



## Homework II

### Exercise 1

20 Points

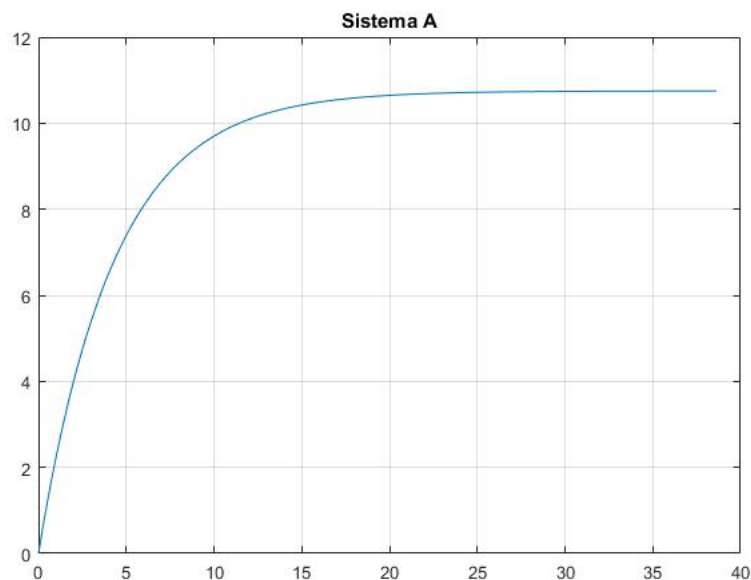
The unit steps responses of two systems  $A$  and  $B$  are recorded and reported in the files `HW2_ex1_dataA.txt` and `HW2_ex1_dataB.txt`, respectively. In each file, the first column gives the time vector  $t$  and the second column gives the output response  $y$ .

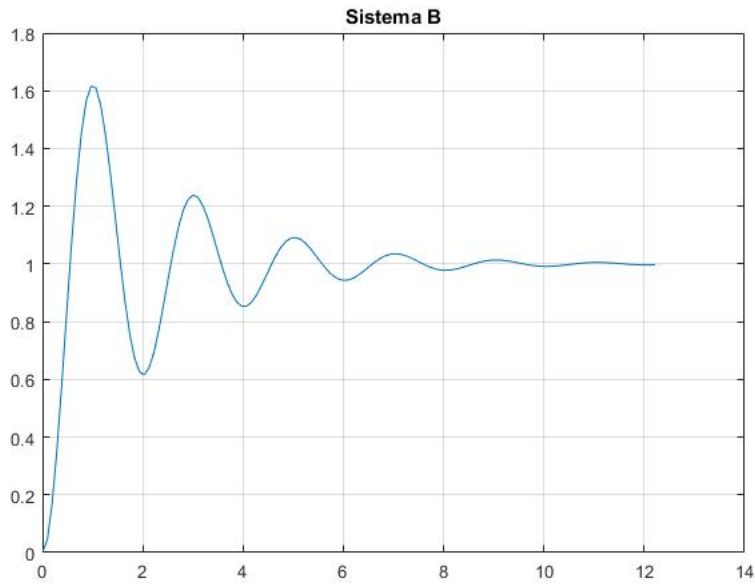
It is required to:

1. Load the data in Matlab/Octave and plot the two responses.
2. Estimate the transfer functions for the system  $A$  and  $B$ .
3. Compare your estimated systems with the ones provided in the data.

### Solution 1

1. Plotando os gráficos:





2. • Observando o gráfico do sistema A, concluímos que o mesmo é um sistema de primeira ordem. Portanto,  $G_a(s) = \frac{k}{s + a}$ .

Para calcularmos  $a$ , fazemos que  $T_s = \frac{4}{a}$ , onde  $T_s$  corresponde ao Tempo de estabilização, que é o tempo quando o sistema atinge 98% do valor final. Como o valor final é 10.7486, então o tempo quando seu valor for 10.5336 será o  $T_s$ . Olhando pelos dados fornecidos, temos então que  $16.8319 = \frac{4}{a} \rightarrow a = 0.2376$ .

Já o  $K$ , podemos calcular a partir do valor final, onde  $V_f = \frac{K}{a}$ . Como o Valor Final é 10.7486, teremos que

$$K = 0.2376 * 10.7486 \rightarrow K = 2.5543$$

$$\text{Sendo assim, } G_a(s) = \frac{K}{s + a} \rightarrow G_a(s) = \frac{2.5543}{s + 0.2376}$$

- Já no sistema B, podemos identificá-lo como um sistema de segunda ordem. Portanto,  $G_b(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ , onde  $\omega_n$  é a

frequência natural do sistema e  $\zeta$  é taxa de amortecimento. Para

calcular o  $\zeta$ , utilizamos a seguinte relação:  $\zeta = \frac{-\ln\left(\frac{\%OS}{100}\right)}{\sqrt{\ln^2\left(\frac{\%OS}{100}\right) + \pi^2}}$ ,

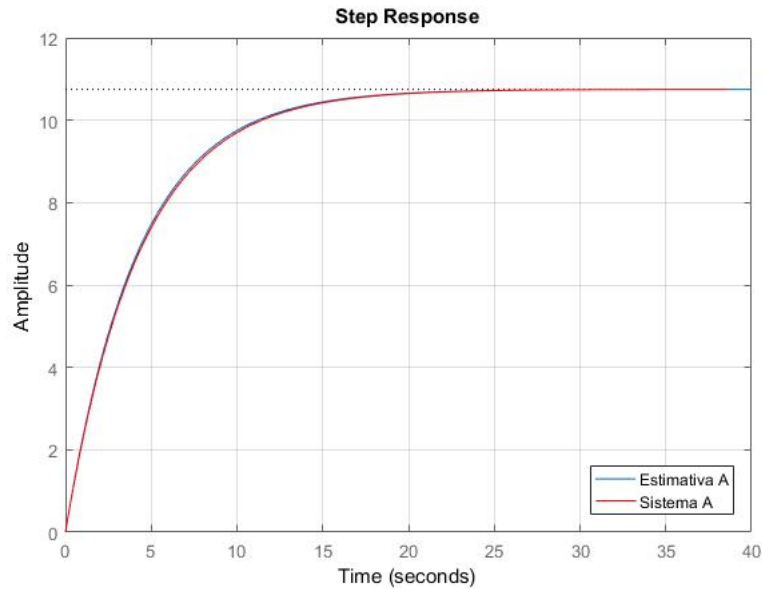
onde  $\%OS$  é a porcentagem de *overshoot*, que é quanto foi ultrapassado o valor final do sistema. Essa porcentagem é calculada através da razão  $\%OS = \frac{C_{\max} - C_{\text{final}}}{C_{\text{final}}} * 100$ . Com isso, temos que  $\zeta = 0.1498$ . Para calcularmos o  $\omega_n$ , utilizamos o Tempo de pico, que nada mais é do que o tempo no qual o valor máximo foi alcançado. Conseguimos identificar que o valor máximo foi 1.6165 e, verificando na tabela de dados, seu tempo de pico será 0.9695. En-

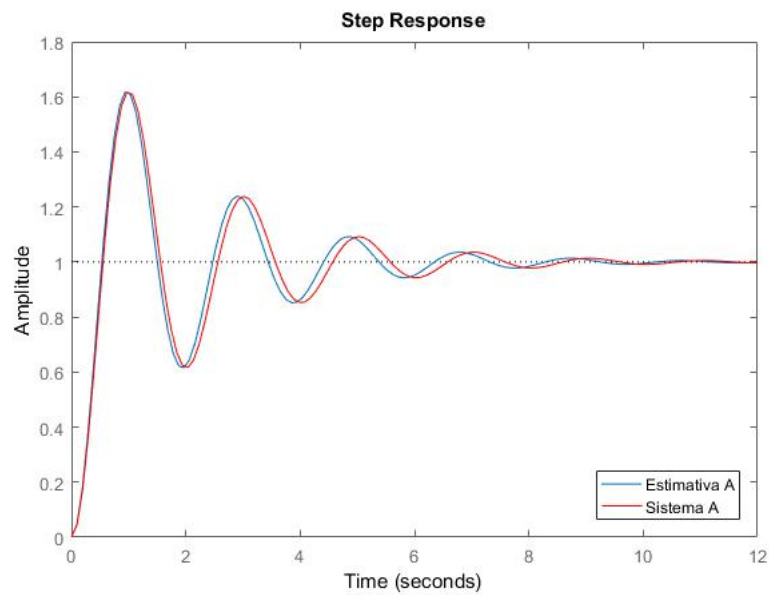
tão, através da relação  $\omega_n = \frac{\pi}{T_p \sqrt{1 - \zeta}}$  achamos que  $\omega_n = 3.2774$ .

Com isso, chegamos que  $G_b(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \rightarrow$

$$G_b(s) = \frac{10.7415}{s^2 + 0.9822s + 10.7415}$$

### 3. Gráficos comparativos:





```
%% Exercise 1
Clear all; clc;

[Xa,Ya] = textread('HW2_ex1_dataA.txt','%f%f');
[Xb,Yb] = textread('HW2_ex1_dataB.txt','%f%f');

%% 1.1
figure(1);
plot(Xa,Ya)
grid on
title('Sistema_A')

figure(2);
plot(Xb,Yb)
grid on
title('Sistema_B')

%% 1.3
a = 4/Xa(86)
k = a*Ya(196)
numa = [ k ]
dena = [ 1 a]
Ga = tf(numa,dena)
figure(3)
step(Ga); hold on;
plot(Xa,Ya,'color','r')
grid on;
legend('Estimativa_A','Sistema_A','Location','southeast')

OS = (max(Yb)-Yb(127))/Yb(127)*100
```

```

damp = -log(OS/100)/sqrt((log(OS/100)^2)+pi^2)
wn = pi/(Xb(11)*sqrt(1-damp^2))
numb = [ wn^2 ]
denb = [ 1 2*damp*wn wn^2]
Gb = tf(numb, denb)
figure(4)
step(Gb); hold on;
plot(Xb,Yb,'color','r')
legend('Estimativa_A','Sistema_A','Location','southeast')

```

## Exercise 2

25 Points

Find the transfer function,  $T(s) = C(s)/R(s)$ , for the system in Figure (1), using the following methods:

1. Block diagram reduction.
2. Matlab/Octave. Use the following transfer functions:

$$G1(s) = \frac{1}{(s+7)}, \quad G2(s) = \frac{1}{(s^2+2s+3)},$$

$$G3(s) = \frac{1}{(s+4)}, \quad G4(s) = \frac{1}{s},$$

$$G5(s) = \frac{5}{(s+7)}, \quad G6(s) = \frac{1}{(s^2+5s+10)},$$

$$G7(s) = \frac{3}{(s+2)}, \quad G8(s) = \frac{1}{(s+6)}.$$

**Hint:** Use the `append` and `connect` commands in Matlab/Octave Control System Toolbox/Package.

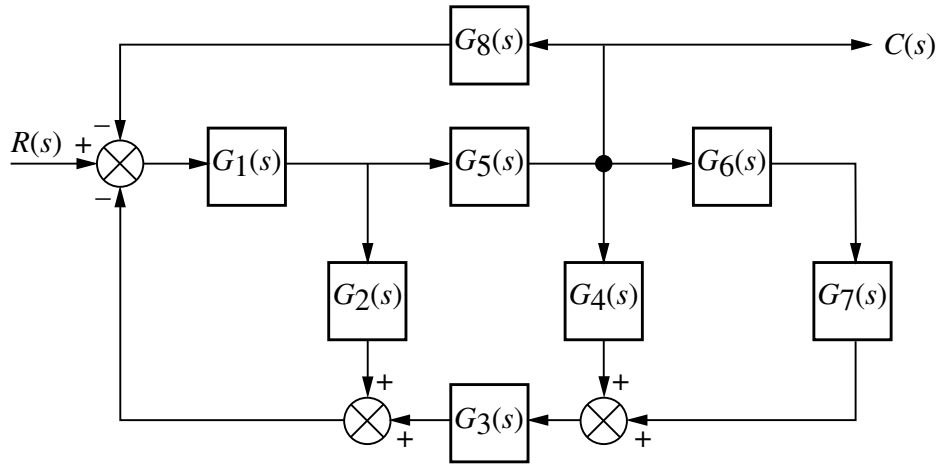


Figura 1: Block diagram for Exercise 2.

## Solution 2

1. Após utilizar as propriedades de soma e concatenação de blocos chegamos no bloco com a função de transferência

$$G(s) = \frac{G_1(s)G_5(s)}{1 + G_1(s)[[G_2(s) + G_5(s)G_3(s)[G_4(s) + G_6(s)G_7(s)]] + G_8(s)G_5(s)]}$$

- 2.

```
s = tf('s');
G1= 1/(s+7); G2= 1/(s^2+2*s+3);
G3= 1/(s+4); G4= 1/s;
G5= 5/(s+7); G6= 1/(s^2+5*s+10);
G7= 3/(s+2); G8= 1/(s+6);

Gm = (G1*G5)/(1+G1*(G2+G3*G5*(G4+G6*G7)+G5*G8))

Sum1 = sumblk('b_u=r_u-a_u-d');
Sum2 = sumblk('d_u=f_u+g');
Sum3 = sumblk('h_u=i_u+j');

G1.u = 'b'; G1.y = 'e';
G2.u = 'e'; G2.y = 'f';
G3.u = 'h'; G3.y = 'g';
G4.u = 'c'; G4.y = 'i';
G5.u = 'e'; G5.y = 'c';
G6.u = 'c'; G6.y = 'k';
G7.u = 'k'; G7.y = 'j';
G8.u = 'c'; G8.y = 'a';

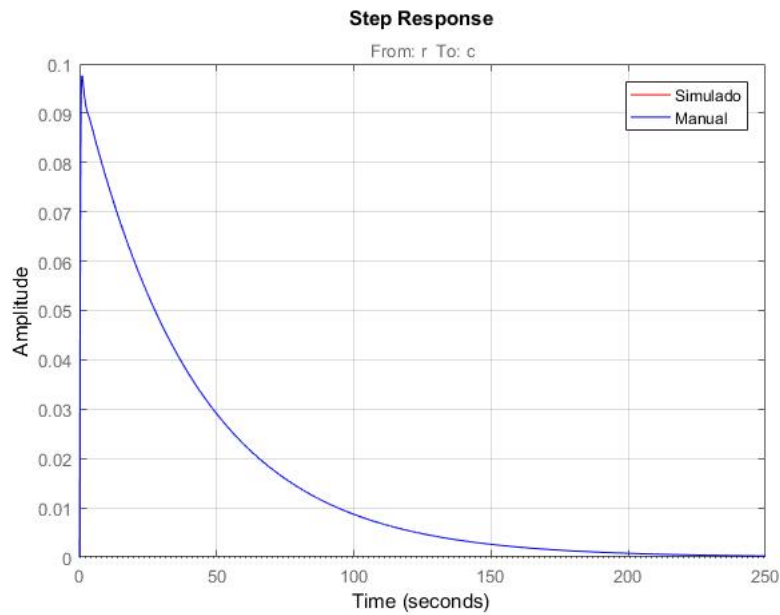
Ss = connect(G1,G2,G3,G4,G5,G6,G7,G8,Sum1,Sum2,
             Sum3,'r','c');

[num, den] = ss2tf(Ss.A,Ss.B,Ss.C,Ss.D);
```

```

Gs = tf(num,den)
figure(1)
step(Gs,'r',Gm,'b'); grid on;
legend('Simulado','Manual');

```



P.S 1: Simulado refere-se ao calculado no MATLAB, enquanto manual refere-se ao feito por diagrama de blocos.

P.S 2: O gráfico do simulado não aparece visível pois a linha do manual está sobrepondo totalmente ela. Contudo, através do MATLAB, pode ser verificado a semelhança entre as funções de transferência.

P.S 3: Os valores concretos das funções de transferência foram omitidos devido ao seu tamanho exageradamente grande. Contudo, as variáveis Gs e Gm indicam as funções de transferência feita no MATLAB e através do diagrama de blocos, respectivamente.

### Exercise 3

20 Points

The system in state space given in (1) represents the forward path of a unity feedback system.

$$\begin{aligned}
 \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -4 & -5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} u \\
 y &= \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \mathbf{x}
 \end{aligned} \tag{1}$$

Constructing the Routh table in Matlab/Octave, use the Routh-Hurwitz criterion to determine if the closed loop is stable.

### Solution 3

### Exercise 4

35 Points

For a system with the state and output equations given in (2):

$$\begin{aligned}\dot{x}(t) &= x^2(t) - u(t)x(t) - 2u(t) \\ y(t) &= x^3(t) + u^3(t)\end{aligned}\tag{2}$$

1. Calculate the state and output equilibrium points when  $u(t) = u_{eq} = 1$ .
2. Define the linearised system around the equilibrium points and analyse the stability of the system.
3. Compare the linearised and nonlinear system responses  $\delta y(t)$  for an input  $\delta u(t) = A \cos(2t)$ , with  $A = 0.1$  applied at  $t = 0$ .
4. **Extra (5 points):** Investigate how the responses differ when the input amplitude assumes the values  $[0.05, 0.15, 0.2]$ .
5. **Extra (5 points):** Calculate the error between linearised and nonlinear system in point (3) and point (4). Comment your findings.

**Hint:** Use the command `ode45` to simulate the nonlinear system.