

Ball and Beam

Bernardo Murta Bastos

Abstract—Neste projeto, foi implementado um controle PID (Proporcional-Integral-Derivativo), cujo objetivo é equilibrar uma bola em uma barra utilizando um Arduino, um servomotor e um sensor ultrassônico. A proposta é desenvolver um controle automático que ajuste o ângulo da barra de forma dinâmica para manter a bola em uma posição central. O sensor mede a posição da bola ao longo da barra, enviando essas informações ao Arduino, que, por sua vez, calcula os ajustes necessários usando o controle PID. O servomotor, então, inclina a barra para corrigir a posição da bola, aplicando as correções proporcionais ao desvio, acumulativas ao longo do tempo e antecipativas para estabilizar o sistema. Esse projeto combina eletrônica, programação e controle de sistemas, proporcionando uma aplicação prática para a teoria do controle PID. Ao final, espera-se obter um sistema capaz de manter a bola em equilíbrio, resistindo a pequenas perturbações e reagindo rapidamente a deslocamentos, promovendo a compreensão do funcionamento dos algoritmos de controle e aprimorando habilidades técnicas em eletrônica e automação.

Index Terms—Bola, Barra, Servomotor, Arduino, PID, Ultrassônico

I. INTRODUÇÃO

A Teoria do controle PID (Proporcional-Integral-Derivativo) é amplamente reconhecida como uma das abordagens mais eficazes e versáteis em sistemas de automação e controle, sendo utilizada em aplicações que vão desde processos industriais até dispositivos eletrônicos de consumo. Sua relevância se deve à capacidade de manter estabilidade e precisão em processos dinâmicos, ajustando variáveis em tempo real para minimizar desvios, corrigir erros acumulados e antecipar mudanças futuras no comportamento do sistema [1] [2].

O controlador PID baseia-se em três componentes principais: o termo proporcional (P), que reage ao erro atual entre o valor desejado e o valor medido, esse termo aplica uma correção proporcional à magnitude do erro, aumentando a rapidez de resposta do sistema; o termo integral (I), acumula os erros passados ao longo do tempo, corrigindo desvios sistemáticos que o componente proporcional isolado não conseguiria eliminar, como offsets persistentes; e o termo derivativo (D), atua como um mecanismo preditivo, reagindo à taxa de variação do erro. Ele ajuda a suavizar a resposta do sistema, reduzindo oscilações e melhorando a estabilidade [3] [4]. Esse equilíbrio entre os três elementos resulta em um controle robusto e preciso, tornando o PID amplamente adotado em sistemas como plataformas de estabilização, pêndulos invertidos, robôs móveis e veículos autônomos [5].

O presente trabalho propõe o desenvolvimento e a implementação de um sistema de controle PID aplicado ao equilíbrio dinâmico de uma bola sobre uma barra, utilizando um microcontrolador Arduino, um servomotor e um sensor ultrassônico. Esse tipo de experimento é amplamente utilizado em ambientes acadêmicos e laboratórios de engenharia devido

à sua capacidade de integrar conceitos teóricos e práticos, além de proporcionar uma plataforma tangível para o estudo de sistemas dinâmicos [6] [7]. O sistema busca estabilizar a posição da bola ajustando dinamicamente o ângulo da barra, utilizando os cálculos do controlador PID para reagir a pequenos deslocamentos e perturbações externas.

A relevância desse tipo de projeto reside na sua aplicabilidade tanto como ferramenta de aprendizado quanto como base para estudos mais avançados. Sistemas de equilíbrio dinâmico são frequentemente utilizados como modelos simplificados de problemas reais em engenharia de controle, como o controle de drones, robôs bípedes e veículos autônomos [8] [9]. Além disso, a implementação do projeto promove o desenvolvimento de habilidades práticas em eletrônica, programação e modelagem de sistemas, consolidando o aprendizado de conceitos fundamentais de controle automático.

Por fim, estudos anteriores evidenciam a eficácia do controle PID em sistemas semelhantes, como o controle de pêndulos invertidos e sistemas de levitação magnética, demonstrando que, com os ajustes adequados dos parâmetros proporcional, integral e derivativo, é possível alcançar estabilidade, rapidez de resposta e precisão [10] [11]. Assim, este trabalho não apenas visa demonstrar a eficiência do controle PID em um sistema prático, mas também proporcionar uma experiência enriquecedora no desenvolvimento de sistemas de automação e controle.

II. METODOLOGIA

A. Materiais

Para este experimento, foram utilizados os seguintes materiais:

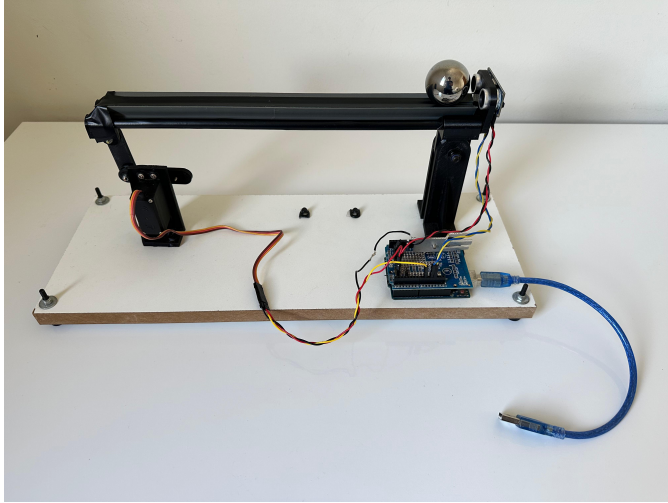
- Arduino UNO;
- Servo motor;
- Sensor ultrassônico;
- Estrutura feita em impressora 3D;
- Barras de ferro;
- Base rígida de madeira;
- Bolinha metálica;
- Software MATLAB/Simulink.

B. Métodos

O projeto utiliza um controlador PID (Proporcional-Integral-Derivativo) para equilibrar uma bolinha no centro de uma estrutura móvel. A estrutura é inclinada por meio de um servomotor, enquanto um sensor ultrassônico mede a posição da bolinha, fornecendo informações de feedback ao controlador.

- Arquitetura do Sistema: O sistema consiste em uma estrutura plana onde a bolinha se move, um sensor ultrassônico posicionado de forma a monitorar a posição

Fig. 1: Projeto físico montado



Fonte: dos autores.

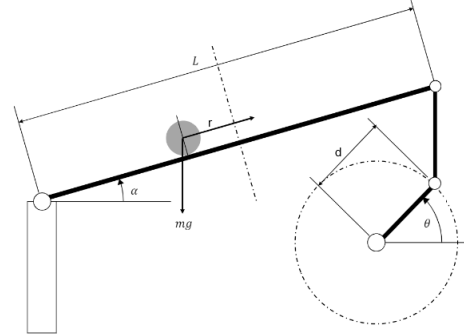
da bolinha em um eixo (horizontal), e um servomotor que inclina a estrutura para corrigir a posição da bolinha. A movimentação da estrutura em resposta ao controlador PID permite que a bolinha seja mantida próxima ao centro da plataforma.

- **Sensoriamento da Posição:** O sensor ultrassônico é posicionado a uma extremidade da estrutura e orientado para medir a distância da bolinha em relação ao sensor. Esse valor é convertido para indicar a posição da bolinha ao longo do eixo da estrutura. O sensor realiza leituras contínuas em uma taxa de amostragem fixa, enviando os dados para o controlador para processamento.
- **Controle PID:** O controlador PID processa as leituras de posição do sensor ultrassônico e calcula o erro, que é a diferença entre a posição atual da bolinha e a posição desejada (centro da estrutura). Com base nesse erro, o controlador aplica três termos principais: Proporcional (P): Proporciona uma correção proporcional ao erro atual, aumentando ou diminuindo a inclinação da estrutura para mover a bolinha em direção ao centro. Integral (I): Corrige erros acumulados ao longo do tempo, úteis para eliminar desvios residuais da posição central. Derivativo (D): Responde à taxa de variação do erro, ajudando a suavizar a resposta e a evitar oscilações excessivas. O valor de controle resultante é então enviado ao servomotor para ajustar a inclinação da estrutura.
- O sistema passa por uma fase de calibração para ajustar os parâmetros K_p , K_i e K_d do controlador PID. A calibração é feita iterativamente, observando-se o comportamento da bolinha e ajustando os valores de K_p , K_i e K_d até que o sistema consiga manter a bolinha de forma estável no centro com o mínimo de oscilação.
- **Execução e Monitoramento:** Durante a execução, o sistema coleta dados de posição e resposta do servo em tempo real. Os dados são analisados para avaliar a estabilidade do sistema, o tempo de resposta e a precisão do controlador PID. Esses dados podem ser usados para

ajustes adicionais no controle PID e para gerar relatórios de desempenho.

Primeiramente, foi observado o modelo físico do sistema, mostrado abaixo:

Fig. 2: Projeto físico montado



Fonte: dos autores.

Segundamente, utilizamos o método de Lagrange, pois ele é mais útil para sistemas com um número maior de graus de liberdade ou com coordenadas generalizadas. Nesse sentido, obtivemos pelo método de Lagrange a seguinte equação que rege o movimento da bola:

$$O = \left(\frac{J}{R^2} + m \right) \ddot{R} + mg \sin(\alpha) - m r \dot{\alpha}^2 \quad (1)$$

$$O = \frac{d}{L} \theta \quad (2)$$

Terceiramente, analisamos como o sistema se comporta utilizando o Simulink no MATLAB.

Para isso, utilizamos os seguintes valores:

- m (massa da bolinha)=0.174 Kg;
- g (gravidade)=-9,8 m/s²;
- l (comprimento)=0,28m;
- d (distância servo e haste)=0,085 m;
- J (inercia)=2,1315e⁻⁵ Kg*m²;
- r (distância)=0,0175m;

Para fazer a simulação, montamos a seguinte estrutura no Simulink:

Fig. 3: Estrutura geral montada

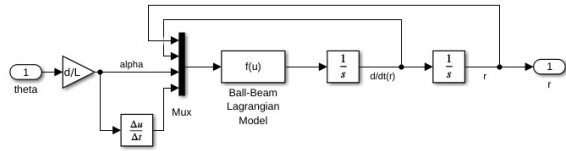


Fonte: dos autores.

Com a estrutura montada e os valores definidos, obtivemos a seguinte resposta ao degrau:

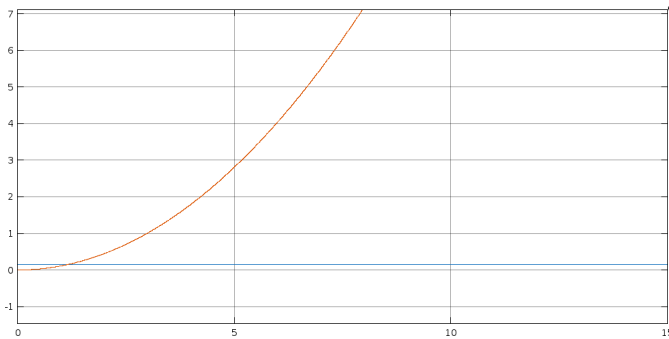
A partir deste gráfico, podemos concluir que o sistema é instável em malha aberta, fazendo com que a bola role diretamente para fora da extremidade da viga. Fazendo-se necessário a implementação do sistema de controle PID.

Fig. 4: Subsistema Ball and Beam



Fonte: dos autores.

Fig. 5: Resposta ao degrau



Fonte: dos autores.

O próximo passo foi achar a função de transferência do sistema, para isto, utilizamos:

$$O = \left(\frac{J}{R^2} + m \right) \ddot{R} + mg \sin(a) - m r \dot{a}^2 \quad (3)$$

Fazendo a linearização em torno de

$$a = 0 \quad (4)$$

, obtemos:

$$Ft = \frac{0.1449}{0.09744s^2} \quad (5)$$

Por fim, é necessário fazer algumas simulações via MATLAB para encontrarmos os valores de K_p , K_i e K_d .

Para que tudo ficasse conectado, era necessário fazer a discretização da planta e do PID, para isso, utilizamos o método Zero-Order Hold (ZOH) para discretizar a função de transferência e Backward Euler para discretizar o PID. Por fim, apesar de a planta e o PID já estarem discretizados, utilizamos o método das Equações de diferença para representar explicitamente a dinâmica do sistema no tempo discreto.

1) *Controle PID*: O controle PID (Proporcional-Integral-Derivativo) é implementado para ajustar a inclinação da viga e manter a bolinha no centro.

A equação do controlador PID é dada por:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t) \quad (6)$$

onde:

- $u(t)$ é o sinal de controle (ângulo da viga θ),
- $e(t)$ é o erro entre a posição desejada e a posição atual da bolinha,
- K_p é o ganho proporcional,

- K_i é o ganho integral,
- K_d é o ganho derivativo.

2) *Método de Backward Euler*: O método de Backward Euler (ou Euler Implícito) é um método numérico implícito para a resolução de equações diferenciais ordinárias (EDOs). Ele é derivado da aproximação para a derivada de primeira ordem:

$$\frac{dy}{dt} \approx \frac{y_{k+1} - y_k}{\Delta t} \quad (7)$$

Dessa forma, a formulação do método de Backward Euler para uma EDO genérica da forma $\frac{dy}{dt} = f(y, t)$ é dada por:

$$y_{k+1} = y_k + \Delta t \cdot f(y_{k+1}, t_{k+1}) \quad (8)$$

onde:

- y_k é a aproximação da solução no instante t_k ,
- y_{k+1} é a aproximação da solução no instante $t_{k+1} = t_k + \Delta t$,
- Δt é o passo de tempo,
- $f(y, t)$ representa a função da equação diferencial.

Como y_{k+1} aparece em ambos os lados da equação, esse método requer a resolução de uma equação implícita para determinar y_{k+1} , o que pode exigir técnicas iterativas como Newton-Raphson. Para o nosso sistema, obtivemos a seguinte equação:

$$Gc = K_p + K_d * \frac{z - 1}{T_s} \quad (9)$$

3) *Método Zero-Order Hold (ZOH)*: O Zero-Order Hold (ZOH) é um método utilizado para reconstrução de sinais discretos em sinais contínuos. Ele é amplamente empregado em sistemas de controle digital e conversores digital-analógico (DACs), pois mantém constante o valor da última amostra até que uma nova amostra seja adquirida.

Dado um sinal discreto $x[k]$, obtido por amostragem de um sinal contínuo $x(t)$, o ZOH mantém cada valor $x[k]$ constante ao longo do intervalo entre duas amostras sucessivas. Assim, o sinal reconstruído $x_{ZOH}(t)$ assume um formato em degraus, sendo definido por:

$$x_{ZOH}(t) = x[k], \quad \text{para } t \in [kT, (k+1)T) \quad (10)$$

onde:

- $x[k]$ representa a amostra do sinal original no instante kT ,
- T é o período de amostragem.

Esse comportamento faz com que a reconstrução do sinal tenha trechos constantes entre os instantes de amostragem, ao invés de uma interpolação suave.

A função de transferência do ZOH no domínio de Laplace é dada por:

$$H(s) = \frac{1 - e^{-sT}}{s} \quad (11)$$

Essa função representa o efeito do ZOH sobre o espectro do sinal amostrado, mostrando que ele atua como um filtro passa-baixas que suaviza a transição entre amostras, mas também introduz atenuações e distorções. Para o nosso sistema, obtivemos a seguinte equação:

$$Pballz = \frac{9.917e^{-05} + 9.917e^{-05}}{z^2 - 2z + 1} \quad (12)$$

4) *Equações de Diferença e Discretização*: Uma equação de diferença é uma expressão matemática utilizada para modelar sistemas discretos. A forma geral de uma equação de diferença linear é:

$$a_N y[k + N] + a_{N-1} y[k + N - 1] + \dots + a_0 y[k] = b_M x[k + M] + \dots + b_0 x[k] \quad (13)$$

onde:

- $y[k]$ representa a saída do sistema no instante k ,
- $x[k]$ representa a entrada do sistema no instante k ,
- a_i e b_i são coeficientes constantes,
- N e M são as ordens da equação de diferença.

A discretização de uma equação diferencial pode ser feita através do método de Euler, onde a derivada é aproximada por diferenças finitas:

$$\frac{dy}{dt} \approx \frac{y[k + 1] - y[k]}{T} \quad (14)$$

Assim, uma equação diferencial simples do tipo:

$$\frac{dy}{dt} + ay = bx \quad (15)$$

pode ser discretizada como:

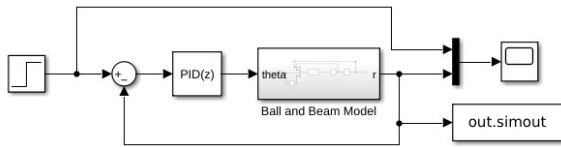
$$y[k + 1] = (1 - aT)y[k] + bTx[k] \quad (16)$$

Essa representação permite a implementação digital de sistemas dinâmicos.

III. RESULTADOS

Para darmos início a este tópico, abaixo apresentaremos o sistema em malha fechada já discretizado no simulink:

Fig. 6: Representação da planta do sistema em malha fechada no simulink

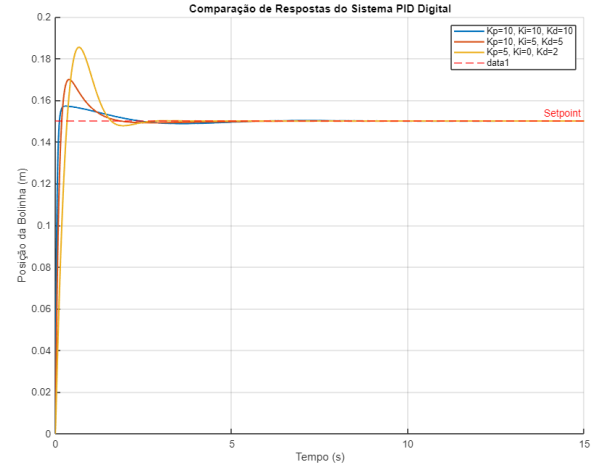


Fonte: dos autores.

Para conseguirmos obter os valores ideais de K_p , K_i e K_d , utilizamos alguns valores iniciais para as variáveis e modificamos iterativamente até obtermos um valor que se aproximasse ao máximo do projeto real.

Para a primeira análise, jogamos valores "aleatórios" de K_p , K_i e K_d , entretanto nenhum deles obteve um resultado próximo do resultado do projeto físico, como podemos ver no gráfico abaixo.

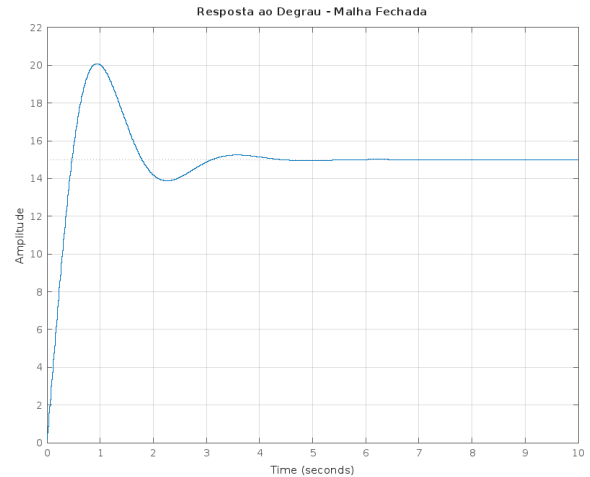
Fig. 7: Resposta do sistema PID para vários K_p , K_i e K_d



Fonte: dos autores.

Ao observar o projeto físico real, conseguimos ver que a resposta que mais se aproxima do é a que utiliza $K_p=5$, $K_i=0$ e $K_d=2$, entretanto ainda não apresenta um resultado satisfatório. Para tentarmos melhorar a resposta, utilizaremos agora $K_p=3$, $K_i=0$ e $K_d=1$. Assim, obtivemos para a resposta em degrau da malha fechada:

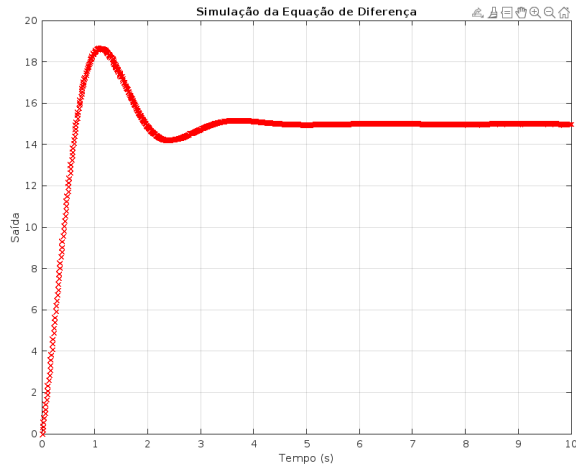
Fig. 8: Resposta ao degrau de 15 em malha fechada



Fonte: dos autores.

Já para a equação das diferenças, encontramos o seguinte resultado gráfico:

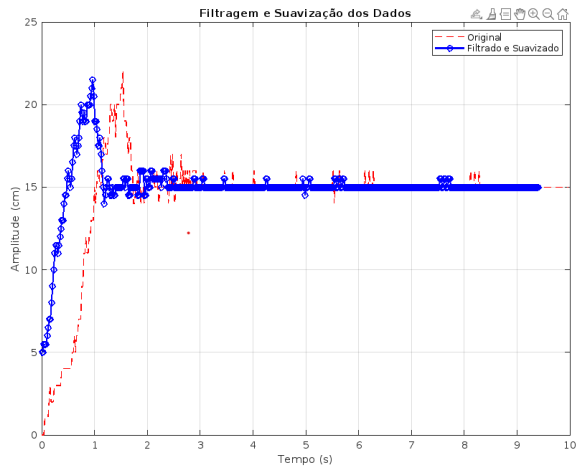
Fig. 9: Simulação da equação de diferença



Fonte: dos autores.

E, para os dados do projeto físico, encontramos o seguinte resultado via MtaLab:

Fig. 10: Dados do projeto físico simulado pelo MatLab

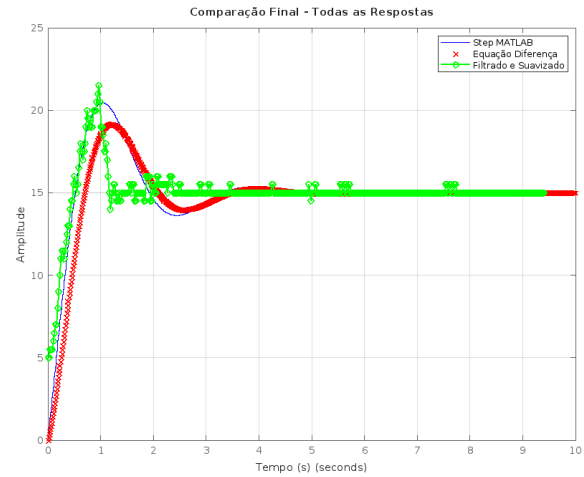


Fonte: dos autores.

Podemos observar que os gráficos são bastante semelhantes. No entanto, no experimento físico, o objeto inicia na posição 5. Isso ocorre porque o sensor ultrassônico não consegue detectar objetos posicionados abaixo desse valor.

Além disso, foi aplicado um filtro aos dados obtidos na simulação real, pois o sensor apresentava pequenas falhas nas medições. A utilização desse filtro se mostrou eficaz, garantindo maior precisão e estabilidade nos resultados. Por fim, foi gerado um gráfico consolidando todas as respostas para $K_p=3$, $K_i=0$ e $K_d=1$, proporcionando uma análise mais clara e facilitando a interpretação dos resultados.

Fig. 11: Resposta ao degrau em malha fechada, Simulação da equação de diferença e Dados do projeto físico simulado pelo MatLab



Fonte: dos autores.

Podemos ver que os resultados se aproximam bastante nas três respostas. Houve algumas distorções, mas ocorreram provavelmente por conta da falha do sensor ultrassônico.

IV. CONCLUSÕES

O desenvolvimento do sistema Ball and Beam utilizando controle PID proporcionou uma aplicação prática dos conceitos teóricos de eletrônica, automação e controle de sistemas. A implementação permitiu a análise do impacto dos ganhos proporcional (K_p), integral (K_i) e derivativo (K_d) na estabilidade do sistema, demonstrando a importância do ajuste fino desses parâmetros para alcançar um desempenho satisfatório.

A partir das simulações realizadas no MATLAB e das validações experimentais no protótipo físico, observou-se que a discretização da planta e do controlador foi essencial para garantir uma implementação eficiente no ambiente digital. O uso do método Zero-Order Hold (ZOH) para discretização da função de transferência e do Backward Euler para a discretização do PID permitiu a integração precisa entre hardware e software, garantindo a resposta esperada do sistema.

Durante os testes, notou-se que o sistema físico apresentou pequenas discrepâncias em relação à simulação, principalmente devido a limitações do sensor ultrassônico e do servomotor, que introduziram erros de medição e atraso na resposta. Para mitigar esses efeitos, técnicas de filtragem e suavização de dados foram aplicadas, contribuindo para a melhoria da precisão do sistema.

Além disso, a implementação da equação de diferença foi fundamental para representar explicitamente a dinâmica do sistema no tempo discreto, possibilitando a análise mais detalhada do comportamento da bola ao longo do tempo. Os resultados finais indicaram que, com os ajustes adequados, o sistema foi capaz de manter a bola equilibrada na barra e resistir a pequenas perturbações externas.

Este trabalho reforçou a importância da integração entre modelagem matemática, simulação computacional e implementação prática no estudo de sistemas dinâmicos. O projeto

não apenas validou o funcionamento do controle PID no problema do Ball and Beam, mas também serviu como uma excelente ferramenta de aprendizado para futuras aplicações em automação e engenharia de controle.

Perspectivas futuras incluem: Otimização do hardware, substituindo o servomotor por um atuador mais preciso. Melhoria no sistema de sensoriamento, explorando sensores ópticos ou de câmera para aumentar a precisão da detecção da posição da bola. Implementação de técnicas de controle mais avançadas, como controle adaptativo ou fuzzy logic, para melhorar a resposta do sistema a variações de carga e perturbações externas.

Por fim, o projeto demonstrou de forma clara como o controle PID pode ser aplicado com sucesso a sistemas físicos, consolidando conhecimentos práticos e teóricos sobre controle dinâmico e engenharia de sistemas.

REFERENCES

- [1] K. Ogata, *Modern Control Engineering*. Prentice Hall, 2010.
- [2] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.
- [3] N. S. Nise, *Control Systems Engineering*. Wiley, 2020.
- [4] J. D. P. Gene F. Franklin and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 2014.
- [5] R. C. Dorf and R. H. Bishop, *Modern Control Systems*. Pearson, 2017.
- [6] A. P. Hub, "Balance ball on a beam using pid." Accessed: January 28, 2025.
- [7] A. Kumar, *Introduction to Control Systems*. New Age International, 2015.
- [8] N. Instruments, "Pid theory explained." Accessed: January 28, 2025.
- [9] R. Ramesh and A. P. Vinod, "Optimization of pid parameters using genetic algorithms," *International Journal of Control Theory and Applications*, vol. 11, no. 4, 2018.
- [10] W. Messner and D. Tilbury, *Control Tutorials for MATLAB and Simulink: A Web-Based Approach*. Addison-Wesley, 2020.
- [11] Q. Su and Others, "Improving ball and beam stability using adaptive pid," *IEEE Transactions on Industrial Electronics*, 2019.