



HOMEWORK II

STUDENT NAME: LUCAS FERREIRA DA ROCHA

STUDENT ID: 542653

STUDENT NUMBER: LEONARDO MÁXIMO VIANA

STUDENT ID: 542656

EXERCISE 1

The unit steps responses of two systems A and B are recorded and reported in the files attached to the homework assignment. In each file, the first column gives the time vector, t , and the second column provides the output response, $y(t)$, for the systems A (in `HW2_ex1_dataA.csv`) and B (in `HW2_ex1_dataB.csv`)¹. Do the following:

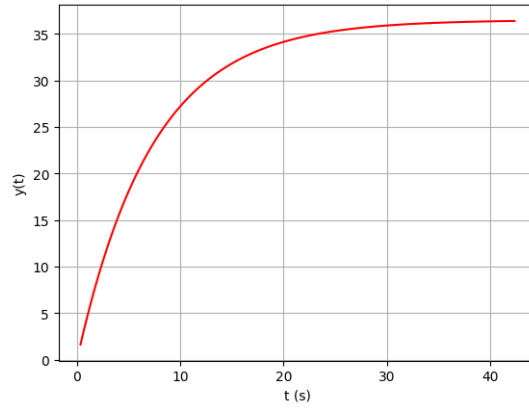
1. Load the data and plot the responses for systems A and B .

RESPOSTA 1

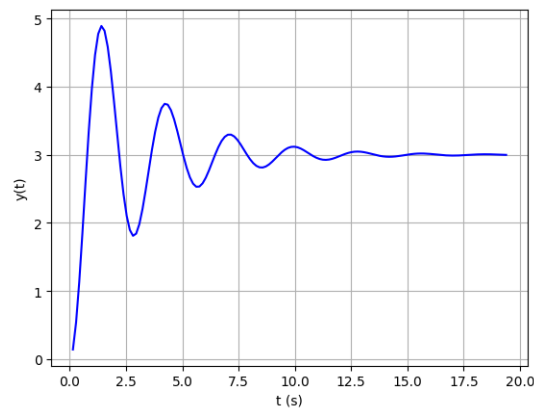
Resolvemos essa questão na linguagem python. Após carregar os dados, obtemos os seguintes gráficos de resposta para os sistemas A e B:

¹Data available in other format if needed!

Figura 1: Gráficos de A e B



(a) Resposta de A



(b) Resposta de B

Inicialmente, baixamos e importamos as bibliotecas necessárias para a resolução da questão.

Listing 1: Inicializando o código

```
1 !pip install control
2
3 import control
4 from control import matlab
5
6 #Importando as bibliotecas
7 import pandas as pd
8 import numpy as np
9 import matplotlib.pyplot as plt
10 from math import exp
```

Para plotar a resposta de A, após carregarmos o dataset com o comando `read_csv`

da biblioteca pandas, criaremos duas variáveis, A_x e A_y , as quais representam os eixos x (vetor de tempo t) e y (resposta $y(t)$) dos valores do dataset. Tais variáveis x e y são as colunas 0 e 0.1 no dataset printado, respectivamente. Depois, plotamos o gráfico com ajuda da biblioteca matplotlib.

Listing 2: Plotando a resposta de A

```
1 A = pd.read_csv('/content/HW2_ex1_dataA(2).csv')
2
3 A_x = A.drop(['0.1'],axis = 1)
4
5 A_y = A.drop(['0'],axis = 1)
6
7 plt.plot(A_x,A_y, color= 'red')
8 plt.xlabel('t (s)')
9 plt.ylabel('y(t)')
10
11 plt.grid()
12 plt.show()
```

Repetimos o mesmo processo com o dataset B, armazenando os eixos x e y em B_x e B_y , respectivamente.

Listing 3: Plotando a resposta de B

```
1 B = pd.read_csv('/content/HW2_ex1_dataB.csv')
2
3 B_x = B.drop(['0.1'],axis = 1)
4
5 B_y = B.drop(['0'],axis = 1)
6
7 plt.plot(B_x,B_y, color= 'blue')
8 plt.xlabel('t (s)')
9 plt.ylabel('y(t)')
10
11 plt.grid()
12 plt.show()
```

2. Identify the order of the systems. Based on the plots, estimate the transient response characteristics, such as time constant, settling time, rise time, peak time and percentage of overshoot. Write the corresponding transfer functions $T_A(s)$ and $T_B(s)$ for the systems A and B , respectively.

SOLUÇÃO 1

Analisando inicialmente o sistema A:

Pelo gráfico, percebe-se que ele é um sistema de 1ª ordem, visto que em nenhum momento há % overshoot e a inclinação inicial é diferente de zero. Para calcular sua constante de tempo, observamos o valor do tempo em que 63% do seu valor final é atingido a partir do momento em que o tempo de resposta ao degrau começa

a ser observado. Ademais, algumas informações importantes sobre essa constante de tempo são que ela caracteriza a performance de um sistema de primeira ordem. Analisando graficamente, percebemos que a constante de tempo τ será a 7.39s - 0.34s (visto que precisamos reduzir o atraso do valor encontrado como o momento em que se atinge 63 % do valor final dessa resposta). Assim, temos que τ é de, aproximadamente, 7 segundos.

Para descobrir o valor de "a", o qual é a taxa inicial de mudança da exponencial, temos que:

$$\tau = \frac{1}{a}$$

$$a = \frac{1}{\tau}$$

$$a = 0.14$$

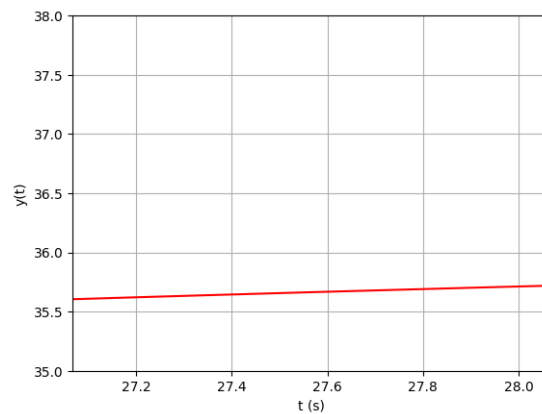
Para calcular o tempo de acomodação - o qual é o tempo para a resposta alcançar e permanecer entre 2% de seu valor final - temos que descobrir o tempo em que a resposta atinge um valor e permanece entre 2% do seu valor final, ou seja, quando:

$$Y_{ts} = 0.98 \times 36.39$$

$$Y_{ts} = 35.66$$

Como a curva é exponencial, ou seja, apenas cresce e depois se estabiliza, o valor do tempo de acomodação quando a resposta é igual a 35.66 é, aproximadamente, 27.57 segundos.

Figura 2: Analisando graficamente o tempo de acomodação

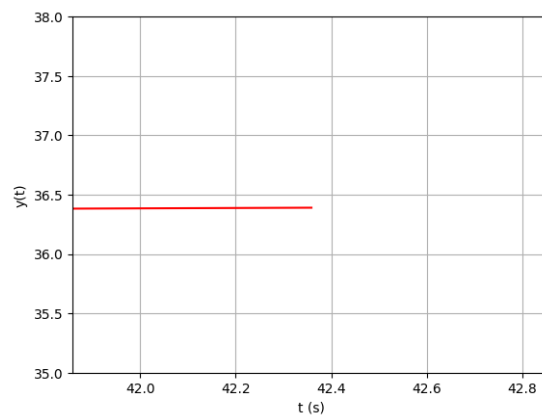


Para realizar a análise gráfica, plotamos um gráfico com as mesmas especificações já usadas no código passado. Assim descobrindo o tempo de acomodação

```
1 for i in range(len(y)):
2
3     if y[i] >= 35.66:
4
5         idx = i
6         break
7
8 plt.plot(A_x,A_y, color= 'red')
9 plt.xlabel('t (s)')
10 plt.ylabel('y(t)')
11
12 plt.xlim(t[i] - 0.5, t[i] + 0.5)
13 plt.ylim(35, 38)
14 plt.grid()
15 plt.show()
```

O tempo de pico é o primeiro momento em que a resposta atinge o seu valor máximo. Pela análise gráfica, temos que o tempo de pico é de, aproximadamente, 42.36 segundos.

Figura 3: Analisando graficamente o tempo de pico



Com o intuito de realizar a análise gráfica, plotamos um gráfico com as mesmas especificações dos códigos antigos e conseguimos obter o tempo de pico.

```
17 t_final = t.iloc[-1]
18
19 print(t_final)
20
21 plt.plot(A_x,A_y, color= 'red')
22 plt.xlabel('t (s)')
23 plt.ylabel('y(t)')
24
25 plt.xlim(t_final - 0.5, t_final + 0.5)
```

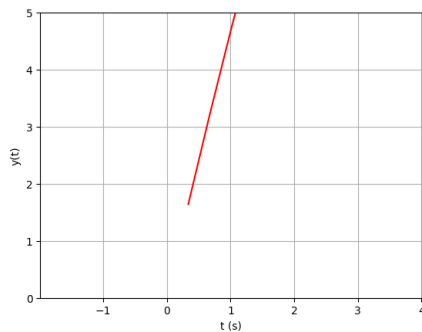
```

26 plt.ylim(35, 38)
27 plt.grid()
28 plt.show()

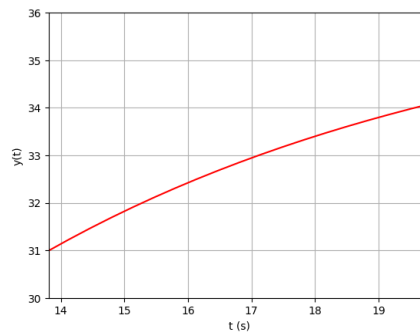
```

Já o tempo de subida é o tempo para a curva sair de 10% do seu valor final (no caso de A, 3.63) até 90% do seu valor final (32.75). No caso desse sistema, por análise gráfica, o tempo para que a curva chegue em um valor de 3.63 é de, aproximadamente, 0.80 segundos, e para que chegue em um valor de 32.75, aproximadamente, 16.80 segundos. Logo, o tempo de subida é $16,80 - 0,80 = 16,00$ segundos.

Figura 4: Análise de gráfico para tempo de subida



(a) Tempo quando $y = 10\%$ cfinal



(b) Tempo quando $y = 90\%$ cfinal

Com o intuito de realizar a análise gráfica, plotamos dois gráficos para obter o tempo de subida do sistema A. Inicialmente, com as mesmas especificações dos gráficos já plotados nesse item, conseguimos obter o tempo em que 10% do valor final do sistema A é atingido. Depois, realizamos o mesmo procedimento para obter o tempo em que 90% do valor final do sistema é atingido.

```

29
30 for i in range(len(y)):
31
32     # 10% do valor total
33
34     if y[i] >= 3.63:
35
36         idx = i
37         break
38
39 plt.plot(A_x,A_y, color= 'red')
40 plt.xlabel('t (s)')
41 plt.ylabel('y(t)')
42
43 plt.xlim(t[idx] - 3, t[idx] + 3)
44 plt.ylim(0, 5)
45 plt.grid()
46 plt.show()

```

```

47 -----
48
49
50 for i in range(len(y)):
51
52     # 90% do valor total
53
54     if y[i] >= 32.75:
55
56         idx = i
57         break
58
59 print(t[idx])
60
61 plt.plot(A_x,A_y, color= 'red')
62 plt.xlabel('t (s)')
63 plt.ylabel('y(t)')
64
65 plt.xlim(t[idx] - 3, t[idx] + 3)
66 plt.ylim(30, 36)
67 plt.grid()
68 plt.show()

```

Para calcularmos a função de transferência de um sistema de 1^a ordem, sabendo que o valor final desse sistema é diferente de 1:

$$G(s) = \frac{K}{\tau \cdot s + 1}$$

Sendo K o ganho de estado estático e τ a constante de tempo já apresentada acima.

Para definir a função de transferência desse sistema, consideramos ainda o atraso no tempo do sistema, tendo então que:

$$G(s) = \frac{K}{\tau \cdot s + 1} \cdot e^{-T_d \cdot s}$$

Sendo T_d o tempo de atraso. Realizando cálculos para descobrir K, temos, utilizando o teorema do valor final, o qual apresenta que o valor de K é o valor que a função de transferência quando s tende a zero, o qual é igual ao valor final do sistema (c_{final}):

$$\lim_{s \rightarrow 0} \frac{K}{\tau \cdot s + 1} \cdot e^{-T_d \cdot s} = c_{final}$$

$$\frac{K}{1} \cdot 1 = 36.39$$

Logo, K = 36.39

Assim:

$$G(s) = \frac{36.39}{7 \cdot s + 1} \cdot e^{-0.34 \cdot s}$$

Já para estimar as características da resposta transiente do dataset B, percebemos que ele é um sistema de 2^a ordem subamortecido, cujos polos do sistema são complexos conjugados e sua resposta transiente consiste em uma amplitude que decai exponencialmente (gerada pela parte real do sistema) e por uma onda senoidal gerada pela parte imaginária do polo do sistema, além de sua constante de amortecimento ζ estar entre 0 e 1. Calculando inicialmente sua % de overshoot, que é a porcentagem em que a resposta ultrapassa o valor de equilíbrio final em seu pico, temos que:

$$\%Os = \frac{c_{max} - c_{final}}{c_{final}} \cdot 100$$

Pelo código apresentado abaixo, obtemos que c_{max} (valor máximo do sistema) é aproximadamente 1.79 e c_{final} (valor final do sistema) é aproximadamente 1. Logo:

$$\%Os = \frac{4.89 - 3.00}{3.00} \cdot 100$$

$$\%Os = 63$$

A porcentagem de overshoot é de 63%.

Para conseguirmos descobrir o ponto máximo e final da senoide, utilizamos o comando `max()` e `iloc()`, retornando aproximadamente 4.89 e 3.00, respectivamente.

```
69 t = B['0']
70 y = B['0.1']
71
72 Cmax = y.max()
73
74 print("Cmax: ", Cmax)
75
76 Cfinal = y.iloc[-1]
77
78 print("\nCfinal: ", Cfinal)
```

Ademais, para descobrir ζ , o coeficiente de amortecimento, temos que:

$$\%Os = e^{\left(\frac{-\zeta \cdot \pi}{\sqrt{1 - \zeta^2}}\right)}$$

$$\zeta = \frac{-\ln(\frac{\%Os}{100})}{\sqrt{\pi^2 + (\ln(\frac{\%Os}{100}))^2}}$$

Para calcular ζ , temos que:

$$\zeta = \frac{-\ln(0.63)}{\sqrt{\pi^2 + (\ln(0.63))^2}}$$

$$\zeta = \frac{+0.46}{\sqrt{\pi^2 + 0.21}}$$

$$\zeta = \frac{+0.46}{\sqrt{9.87}}$$

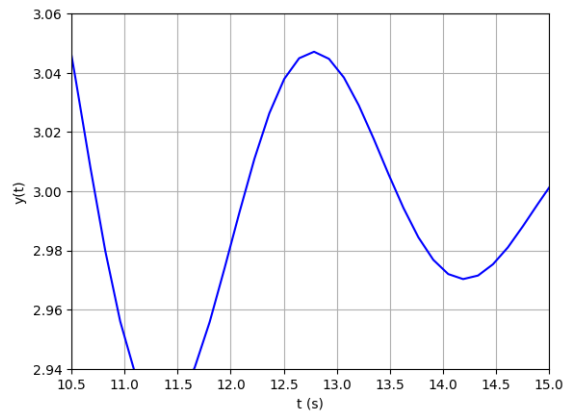
$$\zeta = \frac{+0.46}{3.17}$$

$$\zeta \approx 0.14$$

Como ζ está entre 0 e 1, confirmamos que o sistema B é um sistema de 2ª ordem subamortecido.

Pela análise do gráfico, percebemos que o tempo de acomodação do sistema, ou seja, o tempo para a senoide chegar e se manter entre 0.98 e 1.02 (aproximadamente 2% acima ou abaixo do valor final) é de aproximadamente 11.6 segundos, visto que esse é o primeiro tempo em que o sistema começa a se acomodar no intervalo especificado anteriormente.

Figura 5: Análise de gráfico para tempo de acomodação



Como feito na análise do sistema A, plotamos o gráfico necessário para a análise definindo os limites dos eixos x e y com os comandos `x.lim()` e `y.lim()` da biblioteca `matplotlib`. Após limitarmos os limites das imagens, conseguimos ter uma visão aproximada dos valores do tempo $t(s)$ e encontramos o tempo de acomodação.

```

79 plt.plot(B_x,B_y, color= 'blue')
80 plt.xlabel('t (s)')
81 plt.ylabel('y(t)')
82
83 plt.xlim([10.5, 15])
84 plt.ylim([2.94, 3.06])
85
86 plt.grid()
87 plt.show()

```

Para descobrirmos a frequência natural, ou seja, ω_n , utilizamos a fórmula a seguir:

$$T_s = \frac{4}{\zeta \cdot \omega_n}$$

Onde T_s é o tempo de acomodação. Assim, temos que:

$$11.6 = \frac{4}{0.14 \cdot \omega_n}$$

$$\omega_n = \frac{4}{0.14 \cdot 11.6}$$

$$\omega_n \approx 2.46 \text{ rad/s}$$

Já o tempo de pico, T_p é calculado pela fórmula:

$$T_p = \frac{\pi}{\omega_n \cdot \sqrt{1 - \zeta^2}}$$

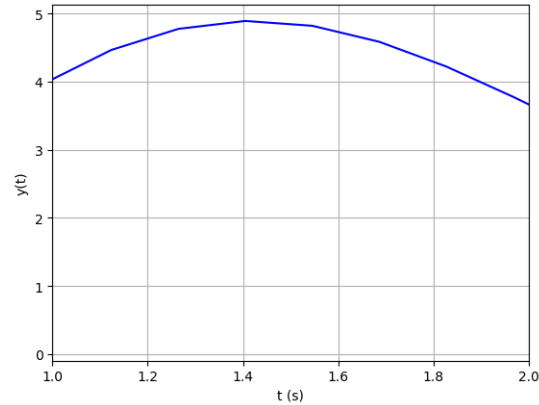
$$T_p = \frac{\pi}{2.46 \cdot \sqrt{0.98}}$$

$$T_p = \frac{\pi}{2.43}$$

$$T_p \approx 1.30 \text{ s}$$

Pelo gráfico, analisamos que o tempo de pico é bem próximo do que foi descoberto analiticamente, sendo aproximado de 1.3 segundos.

Figura 6: Análise de gráfico para tempo de pico

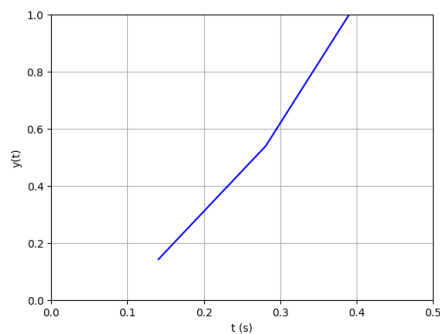


Assim como feito acima, conseguimos obter o tempo de pico pela análise gráfica.

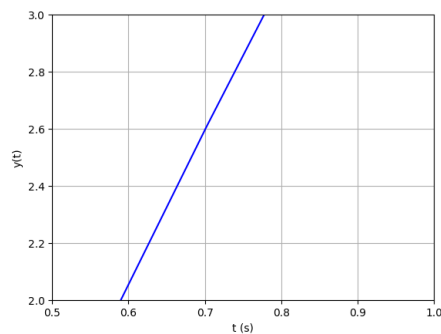
```
88 plt.plot(B_x,B_y, color= 'blue')
89 plt.xlabel('t (s)')
90 plt.ylabel('y(t)')
91
92 plt.xlim([1, 2])
93
94 plt.grid()
95 plt.show()
```

O tempo de subida é o tempo que leva para a senoide amortecida sair de 10% do seu valor final até atingir 90% do seu valor final. Pela análise do gráfico, temos que os 10% desse valor são atingidos inicialmente em 0.20 segundos e os 90% em 0.71 segundos. A diferença entre eles é o tempo de subida, que equivale a 0.51 segundos.

Figura 7: Análise de gráfico para tempo de subida



(a) Tempo quando $y = 10\%$ cfinal



(b) Tempo quando $y = 90\%$ cfinal

Com a mesma lógica dos gráficos acima, conseguimos descobrir o tempo de subida

ao obter analiticamente os primeiros tempos em que a senoide os 10% e 90% do seu valor final, respectivamente.

```

96 plt.plot(B_x,B_y, color= 'blue')
97 plt.xlabel('t (s)')
98 plt.ylabel('y(t)')
99
100 plt.ylim([0, 1])
101 plt.xlim([0, 0.5])
102
103 plt.grid()
104 plt.show()
105
106 -----
107
108 plt.plot(B_x,B_y, color= 'blue')
109 plt.xlabel('t (s)')
110 plt.ylabel('y(t)')
111
112 plt.ylim([2, 3])
113 plt.xlim([0.5, 1])
114
115 plt.grid()
116 plt.show()

```

A função de transferência de um sistema de segunda ordem pode ser calculada por:

$$G(s) = \frac{\omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2}$$

Assim, a função de transferência do sistema B será:

$$G(s) = \frac{2.46^2}{s^2 + 2 \cdot 0.14 \cdot 2.46 \cdot s + 2.46^2}$$

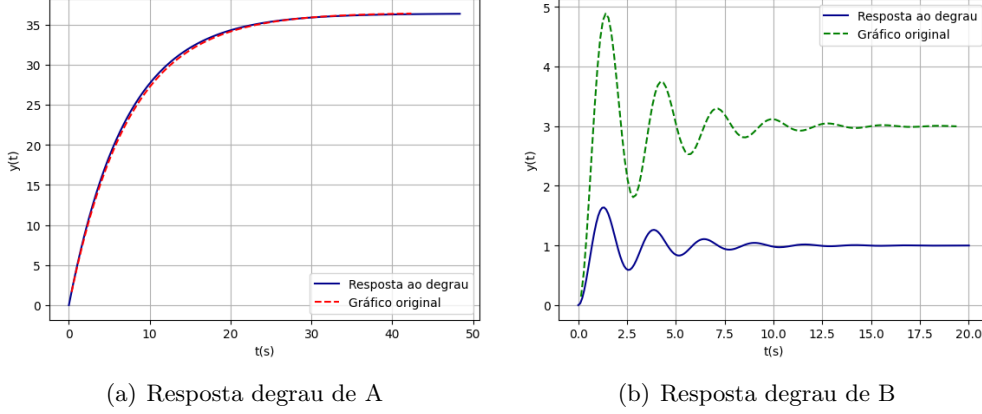
$$G(s) = \frac{6.05}{s^2 + 0.69 \cdot s + 6.05}$$

3. In the plot obtained in item (1) of this exercise, add and compare the unit step response of the systems $T_A(s)$ with the data provided in `HW2_ex1_dataA.csv`. Do the same with the unit step response of $T_B(s)$ with the data provided in `HW2_ex1_dataB.csv`. Comment on your results

RESPOSTA 1

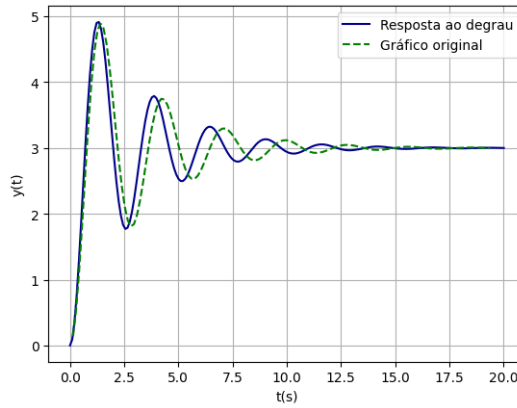
A resposta degrau dos sistemas versus o seu plot original é:

Figura 8: Respostas degrau dos sistemas



É perceptível que a resposta ao degrau do sistema A é extremamente fiel ao gráfico original. Entretanto, a resposta ao degrau do sistema B contém imperfeições em relação ao seu gráfico original. Além disso, é claro que existe uma grande diferença em relação ao offset entre as 2 representações. Contudo, quando multiplicamos $T_B(s)$ por um ganho fixo, conseguimos aproximar e relacionar muito melhor as funções, como mostra a figura abaixo:

Figura 9: Resposta ao degrau de B corrigido



Em consequência disso, é possível levantar hipóteses para analisar a razão da diferença no sistema B. Primordialmente, devemos levar em consideração o tipo de sistema representado, um sistema de segunda ordem, que é muito mais complexo e difícil de modelar do que um sistema de primeira ordem. Além disso, observamos que os dados originais parecem ter um atraso que se propaga durante as oscilações do sistema, o que dificulta a modelagem teórica dele. Não obstante, por usar valores relativizáveis, é possível que a função, quando não multiplicada por um ganho fixo, se mostre diferente do original.

Por fim, abaixo estão os códigos usados para plotar $T_A(s)$ e $T_B(s)$, sendo o último já corrigido.

Listing 4: Plotando a resposta degrau do sistema A

```

1 from control import matlab
2
3 num = [36.39]
4 den = [7,1]
5 sys = control.TransferFunction(num, den)
6
7 T, yout = control.step_response(sys)
8
9 plt.plot(T,yout, color= 'darkblue')
10 plt.plot(A_x,A_y, 'r--')
11 plt.xlabel('t(s)')
12 plt.ylabel('y(t)')
13 plt.legend(['Resposta ao degrau', 'Gráfico original'])
14 plt.grid()
15 plt.show()

```

Listing 5: Plotando a resposta degrau do sistema B

```

1 num = [6.05 * 3]
2 den = [1, 0.69, 6.05]
3 sys = control.TransferFunction(num, den)
4
5 T, yout = control.step_response(sys)
6
7 plt.plot(T, yout, color='darkblue', label='Resposta ao degrau')
8 plt.plot(B_x, B_y, 'g--', label='Gráfico original')
9 plt.xlabel(-4031
10 't(s)')
11 plt.ylabel('y(t)')
12 plt.grid()
13 plt.legend()
14 plt.show()

```

EXERCISE 2

EXERCÍCIO 2 2

É DADO O MODELO DE ESPAÇO DE ESTADOS EM [IT IS GIVEN THE STATE-SPACE MODEL IN]. 2

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

1. Calcule e explique a função de transferência correspondente ao sistema em [Compute and explain the transfer function corresponding to the system in]

Para resolver esta questão A função de transferência $G(s)$ de um sistema representado no espaço de estados é dada pela fórmula:

$$G(s) = C(sI - A)^{-1}B + D$$

Onde I é a matriz identidade de mesma dimensão que A . Para resolver esta questão é preciso inicialmente definir os valores dados pela questão para substituir na equação de $G(s)$.

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = [0]$$

Passo 1: Iniciando pelo cálculo de $(sI - A)$

$$\begin{aligned} sI - A &= s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \\ &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \\ &= \begin{bmatrix} s & -1 \\ 5 & s+2 \end{bmatrix} \end{aligned}$$

Passo 2: Calculando a inversa $(sI - A)^{-1}$

A inversa de uma matriz 2x2 é $\frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ portanto:

$$(sI - A)^{-1} = \frac{1}{s^2 + 2s + 5} \begin{bmatrix} s+2 & 1 \\ -5 & s \end{bmatrix}$$

Passo 3: Multiplicando a inversa por C. $(C(sI - A)^{-1})$

$$\begin{aligned} C(sI - A)^{-1} &= [0 \quad 1] \frac{1}{s^2 + 2s + 5} \begin{bmatrix} s+2 & 1 \\ -5 & s \end{bmatrix} \\ &= \frac{1}{s^2 + 2s + 5} [(0)(s+2) + (1)(-5) \quad (0)(1) + (1)(s)] \\ &= \frac{1}{s^2 + 2s + 5} [-5 \quad s] \end{aligned}$$

Passo 4: Multiplicando por B. ($C(sI - A)^{-1}B$)

$$\begin{aligned} C(sI - A)^{-1}B &= \frac{1}{s^2 + 2s + 5} \begin{bmatrix} -5 & s \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \\ &= \frac{1}{s^2 + 2s + 5} [(-5)(0) + (s)(2)] \\ &= \frac{1}{s^2 + 2s + 5} [2s] \\ &= \frac{2s}{s^2 + 2s + 5} \end{aligned}$$

Como $D = [0]$, a função de transferência final é:

$$\mathbf{G}(s) = \frac{2s}{s^2 + 2s + 5}$$

Explicação da Função de Transferência

A função de transferência deste sistema mostrada acima, possui seu zero quando $s = 0$, para descobrir isto basta olhar para o seu numerado que é $2s$, mas quando olhamos para o denominador vemos o polinômio característico dado por $s^2 + 2s + 5$, por meio das raízes do denominador achamos os polos da função que são:

$$\begin{aligned} s &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ s &= \frac{-2 \pm \sqrt{2^2 - 4(1)(5)}}{2(1)} \\ s &= \frac{-2 \pm \sqrt{4 - 20}}{2} \\ s &= \frac{-2 \pm \sqrt{-16}}{2} \\ s &= \frac{-2 \pm j4}{2} \\ s &= -1 \pm j2 \end{aligned}$$

Ambos os polos são complexos conjugados, localizados em $s = -1 + j2$ e $s = -1 - j2$.

Logo, como os dois polos possuem parte real negativa (-1) o sistema é estável e a parte imaginária mostra que o sistema é subamortecido, ou seja, são oscilações que decaem com o tempo até o sistema atingir o regime permanente.

2. Explique brevemente o conceito de estabilidade *bounded-input, bounded-output* (BIBO). Avalie a estabilidade BIBO para o sistema dado.

Conceito de Estabilidade BIBO

A estabilidade Bounded-Input, Bounded-Output (BIBO) é uma propriedade fundamental para sistemas lineares e invariantes no tempo que diz que um sistema é estável se toda entrada limitada gerar uma saída limitada e um sistema é instável se alguma entrada limitada gerar uma saída ilimitada.

Estabilidade BIBO do Sistema Dado

Para avaliar a estabilidade BIBO, utilizamos os polos descobertos na questão 1 deste exercício:

$$s = -1 \pm j2$$

Analisando os polos vemos que eles estão na parte real negativa, não há polos no semiplano direito do plano s e nenhum está localizado sobre o eixo imaginário. Portanto, ele é considerado BIBO estável já que com uma entrada limitada ele não irá crescer infinitamente.

3. Explique brevemente o conceito de estabilidade de Lyapunov. Avalie a estabilidade de Lyapunov para o sistema dado.

Conceito de Estabilidade de Lyapunov

A estabilidade de Lyapunov é utilizada para analisar sistemas não lineares e dinâmicos onde métodos como Routh-Hurwitz não são aplicáveis. A estabilidade foca no comportamento das soluções de estado de um sistema nas proximidades de um ponto de equilíbrio, o método mais comum é o método direto de Lyapunov que se baseia na construção de uma função de Lyapunov candidata $V(x)$.

Para um ponto de equilíbrio na origem ($x = 0$), uma função dada $V(x)$ é função de Lyapunov se ela é positivamente definida ($V(x) > 0$ para $x \neq 0$ e $V(0) = 0$), e a derivada em relação ao tempo de $V(x)$ é negativamente semidefinida ($\dot{V}(x) \leq 0$ para $x \neq 0$).

Logo, se o sistema seguir estas condições ele será estável, se além destas condições $V(x)$ for negativamente definida ($V(x) < 0$ para $x \neq 0$) então o ponto de equilíbrio é assintoticamente estável, ou seja, ele não apenas permanece perto do ponto de equilíbrio, mas também retorna a ele com o tempo. Por fim, se a estabilidade assintótica é válida para todo o espaço de estados e $V(x)$ é radialmente ilimitado o sistema possui estabilidade assintótica global.

Estabilidade de Lyapunov para o Sistema Dado

Sabendo que o sistema dado é linearmente invariante no tempo, a estabilidade de Lyapunov é calculada pelos autovalores da matriz de estado de A que é:

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}$$

Para encontrar os autovalores (λ), calculamos o determinante da matriz $(\lambda I - A)$, c

$$\begin{aligned} \det(\lambda I - A) &= \det \left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \right) = 0 \\ &= \det \left(\begin{bmatrix} \lambda & -1 \\ 5 & \lambda + 2 \end{bmatrix} \right) = 0 \\ &= (\lambda)(\lambda + 2) - (-1)(5) = 0 \\ &= \lambda^2 + 2\lambda + 5 = 0 \end{aligned}$$

As raízes desta equação são:

$$\begin{aligned} \lambda &= \frac{-2 \pm \sqrt{2^2 - 4(1)(5)}}{2(1)} \\ \lambda &= \frac{-2 \pm \sqrt{4 - 20}}{2} \\ \lambda &= \frac{-2 \pm \sqrt{-16}}{2} \\ \lambda &= \frac{-2 \pm j4}{2} \\ \lambda_1 &= -1 + j2 \\ \lambda_2 &= -1 - j2 \end{aligned}$$

Portanto, como ambas as raízes possuem parte real negativa(-1) para sistemas LTI isso significa que é assintoticamente estável. Como na ausência de entrada externa($u(t) = 0$) os resultados do sistema($x(t)$) convergirão para o ponto de equilíbrio a medida que o tempo tende ao infinito, ou seja, ele é assintoticamente estável no conceito de Lyapunov.

4. Discuta como a teoria da estabilidade BIBO e a estabilidade de Lyapunov diferem para sistemas LTI. Compare e explique os resultados do item (2) e (3) deste exercício.[Discuss how BIBO stability theory and Lyapunov stability differ for LTI systems. Compare and explain the results of item (2) and (3) of this exercise].

A estabilidade BIBO avalia se a saída de um sistema permanece limitada se a entrada é limitada,portanto, no sistema LTI todos os polos da função de transferência $G(s)$ devem ter a parte real negativa como é mostrado na segunda questão deste exercício.

A estabilidade de Lyapunov avalia se os estados internos do sistema retornam ao equilíbrio na ausência de uma entrada,para isto acontecer todos os autovalores da matriz de estado A devem possuir parte real negativa como é calculado na questão anterior.

Neste sistema, os polos da função de transferência são iguais aos autovalores da matriz de estado, como todos eles possuem parte real negativa, o sistema é BIBO estável e assintoticamente estável no sentido de Lyapunov.

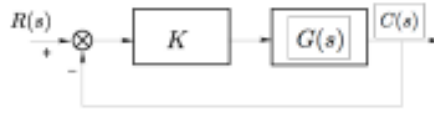


Figura 10: Malha de retroalimentação negativa considerada no Exercício [Negative feedback loop considered in Exercise] 3.

EXERCISE 3

Para o sistema com retroalimentação unitária representado na Figura (1), $G(s)$ é dado conforme a Equação [For the unit feedback system in figure (13), $G(s)$ is given as in] (2).

$$G(s) = \frac{s + 4}{s \cdot (s^2 + 3.2 \cdot s + 2.4)}$$

1. Utilize o critério de Routh-Hurwitz para determinar [Use the Routh-Hurwitz criterion to define]:

A) O intervalo de valores de K que proporciona estabilidade ao sistema em malha fechada [The range of K that provides stability to the closed-loop system].

B) O intervalo de valores de K que faz o sistema oscilar [The range of K that makes

2. Plote e discuta as respostas ao degrau para valores de K estáveis e oscilatórios [Plot and discuss the step responses for stable and oscillatory values of K].

SOLUÇÃO 3

1.A) O critério de Routh-Hurwitz é um método que calcula quantos polos existem em um sistema de malha fechada em cada seção do plano s , ou seja, ele nos mostra informações sobre a estabilidade do sistema mas não sobre suas coordenadas. Para o cálculo do critério é necessário montar uma tabela chamada de tabela de Routh(1), para montar a tabela se começa rotulando as linhas com a maior potência de s até s^0 , em seguida, como se busca sobre os polos para determinar a estabilidade, se utilizam os valores do denominador da função de transferência para preencher a tabela inserindo eles horizontalmente na primeira linha iniciando sempre com o coeficiente da maior potência de s e mas sempre pulando um coeficiente, na segunda linha eles também devem ser listados horizontalmente e pulando um coeficiente, mas se deve iniciar pelo coeficiente que tenha a segunda maior potência de s . O restante dos elementos são preenchidos com o negativo do determinante de elementos das duas linhas anteriores dividido pelo elemento na primeira coluna diretamente acima da linha que está sendo calculada. Para resolver a questão primeiramente se deve achar o denominador da função de transferência da retroalimentação negativa (10), sabendo que a função de transferência da retroalimentação negativa é: $T(s) = \frac{K \cdot G(s)}{1 + K \cdot G(s)}$

Encontrando o denominador característico da função utilizando o $G(s)$ dado na questão:

s^3	1	$(2.4 + k)$
s^2	3.2	$(4 \cdot k)$
s^1	b	$b1$
s^0	c	$c2$

Tabela 1: Tabela de Routh

$1 + K \cdot \frac{s + 4}{s(s^2 + 3.2s + 2.4)} = 0$ Multiplicando a equação por $s(s^2 + 3.2s + 2.4)$:

$$s(s^2 + 3.2s + 2.4) + k(s + 4)$$

$$s^3 + 3.2s^2 + (2.4 + K)s + 4K$$

Portanto o denominador característico é $P(s)s^3 + 3.2s^2 + (2.4 + K)s + 4K$. Com isso inserimos os valores na tabela como explicado anteriormente.

Para descobrir a estabilidade do sistema se analisa a quantidade de mudanças na primeira coluna da tabela de Routh, pois se não houver mudança de sinal o sistema é estável e o critério de Routh-Hurwitz estabelece isso porque o número de raízes do polinômio que estão no semiplano direito é igual ao número de mudanças de sinal na primeira coluna, portanto, se a função de transferência em malha fechada possui todos os polos na metade esquerda do plano s , não haverá mudança de sinal na primeira coluna da tabela logo, o sistema será estável. Resolvendo a Tabela de Routh-Hurwitz:

Cálculo do elemento b :

$$b = \frac{(3.2)(2.4 + K) - (1)(4K)}{3.2}$$

$$b = \frac{7.68 + 3.2K - 4K}{3.2}$$

$$b = \frac{7.68 - 0.8K}{3.2}$$

Cálculo do elemento $b1$:

$$b1 = \frac{3.2 \cdot 0 - 1 \cdot 0}{3.2}$$

$$b1 = \frac{0}{3.2}$$

$$b1 = 0$$

Cálculo do elemento c :

$$c = \frac{(b)(4K) - (3.2)(0)}{b}$$

$$c = 4K$$

Cálculo do elemento c_1 :

$$c_1 = \frac{(b)(0) - (3.2)(0)}{b}$$

$$c_1 = \frac{0}{b}$$

$$c_1 = 0$$

Portanto, sabendo que os dois primeiros números da primeira coluna da tabela são positivos para o sistema ser estável, b e c devem ser positivos.

Cálculo de $b > 0$:

$$b > 0$$

$$b = \frac{7.68 - 0.8K}{3.2}$$

$$0 < \frac{7.68 - 0.8K}{3.2}$$

$$0 < 7.68 - 0.8K$$

$$0.8K < 7.68$$

$$K < 9.6$$

Calculo de $b > 0$

$$c = 4 \cdot K$$

$$c > 0$$

$$4 \cdot K > 0$$

$$K > 0$$

Para este sistema ser estável, de acordo com o Critério de Routh-Hurwitz, K deve estar entre 0 e 9.6.

B)

Para determinar o valor de K que provoca oscilações no sistema, se utiliza o Critério de Routh-Hurwitz. O sistema oscila atinge o limite da estabilidade, o que é identificado na Tabela de Routh quando uma linha completa de zeros aparece na primeira coluna. Isso mostra a presença de polos puramente imaginários no plano s , causando oscilação. O sistema entra em um estado de oscilação quando entra no limite da estabilidade, ou seja, quando um dos valores da primeira coluna é igual a 0 enquanto os demais são positivos. Analisando os valores da primeira coluna da tabela calculada, os valores que K pode ter para ser 0, são $k = 9.6$ e $k = 0$. No caso de $K = 0$, o sistema apresenta um polo na origem ($s = 0$), caracterizando estabilidade marginal, mas não necessariamente oscila com frequência não nula, para o caso de $K = 9.6$ a linha se torna e indica a oscilação no sistema. Para determinar os valores que K oscila, utilizamos o Polinômio Auxiliar, que é formado pelos coeficientes da linha imediatamente acima da linha de zeros, portanto é:

$$A(s) = 3.2s^2 + 4 \cdot K$$

Substituindo $K = 9.6$:

$$A(s) = 3.2s^2 + 4 \cdot 9.6$$

$$A(s) = 3.2s^2 + 38.4$$

Calculando as raízes:

$$3.2s + 38.4 = 0$$

$$s = \sqrt{\frac{-38.4}{3.2}}$$

$$s = \sqrt{-12}$$

$$s = j \cdot \sqrt{3 \cdot 4}$$

$$s = \pm 2j\sqrt{12}$$

Resolução com cálculos do porque o sistema não tem uma oscilação sustentada com frequência não nula quando o k é igual a 0.

Como s_0 é a linha dos zeros, se utiliza a linha s_1 que é composta por bs^1 e b_1s^0 , como b_1 é igual a 0, o polinômio característico se dá por bs^1 :

$$A(s) = \frac{7.68 - 0.8K}{3.2}s$$

$$A(s) = \frac{7.68 - 0.8 \cdot 0}{3.2}s$$

$$A(s) = \frac{7.68}{3.2}s$$

$$A(s) = 2.4s$$

Com isso, é perceptível que a única raiz para s é $s = 0$, isso significa que o sistema possui um único polo na origem do plano s , isso caracteriza um sistema marginalmente estável mas não causa oscilações com frequência diferente de zero.

2) Inicialmente, adicionamos as bibliotecas que serão usadas no código, a função de transferência que a questão pede, o vetor d de tempo para a simulação e os valores de K que serão utilizados, para demonstração, foram escolhidos valores de K de modo que seja analisado a função quando ela é estável (11), marginalmente estável com polo na origem (12), oscilatório (13) e instável (16).

Listing 6: Plotando a Figura.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import signal
4
5 # Polinômio característico do sistema: s^3 + 3.2s^2 + (2.4 + K)s + 4K
6 # Numerador do sistema de malha fechada: K*(s+4)
7
8 # Função para obter a função de transferência de malha fechada para um
   dado K
```

```

9 def get_closed_loop_tf(K):
10     # Numerador N(s) = K*(s+4)
11     num = [K, 4*K]
12     # Denominador D(s) = s^3 + 3.2s^2 + (2.4 + K)s + 4K
13     den = [1, 3.2, (2.4 + K), 4*K]
14     return signal.TransferFunction(num, den)
15
16 # Definir o vetor de tempo para a simulação da resposta ao degrau
17 t = np.linspace(0, 30, 800) # De 0 a 30 segundos, com 800 pontos
18
19 # --- Valores de K para Análise ---
20 # Baseado na análise de Routh-Hurwitz, onde 0 < K < 9.6 para estabilidade
21
22 k_values = {
23     1: "Estável",
24     0: "Polo na origem",
25     9.6: "Oscilatório",
26     10: "Instável"
27 }

```

Sabendo disso utilizamos o código abaixo para plotar cada um dos gráficos e enfim analisá-los.

```

1
2 for K_val, description in k_values.items():
3     tf = get_closed_loop_tf(K_val)
4     tout, yout = signal.step(tf, T=t)
5
6     plt.figure(figsize=(10, 6))
7     plt.title(f'Resposta ao Degrau para K = {K_val} ({description})')
8     plt.xlabel('Tempo (s)')
9     plt.ylabel('Saída C(t)')
10    plt.grid(True)
11    plt.axhline(y=1, color='gray', linestyle='--', label='Valor de Estado Estacionário (para entrada degrau unitário)')
12
13    plt.plot(tout, yout, label=f'K = {K_val}')
14    plt.legend()
15    plt.tight_layout()
16    plt.show()

```

Para $K = 1$: Vemos por análise gráfica que quando K é 1 o sistema é estável e bem amortecido.

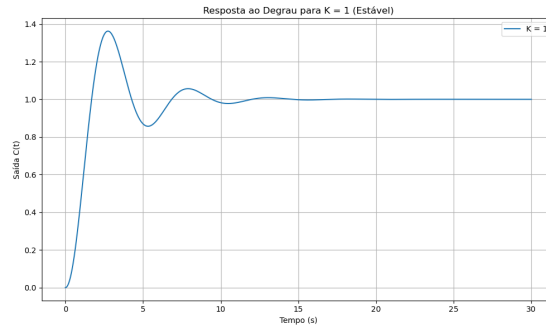


Figura 11: $K = 1$

Para $K = 0$: O sistema é sempre 0 pois, a função de transferência é $T(s) = \frac{K \cdot G(s)}{1 + K \cdot G(s)}$, logo o numerador será sempre 0 e o sistema não sofrerá alterações.

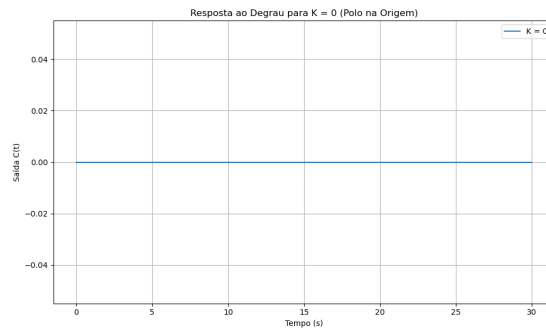


Figura 12: $K = 0$

Para $K = 9.6$: É perceptível que o sistema é oscilatório como mostrado no item B da primeira questão, pois o sistema completa uma linha de zeros na tabela de Routh e isso indica que o sistema possui um par de polos localizados no eixo imaginário do plano complexo.

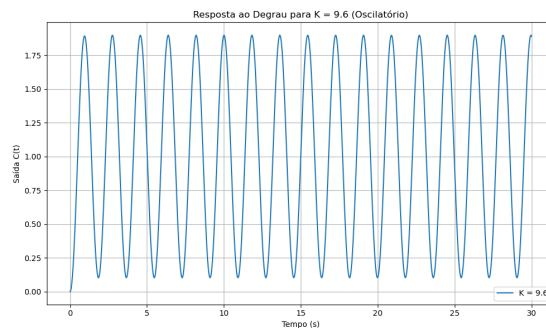


Figura 13: $K = 9.6$

Para $K = 10$: Neste caso como o K é maior que 9.6 o sistema se torna instável e cresce continuamente durante o tempo, ou seja, $k = 10$ levaria a uma mudança de sinal na tabela de Routh.

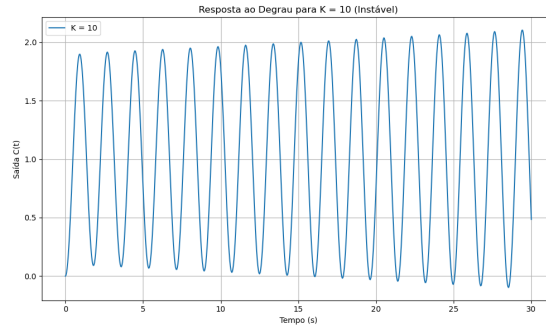


Figura 14: $K = 10$

EXERCISE 4

Given the negative feedback system shown in Figure 15, do the following:

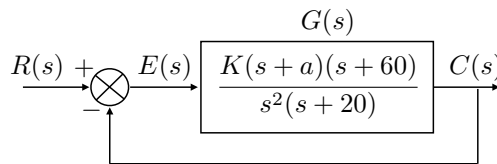


Figura 15: [Negative feedback loop considered in Exercise 4]

1. For $a = 40$:

- (a) Using the programming language of your choice, plot the root locus when the parameter K varies. Explain the main steps required for sketching it by hand.

4.0.1 Resposta:

Primeiramente, é necessário entender que um Root Locus tem como objetivo facilitar a análise de sistemas, quando há a alteração de parâmetros importantes, e suas características. Dessa forma, propriedades foram definidas para que seja possível analisar um Root Locus facilmente.

Em relação ao sistema apresentado no enunciado, antes de representar graficamente o Root Locus do sistema, é imprescindível a observação dos intervalos de pontos válidos, a localização dos polos e dos zeros, o ponto do eixo real

aonde as assíntotas se encontrarão (caso exista) e o ângulo das assíntotas.

Sendo assim, podemos perceber, através da função de transferência, que teremos 3 polos e 2 zeros. Existe um polo duplo em 0 e outro polo em -20, já os zeros se encontram em -40 (dado que $a = 40$) e -60.

$$G(s) = \frac{K(s + 40)(s + 60)}{s^2(s + 20)}$$

Dessa forma, podemos analisar que 1 dos ramos (conjunto de pontos representado por cada polo) irá para o infinito, enquanto 2 polos terão seus respectivos zeros e serão finitos.

Além disso, é possível definir os intervalos de pontos válidos, já que uma das propriedades de um Root Locus é que a quantidade de zeros e polos à direita de um ponto deve totalizar um número ímpar. Sendo assim, o intervalo de -20 a -40 será válido (Polos = 3), o intervalo de -60 ao infinito também será válido (Polos = 3 e Zeros = 2) e os demais intervalos não conterão pontos válidos porque o somatório do número de polos e zeros à sua direita representa um número par.

Com os intervalos definidos, percebemos que o polo duplo precisa extrapolar o eixo real e "fugir" para o eixo imaginário. Em consequência disso, precisamos calcular o ponto de chegada (break-in) deles. Em relação ao ângulo, como sabemos que existe apenas 1 ramo que irá para o infinito, podemos concluir que o ângulo será 180° .

Usando a equação:

$$\sum_{i=1}^n \frac{1}{\sigma - z_i} = \sum_{i=1}^n \frac{1}{\sigma - p_i}$$

Substituindo os valores na fórmula:

$$\frac{1}{\sigma - (-40)} + \frac{1}{\sigma - (-60)} = \frac{1}{\sigma - 0} + \frac{1}{\sigma - 0} + \frac{1}{\sigma - (-20)}$$

Desenvolvendo e manipulando os termos, temos:

$$0 = \sigma^3 + 200\sigma^2 + 9200\sigma + 96000$$

Usando uma calculadora, obtemos 3 possíveis valores para σ : $\sigma_1 = -135.4$, $\sigma_2 = -50.4$ e $\sigma_3 = -14.2$. Analisando os intervalos de pontos válidos definidos anteriormente, é claro que apenas o σ_1 é apto para ser o ponto de chegada do sistema.

Esses são todas as características necessárias para esboçar um Root Locus. Fazendo esse trabalho com python, obtemos os mesmos resultados anteriores e o gráfico representativo.

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 a = 40
5
6 # Coeficientes do numerador: (s+40)(s+60) = s^2 + 100s + 2400
7 num = [1, 100, 2400]
8
9 # Coeficientes do denominador: s^2 * (s+20) = s^3 + 20s^2 + 0s +
   0
10 den = [1, 20, 0, 0]
11
12 G = control.tf(num, den)
13
14 # Plotar o Lugar Geométrico das Raízes
15 plt.figure(figsize=(12, 8))
16
17 # root_locus
18 rlist, klist = control.root_locus(G, plot=True)
19
20 plt.title("Lugar Geométrico das Raízes (a=40)")
21 plt.xlabel("Eixo Real")
22 plt.ylabel("Eixo Imaginário")
23 plt.grid(True)
24 plt.xlim(-200, 10)
25 plt.ylim(-80, 100)
26 plt.show()

```

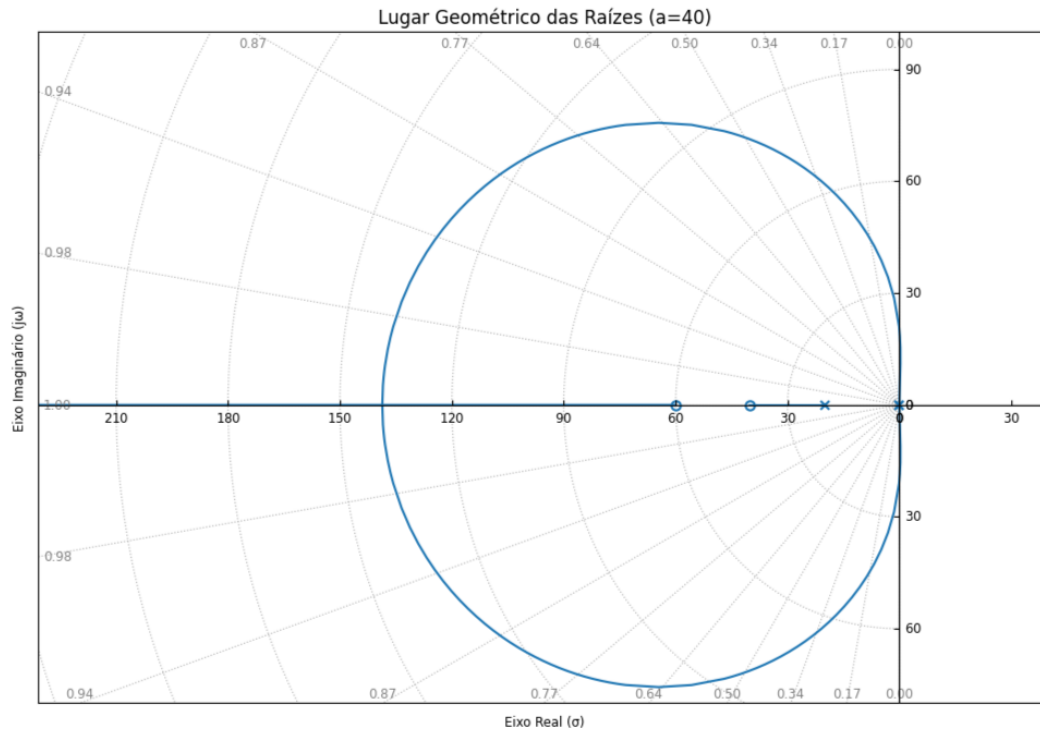


Figura 16: Root Locus

- (b) With exact calculations, verify on the root locus (i) the point and gain where the locus crosses the $j\omega$ -axis; (ii) the breakaway and break-in points on the real axis.

4.0.2 Resposta:

Este é talvez o cálculo de refinamento mais crítico. O ponto onde um ramo cruza o eixo imaginário ($j\omega$) é o ponto onde o sistema deixa de ser estável e se torna oscilatório ou instável. Ao encontrar esse ponto, podemos definir o ganho máximo (K máximo) e a frequência de oscilação máxima (ω máximo). Para isso, é necessário usar o critério de estabilidade de Routh-Hurwitz, que delimita os passos:

- i. **Encontrar a equação Característica:** $1 + G(s) = 0$

$$1 + \frac{K(s^2 + 100s + 2400)}{s^2(s + 20)} = 0$$

$$s^3 + 20s^2 + K(s^2 + 100s + 2400) = 0$$

$$s^3 + (20 + K)s^2 + 100Ks + 2400K = 0$$

ii. **Montar a tabela de Routh:**

Ordem	Coefficiente 1	Coefficiente 2
s^3	1	$100K$
s^2	$20 + K$	$2400K$
s^1	$\frac{(20 + K)(100K) - 2400K \cdot 20}{20 + K}$	K
s^0	$2400K$	0

iii. **Condição para Cruzamento:** Para o sistema estar no limite da estabilidade (cruzando o eixo $j\omega$), a linha s^1 deve ser toda de zeros:

$$(20 + K)(100K) - 2400K = 0$$

$$100K^2 + 2000K - 2400K = 0$$

$$100K^2 - 400K = 0$$

$$100K(K - 4) = 0$$

Isso nos dá $K = 0$ e $K = 4$.

iv. **Encontrando o Ponto de Cruzamento:** Usamos a equação da linha s^2 (a equação auxiliar) com $K = 4$:

$$A(s) = (20 + 4)s^2 + 2400(4) = 0$$

$$24s^2 + 9600 = 0$$

$$s^2 = -400 \Rightarrow s = \pm j20$$

Por fim, como já encontramos o Break-In no item anterior, agora é necessário encontrar o Breakaway.

O Breakaway é o ponto no eixo real onde dois ou mais ramos do Root Locus, que se moviam em direções opostas, ao longo do eixo, colidem e saem do eixo para o plano complexo.

Como já foi mencionado, o Breakaway se encontra em $s = 0$ porque é aonde os ramos do polo duplicado divergem do eixo real, por sua vez, isso se dá porque o intervalo de 0 a -20 é inválido para o sistema, obrigando a "fuga" dos ramos.

2. For $K = 20$

(a) Find the steady state error, $e(\infty)$, as a function of a for the following test inputs (i) $r(t) = 2$; (ii) $r(t) = 3t$; and (ii) $r(t) = t^2$.

4.0.3 Resposta:

Primeiro, o erro estacionário é definido como o erro encontrado após a estabilização do sistema, significando o quão preciso um sistema de controle é. Em consequência da existência do erro estacionário, o Teorema do Valor Final é usado para calcular o comportamento de um sistema quando o tempo tende ao infinito e enuncia:

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s)$$

Em seguida, é necessário identificar os tipos de sistema e os tipos de constantes de erro. Existem 3 tipos de sistema:

Tipo 0 (sem integrador): Apresenta entrada constante, sempre apresenta erro.

Tipo 1 (1 integrador): Apresenta integrador que consegue mitigar erros de entrada constante (degrau).

Tipo 2 (2 integradores): Apresenta 2 integradores que tentam conseguem mitigar erros de degrau e rampa.

Já as constantes podem ser definidas como K_p para a posição, K_v para a velocidade e K_a para a aceleração. São medidas da capacidade de um sistema de anular o erro. Quanto maior a constante, menor o erro. O sistema referenciado pela questão é um sistema do tipo 2, que só apresenta erro em aceleração, sabendo disso, podemos definir:

$$K_v, K_p = \infty$$

Sendo assim, para os itens *i*) e *ii*), ambos os erros são nulos. Entretanto, para o item *iii*), o erro será constante e finito e dado por:

$$e(\infty) = \frac{A}{K_a}$$

$$K_a = \lim_{s \rightarrow 0} s^2 G(s) = \lim_{s \rightarrow 0} s^2 \left[\frac{20(s+a)(s+60)}{s^2(s+20)} \right]$$

Podemos cancelar os termos s^2 :

$$K_a = \lim_{s \rightarrow 0} \frac{20(s+a)(s+60)}{s+20}$$

Agora, substituímos $s = 0$ na expressão:

$$K_a = \frac{20(0+a)(0+60)}{0+20} = \frac{20 \cdot a \cdot 60}{20} = 60a$$

Finalmente, calculamos o erro usando $A = 2$ e $K_a = 60a$:

$$e(\infty) = \frac{A}{K_a} = \frac{2}{60a} = \frac{1}{30a}$$

- (b) With exact calculations, verify on the root locus (i) the point and gain where the locus crosses the $j\omega$ -axis; (ii) the breakaway and break-in points on the real axis.
- (c) With exact calculations, (i) define the position of the asymptotes; (ii) define the point and gain where the root locus crosses the $\zeta = 0.175$ line; (iii) discuss the validity of the second order approximation

4.0.4 Resposta:

$$G_{eq}(s) = \frac{20(s+60)}{s(s^2 + 40s + 1200)}$$

Os polos desta função equivalente são $p_1 = 0$, $p_{2,3} \approx -20 \pm j28.28$ e o zero é $z_1 = -60$.

(i) Posição das Assíntotas

As assíntotas indicam o comportamento dos ramos para valores muito grandes do parâmetro ' a '.

- i. Número de Assíntotas: $n - m = (\text{número de polos}) - (\text{número de zeros}) = 3 - 1 = 2$.
- ii. Ângulos das Assíntotas: $\theta_k = \frac{(2k+1)180^\circ}{n-m}$ para $k = 0, 1$.

- o Para $k = 0 : \theta_0 = \frac{180^\circ}{2} = 90^\circ$
- o Para $k = 1 : \theta_1 = \frac{3 \cdot 180^\circ}{2} = 270^\circ$

iii. **Centroide (Ponto de Interseção no Eixo Real):**

$$\sigma_a = \frac{\sum(\text{polos}) - \sum(\text{zeros})}{n - m}$$

$$\sigma_a = \frac{(0 - 20 - 20) - (-60)}{2} = \frac{-40 + 60}{2} = 10$$

Conclusão: O sistema possui **duas assíntotas** com ângulos de **90° e 270°** que se cruzam no eixo real no ponto $\sigma_a = +10$.

(ii) **Ponto e Ganho no Cruzamento com $\zeta = 0.175$**

- i. Equação Característica (em função de 'a'): A equação do sistema é $s^3 + 40s^2 + 1200s + a(20s + 1200) = 0$.
- ii. Solução Numérica: Resolvendo a equação para um polo s sobre a linha de $\zeta = 0.175$, encontramos:
O ponto de cruzamento ocorre em:

$$s \approx 13.11 \pm j73.76$$

O valor do parâmetro 'a' (o "ganho") que posiciona os polos neste ponto é:

$$a \approx 133.9$$

(iii) **Validade da Aproximação de Segunda Ordem**

A validade da aproximação é analisada no ponto de cruzamento.

- i. **Localização dos Polos** para $a \approx 133.9$:

O par dominante está em $p_{1,2} \approx 13.11 \pm j73.76$. O terceiro polo (p_3) é encontrado pela soma das raízes:

$$p_1 + p_2 + p_3 = -40 \implies 26.22 + p_3 = -40 \implies p_3 \approx -66.22$$

- ii. **Discussão da Validade:**

A aproximação de segunda ordem é inválida porque o par de polos "dominante" está no semiplano direito ($Re(s) > 0$), o que torna a resposta do sistema instável (cresce exponencialmente) e o terceiro polo em $p_3 \approx -66.22$

está muito próximo do zero do sistema original em $s = -60$.

Conclusão: A aproximação de segunda ordem não é válida. A instabilidade do par de polos complexos é o fator que domina o comportamento do sistema.