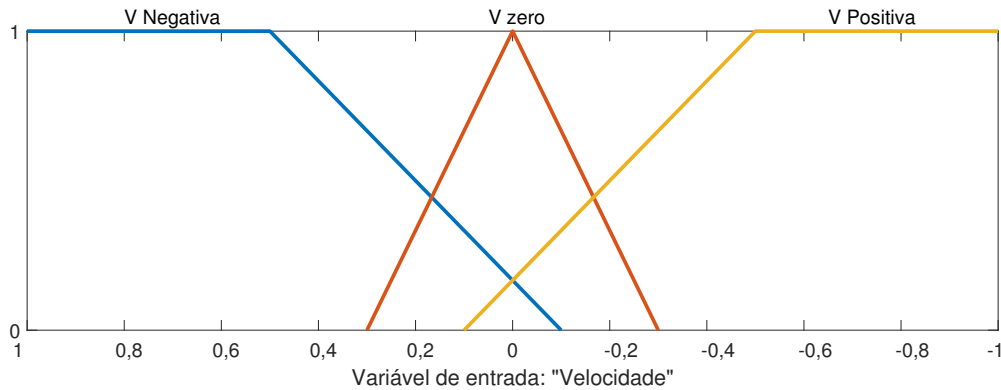
As variáveis linguísticas que representam o erro de posição são: Erro Negativo, Erro Zero e Erro Positivo. As mesmas são trapezoidais e são mostradas no gráfico da função de pertinência (Figura 45), onde o eixo das abscissas representa o valor do erro em metros e o eixo das ordenadas indica o grau de pertinência do valor de erro a um conjunto *fuzzy* específico.

**Figura 45** – Função de pertinência do erro de posição



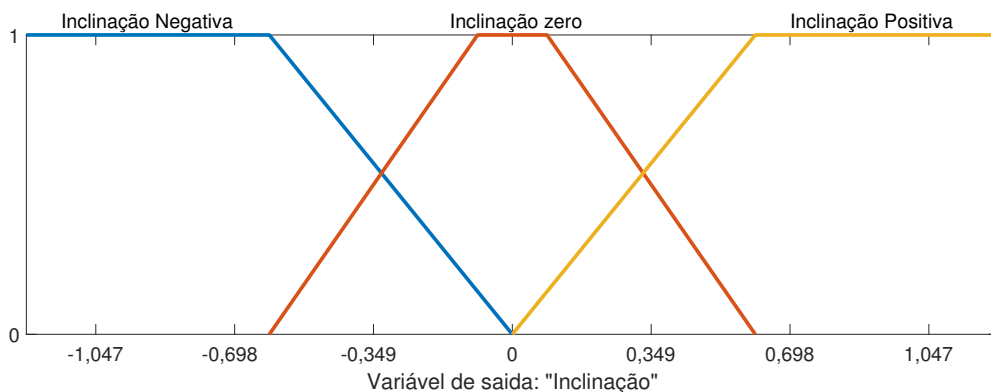
**Fonte:** elaborado pelo próprio autor.

Já as variáveis linguísticas que representam a velocidade da bola são: Velocidade negativa, Velocidade zero e Velocidade positiva; todas trapezoidais e são mostradas na Figura 46 onde o eixo das abscissas representa a velocidade da bola em metros por segundo e o eixo das ordenadas indica o grau de pertinência do valor de velocidade a um conjunto *fuzzy* específico.

**Figura 46** – Função de pertinência da velocidade da bola

**Fonte:** elaborado pelo próprio autor.

E por último, a variável de saída também conhecida como consequente, se divide em 3 variáveis linguísticas: inclinação negativa, inclinação zero e inclinação positiva, as quais são trapezoidais e mostradas na Figura 47 onde o eixo das abscissas representa o valor da inclinação em radianos e o eixo das ordenadas indica o grau de pertinência do valor de inclinação para um conjunto *fuzzy* específico.

**Figura 47** – Função de pertinência do ângulo do servomotor

**Fonte:** elaborado pelo próprio autor.

Finalmente, o conjunto de regras do controlador é baseado nas variáveis linguísticas:

Se (Erro é negativo) então (Inclinação é negativa).

Se (Erro é negativo e Velocidade é positiva) então (Inclinação é zero).

Se (Erro é zero e Velocidade é negativa ) então (Inclinação é negativa).

Se (Erro é zero e Velocidade é zero) então (Inclinação é zero).

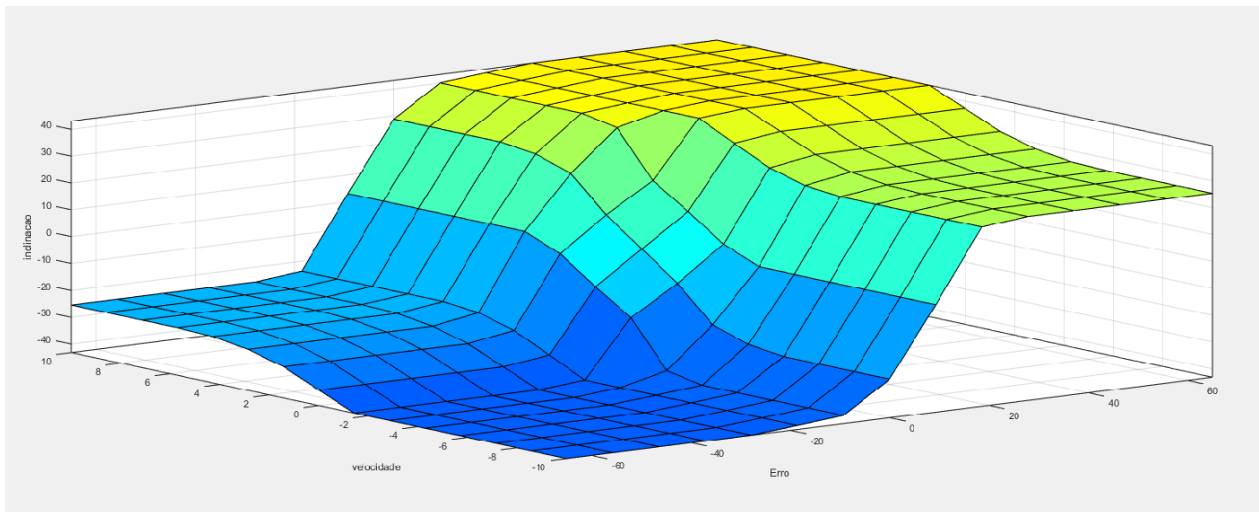
Se (Erro é zero e Velocidade é positiva ) então (Inclinação é positiva).

Se (Erro é positivo e Velocidade é negativa) então (Inclinação é zero).

Se (Erro é positivo) então (Inclinação é positiva).

Usando um sistema de inferência mamdani *Fuzzy* com as funções de pertinência do Erro, Velocidade e Inclinação, chega-se à superfície *fuzzy* mostrada na Figura 48.

**Figura 48** – Superfície *fuzzy*



**Fonte:** elaborado pelo próprio autor.

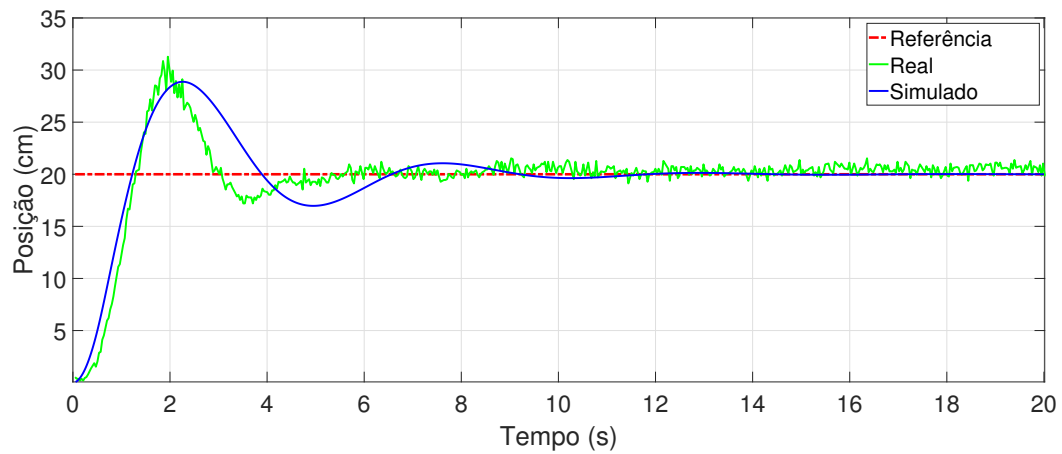
A Figura 49 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo degrau. O sistema simulado obteve um desempenho levemente pior que na prática, como pode ser visto na Tabela 14.

**Tabela 14** – Comparação entre os resultados simulados e práticos para uma referência degrau de 20 cm usando controle *fuzzy*

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	1,22	1,29	+0,07
Máximo sobressinal (%)	44,3	56,4	+12,1
Tempo de pico (s)	2,26	1,96	-0,3
Tempo de assentamento(s)	8,5	4,9	-3,3

Fonte: Elaborado pelo próprio autor.

**Figura 49** – Comparação entre os resultados ante um degrau usando controlador *fuzzy*



**Fonte:** elaborado pelo próprio autor.

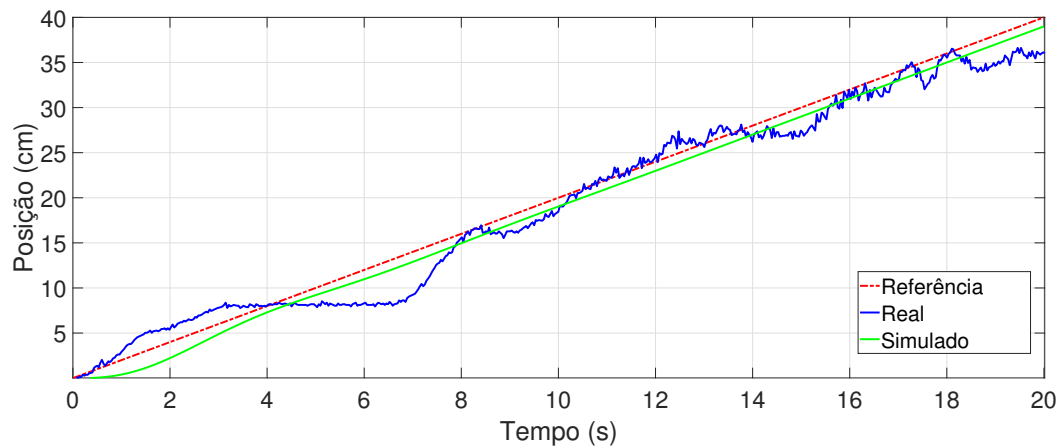
Já se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados na Tabela 15, onde, nos critérios ITSE e IAE o sistema real se saiu levemente melhor que a simulação, sendo um pouco inferior à simulação nos critérios ISE e ITAE.

**Tabela 15** – Critérios integrais para um degrau usando controle *fuzzy*

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0338	0,0389	0,369	0,9146
Real	0,2 m	0,0365	0,0334	0,3638	1,2540

Fonte: Elaborado pelo próprio autor.

Já a Figura 50 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição médio foi de 1,01 centímetros; enquanto nos ensaios práticos, o erro médio foi de 1,6 centímetros, sendo 0,59 centímetros menor do que o esperado.

**Figura 50** – Comparação entre os resultados ante uma rampa usando controlador *fuzzy*

**Fonte:** elaborado pelo próprio autor.

Mesmo o sistema real apresentando mais oscilações que nas simulações, para a avaliação de desempenho por critérios integrais (Tabela 16), ambas as respostas obtiveram índices de desempenho próximos, mesmo com a simulação se saindo um pouco melhor em todos os critérios. Sendo o principal motivo as oscilações da bola no entorno da referência.

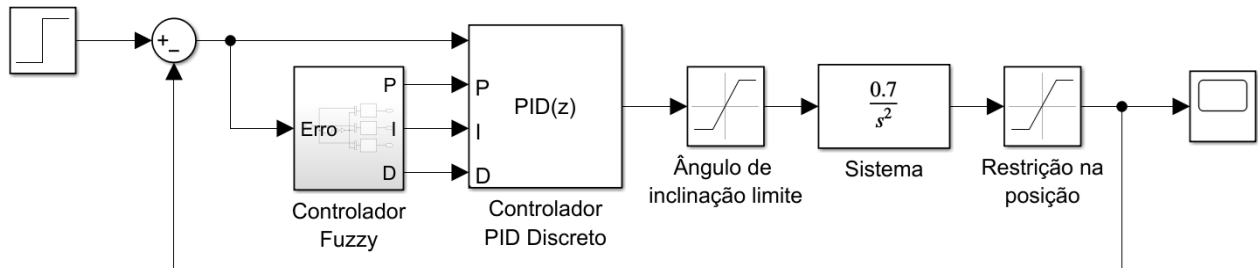
**Tabela 16** – Critérios integrais para uma rampa usando controle *fuzzy*

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0023	0,0205	0,2081	2,011
Real	0 a 0,4 m	0,0060	0,0525	0,2805	2,625

Fonte: Elaborado pelo próprio autor.

#### 4.4 Controle *fuzzy*-PID (FGS)

O controlador *Fuzzy*-PID utiliza a topografia FGS para calcular os novos ganhos proporcional, integral e derivativo do sistema em tempo real de forma dinâmica. A Figura 51 apresenta o diagrama em blocos do sistema modelado no *Simulink*, destacando a interface entre o controlador *fuzzy* e o controlador PID.

**Figura 51** – Diagrama em blocos do controle *Fuzzy*-PID FGS

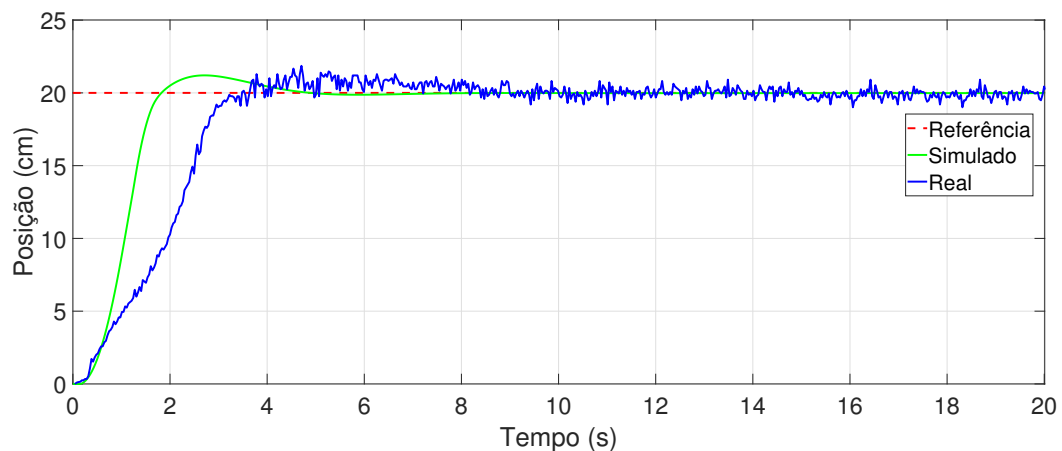
**Fonte:** elaborado pelo próprio autor.

A Figura 52 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo degrau. O sistema simulado obteve um desempenho levemente melhor que o prático, como mostrado na Tabela 17

**Tabela 17** – Especificações da resposta transitória para uma referência degrau de 20 cm usando controle *Fuzzy*-PID FGS

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	1,82	3,50	+1,68
Máximo sobressinal (%)	6,50	8,50	+2,00
Tempo de pico (s)	2,70	4,70	+2,00
Tempo de assentamento(s)	3,9	4,4	0,5

Fonte: Elaborado pelo próprio autor.

**Figura 52** – Comparação entre os resultados ante um degrau usando controle *Fuzzy*-PID FGS

**Fonte:** elaborado pelo próprio autor.

Já se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados

na Tabela 18, onde em todos os critérios, a simulação apresenta desempenho melhor; o que já era de se esperar, uma vez que o sistema real foi significativamente mais lento que o sistema simulado.

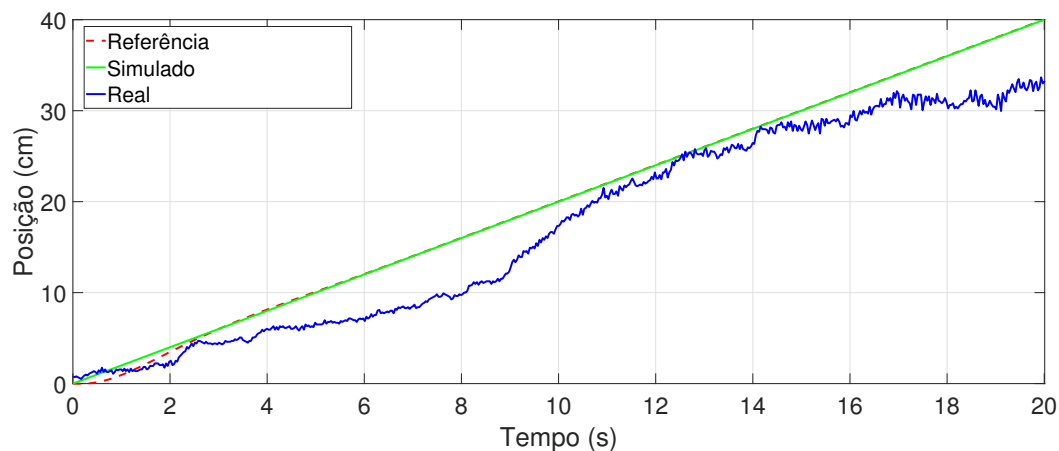
**Tabela 18** – Critérios integrais para um degrau usando controle *Fuzzy*-PID FGS

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0337	0,0166	0,2347	0,2276
Real	0,2 m	0,0529	0,0489	0,4181	0,9532

Fonte: Elaborado pelo próprio autor.

Já Figura 53 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição médio foi de 0,02 centímetros; enquanto nos ensaios práticos o erro médio foi de 2,5 centímetros, sendo 2,48 centímetros maior que o esperado.

**Figura 53** – Comparação entre os resultados ante uma rampa usando controlador *Fuzzy*-PID FGS



Fonte: elaborado pelo próprio autor.

De acordo com a avaliação de desempenho por critérios integrais mostrada na Tabela 19 e com a Figura 53, fica evidente que o teste no sistema real teve um desempenho abaixo do esperado quando comparado com a simulação que obteve índices de desempenho bastante superiores, com destaque para os critérios ISE e ITAE.

**Tabela 19** – Critérios integrais para uma rampa usando controle *Fuzzy*-PID FGS

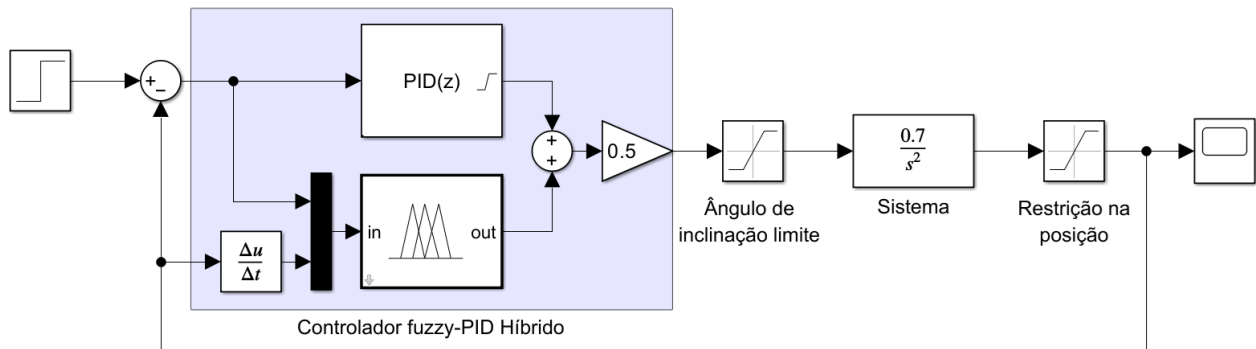
Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0001	0,0002	0,0249	0,0871
Real	0 a 0,4 m	0,0200	0,2311	0,5045	5,5070

Fonte: Elaborado pelo próprio autor.

#### 4.5 Controle *fuzzy*-PID (híbrido)

O controlador *Fuzzy*-PID é mostrado na Figura 54, o qual utiliza uma topografia Híbrida, onde a ação de controle é calculada como a média da soma das ações de controle PID e *Fuzzy*.

**Figura 54** – Diagrama em blocos do controle *Fuzzy*-PID Híbrido



**Fonte:** elaborado pelo próprio autor.

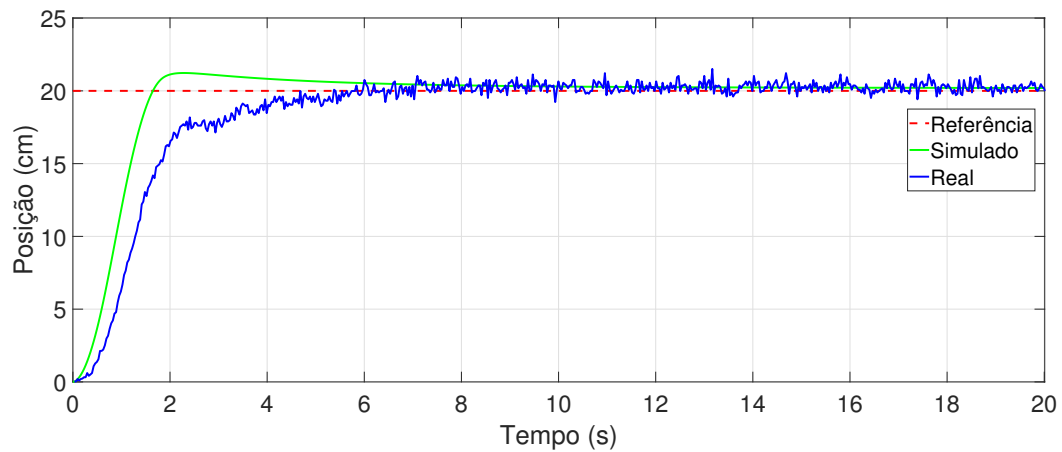
A Figura 55 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo degrau. O sistema simulado obteve um desempenho levemente melhor que o prático, como mostrado na Tabela 20.

**Tabela 20** – Especificações da resposta transitória para uma referência degrau de 20 usando controle *Fuzzy*-PID Híbrido

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	1,70	5,40	+3,70
Máximo sobressinal (%)	5,00	3,75	-1,25
Tempo de pico (s)	2,00	7,00	+5,00
Tempo de assentamento(s)	5,6	5,9	0,3

Fonte: Elaborado pelo próprio autor.

**Figura 55** – Comparação entre os resultados ante um degrau usando controle *Fuzzy*-PID Híbrido



**Fonte:** elaborado pelo próprio autor.

Já se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados na Tabela 21, onde novamente em todos os critérios o sistema simulado se saiu melhor que o sistema real. O principal motivo foi que o sistema real demora um pouco mais para se estabilizar, fazendo com que eu mesmo seja bastante penalizado pelos critérios.

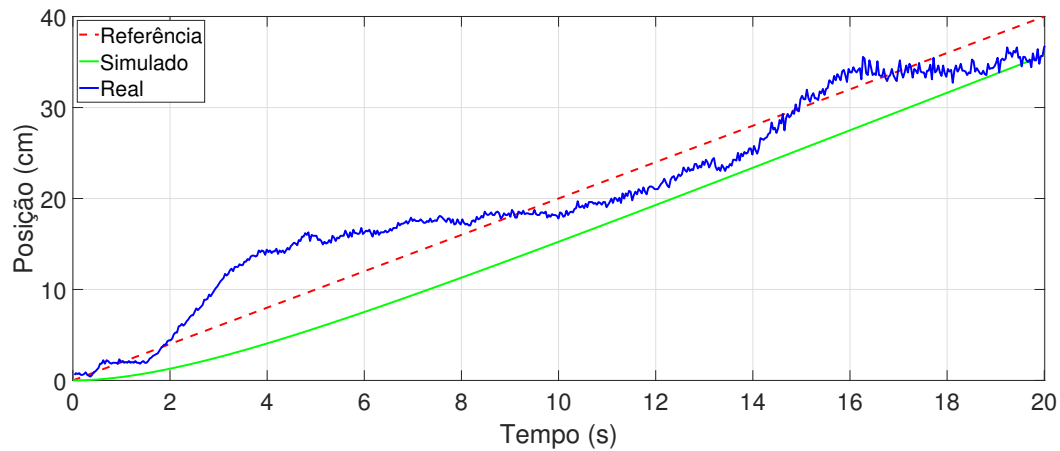
**Tabela 21** – Critérios integrais para um degrau usando controle *Fuzzy* PID Híbrido

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0270	0,0133	0,2233	0,6879
Real	0,2 m	0,0414	0,0300	0,3577	0,9624

Fonte: Elaborado pelo próprio autor.

Já Figura 56 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição médio foi de 2 centímetros; enquanto nos ensaios práticos, o erro médio foi de 2,36 centímetros, sendo 0,36 centímetros maior que o esperado.

**Figura 56** – Comparação entre os resultados ante uma rampa usando controlador *Fuzzy*-PID Híbrido



**Fonte:** elaborado pelo próprio autor.

Já para avaliação de desempenho por critérios integrais mostrados na Tabela 22, mesmo com o erro oscilando mais do que o esperado, o teste se mostrou superior à simulação em todos os aspectos, de acordo com os critérios integrais avaliados.

**Tabela 22** – Critérios integrais para uma rampa usando controle *Fuzzy*-PID Híbrido

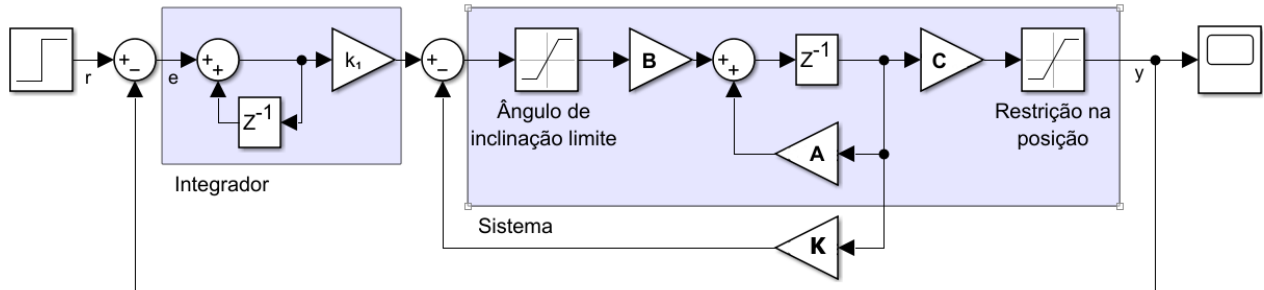
Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0359	0,4021	0,8241	8,9380
Real	0 a 0,4 m	0,0177	0,1240	0,4724	4,1120

Fonte: Elaborado pelo próprio autor.

#### 4.6 Controle Por Realimentação De Estados

O controle por realimentação de estados foi implementado em um sistema digital, utilizando o modelo do sistema no espaço de estados com os ganhos  $\mathbf{K} = (7,7638 \ 7,8835 \ 1,0000)$  e  $k_1 = 2,0159$  obtidos com auxílio computacional. A Figura 57 mostra o diagrama em blocos do sistema simulado no *Simulink*.

**Figura 57** – Diagrama em blocos do controle por realimentação de estados



**Fonte:** elaborado pelo próprio autor.

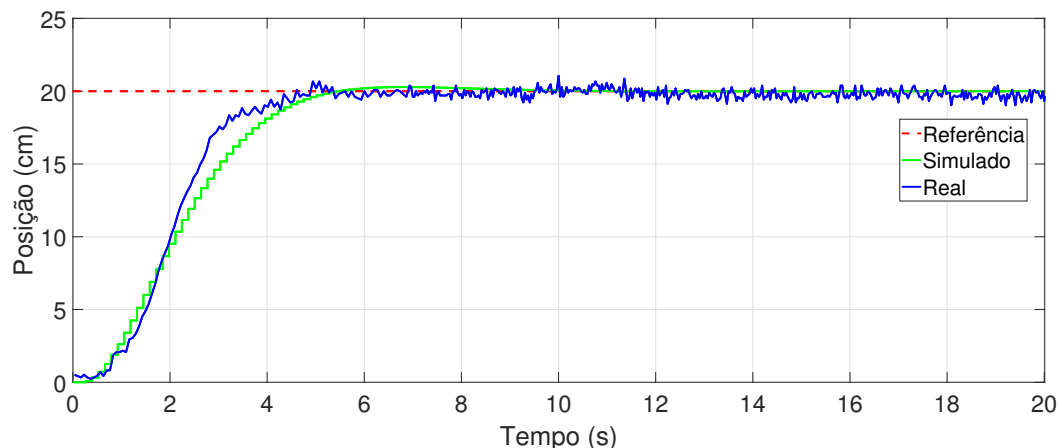
A Figura 58 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo degrau. O sistema simulado obteve um desempenho levemente pior que na prática, mais detalhes podem ser vistos na Tabela 23.

**Tabela 23** – Especificações da resposta transitória para uma referência degrau de 20 cm usando controle por realimentação de estados

Parâmetro	Simulação	Prático	Diferença
Tempo de subida (s)	5,02	4,90	-0,12
Máximo sobressinal (%)	3,00	3,37	+0,37
Tempo de pico (s)	6,70	4,95	-1,75
Tempo de assentamento(s)	5,6	5	-0,6

Fonte: Elaborado pelo próprio autor.

**Figura 58** – Comparação entre os resultados ante um degrau usando controle por realimentação de estados



**Fonte:** elaborado pelo próprio autor.

Já se tratando da avaliação de desempenho por critérios integrais, os resultados são mostrados na Tabela 24, onde, nos critérios ISE, ITSE e IAE o sistema real apresentou melhor desempenho que a simulação, sendo inferior somente no critério ITAE. O motivo mais provável do pior desempenho obtido no critério ITAE é o desvio padrão na leitura do sensor de distância, que faz com que o índice aumente conforme o tempo avança.

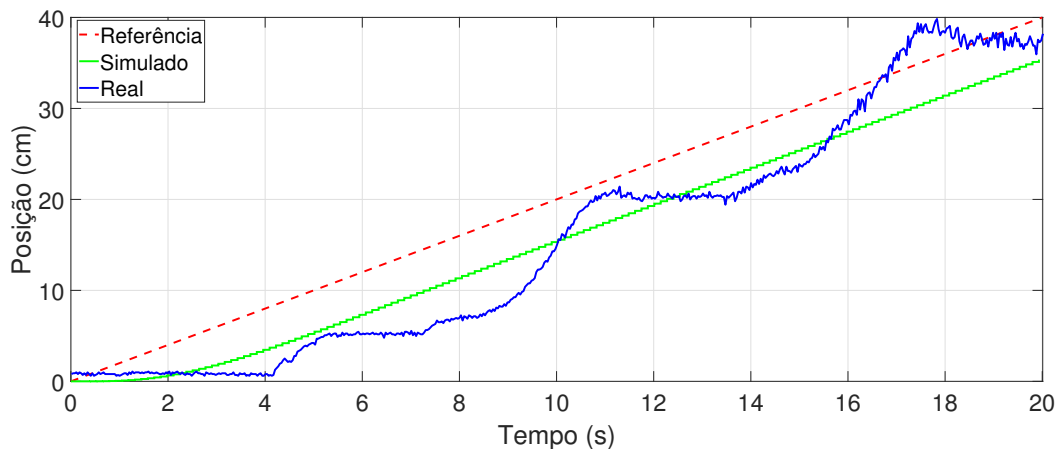
**Tabela 24** – Critérios integrais para um degrau usando controle por realimentação de estados

Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0,2 m	0,0652	0,0668	0,4660	0,7247
Real	0,2 m	0,0641	0,0628	0,4636	1,0650

Fonte: Elaborado pelo próprio autor.

A Figura 59 mostra a comparação entre os resultados simulados e práticos do sistema para um sinal do tipo rampa. Nas simulações, o erro de posição médio foi de 4,75 centímetros; enquanto nos ensaios práticos o erro médio foi de 4,38 centímetros, sendo 0,37 centímetros menor do que o esperado.

**Figura 59** – Comparação entre os resultados ante uma rampa usando controle por realimentação de estados



Fonte: elaborado pelo próprio autor.

Mesmo o sistema real apresentando mais oscilações que nas simulações, para a avaliação de desempenho por critérios integrais (Tabela 25) ambas as respostas obtiveram índices de desempenho próximos, mas com a simulação se saindo um pouco melhor nos critérios ISE, ITSE e IAE. E o teste no sistema real se saiu melhor com o índice de desempenho ITAE.

**Tabela 25** – Critérios integrais para uma rampa usando controle por realimentação de estados

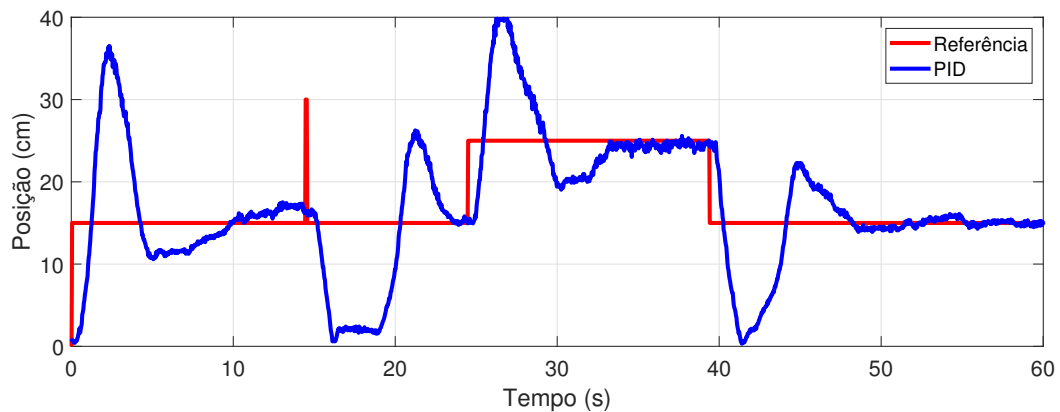
Saída	Referência	ISE	ITSE	IAE	ITAE
Simulada	0 a 0,4 m	0,0387	0,4234	0,8616	9,1870
Real	0 a 0,4 m	0,0521	0,4528	0,8761	8,0550

Fonte: Elaborado pelo próprio autor.

#### 4.7 Comportamento Do Sistema Em Testes Complexos

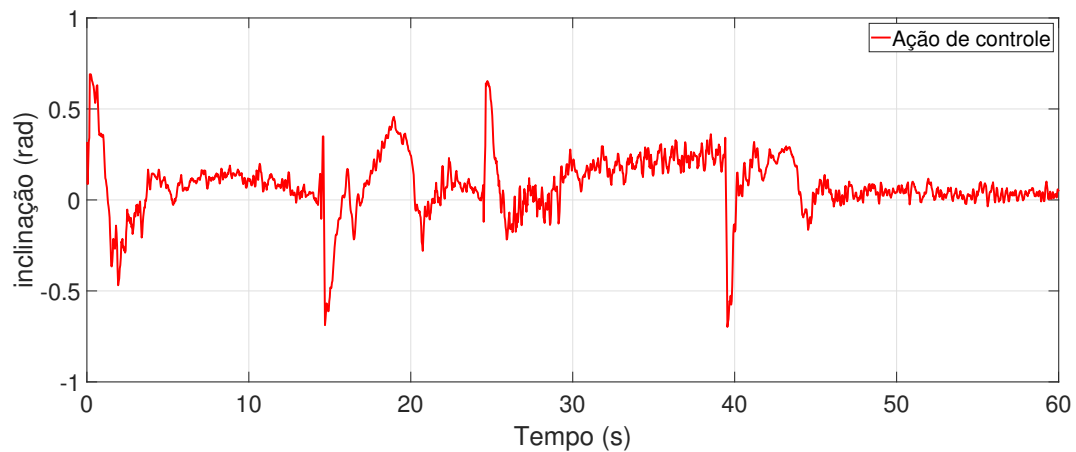
Para avaliar melhor como o sistema se comporta com cada um dos controladores, fez-se necessário um terceiro teste que incluía mudanças de referência, incluindo uma mudança mais rápida que se aproxima de um impulso. Esse teste visa simular o funcionamento em condições não ideais para avaliar melhor os controladores.

Começando com o controlador PID, que não lida muito bem quando há variação na referência (Figura 60), além de apresentar uma grande resposta ao distúrbio que causa uma grande ação de controle (ver Figura 61), causando uma variação significativa na posição, o que faz com que o sistema se afaste da referência por um período considerável antes de recuperar o controle. Esses comportamentos indicam que o PID, embora eficiente em condições estáveis, tem dificuldades em lidar com mudanças bruscas e perturbações externas, exigindo bastante do atuador nesses momentos.

**Figura 60** – Teste complexo com controle PID

Fonte: elaborado pelo próprio autor.

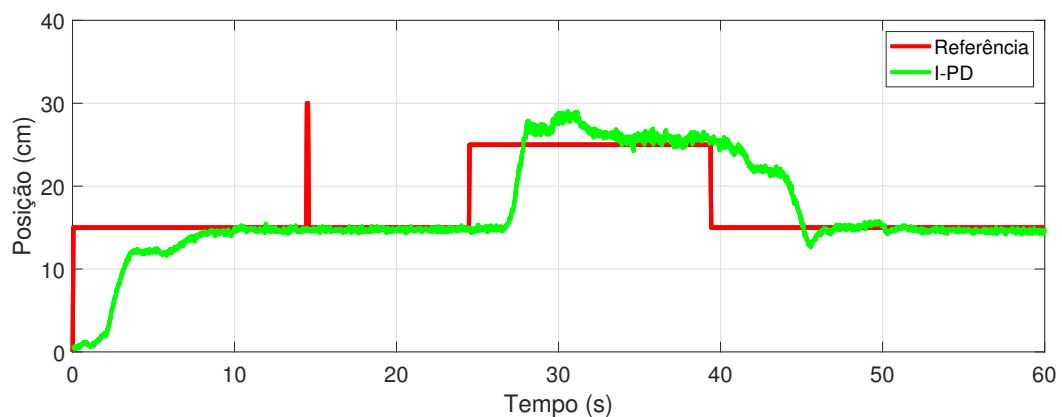
**Figura 61** – Ações de controle para o teste com controle PID



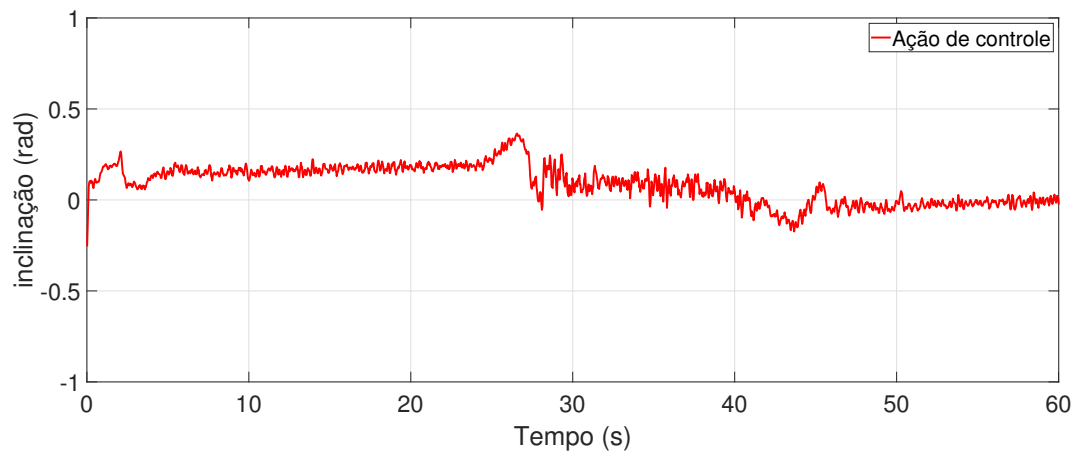
**Fonte:** elaborado pelo próprio autor.

O controlador I-PD (Ver Figura 62), apesar da resposta mais lenta comparada ao PID, consegue seguir a referência de forma suave e com poucas mudanças na ação de controle (Ver Figura 63), além de não ser afetado pelo distúrbio, o que já era de se esperar, uma vez que a parte derivativa desse controlador recebe a posição da bola, diferente do PID que recebe a referência. Essa característica contribui para uma resposta mais estável e controlada, evitando variações bruscas e melhorando o desempenho do sistema em cenários onde a rejeição de distúrbios é essencial, além de estender a vida útil do atuador por ter variações de menor intensidade nas ações de controle.

**Figura 62** – Teste complexo com controle I-PD

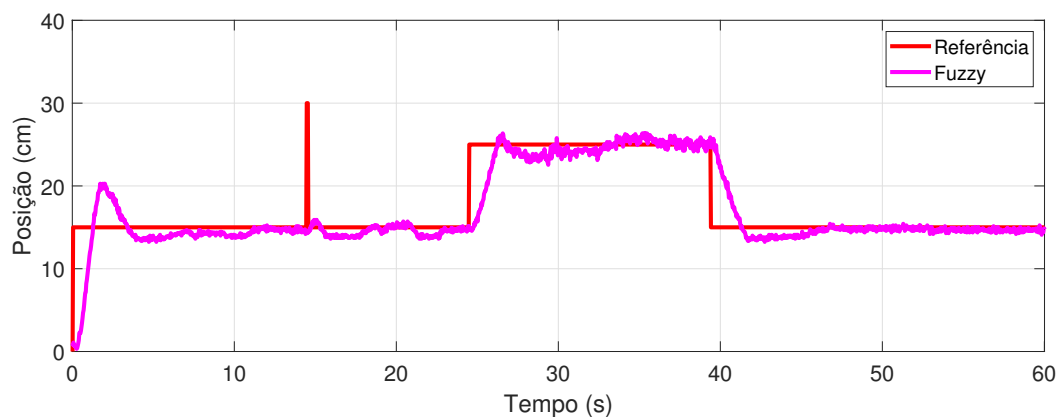


**Fonte:** elaborado pelo próprio autor.

**Figura 63** – Ações de controle para o teste com controle I-PD

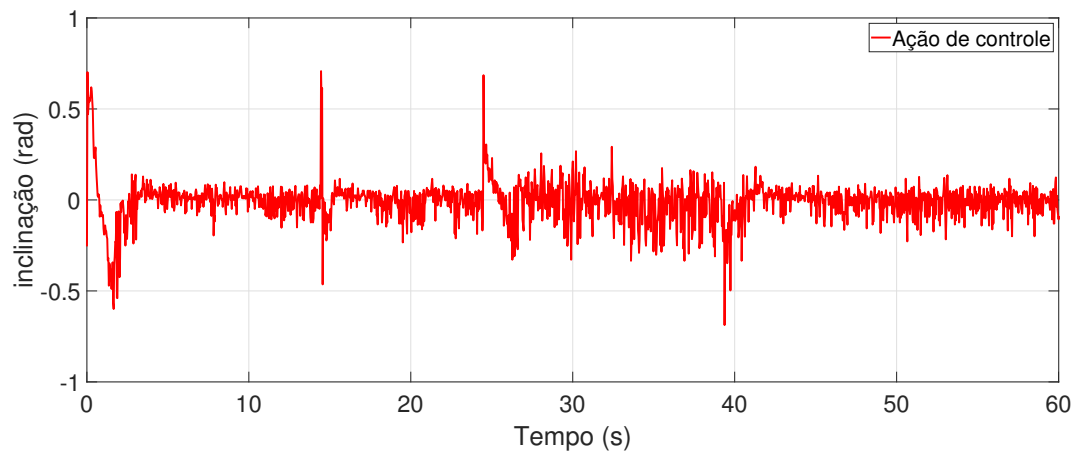
**Fonte:** elaborado pelo próprio autor.

O controlador *Fuzzy* por sua vez, conseguiu seguir a referência de forma satisfatória durante os degraus, e foi pouco afetado pelo distúrbio na referência, apresentando somente um pequeno sobressinal, como mostrado na Figura 64. O controlador também seguiu de forma satisfatória os outros degraus e se mostrou uma alternativa mais rápida e estável do que os controladores anteriores. Se tratando das ações de controle, durante os degraus as ações de controle são mais consistentes, mas durante o estado estacionário a ação de controle varia bastante devido à imprecisão na leitura do sensor (ver Figura 65), o que pode levar ao desgaste mecânico prematuro do atuador.

**Figura 64** – Teste complexo com controle *fuzzy*

**Fonte:** elaborado pelo próprio autor.

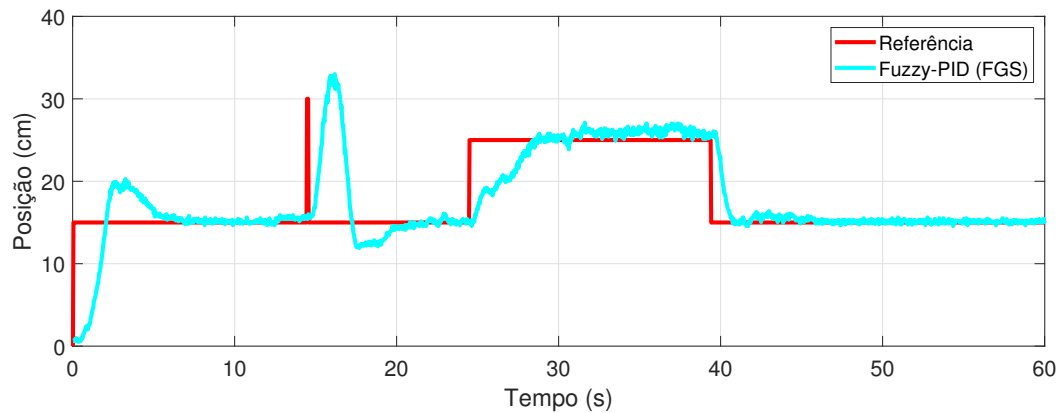
**Figura 65** – Ações de controle para o teste com controle *fuzzy*



**Fonte:** elaborado pelo próprio autor.

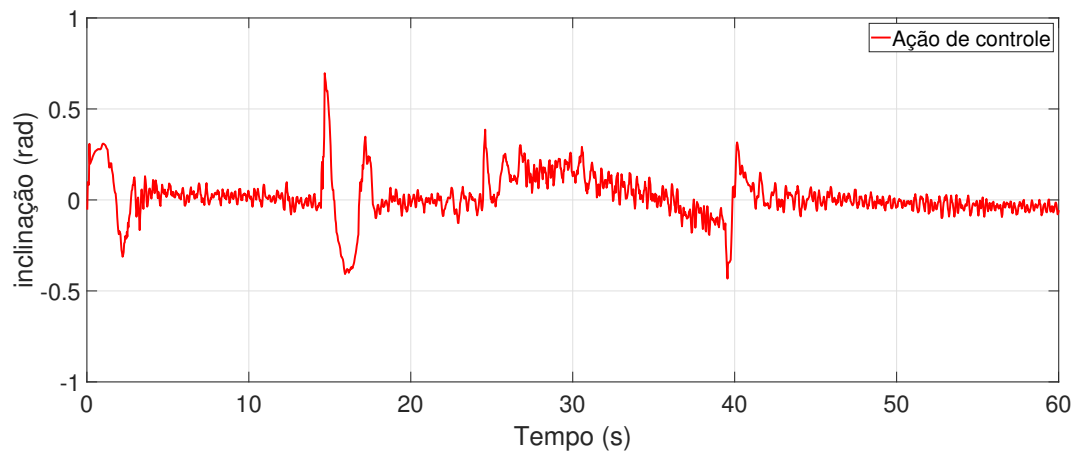
Já o controlador *Fuzzy*-PID (FGS) se mostrou superior aos controladores *Fuzzy* e PID para as variações na referência, principalmente para o segundo e o terceiro degrau, mas herdou o problema com distúrbio na referência do controlador PID, que fez com que o sistema apresentasse um grande sobressinal, como mostrado na Figura 66. Já se tratando de ações de controle (ver Figura 67), mesmo em degraus ou em regime estacionário, o sistema se saiu melhor que os controladores *Fuzzy* e PID, tendo ações de controle de menor intensidade em ambos os casos.

**Figura 66** – Teste complexo com controle *Fuzzy*-PID (FGS)



**Fonte:** elaborado pelo próprio autor.

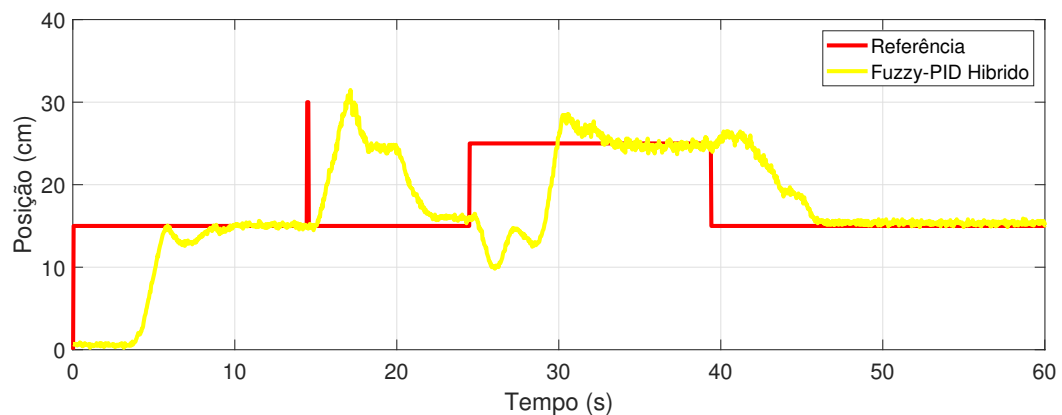
**Figura 67** – Ações de controle para o teste com controle *fuzzy*-PID (FGS)



**Fonte:** elaborado pelo próprio autor.

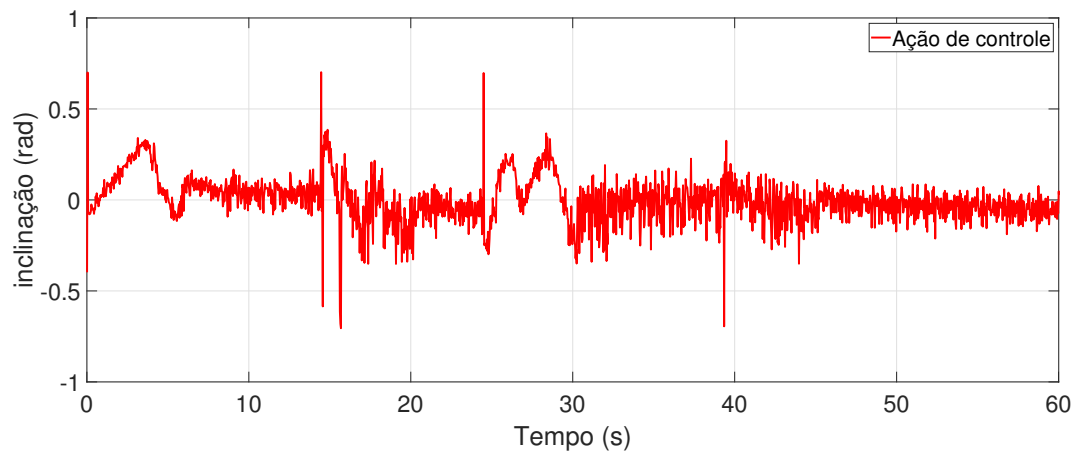
O controlador *Fuzzy*-PID Híbrido, assim como o seu antecessor, mostrou-se superior aos controladores *Fuzzy* e PID para as variações na referência, e novamente apresentou um grande sobressinal para o distúrbio, como mostrado na Figura 68. Porém, como mostrado na Figura 69 o controlador herdou o problema do excesso de variações na ação de controle quando o sistema se encontra em regime estacionário, o que pode causar desgaste prematuro do atuador.

**Figura 68** – Teste complexo com controle *fuzzy*-PID Híbrido



**Fonte:** elaborado pelo próprio autor.

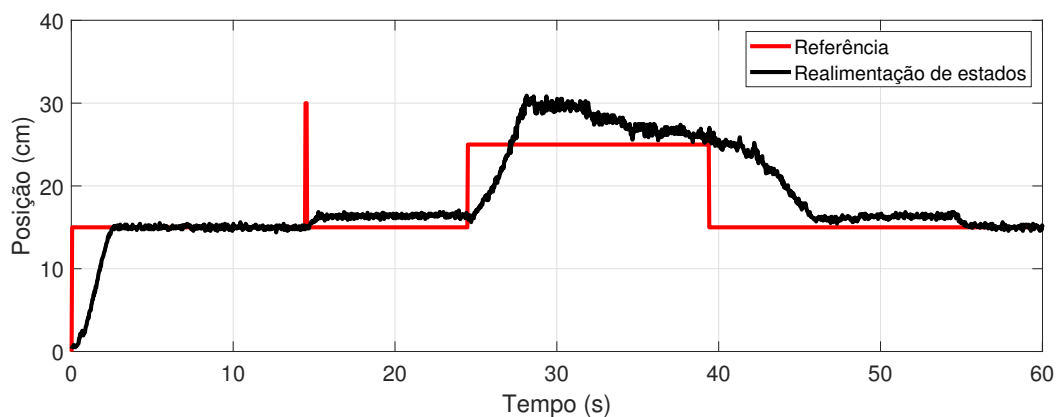
**Figura 69** – Ações de controle para o teste com controle *fuzzy*-PID Híbrido



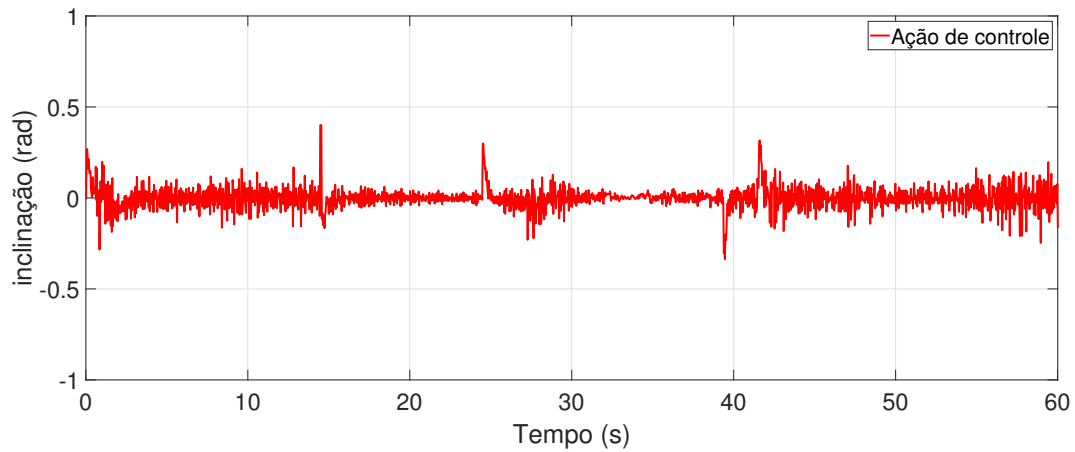
**Fonte:** elaborado pelo próprio autor.

E por fim, o controle por realimentação de estados apresentou um bom desempenho seguindo a referência, assim como o controlador *fuzzy*, apresentou um pequeno sobressinal após o distúrbio, como mostrado na Figura 70. Para os degraus posteriores, o sistema aparentou ter desempenho inferior ao primeiro. O motivo mais provável é a existência prévia de valores no integrador do controlador. Se tratando de ações de controle, esse foi o controlador com as ações menos intensas para um degrau, porém como ocorreu com o controlador *fuzzy*, também apresentou oscilações (de menor amplitude) durante o estado estacionário devido à imprecisão na leitura do sensor.

**Figura 70** – Teste complexo com controle por realimentação de estados



**Fonte:** elaborado pelo próprio autor.

**Figura 71** – Ações de controle para o teste com controle por realimentação de estados

**Fonte:** elaborado pelo próprio autor.

#### 4.8 Comparação Dos Métodos De Controle

Nessa seção serão comparadas as respostas do sistema e os critérios integrais com a intenção de descobrir qual técnica de controle melhor se adaptou ao sistema.

**Resultados obtidos ante um degrau:** A Tabela 26 apresenta os valores de máximo sobressinal obtidos pelos diferentes controladores quando aplicados a uma entrada do tipo degrau. Na mesma, é possível observar que o controlador I-PD apresentou o melhor desempenho, com um sobressinal de apenas 0,75%, seguido pelo controlador por realimentação de estados e *fuzzy*-PID Híbrido, que obtiveram 3% e 5% de máximo sobressinal, respectivamente.

**Tabela 26** – Máximo Sobressinal - ensaio simulado com entrada degrau

Controlador	Máximo Sobressinal (%)
I-PD	0,75
Realimentação de Estados	3,00
Fuzzy-PID Híbrido	5,00
Fuzzy-PID FGS	6,50
Fuzzy	44,30
PID	55,30

Fonte: Elaborado pelo próprio autor.

Já se tratando dos critérios integrais, mostrados na Tabela 27, para o critério ISE, o controlador com melhor resultado foi o *fuzzy*-PID Híbrido, seguido pelo *fuzzy*-PID (FGS) e *fuzzy*, e por último ficaram os controladores PID, realimentação de estados e I-PD. No que diz respeito ao critério

ITSE, novamente o controlador com melhor resultado foi o *fuzzy*-PID Híbrido, seguido pelo *fuzzy*-PID (FGS), *fuzzy* e por último ficaram os controladores realimentação de estados, PID e I-PD.

Para o critério IAE, o melhor resultado pertenceu ao controlador *fuzzy*-PID Híbrido, seguido novamente pelo *fuzzy*-PID (FGS) e *fuzzy* e mais uma vez por último ficaram os controladores por realimentação de estados, PID e I-PD. No critério ITAE a situação muda um pouco com melhor resultado ficando com o controle *fuzzy*-PID (FGS), seguido pelo *fuzzy*-PID Híbrido e pela realimentação de estados, e por último ficam os controladores *fuzzym* I-PD e PID.

**Tabela 27** – Critérios integrais - ensaio simulado com entrada degrau

Saída	Referência	ISE	ITSE	IAE	ITAE
PID	0,2 m	0,0475	0,0787	0,5028	1,598
I-PD	0,2 m	0,0774	0,0909	0,5599	1,201
<i>fuzzy</i>	0,2 m	0,0338	0,0389	0,369	0,9146
Realimentação de Estados	0,2 m	0,0652	0,0668	0,4660	0,7247
<i>Fuzzy</i> -PID (FGS)	0,2 m	0,0337	0,0166	0,2347	0,2276
<i>Fuzzy</i> -PID (Híbrido)	0,2 m	0,0270	0,0133	0,2233	0,6879

Fonte: Elaborado pelo próprio autor.

Os ensaios práticos do sistema para um sinal do tipo degrau foram obtidas respostas levemente diferentes. A Tabela 28 apresenta os valores de máximo sobressinal obtidos para diferentes controladores. O controlador I-PD, assim como nas simulações, demonstrou o melhor desempenho, registrando um sobressinal de 3,25%, seguido pelo controlador por realimentação de estados e *fuzzy*-PID Híbrido, que obtiveram valores ligeiramente superiores, de 3,37% e 8,5%, respectivamente.

**Tabela 28** – Máximo Sobressinal - ensaio prático com entrada degrau

Controlador	Máximo Sobressinal (%)
I-PD	3,25
Realimentação de Estados	3,37
<i>Fuzzy</i> -PID Híbrido	3,75
<i>Fuzzy</i> -PID FGS	8,50
<i>Fuzzy</i>	56,40
PID	63,40

Fonte: Elaborado pelo próprio autor.

Já se tratando dos critérios integrais obtidos nos ensaios práticos, os valores obtidos são mostrados na Tabela 29, se tratando do critério ISE, o controlador com melhor resultado foi o *fuzzy*, seguido pelo *fuzzy*-PID Híbrido e PID, e por último ficaram os controladores *fuzzy*-PID (FGS),

realimentação de estados e I-PD. Para o critério ITSE, o controlador com melhor resultado foi o *fuzzy*-PID Híbrido, seguido pelo *fuzzy* e *fuzzy*-PID (FGS), e por último ficaram os controladores PID, realimentação de estados e I-PD.

No critérios IAE o melhor resultado fica com o controle *fuzzy*-PID Híbrido, seguido pelo *fuzzy* e *fuzzy*-PID (FGS), por último ficam os controladores PID, realimentação de estados e I-PD. Para o critério ITAE o melhor resultado pertenceu ao controlador *fuzzy*-PID (FGS), seguido pelo *fuzzy*-PID Híbrido, realimentação de estados, e por último ficaram os controladores *fuzzy*, PID e I-PD.

**Tabela 29** – Critérios integrais - ensaio prático com entrada degrau

Saída	Referência	ISE	ITSE	IAE	ITAE
PID	0,2 m	0,0432	0,0523	0,4356	1,697
I-PD	0,2 m	0,0979	0,1502	0,7389	2,366
<i>Fuzzy</i>	0,2 m	0,0365	0,0334	0,3638	1,254
Realimentação de Estados	0,2 m	0,0641	0,0628	0,4636	1,065
<i>Fuzzy</i> -PID (FGS)	0,2 m	0,0529	0,0489	0,4181	0,9532
<i>Fuzzy</i> -PID (Híbrido)	0,2 m	0,0414	0,0300	0,3577	0,9624

Fonte: Elaborado pelo próprio autor.

**Resultados obtidos ante uma rampa:** Nos resultados das simulações, o controlador com menor erro de estado estacionário foi o *fuzzy*-PID (FGS), com um valor de 0,02 cm, seguido pelo PID com 0,15 cm e pelo *fuzzy* com 1,1 cm. Em seguida, aparecem o *fuzzy*-PID Híbrido com 2 cm, a realimentação de estados com 4,75 cm e, por último, o I-PD com 5,54 cm, que apresentou o maior erro entre os controladores avaliados.

Já se tratando dos critérios integrais obtidos nas simulações, os valores obtidos são mostrados na Tabela 30. Para o critério ISE, o controlador com melhor desempenho foi o *fuzzy*-PID FGS, seguido pelo PID e *fuzzy*. Os controladores *fuzzy*-PID Híbrido, realimentação de estados e I-PD apresentaram os piores desempenhos, respectivamente. Para o critério ITSE, novamente o *fuzzy*-PID (FGS) se destacou como o melhor, seguido pelo PID e *fuzzy*. Os demais controladores apresentaram desempenhos inferiores.

No critério IAE, o *fuzzy*-PID FGS obteve o melhor resultado, seguido pelo PID e *fuzzy*. Os controladores *fuzzy*-PID Híbrido, realimentação de estados e I-PD apresentaram os piores desempenhos, respectivamente. Por fim, para o critério ITAE, o controlador *fuzzy*-PID (FGS) manteve o melhor desempenho, seguido pelo PID e *fuzzy*. seguido pelos controladores *fuzzy*-PID Híbrido, realimentação de estados e I-PD.

**Tabela 30** – Critérios integrais obtidos a partir de simulações

Saída	Referência	ISE	ITSE	IAE	ITAE
PID	0,2 m	0,0002	0,0005	0,0356	0,1306
I-PD	0,2 m	0,0528	0,5994	1,002	10,92
<i>fuzzy</i>	0,2 m	0,0023	0,0205	0,2081	2,011
Realimentação de Estados	0,2 m	0,0387	0,4234	0,8616	9,187
<i>Fuzzy</i> -PID (FGS)	0,2 m	0,0001	0,0002	0,0249	0,0871
<i>Fuzzy</i> -PID (Híbrido)	0,2 m	0,0359	0,4021	0,8241	8,938

Fonte: Elaborado pelo próprio autor.

Nos testes práticos, o controlador com melhor desempenho foi o PID, com um erro de 1,14 cm, seguido pelo *fuzzy* com 1,6 cm e pelo *fuzzy*-PID Híbrido com 2,36 cm. O *fuzzy*-PID (FGS) apresentou um erro de 2,5 cm, enquanto a realimentação de estados obteve 4,38 cm. O pior desempenho foi registrado pelo I-PD, que apresentou um erro de 5,4 cm, confirmando a maior discrepância em relação aos demais controladores.

Já se tratando dos critérios integrais obtidos nos ensaios práticos, os valores obtidos são mostrados na Tabela 31 para critério ISE, o controlador *fuzzy* obteve o melhor desempenho, seguido pelo PID, *fuzzy*-PID Híbrido. Os controladores *fuzzy*-PID (FGS), realimentação de estados e I-PD foram os piores para esse teste. Para o critério ITSE, o *fuzzy* também apresentou o menor valor, seguido pelo *fuzzy*-PID Híbrido e pelo PID. Já os controladores *fuzzy*-PID (FGS), realimentação de estados e I-PD tiveram desempenhos inferiores.

Analisando o critério IAE, o melhor resultado foi obtido pelo *fuzzy*, seguido pelo *fuzzy*-PID Híbrido e *fuzzy*-PID (FGS). Os demais controladores apresentaram valores superiores. Por fim, para o critério ITAE, o *fuzzy*-PID Híbrido apresentou o menor valor, seguido pelo PID e *fuzzy*-PID (FGS). E por último, ficaram os controladores por realimentação de estados e I-PD.

**Tabela 31** – Critérios integrais obtidos a partir de experimentos

Saída	Referência	ISE	ITSE	IAE	ITAE
PID	0,2 m	0,0114	0,1576	0,3754	4,597
I-PD	0,2 m	0,0756	0,8334	1,174	12,53
<i>fuzzy</i>	0,2 m	0,0060	0,0525	0,2805	2,625
Realimentação de Estados	0,2 m	0,0521	0,4528	0,8761	8,0550
<i>Fuzzy</i> -PID (FGS)	0,2 m	0,0200	0,2311	0,5045	5,5070
<i>Fuzzy</i> -PID (Híbrido)	0,2 m	0,0177	0,1240	0,4724	4,1120

Fonte: Elaborado pelo próprio autor.

## 5 CONCLUSÃO

No presente trabalho foi desenvolvido um protótipo de baixo custo de um sistema barra-bola, com o intuito de estudar a viabilidade de diversas técnicas de controle na estabilização da bola numa posição determinada. Para isto, os resultados foram analisados por meio de parâmetros da resposta temporal, assim como usando critérios integrais, para entradas do tipo degrau e rampa.

Os controladores analisados incluíram: PID, I-PD (uma modificação do PID), *Fuzzy*, *Fuzzy*-PID híbrido, *Fuzzy*-PID (FGS) e o controle por realimentação de estados, cada um com suas particularidades e diferentes abordagens para lidar com as dinâmicas do sistema.

Vale destacar que o PID, apesar de sua simplicidade e facilidade de sintonia, apresentou dificuldades com ruídos no sistema, em especial quando o ganho derivativo é grande, fazendo com que o sistema se torne instável com  $K_d > 2$ . Para um sinal do tipo degrau, o sistema apresentou sobressinal elevado, mas para uma rampa o sistema seguiu a referência de forma satisfatória.

Por sua parte, o controlador I-PD herdou as características do PID, incluindo o problema com o ganho derivativo, mas demonstrou ter uma resposta mais suave a um degrau quando comparado a outros controladores; mas para uma rampa, o sistema possui um dos maiores erros de estado estacionário entre os controladores testados.

Por outro lado, o controlador *Fuzzy*, por sua vez, demonstrou uma melhor estabilidade com ruídos do sistema e manteve uma resposta ante um degrau semelhante ao PID, e também, para uma rampa, apresentou o segundo menor erro de estado estacionário, perdendo apenas para o controlador PID.

O controlador *fuzzy*-PID (FGS) combinou as vantagens de ambas as abordagens, permitindo um controle mais eficiente ao adaptar dinamicamente os ganhos do sistema, resultando em um pequeno sobressinal para um degrau e erro de estado estacionário pequeno para uma rampa.

Já o controlador *Fuzzy*-PID híbrido combinou os dois controladores, resultando em uma resposta mais rápida e sem sobressinal para um degrau, mas para uma rampa o sistema foi penalizado com um erro de estado estacionário um pouco maior.

E por fim, o controlador por realimentação de estados obteve uma resposta com sobressinal mínimo para um degrau, porém, para uma rampa teve desempenho similar ao controlador I-PD.

Com base nos critérios integrais analisados, verificou-se que a escolha do controlador ideal depende da aplicação e das exigências de desempenho do sistema. O controlador *Fuzzy*-PID Híbrido apresentou o desempenho mais consistente em termos de robustez e resposta transitória, enquanto o *Fuzzy* se mostrou promissor, em especial, para distúrbios.

Diante dos resultados obtidos, conclui-se que os controladores baseados em lógica *fuzzy* oferecem uma solução promissora para sistemas dinâmicos, especialmente quando combinados com estratégias híbridas.

## REFERÊNCIAS

- ALVES, F. D. da S. *et al.* Projeto e implementação de um controlador PID digital para um sistema calha e bola. *In: ENCONTRO DE COMPUTAÇÃO DO OESTE POTIGUAR (ECOP)*, 4., 2020, Mossoró. **Anais [...]**. Mossoró: Ufersa, 2020. v. 1, n. 4. Disponível em: <http://periodicos.ufersa.edu.br/ecop/article/view/10121>. Acesso em: 09 novembro 2023.
- AZEVEDO, L. H. da S. P. Projeto de controladores digitais para um protótipo do sistema barra-bola. *In: SEMINÁRIOS DE INICIAÇÃO CIENTÍFICA*, 24., 2020, Feira de Santana. **Anais [...]**. Feira de Santana: UEFS, 2020. Disponível em: <https://periodicos.uefs.br/index.php/semic/article/view/6876>. Acesso em: 17 outubro 2023.
- BANZI, M.; SAFARI, A. O. M. C. **Getting started with Arduino**. 2. ed. [S.l.]: Maker Media, 2011.
- CAMARGO, C. V. d. **Projeto e implementação de um aparato para estudo e pesquisa em controle com base no sistema bola e barra**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica) — Universidade Tecnológica Federal do Paraná, Toledo, 2018 . Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/23875>. Acesso em: 15 outubro 2023.
- LATIF, S.; MUHAMMAD, E.; NAEEM, U. Implementation of ball and beam system using classical and advanced control techniques. *In: INTERNATIONAL CONFERENCE ON APPLIED AND ENGINEERING MATHEMATICS*, 2019. **Anais [...]**. IEEE, 2019. p. 74–79. Disponível em: <https://ieeexplore.ieee.org/document/8853822>. Acesso em: 12 março 2024.
- LATHROP, R. C.; DUNSTAN, G. The synthesis of "optimum" transient response: criteria and standard forms. **Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry**, v. 72, n. 5, p. 273–288, 1953. Disponível em: <http://ieeexplore.ieee.org/document/6371346>. Acesso em: 10 março 2024.
- MARRO, A. A. *et al.* Lógica fuzzy: conceitos e aplicações. **Research Gate**, v. 2, 2010. Disponível em: <https://www.researchgate.net/publication/228595876>. Acesso em: 12 março 2024.
- MOEZZI, R.; VU, T. M.; TAMRE, M. Fuzzy logic control for a Ball and Beam system. **International Journal of Innovative Technology and Interdisciplinary Sciences**, v. 1, n. 1, p. 39–48, 2018. Disponível em: <https://journals.tultech.eu/index.php/ijitis/article/view/17>. Acesso em: 12 março 2024.
- MUFTAH, M. N. *et al.* Intelligent position control for intelligent pneumatic actuator with ball-beam (IPABB) system. **Applied Sciences**, v. 12, n. 21, p. 11089, 2022. Disponível em: <https://www.mdpi.com/2076-3417/12/21/11089>. Acesso em: 20 novembro 2023.
- OGATA, K. **Engenharia de Controle Moderno** 5. ed. [S.l.]: Pearson Prentice Hall, 2011.

PRUNZEL, A. H. **Estudo comparativo de métodos avançados de controle implementados no sistema bola e barra**. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica) — Universidade Tecnológica Federal do Paraná, Toledo, 2019. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/24599>. Acesso em: 25 outubro 2023.

RAHMAT, M. F.; WAHID, H.; WAHAB, N. A. Application of intelligent controller in a ball and beam control system. **International Journal on Smart Sensing and Intelligent Systems**, v. 3, n. 1, p. 45–60, 2010. Disponível em: <https://www.researchgate.net/publication/228824245>. Acesso em: 25 outubro 2023.

SALEEM, M. K. *et al.* Design and implementation of adaptive neuro-fuzzy inference system for the control of an uncertain ball and beam apparatus. **Mehran University Research Journal of Engineering & Technology**, v. 41, n. 2, p. 178–184, 2022. Disponível em: <https://publications.muet.edu.pk/index.php/muetrj/article/view/2450>. Acesso em: 03 fevereiro 2025.

VALENTINE, D.; HAHN, B. **Essential MATLAB for Engineers and Scientists**. [S.l.]: Academic Press, 2022.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **Transactions of the ASME**, v. 64, n. 8, p. 759–765, 1942. Disponível em: [https://davidr.no/iiaav3017/papers/Ziegler\\_Nichols\\_%201942.pdf](https://davidr.no/iiaav3017/papers/Ziegler_Nichols_%201942.pdf). Acesso em: 10 fevereiro 2025.