

TRABALHO 1

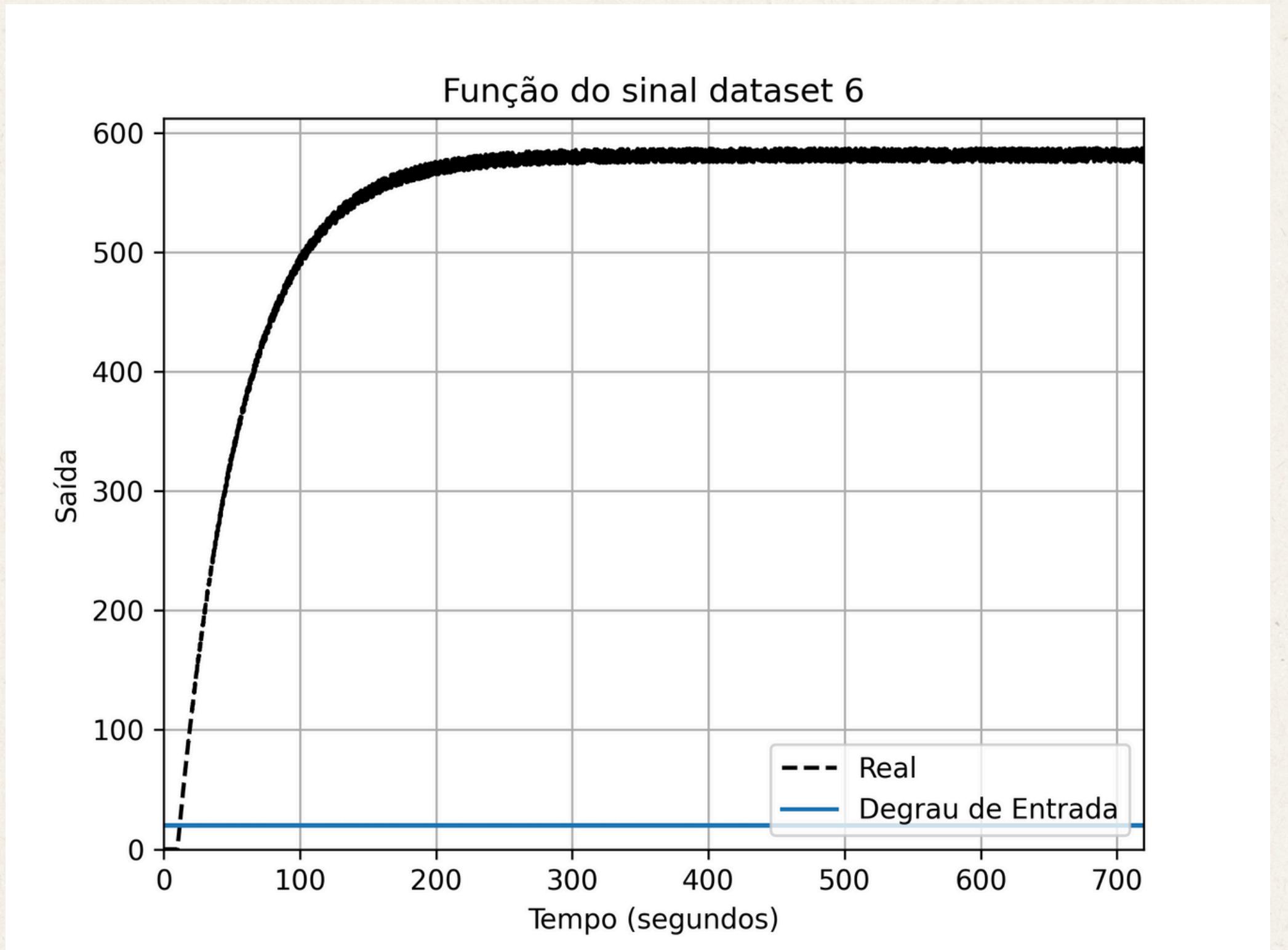
CONTROLE PID

NOMES DOS INTEGRANTES

Davi Rosa Gomes

Matheus Henrique Fonseca Afonso

FUNÇÃO DO SINAL DE DATASET 6

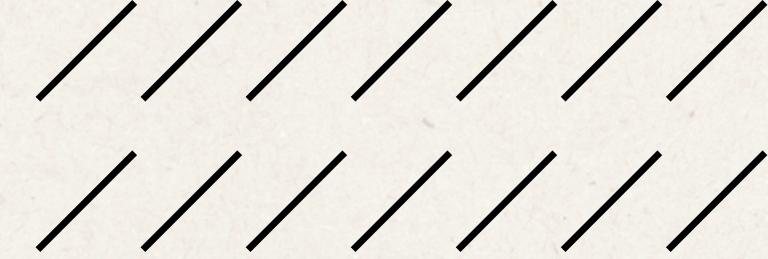


IDENTIFICAÇÃO DO MÉTODO

```
def identificaçãoSmithSundaresan(Step,Tempo, Saida, Method):
    tempo_t1 = 0
    tempo_t2 = 0
    ## SMITH
    if Method == "Smith":
        for i in range(len(Saida)):
            if Saida[i] >= 0.283*valor_final and tempo_t1 == 0:
                tempo_t1 = Tempo[i]
            if Saida[i] >= valor_final*0.632 and tempo_t2 == 0:
                tempo_t2 = Tempo[i]
                break
        tau = 1.5 * (tempo_t2 - tempo_t1)
        theta= tempo_t2 - tau
    else:
        ##SUNDARESAN
        for i in range(len(Saida)):
            if Saida[i] >= 0.353* valor_final and tempo_t1 == 0:
                tempo_t1 = Tempo[i]
            if Saida[i] >= 0.853 * valor_final:
                tempo_t2 = Tempo[i]
                break
        aux = 2/3
        tau = aux * (tempo_t2 - tempo_t1)
        theta = 1.3*tempo_t1 - 0.29*tempo_t2
    return K,tau,theta

K, tau_smith, theta_smith = identificaçãoSmithSundaresan(degrau, tempo, saida_motor, "Smith")
K, tau_sundaresan, theta_sundaresan = identificaçãoSmithSundaresan(degrau, tempo, saida_motor, "Sundaresan")
```

IDENTIFICAÇÃO DO MÉTODO



```
# ERRO QUADRATICO DE SMITH
sistema = ctl.tf(K, [tau_smith, 1])
num_delay, den_delay = ctl.pade(theta_smith, 20)
sistema_com_theta = ctl.series(ctl.tf(num_delay, den_delay),sistema)
tempo_simulado, saida_simulada = ctl.step_response(sistema_com_theta * amplitude_degrau, tempo)

erro_smith = np.sqrt((np.mean((saida_simulada - saida_motor) ** 2)))

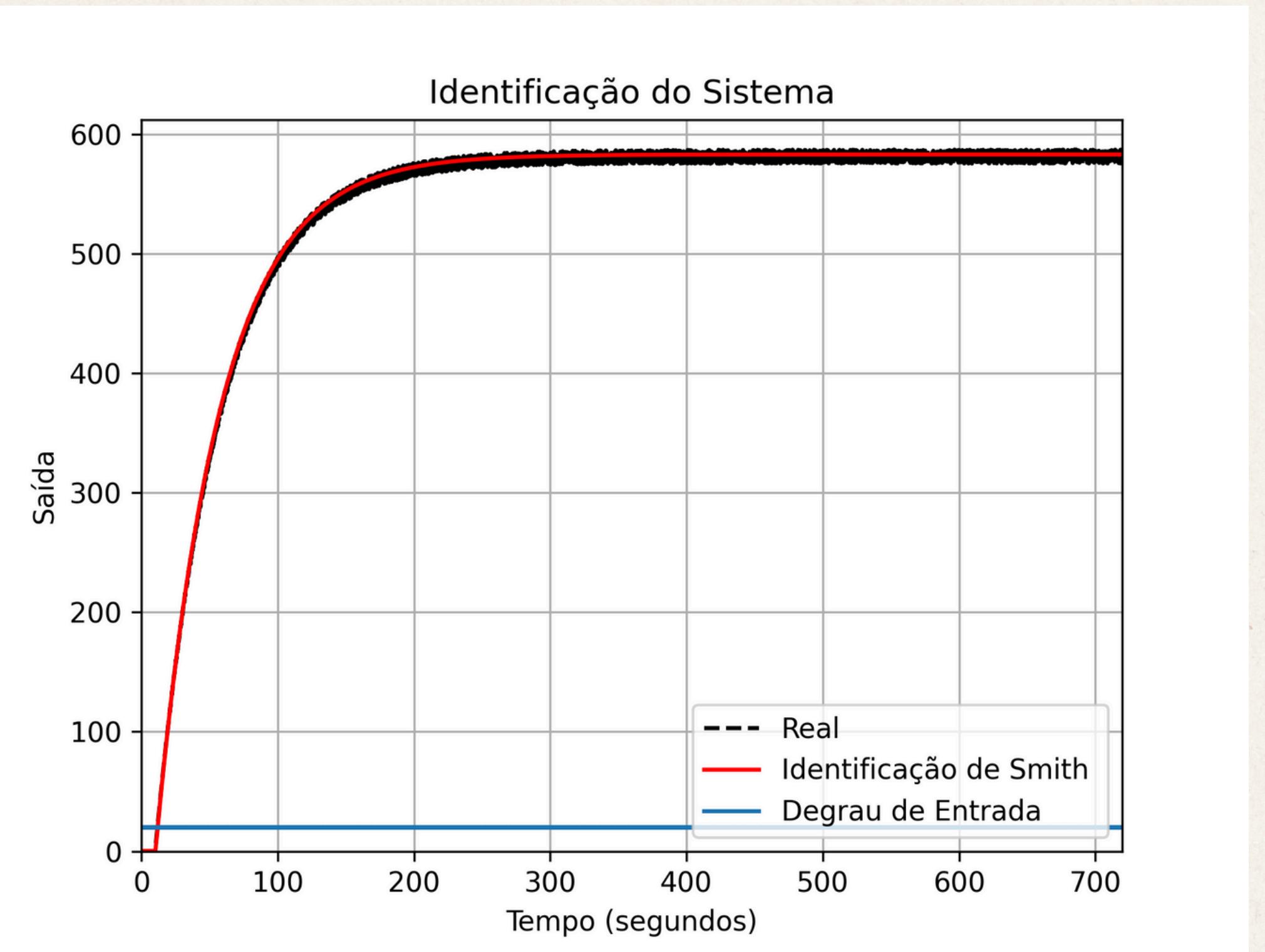
#ERRO QUADRATICO DE SUNDARESAN
sistema = ctl.tf(K, [tau_sundaresan, 1])
num_delay, den_delay = ctl.pade(theta_sundaresan, 20)
sistema_com_theta = ctl.series(ctl.tf(num_delay, den_delay),sistema)
tempo_simulado, saida_simulada = ctl.step_response(sistema_com_theta * amplitude_degrau, tempo)
# Calcular o Erro Quadrático Médio (EQM)
erro_sundaresan= np.sqrt((np.mean((saida_simulada - saida_motor) ** 2)))

if erro_smith < erro_sundaresan:
else:

print(f"K = {K}, Tau = {tau_estimado}, Theta = {theta_estimado}, EQM = {EQM} \nModelo utilizado foi de {nome}")
```

Método escolhido foi o Smith, portanto:
K = 29,05 Tau = 47,25
Theta = 10,35

SINAL REAL X SINAL ESTIMATIVO (SMITH)



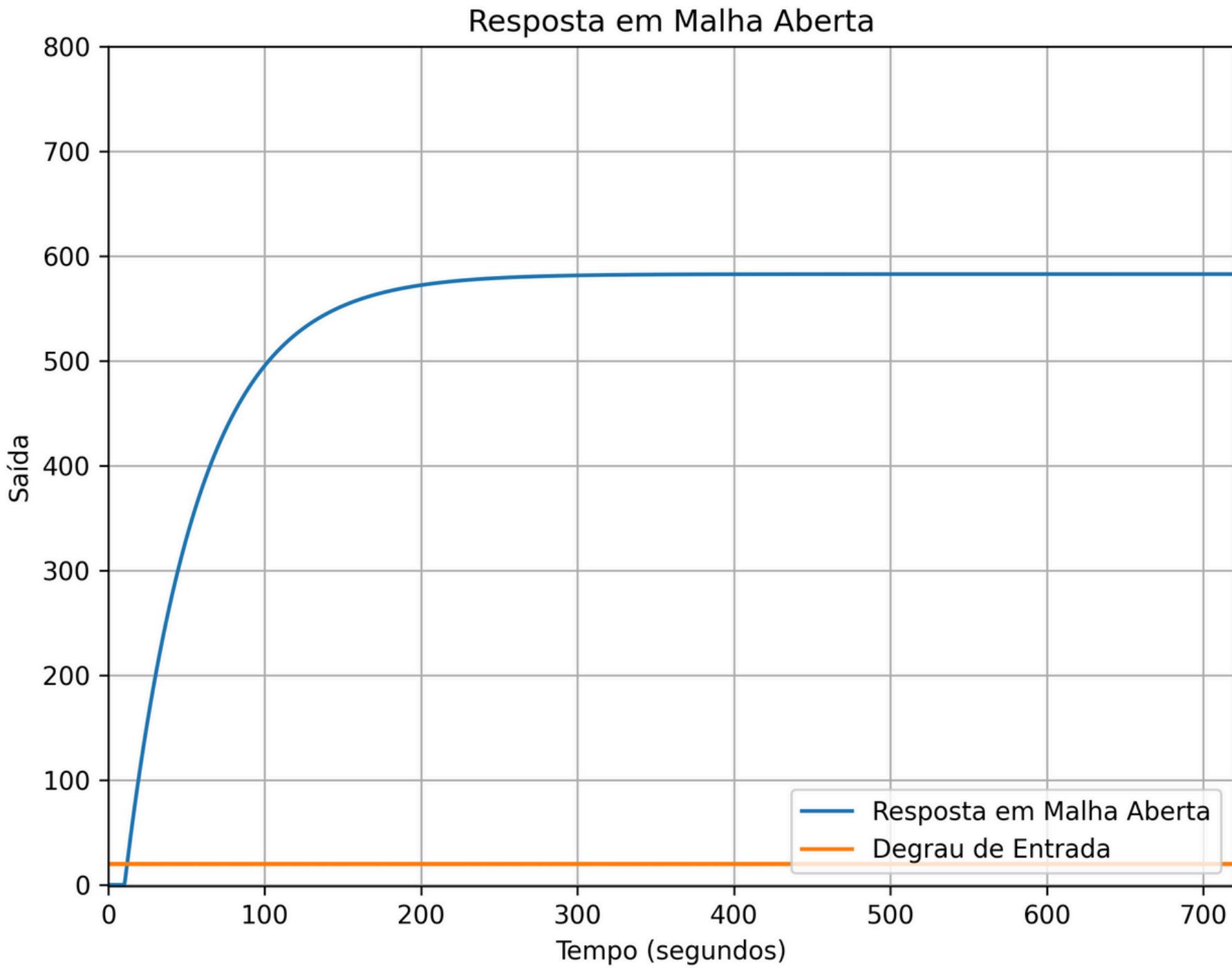
MALHA ABERTA E MALHA FECHADA

```
# Questão 4 -> Malha Aberta
num_delay, den_delay = ctl.pade(theta_estimado, 20)
malha_aberta = ctl.tf([K], [tau_estimado, 1])
malha_aberta_theta = ctl.series(ctl.tf(num_delay, den_delay), malha_aberta)
tempo_aberta, saida_aberta = ctl.step_response(malha_aberta_theta, tempo)
grafico(tempo_aberta, saida_aberta, tempo, "Resposta em Malha Aberta")

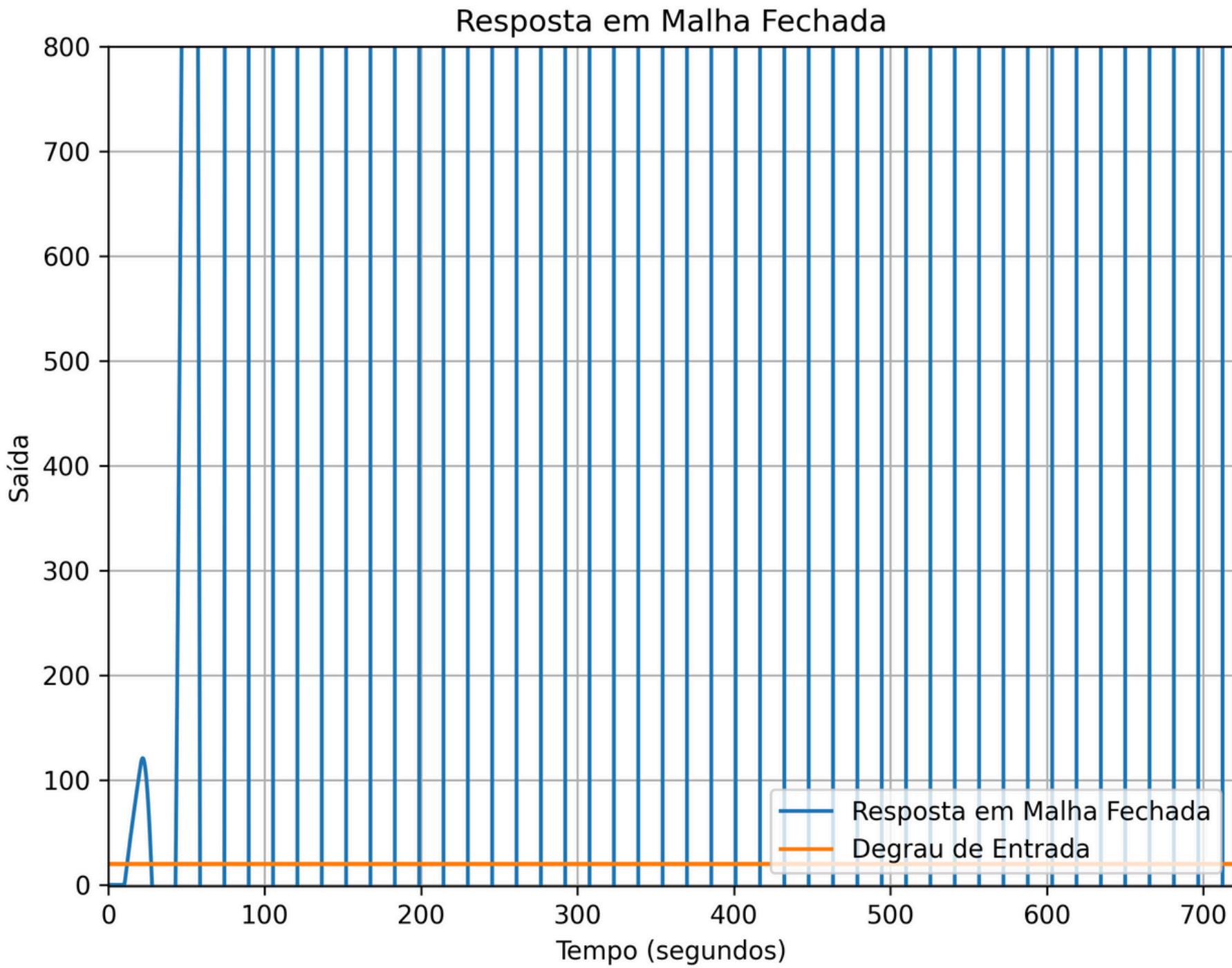
# Questão 4 -> Malha Fechada
# Create the transfer function for the closed loop
malha_fechada = ctl.feedback(malha_aberta_theta, 1)

# closed_loop_with_theta = ctl.series(ctl.tf(num_delay, den_delay), closed_loop)
# Simulate the step response
tempo_closed, saida_closed = ctl.step_response(malha_fechada, tempo)
grafico(tempo_closed, saida_closed, tempo, "Resposta em Malha Fechada")
```

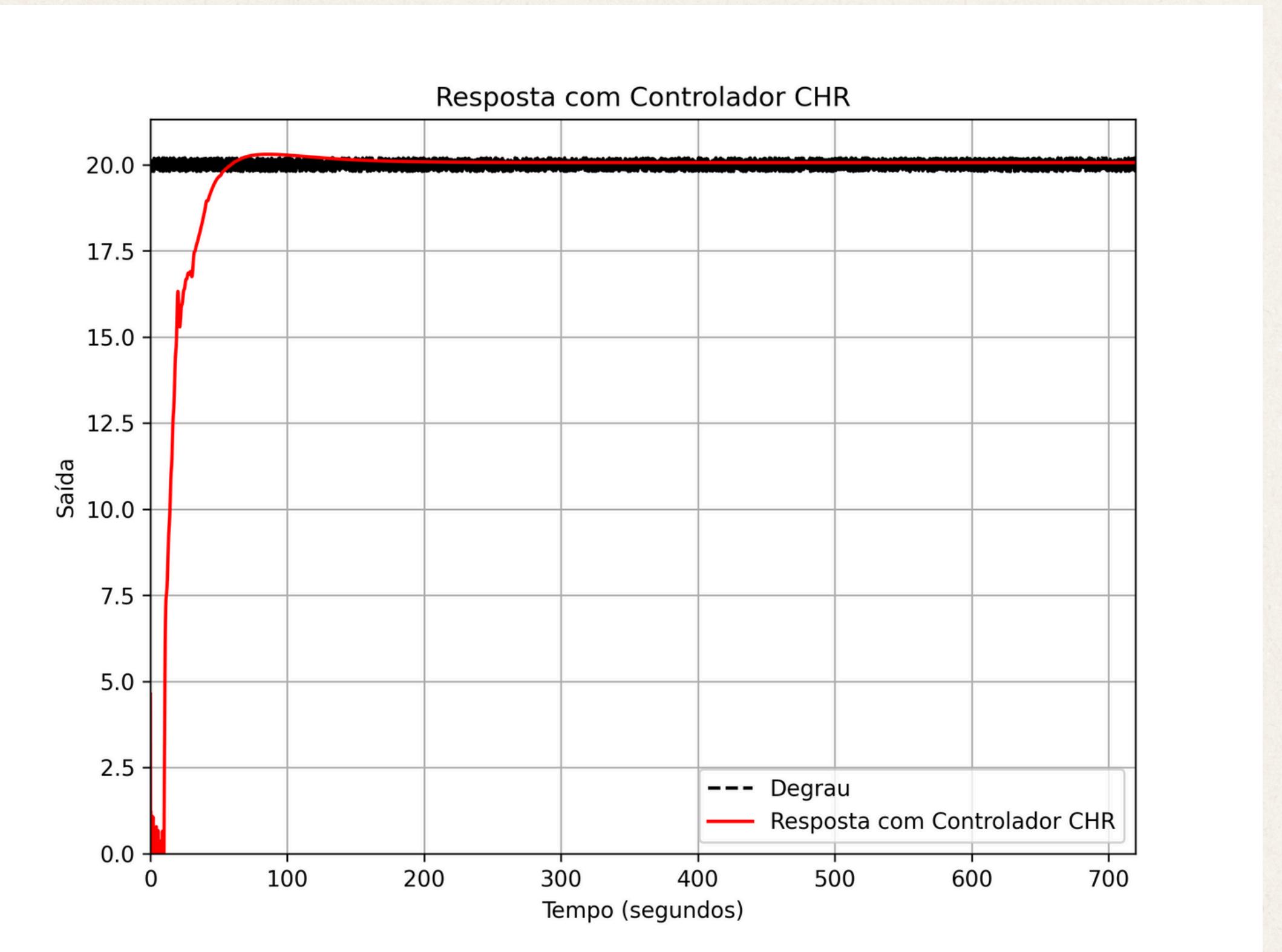
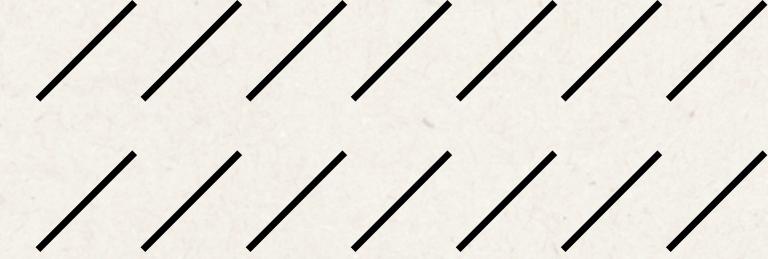
SINAL DE MALHA ABERTA



SINAL DE MALHA FECHADA

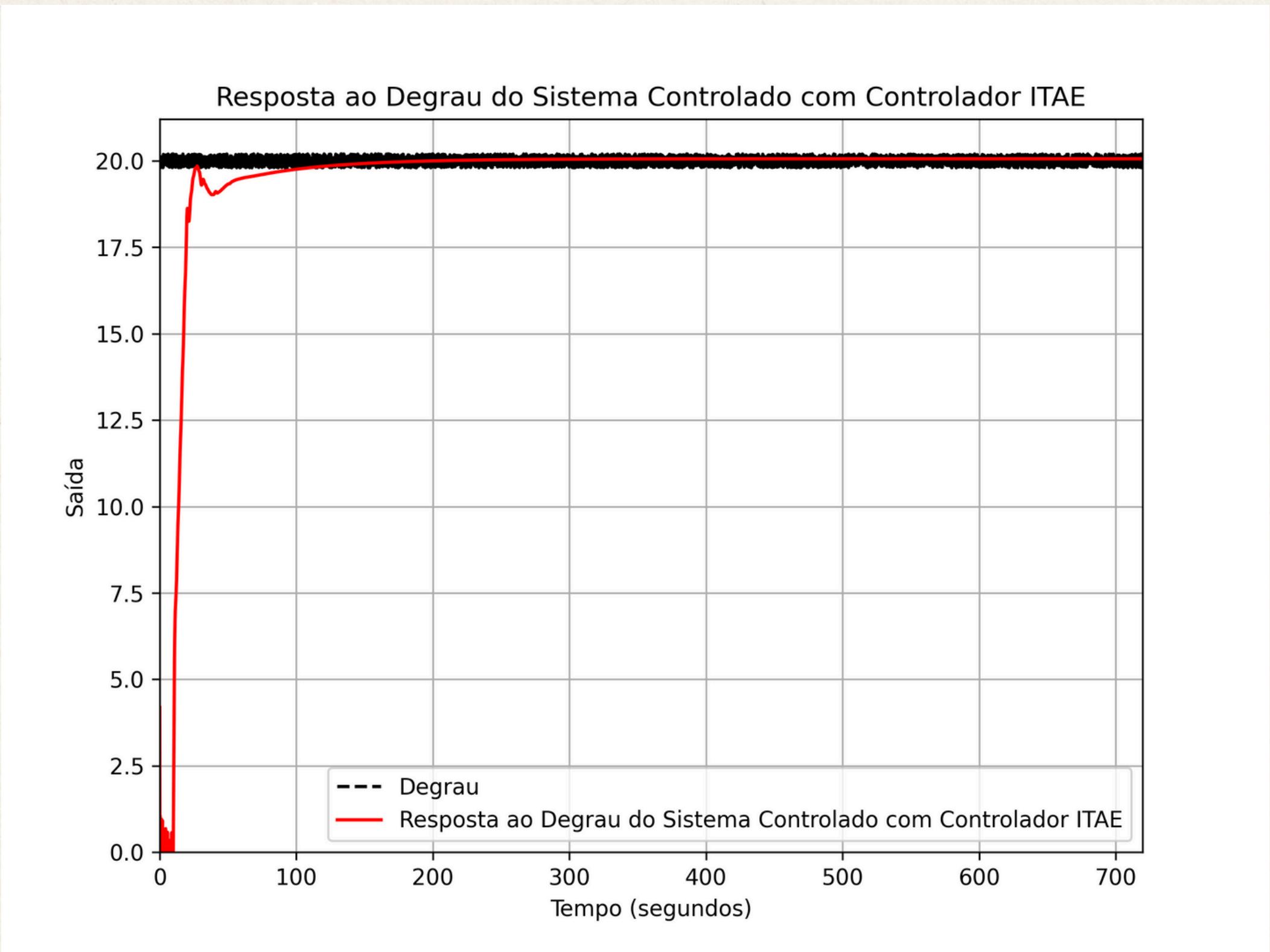


CONTROLE PID - CHR SEM SOBRESINAL



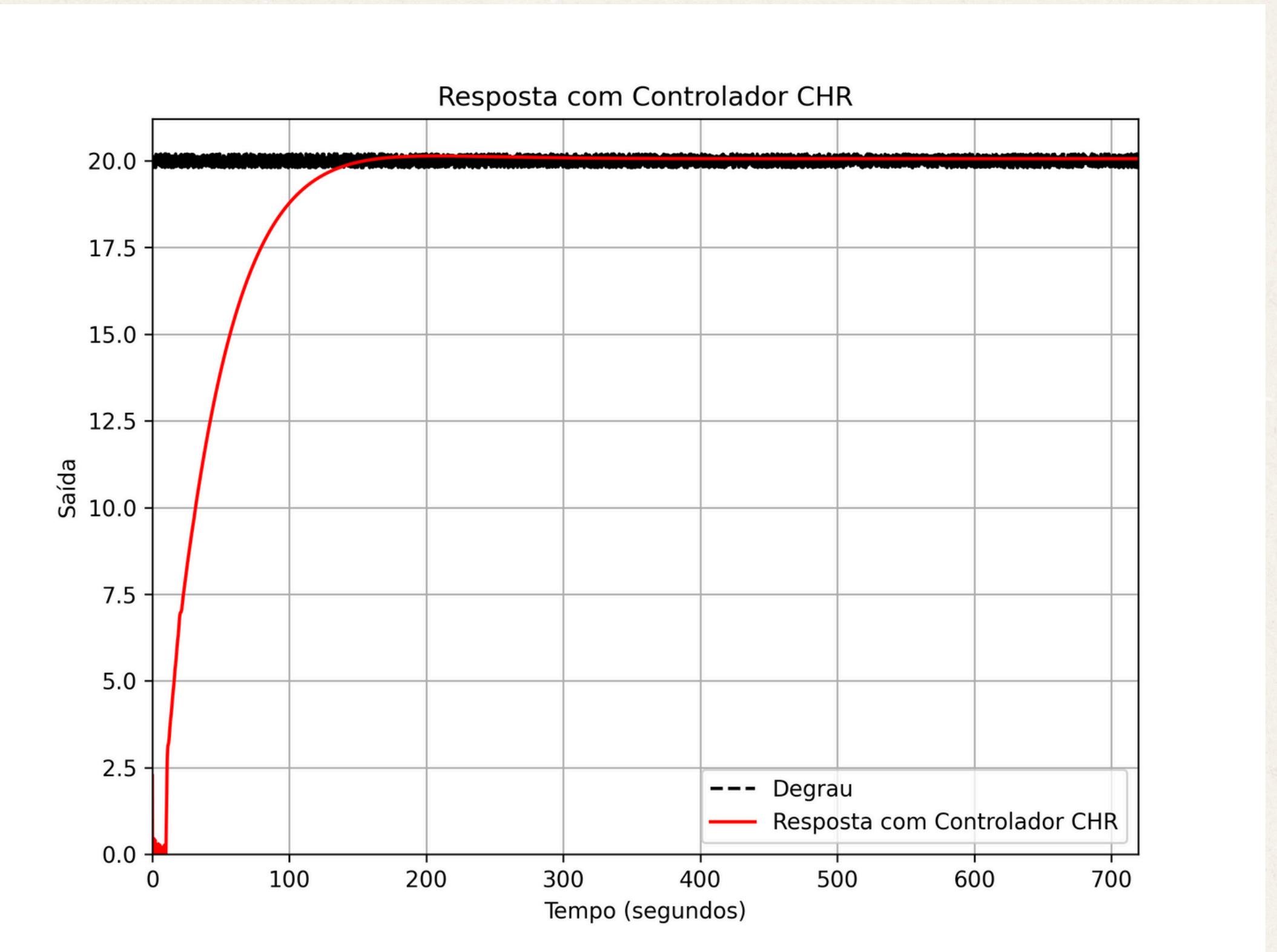
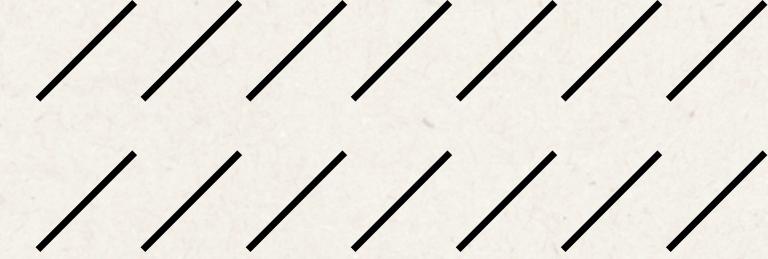
$K_p = 0,09$
 $T_i = 47,25$
 $T_d = 5,175$
Overshoot = 1,2

CONTROLE PID - ITAE



$K_p = 0,12$
 $T_i = 61,86$
 $T_d = 3,55$
Overshoot = 0,0

AJUSTE DO SINAL CHR



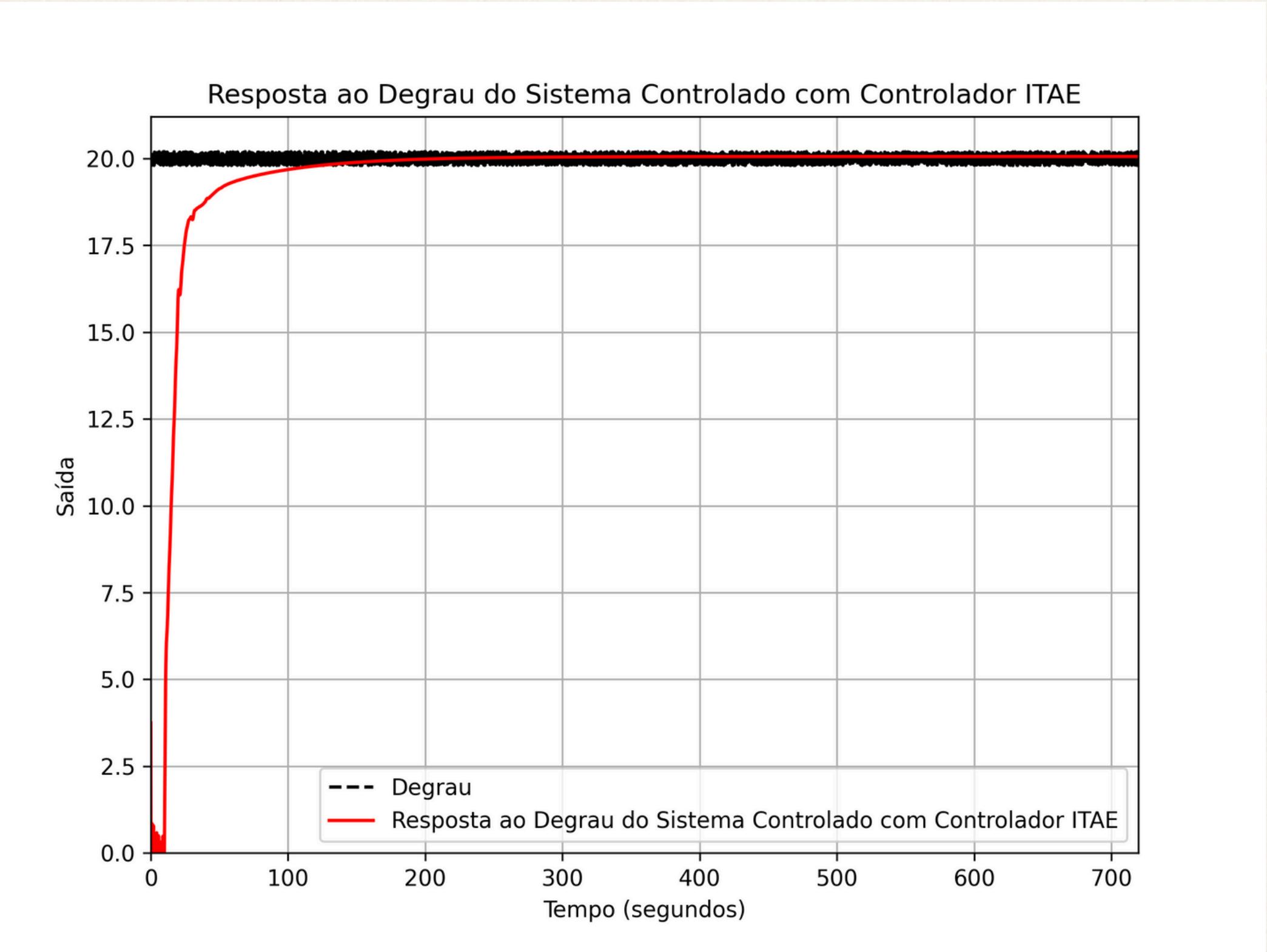
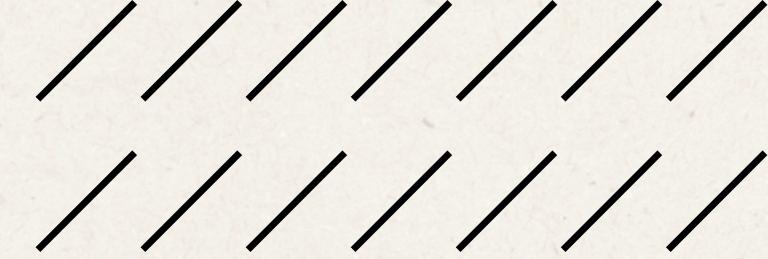
$$K_p = 0,04$$

$$T_i = 47,25$$

$$T_d = 5,175$$

$$\text{Overshoot} = 0,39$$

AJUSTE DO SINAL ITAE

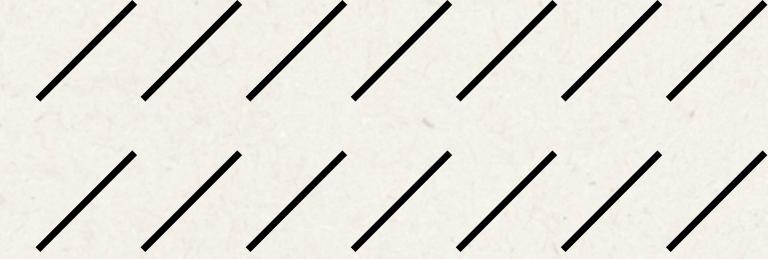


$$K_p = 0,105$$

$$T_i = 61,86$$

$$T_d = 3,55$$

$$\text{Overshoot} = 0,0$$



OBRIGADO