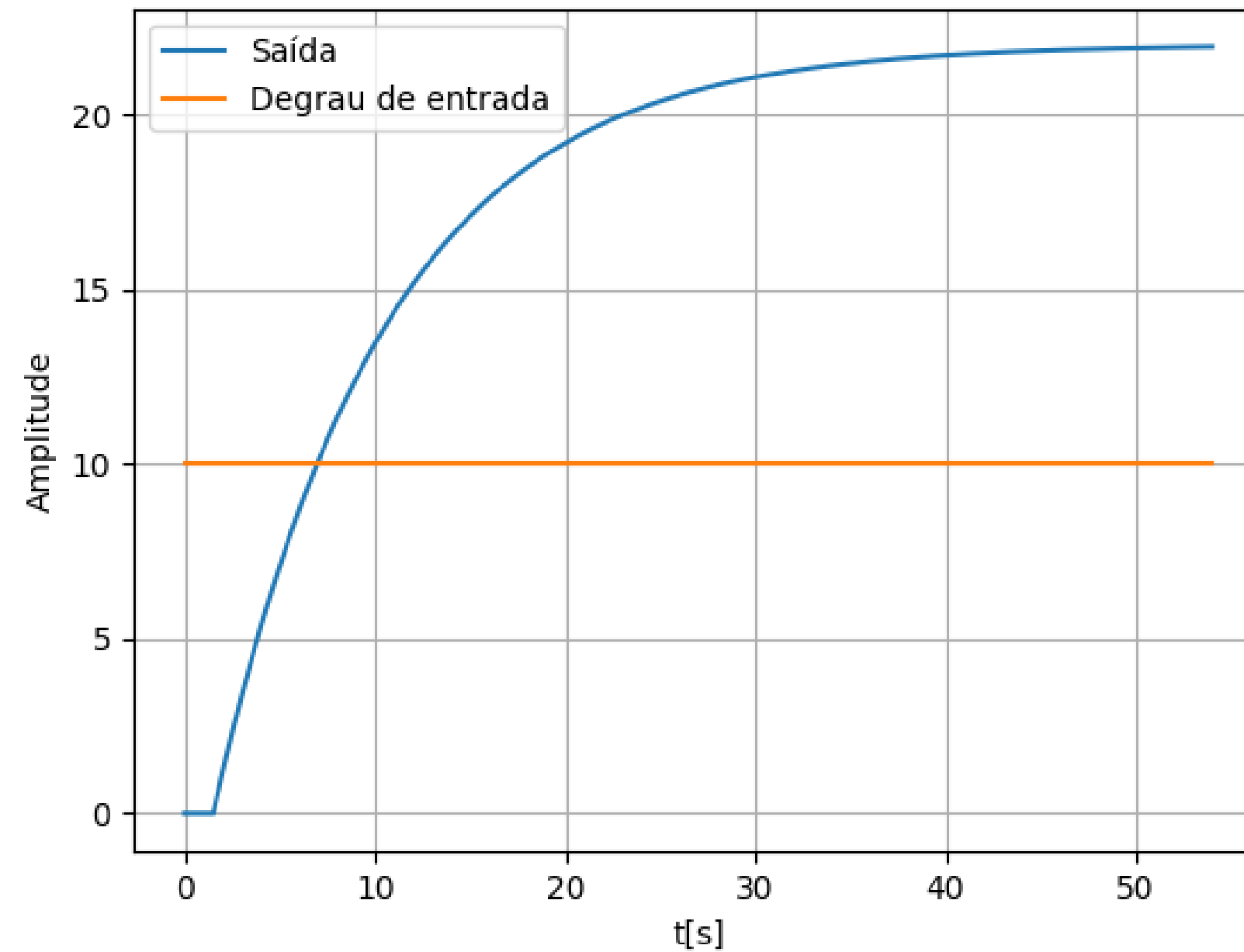


TRABALHO SOBRE CONTROLE CLASSICO

JOÃO PEDRO ANDRADE MOREIRA - 1723

LUCAS FRANQUEIRA CARNEIRO FONSECA-1734

FUNÇÃO DE TRANSFERÊNCIA



CALULANDO K,T E Θ

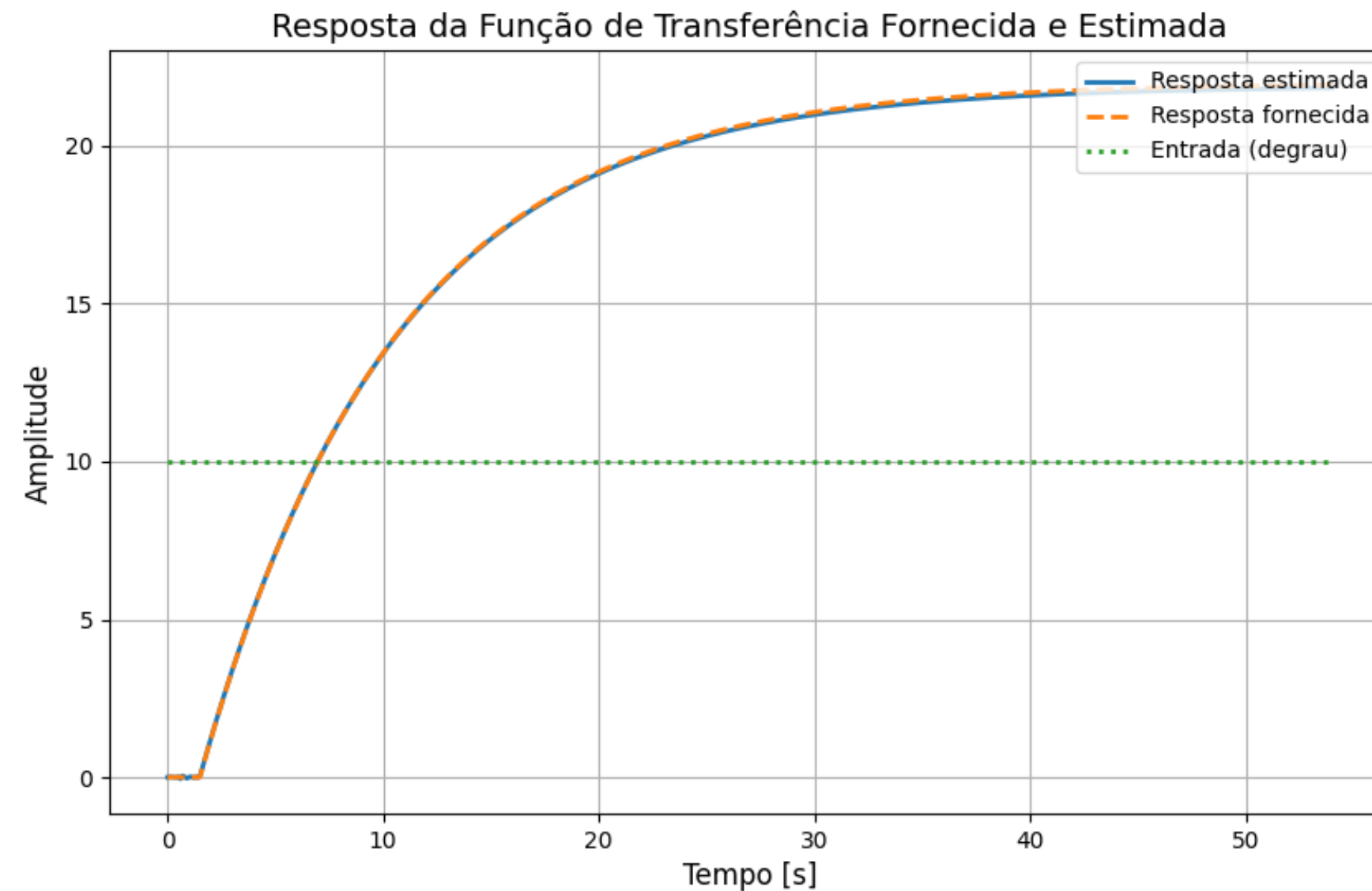
```
#Determinando t1 e t2
~~~~~
y_t1 = max_sinal * 0.283
y_t2 = max_sinal * 0.632

t1 = 4.48
t2 = 10.45
print(f'Y(t1) = {y_t1[0]:.2f}    t1 = {t1}')
print(f'Y(t2) = {y_t2[0]:.2f}    t2 = {t2}')

k = delta_sinal/degrau
tau = 1.5 * (t2 - t1)
theta = t2 - tau
print(f"K: {k[0]:.2f}")
print(f"τ: {tau:.2f}")
print(f"θ: {theta:.2f}")

Y(t1) = 6.21    t1 = 4.48
Y(t2) = 13.86    t2 = 10.45
K: 2.19
τ: 8.95
θ: 1.50
```

RESPOSTA FINAL VS RESPOSTA ESTIMADA



RESPOSTA FINAL VS RESPOSTA ESTIMADA

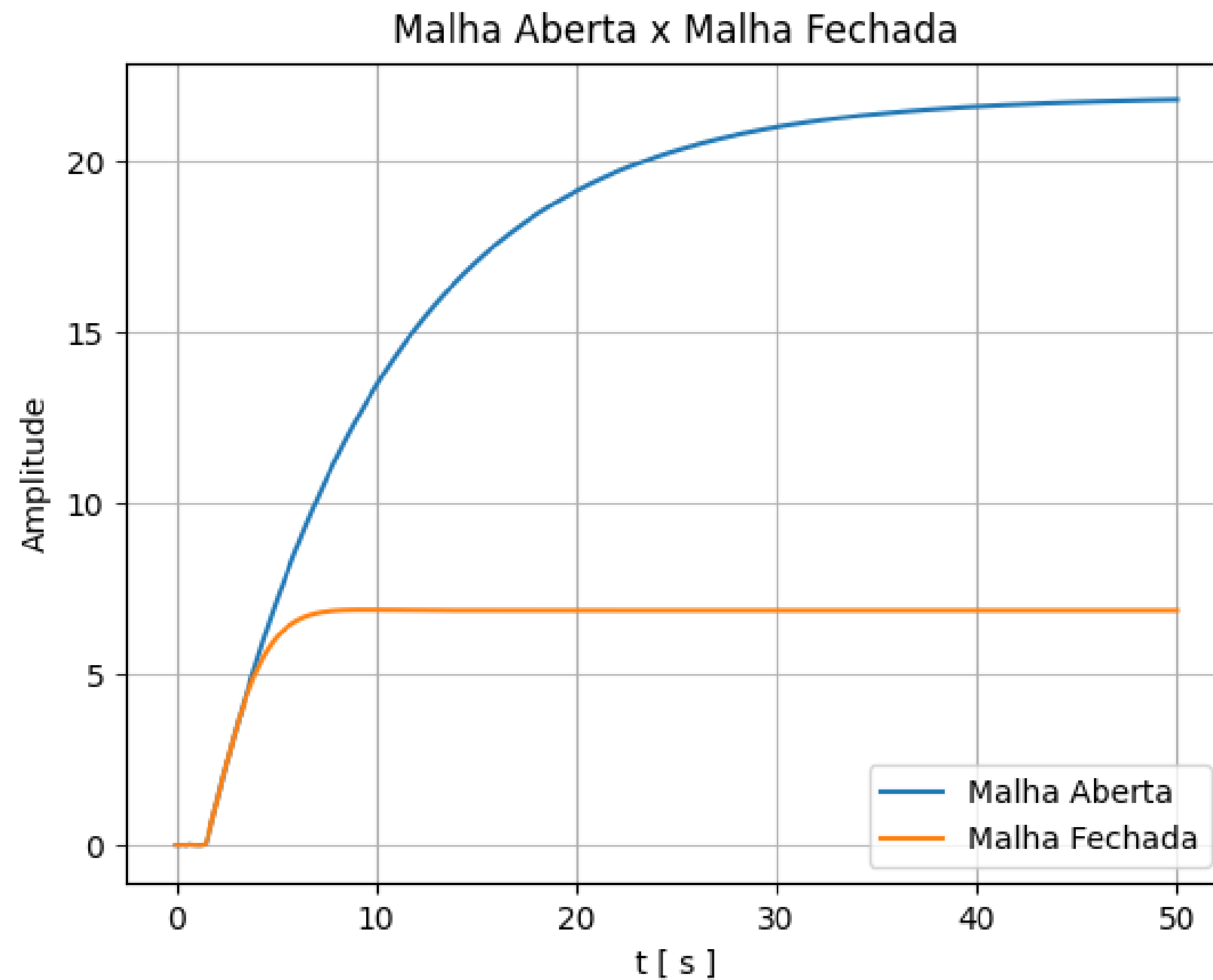
```
# Calcular a diferença entre a resposta estimada e a resposta fornecida em cada ponto do tempo
diff = y - saida.flatten()

# Encontrar o valor máximo absoluto dessa diferença
max_diff = np.max(np.abs(diff))

print("A maior diferença entre as respostas é:", max_diff)
```

A maior diferença entre as respostas é: 0.10026639413950988

MALHA ABERTA VS MALHA FECHADA



MALHA ABERTA VS MALHA FECHADA | CÓDIGO

```
num = np. array ([k]) # Cria um array contendo um único elemento, 'k'
den = np. array ([tau, 1])# Cria um array contendo o valor de 'tau' e 1
H = cnt.tf(num, den)# Cria uma funcao de transferencia

# Montar o sistema usando expansão de Pade
n_pade = 20 # Define a ordem da aproximação de Pade.
(num_pade, den_pade) = cnt.pade(theta, n_pade) # Gera polinômios para aproximação de Padé do atraso "theta".
H_pade = cnt.tf(num_pade, den_pade)# Cria uma função de transferência usando os polinômios obtidos.

Hs = cnt.series(H, H_pade)
Hmf = cnt.feedback(Hs, 1)
t = np.linspace(0, 50, 300)
```

Erro malha aberta: -11.82

Erro malha fechada: 3.13

METODO CHR

- DESENVOLVIDO POR CHIEN, HRONES E RESWICK EM 1952, O MÉTODO CHR É ELOGIADO POR SUA ABORDAGEM SIMPLES E DIRETA NA DETERMINAÇÃO DOS PARÂMETROS DO CONTROLADOR PID.
- UTILIZA A ANÁLISE DA RESPOSTA TRANSITÓRIA DO SISTEMA PARA AJUSTAR OS PARÂMETROS DO CONTROLADOR, CONSIDERANDO CARACTERÍSTICAS COMO TEMPO DE SUBIDA, TEMPO DE ACOMODAÇÃO E OVERSHOOT.
- PROPORCIONA FÓRMULAS EFICAZES PARA O CÁLCULO DOS GANHOS PROPORCIONAL, INTEGRAL E DERIVATIVO, BEM COMO PARA A ESTIMATIVA DO TEMPO INTEGRAL.
- FORNECE ORIENTAÇÕES PRÁTICAS E REGRAS EMPÍRICAS PARA DETERMINAR OS VALORES INICIAIS DOS PARÂMETROS DO CONTROLADOR PID, FACILITANDO A SINTONIA.
- AMPLAMENTE APLICÁVEL A DIVERSOS SISTEMAS DE CONTROLE, O MÉTODO CHR OFERECE RESULTADOS SATISFATÓRIOS EM MUITOS CENÁRIOS PRÁTICOS, ATENDENDO ÀS EXIGÊNCIAS DE DESEMPENHO DO SISTEMA DE CONTROLE.

METODO CHR

```
# Cálculo dos Parâmetros do Controlador PID usando as fórmulas CHR 2
kp_chr = (0.95*tau)/(k*theta)
ti_chr = 1.357*tau
td_chr = 0.473*theta

# Construção das Funções de Transferência para as partes P, I, e D usando CHR 2
numkp_chr = np.array([kp_chr])
denkp_chr = np.array([1])
numki_chr = np.array([kp_chr])
denki_chr = np.array([ti_chr, 0])
numkd_chr = np.array([kp_chr*td_chr, 0])
denkd_chr = np.array([1])
Hkp_chr = cnt.tf(*args: numkp_chr, denkp_chr)
Hki_chr = cnt.tf(*args: numki_chr, denki_chr)
Hkd_chr = cnt.tf(*args: numkd_chr, denkd_chr)
```

METODO COHEN E COON

- "COHEN E COON" É COMUMENTE ASSOCIADO A MÉTODOS DE SINTONIA DE CONTROLADORES EM SISTEMAS DE CONTROLE. NO CONTEXTO DE CONTROLE DE PROCESSOS, ZIEGLER E NICHOLS, COHEN E COON, SÃO NOMES BEM CONHECIDOS POR SUAS REGRAS E MÉTODOS PARA SINTONIZAR CONTROLADORES PID (PROPORCIONAL, INTEGRAL, DERIVATIVO).
- EM ESSÊNCIA, COHEN E COON PROPUSERAM UM MÉTODO DE SINTONIA QUE SE APLICA A SISTEMAS DE CONTROLE EM MALHA FECHADA, ONDE UMA RESPOSTA AO DEGRAU É REQUERIDA PARA DETERMINAR OS PARÂMETROS DO CONTROLADOR. OS PARÂMETROS PID SÃO DERIVADOS COM BASE NAS CARACTERÍSTICAS DA RESPOSTA AO DEGRAU DO SISTEMA, COMO O TEMPO DE SUBIDA, O TEMPO DE ATRASO E A VARIAÇÃO FINAL DA SAÍDA.
- A FORMA DE UTILIZAR AS REGRAS DE SINTONIA DE COHEN E COON É IDENTIFICAR ESSAS CARACTERÍSTICAS NA RESPOSTA AO DEGRAU DO SISTEMA CONTROLADO E, EM SEGUIDA, APLICAR FÓRMULAS ESPECÍFICAS PARA CALCULAR OS PARÂMETROS DO PID. A INTENÇÃO É FORNECER UMA SINTONIA QUE OFEREÇA UMA RESPOSTA DE SISTEMA SATISFATÓRIA, QUE FREQUENTEMENTE É UMA COMPENSAÇÃO ENTRE A VELOCIDADE DE RESPOSTA E A ESTABILIDADE DO SISTEMA.
- EM TERMOS SIMPLES, O MÉTODO COHEN-COON É UMA TÉCNICA USADA PARA DETERMINAR OS PARÂMETROS DE UM CONTROLADOR PID, DE MODO A OTIMIZAR A PERFORMANCE DO SISTEMA DE CONTROLE, FOCANDO EM OBTER UMA RESPOSTA AO DEGRAU SATISFATÓRIA NO SISTEMA DE MALHA FECHADA.

METODO COHEN E COON

```
# Construção das Funções de Transferência para as partes P, I, e D do Controlador PID
numkp = np.array([kp])
denkp = np.array([1])
numki = np.array([kp])
denki = np.array([ti, 0])
numkd = np.array([kp*td, 0])
denkd = np.array([1])

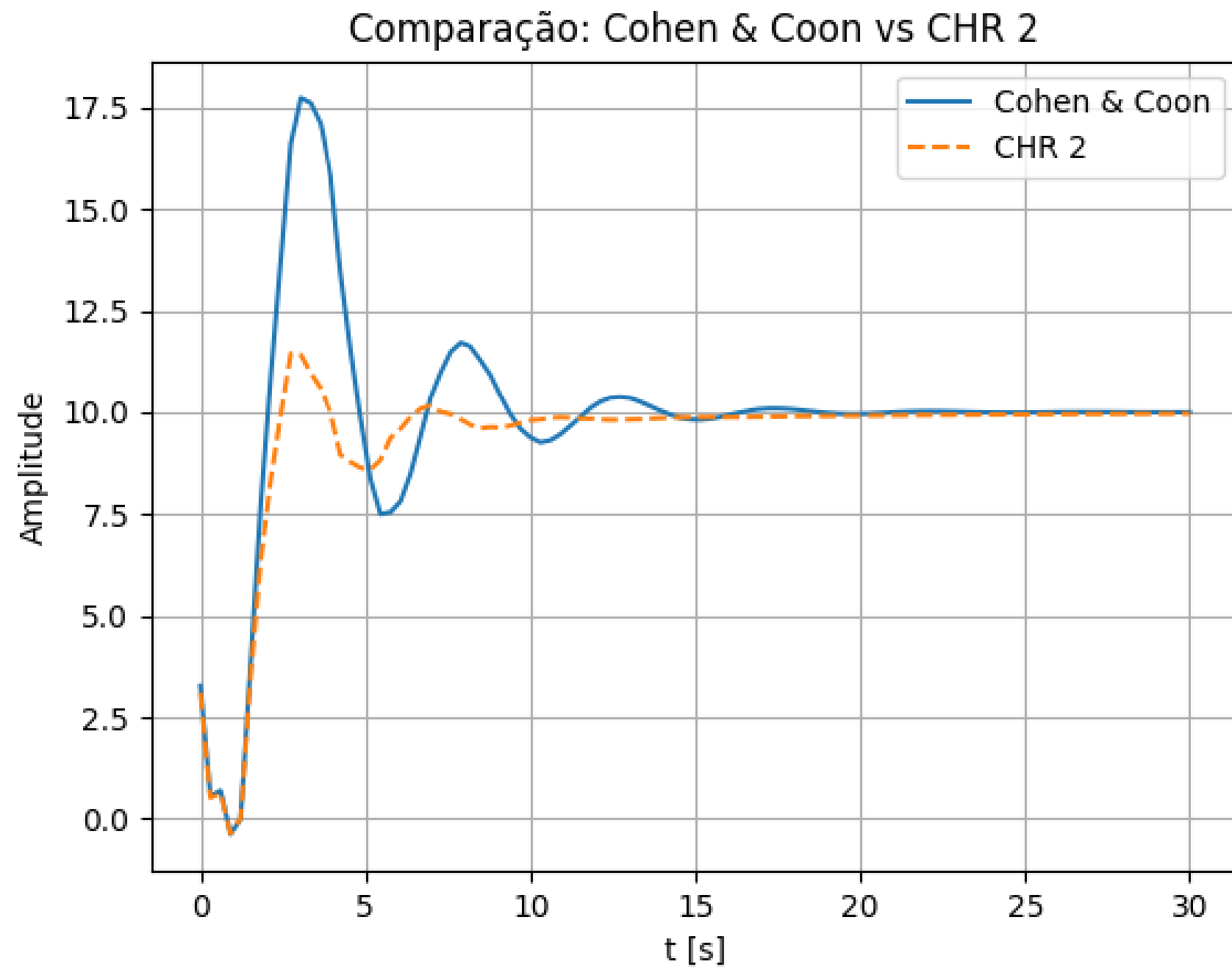
Hkp = cnt.tf(numkp, denkp)
Hki = cnt.tf(numki, denki)
Hkd = cnt.tf(numkd, denkd)

# Combinação das partes P, I, e D para formar o Controlador PID completo
Hctrl1 = cnt.parallel(Hkp, Hki)
Hctrl = cnt.parallel(Hctrl1, Hkd)

# Série da Planta com o Controlador PID
Hdel = cnt.series(Hs, Hctrl)

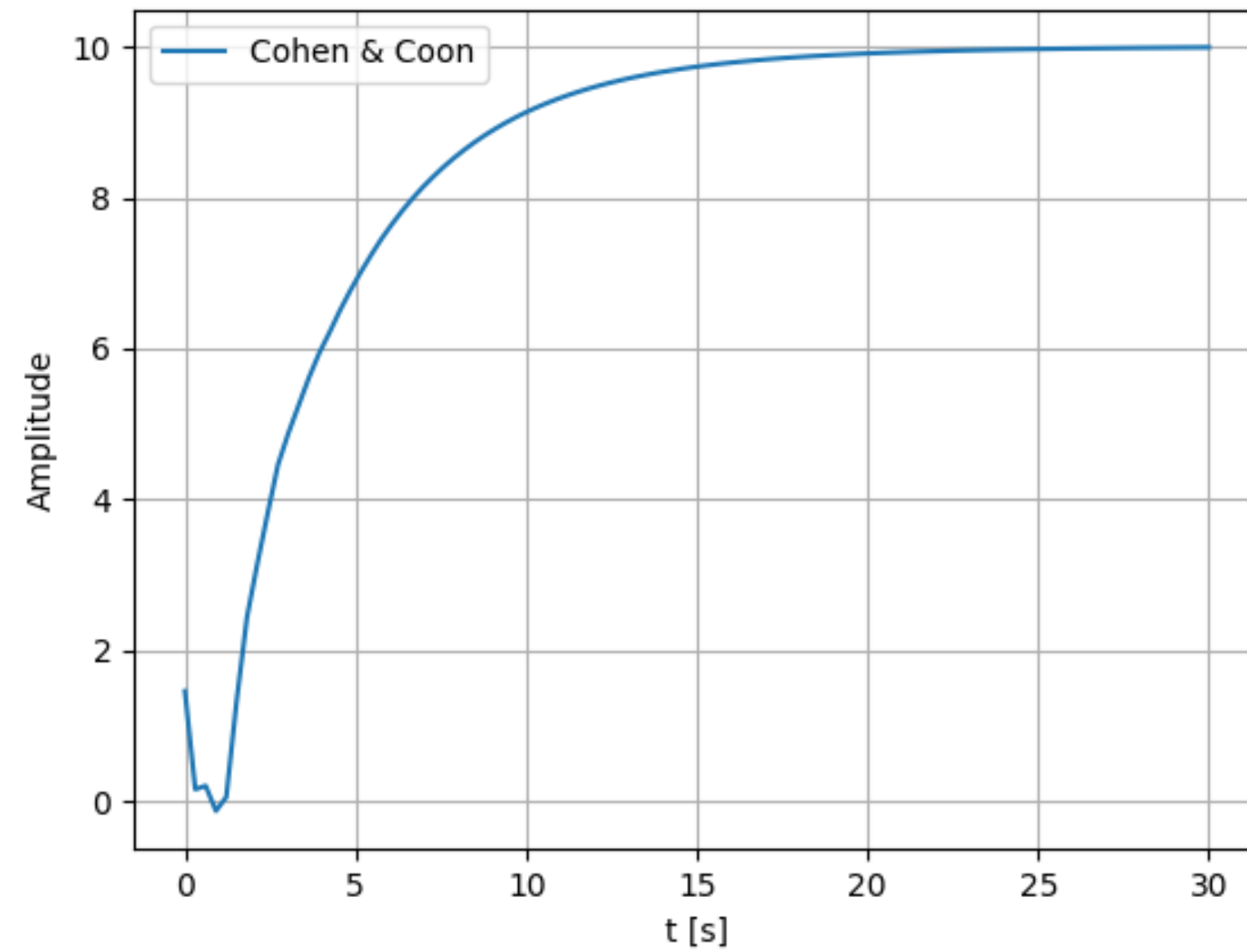
# Construção do Sistema de Controle em Malha Fechada
Hcl = cnt.feedback(Hdel, 1)
```

CHR VS COEN COON



COMO É POSSÍVEL OBSERVAR, O MÉTODO CHR DEMONSTROU UMA ESTABILIZAÇÃO MAIS RÁPIDA EM COMPARAÇÃO COM O MÉTODO COHEN-COON.

AJUSTES FINOS



```
#ajustes finos  
kp = kp - 2.77  
ti = ti + 6.57  
td = td + 0.174
```



**OBRIGADA PELA SUA
ATENÇÃO**