



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
SISTEMAS DE CONTROLE II - ELE0522

Docente:

Gabriel Pereira de Oliveira

Discentes:

Douglas Wilian Lima Silva
Gutembergue Ferreira da Silva
Jeffet Matheus Cardoso da Silva
Pedro Artur Fernandes Varela de Lira
Roger José Zacarias da Conceição

**DESENVOLVIMENTO DO SISTEMA DE
CONTROLE**

Natal - RN
Setembro de 2023

Sumário

1	Referencial Geral do Problema	3
1.1	Sistemas em malha fechada e em malha aberta	3
1.2	Método do lugar geométrico das raízes	3
1.3	O problema	5
2	Desenvolvimento	6
2.1	Tanque 1	6
2.1.1	Modelagem	6
2.1.2	Linearização	7
2.1.3	Parâmetros do sistema de controle	8
2.1.4	Controlador PI	8
2.2	Tanque 2	9
2.2.1	Modelagem	10
2.2.2	Linearização	10
2.2.3	Parâmetros do sistema de controle	11
2.2.4	Controlador PID	12
2.3	Implementações	17
2.3.1	Controlador PI - Tanque 1	17
2.3.2	Controlador PID - Tanque 2	22
2.4	Códigos	27
3	Considerações finais	34
4	Referencial teórico	36

1 Referencial Geral do Problema

1.1 Sistemas em malha fechada e em malha aberta

Os sistemas podem ser de malha aberta, quando o controlador não utiliza sinais de saída para efetuar o controle do processo, ou de malha fechada, quando o sinal de saída é usado para modificar sinais internos, processo ao qual denominamos realimentação.

Neste sistema, a realimentação negativa é caracterizada pela equação que descreve o sinal resultante da realimentação, o qual é chamado sinal de erro, dado pela diferença entre o sinal de referência, $R(s)$, e o sinal de saída $X(s)$ da função de transferência $H(s)$. No domínio da frequência, temos:

$$E(s) = R(s) - X(s) = R(s) - H(s)Y(s)$$

Da equação acima, é possível perceber que o sinal de entrada da planta depende da leitura de seus sinais de saída. O bloco $H(s)$ opera a saída do sistema, $Y(s)$, e resulta em $X(s)$. Este bloco significa a função de transferência de um instrumento de medida ou um sensor, por exemplo.

No controle em malha aberta, temos a seguinte função de transferência:

$$FTMA(s) = \frac{X(s)}{E(s)}$$

No controle em malha fechada, temos a seguinte função de transferência:

$$\begin{aligned} Y(s) &= G(s)E(s) = G(s)(R(s) - X(s)) = G(s)(R(s) - H(s)Y(s)) \\ \Rightarrow Y(s) &= G(s)R(s) - G(s)H(s)Y(s) \\ \Rightarrow (1 + G(s)H(s))Y(s) &= G(s)R(s) \\ \Rightarrow FTMF(s) &= \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \end{aligned}$$

1.2 Método do lugar geométrico das raízes

O método do Lugar das Raízes consiste em um gráfico no qual, a partir da posição dos polos e zeros do sistema de controle em malha aberta, é possível visualizar o comportamento dos polos de malha fechada no plano complexo, quando altera-se o ganho K de malha aberta.

Seja o sistema do diagrama de blocos apresentado, definido por $FTMF(s)$. Os polos de malha fechada do sistema serão as raízes da seguinte equação:

$$1 + G(s)H(s) = 0$$

Assim, o Lugar Geométrico das Raízes é o conjunto das curvas do plano complexo onde podem estar os polos de malha fechada, que satisfazem a equação anterior. Podemos reescrever a equação acima da seguinte maneira:

$$G(s)H(s) = -1 + j0$$

Vejamos que esta é a função de transferência de malha aberta, apresentada anteriormente. Da quinta equação representada anteriormente, extraímos duas condições para que um ponto de teste pertença ao lugar das raízes, isto é, para que este ponto seja um lugar possível para um polo de malha fechada. Uma delas é a condição de módulo, na qual ocorre:

$$|G(s)H(s)| = 1$$

A segunda, é a condição de fase, na qual ocorre:

$$\angle G(s)H(s) = 180^\circ \pm r360^\circ$$

Sendo “r” um número inteiro positivo ou nulo, o que nos dá uma condição de fase que requer um múltiplo ímpar de 180° .

Paralelamente à quinta equação, podemos escrever a $FTMA(s)$ na forma de polos e zeros:

$$G(s)H(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

Logo, teremos as seguintes condições de módulo e de fase, respectivamente.

$$|G(s)H(s)| = K \frac{|s - z_1||s - z_2| \dots |s - z_m|}{|s - p_1||s - p_2| \dots |s - p_n|} = 1 \Rightarrow K = \frac{|s - p_1||s - p_2| \dots |s - p_n|}{|s - z_1||s - z_2| \dots |s - z_m|}$$

$$|G(s)H(s)| = |(s - z_1)| + |(s - z_2)| + \dots + |(s - z_m)|$$

$$- (|(s - p_1)| + |(s - p_2)| + \dots + |(s - p_n)| + (180^\circ \pm r360^\circ))$$

Em resumo, o segmento que contém um ponto de teste pertence ao lugar das raízes se o número de polos ou zeros reais à direita do ponto de teste for ímpar.

No que concerne aos procedimentos para o traçado do lugar das raízes, temos as seguintes especificações.

- I. O número de ramos do lugar das raízes é igual ao número de polos de malha aberta.
- II. Quanto à simetria, o lugar das raízes é sempre simétrico em relação ao eixo real.
- III. Os polos e zeros complexos não afetam a condição de fase.

Contudo, um polo ou zero real de malha aberta contribui com 180° quando à direita de um ponto de teste e não altera a condição quando à esquerda de um ponto de teste. IV. O lugar das raízes inicia nos polos de malha aberta e termina nos zeros de malha aberta. V. Normalmente, o número de polos “n” é maior que o número de zeros “m”. Assim, “m” ramos terminam em zeros e “n - m” ramos vão até infinito, seguindo assíntotas.

1.3 O problema

Com base no apresentado, o objetivo do trabalho é realizar o controle do sistema de bombeamento a seguir utilizando as estratégias de controle apresentadas com os controladores da família PID.

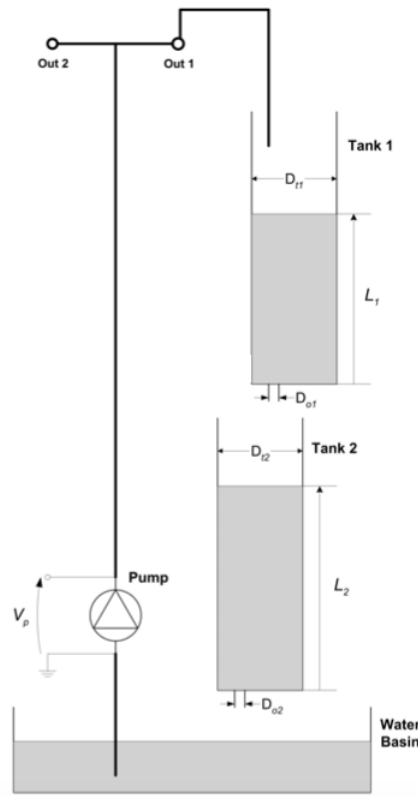


Figura 1: Configuração do problema.

Basicamente o sistema atua de forma que a bomba puxa o fluido do reservatório e o insere no tanque 1, que por sua vez, possui um orifício de saída para o tanque 2. O objetivo é o controle de altura do fluido tanto no tanque 1, quanto no tanque 2, utilizando a variação de tensão da bomba.

Os requisitos a serem atendidos pelo projeto são:

- **(a) Cenário 1** - regular o nível do tanque 1 em 15cm (sem erro em regime), considerando um $T_{s2\%} = 5s$ e um $PO\% = 11\%$;

- **(b) Cenário 2** - regular o nível do tanque 2 em 15cm (sem erro em regime), considerando um $T_{s2\%} = 20s$ e um $PO\% = 10\%$

2 Desenvolvimento

2.1 Tanque 1

Isolando o problema apresentado, teremos que realizar a aplicação das estratégias de controle para ambos os tanques separadamente, o que indica que serão necessárias duas modelagens: uma para cada sistema físico.

Em relação ao tanque 1, é necessário relacionar a vazão de entrada do recipiente - advinda da tensão gerada pela bomba através da tubulação de entrada - e a vazão de saída, que por sua vez, depende da altura do nível da água e do diâmetro de saída do tanque. Para correlacionar as variáveis desejadas, a modelagem dos sistema deve ser especificada.

2.1.1 Modelagem

A variação de volume V do recipiente no tempo depende da diferença entre a vazão de entrada Q_i e a vazão de saída Q_o , assim, pode-se desenvolver:

$$\frac{dV}{dt} = Q_i - Q_o$$

Sabendo que o volume desse tanque é definido como o produto da área da secção transversal A , constante, pela altura da coluna de água h_1 , que, nesse caso, está variando com o tempo, defini-se a variação de altura como sendo:

$$\frac{dh_1}{dt} = \frac{Q_i - Q_o}{A}$$

A vazão de saída é definida como sendo o produto da área da secção transversal do orifício de saída a pela velocidade de saída da água v_o . Observamos que, com a velocidade inicial nula, podemos utilizar a Equação de Torricelli para encontrar essa v_o : $v_o = \sqrt{2gh_1}$, com g sendo a aceleração da gravidade.

De mesma maneira, a bomba introduz uma vazão de entrada ao tanque, porém, não há a regulação direta dessa vazão, mas sim, a tensão da bomba. Assim, é necessário descrever a relação entre a tensão aplicada na bomba e a vazão que ela introduz: $Q_i = V_p \cdot K_p$, onde K_p é a constante de fluxo da bomba.

Alocando os termos adequados, obtém-se que a variação de altura é dada por:

$$\frac{dh_1}{dt} = \frac{K_p V_p - a\sqrt{2gh_1}}{A}$$

Substituindo as áreas das seções da tubulação pela suas relações aos diâmetros fornecidos e chamando a altura h_1 de L_1 , obtém-se a equação diferencial que modela o sistema físico do tanque 1.

$$\frac{dL_1}{dt} = \frac{4K_p V_p - \pi D_{o1}^2 \sqrt{2gL_1}}{\pi D_{t1}^2} \implies \dot{L}_1 = \frac{4K_p V_p - \pi D_{o1}^2 \sqrt{2gL_1}}{\pi D_{t1}^2}$$

2.1.2 Linearização

Após a modelagem do sistema, observa-se o comportamento não linear da planta encontrada. Assim, para encontrar os parâmetros iniciais do controlador, usando uma função de transferência, é necessário linearizar o sistema utilizando um ponto de operação desejado.

Como temos uma função $f(L_1, V_p)$, podemos linearizar através das relações:

$$\dot{L}_1 - \dot{L}_{10} = (L_1 - L_{10}) \frac{\partial f(L_1, V_p)}{\partial L_1} \Big|_{L_1=L_{10}, V_p=V_{p0}} + (V_p - V_{p0}) \frac{\partial f(L_1, V_p)}{\partial V_p} \Big|_{L_1=L_{10}, V_p=V_{p0}}$$

De forma geral, pode-se definir a linearização como sendo:

$$\dot{L}_1 = L_1 \frac{\partial f(L_1, V_p)}{\partial L_1} \Big|_{L_1=L_{10}, V_p=V_{p0}} + V_p \frac{\partial f(L_1, V_p)}{\partial V_p} \Big|_{L_1=L_{10}, V_p=V_{p0}}$$

Por fim, realizando o cálculo das derivadas parciais desejadas, é possível a definição da equação linear para o ponto de operação desejado.

$$\dot{L}_1 = L_1 \frac{-D_{o1}^2 \sqrt{(2gL_1)}}{2D_{t1}^2 L_1} \Big|_{L_1=L_{10}, V_p=V_{p0}} + V_p \frac{4K_p}{\pi D_{t1}^2} \Big|_{L_1=L_{10}, V_p=V_{p0}}$$

Para o ponto de operação, foram utilizadas as informações de referência dadas no problema. Dessa forma, considerando $\dot{L}_1 = 0$ e $L_{10} = 15 \text{ cm}$, conseguimos obter o ponto V_p :

$$\dot{L}_1 = \frac{4K_p V_p - \pi D_{o1}^2 \sqrt{2gL_1}}{\pi D_{t1}^2} \implies 0 = 4K_p V_p - \pi D_{o1}^2 \sqrt{2gL_1}$$

$$\frac{\pi D_{o1}^2 \sqrt{2gL_1}}{4K_p} = V_p \implies V_p = \frac{3.1415 \cdot 0,3175^2 \sqrt{2 \cdot 981 \cdot 15}}{4 \cdot 3,3} \implies V_p = 4,116 \text{ V}$$

Por fim, aplicando o ponto de operação nas derivadas da equação linearizada, pode-se encontrar a equação numérica.

$$\dot{L}_1 = L_1 \cdot (-0,02917) + V_p \cdot (0,212657)$$

Aplicando a transformada de Laplace em ambos os lados:

$$L_1(s) \cdot s = -0,02917L_1(s) + 0,212657V_p(s) \implies G_{p1}(s) = \frac{L_1(s)}{V_p(s)}$$

$$G_{p1}(s) = \frac{0,212657}{s + 0,02917} \quad (1)$$

2.1.3 Parâmetros do sistema de controle

Obtendo a função de transferência do sistema, podemos encontrar os parâmetros que desejamos corrigir com o controlador. Como queremos erro de regime nulo, precisamos descobrir, o valor do erro para a planta e como o sistema de controle do tanque 1 necessita de estabilização em 15 m, assim pode-se utilizar uma entrada degrau com valor 15 na função de transferência 1, para a obtenção do parâmetro.

$$e_r(\infty) = \lim_{s \rightarrow 0} s \frac{15}{s} \frac{1}{1 + G_p(s)} \implies e_r(\infty) = \frac{15}{1 + 7,29} \implies e_r(\infty) = 1,809$$

Além disso, como se trata de um sistema de 1ª ordem, pode-se reajustar a função de transferência para encontrar os parâmetros de valor final e tempo de estabilização.

$$G_{p1}(s) = \frac{0,212657}{s + 0,02917} \cdot \frac{\frac{1}{0,02917}}{\frac{1}{0,02917}} \implies G_{p1} = \frac{7.29026}{34,2817s + 1}$$

Em que $\tau = 34,2817$ é o tempo de estabilização.

Em relação ao *overshoot*, como se trata de um sistema de primeira ordem, inicialmente, não é necessária uma preocupação acerca desse quesito.

Como temos um sistema que apresenta erro de regime e não atende o tempo de estabilização, será necessária a implementação de um controlador proporcional integrativo para esse tanque. Visto que, a parte integrativa tem por objetivo zerar o erro de regime, enquanto a parte proporcional atua no transitório podendo resolver o tempo de estabilização.

2.1.4 Controlador PI

Dessa maneira, podemos projetar um controlador PI que segue a seguinte característica:

$$G_c(s) = \frac{K_p(s + \frac{1}{\tau_i})}{s}$$

O termo do controlador que adiciona um zero à planta pode ser desenvolvido de forma a cancelar o zero da planta original. Dessa maneira, torna-se possível encontrar a variável

τ_i .

A planta G_{p1} , como já mostrado, é dada por:

$$G_{p1}(s) = \frac{0,212657}{s + 0,02917}$$

Logo, para cancelar o polo é necessário que:

$$\frac{1}{\tau_i} = 0,02917 \implies \tau_i = 34,2818$$

Para dimensionar o ganho proporcional K_p , utilizamos a condição de magnitude. Assim, é necessário encontrar a parte real desejada para o projeto do controlador no lugar das raízes.

$$T_{s2\%} = \frac{4}{Re\{P\}} = 5 \implies Re\{P\} = \frac{4}{5}$$

O sistema geral, pode ser descrito como:

$$\frac{K_p(s + \frac{1}{34,2818})}{s} \cdot \frac{0,212657}{s + 0,02917} \implies \frac{K_p \cdot A_z}{A_{p0}} \cdot \frac{0,212657}{A_{p1}} = 1$$

$$K_p = \frac{A_{p0}A_{p1}}{0,212657 \cdot A_z}$$

Como $A_{p1} = A_z$, simplifica-se a expressão, obtendo:

$$K_p = \frac{0,8}{0,212657} \implies K_p = 3,762$$

Com isso, a definição final o controlador proporcional integrativo é dada por:

$$G_c(s) = \frac{3,762(s + \frac{1}{34,2818})}{s}$$

2.2 Tanque 2

Utilizando a mesma abordagem do tanque 1, a análise física do tanque 2 pode ser realizada de forma independente, tendo como objetivo a implementação de um segundo controlador que mantenha a altura em 15cm, com estabilização em 20 segundos e atingindo uma altura máxima de 16,5cm.

O objetivo inicial de análise também será a relação entre as vazões de entrada e saída do tanque em questão.

2.2.1 Modelagem

De maneira similar ao tanque 1, chegamos à expressão abaixo, já que a vazão de saída do tanque 1 é igual à vazão de entrada do tanque 2.

$$\frac{dL_2}{dt} = \frac{Q_{o1} - a\sqrt{2gL_2}}{A}$$

As áreas da secção transversal também podem ser obtidas de maneira similar, ou seja, temos: $A = \pi \frac{D_{t2}^2}{4}$ e $a = \pi \frac{D_{o2}^2}{4}$, onde D_{t2} e D_{o2} são, respectivamente, o diâmetro interno e o diâmetro do orifício de saída do tanque 2. Com isso:

$$\frac{dL_2}{dt} = \frac{Q_{o1} - \pi \frac{D_{o2}^2}{4} \sqrt{2gL_2}}{\pi \frac{D_{t2}^2}{4}} \implies \frac{dL_2}{dt} = \frac{4Q_{o1} - \pi D_{o2}^2 \sqrt{2gL_2}}{\pi D_{t2}^2}$$

Para finalizar a modelagem, podemos escrever a vazão de saída do tanque 1 (Q_{o1}) em função da altura no tanque 1, como fizemos na modelagem do próprio tanque 1:

$$Q_{o1} = \pi \frac{D_{o1}^2}{4} \sqrt{2gL_1}$$

Chegamos a seguinte expressão final:

$$\frac{dL_2}{dt} = \frac{\pi D_{o1}^2 \sqrt{2gL_1} - \pi D_{o2}^2 \sqrt{2gL_2}}{\pi D_{t2}^2} \implies \dot{L}_2 = \frac{D_{o1}^2 \sqrt{2gL_1} - D_{o2}^2 \sqrt{2gL_2}}{D_{t2}^2}$$

Com essa equação, depois da linearização, podemos obter uma função de transferência que relaciona o nível de altura do tanque 2 com o nível de altura do tanque 1.

E, com essa função de transferência em mãos, podemos relacioná-la em série com a função de transferência obtida do tanque 1. A função de transferência resultante dessa associação em série irá expressar diretamente uma relação entre a tensão aplicada na bomba e o nível de altura da água no tanque 2.

2.2.2 Linearização

Novamente, tendo a equação diferencial no tempo, podemos linearizá-la em um ponto de operação definido para obtermos a função de transferência do tanque 2.

Como temos uma função $f(L_1, L_2)$, podemos linearizar através das relações:

$$\dot{x} = \dot{L}_2 - \dot{L}_{20}$$

$$x = L_2 - L_{20}$$

$$L = L_1 - L_{10}$$

Em que as variáveis de índice 0, indicam o ponto de operação utilizado na linearização.

Assim, podemos dizer que:

$$\dot{L}_2 - \dot{L}_{20} = (L_2 - L_{20}) \frac{\partial f(L_1, L_2)}{\partial L_2} \Big|_{L_2=L_{20}, L_1=L_{10}} + (L_1 - L_{10}) \frac{\partial f(L_1, L_2)}{\partial L_1} \Big|_{L_2=L_{20}, L_1=L_{10}}$$

De forma geral, temos:

$$\dot{L}_2 = L_2 \frac{\partial f(L_1, L_2)}{\partial L_2} \Big|_{L_2=L_{20}, L_1=L_{10}} + L_1 \frac{\partial f(L_1, L_2)}{\partial L_1} \Big|_{L_2=L_{20}, L_1=L_{10}}$$

$$\dot{L}_2 = L_2 \frac{-D_{o2}^2 \sqrt{(2gL_2)}}{2D_{t2}^2 L_2} \Big|_{L_2=L_{20}, L_1=L_{10}} + L_1 \frac{D_{o1}^2 \sqrt{(2gL_1)}}{2D_{t2}^2 L_1} \Big|_{L_2=L_{20}, L_1=L_{10}}$$

Assim, através do valor de $L_2 = 15 \text{ cm}$ podemos encontrar L_1 , usando:

$$L_1 = \left(\frac{D_{o2}^2 \sqrt{2g \cdot 15}}{D_{o1}^2 \sqrt{2g}} \right)^2 \implies L_1 = 8,7142 \text{ cm}$$

Por fim, aplicando as alturas desejadas de 15cm e os diâmetros $D_{o2} = 0,47625$, $D_{t1} = D_{t2} = 4,445$ e $D_{o1} = 0,3175$, temos:

$$\dot{L}_2(t) = -0,0656448L_2(t) + 0,0382780L_1(t)$$

Aplicando Laplace na equação:

$$(s + 0,0656448)L_2(s) = 0,0382780L_1(s) \implies G_{p2} = \frac{0,0382780}{s + 0,0656448}$$

Como dito inicialmente, para que a função de transferência se mantenha em relação à tensão da bomba tendo um sistema de FTs em série, iremos multiplicar G_{p1} por G_{p2} encontrando a função de transferência do segundo tanque.

$$G_{p1} \cdot G_{p2} = \frac{0,212657}{s + 0,02917} \cdot \frac{0,0382780}{s + 0,0656448} \implies G_{pt} = \frac{0,00814}{s^2 + 0,0948148s + 0,0019148}$$

$$G_{pt} = \frac{0,00814}{(s + 0,0656464)(s + 0,0291684)}$$

2.2.3 Parâmetros do sistema de controle

Tendo a função de transferência do sistema, podemos verificar se a planta originalmente atende os requisitos desejados para o projeto. Em caso contrário, será necessário pensar acerca do tipo de controlador que será implementado para resolver os requisitos não atendidos inicialmente.

De mesma maneira do caso anterior, podemos verificar o erro de regime da planta para

uma entrada degrau de 15V.

$$e_r(\infty) = \lim_{s \rightarrow 0} s \frac{15}{s} \frac{1}{1 + G_{pt}(s)} \implies e_r(\infty) = \frac{15}{1 + 4,2511} \implies e_r(\infty) = 2,85645$$

Pela função de transferência, podemos ainda identificar os parâmetros ω_n e ξ :

$$\omega_n = \sqrt{0,0019148} \implies \omega_n = 0,0437584$$

$$2\xi\omega_n = 0,0948148 \implies \xi = 1,08339$$

Dessa maneira, o tempo de estabilização da planta é dado por:

$$T_{s2\%} = \frac{4}{\xi\omega_n} \implies T_{s2\%} = 84,375 \text{ s}$$

Com os parâmetros apresentados, pode-se observar que a planta original não atende nenhum dos requisitos de projeto. O percentual de *overshoot* não foi calculado visto que o $\xi > 1$ pode gerar problemas nesse cálculo.

Com essa análise, fica claro que, para esse problema, necessitamos de um controlador proporcional integrativo derivativo (PID), para corrigir tanto os problemas do transitório quanto os de regime permanente.

2.2.4 Controlador PID

O controlador PID é definido da seguinte forma:

$$G_c = \frac{K_c \tau_d \left(s + \frac{1}{2\tau_d} \right)^2}{s}$$

Sendo definido também que $\tau_i = 4\tau_d$.

Para atender os requisitos de projeto, precisamos dimensionar o ponto em que desejamos aplicar os parâmetros do controlador. Para isso, é necessário encontrar o valor de ω_n e ξ que atendem o desejado.

Para um *overshoot* máximo de 10%, temos a seguinte expressão:

$$0,1 = e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}} \implies \xi = 0,591155$$

Assim, podemos definir a frequência natural:

$$T_{s2\%} = \frac{4}{\xi\omega_n} \implies \frac{4}{0,591155\omega_n} = 20 \implies \omega_n = 0,338321$$

Como o produto $\xi\omega_n = 0,2$, o ponto que queremos será dado da seguinte maneira.

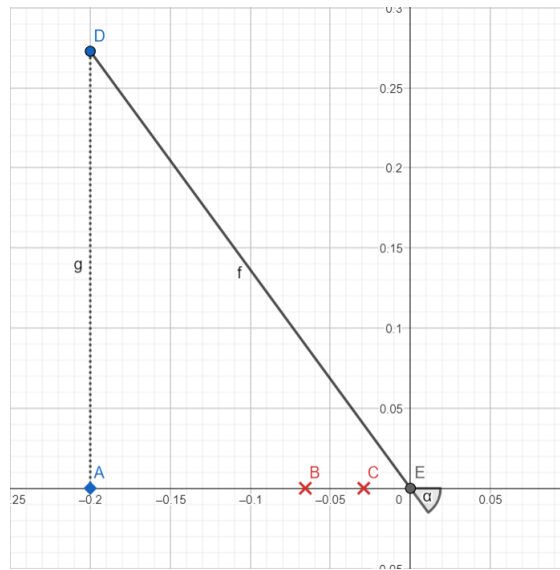


Figura 2: Ponto D desejado.

Podemos encontrar o ângulo α mostrado na figura através da relação do arco cosseno.

$$\alpha = \cos^{-1} \left(\frac{\xi\omega_n}{\omega_n} \right) \Rightarrow \alpha = \cos^{-1}(\xi) \Rightarrow \alpha = 53,76^\circ$$

Tendo esses parâmetros, ao aplicar o controlador PID que adicionará um polo na origem e dois zeros dimensionáveis em $\frac{1}{2\tau_d}$ a configuração no plano s será dada conforme a figura 3 abaixo.

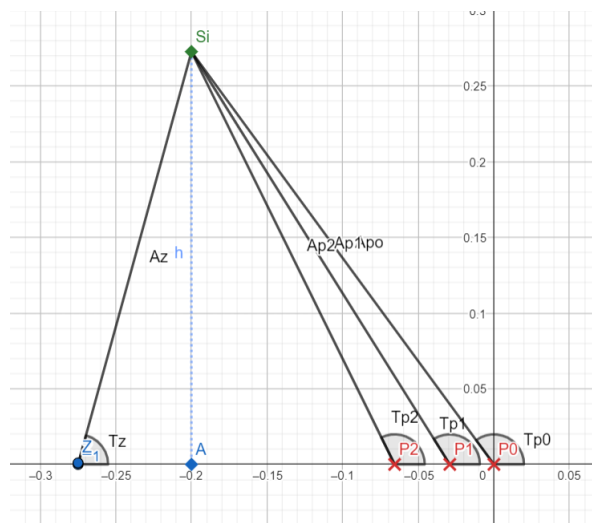


Figura 3: Configuração dos polos e zeros.

Com essa configuração do plano s definida anteriormente, obteremos os ângulos T_{p0} , T_{p1} , T_{p2} e T_z , através das devidas relações trigonométricas, assim como foi feito para a obtenção de α .

Primeiramente, T_{p0} pode ser obtido considerando que este ângulo é o complementar de α para 180° , assim temos:

$$T_{p0} = 180^\circ - \alpha \implies T_{p0} = 180^\circ - 53,76^\circ \implies T_{p0} = 126,24^\circ$$

Por outra análise, obteremos T_{p1} através do ângulo entre a reta A_{p1} e o eixo X do gráfico, que chamaremos de θ_{p1} , sabendo que T_{p1} será o complementar deste ângulo para 180° , assim, temos:

$$\theta_{p1} = \arctan\left(\frac{h}{0,2 - 0,0291684}\right) \implies \theta_{p1} = \arctan\left(\frac{0,2728}{0,2 - 0,0291684}\right)$$

$$\theta_{p1} = 57,94^\circ$$

Logo, sabemos que:

$$T_{p1} = 180^\circ - 57,94^\circ \implies T_{p1} = 122,06^\circ$$

De forma análoga, obteremos T_{p2} através do ângulo entre a reta A_{p2} e o eixo X do gráfico, que chamaremos de θ_{p2} , sabendo que T_{p2} também será o complementar deste ângulo para 180° , assim, temos:

$$\theta_{p2} = \arctan\left(\frac{h}{0,2 - 0,0656464}\right) \implies \theta_{p2} = \arctan\left(\frac{0,2728}{0,2 - 0,0656464}\right)$$

$$\theta_{p2} = 63,78^\circ$$

Portanto, temos:

$$T_{p2} = 180^\circ - 63,78^\circ \implies T_{p2} = 116,22^\circ$$

Seguindo a análise, calcularemos T_z , sabendo que:

$$2T_z - T_{p2} - T_{p1} - T_{p0} = 180^\circ$$

$$2T_z = 180^\circ + 116,22^\circ + 122,06^\circ + 126,24^\circ \implies 2T_z = 544,52^\circ$$

$$2T_z = 544,52^\circ - 360^\circ \implies 2T_z = 184,52^\circ \implies T_z = 92,26^\circ$$

A partir disso, conseguiremos encontrar a distância de $-0,2$ até $-\frac{1}{2\tau_d}$, que chamaremos de D e que nos dará o valor de τ_d . Logo, temos que:

$$\tan(T_z) = \left(\frac{h}{D}\right) \implies \tan(92,26^\circ) = \left(\frac{0,2728}{D}\right)$$

$$D = \left(\frac{0,2728}{\tan(92,26^\circ)}\right) \implies D = 0,0107$$

Dessa forma, sabendo que o ângulo T_z encontrado é superior a 90° , concluímos que $-\frac{1}{2\tau_d}$ estará a direita de $-0,2$ no eixo X, assim, obteremos o valor de τ_d da seguinte maneira:

$$\frac{1}{2\tau_d} = 0,2 - 0,0107 \implies \frac{1}{2\tau_d} = 0,1893 \implies \tau_d = \frac{1}{2 \times 0,1893}$$

$$\tau_d = 2,6413$$

Como $\tau_i = 4\tau_d$, temos:

$$\tau_i = 4 \times 2,6413 \implies \tau_i = 10,5652$$

Por fim, calcularemos, através da relação do triângulo retângulo e seus lados, os valores de A_{p0} , A_{p1} , A_{p2} e A_z .

Para A_z , sabe-se que:

$$A_z^2 = h^2 + D^2 \implies A_z = \sqrt{0,2728^2 + 0,0107^2} \implies A_z = 0,2730$$

Para A_{p0} , sabe-se que:

$$A_{p0} = \omega_n \implies A_{p0} = 0,338321$$

Para A_{p1} , sabe-se que:

$$A_{p1}^2 = h^2 + (0,2 - 0,0291684)^2 \implies A_{p1} = \sqrt{0,2728^2 + 0,1708316^2}$$

$$A_{p1} = 0,3219$$

Para A_{p2} , sabe-se que:

$$A_{p2}^2 = h^2 + (0,2 - 0,0656464)^2 \implies A_{p2} = \sqrt{0,2728^2 + 0,1343536^2}$$

$$A_{p2} = 0,3041$$

Feito isso, finalmente obteremos o valor de K_c , relacionando as funções de transferência da planta (G_{pt}) e do controlador (G_c), da seguinte maneira:

$$G_c \times G_{pt} = 1 \implies \frac{K_c \tau_d \left(s + \frac{1}{2\tau_d}\right)^2}{s} \times \frac{0,00814}{(s + 0,0656464)(s + 0,0291684)} = 1$$

Sabendo que:

$$A_z = \left(s + \frac{1}{2\tau_d}\right) \implies A_{p0} = s \implies A_{p1} = (s + 0,0291684) \implies A_{p2} = (s + 0,0656464)$$

Logo:

$$\frac{K_c \tau_d A_z^2}{A_{p0}} \times \frac{0,00814}{A_{p2} \cdot A_{p1}} = 1 \implies K_c = \frac{A_{p0} \cdot A_{p1} \cdot A_{p2}}{0,00814 \cdot A_z^2 \cdot \tau_d}$$

$$K_c = \frac{0,338321 \cdot 0,3219 \cdot 0,3041}{0,00814 \cdot 0,2730^2 \cdot 2,6413} \implies K_c = 20,6680$$

Por fim, finalmente obteremos G_c :

$$G_c = \frac{K_c \tau_d \left(s + \frac{1}{2\tau_d}\right)^2}{s} \implies G_c = \frac{20,6680 \cdot 2,6413 \left(s + \frac{1}{2 \cdot 2,6413}\right)^2}{s}$$

$$G_c = \frac{54,5903 \cdot (s + 0,3786s + 0,0358)}{s}$$

$$G_c = \frac{54,5903s^2 + 20,6679s + 1,9543}{s}$$

2.3 Implementações

Tendo o projeto dos controladores definidos matematicamente, é possível realizar a implementação computacional do projeto de forma a visualizar os resultados obtidos, tanto da planta original, como da planta ao ser submetida ao sistema de controle.

Com base nisso, foi realizada a implementação computacional utilizando a linguagem de programação Python, tendo em vista sua boa flexibilidade de aplicação em diferentes tipos de problema. Através dela, foi utilizada uma biblioteca chamada control, que permite algumas ferramentas úteis de definição da função de transferência e do método geométrico do lugar das raízes.

2.3.1 Controlador PI - Tanque 1

Trazendo de volta a planta original obtida pela modelagem do tanque 1, temos:

$$G_{p1}(s) = \frac{0,212657}{s + 0,02917}$$

Aplicando o lugar geométrico das raízes LGR, temos apenas um polo, que nos apresentará a seguinte estrutura.

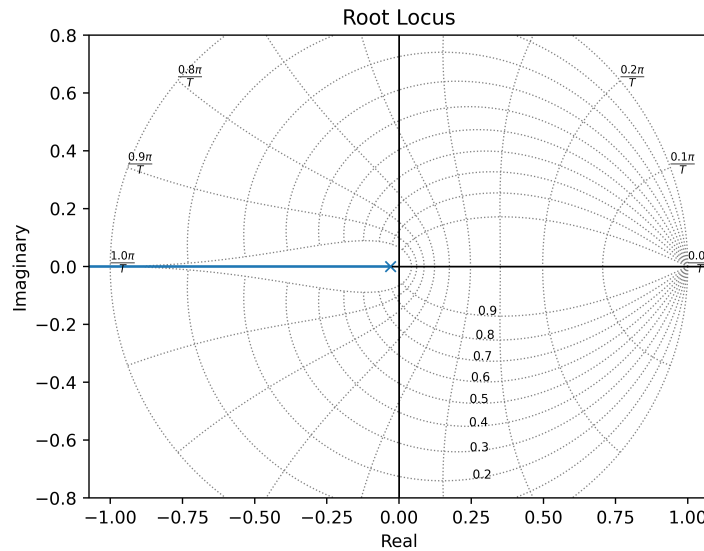


Figura 4: Lugar das raízes - Planta original.

Ao se adicionar o controlador PI, em teoria é adicionado um polo na origem e um zero dimensionável. No entanto, como o zero desenvolvido matematicamente nos tópicos acima foi utilizado para cancelar o polo, a nova planta terá a seguinte característica.

$$G_{PI} = G_c \cdot G_{p1} \implies G_{PI} = \frac{3,762(s + \frac{1}{34,2818})}{s} \cdot \frac{0,212657}{s + 0,02917}$$

$$G_{PI} = \frac{0,8}{s}$$

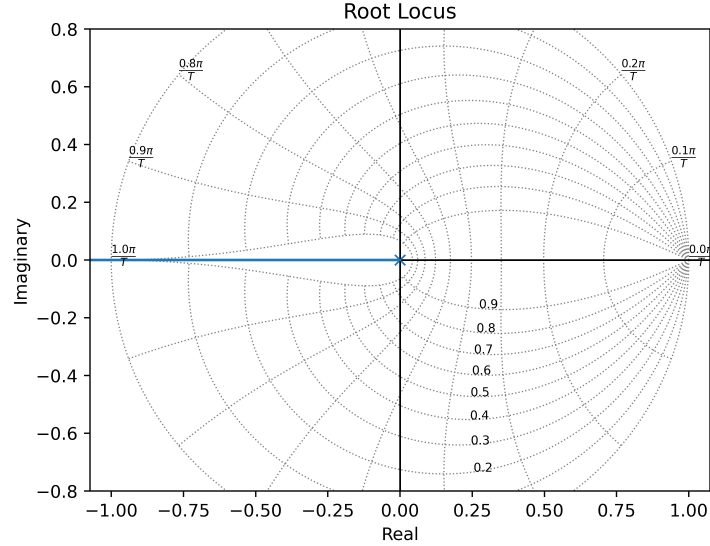


Figura 5: Lugar das raízes - Planta com controle.

Com isso, observando a resposta do sistema ao ser submetido a uma referência degrau em altura de 15 cm, com os parâmetros calculados $K_p = 3,762$ e $\tau_i = 34,2818$, vemos que ele já atende os requisitos de projeto desejados.

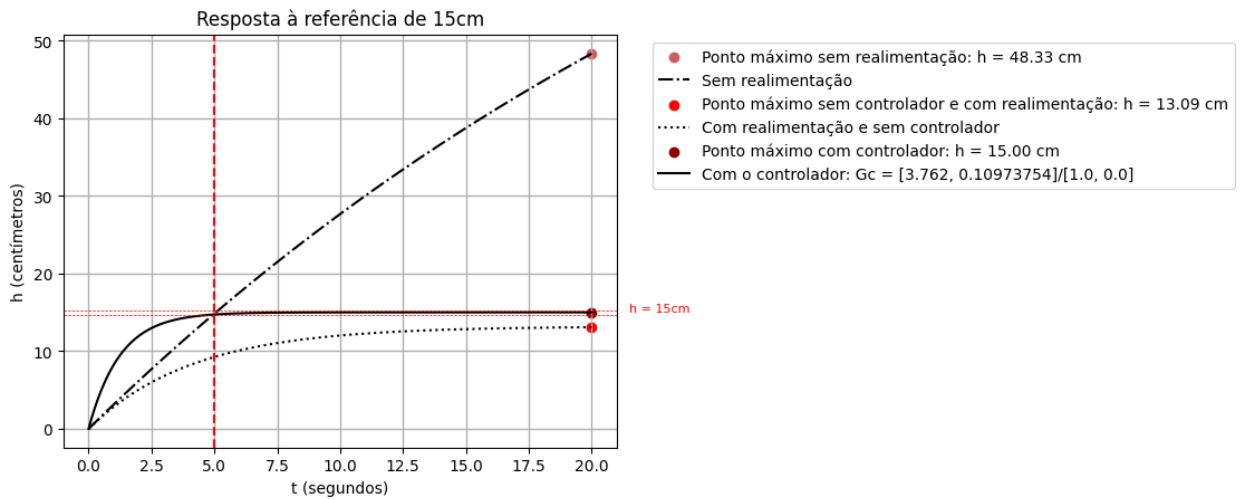


Figura 6: Resposta do tanque 1.

Como mostrado na legenda da figura, a curva em sólida em preto representa a resposta do sistema com o controlador. Essa curva atinge o percentual de 2% da altura antes dos 5 segundos desejados, com um erro de regime nulo e, novamente, como se trata de um sistema de primeira ordem, sem *overshoot*.

As demais curvas representam o sistema sem controle: de forma original (linha traço-ponto), que não atinge estabilização antes dos 34 segundos, como calculado matematicamente; e o sinal com realimentação unitária (pontilhado) que já apresenta uma melhora em relação ao sistema original, mas com um erro de regime também já previsto.

Assim, tendo realizado o controle da planta de forma linearizada, foi possível iniciar o estudo para o controle da planta real, não linear.

Implementando a equação diferencial ordinária que rege o comportamento do tanque 1 em Python, foi possível, da mesma forma, verificar a resposta do sistema ao ser implementado o controlador PI. No entanto, ao parametrizar o controlador da mesma maneira que foi calculado matematicamente, o sistema não linear não apresentou resultados que atendessem todos os requisitos solicitados. Assim, foi necessário o ajuste fino dos termos, para conseguir tal objetivo.

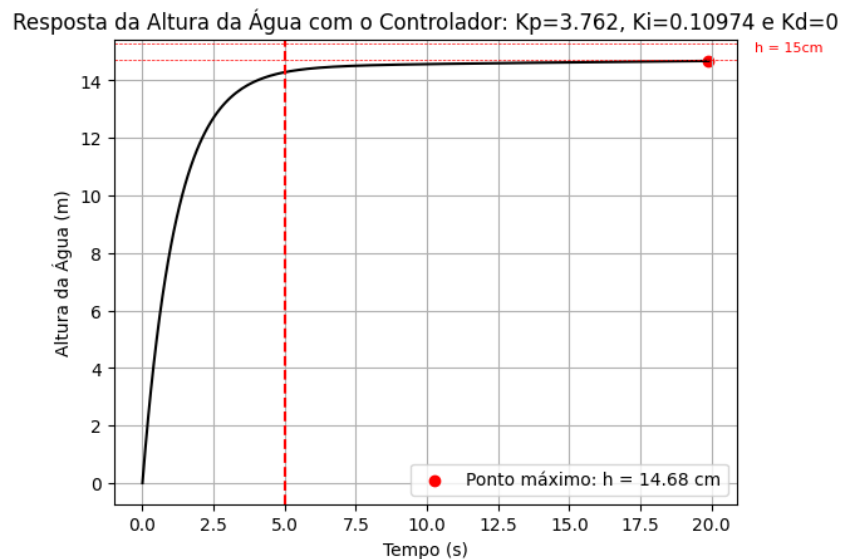


Figura 7: Resposta não linear calculada.

Observando que os parâmetros não foram atendidos, foi-se realizado o ajuste, como já citado, obtendo o resultado apresentado abaixo. Para esse caso, o K_p foi dimensionado como 4,4 e o K_i como sendo 0,25. Valores um pouco acima dos calculados matematicamente.

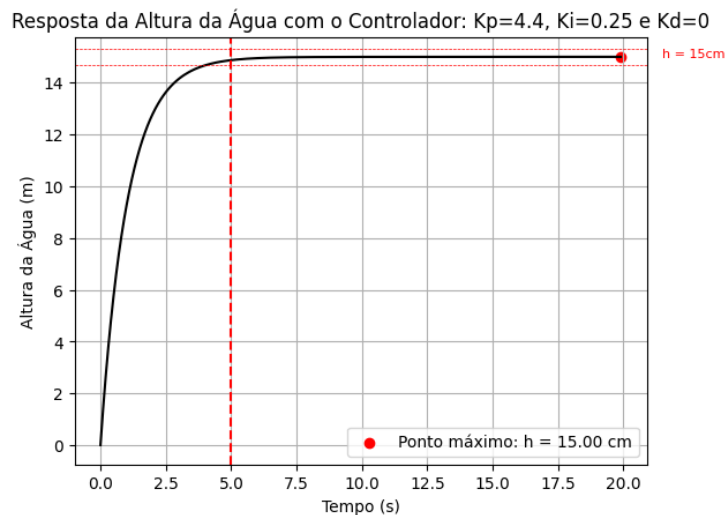


Figura 8: Resposta não linear com ajustes.

Para o caso acima, também foi verificada a tensão da bomba, já que por limitações físicas, ela não pode ultrapassar os 12V. O resultado está apresentado a seguir.

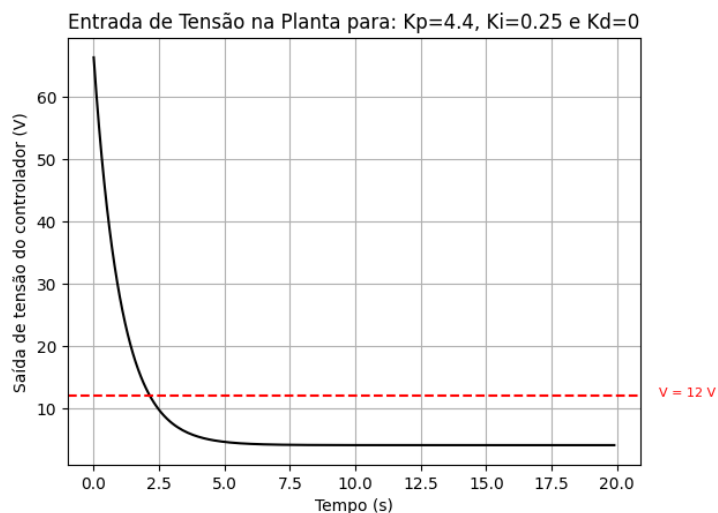


Figura 9: Tensão da bomba.

Tendo em vista o pico de tensão gerado durante o transitório, foi implementado um saturador que tem o objetivo de limitar a tensão da bomba em 12V. Ele atua basicamente da seguinte maneira: Para $V > 12 \implies V = 12$, para $V \leq 12 \implies V = V$.

Dessa forma, implementando essa condição também foi necessário alterar os ganhos do controlador, visto que alterando a variação da bomba, também alteramos a resposta do sistema.

Assim, com o saturador e alterando os parâmetros, o sistema apresentou o seguinte comportamento.

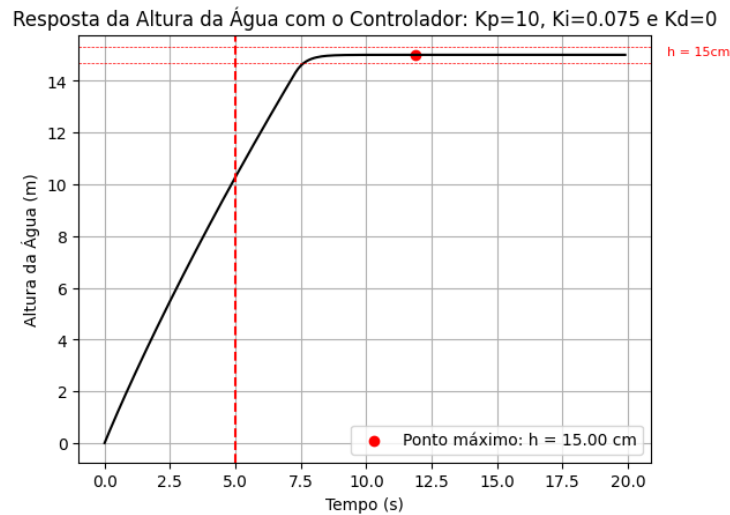


Figura 10: Resposta com saturador.

Com as devidas adequações dos ganhos através do método heurístico, chegamos a um valor de $K_p = 10$ e $K_i = 0,075$. Veja que, com esses valores, embora tenhamos zerado o erro de regime, não conseguimos estabilizar a planta antes de 7,5 segundos. Mesmo alterando os valores e realizando testes, não foi possível encontrar valores que permitissem resposta mais satisfatória.

Então, para finalizar a análise do tanque 1, basta apenas verificar a tensão da bomba após a implementação do saturador. O resultado está apresentado abaixo.

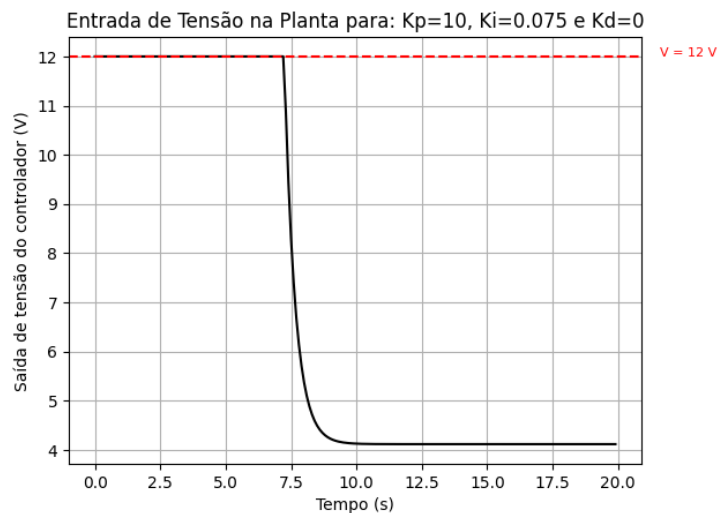


Figura 11: Tensão da bomba com saturador.

Observamos que agora, não há mais ultrapassagem da tensão da bomba de sua tensão de operação, atingindo o objetivo desejado inicialmente.

2.3.2 Controlador PID - Tanque 2

Para o tanque 2, temos uma metodologia semelhante. Trazendo de volta a função de transferência obtida, temos:

$$G_{pt} = \frac{0,00814}{(s + 0,0656464)(s + 0,0291684)}$$

Através da função de transferência acima, aplicando o LGR, temos o seguinte comportamento.

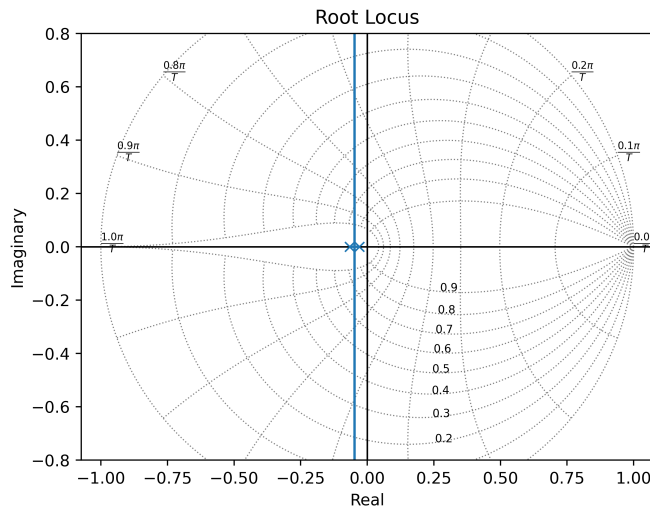


Figura 12: LGR Planta 2.

Observa-se que, como previsto, os dois polos se juntam e partem em assíntotas de 90° e 270° . No entanto, essa configuração é alterada ao se adicionar o controlador PID, já que o mesmo acrescenta 2 zeros e um polo na origem. Assim, com esse acréscimo, o LGR será dado como mostrado na figura 13.

Nesse caso, percebemos um comportamento bem mais complexo tendo em vista o aumento no número de polos e zeros. No entanto, o comportamento do lugar geométrico das raízes ainda condiz com o que deveria ser esperado, devido a configuração encontrada.

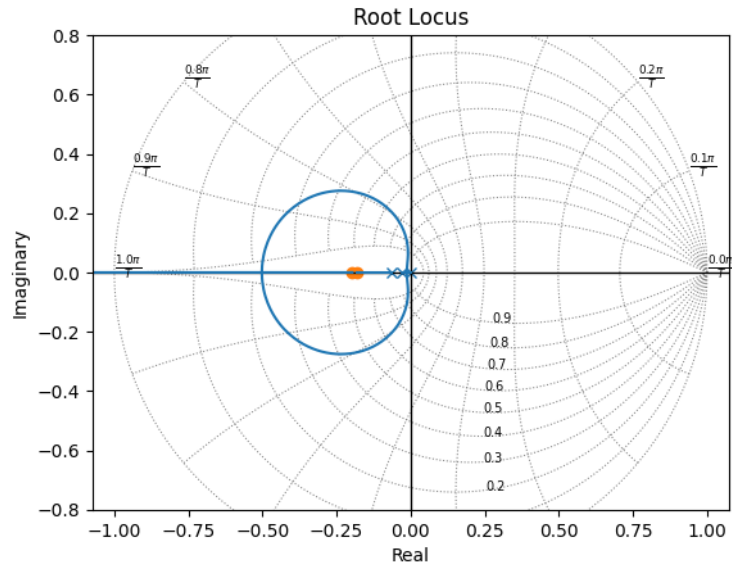


Figura 13: LGR Planta + Controlador.

Partindo então para a resposta da planta linearizada, podemos observar na figura abaixo a forma que se deu a resposta do sistema ao ser submetido a uma entrada degrau de 15cm, usando os parâmetros calculados para K_c , τ_d e τ_i .

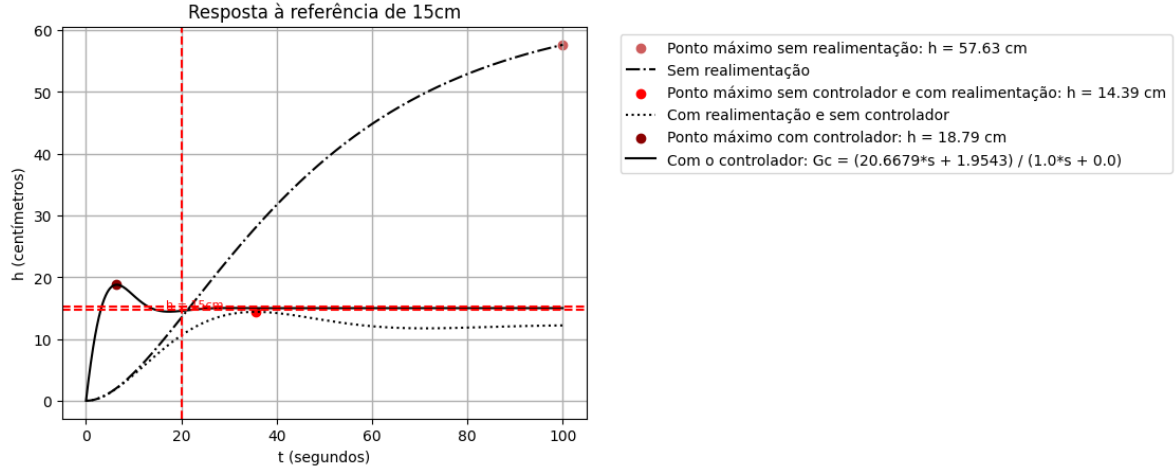


Figura 14: Resposta do sistema, parâmetros calculados.

De mesma maneira da planta 1, foram plotados as 3 repostas do sistema: planta original (curva traço-ponto), planta com realimentação unitária (curva pontilhada) e por fim, a curva sólida indica a resposta com controlador.

Observe que o parâmetro de tempo de estabilização e erro de regime foram atendidos, no entanto, o parâmetro de *overshoot* não, já que seu valor máximo está em 18,79 cm, enquanto deveria chegar no máximo em 16,50 cm. Dessa maneira, foram realizados alguns ajustes nos parâmetros do controlador de forma que, para atender os requisitos, o valor

de K_c utilizado foi de 28 e de τ_d como sendo 8,1. A resposta corrigida pode ser visualizada abaixo

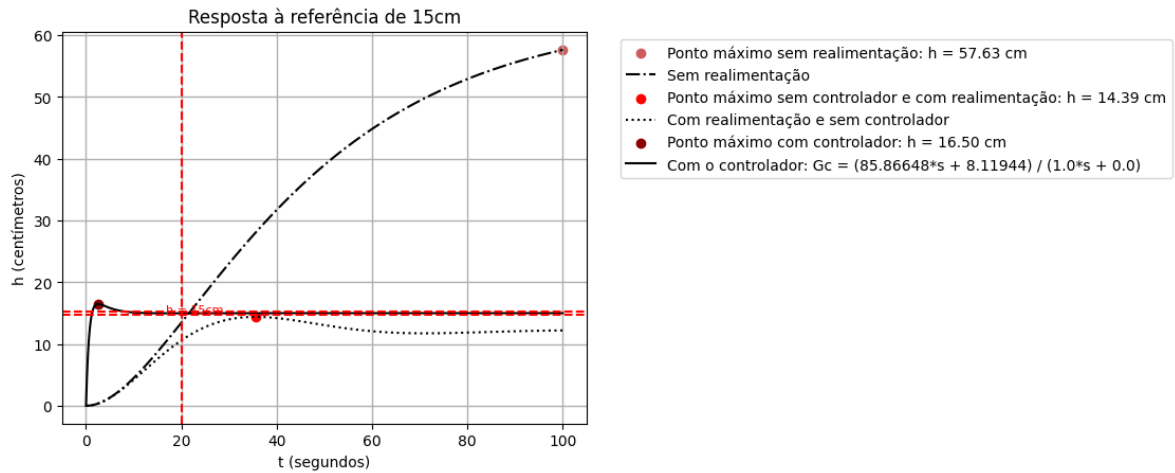


Figura 15: Tanque 2 com ajustes.

Novamente, tendo controlado a planta linearizada, foi realizada a tentativa de controle da planta não linear. E da mesma maneira, a equação diferencial ordinária foi implementada no código de forma que fosse possível a visualização de sua resposta ao ser implementado o PID.

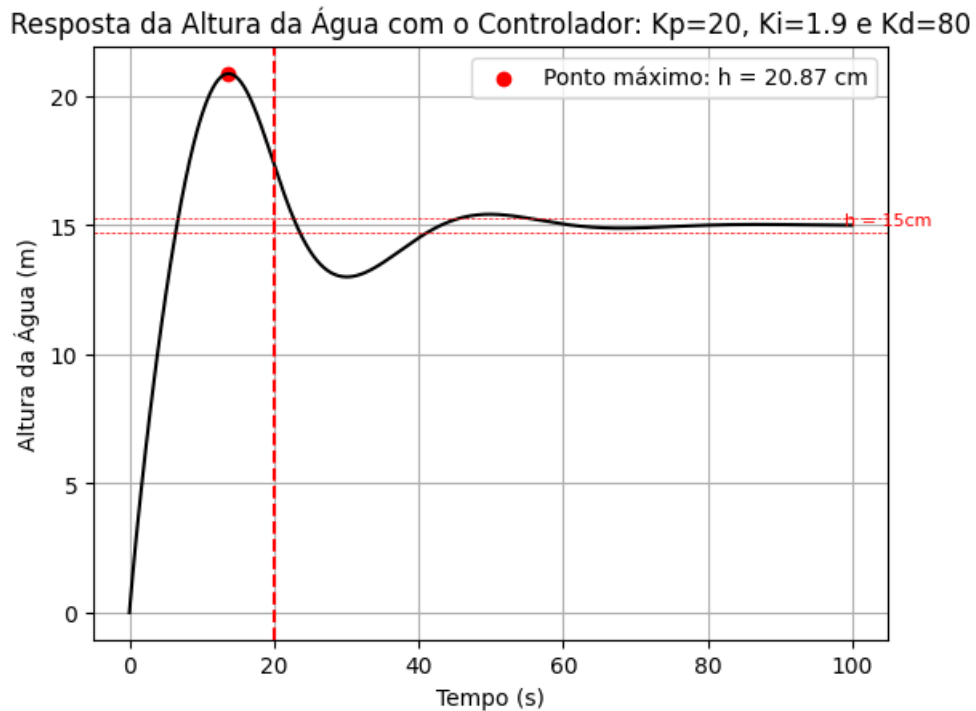


Figura 16: Resposta não linear: tanque 2.

É possível fazer alguns comentários acerca da resposta: 1° O sistema realmente estabilizou, ou seja, a estratégia de controle utilizada surtiu efeito. No entanto, percebe-se que

os requisitos solicitados pelo problema não foram atendidos, temos um *overshoot* alto, assim como um tempo de estabilização. 2° Os parâmetros utilizados para os ganhos do controlador estão diferentes dos calculados e dos ajustados. Isso se deu pelo fato do sistema não responder satisfatoriamente para ambos os casos. Assim foram realizados teste para encontrar valores que pudessem melhorar o desempenho. E dessa maneira foram usados os valores de 20, 1.9 e 80, para K_p , K_i e K_d , respectivamente.

Outro problema encontrado nesse caso, é a tensão da bomba. Usando esses parâmetros mostrados na figura anterior, temos no transitório uma tensão da bomba atingindo o valor de 12kV, algo desproporcional para o problema. A ideia é conseguir limitar isso justamente com a implementação de um saturador. A resposta da tensão pode ser visualizada abaixo.



Figura 17: Tensão da bomba sem saturador.

Tendo em vista isso, foi implementado o saturador, obtendo os resultados mostrados abaixo.

Resposta da Altura da Água com Saturador e Controlador: $K_p=5$, $K_i=1.95623367$ e $K_d=5$

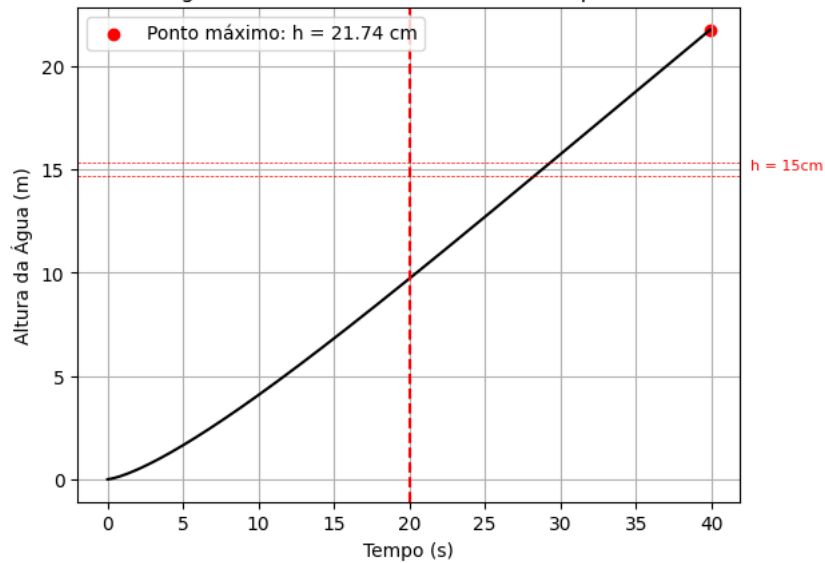


Figura 18: Resposta sistema com saturador.

Aqui, tivemos uma resposta bem pior ao caso anterior. Como a tensão da bomba não dá o pico inicial, a resposta do sistema se dá de forma bem mais lenta, sendo que em 40s de simulação, o sistema ainda nem iniciou o período de oscilação. Nesse caso, independente dos valores de controlador utilizados, não foi possível controlar o sistema do tanque 2.

Por fim, foi verificada a tensão da bomba com saturador apenas para mostrar sua estabilidade em 12V.

Saída de tensão do controlador saturado para: $K_p=5$, $K_i=1.95623367$ e $K_d=5$

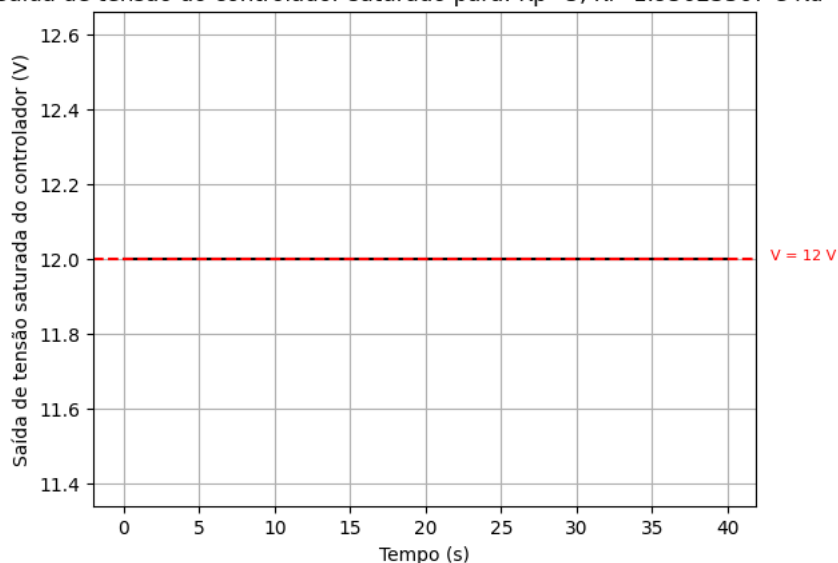


Figura 19: Tensão da bomba com saturador.

Assim, de forma resumida, para o tanque 2 foi atingido o objetivo de controle da planta

linear e não linear (com suas ressalvas), no entanto, não foi possível realizar o controle eficiente utilizando o saturador implementado.

2.4 Códigos

Para a realização da simulação do controle do sistema de controle da altura de água dos tanques, foi utilizada a linguagem de programação Python, que possui diversas bibliotecas que suportam análises matemáticas como a Numpy, plotagem de gráficos como a Matplotlib e a modelagem de sistemas de controle como a Control (Listing 1).

```
1 !pip install control
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import control as ctl
```

Listing 1: Dependências Importadas

Primeiramente, para análise da resposta do tanque à entrada de tensão sem realimentação negativa, depois somente com a realimentação, e, por fim, com o controlador, foi implementada a função `plot_response` (Vista abaixo no Listing 2, os argumentos tiveram seu estado de default com base no tanque 1, que depois foram modificados para o tanque 2). Nessa função, são passados como argumentos as funções de transferência da planta, do controlador, o tempo de simulação, à referência de altura de água que se deseja obter e uma referência de `vertical_line` que serve para análise do tempo de estabilização. A função `plot_response` se utiliza da função `control` para contruir os modelos de função de transferência e feedback com os parâmetros dados e fornece a função `forced_response` para o cálculo das respostas aqui descritas. Nessa célula, alguns outros parâmetros para visualização do gráfico foram definidos, no intuito de analisar se a função se enquadra dentro dos parâmetros estabelecidos.

```
1 def plot_response(numGpd=[0.212657], denGpd=[1, 0.02917], numGc=[8],
2     denGc=[1], T_sim=20, REFERENCE=15, vertical_line=5):
3     # Modelo da planta
4     sysGpd = ctl.tf(numGpd, denGpd)
5     print('Gp = ', sysGpd)
6
7     # Modelo da realimentacao unitaria
8     sysR = ctl.tf([1], [1])
9
10    # Modelo do controlador
11    sysGc = ctl.tf(numGc, denGc)
12
13    # Feedback sem o controlador
14    sysFBs = ctl.feedback(sysGpd, sysR)
15    print('G_malhaFechada_semGp = ', sysFBs)
```

```

16 # Malha Fechada antes do feedback com o controlador
17 sysMFC = ctl.series(sysGc, sysGpd)
18
19 # Feedback com o controlador
20 sysFBc = ctl.feedback(sysMFC, sysR, sign=-1)
21 print('G_MalhaFechada_comGp = ', sysFBc)
22
23 tx = np.linspace(0, T_sim, 1000) # Pontos da simulacao
24 u = REFERENCE * np.ones_like(tx) # Referencia constante de 15cm
25
26 ### Plotagem da resposta a entrada de tensao sem realimentacao
27 t, h = ctl.forced_response(sysGpd, tx, u)
28
29 ## Destacar o ponto maximo para analise overshoot
30 indice_ponto_maximo = np.argmax(h)
31 x_ponto_maximo = t[indice_ponto_maximo]
32 y_ponto_maximo = h[indice_ponto_maximo]
33
34 # Destaque o ponto maximo com um marcador
35 plt.scatter(x_ponto_maximo, y_ponto_maximo, color='indianred',
36             label=f'Ponto maximo sem realimentacao: h = {
37                 y_ponto_maximo:.2f} cm')
38 plt.plot(t, h, color='black', linestyle='-.', label='Sem realimentacao
39 ')
40
41 ### Plotagem da resposta a entrada de tensao com a realimentacao e sem
42 o controlador
43 t, h = ctl.forced_response(sysFBs, tx, u)
44
45 ## Destacar o ponto maximo para analise overshoot
46 indice_ponto_maximo = np.argmax(h)
47 x_ponto_maximo = t[indice_ponto_maximo]
48 y_ponto_maximo = h[indice_ponto_maximo]
49
50 # Destaque o ponto maximo com um marcador
51 plt.scatter(x_ponto_maximo, y_ponto_maximo, color='red',
52             label=f'Ponto maximo sem controlador e com realimentacao:
53                 h = {y_ponto_maximo:.2f} cm')
54 plt.plot(t, h, color='black', linestyle='dotted', label='Com
55             realimentacao e sem controlador')
56
57 ### Plotagem da resposta a entrada de tensao com o controlador
58 t, h = ctl.forced_response(sysFBc, tx, u)
59 ex = t
60 ey = ((REFERENCE * np.ones_like(h)) - h)
61
62 ## Destacar o ponto maximo para analise overshoot

```

```

58 indice_ponto_maximo = np.argmax(h)
59 x_ponto_maximo = t[indice_ponto_maximo]
60 y_ponto_maximo = h[indice_ponto_maximo]
61
62 # Destaque o ponto maximo com um marcador
63 plt.scatter(x_ponto_maximo, y_ponto_maximo, color='darkred',
64             label=f'Ponto maximo com controlador: h = {y_ponto_maximo
65                  :.2f} cm')
66
67 plt.plot(t, h, color='black', label=f'Com o controlador: Gc = {numGc
68     }/{denGc}')
69
70 ##### Eixos do grafico
71 plt.xlabel('t (segundos)')
72 plt.ylabel('h (centimetros)')
73 plt.title(f'Resposta a referencia de {REFERENCE}cm')
74
75 ##### Linha vertical para analise do tempo de estabilizacao
76 plt.axvline(x=vertical_line, color='red', linestyle='--')
77
78 ##### Definir o valor de h onde se deseja a linha horizontal -
79 Estabilizacao
80 ESTAB_MAX = 15.3
81 ESTAB_MIN = 14.7
82 ESTAB = 15
83
84 plt.axhline(y=14.7, color='red', linestyle='--', lw=0.5)
85 plt.axhline(y=15.3, color='red', linestyle='--', lw=0.5)
86 plt.annotate(f'h = {ESTAB}cm', (0, ESTAB), (T_sim+7, ESTAB),
87             color='red', fontsize=8, ha='center')
88
89 ##### Plotagem do grafico
90 plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
91 plt.grid(linewidth=1)
92 plt.savefig('Resposta_a_referencia.png')
93 plt.show()
94
95 return ctl.forced_response(sysGc, ex, ey)

```

Listing 2: Dependências Importadas

Para plotar os gráficos para análise do controlador, um loop foi feito (No Listing 3 é apresentado a estrutura para o primeiro tanque, que tiveram seus argumentos alterados para o tanque 2). Nesse loop, é solicitado ao usuário os parâmetros do numerador e denominador da função de transferência do controlador, para então fornecer o gráfico. Nessa célula, ainda é plotada um gráfico para visualização do nível de tensão na bomba, imposto pelo controlador.

```

1 while True:

```

```

2 numGc = [float(coeffs) for coeffs in input("Coloque os coeficientes do
    numerador de Gc: ").split()]
3 denGc = [float(coeffs) for coeffs in input("Coloque os coeficientes do
    denominador de Gc: ").split()]
4
5 t, v = plot_response(numGc=numGc, denGc=denGc)
6 mudarGc = input('Deseja mudar Gc? (S/N)')
7 if mudarGc.lower() == 's':
8     continue
9 else:
10    break

```

Listing 3: Dependências Importadas

Além disso, como solicitado, foi feito uma análise não linear do sistema. Para isso, utilizamos a biblioteca Scipy para cálculo da equação diferencial ordinária encontrada na modelagem. A solução dessa EDO foi dada pela função `odeint` do Scipy e, depois da montagem do controlador PID com o cálculo das parcelas derivativa, integrativa e proporcional, a resposta gráfica foi gerada (Listing 4). Além do análise do nível de tensão, de maneira similar àquela vista anteriormente (Listing 5).

```

1 # Parametros do sistema
2 Kp = 3.3
3 Vp=4
4 Do1 = 0.3175
5 g=981
6 Dt1 = 4.445
7
8 # Funcao que descreve a equacao diferencial da planta
9 def plant(y, t, u):
10     h = y[0]
11     dhdt = ((4*Kp*u)-(np.pi * Do1*Do1*np.sqrt(2*g*h)))/(np.pi * Dt1*Dt1)
12     return [dhdt]
13
14 # Parametros do controlador PID
15 # ----- ALTERAR -----
16 kp = 3.762
17 ki = 0.10974
18 kd = 0
19 # -----
20
21 dt = 0.1 # Intervalo de amostragem
22 sim_time = 20 # Tempo total de simulacao
23 setpoint = 15.0 # Altura desejada da agua
24
25 # Inicializacao
26 time = np.arange(0, sim_time, dt)
27 height = np.zeros_like(time)

```

```

28 prev_error = 0
29 integral = 0
30 ddp = []
31
32 # Loop de simulacao
33 for i in range(1, len(time)):
34     # Calcular o erro
35     error = setpoint - height[i - 1]
36
37     # Calcular a integral e o derivativo
38     integral += error * dt
39     derivative = (error - prev_error) / dt
40
41     # Calcular o sinal de controle usando o controlador PID
42     control_signal = kp * error + ki * integral + kd * derivative
43
44     ddp.append(control_signal)
45     # Integrar a EDO usando odeint
46     y0 = [height[i - 1]]
47     y = odeint(plant, y0, [0, dt], args=(control_signal,))
48     height[i] = y[-1]
49
50     # Atualizar o erro anterior
51     prev_error = error
52
53 ddp.append(control_signal)
54
55 # Plotar resultados
56 plt.figure()
57 plt.plot(time, height, color='black')
58 plt.xlabel('Tempo (s)')
59 plt.ylabel('Altura da Agua (m)')
60 plt.title(f'Resposta da Altura da Agua com o Controlador: Kp={kp}, Ki={
    ki} e Kd={kd}')
61
62 ## Destacar o ponto maximo para analise overshoot
63 indice_ponto_maximo = np.argmax(height)
64 x_ponto_maximo = time[indice_ponto_maximo]
65 y_ponto_maximo = height[indice_ponto_maximo]
66
67 # Destaque o ponto maximo com um marcador
68 plt.scatter(x_ponto_maximo, y_ponto_maximo, color='red',
69             label=f'Ponto maximo: h = {y_ponto_maximo:.2f} cm')
70
71 #### Linha vertical para analise do tempo de estabilizacao
72 plt.axvline(x=5, color='red', linestyle='--')
73

```

```

74 ##### Definir o valor de h onde se deseja a linha horizontal -
    Estabilizacao
75 ESTAB_MAX = 15.3
76 ESTAB_MIN = 14.7
77 ESTAB = 15
78 plt.axhline(y=14.7, color='red', linestyle='--', lw=0.5)
79 plt.axhline(y=15.3, color='red', linestyle='--', lw=0.5)
80 plt.annotate(f'h = {ESTAB}cm', (0, ESTAB), (sim_time+2.7, ESTAB),
81             color='red', fontsize=8, ha='center')
82
83 plt.legend()
84 plt.grid(True)
85 plt.savefig(f'Resposta da Altura da Agua com o Controlador')
86 plt.show()

```

Listing 4: Dependências Importadas

```

1  ### Para plotagem da tensao
2  def see_ddp(t, v):
3      plt.plot(t, v, color='black')
4
5      plt.xlabel('t (segundos)')
6      plt.ylabel('Tensao (V)')
7      plt.title('Entrada de Tensao na Planta (Entre Gc e Gp)')
8
9      plt.axhline(y=12, color='red', linestyle='--')
10     plt.annotate(f'V = 12 V', (0, 12), (22.7, 12),
11                 color='red', fontsize=8, ha='center')
12     plt.grid(linewidth=1)
13
14     plt.savefig('entrada_tensao.png')
15     plt.show()

```

Listing 5: Dependências Importadas

A fim de uma manipulação mais realista de um modelo de controlador, foi imposto um saturador à tensão da bomba, que, com base nas informações de projetos dadas, não poderia passar de 12 V. Com isso, a implementação ocorreu da mesma maneira, somente com a inserção de um saturador na saída do controlador. Esse saturador foi implementado através da função clip da biblioteca Numpy (Listing 6), essa função, no nosso caso, irá retornar o valor de tensão dentro da faixa de -12V a +12V, se a tensão imposta pelo controlador for maior que 12V, a função manterá a entrada na planta em 12V e, se a tensão imposta for menor que -12V, a função manterá a entrada em -12V. É possível ver nos gráficos mostrados da entrada de tensão na planta ao longo do tempo que a tensão foi corretamente saturada.

```

1  # Limites de tensao na bomba

```



```
2 limite_inferior=-12
3 limite_superior=12
4
5 # Aplicar os limites ao sinal de controle
6 control_signal = np.clip(control_signal, limite_inferior,
    limite_superior)
```

Listing 6: Dependências Importadas

3 Considerações finais

O presente relatório descreveu a aplicação de técnicas de controle em sistemas de tanques interligados, representando um desafio significativo na área de Engenharia Elétrica. As experiências realizadas no controle dos níveis de água em dois tanques demonstraram a aplicação de conceitos teóricos em um cenário prático e destacaram a importância de abordar tanto sistemas lineares quanto não lineares, além, claro, do controle através do método do lugar das raízes.

No primeiro tanque, a abordagem começou com a modelagem da planta através de uma Equação Diferencial Ordinária (EDO) de primeira ordem, seguida pela linearização em torno do ponto de operação desejado. A escolha de um controlador Proporcional-Integral (PI) com base no método do lugar das raízes provou ser eficaz, levando o nível de água a uma estabilização satisfatória em 15cm em menos de 5 segundos. Isso demonstra a robustez e eficácia do controle PI e o êxito no cálculo dos parâmetros.

Essa eficácia para o controle do nível de água do tanque 1 também foi observada no controle através da planta não linearizada no domínio do tempo e também para esse caso com saturação da tensão da bomba na faixa de -12V a 12V. Para ambos os casos, foi necessário a adequação dos parâmetros do PI calculados no modelo linear, para uma estabilização antes de 5 segundos em 15cm e para evitar overshoot.

Ao lidar com o segundo tanque, a complexidade aumentou devido à natureza de segunda ordem, além da não linear da planta e à necessidade de interligar os tanques. A obtenção de uma função de transferência de segunda ordem para o segundo tanque adicionou uma camada de desafio à modelagem. Encontrar uma função de transferência entre o primeiro e o segundo tanque foi um passo importante para relacionar os dois sistemas.

A decisão de implementar um controlador Proporcional-Integral-Derivativo (PID) para o segundo tanque era uma escolha apropriada, dada a natureza de segunda ordem da função de transferência e as necessidades de melhora no overshoot, tempo de estabilização e erro de regime; e funcionaram para o modelo de planta linearizado no domínio da frequência. No entanto, os resultados do controle não linear do segundo tanque não foram tão bem-sucedidos quanto no primeiro tanque. Isso é atribuído às limitações das aproximações de linearização, que podem não capturar totalmente a dinâmica não linear do sistema real, além da propagação dos erros de linearização, já que foi feita considerando duas funções de transferência.

É importante destacar que sistemas não lineares frequentemente desafiam a aplicação de técnicas de controle clássicas, como a linearização. Soluções alternativas, como o controle adaptativo ou o controle não linear, poderiam ser consideradas em situações futuras para melhorar o desempenho do controle do segundo tanque.

Além disso, os testes bem-sucedidos com um controlador que limita a tensão na bomba na faixa de -12V a +12V para o primeiro tanque são promissores. Isso mostra que é

possível integrar considerações práticas, como limitações de hardware, no projeto de controle.

Em conclusão, este relatório enfatiza a importância de abordar sistemas reais com uma compreensão sólida dos princípios de controle e modelagem. O sucesso no controle do primeiro tanque ilustra a eficácia das técnicas tradicionais em sistemas lineares, enquanto os desafios enfrentados no segundo tanque ressaltam a necessidade de abordagens mais avançadas em sistemas não lineares. A pesquisa contínua e a exploração de técnicas de controle mais avançadas são essenciais para lidar com sistemas complexos e melhorar o desempenho de controle em ambientes do mundo real. Este experimento fornece uma base sólida para futuras investigações e melhorias no controle de sistemas de tanques interligados e para o entendimento geral da disciplina Sistemas de Controle 2.

4 Referencial teórico

- [1] KATSUHIKO OGATA. Engenharia de controle moderno. São Paulo (Sp): Pearson Prentice Hall, 2011.
- [2] MARTINS, Allan de Medeiros. Notas de aula: Linearização. LACI - UFRN, [s. l.], [202-?]. Disponível em: <https://www.dev-mind.blog/apps/board/>. Acesso em: 12 set. 2023.
- [3] NISE, N. S. Control systems engineering. [s.l.] Weinheim Wiley, 2019.
- [4] LINK DO VÍDEO ACERCA DA IMPLEMENTAÇÃO.