LISTA D

GABRIEL PINHEIRO **10336595**

PME 3380

Dr.Agenor T. Fleury
Dr. Décio C. Donha

06/09/2020

Exercícios:

1. Utilizando o programa *scilab* foram elaborados códigos para análise da solução numérica dos sistemas linearizados e não linearizados do problema do enchimento de um reservatório.
   A seguir temos os resultados obtidos:

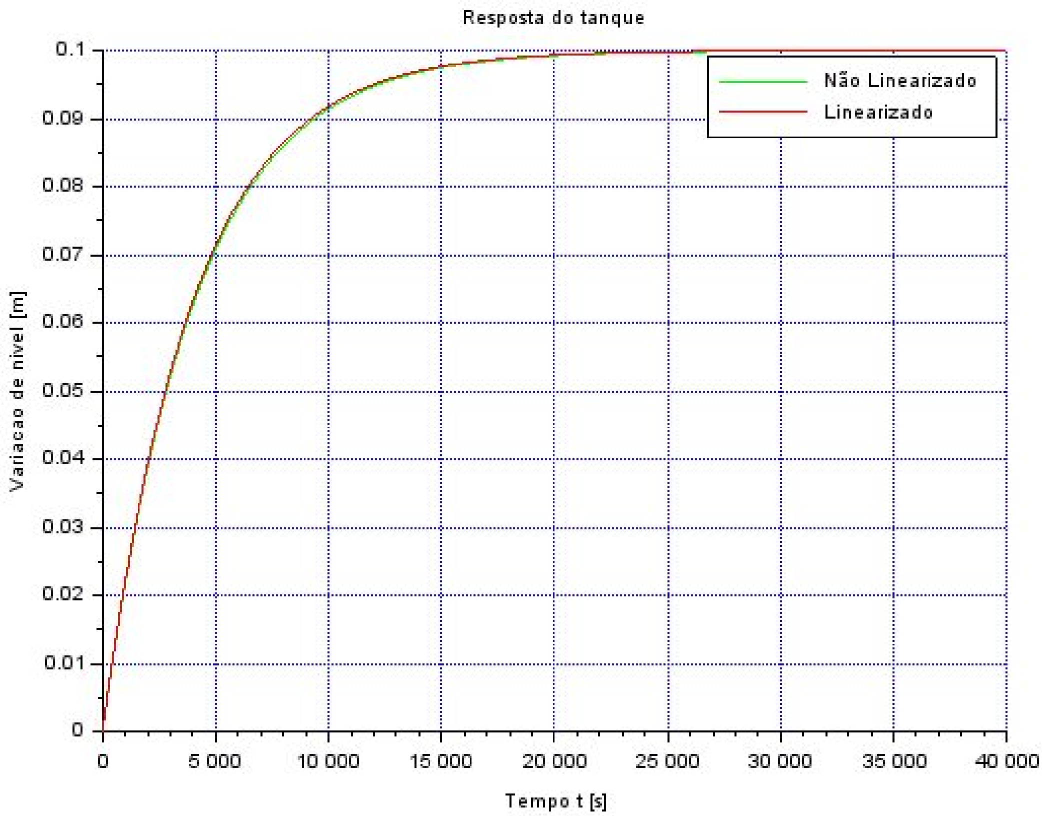

Figura 3 - Simulação 1 reservatórios

Como podemos ver, a solução linearizada possui apenas um pequeno desvio se comparada a solução não linearizada. Isso indica que o sistema linearizado é capaz de descrever o sistema não linearizado de forma bastante precisa.

2. A partir do método das malhas, podemos encontrar a seguinte equação que descreve o circuito:

$$Ri = \frac{-1}{c}\int(i\partial t) + e_i - e_o \tag{1}$$

Com a equação (1), podemos criar uma associação com a equação linearizada de um reservatório da seguinte forma:

$$R = 1 \tag{2}$$

$$c = 2S\sqrt{\frac{R\,h_o}{\rho\,g}} \tag{3}$$

$$e_i = Q_{ei}/S \tag{4}$$

$$e_o = 0 \tag{5}$$

$$i = dh \tag{6}$$

Com essas relações, podemos encontrar a seguinte simulação:

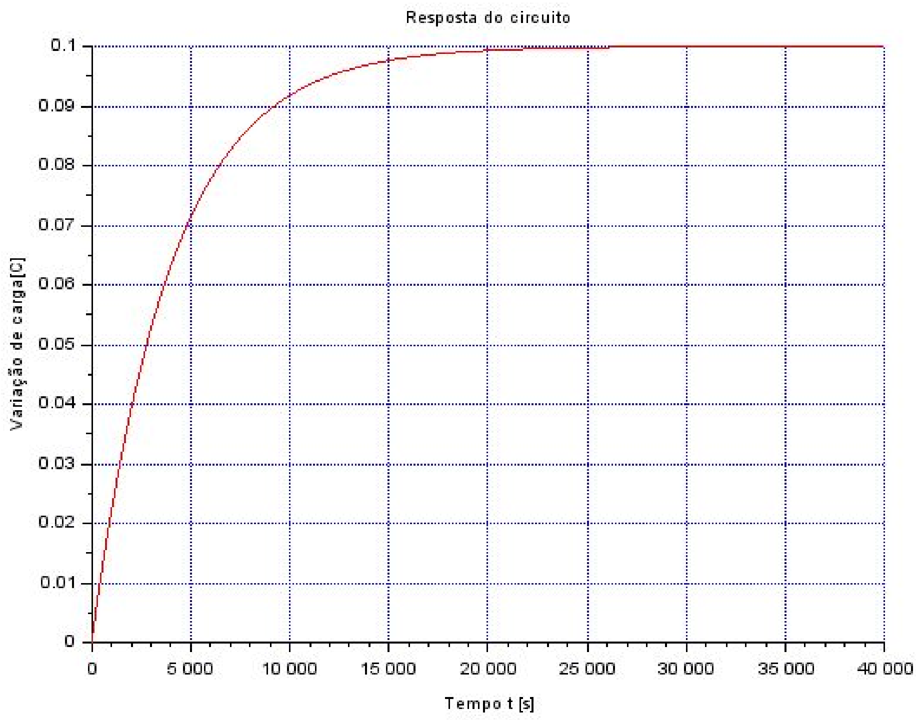

Figura 2 - Simulação do circuito elétrico

Esta simulação nos mostra que através do circuito elétrico mostrado abaixo, somos capazes de descrever o sistema linearizado do reservatório.

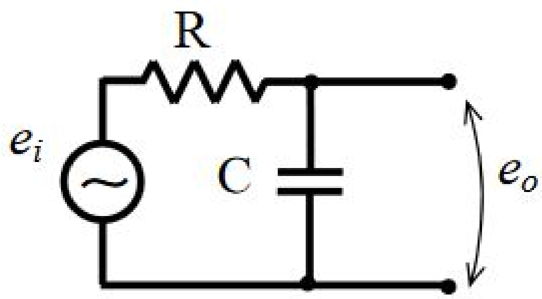

Figura 3 - Esquema do circuito elétrico

Lição de Casa:

Utilizando a linguagem de programação *Python* somada à biblioteca *Scipy*, somos capazes de resolver o sistema linearizado de 2 reservatórios encontrado na última lista.

Recordando:

$$A_1 = \frac{-1}{2S_1}\sqrt{\frac{\rho g}{R_a(h_{1o} - h_{2o})}} \tag{7}$$

$$A_2 = \frac{1}{2S_1}\sqrt{\frac{\rho g}{R_a(h_{1o} - h_{2o})}} \tag{8}$$

$$A_3 = \frac{1}{2S_2}\sqrt{\frac{\rho g}{R_a(h_{1o} - h_{2o})}} \tag{9}$$

$$A_4 = \frac{-1}{2S_2}\left(\sqrt{\frac{\rho g}{R_a(h_{1o} - h_{2o})}} + \sqrt{\frac{\rho g}{R_s h_{2o}}}\right) \tag{10}$$

$$\dot{x} = Ax + Bu \tag{11}$$

$$y = Cx + Du \tag{12}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{S_1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{14}$$

Com auxílio das funções "*signal.StateSpace*" e "*signal.step*", chegamos no seguinte gráfico:

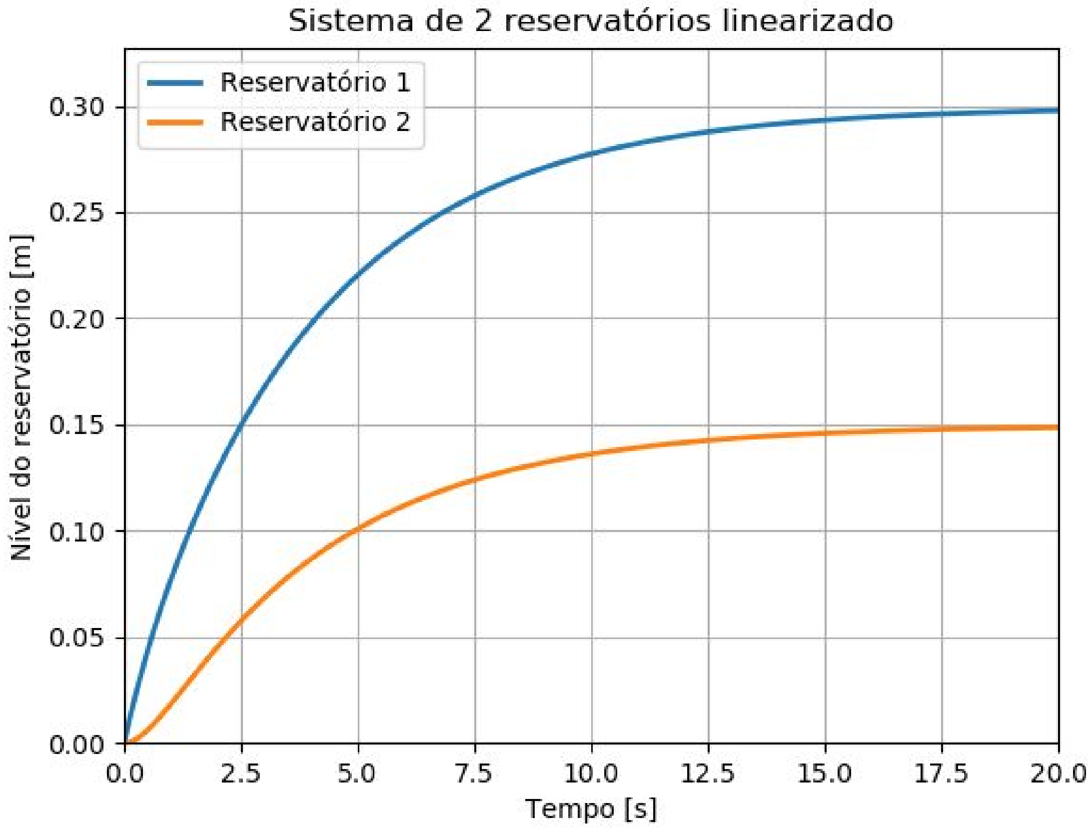

Figura 4 - Simulação 2 reservatórios

Esta simulação se assemelha bastante com a que obtivemos na última lista, corroborando com a aplicabilidade de linearização para este problema.
De forma análoga ao modelo de um reservatório, podemos utilizar o seguinte circuito elétrico para descrever o sistema linearizado para o de dois reservatórios:

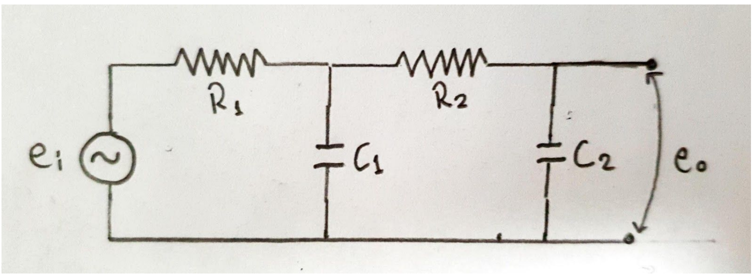

Figura 5 - Circuito elétrico para dois reservatórios

Códigos:

Algoritmo do “Exercício 1”:

```
clear

exec("C:\Users\gabri\Desktop\Modelagem\Lista_C\reservatorio.sci")

S=10;
rho=1000;
g=10;
R=2*10^8;
ho=2;
hi=0.1;
Qei=sqrt(rho*g*(ho+hi)/R);

h0=2;

t=0:10:40000;

h=ode(h0,t(1),t,list(tanque,entrada));
for i=1:length(h)
   h(i)=h(i)-ho;
end

S=10;
rho=1000;
g=10;
R=2*10^8;
ho=2;
hi=0.1;
Qei=(1/2)*sqrt(rho*g/(R*ho))*hi;

A=(-1/(2*S))*sqrt(rho*g/(R*ho));
B=1/S;
C=1;
D=0;
tanque=syslin('c',A,B,C,D);

x0=0;
t=0:10:40000;
u=Qei*ones(t);
[y,x]=csim(u,t,tanque,x0);
```

```
scf(0);
plot2d(t,h,3)
plot2d(t,y,5)
xtitle("Resposta do tanque","Tempo t [s]","Variacao de nivel [m]");
xgrid(2)
hl=legend(['Não Linearizado';'Linearizado']);
xs2png(gcf(),'plot.png');
```

Algoritmo do “Exercício 2”:

```
clear

c=2*10*sqrt(2*10^8*2/(1000*10));

A=-1/c;
B=1;
C=1;
D=0;
tanque=syslin('c',A,B,C,D);

x0=0;
t=0:10:40000;
u=((1/2)*sqrt(1000*10/(2*10^8*2))*0.1/10)*ones(t);
[y,x]=csim(u,t,tanque,x0);

plot2d(t,y,5)
xtitle("Resposta do circuito","Tempo t [s]","Variação de carga[C]");
xgrid(2)
xs2png(gcf(),'plot.png');
```

Algoritmo da “Lição de casa”:

```
#Import das bibliotécas que serão utilizadas no código
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
from math import sqrt

#Criação das variáveis do problema
S1=10
S2=10
```

```python
rho=1000
g=10
Ra=2*10^8
Rs=2*10^8
h1o=4
h2o=2
hi=0.1
Qei=(1/2)*sqrt(rho*g/(Ra*h1o))*hi

#Declaração dos valores que entrarão nas células da Matriz A
A1=(-1/(2*S1))*sqrt(rho*g/(Ra*(h1o-h2o)))
A2=(1/(2*S1))*sqrt(rho*g/(Ra*(h1o-h2o)))
A3=(1/(2*S2))*sqrt(rho*g/(Ra*(h1o-h2o)))
A4=(-1/(2*S2))*(sqrt(rho*g/(Ra*(h1o-h2o)))+sqrt(rho*g/(Rs*h2o)))

#Criação das Matrizes A, B, C e D
A = [[A1,A2], [A3,A4]]
B = [[1/S1], [0]]
#Devemos criar duas matrizes C, para assim podermos coletar as diferentes
#células do vetor y que será criado mais a frente
C1 = [1,0]
C2 = [0,1]
#As entradas u não causam impacto na sáida y
D = 0

#Criamos o vetor para o tempo
t=np.linspace(0,20,40001)

#Utilizamos as funções signal.StateSpace e signal.step, da bibliotéca Scipy, para simular o
sistema
sys1 = signal.StateSpace(A, B, C1, D)
t1, y1 = signal.step(sys1,T=t)
sys2 = signal.StateSpace(A, B, C2, D)
t2, y2 = signal.step(sys2,T=t)

#Em seguida, utilizamos a bibliotéca matplotlib.pyplot, para plotar nossa simulação
plt.figure()
plt.plot(t1,y1,linewidth=2,label='Reservatório 1')
plt.plot(t2,y2,linewidth=2,label='Reservatório 2')
plt.title('Sistema de 2 reservatórios linearizado')
plt.xlabel('Tempo [s]')
plt.ylabel('Nível do reservatório [m]')
plt.legend(loc='best')
plt.xlim([0,max(t1)])
```

```
plt.ylim([0,1.1*max(y1)])
plt.grid()
plt.savefig('2reservatórios.png')
plt.show()
```
