



UNIVERSIDADE FEDERAL DO PARÁ - CAMPUS TUCURUÍ

FACULDADE DE ENGENHARIA ELÉTRICA

LABORATÓRIO DE SISTEMAS DE CONTROLE

Professor: Cleison Daniel Silva

Curso: FEE 2020

Aluno 1: Jeimisson Junior Leão do Carmo

Matrícula: 202033940016

Aluno 2: Leonardo França da Cruz

Matrícula: 202033940010

Turma: 02

Bancada: 05

Experimento 1: SISTEMA CONTÍNUO DE PRIMEIRA ORDEM

Experimento 2: IMPLEMENTAÇÃO DO SINAL PRBS

Experimento 3: MÉTODO DE ZIEGLER-NICHOLS

RELATÓRIO DE EXPERIMENTOS

EXPERIMENTO 1 - SISTEMA CONTÍNUO DE PRIMEIRA ORDEM

1.1 Descrição do Experimento

O Experimento 1 visa a análise de um sistema de primeira ordem contínuo, empregando a bancada motor-gerador alimentada por uma fonte DC de 15 Volts, conforme ilustrado na Figura 1. Para executar o experimento, utilizaremos o Visual Studio Code (VSCode), uma plataforma de desenvolvimento avançada. A combinação desses elementos proporciona um ambiente controlado e eficiente para a coleta e análise de dados. Ao longo do procedimento, desde a configuração da bancada até a implementação de códigos no VSCode, serão registradas variáveis cruciais, permitindo uma compreensão detalhada do comportamento do sistema de primeira ordem contínuo.

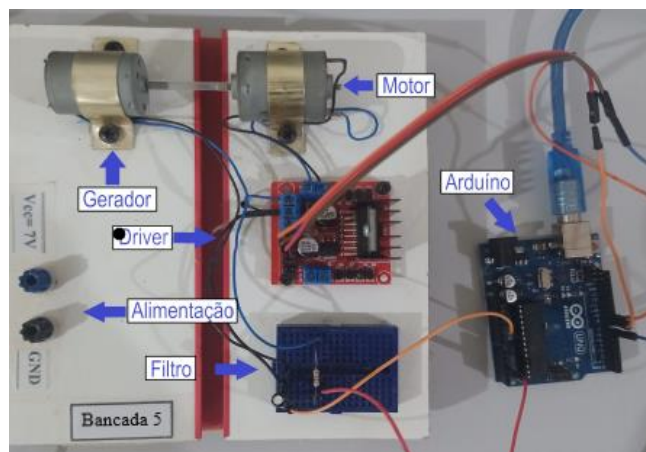


Figura 1: Bancada Motor-Gerador DC

Ademais, iremos coletar dados de uma onda quadrada, para encontrar uma função de transferência contínua de primeira ordem. Abordaremos a análise do comportamento dinâmico e estático de um motor de corrente contínua real, identificando suas características não-lineares, incluindo a modelagem matemática por meio de uma Função de Transferência de primeira ordem em torno de um ponto de operação específico, possibilitando a comparação entre a dinâmica observada e o modelo no domínio do tempo. A complexidade prática do motor, decorrente de elementos elétricos e mecânicos não ideais, bem como a presença de fenômenos físicos não considerados na equação diferencial, torna desafiadora a obtenção de um modelo linear ideal. Diante disso, buscamos um modelo matemático simplificado que capture de maneira aproximada as características essenciais do sistema, reconhecendo a dificuldade na determinação exata dos parâmetros e destacando a importância de um modelo linear para análises específicas em torno do ponto de operação.

Toda Função de Transferência de primeira ordem, expressa como

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

descreve um sistema dinâmico que pode ser caracterizado por dois parâmetros principais: K , o ganho estático, e τ , a constante de tempo. Supondo que $G(s)$ seja estável $\tau > 0$, consideramos uma entrada $x(t) = Au(t)$, em que $u(t)$ é o degrau unitário e A é uma constante. A saída correspondente é

$$y(t) = KA(1 - e^{-t/\tau})$$

válida para $t \geq 0$. O valor da saída em regime permanente ($t \rightarrow \infty$) é $y(\infty) = KA$, e o instante t_0 em que a saída atinge 63% de $y(\infty)$ é dado por:

$$0.6321 \cdot KA = KA(1 - e^{-t_0/\tau}) = \tau$$

Em outras palavras, o ganho estático K é a razão entre a saída em regime permanente e a amplitude da entrada, enquanto a constante de tempo τ representa o tempo necessário para alcançar 63% da resposta em regime permanente.

A Figura 2 esboça o procedimento para a identificação experimental de um modelo $G(s)$ de primeira ordem para o sistema em torno do ponto de operação (\bar{x}, \bar{y}) . Isso implica que tal modelo é válido quando $x(t)$ se aproxima de \bar{x} e $y(t)$ se aproxima de \bar{y} , ou seja, pequenas variações, representadas por $\Delta x(t) = x(t) - \bar{x} \approx 0$ e $\Delta y(t) = y(t) - \bar{y} \approx 0$, na entrada e na saída. Portanto, temos que:

$$G(s) = \frac{\Delta Y(s)}{\Delta X(s)} = \frac{K}{\tau s + 1}, \quad K = \frac{\delta y}{\delta x}, \quad \tau = t(63\%)$$

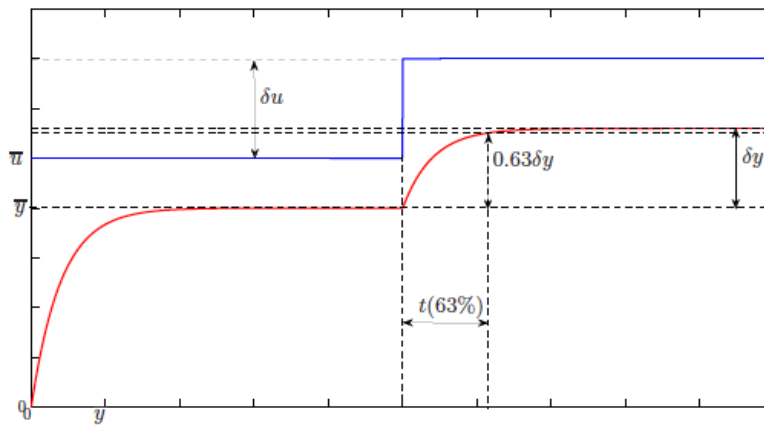


Figura 2: Identificação de primeira ordem em torno do ponto de operação.

O diagrama de blocos da Figura 3, delineia a estrutura fundamental dos sistemas de controle em malha fechada com realimentação negativa. Esse esquema evidencia quatro sinais essenciais: o sinal de referência $r(t)$, o sinal de erro de controle $e(t)$, o sinal de controle $u(t)$ e o sinal de saída $y(t)$.

Destacam-se também dois componentes cruciais: o **processo**, que representa a dinâmica a ser controlada, e o **controlador**. Neste experimento, o foco está na concepção, simulação e implementação dos controladores Proporcional (P) e Proporcional-Integral (PI) para a bancada Motor-gerador.

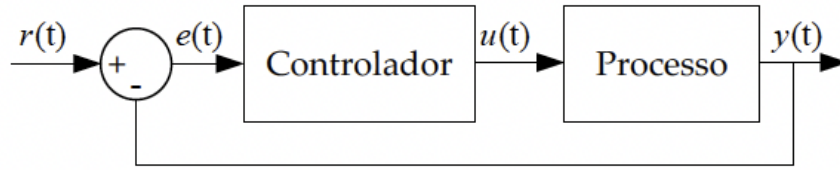


Figura 3: Diagrama de blocos em malha fechada com realimentação negativa.

O controlador P atua de maneira proporcional ao erro, garantindo que, em situações de erro significativo, a saída seja ajustada de forma proporcional. Embora efetivamente reduza o erro entre o valor desejado e o valor real a zero, o controlador P pode resultar em um pequeno erro estacionário, deixando uma discrepância mínima entre o setpoint e a resposta do sistema. Sua capacidade de resposta rápida a distúrbios é evidenciada pela prontidão em ajustar a saída proporcionalmente ao erro instantâneo em face de mudanças súbitas no sistema.

A função de transferência de malha fechada $H(s)$, considerando $G_c(s) = Kp$, com controlador proporcional, é:

$$H(s) = \frac{K_p G(s)}{1 + K_p G(s)}$$

Substituindo $G(s)$ em $H(s)$

$$H(s) = \frac{\frac{K_p K_m}{1 + K_p K_m}}{\frac{\tau}{1 + K_p K_m} s + 1} \quad (1)$$

Para uma redução de 15% no tempo de subida (T_{smf}):

$$T_{smf} = (1 - 15\%) \cdot T_s$$

$$\tau_{mf} = 2,2 T_{smf}$$

Substituindo τ_{mf} em (1) e isolando Kp :

$$Kp = \frac{\frac{\tau}{\tau_{mf}} - 1}{Km}$$

$$\tau_{mf} = \frac{\tau}{(1 + Kp Km)}$$

Função de Transferência de Malha Aberta:

$$G(s) = \frac{K_m}{\tau s + 1} = \frac{Kb}{s + b}$$

Em que $K = K_m$ e $b = \frac{1}{\tau}$

Função de Transferência de Controlador PI:

$$G_c(s) = \frac{K_c(s+a)}{s}$$

Em Malha Fechada:

$$H(s) = \frac{K_c K b (s + a)}{s^2 + (b + K_c K b) s + K_c K b a}$$

Função de Transferência de Referência Genérica:

$$H_{\text{ref}}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

em que: $a = \frac{\omega_n^2}{K_c \cdot K \cdot b}$ e $K_c = \frac{\frac{2\zeta\omega_n - 1}{b}}{K}$

Ultrapassagem percentual (%UP):

$$\%UP = 100 \times e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\zeta = \frac{-\ln(\%UP/100)}{\sqrt{\pi^2 + \ln^2(\%UP/100)}}$$

Tempo de subida: Tempo necessário para que a forma de onda vá de 0,1 a 0,9 do valor final:

$$T_r \approx \frac{1,8}{\omega_n}$$

Tempo de assentamento: Instante em que a resposta do sistema alcança e permanece dentro da faixa de 2% em torno do valor de regime permanente.

$$T_s = \frac{4}{\zeta\omega_n}$$

Para o critério de 1% , consideramos:

$$T_s = \frac{4,6}{\zeta\omega_n}$$

1.2 Procedimentos de Projeto

Ao iniciar o projeto, conectamos uma fonte de tensão DC ajustada para fornecer 15 volts à bancada e estabelecemos as conexões entre o Arduino e o VSCode para garantir a comunicação necessária. Para a coleta inicial de dados, escolhemos o ambiente Jupyter do Python. Durante a medição, configuramos um sinal de onda quadrada com um setpoint de 50% da alimentação, frequência de 0.5 Hz e amplitude de 1. Os resultados foram registrados e analisados, sendo visualizados graficamente na Figura 4.

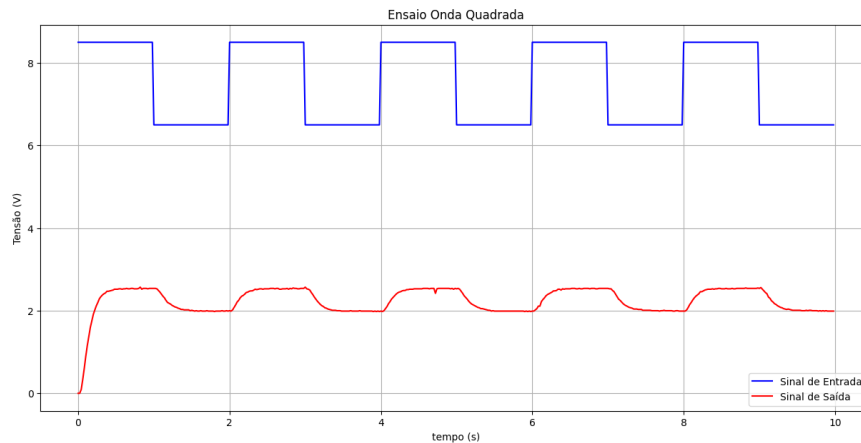


Figura 4: Ensaio de Onda Quadrada

Após calcular os níveis DC de entrada e saída como 7.5 e 2.271, respectivamente, procedemos à remoção desses componentes para proporcionar uma visualização gráfica mais clara, conforme evidenciado na Figura 5. Na Figura 6, ajustamos o intervalo de tempo para uma análise mais detalhada, visando determinar os parâmetros K_m e τ . Essa abordagem visa destacar as variações dinâmicas e os aspectos oscilatórios do sinal, aprimorando, assim, a capacidade de análise das características fundamentais do sistema.

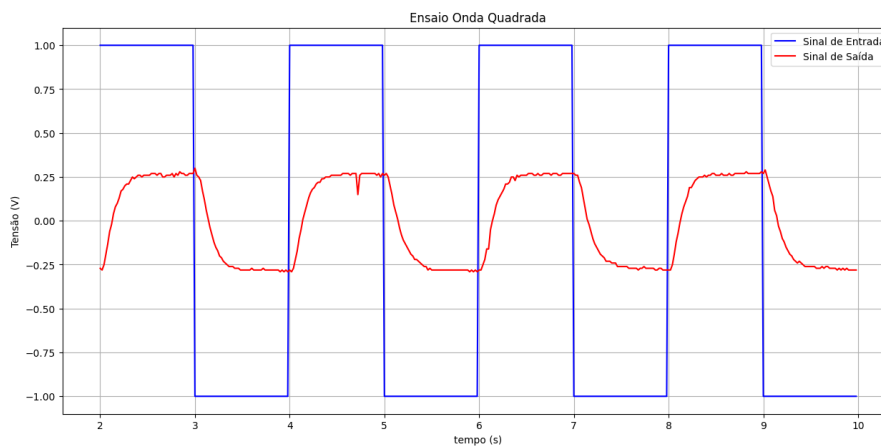


Figura 5: Ensaio de onda quadrada sem o nível DC.

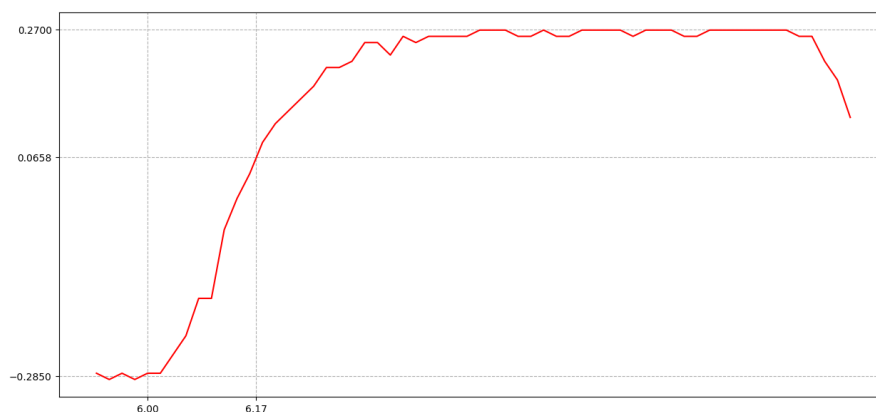


Figura 6: Visualização do sinal entre os instantes 5.9 e 7.1 s.

Com os dados extraídos do gráfico da Figura 6, identificamos os valores de τ e K_m , essenciais para a caracterização do sistema. Utilizando esses parâmetros, formulamos a função de transferência, oferecendo uma representação matemática que descreve de maneira concisa a relação entre a entrada e a saída do circuito.

$$G(s) = \frac{0.277}{0.17s + 1}$$

Com a função de transferência $G(s)$ determinada, desenvolvemos um modelo e conduzimos simulações, comparando os resultados com as medições reais da bancada. Essa análise permite avaliar a precisão do modelo na representação do comportamento do sistema, identificando discrepâncias e refinando o modelo conforme necessário para uma correspondência mais precisa entre a simulação e a realidade.

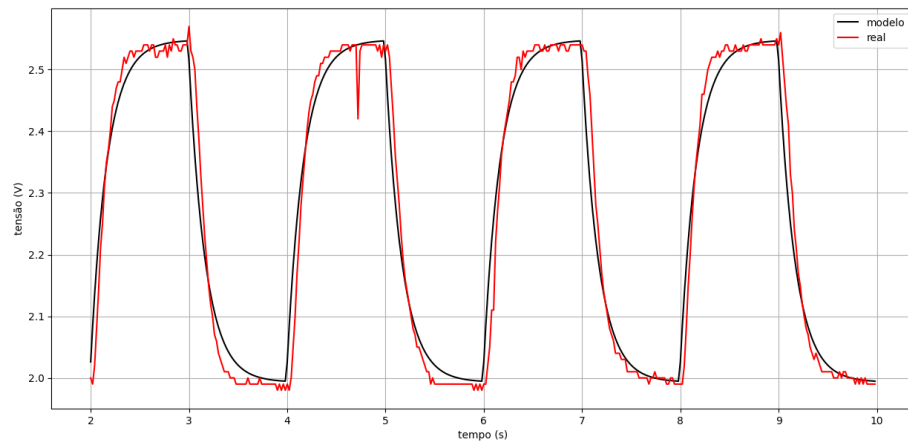


Figura 7: Comparação simulada do modelo com o real.

Para atender às especificações do projeto, inicialmente, no controlador proporcional (P), buscamos reduzir o tempo de subida em 15% ajustando o ganho proporcional K_p . No entanto, como o P sozinho pode resultar em erro em regime permanente, após esses ensaios e coleta de dados, vamos adicionar um termo integral (PI). Ajustando K_i no controlador PI, conseguimos eliminar o erro em regime permanente. Durante esse processo, observamos o impacto nos parâmetros de desempenho, especialmente no tempo de assentamento, que visamos reduzir para 0.8.

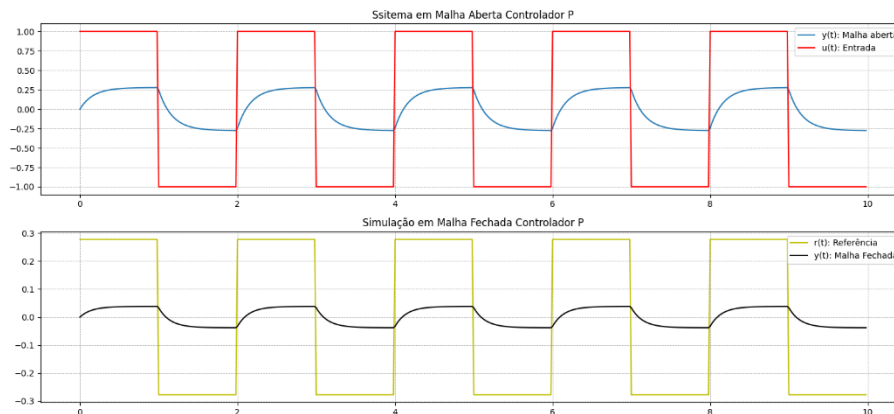


Figura 8: Simulação do Sistema em malha Aberta e Fechada para o controlador P.

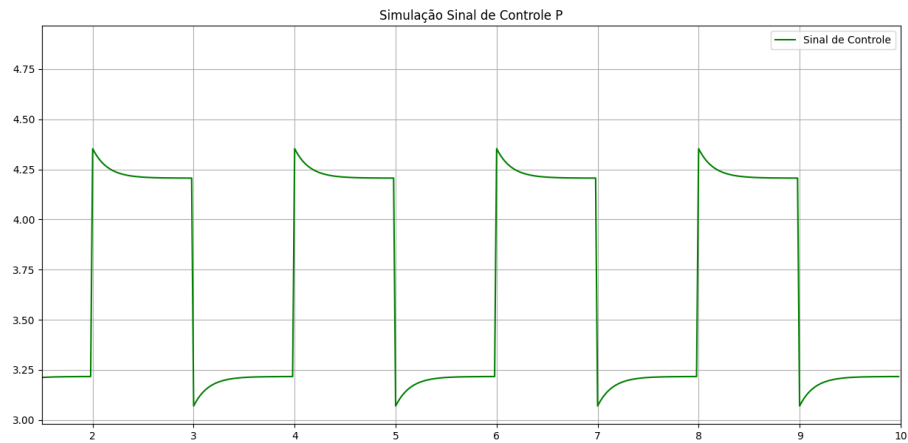


Figura 9: Simulação do sinal de controle Proporcional.

Ganho proporcional $K_p = 0.574$ e ganho DC = 0.137.

$$H_{mf}(s) = \frac{0.1592}{0.17s + 1.159}$$

E sinal de controle P:

$$C(s) = \frac{0.09752s + 0.5736}{0.17s + 1.159}$$

Para o controlador PI, iremos seguir passos parecidos, onde iremos simular o controlador para então poder implementar na bancada, e observando esse sinal para que não se possa queimar algum componente.

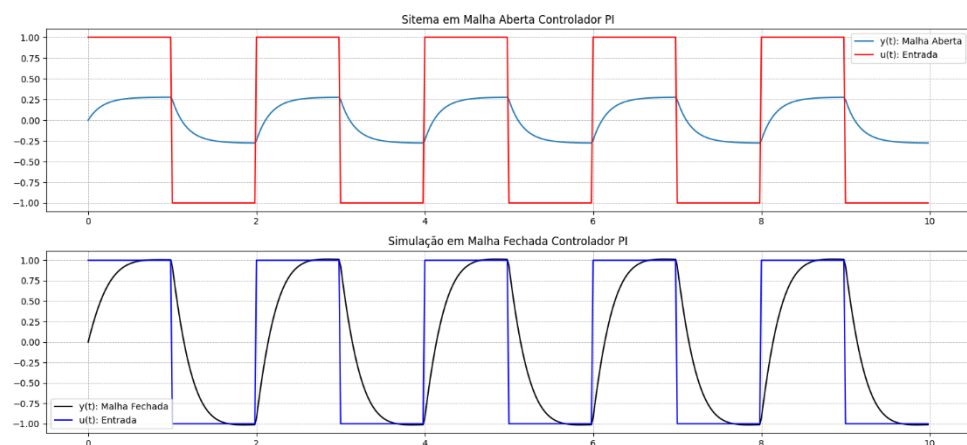


Figura 10: Simulação em malha Aberta e Fechada do controlador PI.

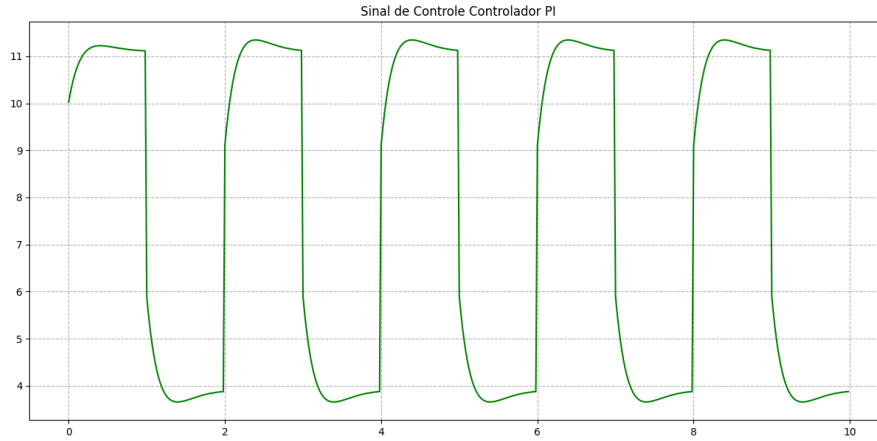


Figura 11: Sinal de controle simulado para o controlador PI.

$$C_{PI}(s) = \frac{2.523s + 19.07}{s}$$

$$H_{mfpi}(s) = \frac{0.17s + 5.291}{1.7s^2 + 1.7s + 5.291}$$

Para implementar o controlador PI na bancada, o primeiro passo é discretizar o controlador contínuo para o domínio discreto com um intervalo de amostragem $dt = 0.02$. A discretização do controlador permitirá sua implementação prática na bancada, garantindo que as características de desempenho desejadas sejam mantidas no domínio discreto.

$$C_{PI}(z) = \frac{2.713z - 2.332}{z - 1} \quad dt = 0.02$$

Onde iremos passar os valores de b_0 e b_1 que são 2.713 e -2.332 respectivamente, para a bancada 5. Os dados observados após a implementação dos controladores são mostrados nas Figuras 12 e 13:

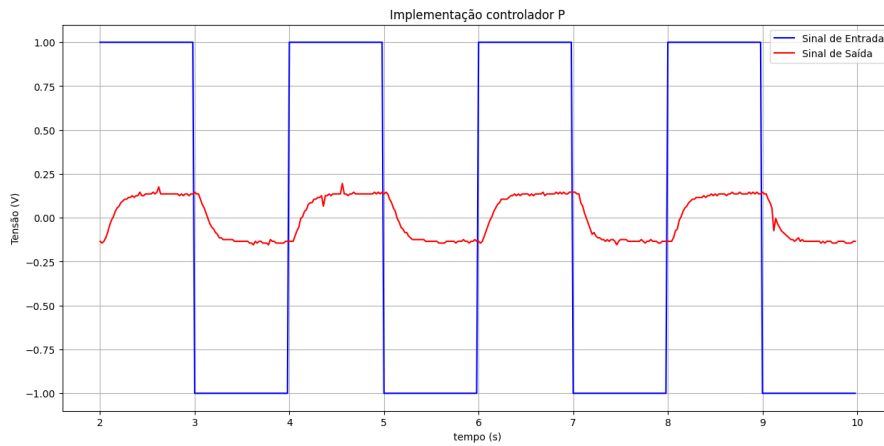


Figura 12: Implementação do controlador P na bancada

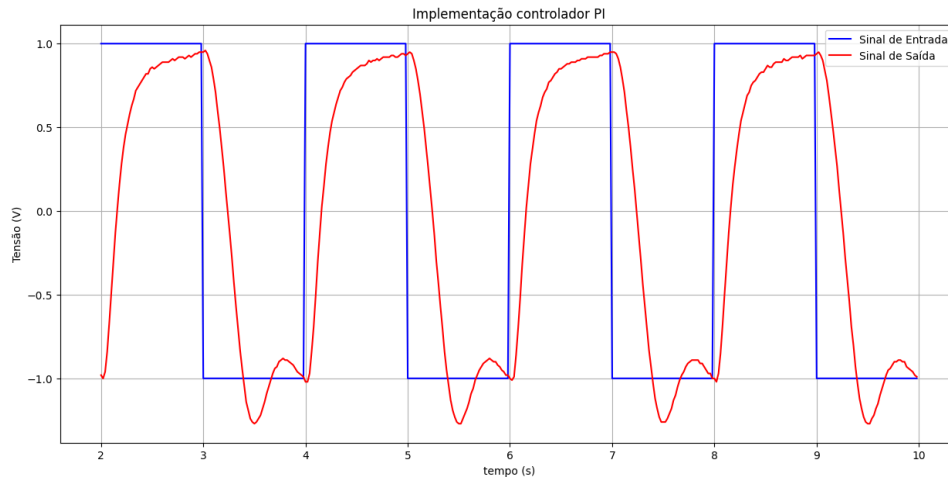


Figura 13: Implementação do controlador PI na bancada

É encorajador perceber que o sistema está comportado e respondendo aos controladores de maneira satisfatória. A implementação dos controladores, especialmente o PI com a discretização adequada, parece estar alinhada com as expectativas do projeto.

EXPERIMENTO 2 - IMPLEMENTAÇÃO DO SINAL PRBS

2.1 Descrição do Experimento

O sinal PRBS é uma sequência binária pseudoaleatória que bastante usada para testes de sistemas. Este sinal é projetado para se assemelhar a uma sequência aleatória, em engenharia o PRBS é frequentemente utilizado para testar sistemas dinâmicos. Dada a sua característica pseudoaleatória, são úteis para identificar comportamentos não lineares e para a análise de resposta ao degrau.

O objetivo deste experimento é testar como a bancada se comporta ao ser aplicado um sinal de entrada do tipo PRBS. Além disto, a equipe deve a partir dos dados coletados, projetar um modelo discreto de primeira e segunda ordem que se aproxime do sinal de saída real do sistema. Por fim devemos projetar o controlador Proporcional e o Proporcional integral para atender as especificações de projeto.

Para este experimento utilizamos a bancada como os motores, o microcontrolador Arduino, e uma fonte de bancada.

2.2 Procedimentos de Projeto

Primeiramente implementamos no python o código fornecido pelos monitores para aplicar o sinal de entrada PRBS na bancada. Os dados coletados foram salvos e armazenados em três variáveis, tempo, sinal de entrada e sinal de saída, como mostra a figura 14, utilizamos as bibliotecas do python para plotar o sinal de entrada e o sinal de saída.

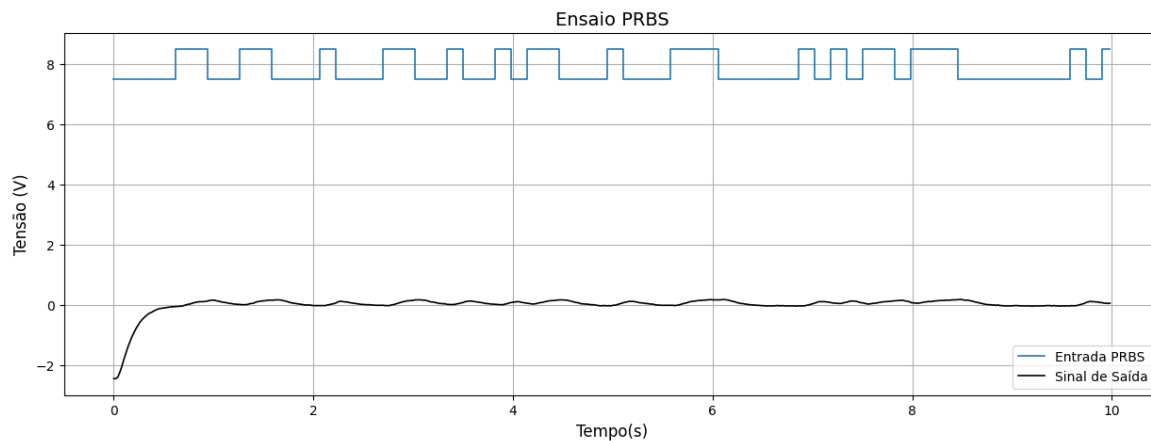


Figura 14 - Sinal de entrada e saída coletado.

Com base nesses dados e utilizando o material fornecido pelos monitores, a equipe primeiramente fez um modelo discreto de primeira ordem para tentar se aproximar do sinal de saída real. O resultado pode ser visto da figura 15.

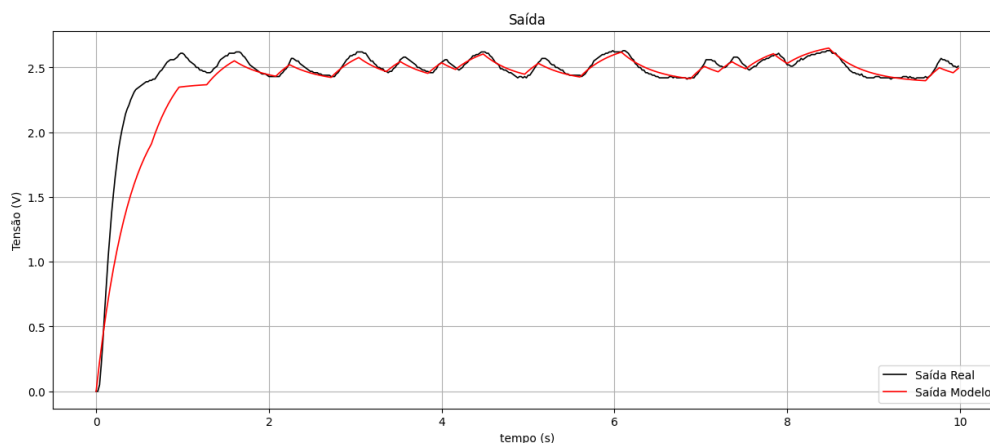


Figura 15 - Sinal real e modelo de primeira ordem.

Para uma melhor visualização podemos definir um intervalo de amostragem no gráfico e plotar novamente a comparação feita anteriormente.

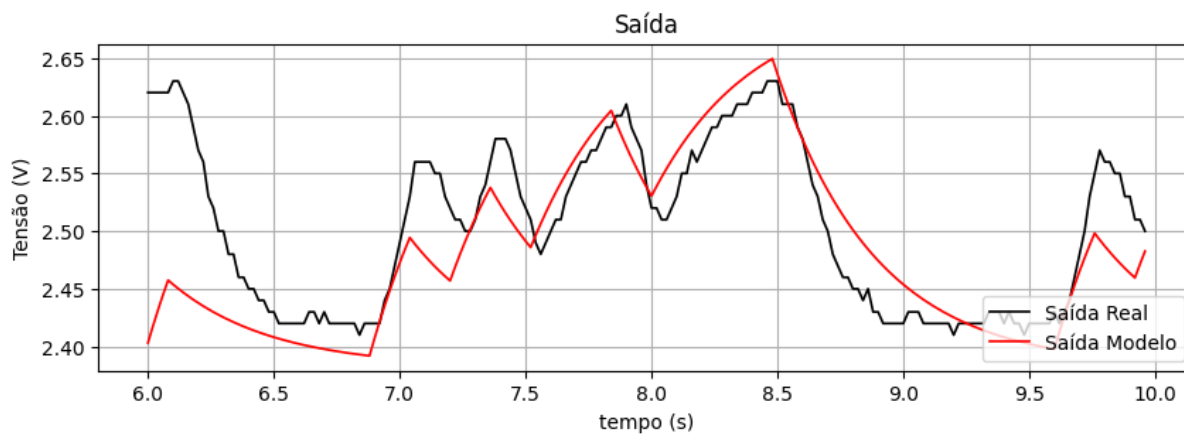


Figura 16 - Comparação em um intervalo amostrado.

Como podemos observar o modelo de primeira ordem até se aproxima bem do real em alguns pontos, mas ainda não é o desejado. Com isso partimos para o projeto do modelo de segunda ordem discreto, a comparação foi feita novamente e pode ser vista na figura 17.

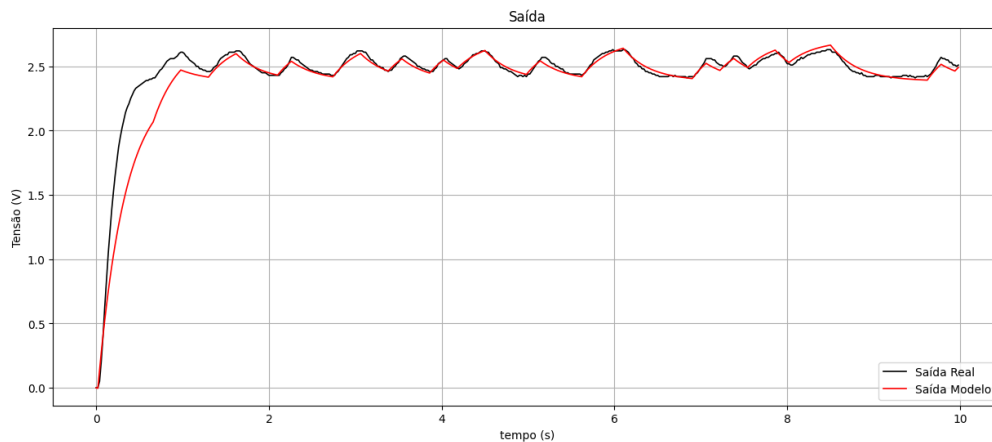


Figura 17- Modelo de segunda ordem.

Podemos observar que agora o modelo de segunda ordem conseguiu se aproximar melhor do sinal de saída real da planta, e por essa razão iremos utilizar este modelo da planta para o projeto dos controlados P e PI.

2.2.1 Controlador Proporcional

Para o projeto do controlador proporcional, primeiramente aplicamos uma resposta ao degrau para o modelo de segunda ordem em malha aberta, como mostra a figura 18. Com os dados em malha aberta, iniciamos o projeto do controlador de acordo com as especificações que previam uma redução de 15% do tempo de subida, erro nulo em regime permanente e não saturar o sinal de controle.

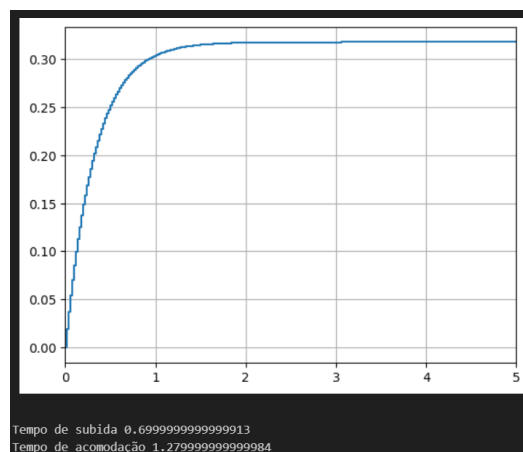


Figura 18 – Simulação em malha aberta.

Para este projeto, a equipe decidiu especificar uma ultrapassagem percentual em malha fechada de 4% e um tempo de acomodação de 0.7s.

Com os dados coletados e as especificações de projeto, a equipe optou por usar o LGR (Lugar Geométrico das Raízes) para projetar o controlador proporcional. Esta é uma técnica bastante utilizada pois permite uma compressão e visualização gráfica melhor de como a mudança de dos parâmetros afeta a resposta transitória e a sua estabilidade.

Com base no LGR e na observação dos polos em malha fechada do sistema, a equipe sintonizou de forma a atingir os requisitos de projeto, obtendo assim um ganho $K_p = 4.1$. O LGR para esse ganho é visto na figura 19.

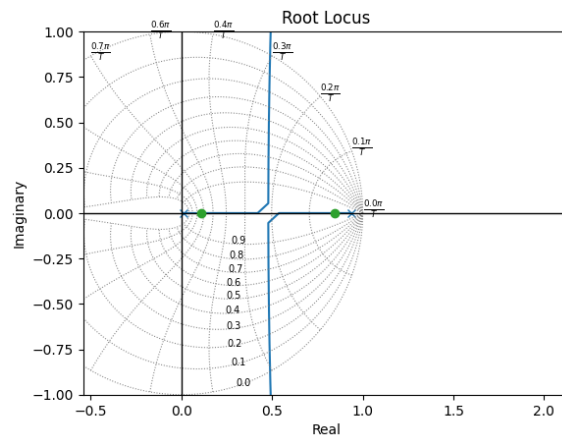


Figura 19 - LGR controlador proporcional

Por fim usando a função *step_response*, plotamos a resposta da função em malha fechada ao degrau unitário, como mostra a figura 20. Com a função *step_info*, observamos que redução de 15% do tempo de subida foi atendida na simulação e também apresentou erro nulo em regime permanente.

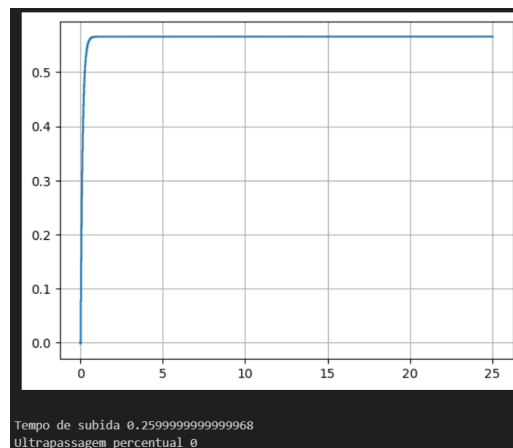


Figura 20 - Simulação em malha fechada.

Por fim com o ganho proporcional projetado, a equipe partiu para a implementação real na bancada. Definimos os níveis DC de entrada e saída, o ganho K_p e a alimentação da bancada, o resultado está na figura 21.

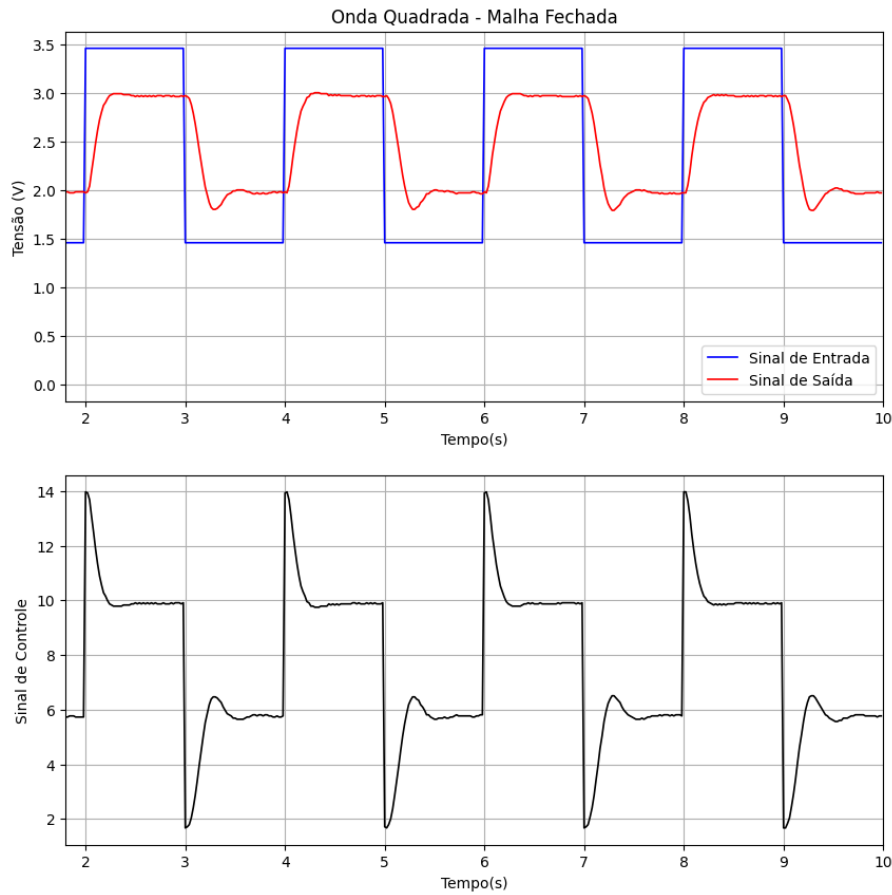


Figura 21 - Implementação do controlador Proporcional.

Podemos observar pelo gráfico que a bancada respondeu bem ao controlador proporcional e o sinal de controle apesar de estar ligeiramente alto não saturou, atendendo ao requisito de projeto.

Para encontrar o controlador proporcional-integrativo (PI) para um sistema de segunda ordem discreto sujeito a um sinal pseudoaleatório de sequência binária de bits (PRBS), começaremos considerando um modelo genérico de controlador no domínio s , representado por:

$$C_{pi}(s) = \frac{K(s + a)}{s}$$

Onde podemos identificar um valor atribuído a K igual a 4 e a igual a 10, considerando um coeficiente de amortecimento ζ de 0,5 e uma frequência natural w_n situada entre $\frac{0,1\pi}{T}$ e 0, conforme ilustrado na Figura 22.

Onde sua equivalente discreta é:

$$C_{pi}(z) = \frac{4,4z - 3,6}{z - 1}$$

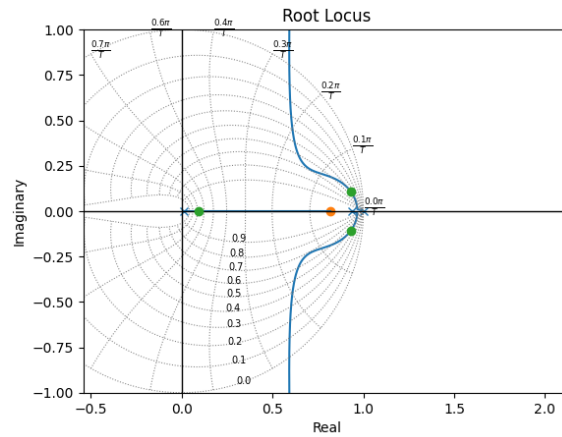


Figura 22: LGR do sistema de segunda ordem discreto.

A simulação do sistema em malha fechada e o sinal de controle foram representados pela Figura 23. Nessa representação, notamos que o sinal de controle, apesar de apresentar uma amplitude elevada, não atingiu a saturação, atendendo assim às especificações estabelecidas.

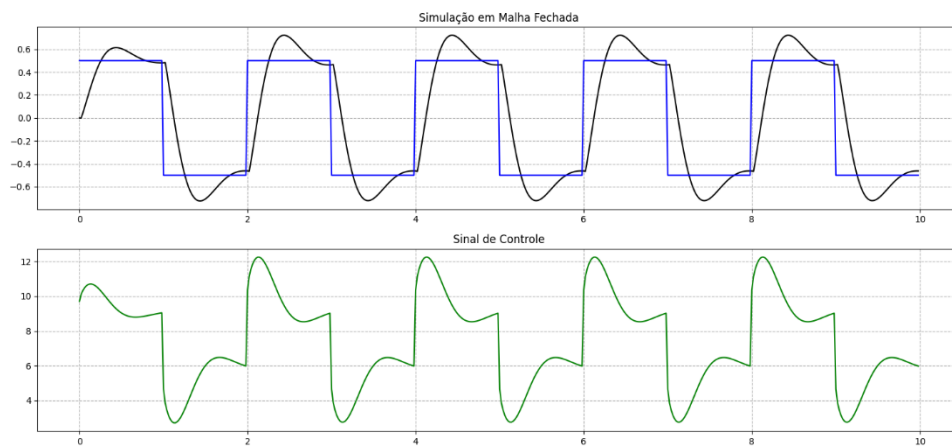


Figura 23: Simulação do sinal de controle e do sistema em malha fechada.

Após obter esses valores para o controlador $C_{pi}(z)$, iremos implementar os valores de $b_0 = 4,4$ e $b_1 = 3,6$ na bancada 5 para se observar os sinais de saída. Podemos observar na figura 24, os valores para o controlador foram satisfeitos, observado pelos sinais de saída após a implementação, pois mesmo tendo uma leve diferença comparado com o simulado, é notório que o sistema respondeu bem ao controlador projetado.

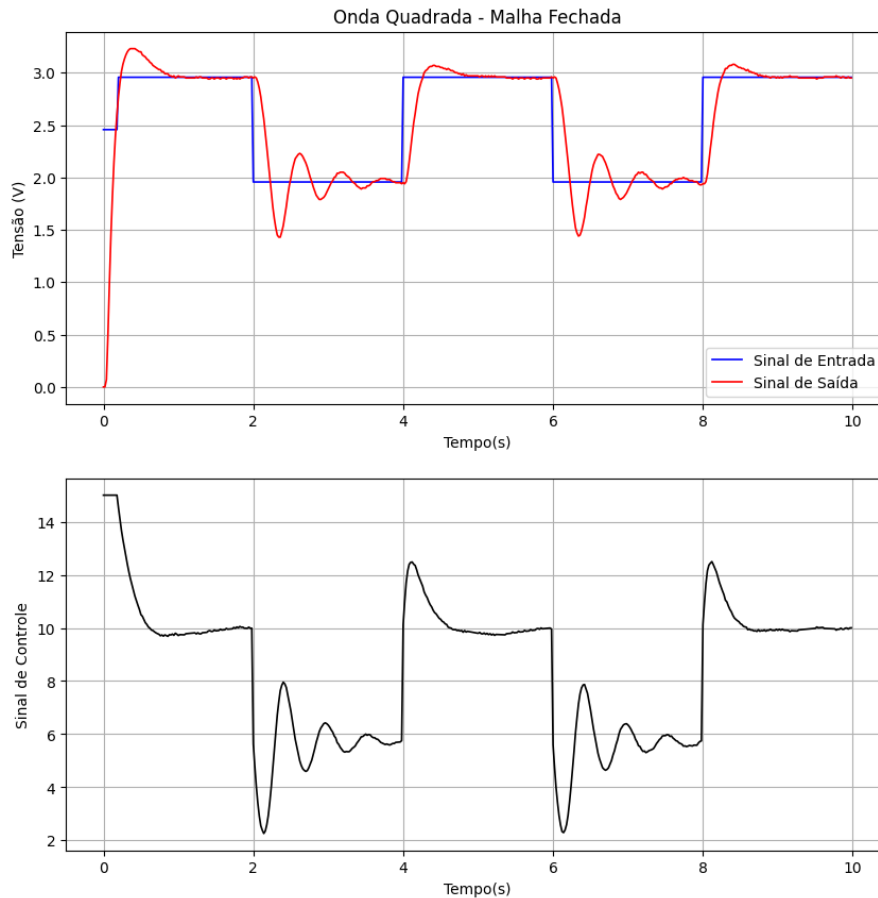


Figura 24: Sinais obtidos com a implementação do controlador PI.

EXPERIMENTO 3 - MÉTODO DE ZIEGLER-NICHOLS

3.1 Descrição dos Experimentos

O método Ziegler – Nichols, é um ajuste empírico de controladores, utilizado para determinar os parâmetros proporcionais, integrais e derivativos de um PID. Este método basicamente consiste em aplicar uma resposta ao degrau a planta e analisar a resposta para obter os parâmetros do controlador que se deseja projetar.

Existem dois métodos propostos por Ziegler – Nichols, para este experimento será utilizado apenas o método da resposta ao degrau para encontrar os parâmetros do controlador. Os ganhos calculados a partir deste método são o ponto de partida para procedimentos de ajustes usado por fabricante e por indústrias de processos.

Muitas plantas presentes em indústrias de processos, tem resposta ao degrau em forma de S, como mostra a figura 25, isso significa que esse sistema é estável em malha aberta, sem integrador e sem polos complexos dominantes.

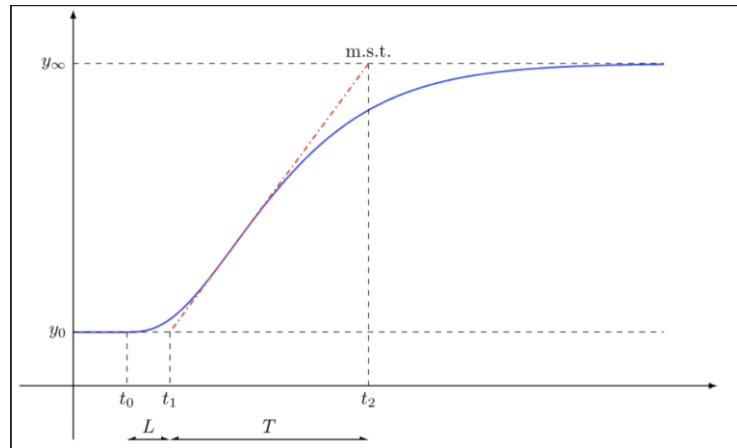


Figura 25 - Método da resposta ao degrau.

Para implementar esse método no experimento, devemos:

- Com a planta em malha aberta, deixe a planta funcionar em condições normais de operação com entrada constante u_0 e saída em estado estacionário y_0 .
- No momento inicial t_0 aplique uma mudança de etapa á planta, de u_0 para u_∞ .
- Registre a resposta de saída para obter a curva de reação. Na figura 20, o mst representa a tangente máxima da inclinação.
- Calcular:

$$K = \frac{y_\infty - y_0}{u_\infty - u_0}, L = t_1 - t_0, T = t_2 - t_1$$

Onde K é o ganho do sistema, L e T são chamados de tempo morto aparente e constante de tempo aparente. Os parâmetros do PID são calculados seguindo as equações da tabela 1.

Controlador	K_p	T_i	T_d
P	$\frac{T}{KL}$	-	-
PI	$\frac{0.9 \cdot T}{KL}$	$3L$	-
PID	$\frac{1.2 \cdot T}{KL}$	$2L$	$0.5L$

Tabela 1 - Ganhos do controlador PID.

O objetivo deste experimento é projetar de forma empírica os controladores P e PI, usando o método da resposta ao degrau. Para este experimento foram usados basicamente os mesmos equipamentos utilizados anteriormente no experimento 1, tendo em vista que iremos aplicar um degrau a planta.

3.2 Procedimentos de Projeto

Primeiramente a equipe utilizou o código do motor gerador para aplicar um sinal do tipo degrau na bancada. Após o ensaio, os dados foram coletados e salvos para serem usados no experimento.

Com os dados, iniciamos o projeto dos parâmetros dos controladores P e PI, estes que foram armazenados em três variáveis para facilitar a visualização dos dados, neste caso foram nomeadas de tempo, sinal de entrada e sinal de saída. Plotando o sinal de entrada e o sinal de saída em função da variável tempo temos o seguinte resultado:

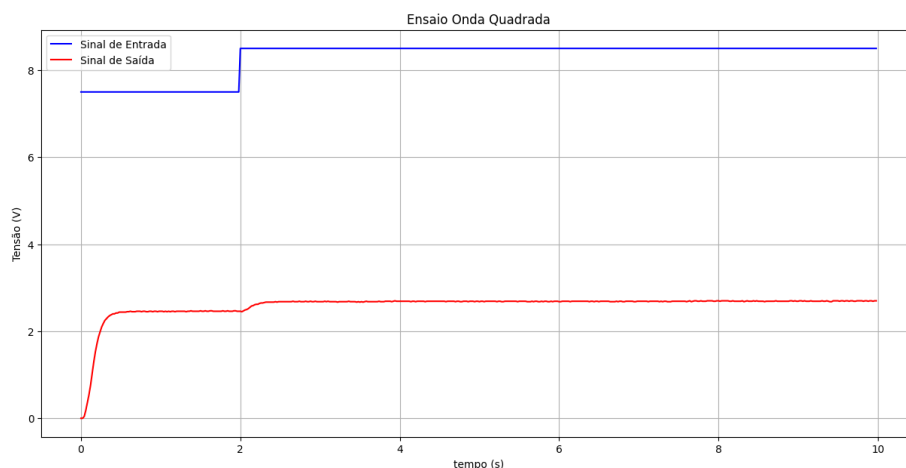


Figura 26 - Sinal de entrada e saída.

Para determinar os parâmetros do controlador, precisamos observar apenas um degrau do sinal de saída e com base em sua forma definir os valores para encontrar K, L e T. Desta forma implementamos no código uma janela de observação para observar o degrau q se inicia em 2s, assim escolhemos uma janela que começa em 1.7s e vai até 3.5s.

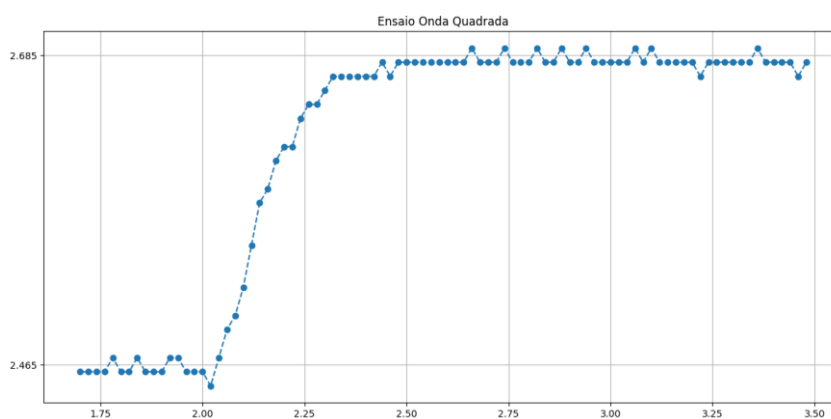


Figura 27 - Sinal de saída para uma janela de tempo.

Observando o gráfico já é possível determinar os instantes y_0 e y_{∞} , pois basta marcar onde o sinal se encontra antes da bancada ser excitada pelo degrau e depois em regime permanente, como mostra a figura 27.

Para os próximos pontos precisamos primeiramente definir a reta tangente de máxima inclinação. Esta reta pode ser encontrada de duas maneiras, a primeira é observando diretamente no gráfico para tentar ver onde essa reta se encontra, o que pode levar erros de imprecisão. Outra forma

mais eficaz é elaborar um código para realizar um método numérico para encontrar a derivada onde ocorre a maior taxa de inclinação. O grupo utilizou os dois métodos que para determinar a curva, comparando posteriormente os resultados vimos que a diferença entre um e outro era relativamente pequena. Esta diferença se mostrou significativa no momento de encontrar os ganhos do controlador e por essa razão optamos por usar a reta obtida pelo método numérico.

Implementando o código obtivemos o seguinte resultado da figura 28, porém analisando os valores dos ganhos dos controladores, percebemos que estavam muito altos e decidimos escolher uma outra reta tangente com uma inclinação maior, desta forma a equipe conseguiu sintonizar de maneira mais adequada os ganhos do controlador. definimos o instante $t_1 = 2.09$ que ocorre após se passar um período de amostragem da intercessão de y_0 com a reta tangente, e o instante $t_2 = 2.18$, também definimos o instante $t_0 = 2$ que representa o um momento antes da bancada ser excitada pelo degrau.

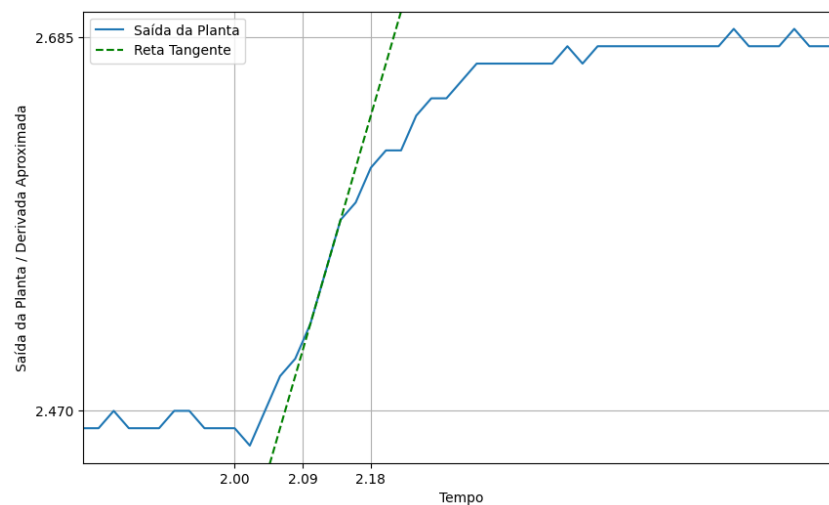


Figura 28 - Gráfico com a tangente de inclinação.

Assim substituindo nas equações temos:

$$K = 2.685 - 2.470$$

$$L = 2.09 - 2$$

$$T = 2.18 - 2.09$$

3.2.1 Controlador Proporcional

Com base nesses dados projetamos o ganho K_p do controlador Proporcional, dado por:

$$K_p = 4.65$$

Com o ganho do controlador definido já poderíamos ter partido para a implementação do controlador na bancada, porém a equipe decidiu por primeiramente simular a resposta do sistema em malha fechada para um degrau unitário e observar se o sinal de controle estava ou não saturando. Para realizar essa simulação fizemos um modelo discreto de segunda ordem da planta com base nos dados de saída dos sistemas, como mostra a figura 29.

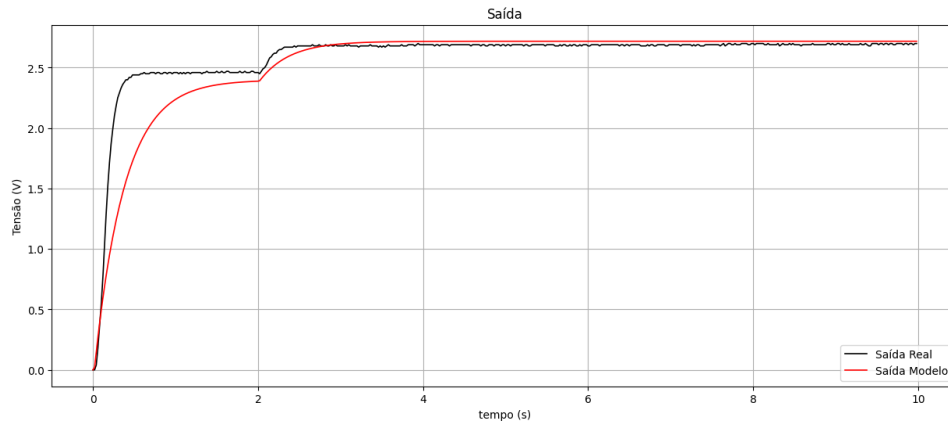


Figura 29- Modelo de segunda ordem.

Podemos ver que o modelo não se aproximou bem do sinal de saída real, uma solução seria criar um modelo de terceira ordem que provavelmente se aproximaria melhor, mas como estamos apenas interessados em observar como a planta está se comportando para esse controlador, este modelo deve ser suficiente.

Deste modo, projetamos a função de transferência do controlador P e em seguida fizemos a função de transferência de malha fechada. O resultado da simulação do sistema respondendo a um degrau é vista na figura 30. Como podemos observar o controlador proporcional não é capaz de corrigir o erro em regime permanente, mas respondeu bem ao degrau.

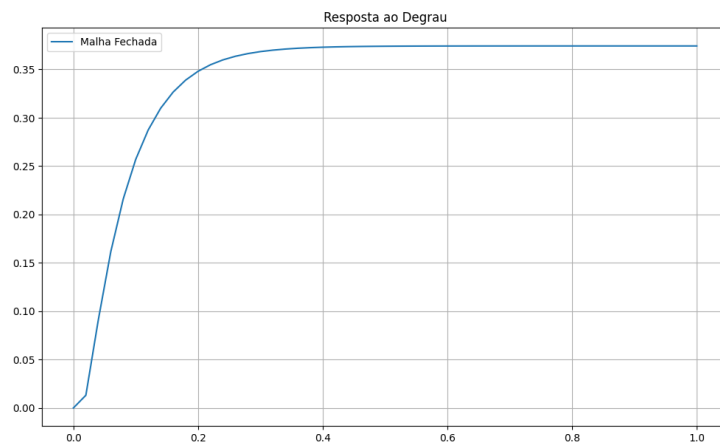


Figura 30 - Resposta ao degrau.

Agora observando o sinal de controle da figura 31, vemos que não está saturando, tendo em vista que a bancada é de 15 V.

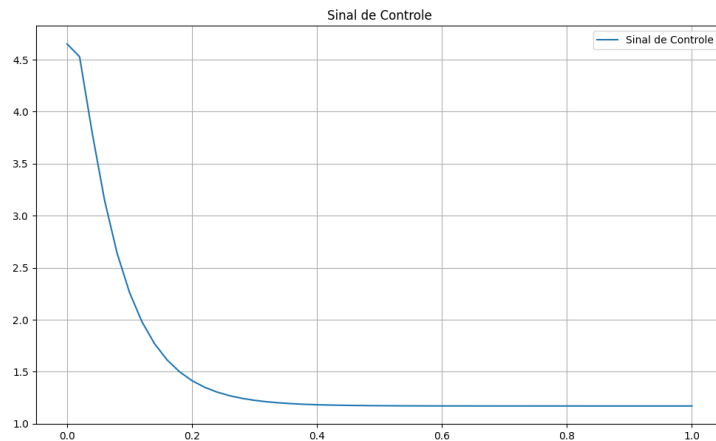


Figura 31 - Sinal de controle.

Visto que as simulações mostram que o controlador está coerente, partimos para a implementação na bancada, colocamos o ganho K_p no código, os níveis DC de entrada e saída e o setpoint. Os resultados estão na figura 32.

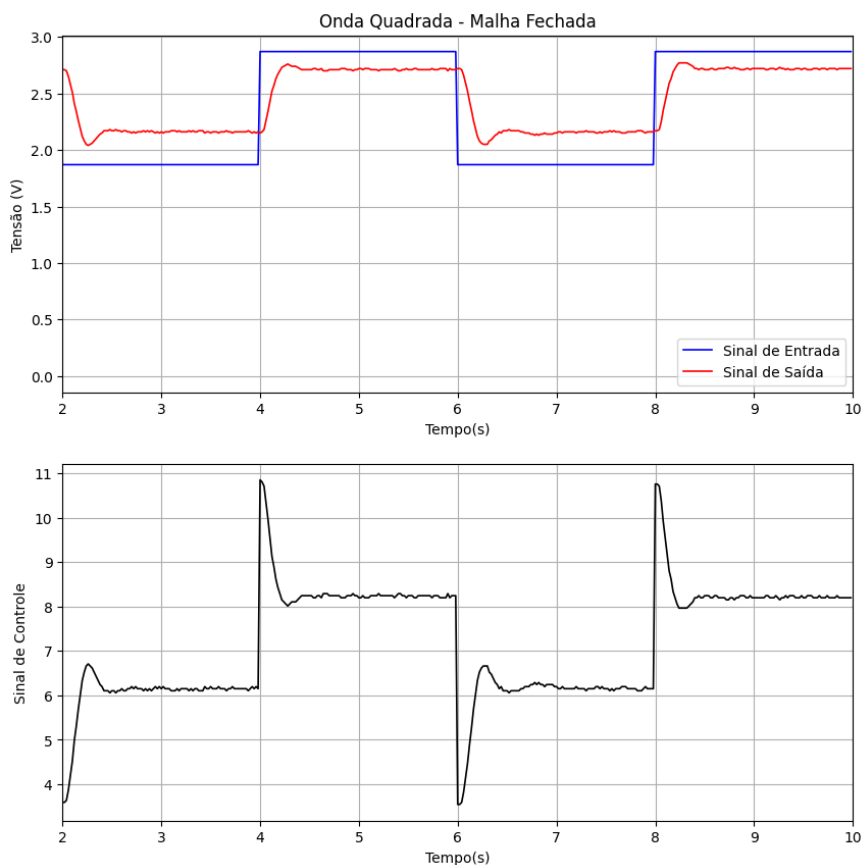


Figura 32 - Implementação controle proporcional.

Em uma primeira análise, notamos que o controlador atuou de maneira satisfatória na bancada e o sinal de controle, apesar de estar relativamente alto, não ficou saturado.

3.2.2 Controlador Proporcional Integral

Seguindo os mesmos passos descritos anteriormente, projetamos o controlador PI, assim como no controle P, realizamos uma simulação previa do sistema em malha fechada e também observamos

o sinal de controle. Após calcular a função do controlador e discretizar esta, realizamos a simulação da resposta ao degrau, vista na figura 33. Notamos que para o controle PI o sistema se estabiliza em 1 no regime permanente, sem ter um Overshoot elevado o que caracteriza um sistema sub-amortecido.

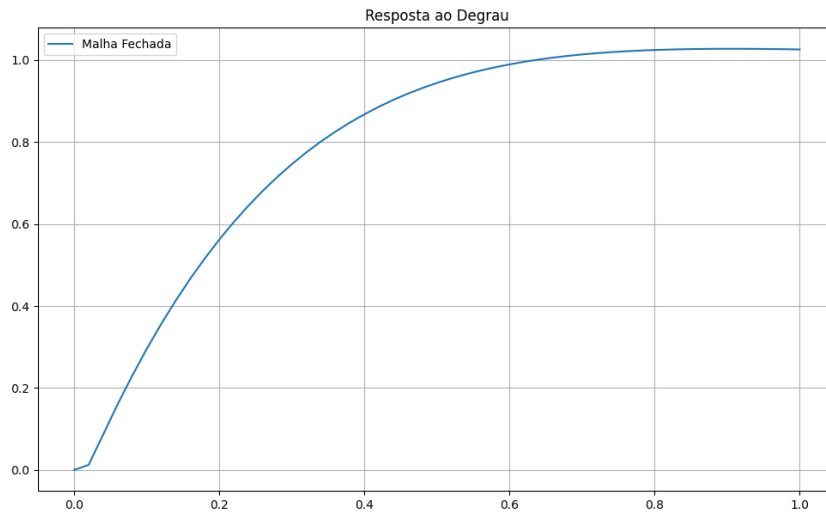


Figura 33 - Resposta ao degrau PI.

Agora analisando o sinal de controle da figura 34, observamos que o sinal de controle também não está saturando para o controlador PI.

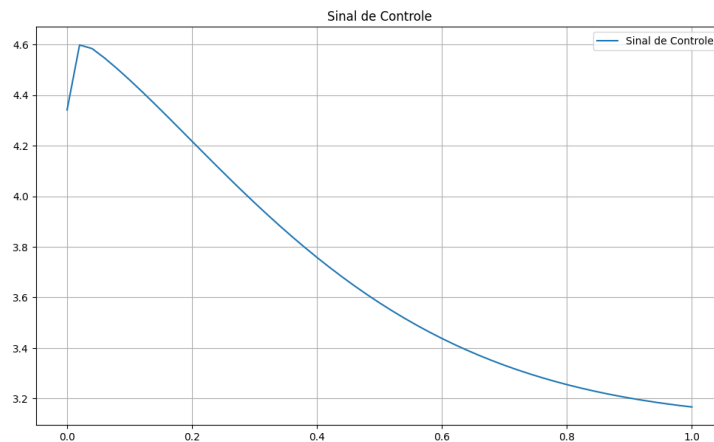


Figura 34 - Sinal de controle PI.

A equação discreta do controlador PI é dada por:

$$\frac{4.341z - 4.031}{z - 1}$$

$$dt = 0.02$$

Com isso definimos o coeficiente $b_0 = 4.341$ e $b_1 = -4.031$ para a implementação da bancada. O resultado é mostrado na figura 35.

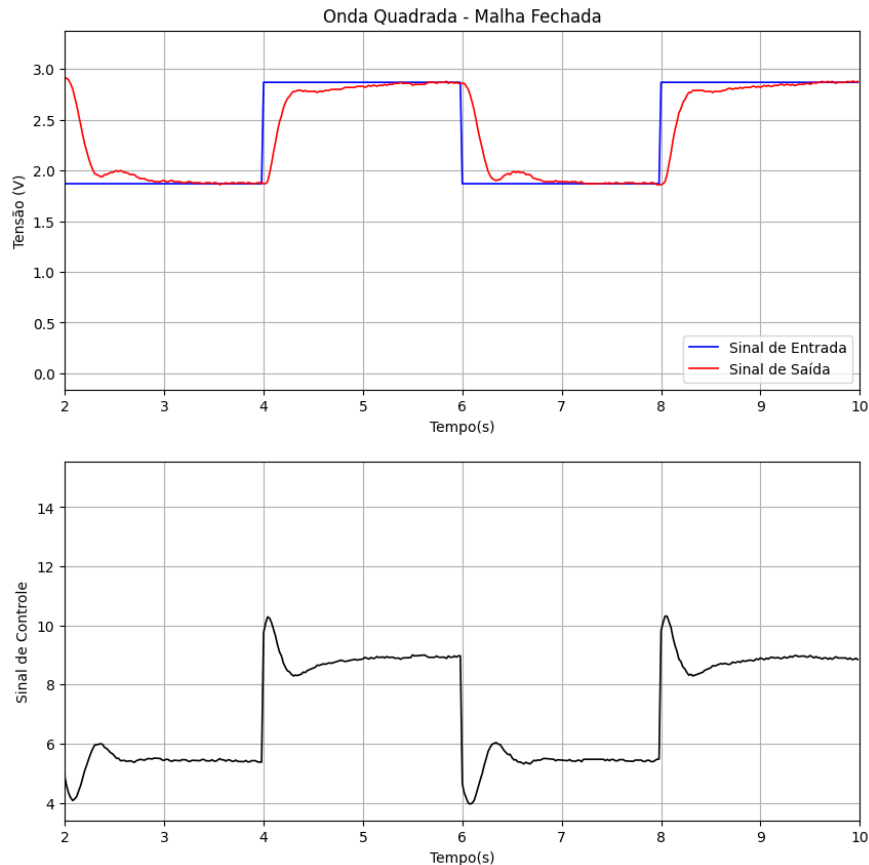


Figura 35 - Implementação Controle PI

Como podemos observar pelo gráfico o controlador PI respondeu melhor a planta, inclusive diminuindo o sinal de controle necessário, em comparação ao controlador proporcional.

RESULTADOS E DISCURSSÕES

Analisando os três resultados para controlador proporcional, podemos inferir que o melhor controle foi obtido através do método de Ziegler-Nichols, pois mostrou ter uma resposta mais rápida ao controlador e o sinal de controle tem uma amplitude melhor em comparação aos outros controladores do tipo P implementados na bancada.

Em relação ao controlador PI, comparando os resultados notamos que o método Ziegler-Nichols tem uma melhor resposta para a bancada, e seu sinal de controle é bem menor em comparação ao sistema contínuo de primeira ordem e o controlador PI obtido através do sinal PRBS.

CONSIDERAÇÕES FINAIS

Nos experimentos conduzidos, observamos uma notável semelhança entre a simulação computacional e a implementação na bancada, embora tenham surgido algumas divergências nos sinais obtidos. A execução dos códigos e a geração dos gráficos foram bem-sucedidas, atendendo às condições de projeto estabelecidas para ambos os controladores. Entretanto, durante a implementação prática, encontramos desafios específicos, como a incapacidade do sinal de entrada em acompanhar o controlador devido à sua rapidez, e este problema foi abordado por meio de uma revisão e refinamento do projeto do controlador PI na fase contínua.

Além disso, vale ressaltar que a abordagem do método de Ziegler-Nichols revelou-se particularmente mais simples que os outros 2 para os projetos dos controladores proporcional P e PI, devido a sua falta de se necessitar de um modelo, sendo então apenas analisando o gráfico. Esta facilidade decorreu da não necessidade de derivar uma função de transferência, sendo possível realizar a análise gráfica dos dados coletados na bancada. Esse processo seguiu um padrão semelhante ao ensaio de onda quadrada, facilitando a determinação dos parâmetros essenciais, como a constante de ganho k_m e a constante de tempo τ .

Essas experiências contribuíram não apenas para a validação dos modelos teóricos por meio da comparação entre simulação e implementação prática, mas também destacaram a importância de ajustes específicos na implementação para garantir a estabilidade e desempenho desejados. O emprego do método de Niegler-Nichols, por sua vez, simplificou significativamente o processo de ajuste dos controladores, proporcionando uma abordagem prática e fácil para alcançar os objetivos de controle estabelecidos.