**Refaça o exemplo A.10.11. Considere um controlador PID onde Gc = k (s+a)² / a que deve controlar a**

**planta Gp = 1 / s(s+1)(s+5). Obtenha:**

**a) Um algoritmo que faça a busca automática dos ganhos. Determine o valores dos ganhos k e a de modo que o máximo sobre-sinal esteja dentro do intervalo 5% < M < 10% com tempo de acomodação ts < 3s. Adote a região de busca 2 ≤ k ≤ 50 e 0, 05 ≤ a ≤ 2.**

Foi desenvolvido no matlab o seguinte código abaixo:

clear all; %deletar todas as variaveis e seus valores previamente existentes.

clc; % apagar o texto atual da janela de comando.

% 2 <= Kc <= 50

min\_search\_kc = 2; % valor minimo de teste de kc

max\_search\_kc = 50; % valor máximo de teste de kc

step\_kc = 0.2; % quanto k irá varia a cada iteração

% 0.05 <= a <= 2

min\_search\_a = 0.05; %valor minimo de teste de a

max\_search\_a = 2; %valor máximo de teste de kc

step\_a = 0.2; % quanto k irá varia a cada iteração

min\_Mp = 5; %valor minimo de Mp(máximo sobressinal) desejado (Entrada em porcentagem!!!)

max\_Mp = 10; %valor máximo de Mp(máximo sobressinal) desejado (Entrada em porcentagem!!!)

max\_ts = 7.9; % valor máximo do tempo de acomodação em segundos

% entrada da FT pelos coeficientes equa. polinomial

% num = [1]

% den = [1 6 5 0]

% Gp = tf(num, den)

% entrada da FT pelos polos e zeros conhecidos

Gp = zpk([],[0 -1 -5], 1);

S = tf('s');

flag\_value\_found = 0;

t = 0 : 0.01 : 8; %t é uma variavel de tempo contendo uma lista de 0 a 8 com intervalos de 0.01

for kc = max\_search\_kc : -step\_kc : min\_search\_kc %realiza um loop onde k varia de 5 a 2 reduzindo -0.2 a cada iteração

for a = max\_search\_a : -step\_a : min\_search\_a %realiza um loop onde a varia de 1.5 a 0.5 reduzindo -0.2 a cada iteração

Gc = kc\*((S+a)^2)/S;

G = Gc\*Gp;

FTMF = feedback(G,1);

[num, den] = tfdata(FTMF, 'v');

%y = step(num,den,t); %executa uma entrada degrau com a função de transferência armazenada em num e den nos intervalos da lista t e salva na variável y os resultados

performance\_espcs = stepinfo(FTMF); %obtem as especificações de desempenho da resposta a entrada degrau

Mp = performance\_espcs.Overshoot; %verifica o valor máximo atingido(máximo sobressinal) na resposta da entrada degrau.

ts = performance\_espcs.SettlingTime;

if Mp <max\_Mp && Mp > min\_Mp && ts < max\_ts%teste se a condição de máximo sobressinal e tempo de acomodação foi atendida

% if Mp <max\_Mp && Mp > min\_Mp%teste se a condição de máximo sobressinal e tempo de acomodação foi atendida

flag\_value\_found = 1;

break; %atendida condição de máximo sobressinal, interrompe do loop interno que varia a

end

end

if Mp <max\_Mp && Mp > min\_Mp && ts < max\_ts %teste se a condição de máximo sobressinal e tempo de acomodação foi atendida

% if Mp <max\_Mp && Mp > min\_Mp %teste se a condição de máximo sobressinal e tempo de acomodação foi atendida

break; %atendida condição de máximo sobressinal, interrompe do loop externo que varia k

end

end

if not(flag\_value\_found)

disp("<! Não foi possível encontrar um valor 'Kc' e 'a' para as especificações pedidas !>\n");

else

fprintf("# Encontrados os seguintes valores: kc = %.3f ; a = %.3f\n", kc, a);

step(FTMF);

grid % acrescenta linhas de grade

title('Resposta ao degrau unitário') % acrescenta titulo ao gráfico

xlabel('t(s)') % acrescenta legenda do eixo x

ylabel('Saída') % acrescenta legenda do eixo y

str\_k = num2str(kc); % valor de k em formato string prar impressão no gráfico

str\_a = num2str(a); % valor de a em formato string prar impressão no gráfico

str\_Mp = num2str(Mp); % valor do máximo sobresinal em formato string prar impressão no gráfico

text(4.25, 0.54, 'kc = '), text(4.75, 0.54, str\_k) % mostra o valor de K encontrado no gráfico

text(4.25, 0.46, 'a = '), text(4.75, 0.46, str\_a) % mostra o valor de a encontrado no gráfico

text(4.25, 0.38, 'Mp = '), text(4.75, 0.38, str\_Mp) % mostra o valor do máximo sobresinal encontrado no gráfico

fprintf("\n# A função de transferência de malha fechada com controlador é: ")

FTMF

fprintf("# A equação caracteristica do sistema é:\n")

size\_den = size(den, 2);

for ind = 1 : 1 : size\_den

if ind < size\_den

fprintf("%.2f\*S^%d + ", den(ind), size\_den-ind);

else

fprintf("%.2f", den(ind));

end

end

poles = roots(den);

real\_poles = real(poles);

flag\_instable\_system = 0;

for ind = 1 : 1 : size(real\_poles, 2)

if real\_poles(ind) >= 0

flag\_instable\_system = 1;

end

end

if flag\_instable\_system

fprintf("\n\n<! O sistema é instavel !>\n\n");

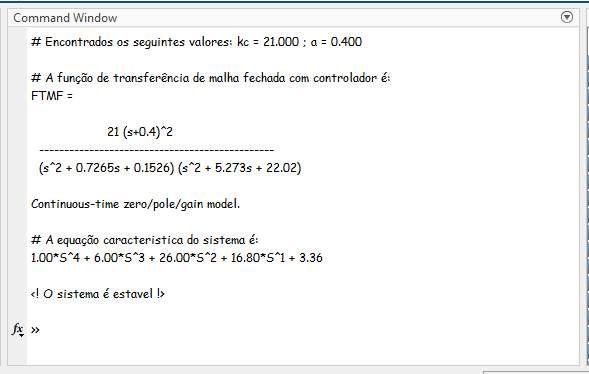
else

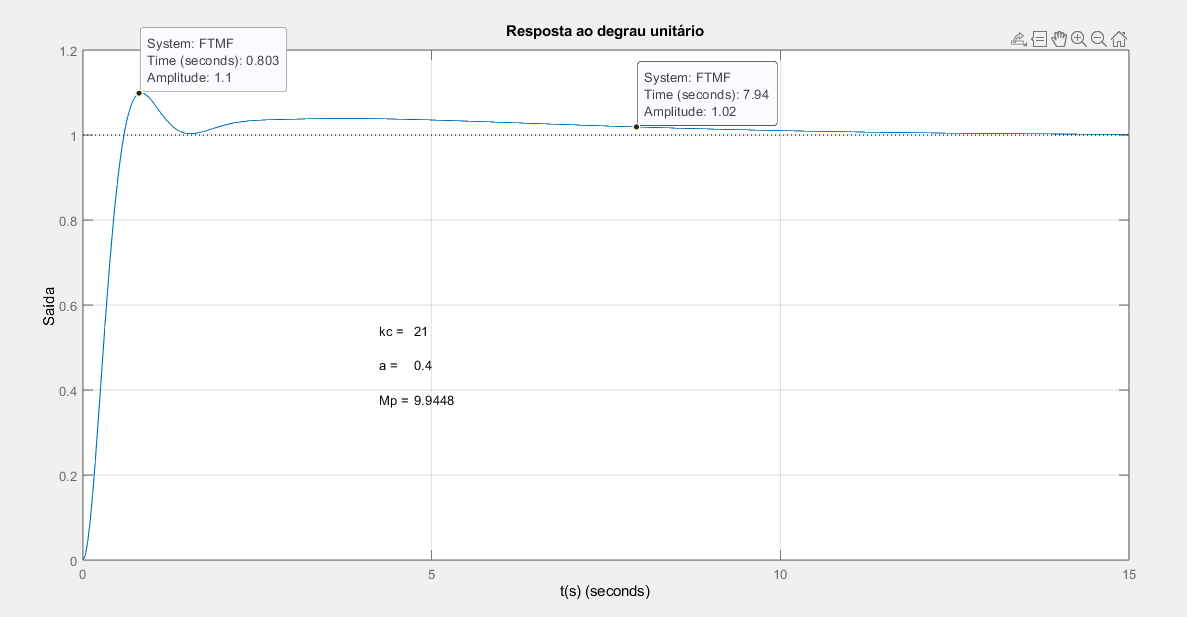
fprintf("\n\n<! O sistema é estavel !>\n\n");

end

end

Não foi possível, porém, obter parâmetros k e a que a atendessem, simultaneamente o máximo sobressinal e o tempo de acomodação pedido, alterando o melhor resultado obitido foi de 9.9% de sobressinal e 7.6s de tempo de acomodação, conforme demonstrado no resultado abaixo:

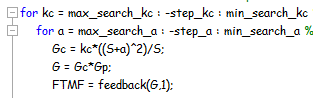


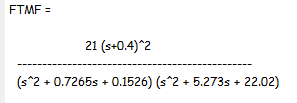


**b) A função de transferência G(s);**

A função de transferência de malha fechada com controlador foi obitida via escript atráves (armazenada na variável FTMF) das seguintes linhas de código:



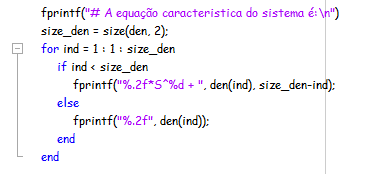




**c) A equação característica;**

A equação característica foi obtida via script nas seguintes linhas de código abaixo:







**d) Determine o máximo valor de ganho para que a resposta a uma entrada degrau seja estável. Lembre-se que existem algumas maneiras de verificar a estabilidade de uma função, pelo método de Routh ou pela posição dos pólos da equação característica. Os pólos de uma função correspondem as suas raízes. As raízes de um polinômio pode ser obtido utilizando a função roots().**