Primeiro Relatorio PCO119

João Vitor Yukio Bordin Yamashita

September 2, 2022

1 Exercício 1

1. É um sistema de terceira ordem e instável.
2. Transformada inversa de:
$$Y(s) = \frac{5}{s^3 + 3s^2 + 4s}$$
Vou fazer duas abordagens, uma resolvendo usando trabalho 'braçal' e outra numericamente. Para a solução 'braçal', temos:
$$Y(s) = \frac{5}{s^3 + 3s^2 + 4s} = \frac{5}{s(s^2 + 3s + 4)} = \frac{A}{s} + \frac{Bs + C}{(s^2 + 3s + 4)} \tag{1}$$
Isolando A:
$$\frac{A}{s} = \frac{5}{s^3 + 3s^2 + 4s} - \frac{Bs + C}{(s^2 + 3s + 4)} = \frac{Bs^2 + Cs + 5}{s(s^2 + 3s + 4)}$$
$$A = \frac{s(Bs^2 + Cs + 5)}{s(s^2 + 3s + 4)} = \frac{Bs^2 + Cs + 5}{s^2 + 3s + 4}$$
Avaliando para s = 0:
$$A = \frac{5}{4}$$
Substituindo A em 1:
$$Y(s) = \frac{5}{4}\frac{1}{s} + \frac{Bs + C}{s^2 + 3s + 4} = \frac{5}{s(s^2 + 3s + 4)}$$
Isolando B e C:
$$\frac{Bs + C}{s^2 + 3s + 4} = \frac{5}{s(s^2 + 3s + 4)} - \frac{5}{4}\frac{1}{s}$$
$$\frac{Bs + C}{s^2 + 3s + 4} = \frac{5 - \frac{5}{4}(s^2 + 3s + 4)}{s(s^2 + 3s + 4)}$$
$$Bs + C = \frac{(5 - \frac{5}{4}(s^2 + 3s + 4))(s^2 + 3s + 4)}{s(s^2 + 3s + 4)}$$
$$Bs + C = \frac{5 - \frac{5}{4}(s^2 + 3s + 4)}{s}$$
$$(Bs + C)s = 5 - \frac{5}{4}s^2 - \frac{15}{4}s - 5 \tag{2}$$
A partir da eq. 2, temos:
$$\begin{cases} Bs^2 = -\frac{5}{4}s^2 \\ Cs = -\frac{15}{4}s \end{cases} \tag{3}$$
Com 3 temos:
$$\begin{cases} B = -\frac{5}{4} \\ C = -\frac{15}{4} \end{cases} \tag{4}$$

Substituindo 4 em 1, temos:

$$Y(s) = \frac{5}{4}\frac{1}{s} + \frac{-\frac{5}{4}s + -\frac{15}{4}}{s^2 + 3s + 4}$$

$$Y(s) = \frac{5}{4}\frac{1}{s} + -\frac{5}{4}\frac{s+3}{s^2 + 3s + 4} \tag{5}$$

Resolvendo:

$$T(s) = \frac{s+3}{s^2 + 3s + 4} \tag{6}$$

Temos que:

$$\frac{s+3}{s^2+3s+4} = \frac{s+3}{s^2+3s+(\frac{3}{2})^2+\frac{7}{4}} = \frac{s+\frac{3}{2}+\frac{3}{2}}{(s+\frac{3}{2})^2+\frac{7}{4}}$$

$$\frac{s+\frac{3}{2}+\frac{3}{2}}{(s+\frac{3}{2})^2+\frac{7}{4}} = \frac{s+\frac{3}{2}}{(s+\frac{3}{2})^2+\frac{7}{4}} + \frac{\frac{3}{2}}{(s+\frac{3}{2})^2+\frac{7}{4}}$$

Usando a tabela temos:

$$\mathcal{L}^-1\{T(s)\} = \frac{5}{4} - \frac{5}{4}(\exp{(-\frac{3}{2}t)}\cos{(\frac{\sqrt{7}}{2}t)} + \frac{3}{\sqrt{7}}\exp{(-\frac{3}{2}t)}\sin{(\frac{\sqrt{7}}{2}t)})$$

As imagens das contas podem ser encontradas no anexo.

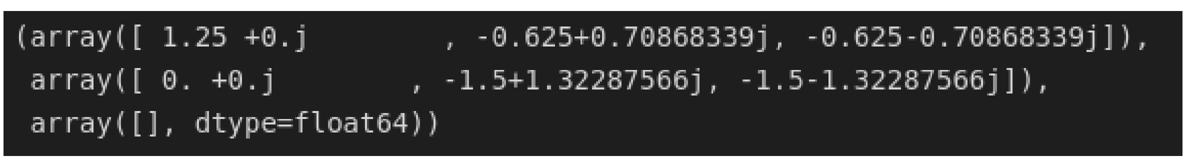
Usando o método numérico:

Primeiramente calculamos os resíduos:

```
import control
import scipy
from scipy import signal

num, den = [5],[1,3,4,0]
Ys = control.TransferFunction(num, den)
res1 = signal.residue(num, den)
res1
```

Listing 1: Cálculo dos residuos



```
(array([ 1.25 +0.j         , -0.625+0.70868339j, -0.625-0.70868339j]),
 array([ 0. +0.j         , -1.5+1.32287566j, -1.5-1.32287566j]),
 array([], dtype=float64))
```

Figure 1: Resíduos

Com os resíduos usamos sympy para ter uma visualização melhor da equação:

```
import sympy
from sympy import Symbol

t = Symbol('t')
yt = Symbol('yt')
e = Symbol('e')


yt = res1[0][0]*e**(res1[1][0]*t) + res1[0][1]*e**(res1[1][1]*t) + res1[0][2]*
e**(res1[1][2]*t)
yt
```

Listing 2: Representação da função no tempo

$$e^{t(-1.5-1.3228756555323i)}\left(-0.625-0.708683386892301i\right)+e^{t(-1.5+1.3228756555323i)}\left(-0.625+0.708683386892301i\right)+1.25$$

Figure 2: Saída para representação da função

Para confirmar o resultado vamos realizar três simulações, usando os três métodos, considerando a resposta ao impulso como entrada:

```python
import numpy as np
import matplotlib.pyplot as plt

yout , T = control.impulse_response(Ys)
T1 = np.arange(0, 25, 0.01)
res = []

for t1 in T1:
    yt1 = res1[0][0]*cmath.exp(res1[1][0]*t1) + res1[0][1]*cmath.exp(res1[1][1]*t1) + res1[0][2]*cmath.exp(res1[1][2]*t1)
    res.append(yt1)

def createVal(T = None):
    import math
    from math import e, cos, sin, sqrt
    res = []
    for t in T:
        yt = (5/4) - ((5/4)*((e**(-3/2*t)*cos((sqrt(7)/2)*t))+((3/(sqrt(7))*e**((-3/2)*t)*sin((sqrt(7)/2)*t)))))
        res.append(yt)
return res

yout2 = createVal(T1)

import matplotlib.pyplot as plt
plt.subplot(1,3,1)
plt.plot(yout, T)
plt.xlabel("Resposta simulada", fontsize = 8)
plt.grid()
#plt.title("Resposta simulada")
plt.subplot(1,3,2)
plt.plot(T1, res, 'r')
plt.xlabel("Resposta usando os residuos", fontsize = 8)
plt.title("Respostas ao impulso de diferentes metodos")
plt.grid()
#plt.title("Resposta usando os residuos")
plt.subplot(1,3,3)
plt.plot(T1, yout2, 'k')
plt.xlabel("Resposta usando as contas", fontsize = 8)
plt.grid()
#plt.title("Resposta usando as contas")
```

Listing 3: Representação da função no tempo

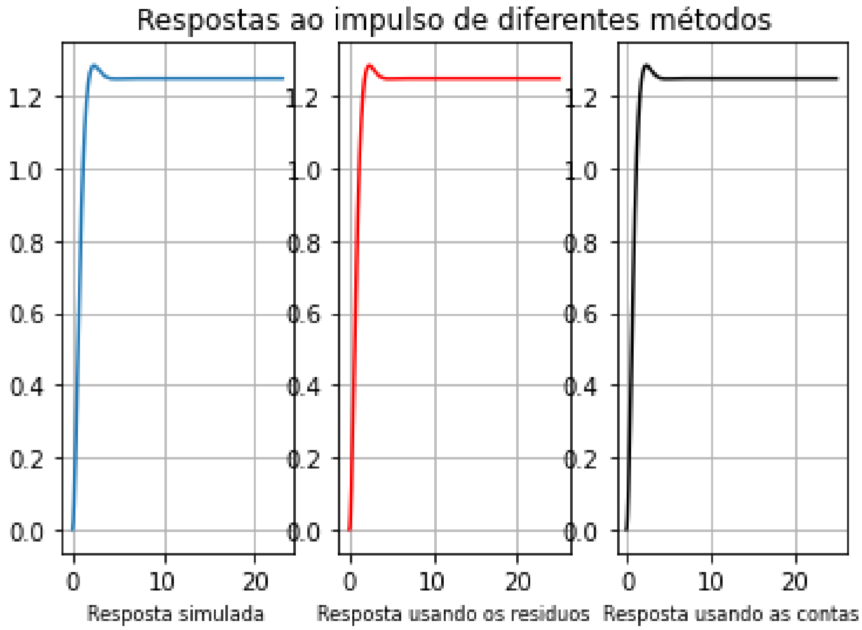


Figure 3: Comparação entre as saídas

3. Simular a resposta ao degrau usando a transformada inversa, para isso devemos calcular a transformada inversa de Y(s)*(1/s), vou usar o método numérico, mas o método 'braçal' pode ser encontrado nos anexos.

```
#Considerando que a biblioteca de controle ja foi importada acima
Ys2 = control.TransferFunction([1], [1, 0]) * Ys # 1/s * Y(s)
Ys2
res2 = signal.residue(Ys2.num[0][0][:], Ys2.den[0][0][:])
res2
import sympy
from sympy import Symbol

t = Symbol('t')
yt = Symbol('yt')
e = Symbol('e')

yt = res2[0][0]*e**(res2[1][0]*t) + t*res2[0][1]*e**(res2[1][1]*t) + res2[0][2]*e**(res2[1][2]*t) + res2[0][3]*e**(res2[1][3]*t) #Coloquei *t pq tem dois polos iguais https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.residue.html
yt

```

Listing 4: Representação da função no tempo

$$e^{t(-1.5-1.3228756555323i)}(0.46875+0.0590569489076918i)+e^{t(-1.5+1.3228756555323i)}(0.46875-0.0590569489076918i)+1.25t-0.9375$$

Figure 4: Resposta ao degrau no tempo

```
import numpy as np
import cmath
import matplotlib.pyplot as plt

yout2 , T3 = control.impulse_response(Ys2)

res =[]
```

```
    T1 = np.arange(0, 25, 0.01)
    for t1 in T1:
        yt1 = res2[0][0]*cmath.exp(res2[1][0]*t1) + t1*res2[0][1]*cmath.exp(res2[1][1]*t1) + res2[0][2]*cmath.exp(res2[1][2]*t1) + res2[0][3]*cmath.exp(res2[1][3]*t1)
        res.append(yt1)

    import matplotlib.pyplot as plt
    plt.figure(0)
    plt.subplot(1,2,1)
    plt.plot(yout2, T3)
    plt.grid()
    plt.title("Resposta simulada")
    plt.subplot(1,2,2)
    plt.plot(T1, res, 'r')
    plt.grid()
    plt.title("Resposta usando os residuos")

```

Listing 5: Respostas do sistema

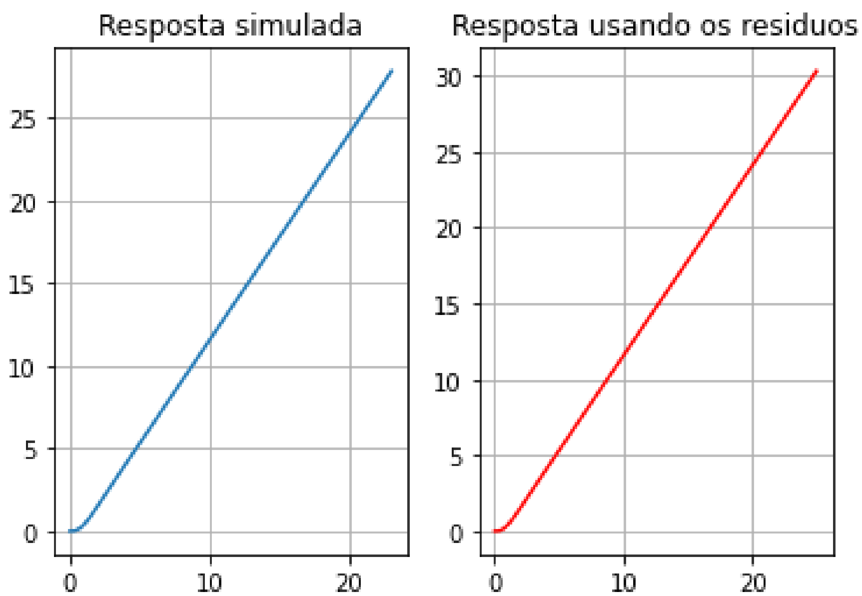


Figure 5: Respostas simulada e calculada

2 Exercício 2

Usando a função pzmap[1] obtemos os seguintes zeros e polos:

```
    from control.matlab import *
    num = [5, 3]
    den = [1,2,-5,-6]
    Ys = control.TransferFunction(num,den)
    pzmap(Ys)

    Output:
    (array([ 2.+0.j, -3.+0.j, -1.+0.j]), array([-0.6+0.j]))

```

Listing 6: Polos e zeros

[1] https://pythoncontrol.readthedocs.io/en/latest/generated/control.pzmap.html

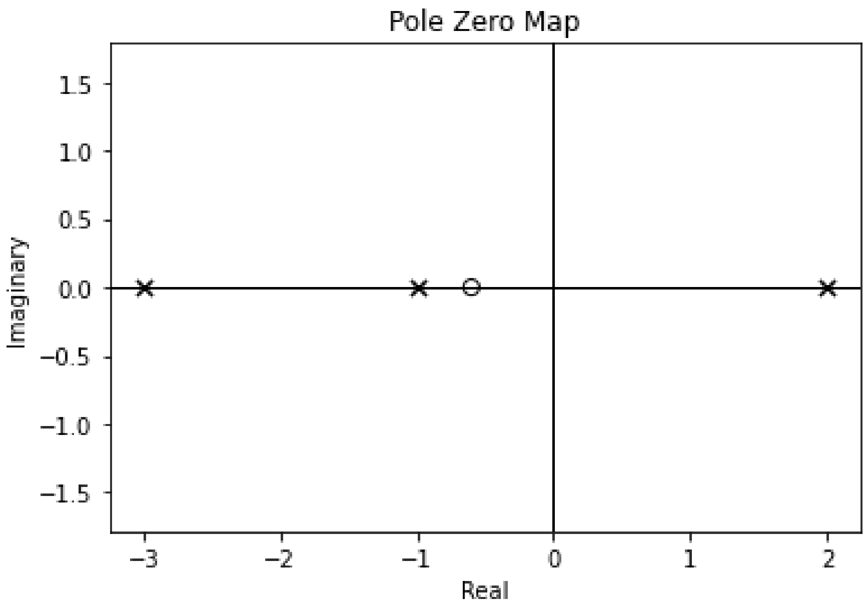


Figure 6: Polos e zeros da função

Com base nos polos do sistema podemos concluir que é um sistema instável, podemos observar isso a partir da resposta do sistema ao impulso:

```python
    import matplotlib.pyplot as plt
    yout , T = control.impulse_response(Ys)
    plt.plot(yout , T)

```

Listing 7: Resposta ao impulso

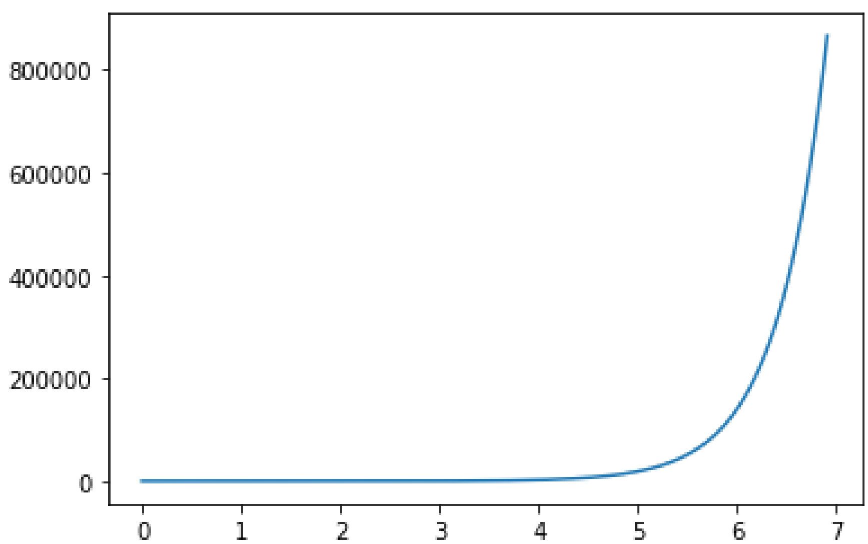


Figure 7: Resposta ao impulso

3 Exercício 3

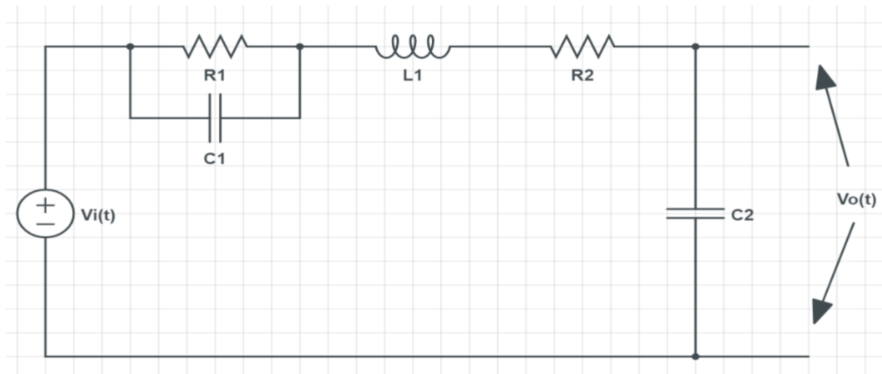


Figure 8: Circuito do exercício

Usando:

$$\begin{cases} R = R \\ C = \frac{1}{Cs} \\ L = sL \end{cases} \tag{7}$$

Temos:

$$Z_t = R1//C1 + L1 + R2 + C2 = \left(\frac{R1\frac{1}{C1s}}{R1 + \frac{1}{C1s}}\right) + sL1 + R2 + \frac{1}{C2s}$$

$$Z_t = \left(\frac{R1}{C1s}\frac{C1s}{R1C1s+1}\right) + sL1 + R2 + \frac{1}{C2s} = \frac{R1}{R1C1s+1} + sL1 + R2 + \frac{1}{C2s}$$

$$Z_t = \frac{R1C2s + (R1C1s+1)(sL1)(C2s) + (R1C1s+1)(R2)(C2s) + (R1C1s+1)}{(R1C1s+1)C2s}$$

$$Z_t = \frac{R1C2s + R1C1C2L1s^3 + L1C2s^2 + R1R2C1C2s^2 + R2C2s + (R1C1s+1)}{R1C1C2s^2 + C2s}$$

$$Z_t = \frac{R1C1C2L1s^3 + (L1C2 + R1R2C1C2)s^2 + (R1C2 + R2C2 + R1C1)s + 1}{R1C1C2s^2 + C2s} \tag{8}$$

Com 8 conseguimos calcular a corrente total do circuito:

$$I_t = \frac{V_i}{Z_t}$$

A tensão sobre o capacitor C2 é dada por:

$$V_o(t) = \frac{1}{C2}\int I_t(t)dt$$

$$V_o(s) = \frac{1}{C2s}I_t(s)$$

Temos, então:

$$V_o(s) = \frac{V_i(s)}{C2s(Z_t)} \tag{9}$$

Substituindo 8 em 9:

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{C2s}\frac{(R1C1s+1)C2s}{R1C1C2L1s^3 + (L1C2 + R1R2C1C2)s^2 + (R1C2 + R2C2 + R1C1)s + 1}$$

Substituindo os valores fornecidos no exercício:

$$\frac{V_o(s)}{V_i(s)} = \frac{10s+1}{100s^3 + 170s^2 + 46s + 1}$$

Simulando, temos:

```python
import control

num = [10,1]
den = [100,170,46,1]
Vo = control.TransferFunction(num, den)
Vo
```

Listing 8: Função de transferência

$$\frac{10s+1}{100s^3+170s^2+46s+1}$$

Figure 9: Função de transferência do circuito

```python
import matplotlib.pyplot as plt

yout, T = control.step_response(Vo)

plt.plot(yout, T)
plt.grid()
plt.title("Resposta do circuito a uma entrada degrau unitaria")
plt.show()
```

Listing 9: Resposta do sistema

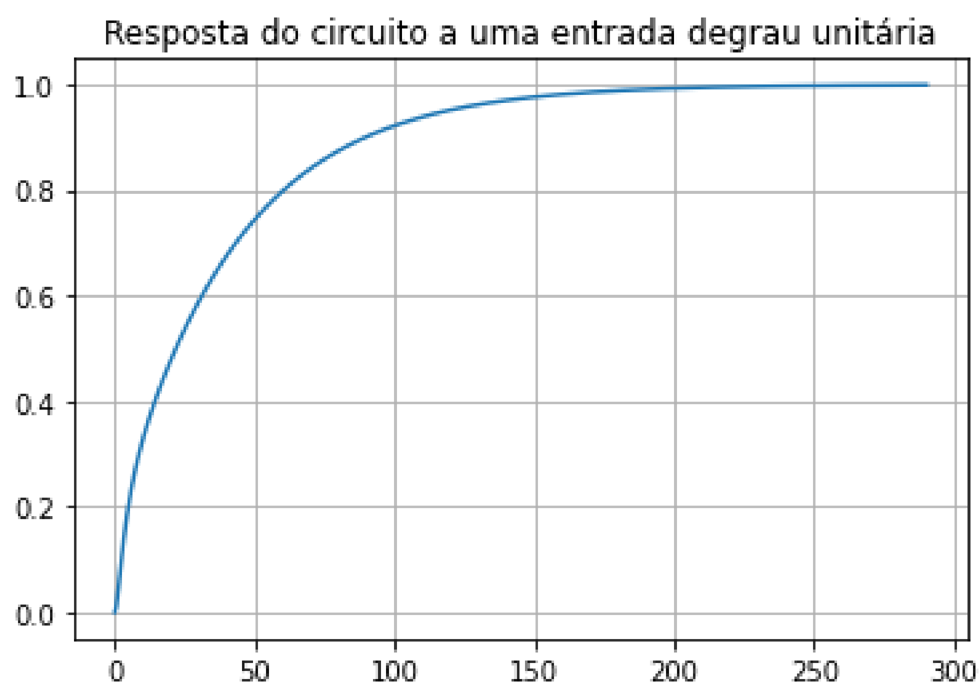


Figure 10: Resposta ao degrau unitário

Podemos perceber que o sistema basicamente não possui *overshoot*, isso se deve ao fato dele possuir um polo bem perto de zero e outros polos distantes, como podemos ver abaixo:

```python
from control.matlab import *

pzmap(Vo)

Output:
(array([-1.369425  +0.j, -0.30677114+0.j, -0.02380385+0.j]), array([-0.1+0.j]))
```

Listing 10: Polos e zeros

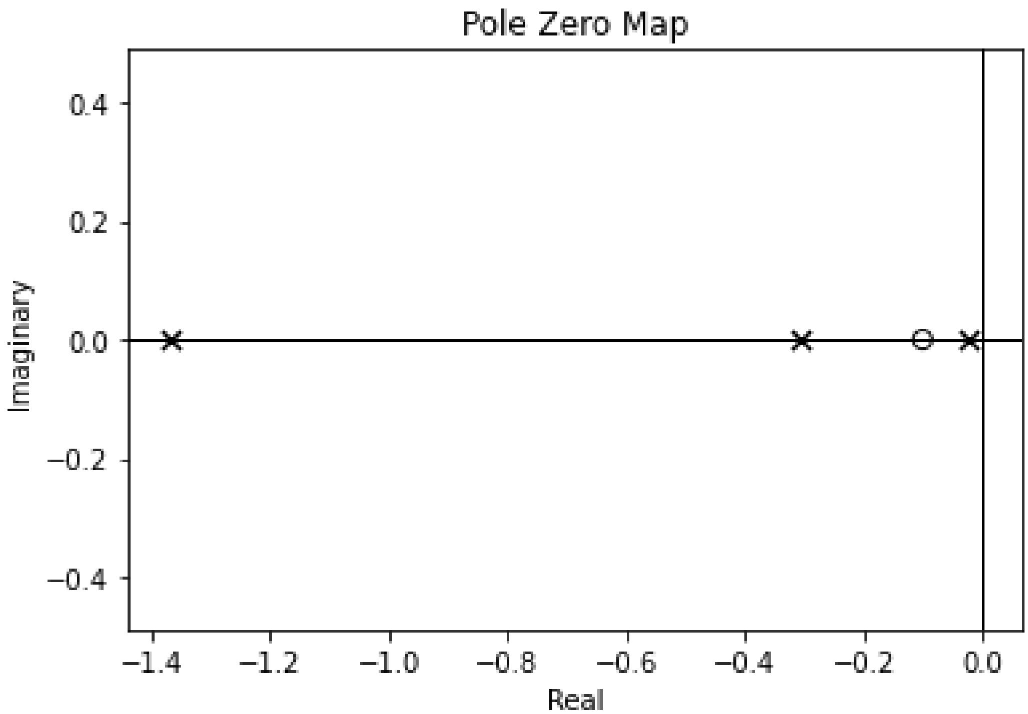


Figure 11: Polos e zeros do sistema