INSTITUTO FEDERAL DO AMAZONAS – IFAM
CAMPUS MANAUS DISTRITO INDUSTRIAL
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

6° Laboratório de Controle Moderno

Busca Numérica de Ganhos Otimizados

MINJAE LINS CHUNG – 2020001395

GABRIEL ALMEIDA SANTOS DE OLIVEIRA – 2021000042

Trabalho solicitado pelo professor Dr. Flávio José Aguiar Soares para atribuição de nota avaliativa na disciplina de Controle Moderno do curso superior de ECAT do IFAM/CMDI.

MANAUS/AM
2023

1. **B.10.11 utilizando o MatLab**

   O seguinte código MatLab foi utilizado:

```
time_range = 0:0.01:8;

for K = 4:-0.05:1;
    for a = 4:-0.05:0.4;
        num = [0 0 1.2*K 2.4*K*a 1.2*K*a^2];
        den = [0.36 1.86 2.5+1.2*K 1+2.4*K*a 1.2*K*a^2];
        sys = tf(num,den);
        y = step(sys, time_range);
        max_y = max(y);
        if max_y < 1.1 & max_y > 1.02
            break;
        end
    end
    if max_y < 1.1 & max_y > 1.02
        break;
    end
end

solution = [K a]

figure
plot(time_range, y)
grid()

title("Resposta ao degrau unitário")
ylabel("Tempo (s)")
xlabel("Saída")
```

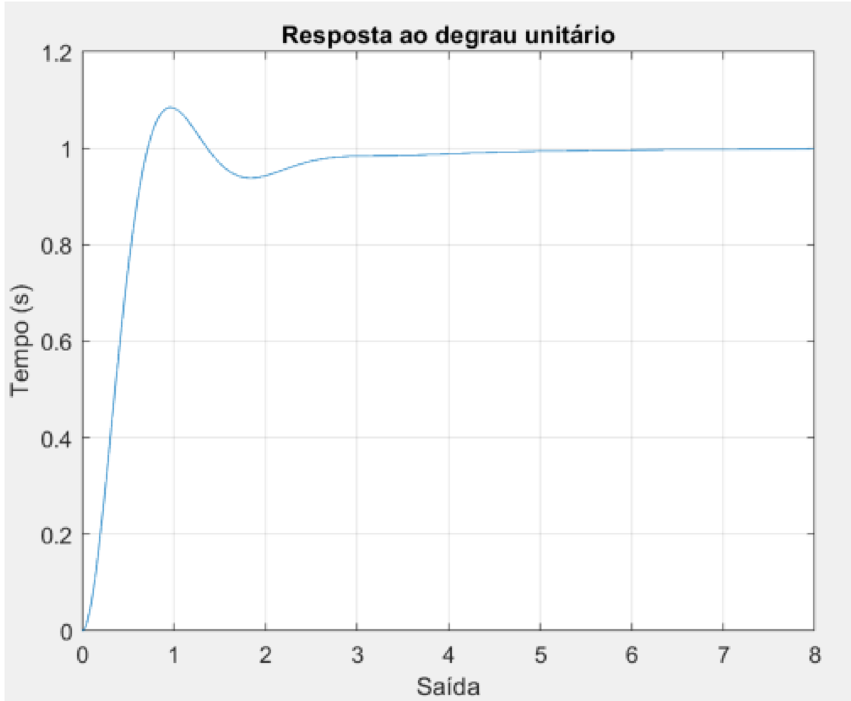

Figura 1. Resposta à entrada degrau unitária. Produzido no MatLab

```
solution =

    4.0000    0.7000
```

Figura 2. Parâmetros *K* e *a,* soluções da questão. Produzido no MatLab

2. B.10.12 utilizando o MatLab

O seguinte código MatLab foi utilizado:

```
clc;
clear all;

time_range = 0:0.01:8;

for K = 4:-0.05:2;
    for a = 3:-0.05:0.5;
        num = [0 0 1.2*K 2.4*K*a 1.2*K*a^2];
        den = [0.36 1.86 2.5+1.2*K 1+2.4*K*a 1.2*K*a^2];
        sys = tf(num,den);
        y = step(sys, time_range);
        max_y = max(y);
        s = 8;
        while y(s)>0.98 & y(s)<1.02;
            s = 1-1;
        end
```

```
        ts = (s-1)*0.01;

        if max_y < 1.08 & max_y > 1.03 & ts < 2.0
            break;
        end
    end
    if max_y < 1.08 & max_y > 1.03 & ts < 2.0
        break;
    end
end

solution = [K a max_y ts]

figure
plot(time_range, y)
grid()

title("Resposta ao degrau unitário")
ylabel("Tempo (s)")
xlabel("Saída")
```

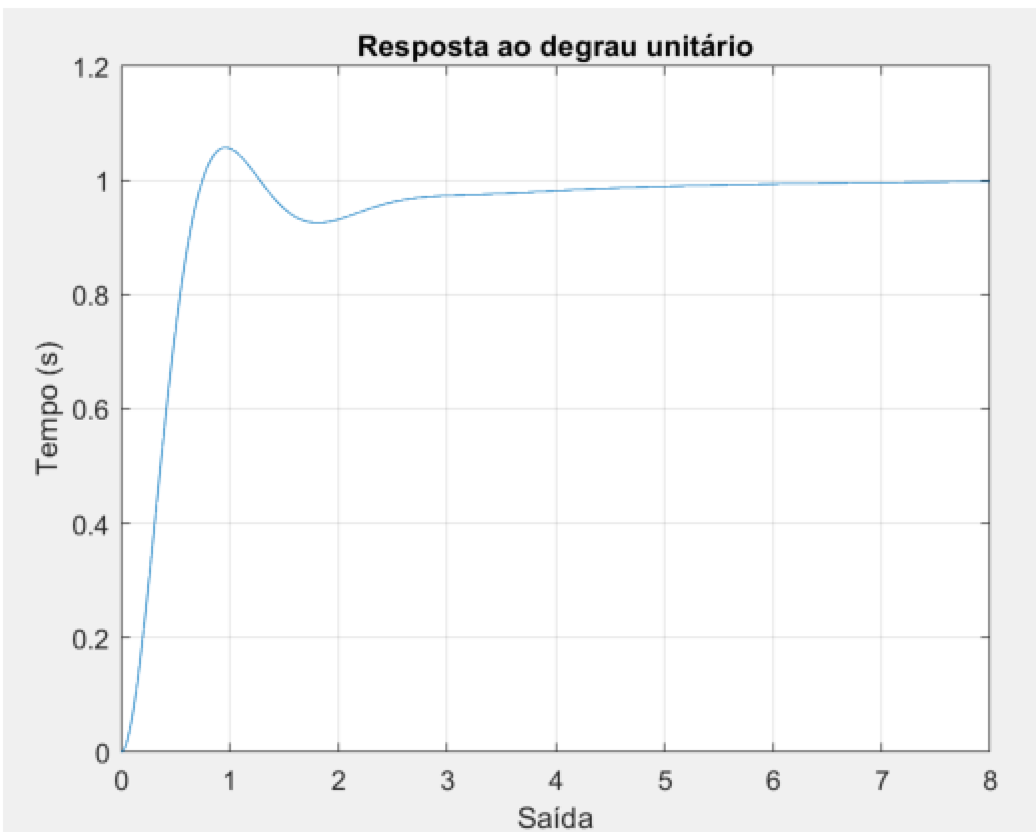

Figura 3. Resposta à entrada degrau unitária. Produzido no MatLab.

```
solution =

    4.0000    0.6500    1.0573    0.0700

>>
```

Figura 4. Solução para K, a, máximo do sistema e ts. Produzido no MatLab.

3. Refaça o exemplo **A.10.11**. Considere um controlador PID onde $G_c(s) = \frac{k(s+a)^2}{a}$ que deve controlar planta $G_p(s) = \frac{1}{s(s+1)(s+5)}$. Obtenha:

a) Um algoritmo que faça a busca automática dos ganhos. Determine os valores dos ganhos k e a de modo que o máximo sobressinal esteja dentro do intervalo $5\% < M < 10\%$ com tempo de acomodação $t_s < 3s$. Adote a região de busca $2 \leq k \leq 50$ *e* $0,05 \leq a \leq 2$.

Resposta: 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 obtido foi de 9.9% de sobressinal e 7.6s de tempo de acomodação, conforme demonstrado no resultado abaixo:

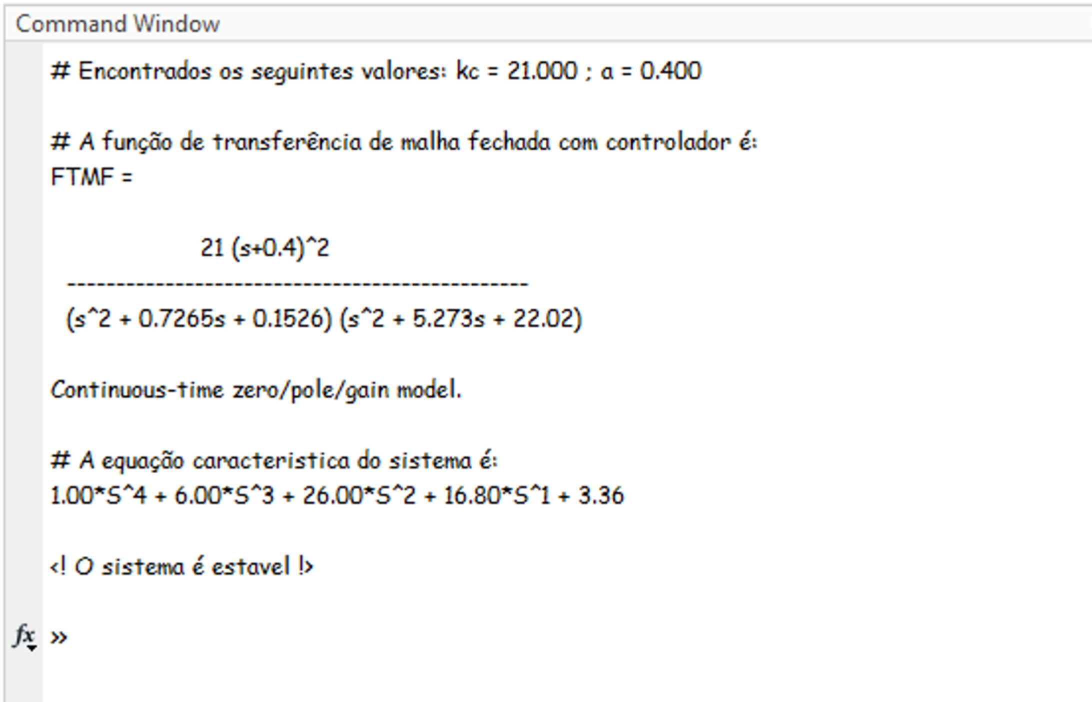

Figura 5. Resultados dos parâmetros do sistema. Produzido no MatLab.

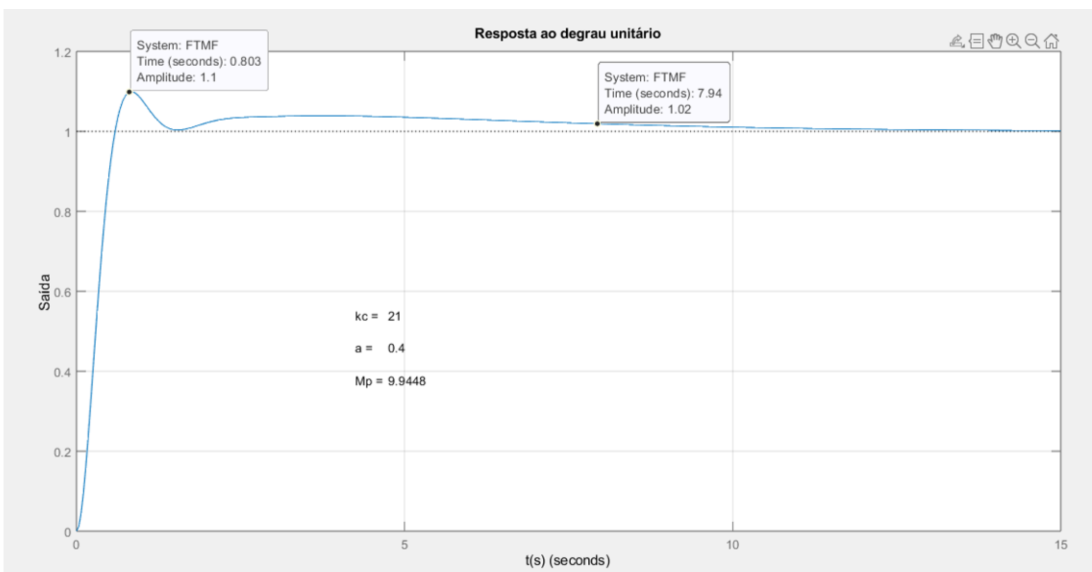

Figura 6. Resposta ao degrau unitário.

b) A função de transferência $G(s)$.

Resposta: A função de transferência de malha fechada com controlador foi obtida via script através (armazenada na variável FTMF) das seguintes linhas de código:

```
% entrada da FT pelos polos e zeros conhecidos
Gp = zpk([],[0 -1 -5], 1);

for kc = max_search_kc : -step_kc : min_search_kc
    for a = max_search_a : -step_a : min_search_a %
        Gc = kc*((S+a)^2)/S;
        G = Gc*Gp;
        FTMF = feedback(G,1);
```

```
FTMF =

               21 (s+0.4)^2
  ---------------------------------------------
  (s^2 + 0.7265s + 0.1526) (s^2 + 5.273s + 22.02)
```

Figura 7. Linhas de código MatLab para obtenção da FT global. Produzido no MatLab.

c) A equação característica:

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

```
[num, den] = tfdata(FTMF, 'v');
%y = step(num,den,t); %executa

    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
```

```
# A equação caracteristica do sistema é:
1.00*S^4 + 6.00*S^3 + 26.00*S^2 + 16.80*S^1 + 3.36
```

Figura 8. Equação característica da planta. Produzido no MatLab.

d) Determine o máximo valor de ganho para que a resposta a uma entrada degrau seja estável.

Resposta: Considerando $a = 0.4$, temos:

$$G_c(s) = k * \frac{(s + 0.4)^2}{s}, G_p(s) = \frac{1}{s(s+1)(s+5)}$$

$$G(s) = G_c(s) * G_p(s) \therefore G(s) = \frac{k(s + 0.4)^2}{s * s(s+1)(s+5)}$$
$$\therefore G(s) = \frac{k * (s^2 + s * 0.8\ + 0.16)}{s^4 + 6s^3 + 5s^2}$$

Fechando a malha, obtém-se:

$$G_{mf} = \frac{k(s^2 + s * 0.8 + 0.16)}{s^4 + 6s^3 + s^2(5 + k) + s * 0.8 * k + 0.16k}$$

Analisando a equação característica de MF e aplicando o critério de estabilidade de Routh-Hurwitz, teremos:

$$s^4 + 6s^3 + s^2(5 + k) + s * 0.8 * k + 0.16k$$

$$\begin{array}{cccc} s^4 & 1 & 5 + k & 0.16 * k \\ s^3 & 6 & 0.8 * k & 0 \\ s^2 & b_1 & b_2 & b_3 \\ s^1 & c_1 & c_2 & \\ s^0 & d_1 & & \end{array} \rightarrow \begin{cases} b_1 = 5 + 0.87k \\ b_2 = 0.16k \\ b_3 = 0 \\ c_1 = \dfrac{0.69k^2 + 3.04k}{5 + 0.81k} \\ c_2 = 0 \\ d_1 = 0.16k \end{cases}$$

Dado que b_1, c_1 e d_1 devem ser maiores que zero, logo:

$$\begin{aligned} &b_1 \rightarrow k > -5.74 \\ &c_1 \rightarrow -5.7 < k < -4.4 \\ &d_1 \rightarrow k > 0.8 \end{aligned}$$

Portanto, devido d_1, k deve ser maior que zero e o sistema é estável para qualquer valor de $k > 0$.