```
% Instituto Federal da Paraíba
% Data: 11/05/2023
% Ademar Gonçalves da Costa Júnior
% Sistemas de Controle I
% Projeto 2 - Sistemas de Controle I
% Grupo 3: Alysson, Fabrício,Gabriel


close all
clear all
% sistema em malha aberta
num = [5.56];
den = [1 11.55 5.56];
disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
% gerar gráfico
sysc = tf(num,den); % objeto função de transferência
 opt = stepDataOptions('StepAmplitude',1.5);
[y,t] = step(sysc); % gera o gráfico para um degrau UNITÁRIO
figure() % cria objeto figura
plot(t,y,'LineWidth',1.3) % gera gráfico
title('Malha aberta');
xlabel('tempo(s)') % título do eixo x
ylabel('Amplitude do sistema') % título do eixo y
grid % coloca grid no gráfico

%###########################################################################
%###########################################################################

% 2a. Calculem por meio de uma rotina computacional em Matlab, os parâmetros
%dos itens 1a e 1b;
%--------------------------------------------------------------------------
% 1a. Nesse item, acrescentem o ganho K antes do sistema dinâmico
% (ramo direto),% e projetem para que o sistema em malha fechada apresente
% overshoot máximo de 5% (cálculo do K que atenda a essa premissa)
% (ver tema: diagrama de blocos). Obs: analisem para a entrada do tipo
% degrau (amplitude dada para cada grupo).
%--------------------------------------------------------------------------

% para um overshoot de 5%, temos:
% k_ov = 11.64
% num_e = [k_ov*5.56];
% poly_den_e = [1 11.55 5.56+5.56*k_ov];
num_e = [64.71];
poly_den_e = [1 11.55 70.27];

%--------------------------------------------------------------------------
% 1b. Com o valor do ganho K calculado no item 1a, estimem o tempo de pico,
% o overshoot e o tempo de assentamento para o sistema em malha fechada.
%--------------------------------------------------------------------------

omegan = sqrt(poly_den_e(3)); % frequência natural
csi = poly_den_e(2)/(2*omegan); % coeficiente de amortecimento
```

```
Ts = 4/(csi*omegan); % tempo de acomodação
Tp = pi/(omegan*sqrt(1-csi^2)); % tempo de pico
overshoot = 100*exp((-csi*pi)/sqrt(1-csi^2)); % overshoot ou sobressinal
raizes_den = roots(poly_den_e); % raízes do denominador
disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause;
disp('Omega_n:'); disp(omegan);
disp('csi:'); disp(csi);
disp('Tr:'); disp(Ts);
disp('Tp:'); disp(Tp);
disp('Overshoot:'); disp(overshoot);
disp('Raízes_denominador:'); disp(raizes_den);

%##########################################################################
%##########################################################################

% 2b. Gerem o gráfico da saída do sistema em malha fechada, de acordo com o
% ganho K encontrado no item 1a para a entrada do tipo degrau (amplitude
% dada para cada grupo).

disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
% gerar gráfico
sysc6 = tf(num_e,poly_den_e); % objeto função de transferência
[y6,t6] = step(sysc6,opt);
figure()
te = 0:0.01:15;
s=tf('s');
sys3_ordem= tf(1.5*[64.61],[1 11.55 70.17]); % sistema em malha fechada
sys2_ordem= tf([5.56],[1 11.55 5.56]); % sistema em malha aberta
hold on
f3 = step(sys3_ordem,te);
f2 = step(sys2_ordem,te);
plot(te,f3,'LineWidth',1.3)
plot(te,f2,'LineWidth',1.3)
title('Comparação do sistema dinâmico com ganho K e com overshoot máximo de 5% ');
xlabel('tempo(s)')
ylabel('Amplitude')
grid on
legend('Malha fechada','Malha aberta')

%##########################################################################
%##########################################################################

% 2c. Gerem o diagrama de polos e zeros dos sistemas em malha aberta e malha
% fechada (vejam o item 1c), em um mesmo gráfico;

disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
figure()
subplot(2,1,1)
h1 = pzplot(sysc(:,:,1),'r');grid on
title('Malha aberta')
```

```
subplot(2,1,2)
h2 = pzplot(sysc6(:,:,1),'r'); grid on
title('Malha fechada')

%##########################################################################
%##########################################################################

% 2d. Analisem o item 1d por meio de gráficos da saída do sistema em malha
% fechada;

%--------------------------------------------------------------------------
% 1d. Determinem a faixa de valores do ganho K do sistema em malha fechada,
% para que o sistema seja estável, instável e marginalmente estável (ver
% tema: estabilidade);
%--------------------------------------------------------------------------

disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
% Critério de Estabilidade Routh-Hurwitz:
K_temp = input('Digite o K de acordo com a condições abaixo: \n K > -1 => estável ↙
\n K < -1 => instável \n K = -1 => marginalmente estável  \n ');
entrada = input('Você quer calcular uma nova tabela de Routh-Hurwitz S/N ', 's');
if entrada == 's' || entrada == 'S'

 % Pega coef do vetor e constroi as duas primeiras linhas
 syms k
 num_temp = [K_temp*5.56];
 coeffVector = [1 11.55 5.56+5.56*K_temp];
 ceoffLength = length(coeffVector);
 rhTableColumn = round(ceoffLength/2);
 % Inicializa a tabela Routh-Hurwitz com vetor nulo
 rhTable = zeros(ceoffLength,rhTableColumn);
 % Computa a primeira linha da tabela
 rhTable(1,:) = coeffVector(1,1:2:ceoffLength);
 % Verifica se o comprimento do vetor de coeficientes é par ou ímpar
 if (rem(ceoffLength,2) ~= 0)
 % se par, a segunda linha da tabela será
 rhTable(2,1:rhTableColumn - 1) = coeffVector(1,2:2:ceoffLength);
 else
 % se impar, a segunda linha da tabela será
 rhTable(2,:) = coeffVector(1,2:2:ceoffLength);
 end

disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
figure()
func_temp = tf(num_temp,coeffVector);
h1 = pzplot(func_temp(:,:,1),'r');grid on
disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
plot_temp = step(func_temp);
figure()
plot(plot_temp,'LineWidth',1.3);
```

```
title('Análise de estabilidade em malha fechada')
xlabel('tempo(s)')
ylabel('Amplitude')
 %% Calcular as linhas da tabela de Routh-Hurwitz
 % Define epss como um valor pequeno
 epss = 0.01;
 % Calcula outros elementos da tabela
 for i = 3:ceoffLength
 % caso especial: linhas de zeros
 if rhTable(i-1,:) == 0
 order = (ceoffLength - i);
 cnt1 = 0;
 cnt2 = 1;
 for j = 1:rhTableColumn - 1
 rhTable(i-1,j) = (order - cnt1) * rhTable(i-2,cnt2);
 cnt2 = cnt2 + 1;
 cnt1 = cnt1 + 2;
 end
 end
 for j = 1:rhTableColumn - 1
 % fprimeiro elemento da linha superior
 firstElemUpperRow = rhTable(i-1,1);
 % computa cada elemento da tabela
 rhTable(i,j) = ((rhTable(i-1,1) * rhTable(i-2,j+1)) - ....
 (rhTable(i-2,1) * rhTable(i-1,j+1))) / firstElemUpperRow;
 end
 % caso especial: zero na primeira coluna
 if rhTable(i,1) == 0
 rhTable(i,1) = epss;
 end
 end


 %% Calcula o número de pólos do lado direito (pólos instáveis)
 % Inicializa pólos instáveis ??com zero
 unstablePoles = 0;
 % Verifica a mudança do sinal
 for i = 1:ceoffLength - 1
 if sign(rhTable(i,1)) * sign(rhTable(i+1,1)) == -1
 unstablePoles = unstablePoles + 1;
 end
 end
 % Print dados na command window
 fprintf('\n Tabela Routh-Hurwitz:\n')
 rhTable
 % Printa o resultado
 if unstablePoles == 0 & K_temp > -1
 fprintf('~~~~~> É um sistema estável! <~~~~~\n')
 else
     if K_temp == -1
        fprintf('~~~~~> É um sistema marginalmente estável! <~~~~~\n')
     else
        fprintf('~~~~~> É um sistema instável! <~~~~~\n')
```

```
    end
 end
 fprintf('\n Número de polos no lado direito =%2.0f\n',unstablePoles)
 reply = input('Você quer ver as raízes do sistema? S/N ', 's');
 if reply == 's' || reply == 'S'
 sysRoots = roots(coeffVector);
 fprintf('\n Raízes de coeficientes polinomiais dadas :\n')
 sysRoots


 end
end


%###########################################################################
%###########################################################################

% 2e. Calculem por meio de uma rotina computacional em Matlab, os
% parâmetros dos itens 1e e 1f;

%---------------------------------------------------------------------------
% 1e. Nesse item, acrescentem o ganho K novamente antes do sistema dinâmico
% (ramo direto), e projetem para que o sistema em malha fechada, possua erro
% em regime permanente de, no máximo, 5% (não utilizar os valores de K
% estimados anteriormente).
%---------------------------------------------------------------------------

syms s k
E_estacionario = limit(1/(1+k*(5.56/(s^2+11.55*s+5.56))),s,0);
%E_estacionario = 5% = 5/100 = 1/(1+k)
% logo E_estacionario = 1/(1+k) = 0.05
k_parcial = 1/0.05; %k_parcial = 20
k = k_parcial - 1;

%---------------------------------------------------------------------------
% 1f. Com o valor do ganho K calculado no item 1e, calculem as constantes de
% erro estático e o erro em regime permanente para as entradas do tipo degrau
% (amplitude dada para cada grupo) e rampa unitária (ver tema: erro em regime
% permanente).
%---------------------------------------------------------------------------

disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause
syms s
% Aplicado o degrau R(s)=0.7/s
kp = k; %Constante de posição
if(1+kp == 0)
 disp('O erro em regime permanente quando aplicado o degrau = infinito' );
else
 ErD = vpa(round(10000*1*1.5/(1+kp))/10000);
 disp('o erro em regime permanente quando aplicado o degrau 1.5 é:');
 disp(ErD);
end
```

```
disp('A constante de posição é :');
disp(kp);
% Aplicando a Rampa unitaria R(s)=1/s^2
kv = vpa(round(1000*limit(s*5.56/(s^2+1.56*s+5.56),s,0))/1000); %Constande de
velocidade
if(kv == 0)
 disp('o erro em regime permanente quando aplicado a rampa = infinito' );
else
ErR = 1./(kv); % erro para a rampa
disp('o erro em regime permanente quando amplicado a rampa é:');
disp(ErR);
end
disp('A constante de velocidade é :');
disp(kv);
ka = vpa(round(1000*limit(s^2*5.56/(s^2+1.56*s+5.56),s,0))/1000); %Constante de
Aceleração
disp('A constante de aceleração é :');
disp(ka);

%###########################################################################
%###########################################################################

% 2f. Analisem por meio de gráficos, comparando com o sistema em malha fechada
% original, como a saida do sistema em malha fechada e o sinal de erro se
% comportam com variações de +-10% nos valores de 'a' e de 'K'. Aproveitamento
% do item 1g.

%---------------------------------------------------------------------------
% 1g. Utilizando a mesma estrutura do item 1e, calculem a sensibilidade do
% erro em regime permanente em malha fechada para variações no parâmetro do
% ganho "K" e no parâmetro "b" da função de transferência em malha aberta
% (ex: G(s) = num(s)/den(s), com den(s)=s^2+as+b), com a aplicação da entrada
% do tipo degrau (amplitude dada para cada grupo).
%---------------------------------------------------------------------------
disp(' '); disp('DIGITE QUALQUER TECLA PARA CONTINUAR'); disp(' ');
pause

syms s k b
E_estacionario = limit(1.5/(1+k*(5.56/(s^2+11.55*s+5.56*b))),s,0);
disp('Sensibilidade do erro em regime permanente para o parâmetro K: ' );
parcial_k = (k/E_estacionario);
sensk = parcial_k *  diff(E_estacionario,k);
pretty(sensk);
disp('Sensibilidade do erro em regime permanente para o parâmetro b: ' );
parcial_b = (b/E_estacionario);
sensb = parcial_b *  diff(E_estacionario,b);
pretty(sensb)
```