PMR3404 Exercício 14

Eduardo Eiras de Carvalho, 9288209

July 9, 2020

Espaço de Estados

Considere o sistema:

$$\dot{x}(t) = Ax(t) + Bu(t)y(t) = Cx(t) + Du(t)$$

No qual

$$A = \begin{pmatrix} 0.5 & 1 \\ 1 & 0 \end{pmatrix} ; B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} ; C = \begin{pmatrix} 0 & 1 \end{pmatrix} ; D = 0$$

Projete um controlador por realimentação de estados que garanta que a dinâmica em malha fechada apresente autovalores [-0.3;-0.7] e que garanta que a saída $y(t) = x_2(t)$ siga um valor de referência r para a entrada degrau.

Resolução

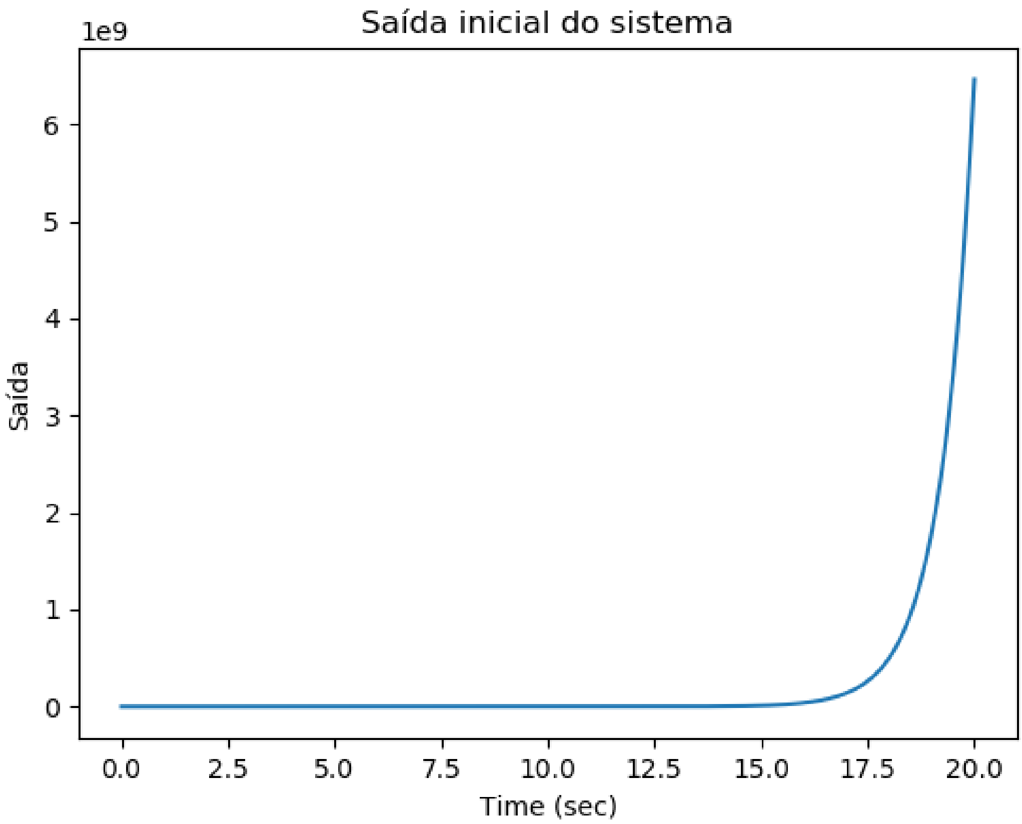
Os autovalores em malha fechada são os polos dominantes do sistema, assim, queremos que nosso sistema sigam a dinâmica de um sistema com os polos [-0.3;-0.7].
O segundo requisito é que a saída siga um valor de referência para a entrada degrau, ou seja, queremos que nosso sistema tenha erro em regime para uma entrada degrau.

Passos a serem seguidos

- Verificar se o sistema é controlável
- Posicionar os polos de requisito
- Encontrar os autovalores da matriz A em malha fechada, ou seja: $(A - B \cdot K)$
- Fazer o sistema em malha fechada $sys(A - B \cdot K, K_r \cdot B, C - D \cdot K, K_r \cdot D)$ com $K_r = 1$
- Verificar o ganho em malha fechada
- Se não for nulo, refazer com $K_r = \frac{1}{ganho}$, sendo *ganho* o ganho em malha fechada obtido

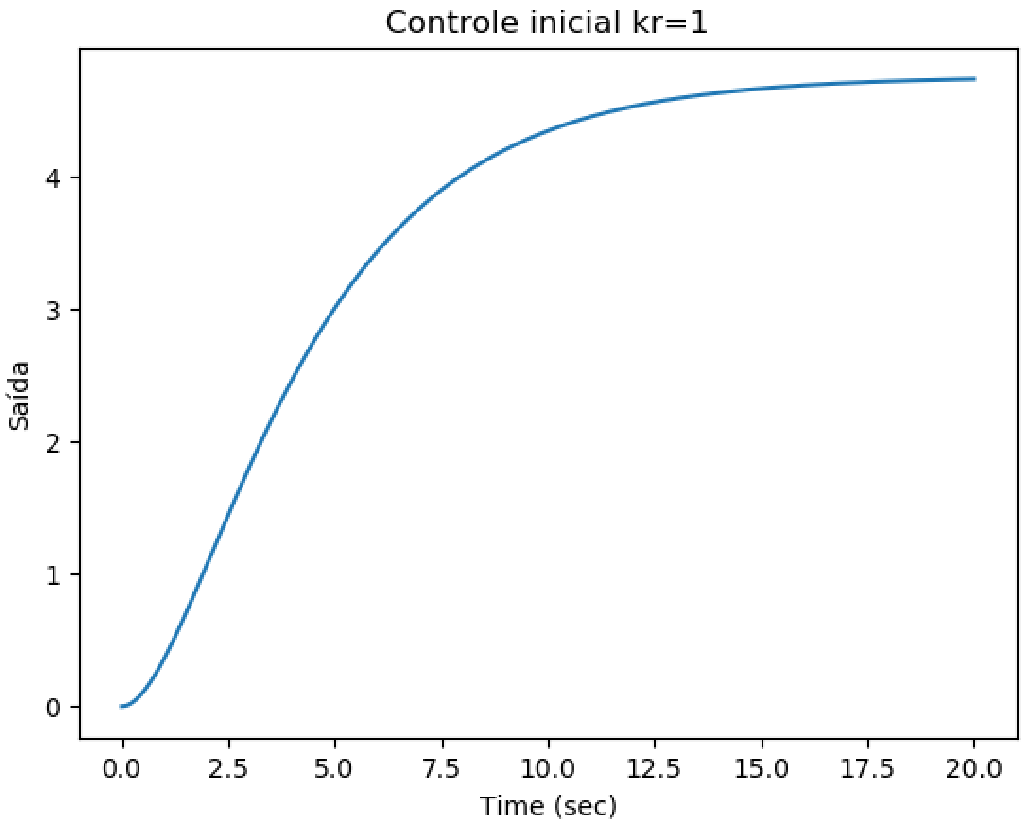
Resolução em Python

```
import numpy as np
import cmath as cm
import control as co_general
import matplotlib.pyplot as plt
from control.matlab import *

A = [[0.5,1],
     [1, 0]]

B = [[1],
     [0]]

C = [0, 1]

D = 0

sys = ss(A,B,C,D)

#simulacao com condicoes iniciais nao nulas
X0 = [0.1,0]
n=1000
T = np.linspace(0,20,n)
yout,t,xout = initial(sys,T,X0,return_x=True)
plt.figure()
plt.plot(t,yout)
plt.title('Saída inicial do sistema')
plt.xlabel('Time (sec)')
plt.ylabel('Saída')

#Controlabilidade
CT = ctrb(A,B)
print("A controlabilidade do sistema é:", np.linalg.matrix_rank(CT))
"Como rank(CT) é igual ao número de estados, o sistema é controlável"
```



Podemos ver que o sistema simulado com condições iniciais não nulas vai para o "infinito", ou seja, é instável.
Além disso o sistema é controlável pois a controlabilidade é 2 que é igual ao número de estados.
Vamos compensar o sistema:

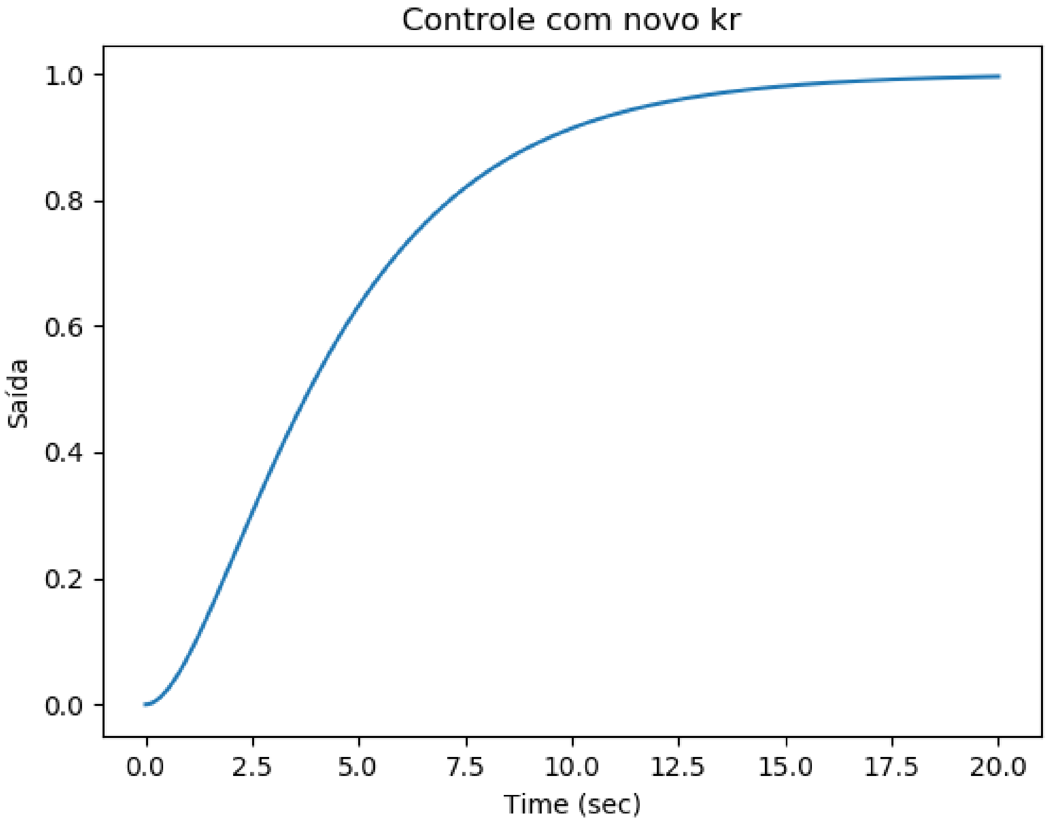
```
pk = [-0.3,-0.7] #Polos do requisito
K = place(A,B,pk) #Matriz com os polos desejados
kr = 1
sys_mf = ss(A-B*K , kr*B , C-D*K , kr*D) #Sistema compensado

#Verificar saída do sistema

yout, y = step(sys_mf, T)
plt.figure()
plt.plot(t,yout)
plt.title('Controle inicial kr=1')
plt.xlabel('Time (sec)')
plt.ylabel('Saída')

"Sistema ainda não atingiu erro nulo em regime"
```



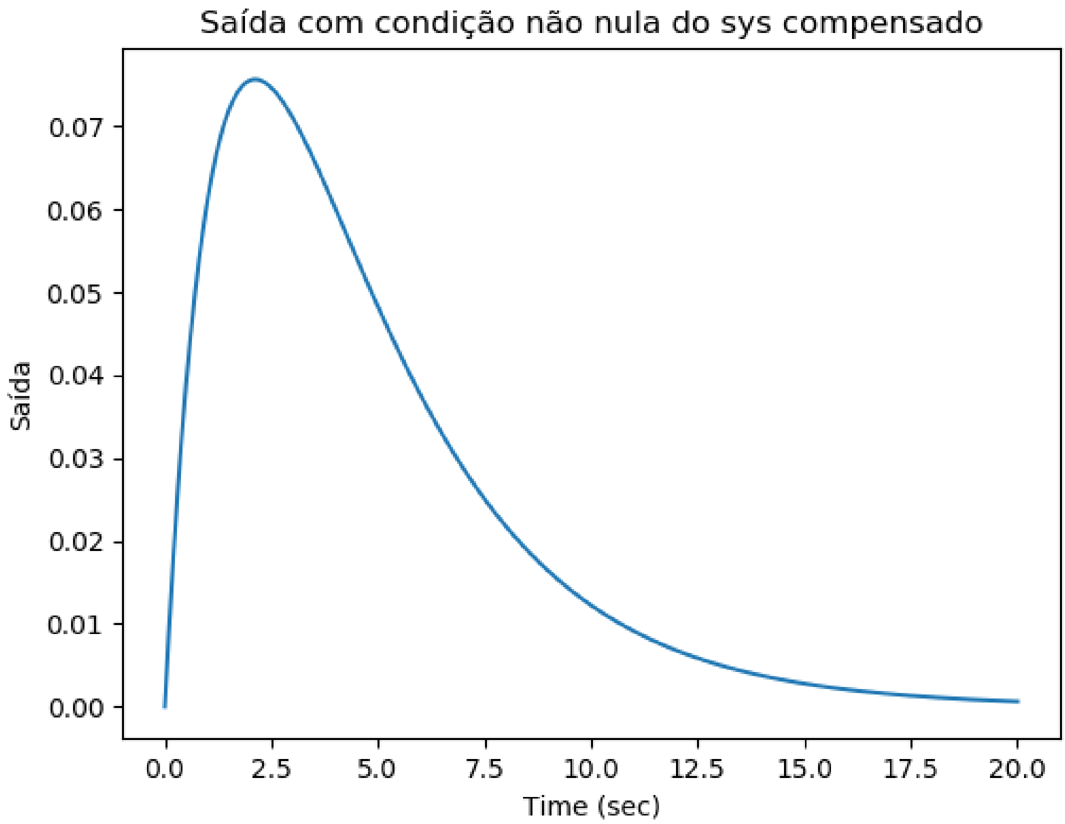
Vemos que o sistema está estável mas não atingiu erro nulo, assim devemos utilizar um novo K_r:

```
#Utilizar outro kr

Kr = 1/dcgain(sys_mf)
print("kr = ", Kr)
sys_mf = ss(A-B*K,Kr*np.asarray(B),C,D)
print("Novo sistema:")
print(sys_mf)

yout, y = step(sys_mf, T)
plt.figure()
plt.plot(t,yout)
plt.title('Controle com novo kr')
plt.xlabel('Time (sec)')
plt.ylabel('Saída')

yout,t,xout = initial(sys_mf,T,X0,return_x=True)
plt.figure()
plt.plot(t,yout)
plt.title('Saída com condição não nula do sys compensado')
plt.xlabel('Time (sec)')
```

```
plt.ylabel('Saída')
```



O sistema agora atingiu os requisitos de projeto e vemos que se for simulado com condições iniciais não nulas ele volta a condição inicial:



O sistema final compensado é:

$$A = \begin{pmatrix} -1 & -0.21 \\ 1 & 0 \end{pmatrix} ; B = \begin{pmatrix} 0.21 \\ 0 \end{pmatrix} ; C = \begin{pmatrix} 0 & 1 \end{pmatrix} ; D = 0$$

Que possui os polos de requisito:

```
In [32]: damp(sys_mf)
_____Eigenvalue______ Damping___ Frequency_
      -0.7                     1         0.7
      -0.3                     1         0.3
```