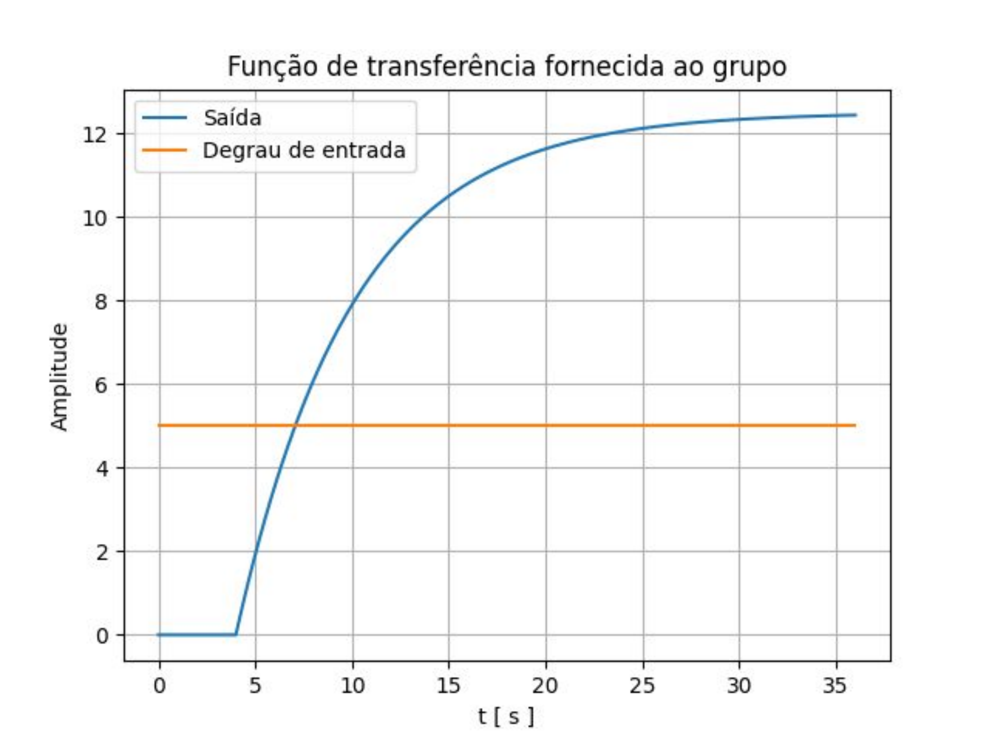
Trabalho de Sistemas Embarcados (C213) Controle Clássico

Gualter Machado Mesquita
Isabela Rezende Barbosa da Silva

Função de Transferência do Grupo



Parâmetros observáveis

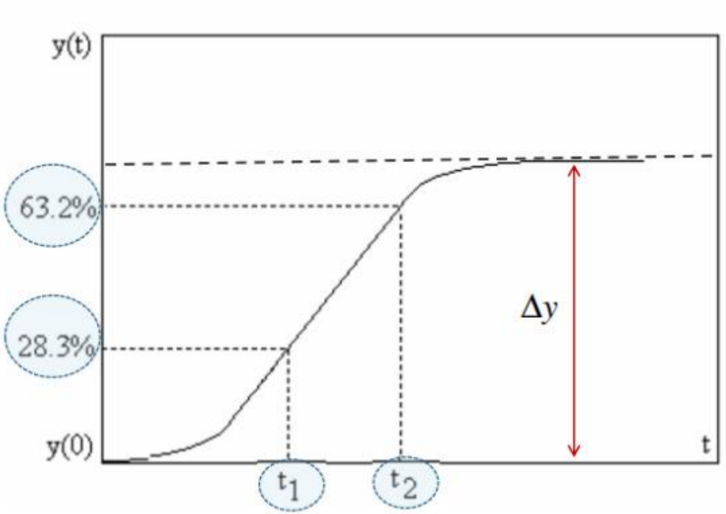
Valor do degrau: 5

Valor máximo da saída: 12.44

Tempo de atraso: 4s

Encontrando os valores de k, ϴ e τ

Método de Smith



$$K = \frac{\Delta y}{\Delta u}$$

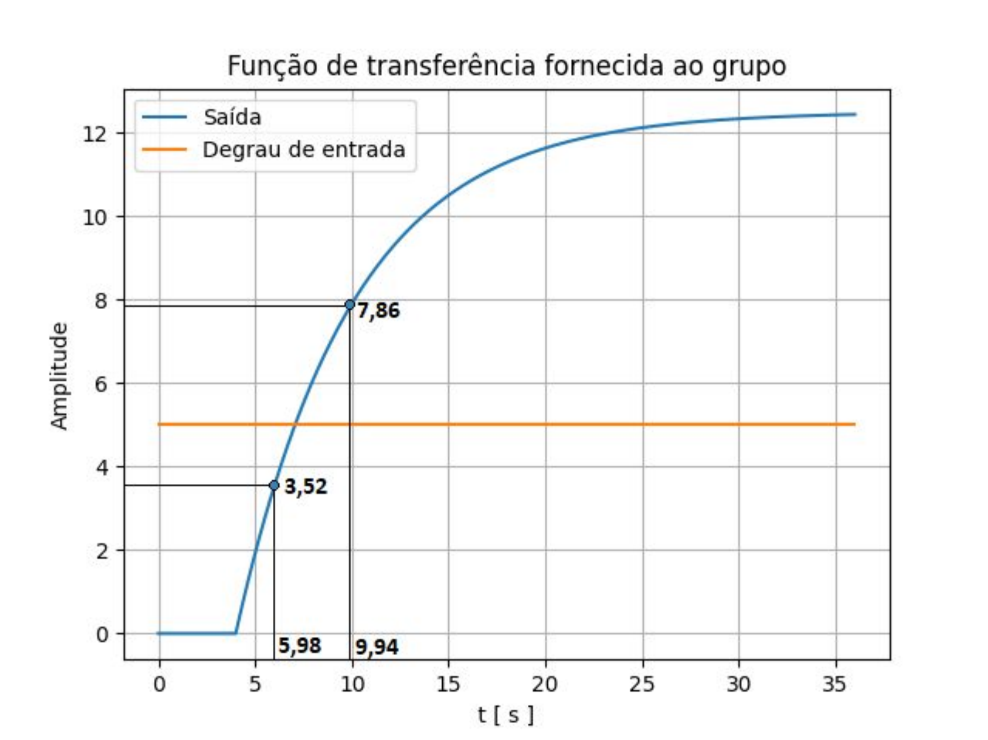
$$\tau = 1.5(t_2 - t_1)$$

$$\theta = t_2 - \tau$$

Encontrando os valores de k, ϴ e τ

```python
# Exercício 2 - Escolha o método de identificação da planta e com isso encontre os valores de k, θ e τ

# Valores obtido pelo método de Smith
y_max = max(saida)                    # Valor máximo do sinal
d_Y = y_max - min(saida)              # Delta do sinal
d_u = 5                               # Valor do degrau unitário

# Obtendo o valor do sinal em 28,3% para determinar t1
y_t1 = y_max * 0.283                  # y(t1) = 3.52
t1 = 5.98                             # Obtido por observação do gráfico

# Obtendo o valor do sinal em 63,2% para determinar t2
y_t2 = y_max * 0.632                  # y(t2) = 7.86
t2 = 9.94                             # Obtido por observação do gráfico

print(y_t1, y_t2)
# Obtendo valores de k, θ e τ
k = d_Y/d_u                           # k = 2.488
tau = 1.5 * (t2 - t1)                 # τ = 5.94
theta = t2 - tau                      # θ = 4

print("Ganho: ", k, "Constante de Tempo: ", tau, "Tempo de atraso: ", theta)
```

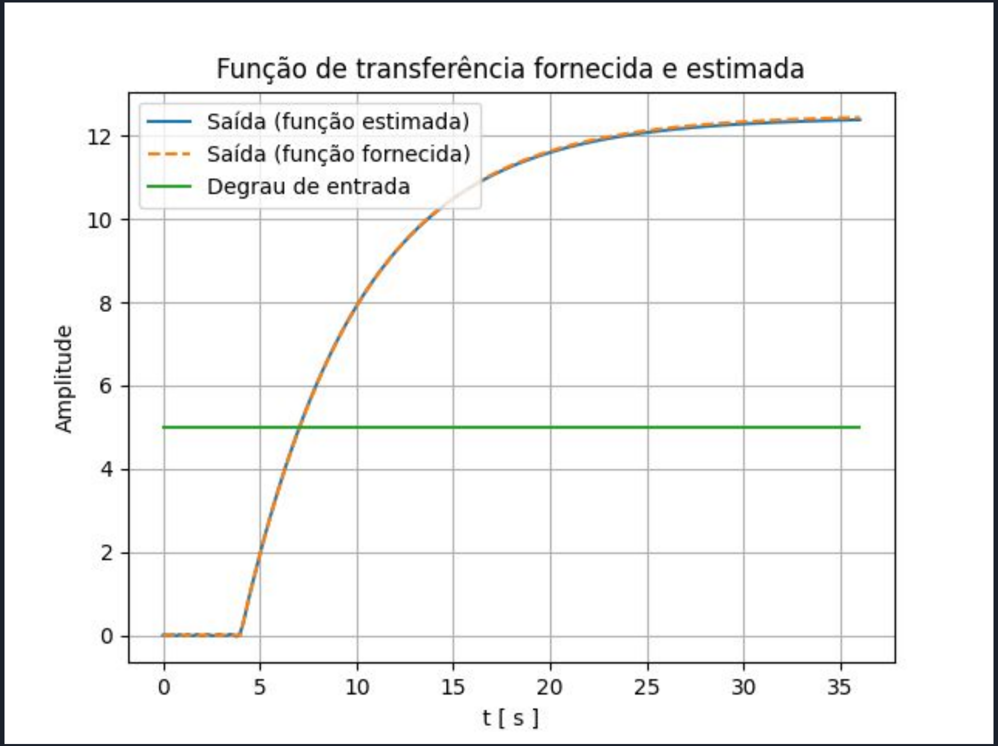
Encontrando os valores de k, ϴ e τ



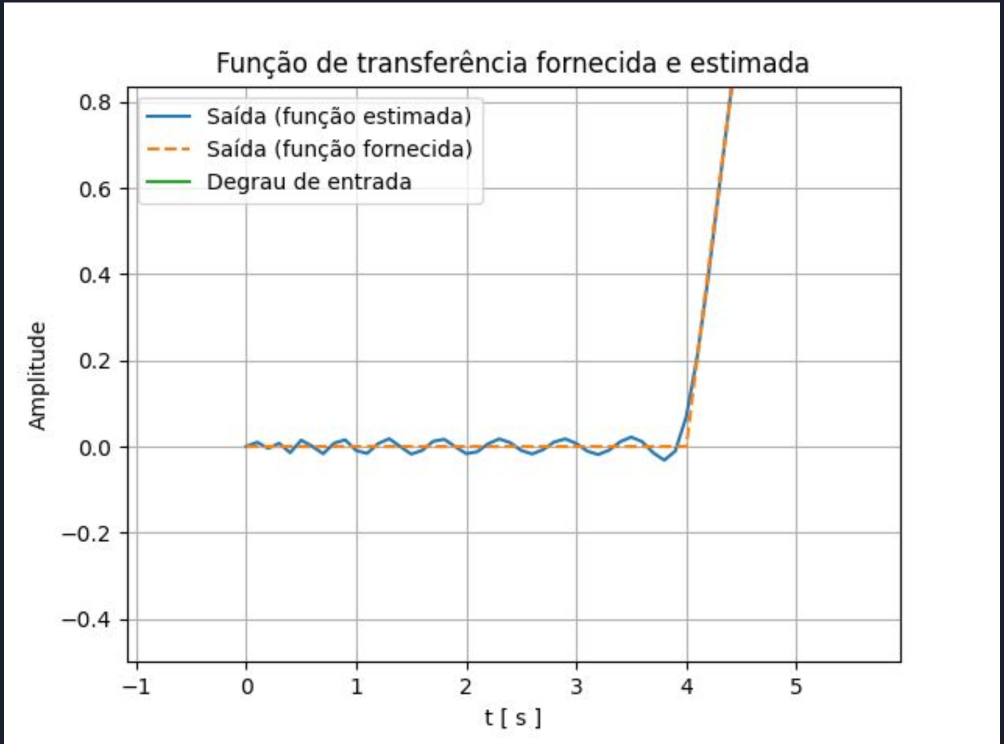
Estimando a função de transferência

```
# Definindo as variáveis da função de transferência em primeira ordem
k = 2.488
tau = 5.94
theta = 4

# Construindo a função de transferência
num = np.array([k])
den = np.array([tau, 1])
H = cnt.tf(num, den)

# Montando o sistema
n_pade = 20
(num_pade, den_pade) = cnt.pade(theta, n_pade)
H_pade = cnt.tf(num_pade, den_pade)
Hs = cnt.series(H, H_pade)

# Simulando a resposta da função de transferência estimada
time, y = cnt.step_response(5*Hs, T=t1)
```

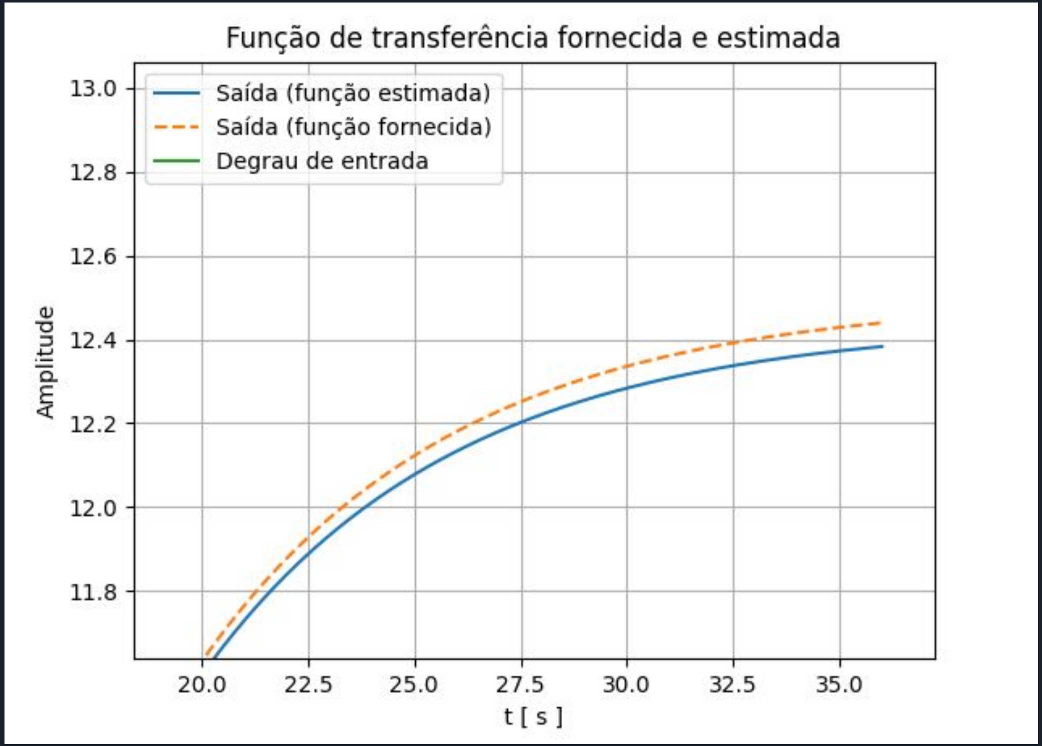
Plotando a função fornecida e a estimada



Ruído no início da função fornecida durante tempo morto



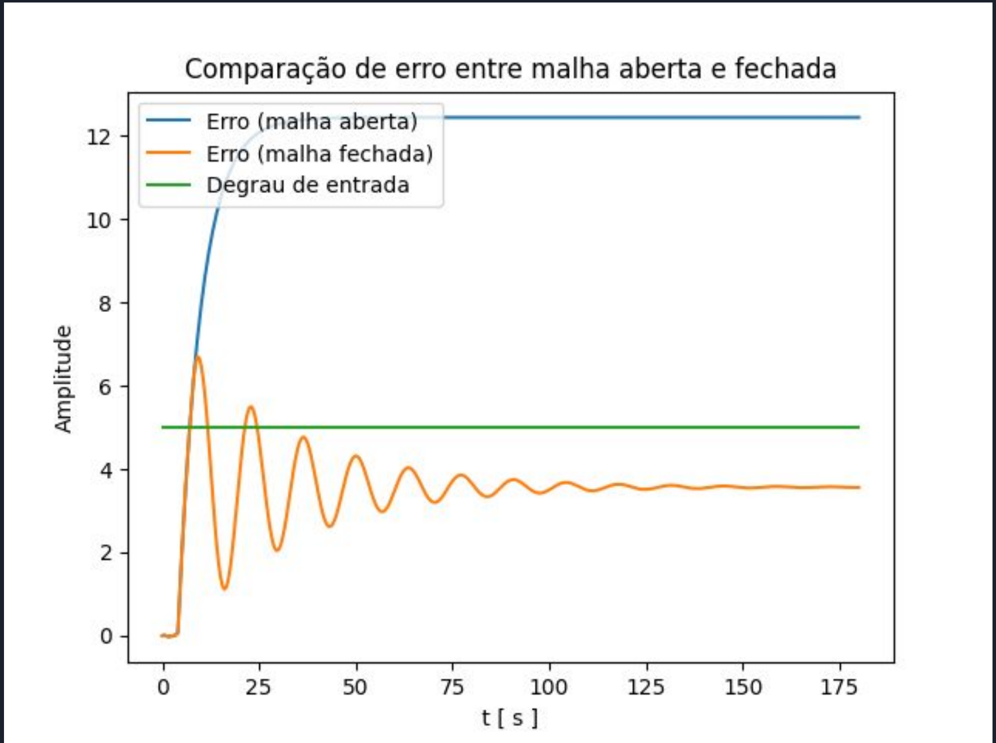
Erro em relação a função fornecida



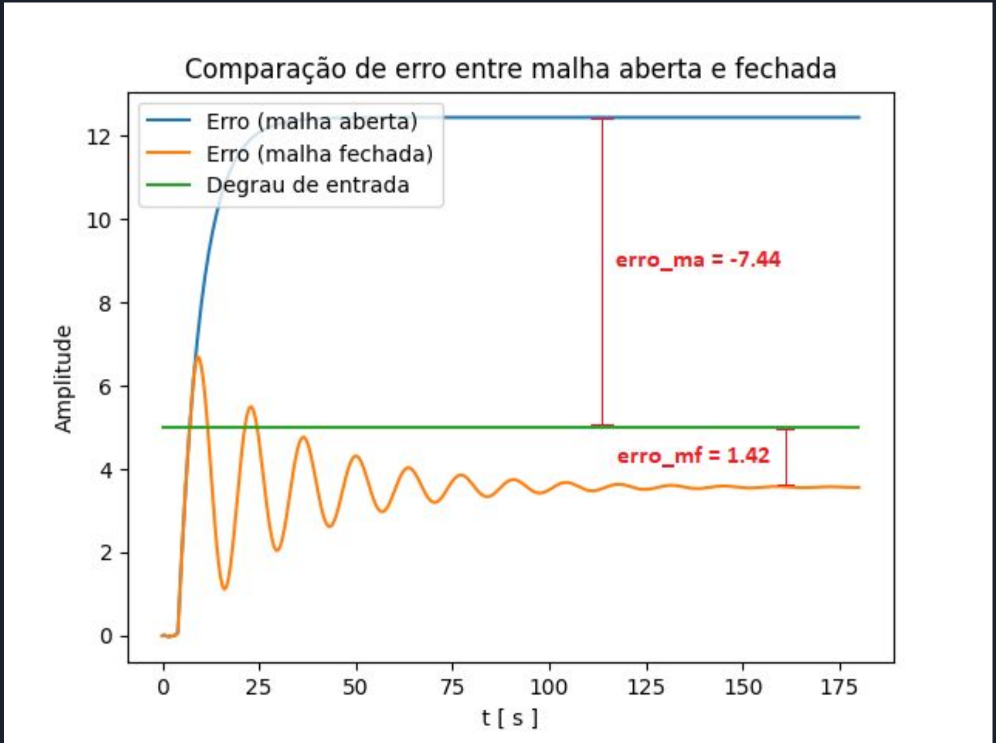
Calculando o erro em malha aberta e fechada

```python
# Definindo as variáveis da função de transferência em primeira ordem
k = 2.488
tau = 5.94
theta = 4

# Escrevendo a função de transferência da planta
num = np. array([k])
den = np. array([tau, 1])
H = cnt.tf(num, den)
n_pade = 20
(num_pade, den_pade) = cnt.pade(theta, n_pade)
H_pade = cnt.tf(num_pade, den_pade)

# Simulando funções para malha aberta e fechada, respectivamente
Hs = cnt.series(H, H_pade)
Hmf = cnt.feedback(Hs, 1)
```

Plotando os gráficos



Análise do erro pelo gráfico (Setpoint - Valor final)



Cálculo do erro pelo código

```python
# Calculando o erro em malha aberta e fechada
erro_ma = 5 - max(saida)
erro_mf = 5 - 3.58
print(f'Erro da malha aberta: {erro_ma[0]}')
print(f'Erro da malha fechada: {erro_mf}')
```

```
Erro da malha aberta: -7.4396506250770855
Erro da malha fechada: 1.42
```

Sobre os valores de erro obtidos

- A função em malha aberta possui um erro constante de -7.44, pois não há realimentação. Isso significa que a resposta temporal em regime estacionário permanece em 12.44.
- Com a realimentação na malha fechada, há uma melhoria no valor final com um erro de 1.42, ou seja, a resposta temporal em regime estacionário atinge o valor 3.58.
- Enquanto o valor da malha fechada é uma melhoria comparado a malha aberta, podemos chegar no valor de referência utilizando um controlador PID.
- Os métodos utilizados para construir os controladores foram o IMC (Clássico) e Integral do Erro (Novo)

Método do Modelo Interno (IMC)

- Foi proposto por Rivera et al (1986).
- Utiliza um modelo interno do processo, que utiliza a função de transferência da planta para determinar os ajustes dos parâmetros PID.
- Seu uso presume-se um processo de baixa ordem sem atraso de resposta.
- A velocidade de resposta depende de um parâmetro λ. Kp, Ti e Td tornam-se funções deste parâmetro.
- Quanto menor o valor de λ, mais rápida é a resposta e melhor é o desempenho. Porém, quanto mais baixo for λ, mais sensível o processo será às perturbações.
- Para um controlador PID, sugere-se que $\lambda / \theta > 0.8$, ou seja, $\lambda > 0.8*\theta$.

IMC - Equações

Controlador	K_P	T_I	T_D	Sugestão para o Desempenho
PID	$\frac{2\tau+\theta}{K\times(2\lambda+\theta)}$	$\tau+\left(\frac{\theta}{2}\right)$	$\frac{\tau\times\theta}{(2\tau+\theta)}$	$\frac{\lambda}{\theta}>0.8$
PI	$\frac{(2\tau+\theta)}{K\times 2\lambda}$	$\tau+\left(\frac{\theta}{2}\right)$	—	$\frac{\lambda}{\theta}>1.7$

[Rivera et al., 1986]

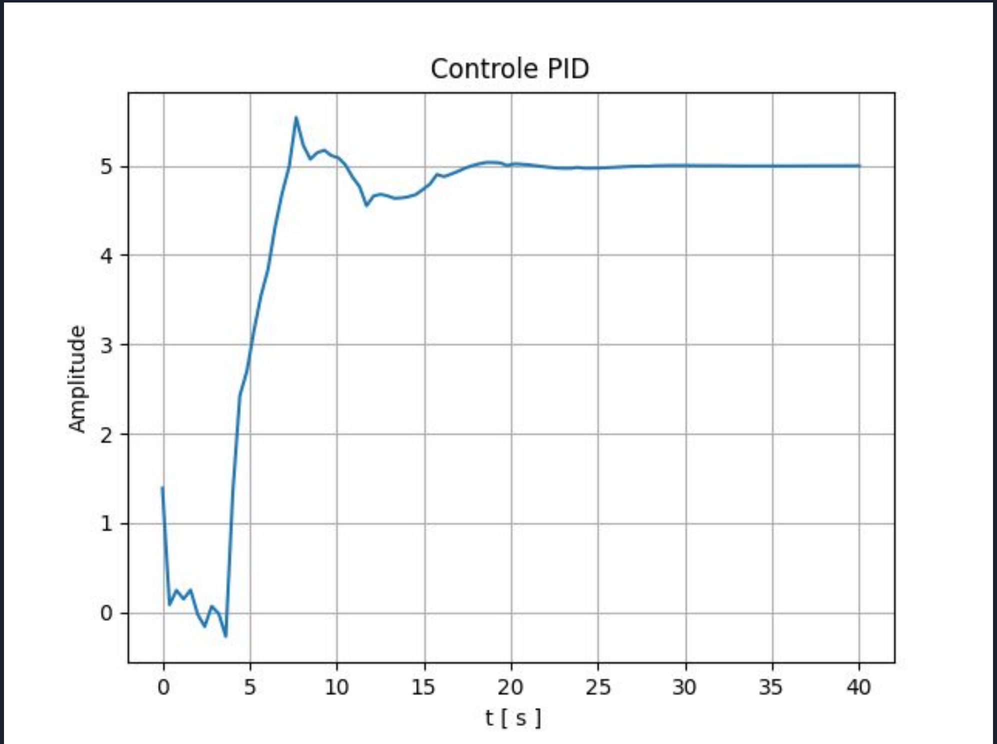
Implementando o IMC

```python
# MÉTODO TRADICIONAL - IMC com Controlador PID
lambda_var = 3.21     # lambda deve ser maior que 3.2
kp = ((2 * tau + theta) / (k * (2 * lambda_var + theta)))
ti = tau + (theta / 2)
td = (tau * theta) / (2 * tau + theta)

# Escrevendo a função de transferência da planta
num = np.array([k])
den = np.array([tau , 1])
H = cnt.tf(num , den)
n_pade = 20
( num_pade , den_pade ) = cnt.pade ( theta , n_pade )
H_pade = cnt.tf( num_pade , den_pade )
Hs = cnt.series (H , H_pade)
```

Implementando o IMC (sem ajustes)

```
# Controlador proporcional
numkp = np.array([kp])
denkp = np.array([1])

# Controlador integral
numki = np.array([kp])
denki = np.array([ti,0])

# Controlador derivativo
numkd = np.array([kp*td,0])
denkd = np.array([1])

# Construindo o controlador PID
Hkp = cnt.tf(numkp , denkp)
Hki = cnt.tf(numki , denki)
Hkd = cnt.tf(numkd , denkd)
Hctrl1 = cnt.parallel (Hkp , Hki)
Hctrl = cnt.parallel (Hctrl1 , Hkd)
Hdel = cnt.series (Hs , Hctrl)

# Fazendo a realimentação
Hcl = cnt.feedback(Hdel, 1)

t = np.linspace(0 , 40 , 100)
( t , y ) = cnt.step_response (5 * Hcl, t )
```

Implementando o IMC (sem ajustes)

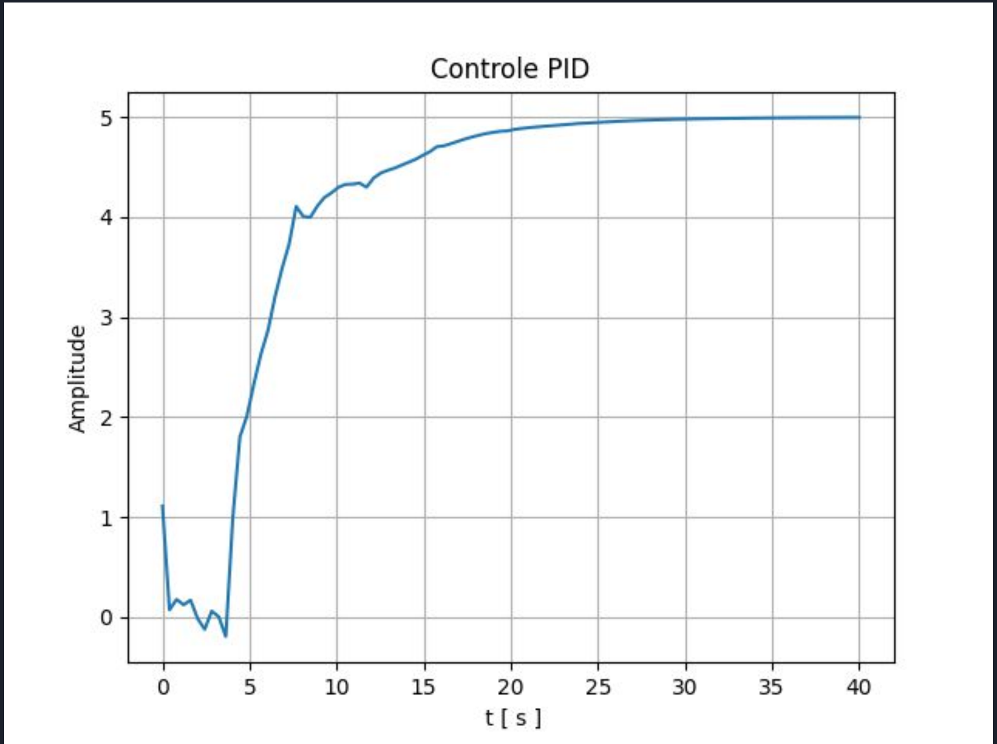


Lambda = 3.21

Kp, Ti, Td sem ajustes.

Como o IMC não prevê alterações para o tempo de atraso, temos este perfil do sinal, com um pouco de sobressinal e tempo de acomodação de aproximadamente 25s.

Implementando o IMC (com ajustes)

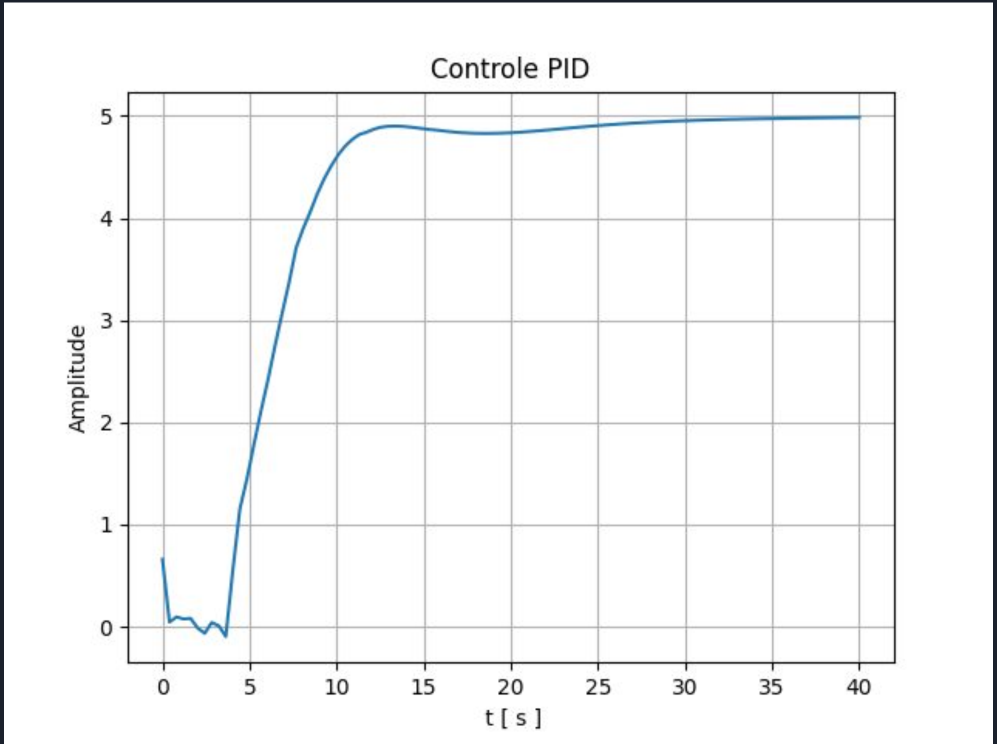


Lambda = 5

Kp, Ti, Td sem ajustes.

Ajustes maiores no Lambda causam maior atraso para chegar no tempo de acomodação e, para valores muito maiores, o erro aumenta. Tempo de acomodação em 30s.

Implementando o IMC (com ajustes)



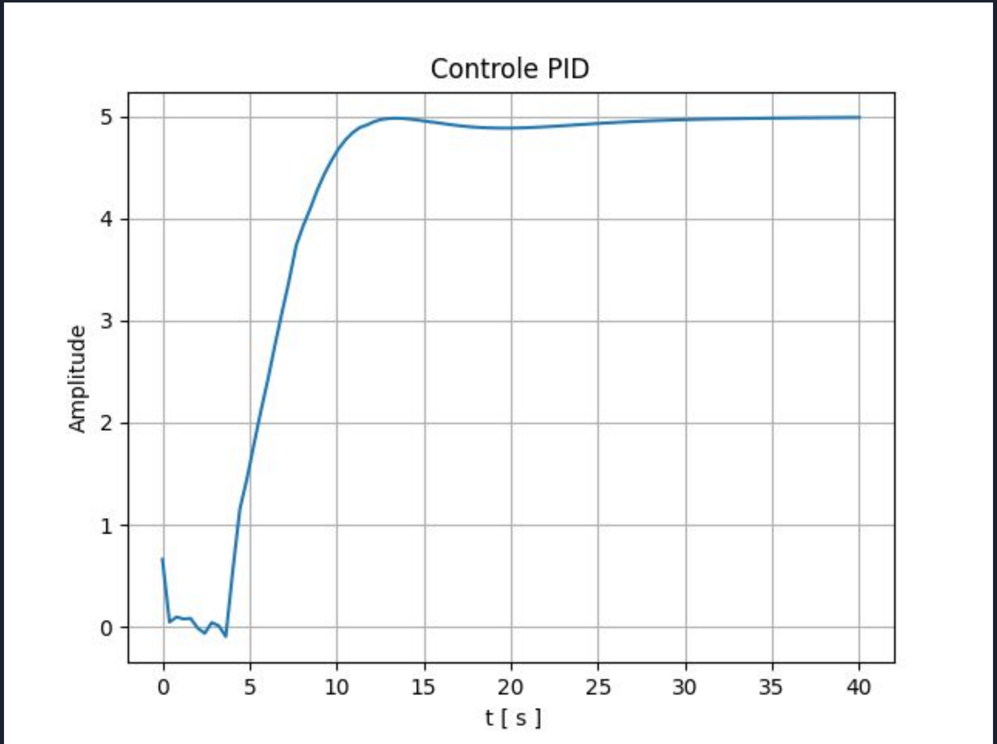
Lambda = 5

Td = -0.7

Kp, Ti sem ajustes.

O perfil do sinal antes do tempo de acomodação, com exceção do tempo de atraso de resposta, fica mais uniforme.

Implementando o IMC (com ajustes)



Lambda = 5

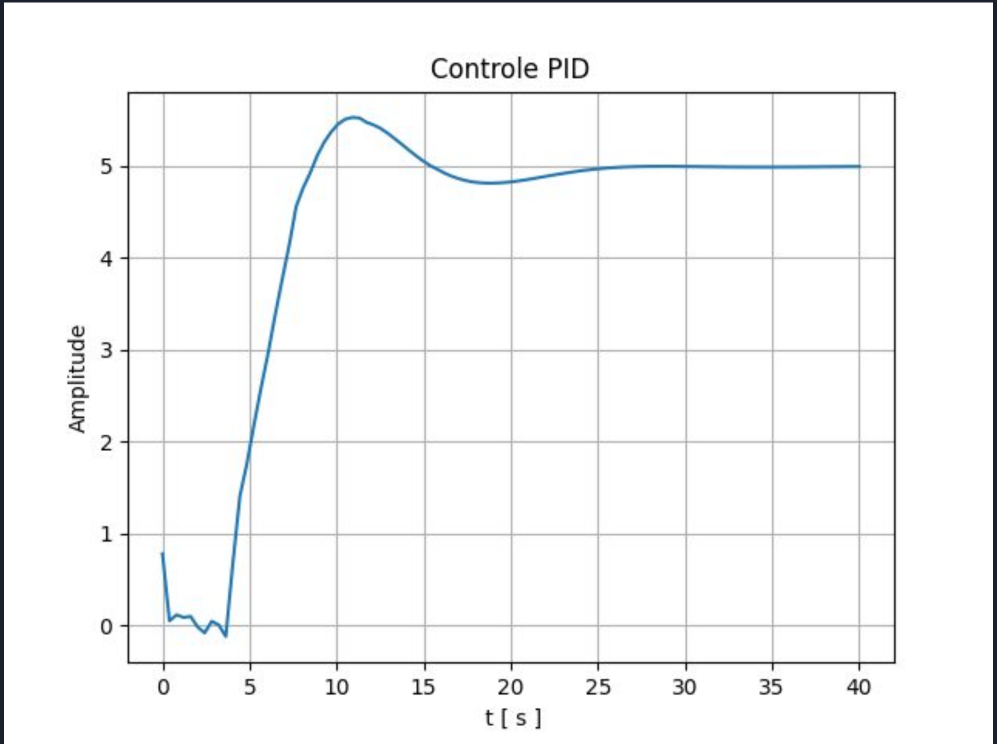
Td = -0.7

Ti = -0.3

Kp sem ajustes.

O sinal atinge o valor de referência mais cedo.

Implementando o IMC (com ajustes)



Lambda = 5

Td = -0.7

Ti = -0.3

Kp = +0.1

Introduz sobressinal e o tempo de acomodação cai para aproximadamente 25s.

Método da Integral do Erro

- O Método da Integral do Erro é baseado na redução do índice de desempenho do sistema a ser controlado e pode ser dividido basicamente em dois grupos: Método da Integral do Erro Quadrático e Método do Erro Absoluto.
- Cada um desses métodos possuem versões diferentes e as configurações possíveis para o controlador podem ter como critério de projeto a rejeição de distúrbios ou variações no set-point.

IAE (Integral do Erro Absoluto) - Equações

Fator Adimensional	IAE
$K_P \times K =$	$1 \Big/ \left(\left(\theta/\tau \right) + 0.2 \right)$
$\dfrac{T_I}{\theta} =$	$\left(0.3 \times \left(\theta/\tau\right) + 1.2\right) \Big/ \left(\left(\theta/\tau\right) + 0.08\right)$
$\dfrac{T_D}{\theta} =$	$1 \Big/ \left(90 \times \left(\theta/\tau\right)\right)$

IAE - Equações alternativas

Tabela 5. Configurações do controlador utilizando a integral do erro absoluto (Ho, Gan, B.Tay, & Ang., 1996).

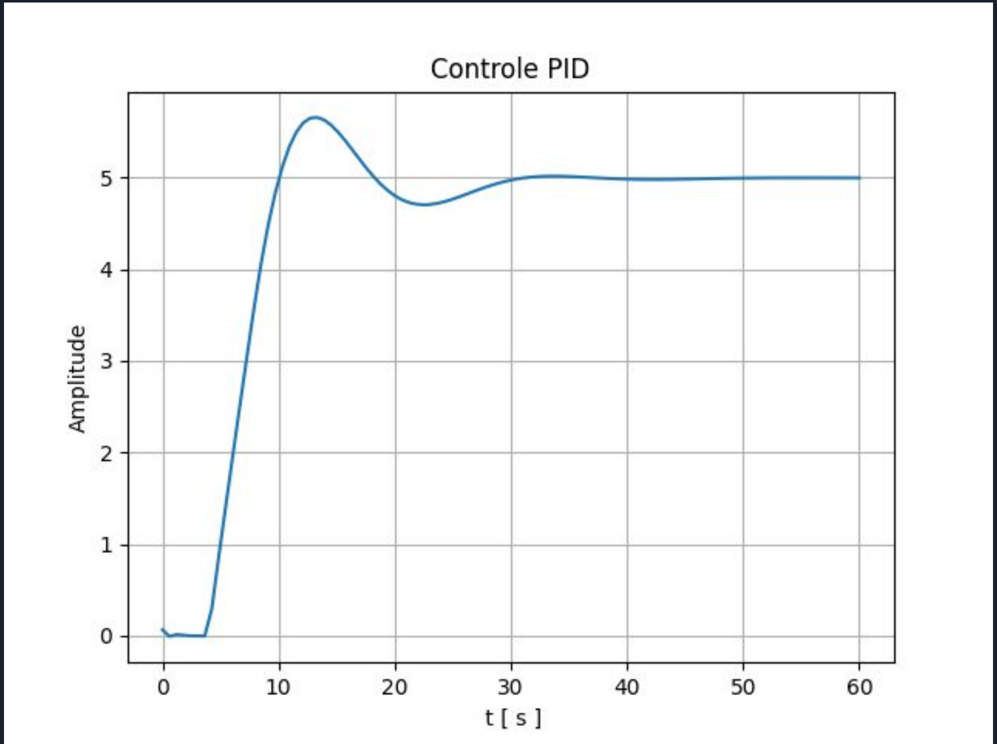
	Controlador	Kp	Ti	Td
e) IAE - variações de set-point	PID	$\frac{0,65}{K}\left(\frac{\theta}{\tau}\right)^{-1,04432}$	$\frac{T}{0,9895-\frac{0,09539\theta}{\tau}}$	$0,50814\tau\left(\frac{\theta}{\tau}\right)^{1,08433}$
f) IAE - rejeição de distúrbios	PID	$\frac{0,980890}{K}\left(\frac{\theta}{\tau}\right)^{-0,76167}$	$\frac{\tau}{0,91032}\left(\frac{\theta}{\tau}\right)^{1,05211}$	$0,59974\tau\left(\frac{\theta}{\tau}\right)^{0,89819}$
g) ITAE - variações de set-point	PID	$\frac{1,12762}{K}\left(\frac{\theta}{\tau}\right)^{-0,80368}$	$\frac{T}{0,99783-\frac{0,02860\theta}{\tau}}$	$0,42844\tau\left(\frac{\theta}{\tau}\right)^{1,0081}$
h) ITAE - rejeição de distúrbios	PID	$\frac{0,77902}{K}\left(\frac{\theta}{\tau}\right)^{-1,06401}$	$\frac{\tau}{1,14311}\left(\frac{\theta}{\tau}\right)^{0,70949}$	$0,57137\tau\left(\frac{\theta}{\tau}\right)^{1,03826}$

Fonte : https://www.abcm.org.br/upload/files/PII_IV_06.pdf

Implementando o IAE

```
# Definindo as variáveis da função de transferência em primeira ordem
k = 2.488
tau = 5.94
theta = 4

# MÉTODO Novo - Integral do Erro
kp = (1 / ((theta / tau) + 0.2) / k)
ti = (((0.3 * (theta / tau)) + 1.2)/((theta / tau) + 0.08)) * theta
td = (1 / (90 * (theta / tau))) * theta
```

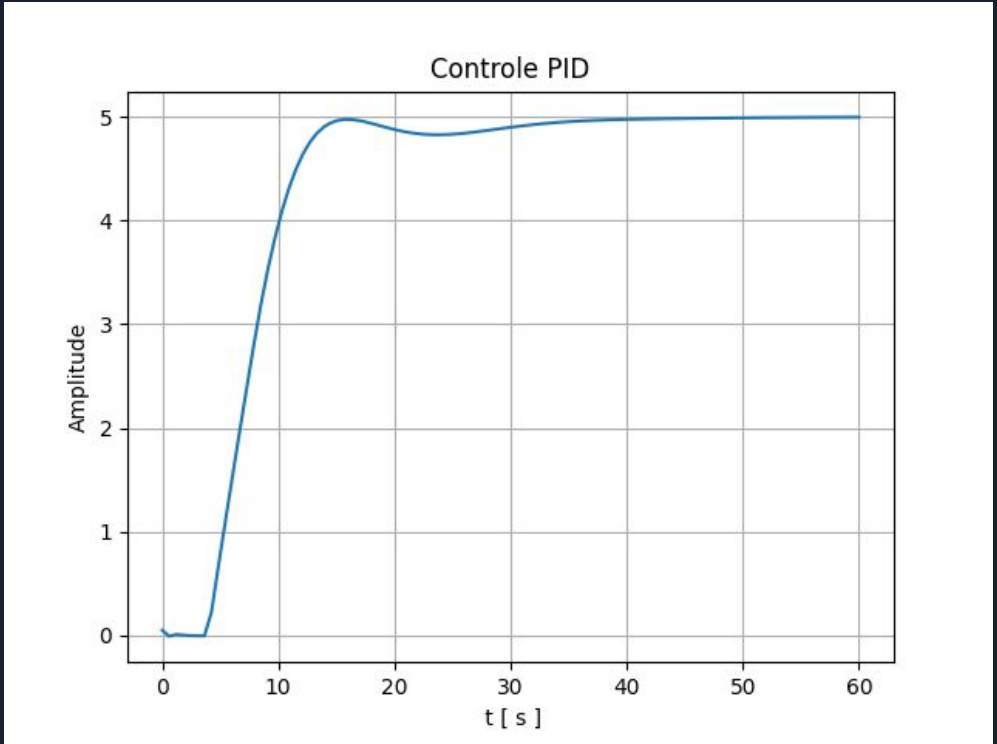
Implementando o IAE (sem ajustes)



Kp, Ti, Td sem ajustes.

Sem nenhum ajuste, o método da Integral do Erro Absoluto deixa o sinal muito parecido com o do IMC, com a exceção do tempo de acomodação. Nota-se que não há mais ruído no tempo de atraso.

Implementando o IAE (com ajustes)



Kp = -0.1

Ti, Td sem ajustes.

O ajuste em Kp remove o sobressinal, mantendo o mesmo tempo de acomodação. Ajustes maiores neste ou nos outros parâmetros introduzem sobressinal ou ruído no tempo de atraso.

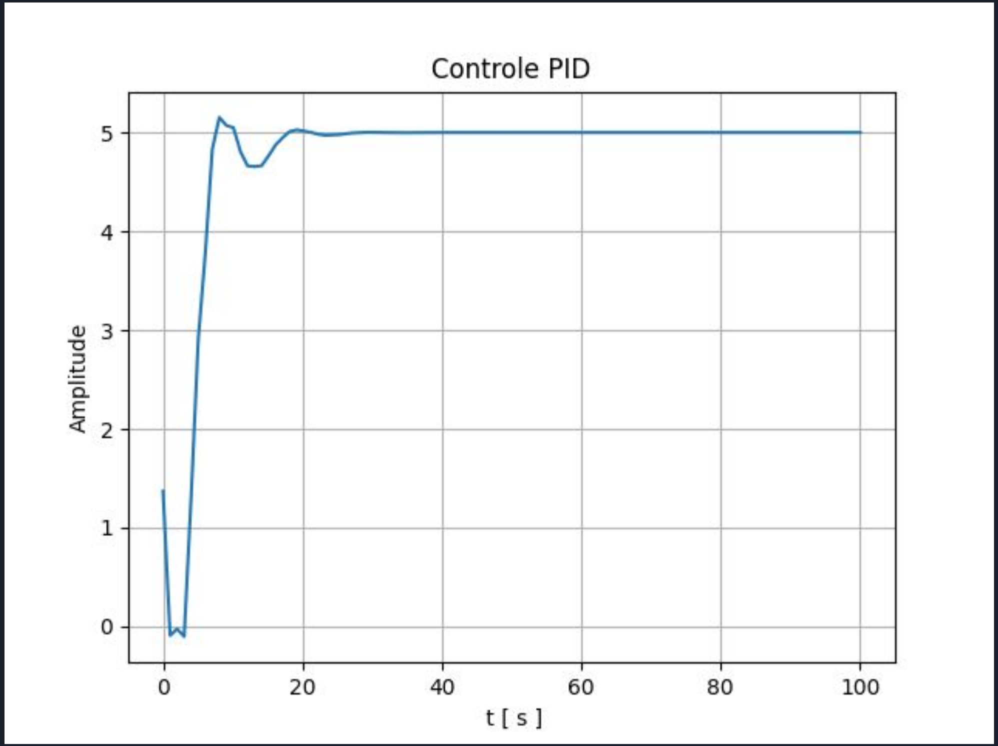
Comparativo entre o método IMC e o IAE

- Utilizando um controlador PID que implementa o modelo clássico de sintonia IMC faz com que o processo atinja o valor de referência com sucesso.
  - Os ajustes melhoram o perfil do sinal, mantendo o mesmo tempo de acomodação sem instabilidade. O Kp pode ser ajustado para melhorar o tempo de acomodação a custo de introduzir sobressinal.
  - O sinal durante o tempo de atraso sofre ruído.
- Um controlador PID que implementa o modelo IME também garante que o processo atinja o valor de referência, porém o sinal se mantém sem nenhum ruído com mínimas alterações no parâmetro Kp.
  - O tempo de acomodação nesse modelo foi maior do que o IMC.

Interface do usuário - IMC

```
Selecione o número do método de sintonia do controlador PID desejado:
{1}: Método do Modelo Interno (IMC)
{2}: Método da Integral de Erro Absoluto (IAE)
{3}: Sair
1
Entre com os parâmetros abaixo:
K (Ganho estático em malha aberta): 2.488
Tau (Constante de tempo): 5.94
Theta (Atraso de transporte): 4
Setpoint (Valor de referência): 5
```

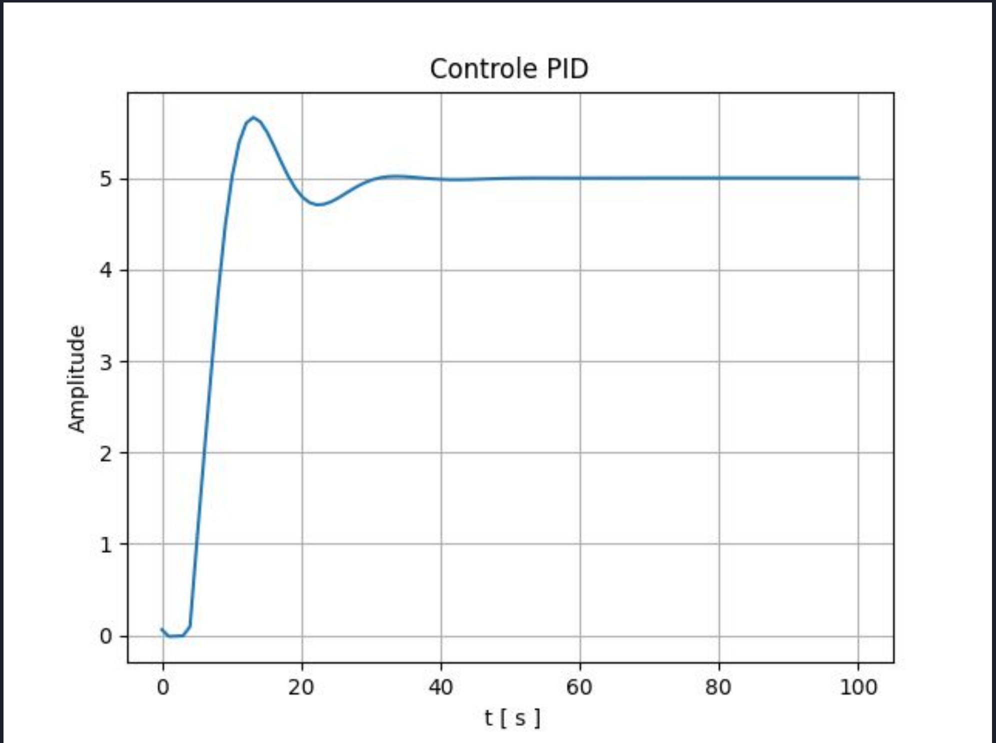
Interface do usuário - IMC Resultado



Interface do usuário - IAE

```
Selecione o número do método de sintonia do controlador PID desejado:
{1}: Método do Modelo Interno (IMC)
{2}: Método da Integral de Erro Absoluto (IAE)
{3}: Sair
2
Entre com os parâmetros abaixo:
K (Ganho estático em malha aberta): 2.488
Tau (Constante de tempo): 5.94
Theta (Atraso de transporte): 4
Setpoint (Valor de referência): 5
```

Interface do usuário - IAE Resultado



Obrigado pela atenção!