

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO  
CAMPUS SÃO PAULO  
CURSO DE GRADUAÇÃO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

GABRIEL PEREIRA CAMPANHA

**DESENVOLVIMENTO DE UM CONTROLADOR PARA UMA PLANTA BARRA-BOLA**

SÃO PAULO – SP

2025

GABRIEL PEREIRA CAMPANHA

**DESENVOLVIMENTO DE UM CONTROLADOR PARA UMA PLANTA BARRA-BOLA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação de Engenharia de Controle e Automação do Instituto Federal de São Paulo, como requisito parcial à obtenção do grau de bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Alexandre Brincalepe

Orientador: Prof. \_\_\_\_\_

Co-orientador: Prof. \_\_\_\_\_

Orientador: Prof. \_\_\_\_\_

Co-orientador: Prof. \_\_\_\_\_

SÃO PAULO – SP

2025

GABRIEL PEREIRA CAMPANHA

**DESENVOLVIMENTO DE UM CONTROLADOR PARA UMA PLANTA BARRA-BOLA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação de Engenharia de Controle e Automação do Instituto Federal de São Paulo, como requisito parcial à obtenção do grau de bacharel em Engenharia de Controle e Automação.

Aprovado em:

**Banca Examinadora**

---

Dr. Prof. Alexandre Brincalepe Campo (Orientador)  
Instituto Federal de São Paulo (IFSP)

---

Dr. Prof. Haroldo Issao Guibu  
Universidade de São Paulo (USP)

---

Dr. Prof. Eduardo Alves da Costa  
Instituto Federal de São Paulo (IFSP)

## RESUMO

A automação é uma área que procura gerir e ajustar variáveis físicas ao juntar software e hardware, com o fim de melhorar os métodos e assegurar exatidão e firmeza. Um dos modelos básicos usados para aprender e usar métodos de controle é o sistema barra-bola, que se baseia em manter uma bola equilibrada em cima de uma barra que pode se inclinar. Saber bem as dinâmicas em jogo, como o giro, o atrito e as forças presentes, são essenciais para criar um bom controlador.

Esta monografia tem como objetivo explorar a automação e uma possível estratégia de controle aplicada ao sistema barra-bola, demonstrando técnicas como controle PID (Proporcional-Integral-Derivativo), e modelagem de sistemas que podem ser implementadas para garantir a estabilidade e o desempenho desejados. Além disso, o objetivo é desenvolver uma solução de baixo custo, utilizando componentes acessíveis e técnicas de programação eficientes, que possa ser replicada em ambientes educacionais ou industriais. O objetivo é que a automação proposta permita alcançar resultados experimentais que se aproximem das simulações teóricas para validar a eficácia do controlador e abrindo portas para aplicações mais complexas.

**Palavras-chaves:** Barra-Bola. Controlador. Automação. Equilíbrio. Controle.

## SUMÁRIO

1	INTRODUÇÃO.....	7
1.1	Justificativa .....	7
1.2	Objetivos gerais .....	8
1.3	Objetivos específicos .....	8
2	FUNDAMENTAÇÃO TEÓRICA .....	9
2.1	Modelagem .....	9
2.2	Técnica de Controle com PID.....	10
2.2.1	Ação Proporcional (P) .....	10
2.2.2	Ação Integral (I) .....	10
2.2.3	Ação Derivativa (D) .....	11
2.2.4	Ações PID Combinadas.....	11
2.3	Equipamentos para Implementação das Técnicas de Controle.....	12
2.3.1	Servo Motor .....	12
2.3.2	Sensores .....	14
2.3.3	ESP32 .....	15
2.3.4	Impressão 3D.....	16
3	METODOLOGIA.....	17
3.1	Desenvolvimento do Projeto.....	17
3.1.1	Modelagem .....	17
3.1.2	Escolhendo os Valores de $K_p$ , $K_i$ e $K_d$ .....	19
3.1.3	Montagem da Planta .....	22
3.1.4	Código .....	24
3.1.5	Código para definição do set-point.....	26
3.2	Montagem Final .....	27
3.3	Resultados Experimentais.....	28

4	CONCLUSÃO.....	30
	REFERÊNCIAS .....	31

# **1 INTRODUÇÃO**

O controle de sistemas dinâmicos é uma área fundamental da engenharia, com aplicações que vão desde a automação industrial até a robótica e aeroespacial. Dentre os diversos sistemas utilizados para estudo e desenvolvimento de técnicas de controle, o sistema Barra-Bola destaca-se como um clássico exemplo de problema de equilíbrio e estabilização. Este sistema consiste em uma bola que se move sobre uma barra, cuja inclinação pode ser controlada para manter a bola em uma posição desejada, geralmente o centro da barra.

Para desenvolver um controlador para o sistema Barra-Bola envolve várias etapas, como a modelagem do sistema, a análise das grandezas físicas envolvidas e a implementação de estratégias de controle que garantam a estabilidade e um bom desempenho. Este trabalho tem como objetivo detalhar essas etapas, desde a modelagem do sistema até a implementação de um controlador suficientemente bom e capaz de manter a bola em equilíbrio sobre a barra, mesmo na presença de perturbações externas.

## **1.1 Justificativa**

Esse sistema possui um caráter didático e é usado em diversos exemplos de um controlador que usa PID. Assim, o estudo do sistema Barra-Bola possui relevância prática, uma vez que problemas de equilíbrio e estabilização são comuns em diversas aplicações na indústria e até no entretenimento, como veículos autônomos, drones e sistemas de controle de posicionamento.

## **1.2 Objetivos gerais**

Dessa forma, esse trabalho se propõe a apresentar de forma detalhada o desenvolvimento de um controlador para o sistema Barra-Bola, unindo conceitos teóricos e práticos de controle estudados ao longo do curso de Engenharia de Controle e Automação, com o intuito de apresentar uma solução viável e eficaz para o problema. Para isso, serão abordados métodos de modelagem do sistema, análise de suas dinâmicas e a aplicação de técnicas de controle, como o controle PID (Proporcional-Integral-Derivativo), que é amplamente utilizado na indústria por sua simplicidade e eficiência. Além disso, serão realizados testes e simulações para validar o desempenho do controlador, garantindo que ele seja capaz de manter a estabilidade do sistema mesmo com perturbações externas.

## **1.3 Objetivos específicos**

Os objetivos específicos deste trabalho estão listados a seguir:

- Desenvolver um controlador que atenda ao sistema proposto;
- Realizar os testes do sensor para validação da posição da bola com erro dentro da margem do aceitável;
- Manipulação da variável da posição da bola no código para o uso de métodos de amortecimento dos sinais de picos (leituras incorretas) e na implementação do PID no projeto.
- Realizar a codificação do projeto no software Arduino IDE.



## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Modelagem**

A modelagem é uma etapa essencial no projeto e análise de sistemas que visam monitorar e regular o comportamento de processos dinâmicos, como em automação industrial, robótica, ou controle de processos químicos. Essa modelagem consiste em representar matematicamente o sistema físico, capturando suas dinâmicas e interações por meio de equações diferenciais, funções de transferência ou modelos em espaço de estados. Essas representações permitem prever como o sistema responderá a diferentes entradas e perturbações, garantindo estabilidade, desempenho e robustez (MARINHO, 2016).

Assim, técnicas como a linearização em torno de pontos de operação e a identificação de sistemas a partir de dados experimentais são comumente empregadas para aprimorar a precisão dos modelos. A modelagem adequada é fundamental para o sucesso de estratégias de controle, como o controle PID, controle preditivo ou controle robusto, pois permite o ajuste preciso dos parâmetros do controlador e a garantia de que o sistema atenda aos requisitos de desempenho desejados (MARINHO, 2016).

## **2.2 Técnica de Controle com PID**

O controle do sistema Barra-Bola pode ser realizado utilizando técnicas clássicas, como o controle PID (Proporcional-Integral-Derivativo). O controle PID é amplamente utilizado na indústria devido à sua simplicidade e eficácia. Ele atua com base no erro, que é calculado como a diferença entre o valor desejado (set-point) e o valor medido (feedback do sensor), ajustando a saída do controlador de forma proporcional, integral e derivativa. A ação proporcional responde ao erro atual (penaliza erros grandes), a ação integral elimina erros acumulados ao longo do tempo (penaliza erros longos) e a ação derivativa pode evitar erros grandes prevendo possíveis sobressinais (penaliza movimentos muito bruscos), melhorando a resposta dinâmica do sistema (OGATA, 2010).

### **2.2.1 Ação Proporcional (P)**

A ação proporcional é a parte mais simples e direta do controlador PID. Ela atua com base no erro atual, que é a diferença entre o valor desejado (set-point) e o valor medido (feedback). O controlador aplica uma correção proporcional ao erro, multiplicando-o por um ganho  $K_p$ . Quanto maior o erro, maior será a correção aplicada. (OGATA, 2010). Assim, no sistema Barra-Bola, a ação proporcional ajuda a reduzir rapidamente o deslocamento da bola em relação ao centro da barra. No entanto, se usada isoladamente, a ação proporcional pode levar a oscilações ou a um erro residual, pois não é capaz de eliminar completamente o erro em regime permanente, deixando assim, um sistema marginalmente estável ou instável.

### **2.2.2 Ação Integral (I)**

A ação integral tem como objetivo eliminar o erro residual que persiste quando apenas a ação proporcional é utilizada. Ela atua sobre o acúmulo de erros ao longo do tempo, integrando o erro e multiplicando-o por um ganho  $K_i$ . Isso permite que o controlador corrija desvios pequenos, mas persistentes, que a ação proporcional não consegue resolver (OGATA, 2010). No sistema Barra-Bola, a ação integral é essencial para garantir que a bola permaneça exatamente no centro da barra, mesmo após pequenas perturbações. No entanto, um ganho integral muito alto pode causar instabilidade, pois o controlador pode reagir de forma excessiva a erros acumulados.

### **2.2.3 Ação Derivativa (D)**

A ação derivativa tem a função de antecipar mudanças futuras no erro, atuando sobre a taxa de variação do erro (derivada do erro). Ela multiplica a derivada do erro por um ganho  $K_d$ , ajudando a amortecer oscilações e a melhorar a resposta dinâmica do sistema (OGATA, 2010). No sistema Barra-Bola, a ação derivativa é crucial para suavizar o movimento da barra, evitando que o controlador reaja de forma abrupta a mudanças rápidas na posição da bola. Isso resulta em um sistema mais estável e com menor overshoot (ultrapassagem do valor desejado). No entanto, um ganho derivativo muito alto pode amplificar ruídos no sinal de feedback, prejudicando o desempenho do controlador.

### **2.2.4 Ações PID Combinadas**

A combinação das ações proporcional, integral e derivativa permite que o controlador PID atue de forma eficiente no sistema Barra-Bola. Enquanto a ação proporcional responde ao erro atual, a ação integral elimina erros acumulados e a ação derivativa prevê e suaviza mudanças futuras (OGATA, 2010). O ajuste adequado dos ganhos  $K_p$ ,  $K_i$  e  $K_d$  é essencial para garantir que o sistema atinja o equilíbrio de forma rápida, precisa e estável, demonstrando a importância do controle PID em sistemas dinâmicos como o Barra-Bola.

## **2.3 Equipamentos para Implementação das Técnicas de Controle**

### **2.3.1 Servo Motor**

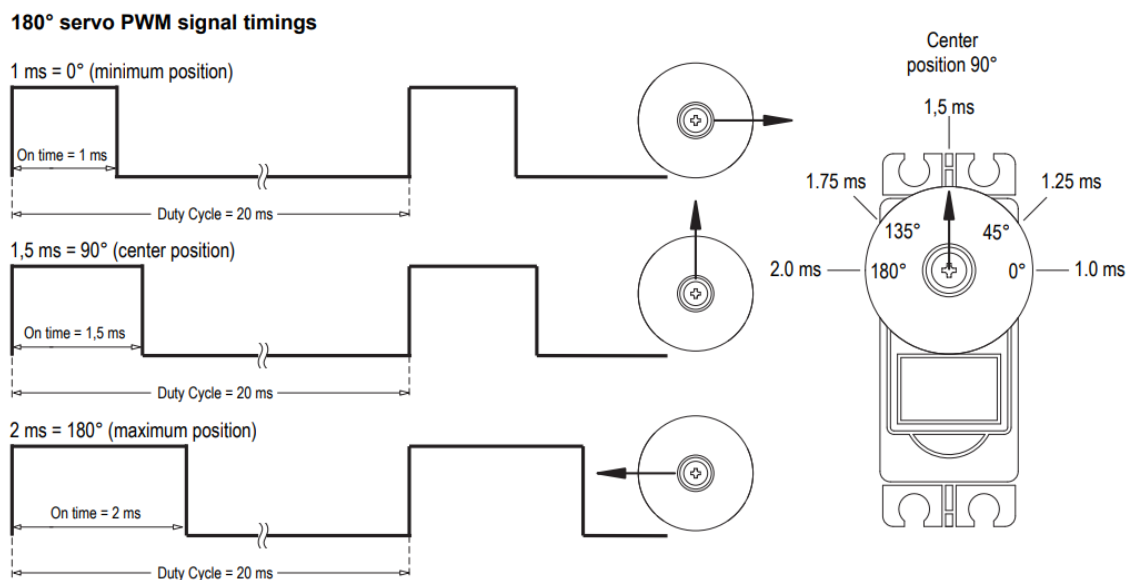
Os servomotores são dispositivos eletromecânicos amplamente utilizados em sistemas de controle de posição, velocidade e aceleração. Eles são compostos por um motor, um sistema de engrenagens, um potenciômetro ou encoder para feedback de posição e um circuito de controle interno. O funcionamento de um servo motor baseia-se no princípio de realimentação (feedback), onde o circuito interno compara a posição atual do eixo (informada pelo potenciômetro ou encoder) com a posição desejada, enviada por meio de um sinal PWM (Pulse Width Modulation). Com base nessa comparação, o motor é acionado para corrigir qualquer diferença, garantindo um posicionamento preciso e controlado. Essa capacidade de ajuste fino, e não apenas rotacionar sem objetivo de chegar em uma posição em específico, torna os servos motores ideais para aplicações que exigem movimentos precisos e repetitivos, como robótica, automação industrial e sistemas de controle (UNIVERSAL ROBOTS, SERVO MOTOR).

### 2.3.1.1 Servo Motor Futaba S3003

O servo motor Futaba S3003 é um dispositivo eletromecânico utilizado para controlar a posição angular da barra. Ele opera com base em sinais PWM (Pulse Width Modulation), onde a largura do pulso determina o ângulo de rotação do eixo do motor. O S3003 possui um circuito interno de controle que ajusta a posição do eixo com uma alta precisão, fazendo com que ele seja um bom servo motor para movimentos precisos e repetitivos. No sistema Barra-Bola, o servo motor é responsável por ajustar a inclinação da barra com base nos comandos enviados pelo controlador (S3003 STANDARD SERVO, DATASHEET).

A imagem a seguir ilustra como o sinal PWM afeta o motor, de acordo com o tempo em ON do sinal com relação ao Duty Cycle (ciclo de trabalho do sinal).

Figura 1: Funcionamento do Servo Motor.



Fonte: (Datasheet S3003 Standard Servo).

### **2.3.2 Sensores**

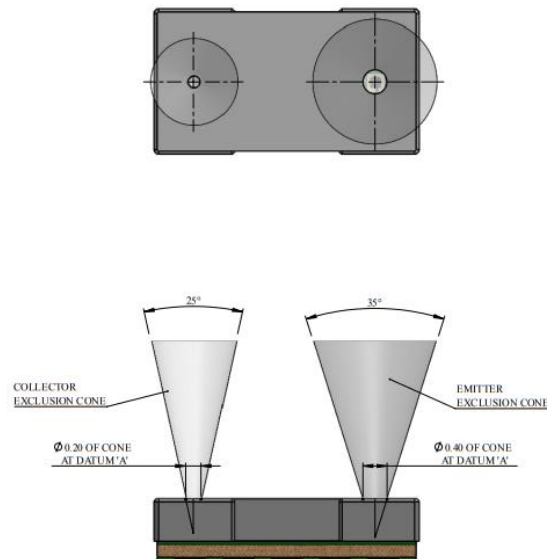
Sensores são componentes fundamentais em sistemas de controle, fornecendo informações em tempo real sobre o estado do processo controlado. Eles convertem grandezas físicas, como posição, velocidade, temperatura, pressão ou distância, em sinais elétricos que podem ser processados pelo controlador. No contexto de sistemas de controle, como o sistema Barra-Bola, os sensores desempenham um papel crítico ao fornecer o feedback necessário para que o controlador ajuste suas ações e mantenha o sistema operando dentro dos parâmetros desejados. Por exemplo, um sensor de distância pode medir a posição da bola sobre a barra, enquanto um encoder pode monitorar o ângulo da barra, permitindo que o controlador calcule o erro e ajuste a inclinação da barra de forma precisa (WEG, SENSORES INDUSTRIAIS).

#### **2.3.2.1 Sensor de Distância Infravermelho VL53L0X**

O sensor VL53L0X é um dispositivo de medição de distância baseado em tecnologia de tempo de voo (ToF - Time of Flight) que emite pulsos de luz infravermelha e mede o tempo que a luz leva para refletir em um objeto e retornar ao sensor. A partir desse tempo, o sensor calcula a distância entre ele e o objeto, mesmo em condições de baixa luminosidade a precisão ainda pode ser considerada aceitável para a maioria das aplicações. No sistema Barra-Bola, o VL53L0X é utilizado para medir a posição da bola sobre a barra, fornecendo o feedback necessário para o controlador ajustar a inclinação da barra e manter a bola próxima do set-point (VL53L0X, DATASHEET).

A imagem a seguir ilustra o funcionamento do sensor mostrando o emissor (a direita) e o receptor (a esquerda) e seus respectivos graus do ângulo de abertura em que o sinal pode ser emitido e recebido pelo sensor (VL53L0X, DATASHEET).

Figura 2: Funcionamento do sensor VL53L0X.



Fonte: (Datasheet VL53L0X).

### 2.3.3 ESP32

O ESP32 é um microcontrolador altamente versátil e popular, desenvolvido pela Espressif Systems, que combina capacidade de processamento, conectividade Wi-Fi e Bluetooth em um único chip. O ESP32 possui dois núcleos de processamento, clock de até 240 MHz, memória RAM suficiente para aplicações embarcadas, o que o torna ideal para projetos que exigem controle de hardware e comunicação com outros dispositivos (ESP32, DATASHEET).

#### **2.3.4 Impressão 3D**

A impressão 3D é uma tecnologia revolucionária que permite a criação de objetos físicos a partir de modelos digitais, camada por camada. A impressão 3D é amplamente utilizada em diversas áreas, como prototipagem rápida, medicina, arquitetura, engenharia e até mesmo na produção de peças funcionais para dispositivos eletrônicos e mecânicos. Sua acessibilidade e custo relativamente baixo a tornam uma ferramenta valiosa para projetos de pequena escala e personalizados (UNIFAL, USOS E APLICAÇÕES DA IMPRESSÃO 3D).



### **3 METODOLOGIA**

Neste capítulo será abordado como foi feita a integração dos componentes utilizados para a montagem do sistema barra-bola, bem como sua codificação foi desenvolvida e os testes necessários para a verificação adequada do funcionamento de cada componente utilizado nesse projeto.

#### **3.1 Desenvolvimento do Projeto**

Para a montagem do projeto são necessários alguns componentes fundamentais, tais como: barra, bola, sensor, suporte do sensor, servo motor, suporte do servo motor, ESP32, fios de conexão e outras peças de impressão 3D auxiliares.

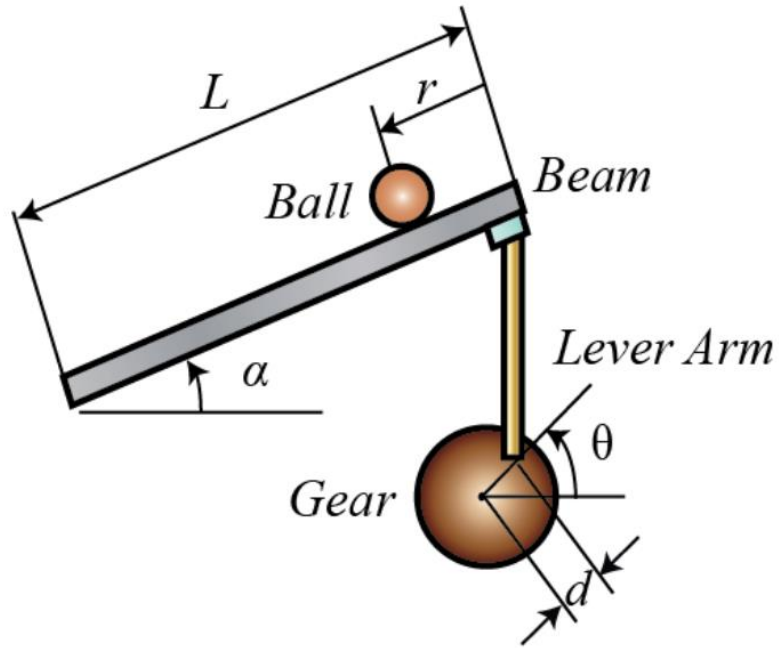
##### **3.1.1 Modelagem**

A modelagem do sistema envolve alguns parâmetros físicos como, ângulo, comprimento dos objetos envolvidos, peso, aceleração, etc. A lista de todos os parâmetros para esse projeto está listada a seguir:

- Massa da bola ( $m$ )
- Raio da Bola ( $R$ )
- Braço de giro do servo motor ( $d$ )
- Aceleração da gravidade ( $g$ )
- Comprimento da barra ( $L$ )
- Momento de inercia da bola ( $J$ )
- Posição da bola em relação ao sensor ( $r$ )
- Ângulo da barra ( $\alpha$ )
- Ângulo do servo motor ( $\theta$ )

A imagem a seguir ilustra os parâmetros variáveis do sistema barra-bola:

Figura 4: Esquemático do sistema barra-bola.



Fonte: (Control Tutorials for Matlab e Simulink).

Para esse sistema, a equação de Lagrange é dada por:

$$0 = \left( \frac{J}{R^2} + m \right) \ddot{r} + mg \sin \alpha - mr\dot{\alpha}^2$$

Como essa equação não é linear, vamos aproximar  $\alpha = 0$ . A equação linearizada do sistema é apresentada a seguir:

$$\left( \frac{J}{R^2} + m \right) \ddot{r} = -mg\alpha$$

A equação aproximada da relação entre os ângulos  $\alpha$  e  $\theta$  é:

$$\alpha = \frac{d}{L}\theta$$

Substituindo alfa na equação anterior, obtemos:

$$\left( \frac{J}{R^2} + m \right) r = -mg \frac{d}{L} \theta$$

Com a transformada de Laplace da equação acima, temos:

$$\left( \frac{J}{R^2} + m \right) R(s) s^2 = -mg \frac{d}{L} \Theta(s)$$

Reorganizando a equação:

$$P(s) = \frac{R(s)}{\Theta(s)} = - \frac{mgd}{L \left( \frac{J}{R^2} + m \right)} \frac{1}{s^2} \quad \left[ \frac{m}{rad} \right]$$

### 3.1.2 Escolhendo os Valores de Kp, Ki e Kd

Dados dos valores constantes da planta real:

- Massa da bola (m) = 0,058 Kg
- Raio da Bola (R) = 0,0225 m
- Braço de giro do servo motor (d) = 0,09 m
- Aceleração da gravidade (g) = 9,8 m/s<sup>2</sup>
- Comprimento da barra (L) = 0,4 m
- Momento de inércia da bola (J) = 0,00004698 Kg.m<sup>2</sup>

Para calcular a função de transferência da planta foi adicionado o seguinte código ao software Matlab no command window, como mostra a figura a seguir:

Figura 5: Código do Matlab para obter a função de transferência do sistema.

```
>> m = 0.058;  
R = 0.0225;  
g = -9.8;  
L = 0.4;  
d = 0.09;  
J = 46.98e-6;  
  
s = tf('s');  
P_ball = -m*g*d/L/(J/R^2+m)/s^2  
  
P_ball =  
  
0.8481  
-----  
s^2
```

Fonte: Autor.

Agora com a função de transferência do sistema em mãos, foi feito o seguinte diagrama de blocos no Simulink:

Figura 6: Diagrama de blocos montado no Simulink.

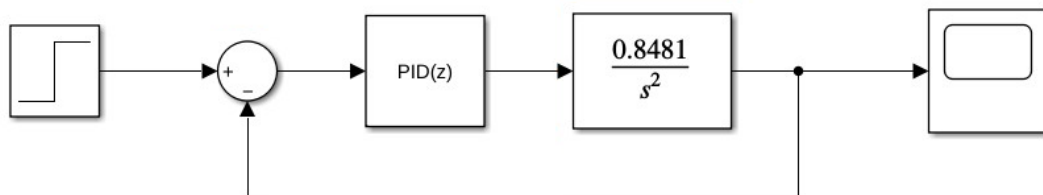
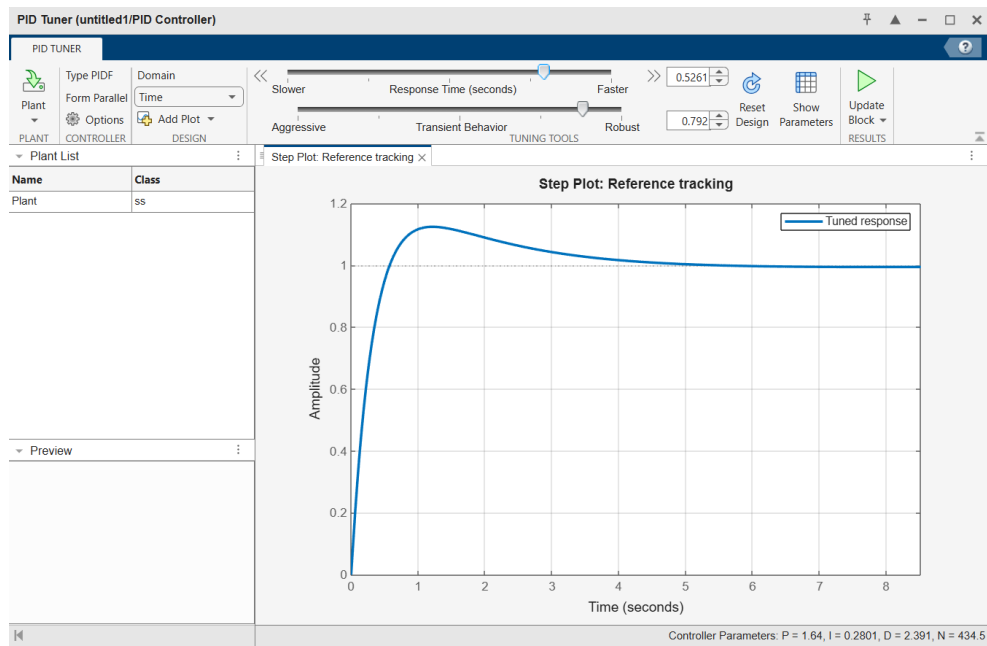


Figura: Autor.

Agora com o diagrama de blocos estruturado, é possível usar o PID Tuner do Simulink, uma função do software que escolhe valores de  $K_p$ ,  $K_i$  e  $K_d$ . A figura a seguir mostra o funcionamento dessa funcionalidade:

Figura 7: PID Tuner do software Simulink.



Fonte: Autor.

Como mostrado na imagem, os valores encontrados foram:

- $K_p = 1,64$
- $K_i = 0,2801$
- $K_d = 2,391$

### 3.1.3 Montagem da Planta

A montagem da planta consiste na combinação das peças impressas 3D com os componentes eletrônicos. A imagem a seguir ilustra uma montagem usada como base para esse projeto.

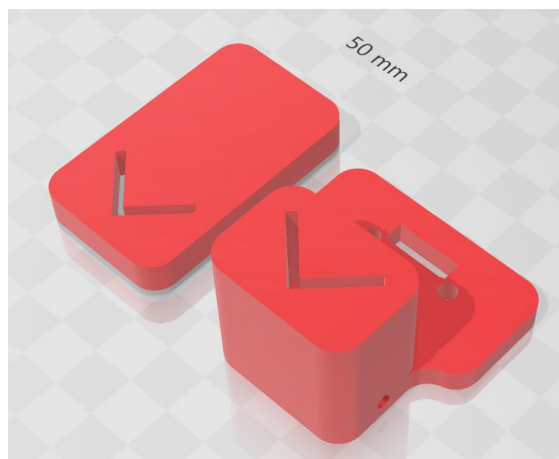
Figura 8: Esquemático de montagem.



Fonte: (Instructables: Ball and Beam W/LabVIEW & Arduino).

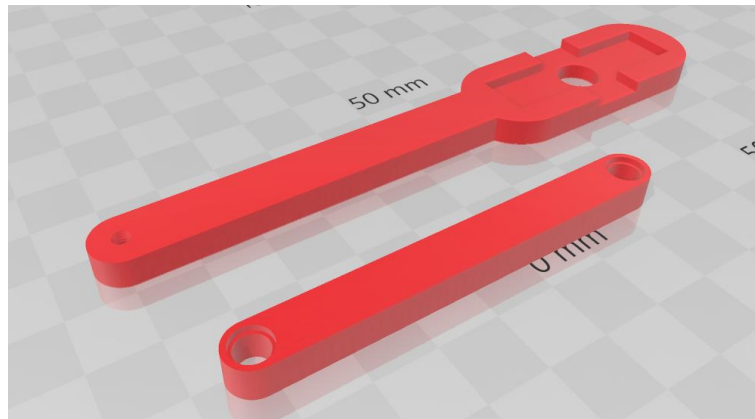
#### 3.1.3.1 Peças Impressas

Figura 9: Peça limitador final e inicial da barra com suporte para sensor.



Fonte: Autor.

Figura 10: Peças de braço do servo motor.



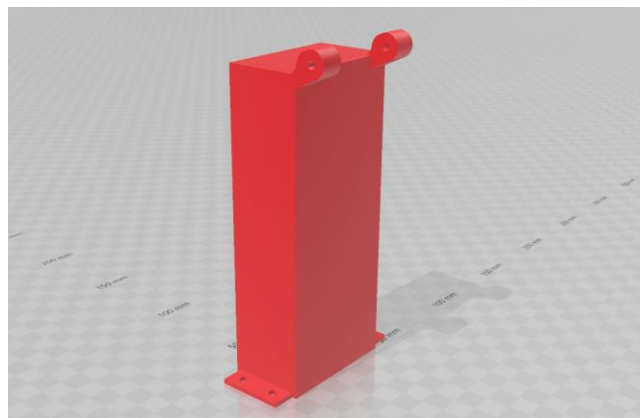
Fonte: Autor.

Figura 11: Peça de suporte do servo motor e encaixe do braço do servo na barra.



Fonte: Autor.

Figura 12: Peça de suporte para a barra e sensor.



Fonte: Autor.

### 3.1.4 Código

Para o projeto, foi escolhida a linguagem em C++ pela sua facilidade de entendimento e sua alta capacidade de aplicações complexas e precisas. As imagens a seguir ilustram o código aplicado ao controlador ESP32.

Figura 13: Início do código e declaração de variáveis.

```
1  #include <Wire.h>
2  #include "Adafruit_VL53L0X.h"
3  #include <ESP32Servo.h>
4
5  Servo myservo;
6  Adafruit_VL53L0X lox = Adafruit_VL53L0X();
7
8  // Configuração dos pinos I2C
9  #define SDA_PIN 21 // Pino SDA (D21)
10 #define SCL_PIN 23 // Pino SCL (D23)
11
12 float dist; // Distância em centímetros
13 float kp = 1.64, ki = 0.2801, kd = 2.391;
14 float setpoint = 18;
15 float erro = 0, erro_anterior = 0;
16 float saidaPID = 0;
17 float PID_i = 0;
18 int angulo = 140;
19 unsigned long tempoAnterior = 0;
20 unsigned long intervaloControle = 70; // Intervalo ajustado para 70 ms
```

Fonte: Autor.

Figura 14: void setup do código.

```
22 void setup() {
23     Serial.begin(115200);
24     while (!Serial) {delay(1);}
25
26     // Inicializa a comunicação I2C com os pinos customizados
27     Wire.begin(SDA_PIN, SCL_PIN);
28
29     if (!lox.begin()) {
30         Serial.println(F("Falha ao inicializar o sensor"));
31         while(1);
32     }
33
34     myservo.attach(13);
35     myservo.write(angulo);
36 }
```

Fonte: Autor.



Figura 15: void loop do código.

```
38 void loop() {
39     unsigned long tempoAtual = millis();
40     if (tempoAtual - tempoAnterior >= intervaloControle) {
41         tempoAnterior = tempoAtual;
42
43         dist = get_dist(5); // 5 leituras por ciclo
44
45         erro = setpoint - dist;
46
47         if (erro < 1 && erro > -1) {myservo.write(140);}
48
49         else if (dist != -1) { // Ignora leituras inválidas
50             // Cálculo do PID
51
52             if (-5 < erro && erro < 5) {
53                 PID_i += ki * erro;
54             } else {
55                 PID_i = 0;
56             }
57
58             float derivadaErro = erro - erro_anterior;
59
60             saidaPID = (kp * erro) + PID_i + (kd * derivadaErro);
61
62             saidaPID = constrain(saidaPID, -45, 45); // Limita a saída
63             myservo.write(140 + saidaPID);
64
65             erro_anterior = erro;
66         }
67     }
68 }
```

Fonte: Autor.

Figura 16: Função para obter a distância da bola.

```
70 float get_dist(int n) {
71     long soma = 0;
72     int validas = 0;
73
74     for (int i = 0; i < n; i++) {
75         VL53L0X_RangingMeasurementData_t medida;
76         lox.rangingTest(&medida, false);
77
78         if (medida.RangeStatus != 4) { // Verifica leitura válida
79             soma += medida.RangeMilliMeter;
80             validas++;
81
82             delay(5); // Pequeno delay para estabilidade
83             float distancia_cm = (soma / validas) / 10.0; // Converte para cm
84             Serial.print("Distancia (cm): ");
85             Serial.println(distancia_cm);
86             return distancia_cm;
87         }
88
89         else {
90             Serial.println("Leitura invalida");
91             return -1; // Retorna -1 se todas as leituras forem inválidas
92         }
93     }
94 }
```

Fonte: Autor.

### 3.1.5 Código para definição do set-point

Figura 17: Código para definição do set-point.

```
int valorAnalogico = analogRead(34);  
if (valorAnalogico == 0){myservo.write(140);}  
else{  
  setpoint = (valorAnalogico * 0.0061) + 5;
```

Fonte: Autor.

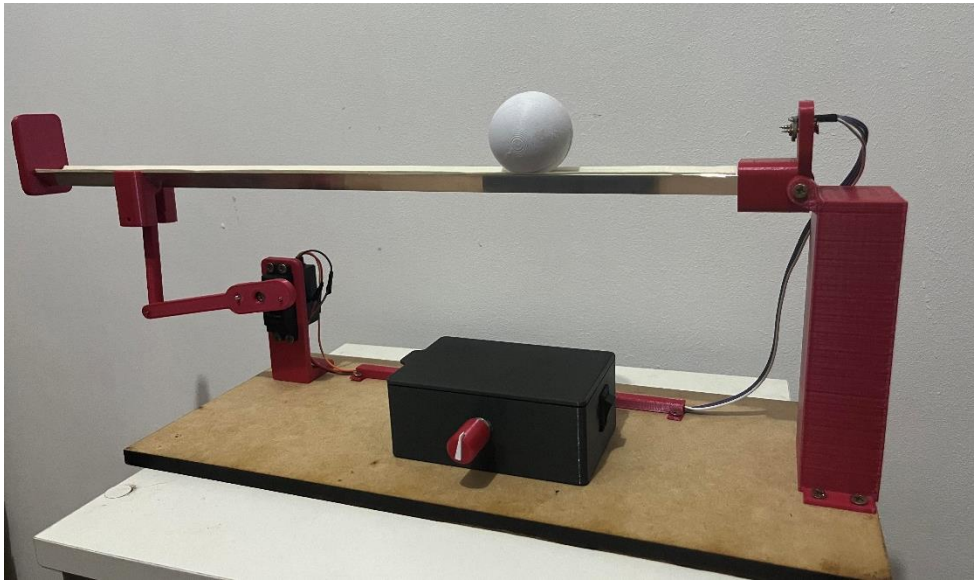
A figura 17 mostra como foi feito o código para definir um set-point variável de acordo com a tensão do potenciômetro conectado ao pino 34 do ESP32.

Em seguida, o valor é convertido para o sinal do set-point, variando de 0 até 22 centímetros do sensor. O erro é então calculado a partir do set-point definido pelo potenciômetro.

### 3.2 Montagem Final

A montagem final do projeto barra bola é mostrado na imagem a seguir:

Figura 18: Montagem final do projeto.



Fonte: Autor.

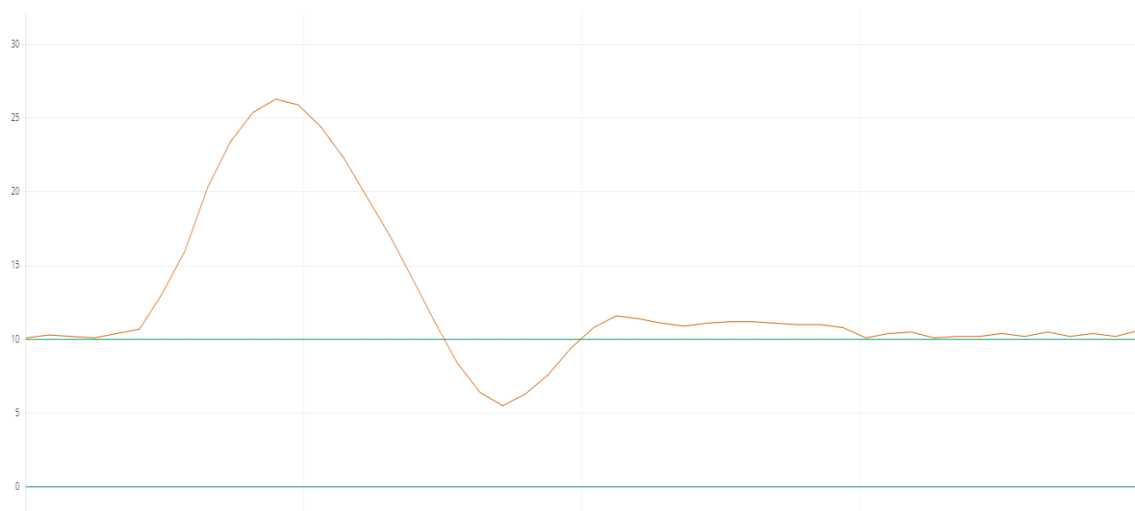
### 3.3 Resultados Experimentais

A partir da planta e do código apresentados, foram realizados testes com o controlador PID, conforme ilustrado na Figura 15. O comportamento do sistema com os ganhos PID iniciais, mostrados na Figura 13, apresentou oscilações excessivas e não conseguiu atingir o set-point de forma satisfatória, resultando em um erro de estado estacionário significativo.

Para melhorar o desempenho do sistema, buscando uma resposta mais amortecida e menos oscilatória, o ganho  $K_d$  foi aumentado em dez vezes, atingindo um novo valor de 23,91. Esse ajuste permitiu uma melhoria significativa na resposta do sistema.

A figura a seguir mostra o comportamento do sistema, por 6 segundos, com o ganho  $K_d$  atualizado.

Figura 19: Gráfico do Comportamento do Sistema.

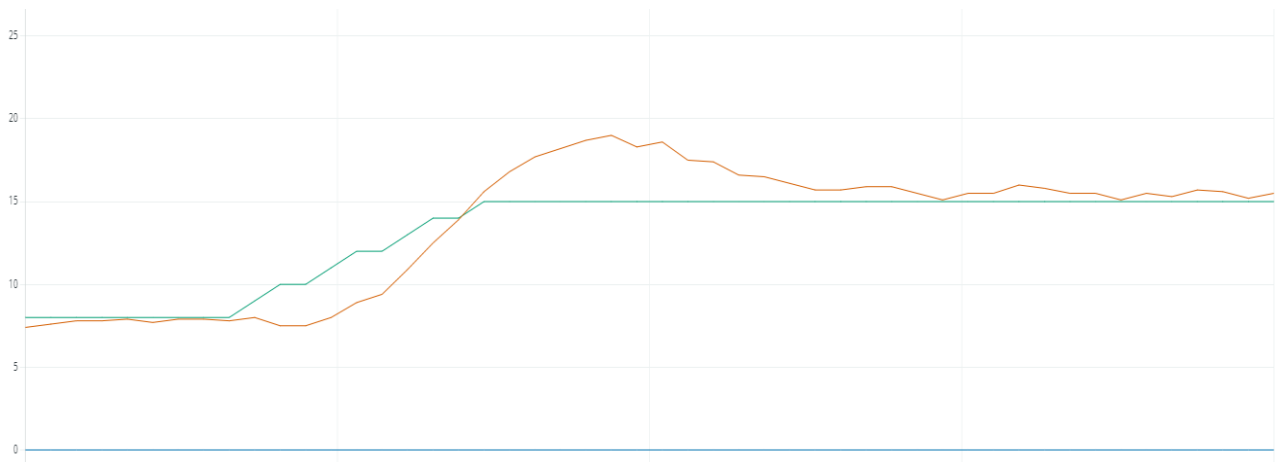


Fonte: Autor.

O gráfico da figura 19 apresenta a estabilização do sistema após uma perturbação externa, nesse caso, um impulso na bola para longe do set-point. Assim, podemos observar no gráfico um sobressinal e tempo de estabilização aceitável para um sistema barra-bola.

Dessa forma, mesmo com as perturbações externas o sistema se mostrou controlar a posição da bola de forma satisfatória, mostrando que os ganhos PID definidos a partir da modelagem do sistema foram validados como funcional de forma geral, mesmo que tenha sido necessário alterações nos testes experimentais feitos posteriormente.

Figura 20: Gráfico do comportamento do sistema com mudança de set-point.



Fonte: Autor.

A figura 20 mostra o gráfico do comportamento do sistema, em um total de 6 segundos, a uma mudança do set-point utilizando o potenciômetro, variando de aproximadamente 8cm à 15cm. No gráfico, o set-point é ilustrado pela linha verde e a posição da bola é ilustrado pela linha laranja.

## 4 CONCLUSÃO

O desenvolvimento de um controlador para o sistema Barra-Bola apresentou-se como um desafio enriquecedor, permitindo a aplicação prática de conceitos teóricos estudados ao longo do curso de Engenharia de Controle e Automação. A implementação de um controlador PID (Proporcional-Integral-Derivativo) mostrou-se eficaz na manutenção do equilíbrio da bola sobre a barra, mesmo na presença de perturbações externas. A combinação das ações proporcional, integral e derivativa permitiu um controle preciso e estável, demonstrando a importância do ajuste adequado dos ganhos  $K_p$ ,  $K_i$  e  $K_d$  para o desempenho do sistema.

A utilização de componentes como o sensor de distância VL53L0X, o servo motor Futaba S3003 e o microcontrolador ESP32 proporcionou uma solução de baixo custo e alta eficiência, viável para aplicações educacionais e até industriais.

Além disso, a impressão 3D teve um papel fundamental na fabricação de peças personalizadas, como suportes para o sensor e o servo motor, além de permitir o posicionamento específico de peças, o que é indispensável para o sensor que se demonstrou extremamente sensível a variações de ângulo de leitura da bola. Assim, a flexibilidade da impressão 3D permitiu aplicar e trocar as peças rapidamente na planta, testando diferentes configurações e tamanhos até alcançar a solução que mais se adequa às necessidades sem precisar gastar muito tempo com isso.

A integração do sensor, servo motor e o microcontrolador foram muito simbólicos à diversos conceitos da engenharia de controle e automação, como por exemplo a obtenção de dados da planta, processamento desses dados pelo controlador e atuação na planta a partir desses dados processados são conceitos fundamentais de sistemas de controle. Esses processos ilustram de maneira prática como a teoria se materializa em sistemas reais, destacando a importância da realimentação (feedback) e do controle em malha fechada para garantir precisão, estabilidade e desempenho em aplicações dinâmicas.

Portanto, o desenvolvimento do projeto se demonstrou extremamente didático para a aplicação de um controlador em um sistema dinâmico relativamente rápido. Além de reforçar a importância da aplicação de conceitos teóricos na prática, como a modelagem do sistema barra bola aplicada neste projeto.

## REFERÊNCIAS

CONTROL TUTORIALS FOR MATLAB E SIMULINK. Disponível em: <  
<https://ctms.engin.umich.edu/CTMS/?example=BallBeam&section=SystemModeling>>.

Acesso em: 02/03/2025.

INSTRUCTABLES. Ball and Beam W/LabVIEW & Arduino. Disponível em: <  
<https://www.instructables.com/Ball-and-Beam-WLabVIEW-Arduino/>>.

Acesso em: 02/03/2025.

ESP32 SERIES. Datasheet. Disponível em: <  
[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)>.

Acesso em: 05/03/2025.

S3003 Standard Servo. Datasheet. Disponível em: <  
[https://www.kjell.com/globalassets/mediaassets/701905\\_87902\\_datasheet\\_en.pdf](https://www.kjell.com/globalassets/mediaassets/701905_87902_datasheet_en.pdf)>.

Acesso em: 05/03/2025.

VL53L0X. Datasheet. Disponível em:  
<<https://www.st.com/resource/en/datasheet/vl53l0x.pdf>>.

Acesso em: 05/03/2025.

UNIVERSAL ROBOTS, SERVO MOTOR. Disponível em: < <https://www.universal-robots.com/br/blog/servo-motor-o-que-%C3%A9-como-funciona-e-seu-uso-na-rob%C3%B3tica-industrial/>>.

Acesso em: 11/03/2025.

WEG, SENSORES INDUSTRIAIS. Disponível em: <  
<https://www.weg.net/digital/blog/o-que-sao-sensores-industriais/>>.  
Acesso em: 11/03/2025.

UNIFAL, USOS E APLICAÇÕES DA IMPRESSÃO 3D. Disponível em: <  
<https://publicacoes.unifal-mg.edu.br/revistas/index.php/cei/article/view/2097/1625>>.  
Acesso em: 11/03/2025.

MARINHO, ANTONIO. ANÁLISE E MODELAGEM DE SISTEMAS. Pearson, 2016.

OGATA, K. ENGENHARIA DE CONTROLE MODERNO. 5a ed. Pearson, 2010.