
Table of Contents

.....	1
PROJETO 01- IDENTIFICACAO DE SISTEMAS	1
EXTRAÇÃO DE DADOS	1
CODIGO REFERENTE AO PROJETO 1- IDENTIFICACAO DE SISTEMAS	4
PLOTANDO OS GRAFICOS DAS RESPOSTAS DAS MALHAS AO AQUECIMENTO	5
PLOTANDO OS GRAFICOS DAS RESPOSTAS DAS MALHAS AO RESFRIAMENTO	9
CALCULO DOS ERROS MEDIO QUADRATICOS PARA O CASO DO AQUECIMENTO	14
CALCULO DOS ERROS MEDIO QUADRATICOS PARA O CASO DO RESFRIAMENTO	14
MODELOS MÉDIOS	15
PROJETO 2 - CONTROLADOR PID	18
MÉTODO SIMC PARA G11 e G22	18
EXTRAÇÃO DE DADOS	19
CODIGO REFERENTE AO PROJETO 2- IDENTIFICACAO DE SISTEMAS	22
AQUECIMENTO	22
IAE (INTEGRAL ERRO ABSOLUTO) - AQUECIMENTO	29
Resfriamento	30
IAE (INTEGRAL ERRO ABSOLUTO) - RESFRIAMENTO	37

```
clear all
clc
```

PROJETO 01- IDENTIFICACAO DE SISTEMAS

Nome: Edilberto Elias Xavier Junior Matricula: 120210134 Turma: 02

EXTRAÇÃO DE DADOS

```
dados_coletados = load('dados_20250313T091841.mat');

figure(1)
subplot(1,2,1)
plot(dados_coletados.mv1) %comportamento geral
title('Comportamento Geral- MALHA 1')
grid on

subplot(1,2,2)
plot(dados_coletados.mv2) %comportamento geral
title('Comportamento Geral- MALHA 2')
grid on

figure(2)
subplot(1,2,1)
plot(dados_coletados.pv1) %comportamento geral
title('Comportamento Geral- Sistema - Malha 1')
grid on

subplot(1,2,2)
```

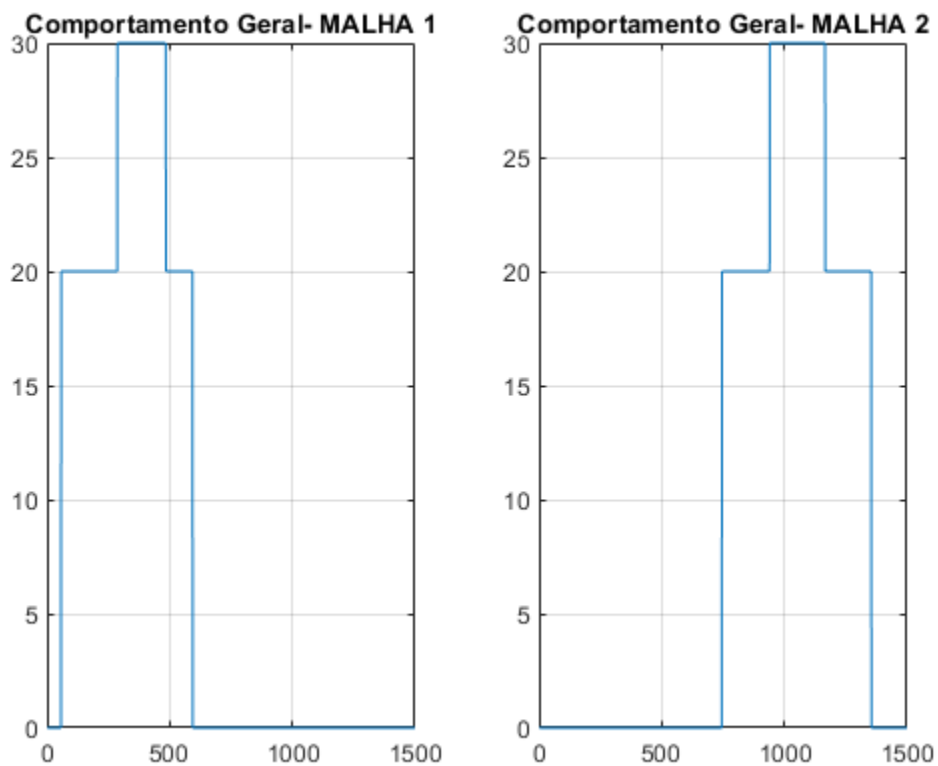
```
plot(dados_coletados.pv2) %comportamento geral
title('Comportamento Geral- Sistema - Malha 2')
grid on
```

```
figure(3)
subplot(1,2,1)
plot(dados_coletados.mv1(284:485)) %aquecimento
title('Comportamento do MV1 no Aquecimento')
grid on
```

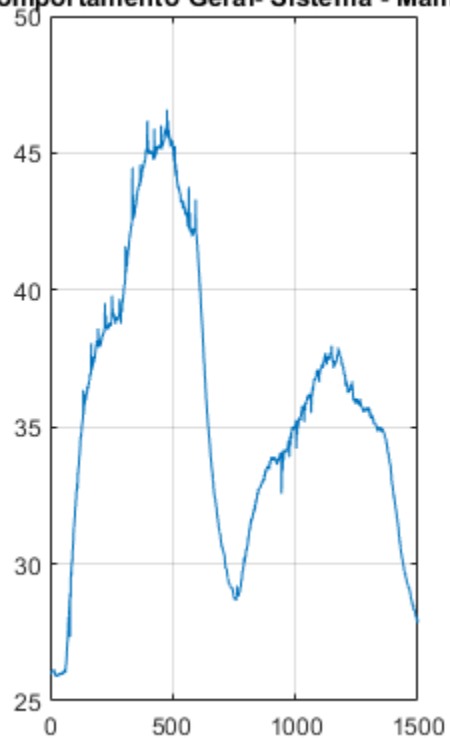
```
subplot(1,2,2)
plot(dados_coletados.pv1(284:485))%aquecimento
title('Comportamento do PV1 no Aquecimento')
grid on
```

```
figure(4)
subplot(1,2,1)
plot(dados_coletados.mv1(485:592)) %resfriamento
title('Comportamento do MV1 no Resfriamento')
grid on
```

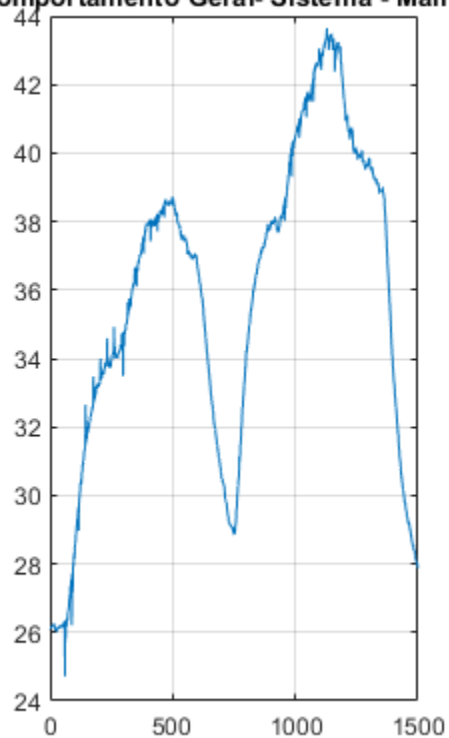
```
subplot(1,2,2)
plot(dados_coletados.pv1(485:592))%resfriamento
title('Comportamento do PV1 no Resfriamento')
grid on
```



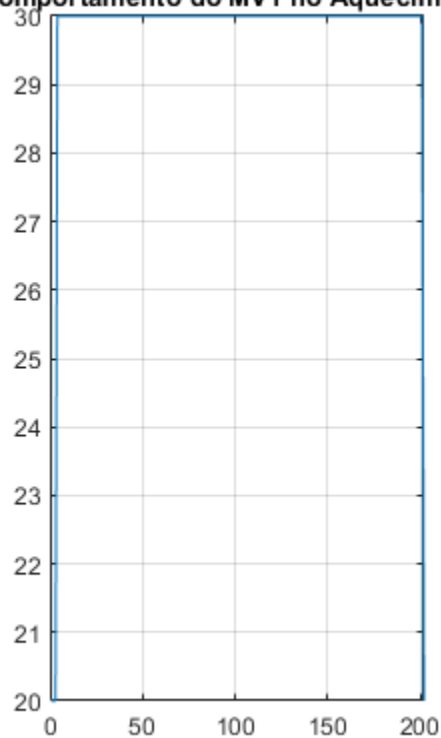
Comportamento Geral- Sistema - Malha



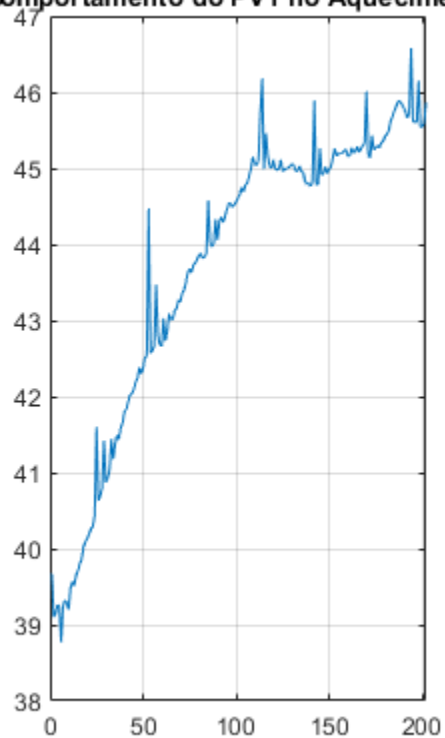
Comportamento Geral- Sistema - Malha 2

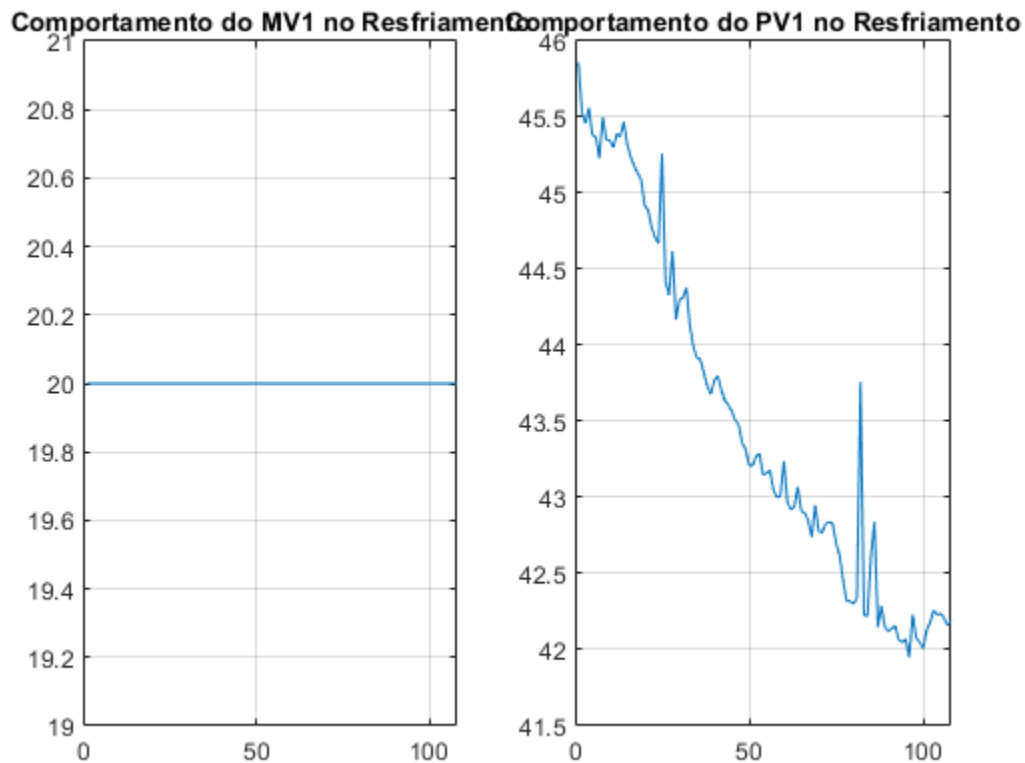


Comportamento do MV1 no Aquecimento



Comportamento do PV1 no Aquecimento





CODIGO REFERENTE AO PROJETO 1- IDENTIFICACAO DE SISTEMAS

```
DeltaT= 2;  
h_subida= 10;  
h_descida= -10;  
  
%intervalos do sistema  
pv_subida_11 = dados_coletados.pv1(284:485);  
pv_descida_11 = dados_coletados.pv1(485:592);  
  
pv_subida_22 = dados_coletados.pv2(940:1167);  
pv_descida_22 = dados_coletados.pv2(1167:1354);  
  
pv_subida_21 = dados_coletados.pv1(940:1167);  
pv_descida_21 = dados_coletados.pv1(1167:1354);  
  
pv_subida_12 = dados_coletados.pv2(284:485);  
pv_descida_12 = dados_coletados.pv2(485:592);
```

PLOTANDO OS GRAFICOS DAS RESPOSTAS DAS MALHAS AO AQUECIMENTO

```
%Para 11

y = pv_subida_11 - pv_subida_11(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_subida * ones(size(y)); %criando uma entrada degrau

[G011sub, T11sub, L11sub] = parametrosFOPTD(y, h_subida, DeltaT); %obtendo
os parametros
G11subida = tf(G011sub, [T11sub, 1], 'iodelay', L11sub)
G11_SIMULADO_subida = lsim(G11subida, u, t);

figure(5)
plot(t, y, t, G11_SIMULADO_subida);
title('Resposta do Sistema ao Aquecimento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G11 Simulado');
grid on

%Para 12

y = pv_subida_12 - pv_subida_12(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_subida * ones(size(y)); %criando uma entrada degrau

[G012sub, T12sub, L12sub] = parametrosFOPTD(y, h_subida, DeltaT); %obtendo
os parametros
G12subida = tf(G012sub, [T12sub, 1], 'iodelay', L12sub)
G12_SIMULADO_subida = lsim(G12subida, u, t);

figure(6)
plot(t, y, t, G12_SIMULADO_subida);
title('Resposta do Sistema ao Aquecimento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G12 Simulado');
grid on

%Para 21

y = pv_subida_21 - pv_subida_21(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_subida * ones(size(y)); %criando uma entrada degrau

[G021sub, T21sub, L21sub] = parametrosFOPTD(y, h_subida, DeltaT); %obtendo
os parametros
G21subida = tf(G021sub, [T21sub, 1], 'iodelay', L21sub) %criando a funcao de
transferencia
G21_SIMULADO_subida = lsim(G21subida, u, t);

figure(7)
```

```

plot(t, y, t, G21_SIMULADO_subida);
title('Resposta do Sistema ao Aquecimento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G21 Simulado');
grid on

%Para 22
y = pv_subida_22 - pv_subida_22(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_subida * ones(size(y)); %criando uma entrada degrau

[G022sub, T22sub, L22sub] = parametrosFOPTD(y, h_subida, DeltaT); %obtendo
os parametros
G22subida = tf(G022sub, [T22sub, 1], 'iodelay', L22sub) %criando a funcao de
transferencia
G22_SIMULADO_subida = lsim(G22subida, u, t);

figure(8)
plot(t, y, t, G22_SIMULADO_subida);
title('Resposta do Sistema ao Aquecimento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G22 Simulado');
grid on

G11subida =

          0.6044
exp(-31.6*s) * -----
          106.7 s + 1

Continuous-time transfer function.

G12subida =

          0.4567
exp(-24.7*s) * -----
          146.7 s + 1

Continuous-time transfer function.

G21subida =

          0.3391
exp(-55.7*s) * -----
          139.6 s + 1

Continuous-time transfer function.

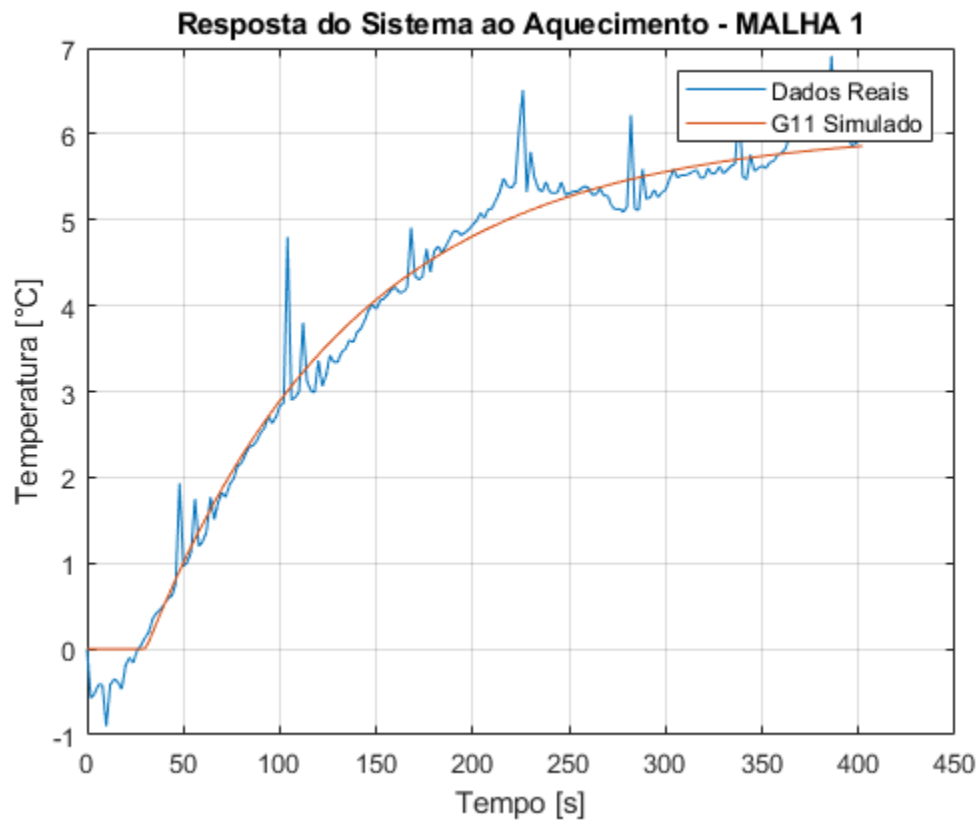
G22subida =

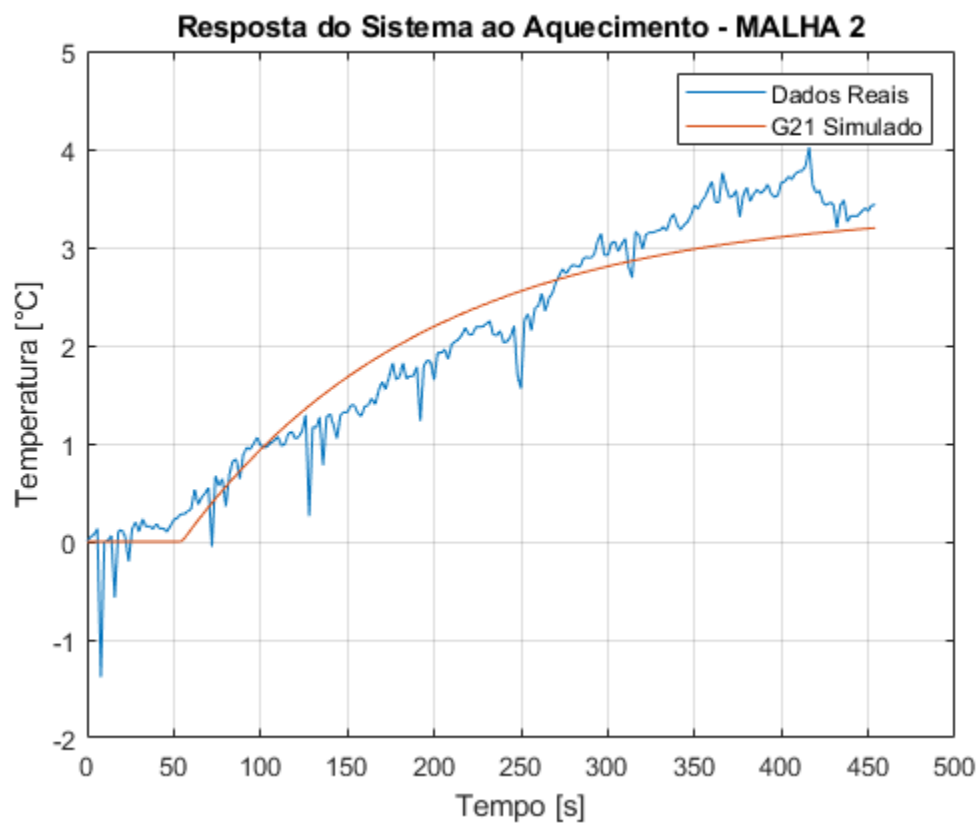
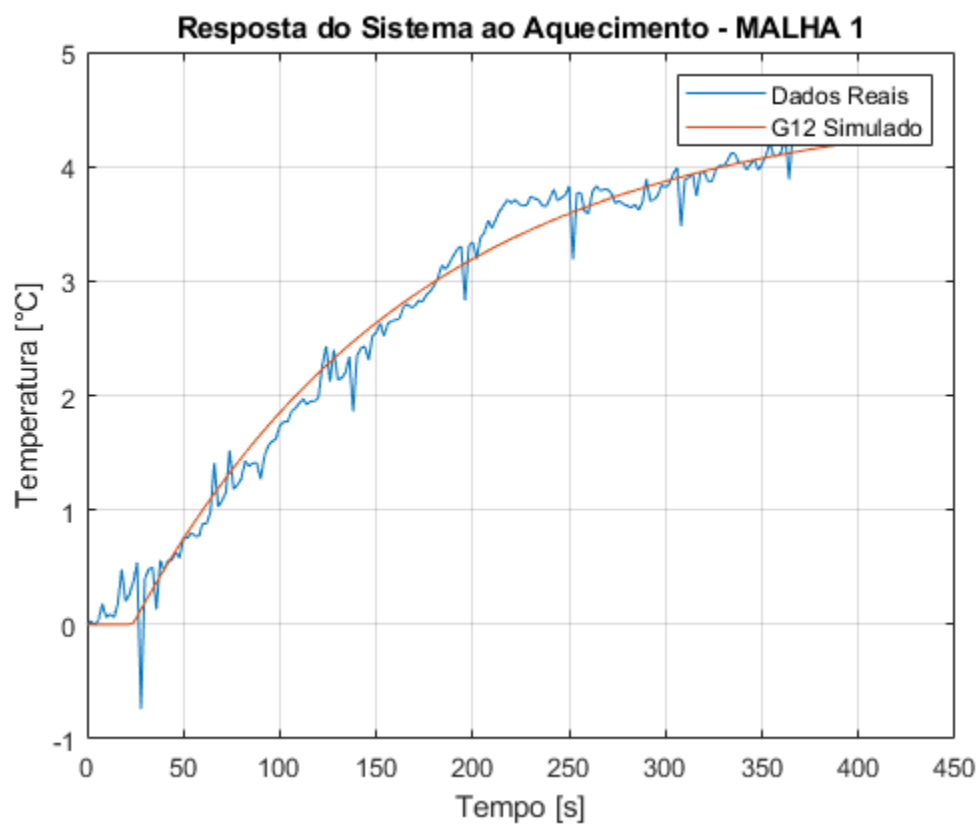
          0.5629

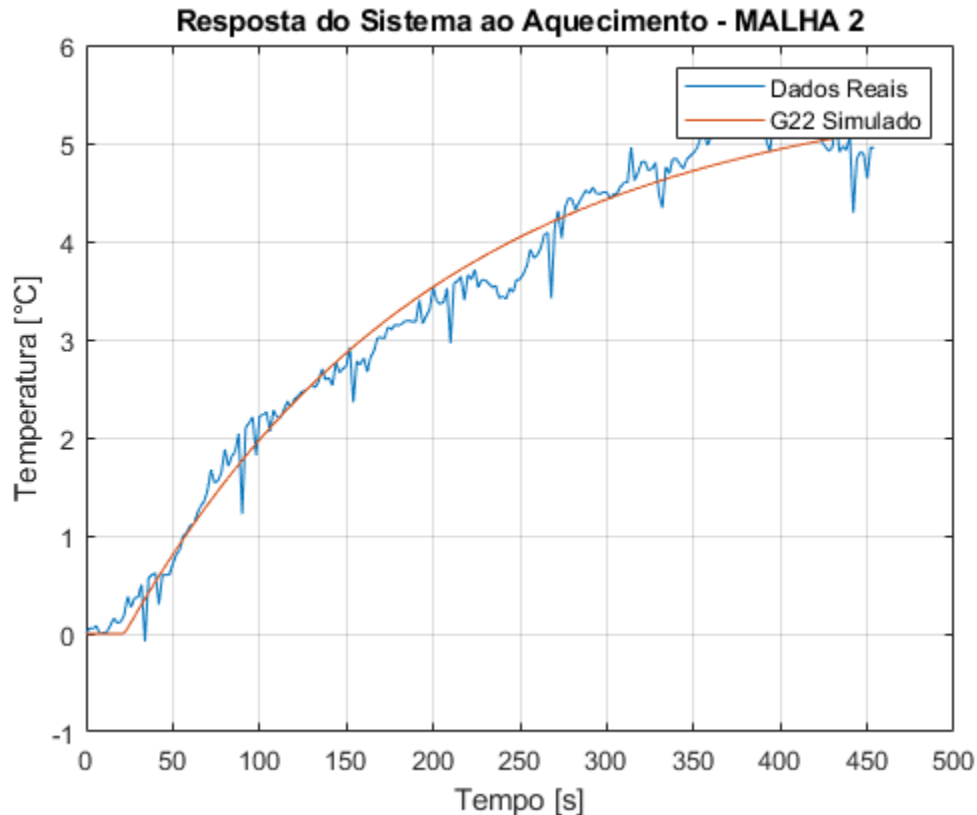
```

$$\exp(-23.2*s) * \frac{\text{-----}}{179.8 s + 1}$$

Continuous-time transfer function.







PLOTANDO OS GRAFICOS DAS RESPOSTAS DAS MALHAS AO RESFRIAMENTO

```
%Para 11
y = pv_descida_11 - pv_descida_11(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_descida * ones(size(y)); %criando uma entrada degrau

[G011desc, T11desc, L11desc] = parametrosFOPTD(y, h_descida, DeltaT);
%obtendo os parametros
G11descida = tf(G011desc, [T11desc, 1], 'iodelay', L11desc) %criando a
funcao de transferencia
G11_SIMULADO_descida = lsim(G11descida, u, t);

figure(9)
plot(t, y, t, G11_SIMULADO_descida);
title('Resposta do Sistema ao Resfriamento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G11 Simulado');
grid on

%Para 12
y = pv_descida_12 - pv_descida_12(1); %retirando o valor inicial
```

```

t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_descida * ones(size(y)); %criando uma entrada degrau

[G012desc, T12desc, L12desc] = parametrosFOPTD(y, h_descida, DeltaT);
%obtendo os parametros
G12descida = tf(G012desc, [T12desc, 1], 'iodelay', L12desc) %criando a
funcao de transferencia
G12_SIMULADO_descida = lsim(G12descida, u, t);

figure(10)
plot(t, y, t, G12_SIMULADO_descida);
title('Resposta do Sistema ao Resfriamento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G12 Simulado');
grid on

%Para 21
y = pv_descida_21 - pv_descida_21(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_descida * ones(size(y)); %criando uma entrada degrau

[G021desc, T21desc, L21desc] = parametrosFOPTD(y, h_descida, DeltaT);
%obtendo os parametros
G21descida = tf(G021desc, [T21desc, 1], 'iodelay', L21desc) %criando a
funcao de transferencia
G21_SIMULADO_descida = lsim(G21descida, u, t);

figure(11)
plot(t, y, t, G21_SIMULADO_descida);
title('Resposta do Sistema ao Resfriamento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G21 Simulado');
grid on

%Para 22
y = pv_descida_22 - pv_descida_22(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
u = h_descida * ones(size(y)); %criando uma entrada degrau

[G022desc, T22desc, L22desc] = parametrosFOPTD(y, h_descida, DeltaT);
%obtendo os parametros
G22descida = tf(G022desc, [T22desc, 1], 'iodelay', L22desc) %criando a
funcao de transferencia
G22_SIMULADO_descida = lsim(G22descida, u, t);

figure(12)
plot(t, y, t, G22_SIMULADO_descida);
title('Resposta do Sistema ao Resfriamento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G22 Simulado');
grid on

```

G11descida =

$$\exp(-14*s) * \frac{0.3796}{73.21 s + 1}$$

Continuous-time transfer function.

G12descida =

$$\exp(-33.1*s) * \frac{0.1441}{56.28 s + 1}$$

Continuous-time transfer function.

G21descida =

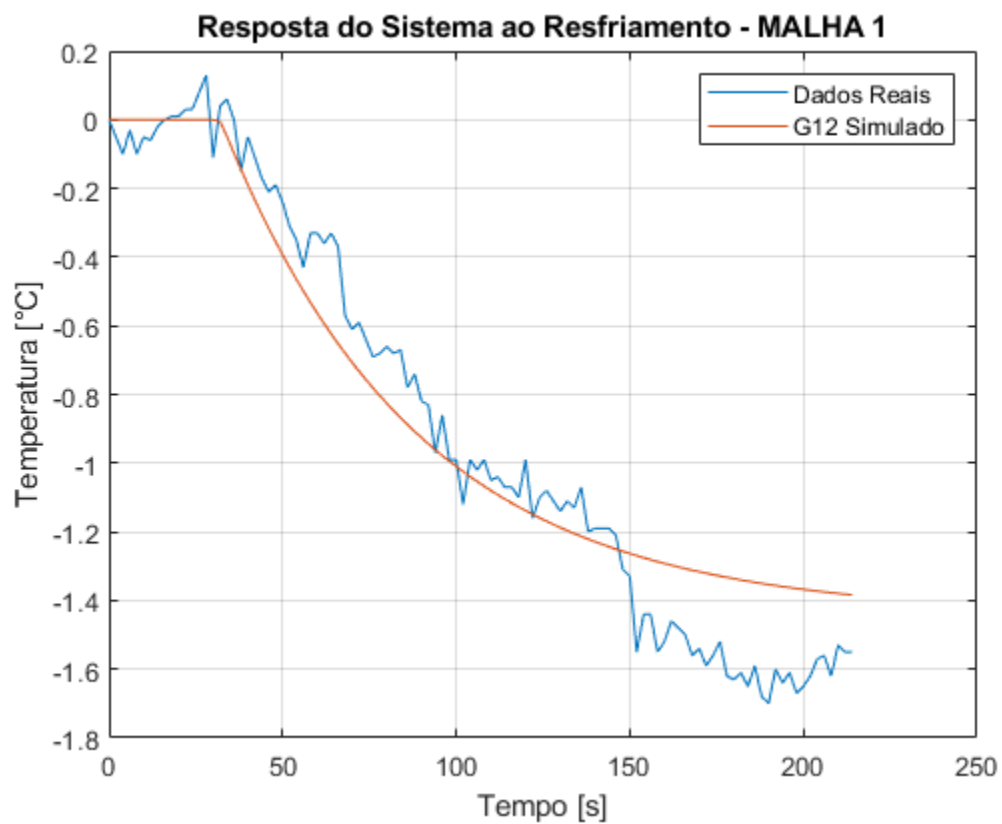
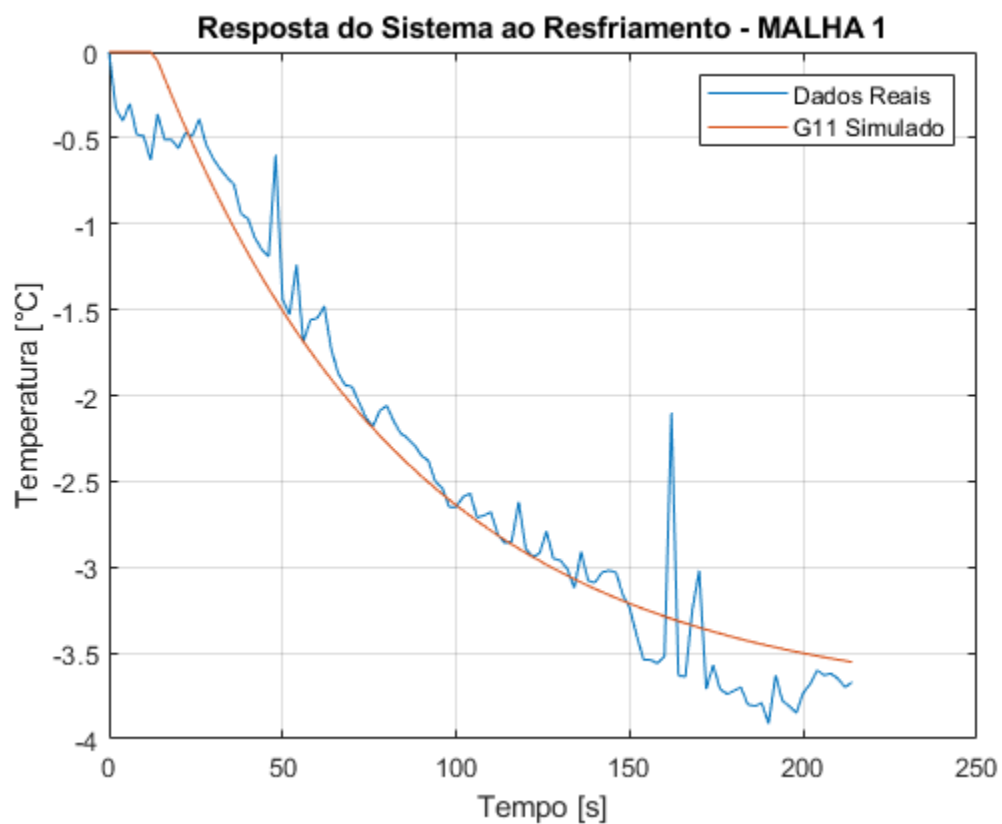
$$\exp(-45.7*s) * \frac{0.2322}{122.2 s + 1}$$

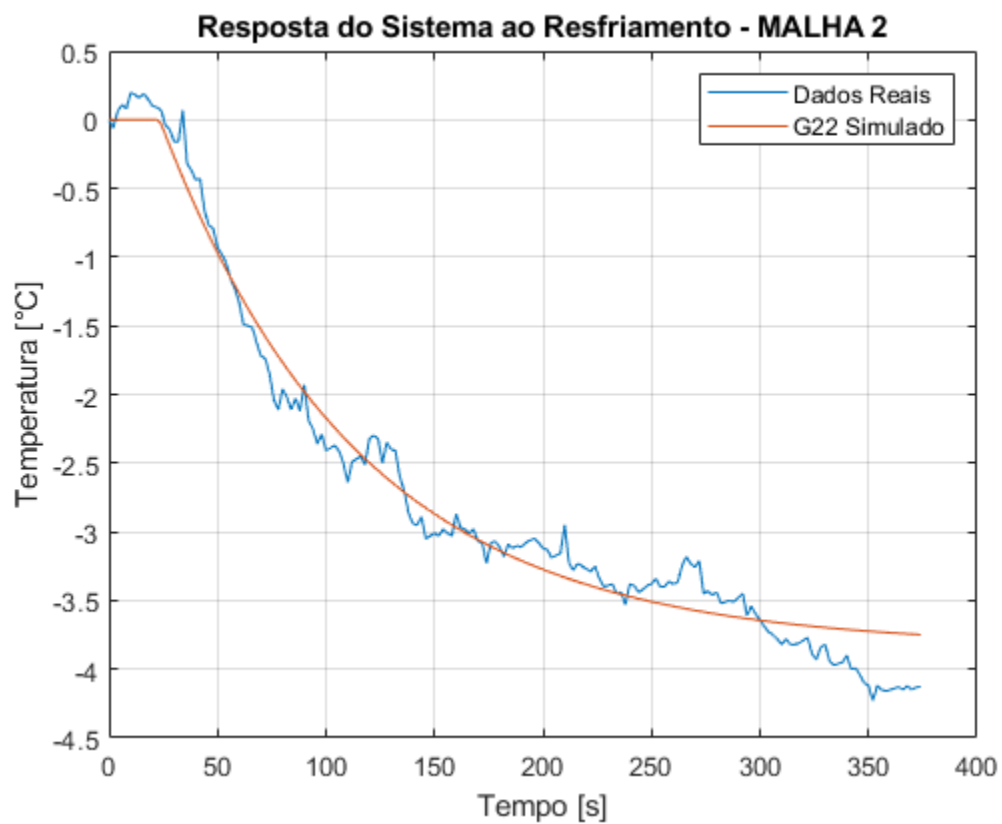
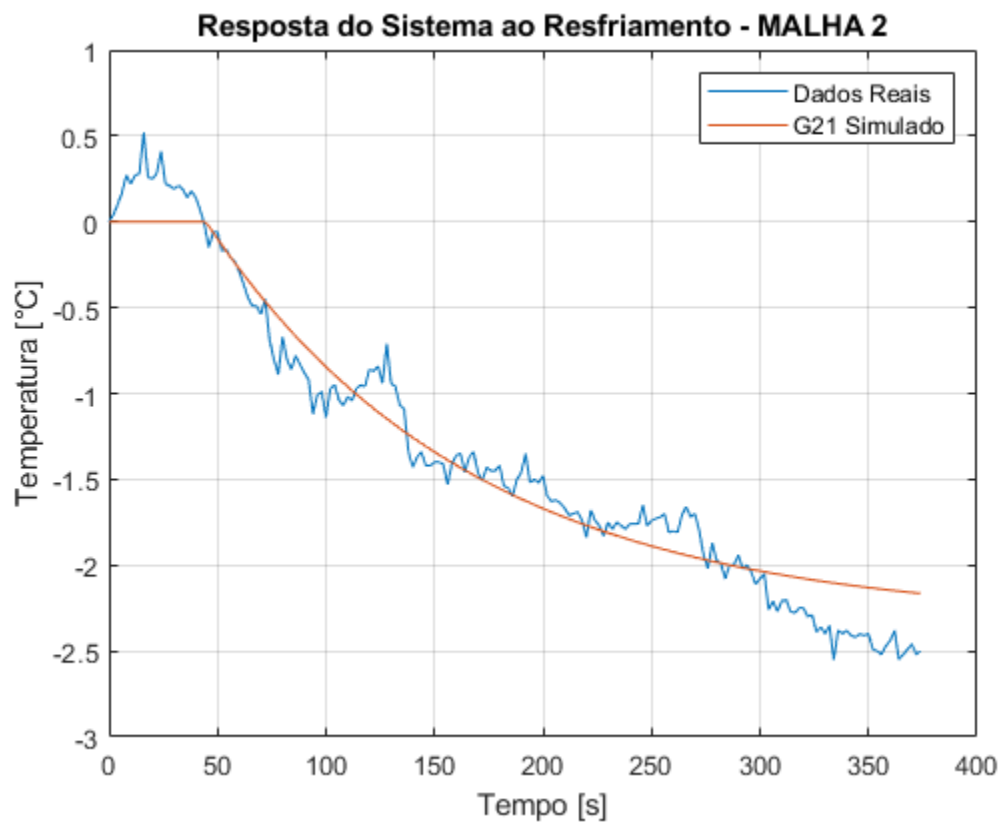
Continuous-time transfer function.

G22descida =

$$\exp(-24.3*s) * \frac{0.3832}{91.68 s + 1}$$

Continuous-time transfer function.





CALCULO DOS ERROS MEDIO QUADRATICOS PARA O CASO DO AQUECIMENTO

```
disp(['PARA O CASO DE AQUECIMENTO']);

%PARA G11
real_11subida = pv_subida_11 - pv_subida_11(1);
emq_11subida = sqrt(mean((G11_SIMULADO_subida(:) - real_11subida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G11 é = ',
num2str(emq_11subida)]);

%PARA G12
real_12subida = pv_subida_12 - pv_subida_12(1);
emq_12subida = sqrt(mean((G12_SIMULADO_subida(:) - real_12subida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G12 é = ',
num2str(emq_12subida)]);

%PARA G21
real_21subida = pv_subida_21 - pv_subida_21(1);
emq_21subida = sqrt(mean((G21_SIMULADO_subida(:) - real_21subida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G21 é = ',
num2str(emq_21subida)]);

%PARA G22
real_22subida = pv_subida_22 - pv_subida_22(1);
emq_22subida = sqrt(mean((G22_SIMULADO_subida(:) - real_22subida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G22 é = ',
num2str(emq_22subida)]);

PARA O CASO DE AQUECIMENTO
O Erro Médio Quadrático para o caso G11 é = 0.31209
O Erro Médio Quadrático para o caso G12 é = 0.1807
O Erro Médio Quadrático para o caso G21 é = 0.35387
O Erro Médio Quadrático para o caso G22 é = 0.25051
```

CALCULO DOS ERROS MEDIO QUADRATICOS PARA O CASO DO RESFRIAMENTO

```
disp(['PARA O CASO DE RESFRIAMENTO']);

%PARA G11
real_11descida = pv_descida_11 - pv_descida_11(1);
emq_11descida = sqrt(mean((G11_SIMULADO_descida(:) - real_11descida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G11 é = ',
num2str(emq_11descida)]);

%PARA G12
real_12descida = pv_descida_12 - pv_descida_12(1);
emq_12descida = sqrt(mean((G12_SIMULADO_descida(:) - real_12descida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G12 é = ',
```

```

num2str(emq_12descida)]);

%PARA G21
real_21descida = pv_descida_21 - pv_descida_21(1);
emq_21descida = sqrt(mean((G21_SIMULADO_descida(:) - real_21descida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G21 é= ',
num2str(emq_21descida)]);

%PARA G22
real_22descida = pv_descida_22 - pv_descida_22(1);
emq_22descida = sqrt(mean((G22_SIMULADO_descida(:) - real_22descida(:)).^2));
disp(['O Erro Médio Quadrático para o caso G22 é= ',
num2str(emq_22descida)]);

PARA O CASO DE RESFRIAMENTO
O Erro Médio Quadrático para o caso G11 é= 0.26053
O Erro Médio Quadrático para o caso G12 é= 0.16121
O Erro Médio Quadrático para o caso G21 é= 0.18659
O Erro Médio Quadrático para o caso G22 é= 0.19414

```

MODELOS MÉDIOS

```

%G11
K11 = (G011sub+G011desc)/2
T11 = (T11sub+T11desc)/2
L11 = (L11sub+L11desc)/2

num_aprox = K11;
den_aprox = [T11 1];

G11 = tf(num_aprox, den_aprox, 'InputDelay', L11);
disp('Sistema Aproximado G11(s):');
display(G11);

%G12
K12 = (G012sub+G012desc)/2
T12 = (T12sub+T12desc)/2
L12 = (L12sub+L12desc)/2

num_aprox = K12;
den_aprox = [T12 1];

G12 = tf(num_aprox, den_aprox, 'InputDelay', L12);
disp('Sistema Aproximado G12(s):');
display(G12);

%G21
K21 = (G021sub+G021desc)/2
T21 = (T21sub+T21desc)/2
L21 = (L21sub+L21desc)/2

num_aprox = K21;
den_aprox = [T21 1];

```

```
G21 = tf(num_aprox, den_aprox, 'InputDelay', L21);
disp('Sistema Aproximado G21(s):');
display(G21);
```

```
%G22
K22 = (G022sub+G022desc)/2
T22 = (T22sub+T22desc)/2
L22 = (L22sub+L22desc)/2
```

```
num_aprox = K22;
den_aprox = [T22 1];
```

```
G22 = tf(num_aprox, den_aprox, 'InputDelay', L22);
disp('Sistema Aproximado G22(s):');
display(G22);
```

```
K11 =

    0.4920
```

```
T11 =

    89.9352
```

```
L11 =

    22.7880
```

Sistema Aproximado G11(s):

```
G11 =

    0.492
exp(-22.8*s) * -----
    89.94 s + 1
```

Continuous-time transfer function.

```
K12 =

    0.3004
```

```
T12 =

    101.5112
```

```
L12 =
```

28.9002

Sistema Aproximado $G_{12}(s)$:

$G_{12} =$

$$\exp(-28.9*s) * \frac{0.3004}{101.5 s + 1}$$

Continuous-time transfer function.

$K_{21} =$

0.2857

$T_{21} =$

130.9423

$L_{21} =$

50.6993

Sistema Aproximado $G_{21}(s)$:

$G_{21} =$

$$\exp(-50.7*s) * \frac{0.2857}{130.9 s + 1}$$

Continuous-time transfer function.

$K_{22} =$

0.4731

$T_{22} =$

135.7595

$L_{22} =$

23.7477

Sistema Aproximado $G_{22}(s)$:

$G_{22} =$

$$\exp(-23.7*s) * \frac{0.4731}{135.8 s + 1}$$

Continuous-time transfer function.

PROJETO 2 - CONTROLADOR PID

Nome: Edilberto Elias Xavier Junior Matricula: 120210134 Turma: 02

MÉTODO SIMC PARA G11 e G22

Metodo SIMC G11

```
t_1= T11;
theta= L11;
t_c = theta;

L = L11;

Kp_G11= (1/K11)*(t_1/(t_c + theta));
Ti_G11= min(t_1, 4*(t_c + theta));
Td_G11= 0;

Kp = Kp_G11;
Ti = 1/Ti_G11;
Td = Td_G11;

disp('Para G11, o método de SIMC:')
disp(['Controlador PI, ',char(964), 'c = ', char(952), ':'])
disp(['O ganho Kp = ' num2str(Kp)]);
disp(['O valor Ti = ' num2str(Ti_G11)]);
disp(['O ganho Ki = ' num2str(Kp*Ti)]);

%Metodo SIMC G22
t_1= T22;
theta= L22;
t_c = theta;

L = L22;

Kp_G22= (1/K22)*(t_1/(t_c + theta));
Ti_G22= min(t_1, 4*(t_c + theta));
Td_G22= 0;

Kp = Kp_G22;
Ti = 1/Ti_G22;
Td = Td_G22;

disp('Para G22, o método de SIMC:')
disp(['Controlador PI, ',char(964), 'c = ', char(952), ':'])
disp(['O ganho Kp = ' num2str(Kp)]);
```

```
disp(['O valor Ti = ' num2str(Ti_G22)]); disp(['O  
ganho Ki = ' num2str(Kp*Ti)]);
```

```
Para G11, o método de SIMC:  
Controlador PI,  $\tau_c = \theta$ :  
O ganho  $K_p = 4.0108$   
O valor  $T_i = 89.9352$   
O ganho  $K_i = 0.044596$   
Para G22, o método de SIMC:  
Controlador PI,  $\tau_c = \theta$ :  
O ganho  $K_p = 6.0422$   
O valor  $T_i = 135.7595$   
O ganho  $K_i = 0.044506$ 
```

EXTRAÇÃO DE DADOS

```
dados_coletados = load('dados_20250403T092605.mat');
```

```
figure(13)  
subplot(1,2,1)  
plot(dados_coletados.mv1) %comportamento geral  
title('Comportamento Geral- MALHA 1')  
grid on
```

```
subplot(1,2,2)  
plot(dados_coletados.mv2) %comportamento geral  
title('Comportamento Geral- MALHA 2')  
grid on
```

```
figure(14)  
subplot(1,2,1)  
plot(dados_coletados.pv1) %comportamento geral  
title('Comportamento Geral- Sistema - Malha 1') grid  
on
```

```
subplot(1,2,2)  
plot(dados_coletados.pv2) %comportamento geral  
title('Comportamento Geral- Sistema - Malha 2') grid  
on
```

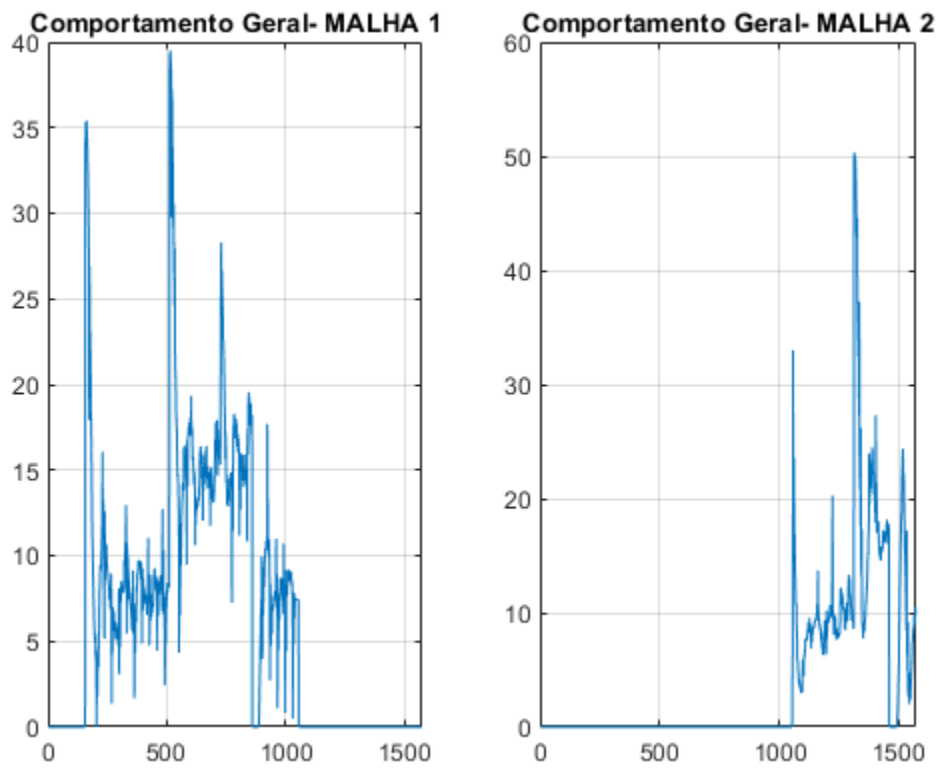
```
figure(15)  
subplot(1,2,1)  
plot(dados_coletados.mv1(532:865)) %aquecimento  
title('Comportamento do MV1 no Aquecimento') grid on
```

```
subplot(1,2,2)  
plot(dados_coletados.pv1(532:865)) %aquecimento  
title('Comportamento do PV1 no Aquecimento') grid on
```

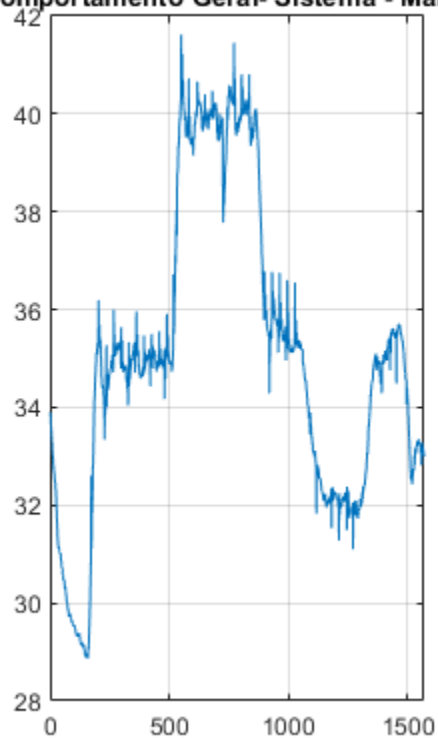
```
figure(16)  
subplot(1,2,1)
```

```
plot(dados_coletados.mv1(896:1061)) %resfriamento  
title('Comportamento do MV1 no Resfriamento')  
grid on
```

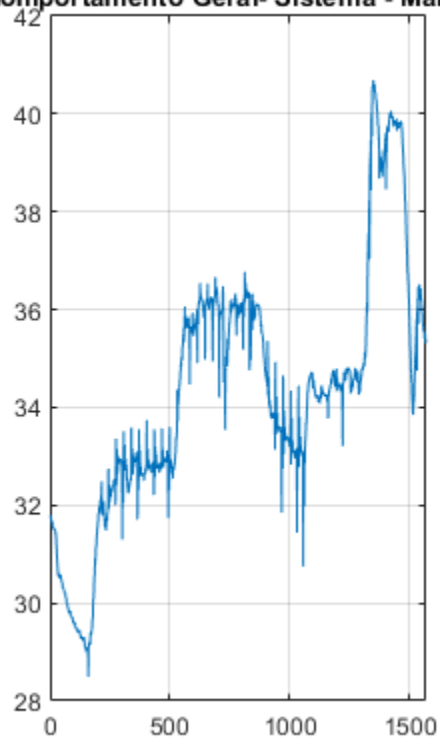
```
subplot(1,2,2)  
plot(dados_coletados.pv1(896:1061))%resfriamento  
title('Comportamento do PV1 no Resfriamento')  
grid on
```



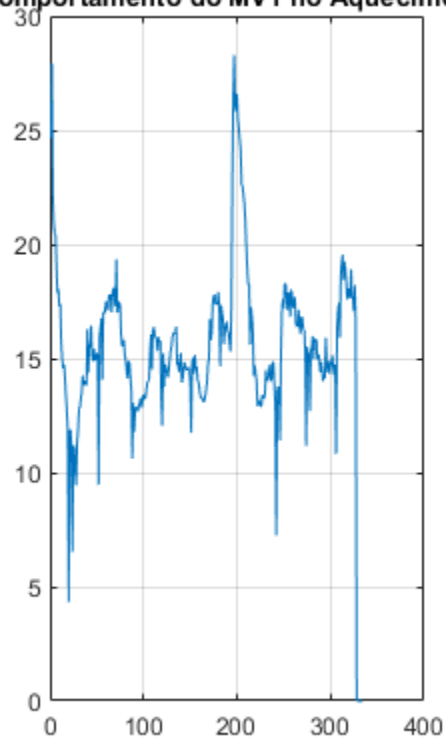
Comportamento Geral- Sistema - Malha 1



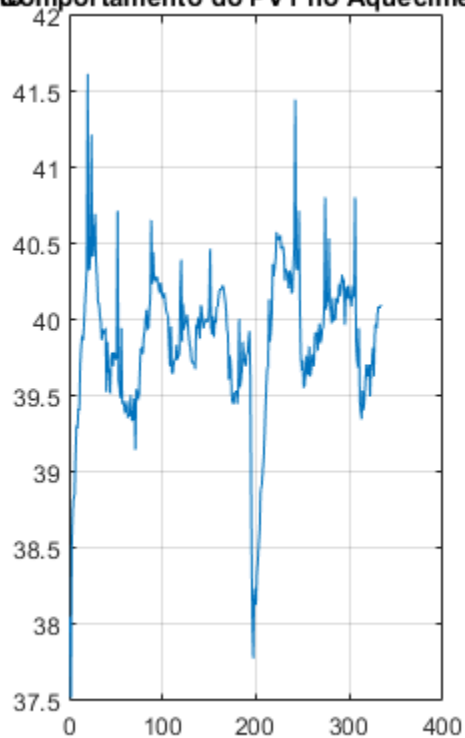
Comportamento Geral- Sistema - Malha 2

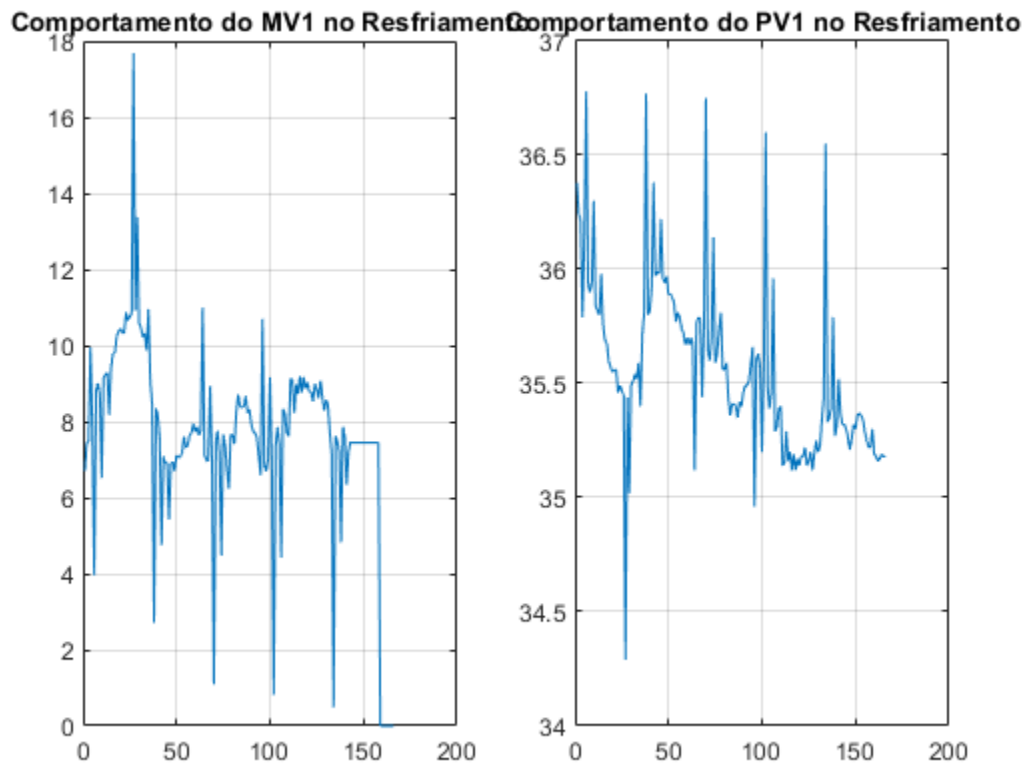


Comportamento do MV1 no Aquecimento



Comportamento do PV1 no Aquecimento





CODIGO REFERENTE AO PROJETO 2- IDENTIFICACAO DE SISTEMAS

```
DeltaT= 2;
```

```
%intervalos do sistema
```

```
pv_subida_11 = dados_coletados.pv1(532:865);  
pv_descida_11 = dados_coletados.pv1(896:1061);
```

```
pv_subida_22 = dados_coletados.pv2(1336:1470);  
pv_descida_22 = dados_coletados.pv2(1339:1570);
```

```
pv_subida_21 = dados_coletados.pv1(1336:1470);  
pv_descida_21 = dados_coletados.pv1(1339:1570);
```

```
pv_subida_12 = dados_coletados.pv2(532:865);  
pv_descida_12 = dados_coletados.pv2(896:1061);
```

AQUECIMENTO

PARA G11

```
y = pv_subida_11 - pv_subida_11(1); %retirando o valor inicial  
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo
```

```

Kp = Kp_G11;
Ti = 1/Ti_G11;
Td = Td_G11;

num_aprox = K11;
den_aprox = [T11 1];
stop_time = t(end);
degrau = 40 - pv_subida_11(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_11 = out.simout.time;
y_11 = out.simout.signals.values;
Ts = t_11(2) - t_11(1); %tempo de amostragem
n_reposo = round(L11 / Ts); % atraso na referencia
u_11 = degrau*[zeros(1, n_reposo), ones(1, length(t_11) - n_reposo)];
%referencia

figure(17)
plot(t, y, t_11, y_11, t_11, u_11);
title('Resposta do Sistema ao Aquecimento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G11 Simulado', 'Referência');
grid on

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['Dados Reais G11:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_11(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

y = y_11;
t = t_11;

```

```

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['SIMULAÇÃO G11:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_11(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

%PARA G12
y = pv_subida_12 - pv_subida_12(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G11;
Ti = 1/Ti_G11;
Td = Td_G11;

num_aprox = K12;
den_aprox = [T12 1];
stop_time = t(end);
degrau = 36 - pv_subida_12(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_12 = out.simout.time;
y_12 = out.simout.signals.values;
Ts = t_12(2) - t_12(1); %tempo de amostragem
n_reposo = round(L12 / Ts); % atraso na referencia
u_12 = degrau*[zeros(1, n_reposo), ones(1, length(t_12) - n_reposo)];
%referencia

figure(18)
plot(t, y, t_12, y_12, t_12, u_12);
title('Resposta do Sistema ao Aquecimento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G12 Simulado', 'Referência');
grid on

```

```

% PARA G21
y = pv_subida_21 - pv_subida_21(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G22;
Ti = 1/Ti_G22;
Td = Td_G22;

num_aprox = K21;
den_aprox = [T21 1];
stop_time = t(end);
degrau = 36 - pv_subida_21(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_21 = out.simout.time;
y_21 = out.simout.signals.values;
Ts = t_21(2) - t_21(1); %tempo de amostragem
n_reposo = round(L21 / Ts); % atraso na referencia
u_21 = degrau*[zeros(1, n_reposo), ones(1, length(t_21) - n_reposo)];
%referencia

figure(19)
plot(t, y, t_21, y_21, t_21, u_21);
title('Resposta do Sistema ao Aquecimento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G21 Simulado', 'Referência');
grid on

%PARA G22
y = pv_subida_22 - pv_subida_22(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G22;
Ti = 1/Ti_G22;
Td = Td_G22;

num_aprox = K22;
den_aprox = [T22 1];
stop_time = t(end);
degrau = 40 - pv_subida_22(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_22 = out.simout.time;
y_22 = out.simout.signals.values;
Ts = t_22(2) - t_22(1); %tempo de amostragem
n_reposo = round(L22 / Ts); % atraso na referencia
u_22 = degrau*[zeros(1, n_reposo), ones(1, length(t_22) - n_reposo)];
%referencia

```

```

figure(20)
plot(t, y, t_22, y_22, t_22, u_22);
title('Resposta do Sistema ao Aquecimento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G22 Simulado', 'Referência');
grid on

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['Dados Reais G22:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_22(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

y = y_22;
t = t_22;

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

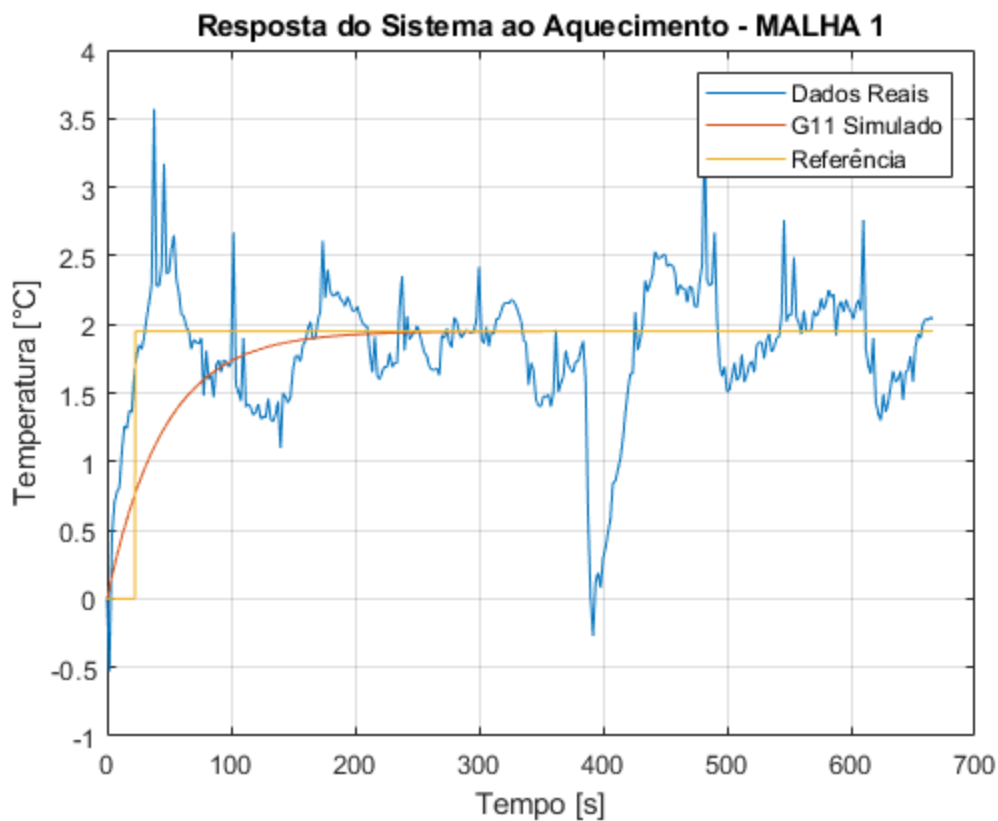
% Overshoot;
over = max(y) - y(end);

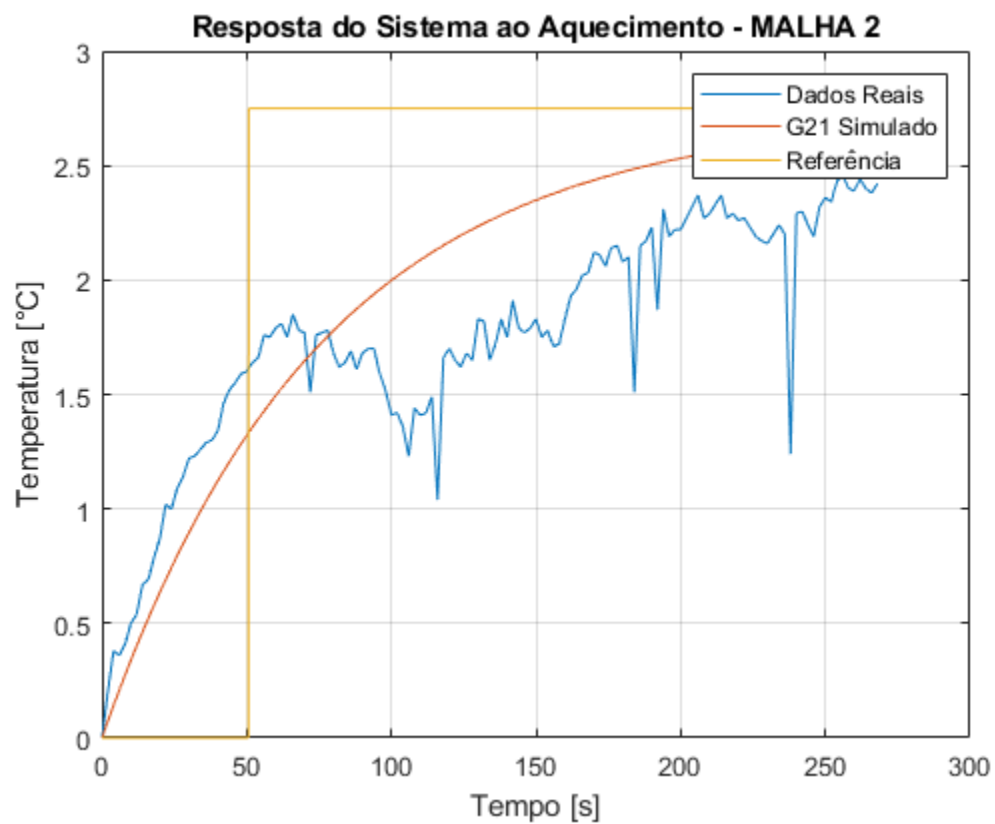
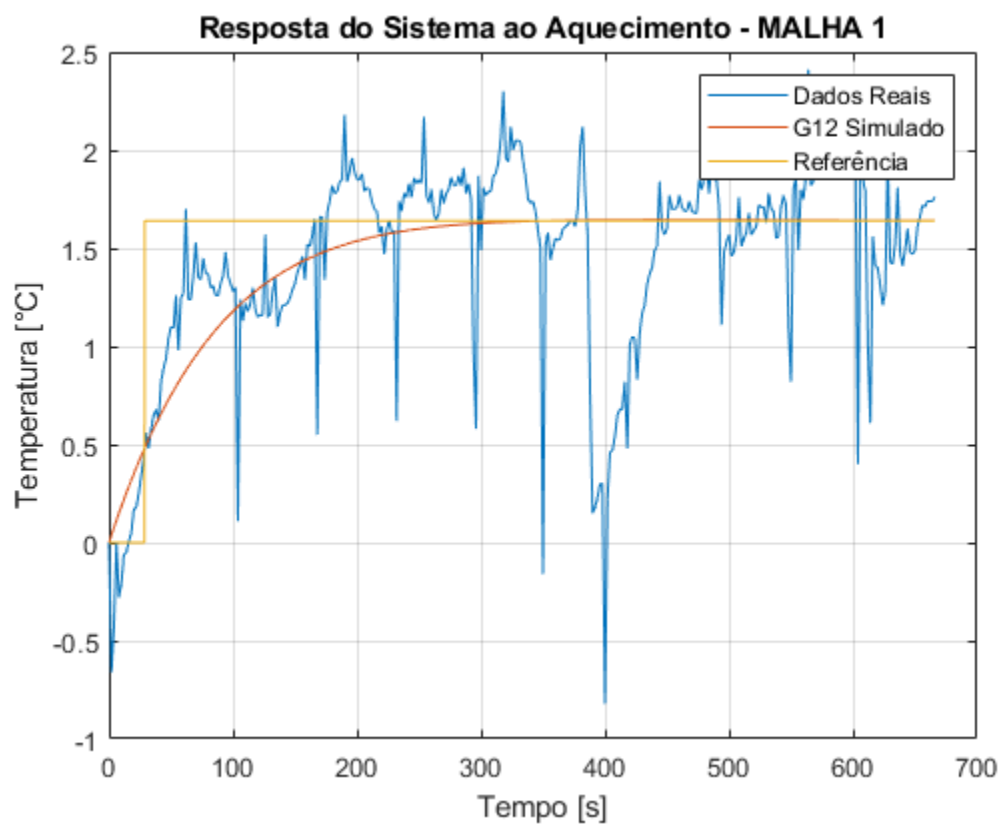
disp(['SIMULAÇÃO G22:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_22(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);

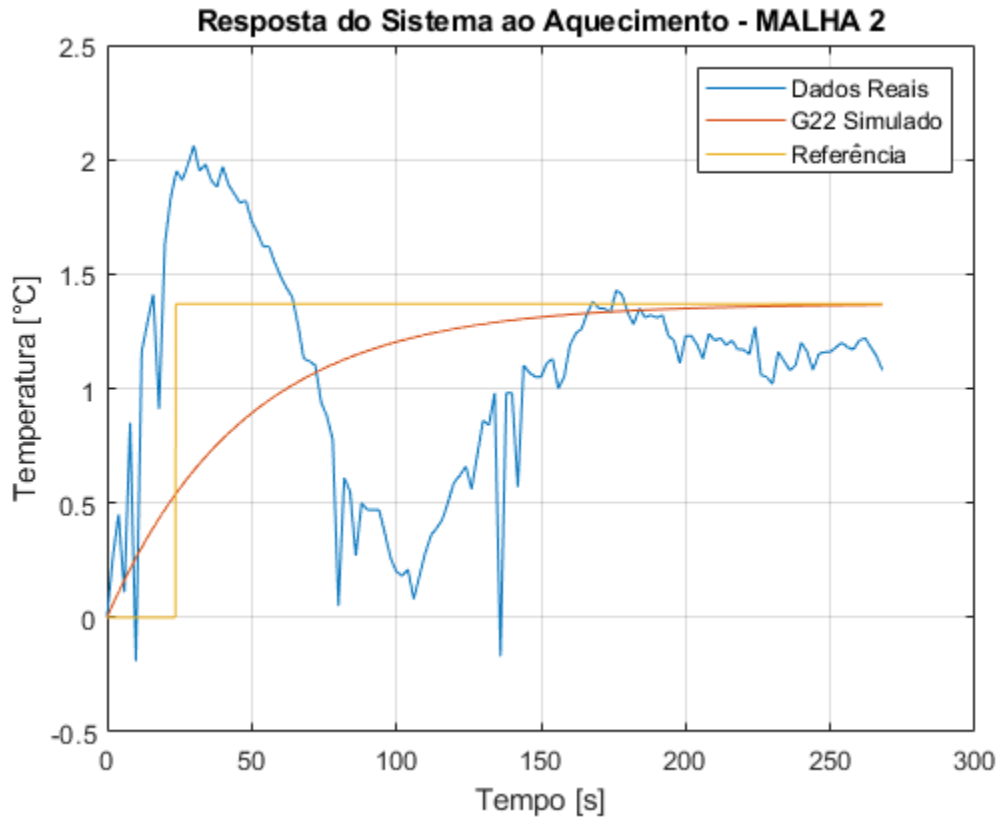
```

```
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);  
disp(['Overshoot: ', num2str(over)]);
```

```
Dados Reais G11:  
Valor Final: 40.09  
Tempo de subida: 26 s  
Tempo de acomodação: 32 s  
Overshoot: 1.53  
SIMULAÇÃO G11:  
Valor Final: 40  
Tempo de subida: 104.95 s  
Tempo de acomodação: 178.3 s  
Overshoot: 0  
Dados Reais G22:  
Valor Final: 39.71  
Tempo de subida: 12 s  
Tempo de acomodação: 72 s  
Overshoot: 0.98  
SIMULAÇÃO G22:  
Valor Final: 39.9951  
Tempo de subida: 107.88 s  
Tempo de acomodação: 178.2 s  
Overshoot: 0
```







IAE (INTEGRAL ERRO ABSOLUTO) - AQUECIMENTO

```
disp(['PARA O CASO DE AQUECIMENTO']);  
% PARA G11  
% SIMULADO  
u_11 = u_11(:);  
% erro absoluto  
erro = abs(u_11 - y_11);  
  
% Calcular tempo de amostragem Δt  
dt = t_11(2) - t_11(1);  
  
% Calcular IAE  
IAE = sum(erro) * dt;  
disp(['IAE G11 simulado: ', num2str(IAE)])  
  
%REAL  
y = pv_subida_11 - pv_subida_11(1);  
t = 0:DeltaT:(length(y)-1)*DeltaT;  
  
u_11 = u_11';  
u_11 = u_11(1:length(t));
```

```

% erro absoluto
erro = abs(u_11 - y);

% Calcular tempo de amostragem Δt
dt = t(2) - t(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G11 real: ', num2str(IAE)])

% PARA G22
% SIMULADO
u_22 = u_22(:);
% erro absoluto
erro = abs(u_22 - y_22);

% Calcular tempo de amostragem Δt
dt = t_22(2) - t_22(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G22 simulado: ', num2str(IAE)])

%REAL
y = pv_subida_22 - pv_subida_22(1);
t = 0:DeltaT:(length(y)-1)*DeltaT;

u_22 = u_22';
u_22 = u_22(1:length(t));

% erro absoluto
erro = abs(u_22 - y);

% Calcular tempo de amostragem Δt
dt = t(2) - t(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G22 real: ', num2str(IAE)])

```

```

PARA O CASO DE AQUECIMENTO
IAE G11 simulado: 63.3732
IAE G11 real: 1214.24
IAE G22 simulado: 46.1682
IAE G22 real: 290.88

```

Resfriamento

PARA G11

```

y = pv_descida_11 - pv_descida_11(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G11;

```

```

Ti = 1/Ti_G11;
Td = Td_G11;

num_aprox = K11;
den_aprox = [T11 1];
stop_time = t(end);
degrau = 35 - pv_descida_11(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_11 = out.simout.time;
y_11 = out.simout.signals.values;
Ts = t_11(2) - t_11(1); %tempo de amostragem
n_reposo = round(L11 / Ts); % atraso na referencia
u_11 = degrau*[zeros(1, n_reposo), ones(1, length(t_11) - n_reposo)];
%referencia

figure(17)
plot(t, y, t_11, y_11, t_11, u_11);
title('Resposta do Sistema ao Resfriamento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G11 Simulado', 'Referência');
grid on

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['Dados Reais G11:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_11(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

y = y_11;
t = t_11;

% Valor final (Regime Permanente)
v_f = y(end);

```

```

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['SIMULAÇÃO G11:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_11(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

%PARA G12
y = pv_descida_12 - pv_descida_12(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G11;
Ti = 1/Ti_G11;
Td = Td_G11;

num_aprox = K12;
den_aprox = [T12 1];
stop_time = t(end);
degrau = 33 - pv_descida_12(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_12 = out.simout.time;
y_12 = out.simout.signals.values;
Ts = t_12(2) - t_12(1); %tempo de amostragem
n_reposo = round(L12 / Ts); % atraso na referencia
u_12 = degrau*[zeros(1, n_reposo), ones(1, length(t_12) - n_reposo)];
%referencia

figure(18)
plot(t, y, t_12, y_12, t_12, u_12);
title('Resposta do Sistema ao Resfriamento - MALHA 1');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G12 Simulado', 'Referência');
grid on

% PARA G21
y = pv_descida_21 - pv_descida_21(1); %retirando o valor inicial

```

```

t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G22;
Ti = 1/Ti_G22;
Td = Td_G22;

num_aprox = K21;
den_aprox = [T21 1];
stop_time = t(end);
degrau = 33 - pv_subida_21(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_21 = out.simout.time;
y_21 = out.simout.signals.values;
Ts = t_21(2) - t_21(1); %tempo de amostragem
n_reposo = round(L21 / Ts); % atraso na referencia
u_21 = degrau*[zeros(1, n_reposo), ones(1, length(t_21) - n_reposo)];
%referencia

figure(19)
plot(t, y, t_21, y_21, t_21, u_21);
title('Resposta do Sistema ao Aquecimento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G21 Simulado', 'Referência');
grid on

%PARA G22
y = pv_descida_22 - pv_descida_22(1); %retirando o valor inicial
t = 0:DeltaT:(length(y)-1)*DeltaT; %vetor de tempo

Kp = Kp_G22;
Ti = 1/Ti_G22;
Td = Td_G22;

num_aprox = K22;
den_aprox = [T22 1];
stop_time = t(end);
degrau = 35 - pv_descida_22(1);

open('Projeto_Final_Simulink.slx');
out = sim('Projeto_Final_Simulink.slx');

t_22 = out.simout.time;
y_22 = out.simout.signals.values;
Ts = t_22(2) - t_22(1); %tempo de amostragem
n_reposo = round(L22 / Ts); % atraso na referencia
u_22 = degrau*[zeros(1, n_reposo), ones(1, length(t_22) - n_reposo)];
%referencia

figure(20)
plot(t, y, t_22, y_22, t_22, u_22);

```

```

title('Resposta do Sistema ao Resfriamento - MALHA 2');
xlabel('Tempo [s]');
ylabel('Temperatura [°C]');
legend('Dados Reais', 'G22 Simulado', 'Referência');
grid on

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

% Overshoot;
over = max(y) - y(end);

disp(['Dados Reais G22:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_22(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

y = y_22;
t = t_22;

% Valor final (Regime Permanente)
v_f = y(end);

% Encontrar o tempo correspondente ao instante em que a resposta atinge 90%
y_90 = y(end) - (y(end) - min(y))*0.1;
idx = find(y >= y_90, 1);
t_s = t(idx);

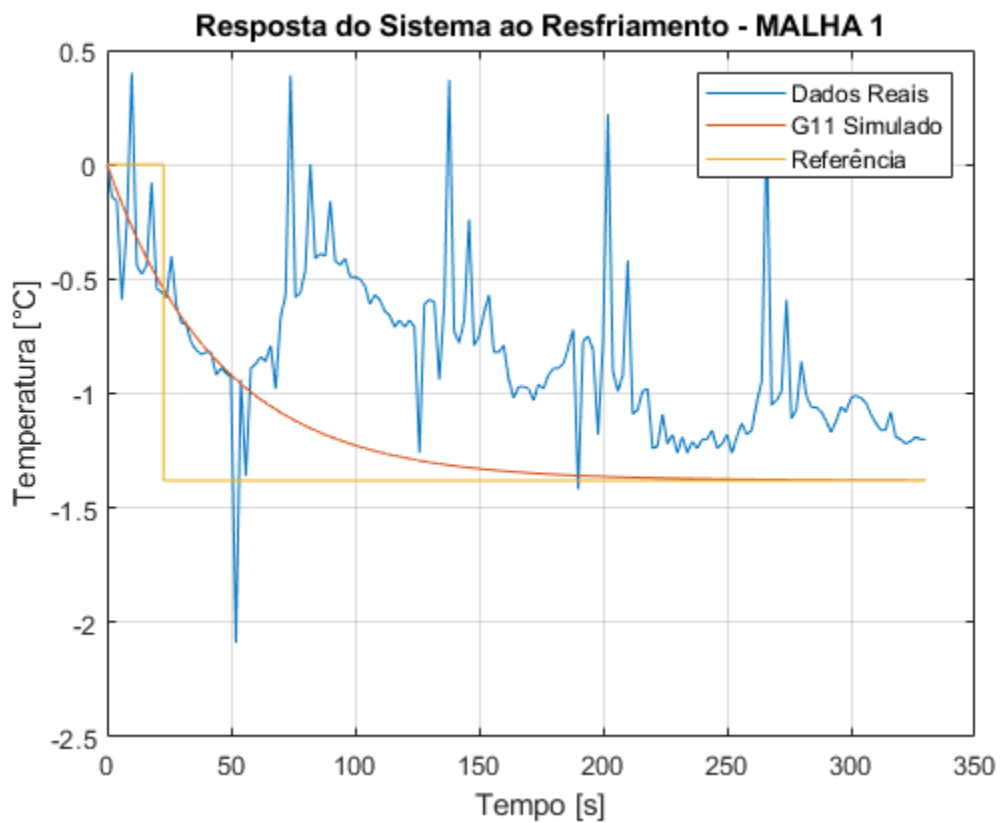
% Encontrar o tempo correspondente ao instante em que a variação da resposta
atinge 2%
tolerance = 0.02 * (y(end) - min(y));
idx = find(abs(y - y(end)) <= tolerance, 1);
t_r = t(idx);

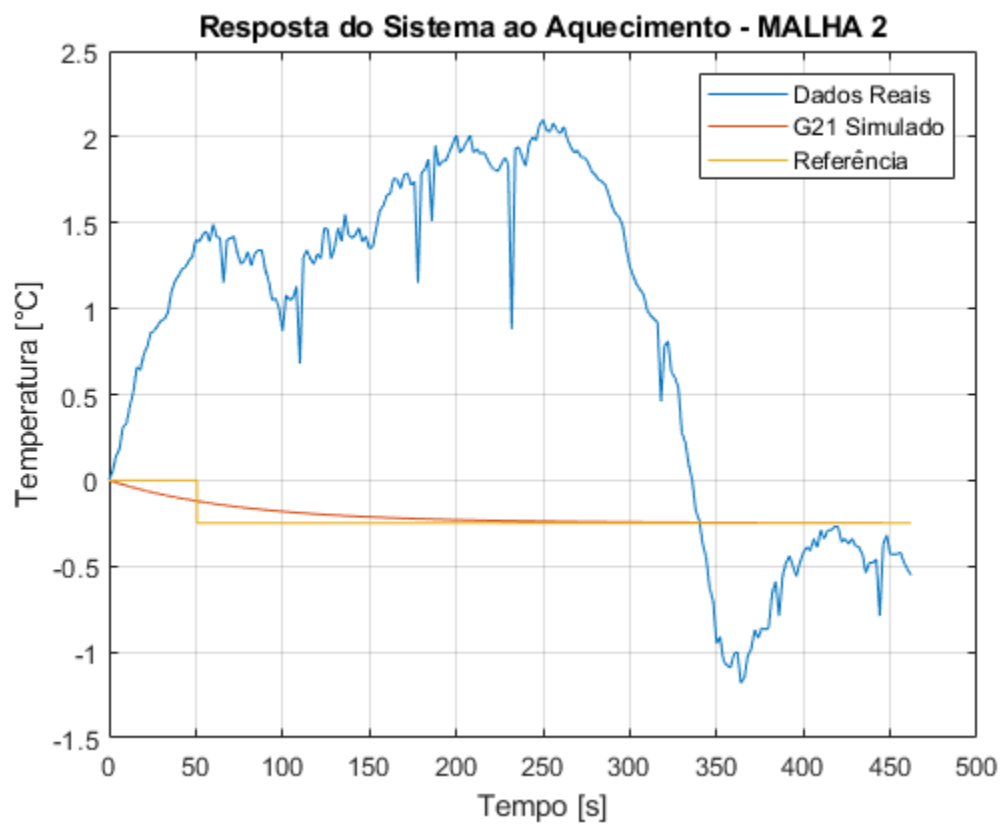
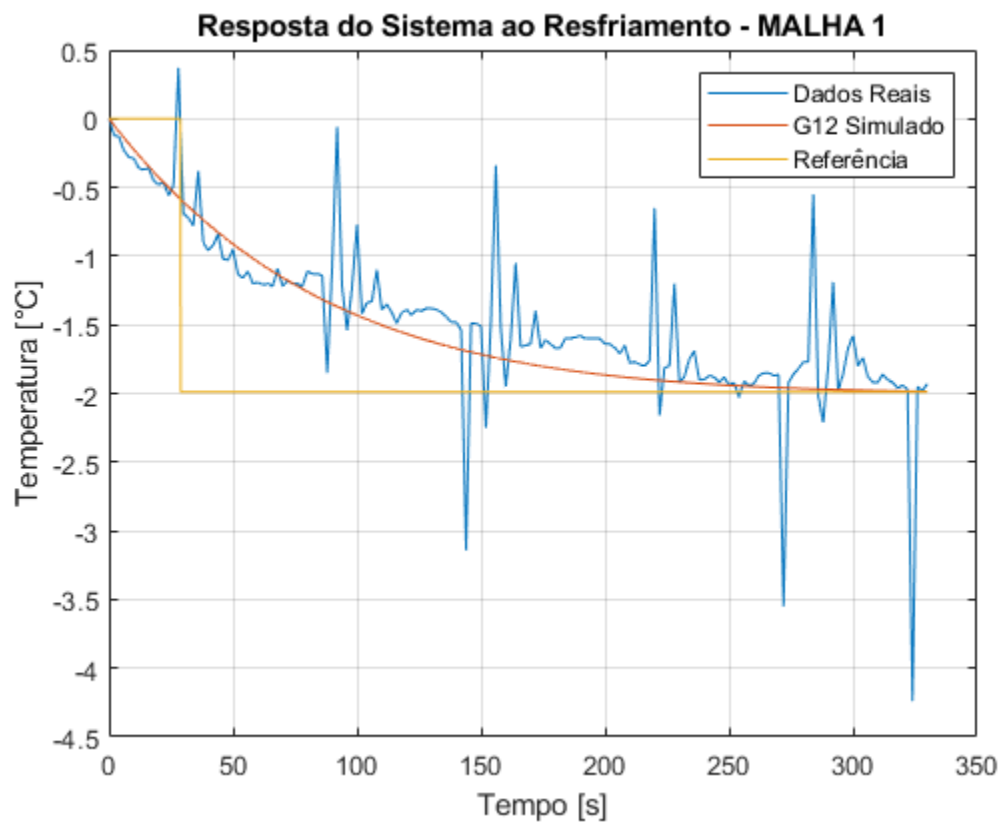
% Overshoot;
over = max(y) - y(end);

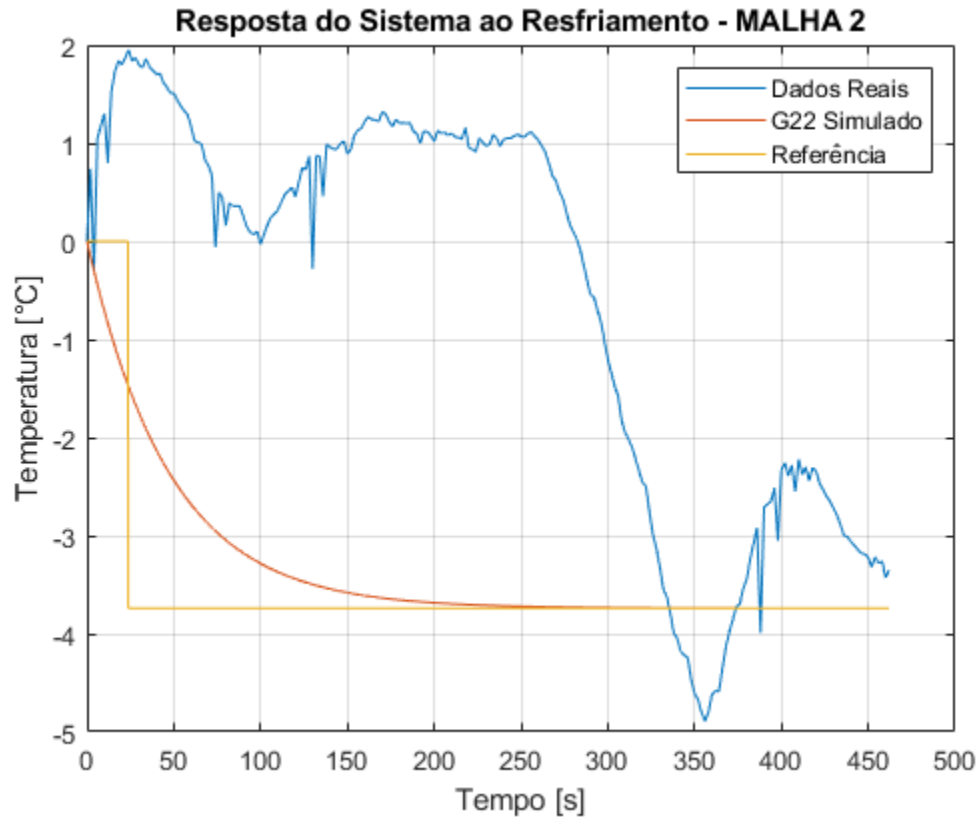
disp(['SIMULAÇÃO G22:'])
disp(['Valor Final: ', num2str(v_f + pv_subida_22(1))]);
disp(['Tempo de subida: ', num2str(t_s), ' s']);
disp(['Tempo de acomodação: ', num2str(t_r), ' s']);
disp(['Overshoot: ', num2str(over)]);

```

Dados Reais G11:
Valor Final: 36.85
Tempo de subida: 0 s
Tempo de acomodação: 232 s
Overshoot: 1.6
SIMULAÇÃO G11:
Valor Final: 36.671
Tempo de subida: 0 s
Tempo de acomodação: 330 s
Overshoot: 1.379
Dados Reais G22:
Valor Final: 35.28
Tempo de subida: 0 s
Tempo de acomodação: 330 s
Overshoot: 5.3
SIMULAÇÃO G22:
Valor Final: 34.8902
Tempo de subida: 0 s
Tempo de acomodação: 462 s
Overshoot: 3.7398







IAE (INTEGRAL ERRO ABSOLUTO) - RESFRIAMENTO

```
disp(['PARA O CASO DE RESFRIAMENTO']);  
% PARA G11  
% SIMULADO  
u_11 = u_11(:);  
% erro absoluto  
erro = abs(u_11 - y_11);  
  
% Calcular tempo de amostragem Δt  
dt = t_11(2) - t_11(1);  
  
% Calcular IAE  
IAE = sum(erro) * dt;  
disp(['IAE G11 simulado: ', num2str(IAE)])  
  
%REAL  
y = pv_descida_11 - pv_descida_11(1);  
t = 0:DeltaT:(length(y)-1)*DeltaT;  
  
u_11 = u_11';  
u_11 = u_11(1:length(t));
```

```

% erro absoluto
erro = abs(u_11 - y);

% Calcular tempo de amostragem Δt
dt = t(2) - t(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G11 real: ', num2str(IAE)])

% PARA G22
% SIMULADO
u_22 = u_22(:);
% erro absoluto
erro = abs(u_22 - y_22);

% Calcular tempo de amostragem Δt
dt = t_22(2) - t_22(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G22 simulado: ', num2str(IAE)])

%REAL
y = pv_descida_22 - pv_descida_22(1);
t = 0:DeltaT:(length(y)-1)*DeltaT;

u_22 = u_22';
u_22 = u_22(1:length(t));

% erro absoluto
erro = abs(u_22 - y);

% Calcular tempo de amostragem Δt
dt = t(2) - t(1);

% Calcular IAE
IAE = sum(erro) * dt;
disp(['IAE G22 real: ', num2str(IAE)])

PARA O CASO DE RESFRIAMENTO
IAE G11 simulado: 44.8037
IAE G11 real: 279.3
IAE G22 simulado: 126.6547
IAE G22 real: 790.8

```

Published with MATLAB® R2024a