

Modelagem de Sistemas Dinâmicos - Trabalho N^o4

Leonardo Soares da Costa Tanaka - DRE: 121067652

Engenharia de Controle e Automação/UFRJ

Rio de Janeiro, Brasil

Julho de 2023

Para este trabalho, vamos utilizar o arquivo “trabalho4-2023-1.mat” que tem os sinais de entrada $u(t)$ e de saída $y(t)$ de um sistema linear contínuo com função de transferência $G(s)$. Os sinais u e y foram aplicados e aquisitados com uma frequência de amostragem $f_s = 2\text{Hz}$ (período de amostragem $T = 0.5\text{s}$). A variável independente tempo t é o vetor com os instantes que foram realizadas as amostragens dos sinais $u(t)$ e $y(t)$.

Vale notar que o sinal de saída $y(t)$ está quantizado e contaminado com ruído.

1 FFT

Determinando, utilizando a FFT (Fast Fourier Transform), o espectro do sinal de entrada (módulo e fase) em função da frequência em Hz. Utilizando Python e as bibliotecas disponíveis (NumPy, Matplotlib e Scipy) que auxiliam na utilização da FFT, na plotagem dos espectros e na coleta dos dados do arquivo .mat.

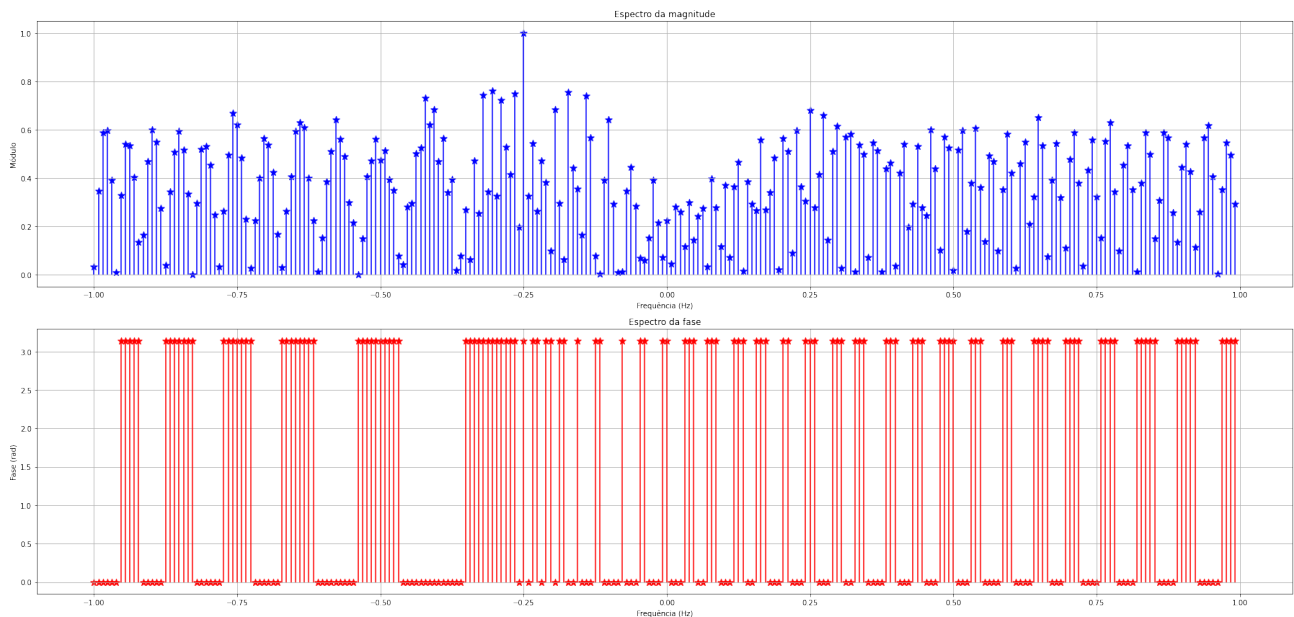


Figura 1: Espectros dos sinais da entrada

É possível observar a magnitude que varia entre 0 e 1 com uma maior concentração de amostras abaixo de 0.6, um padrão de oscilação mais lento na parte esquerda do espectro e um padrão de oscilação mais rápido na parte direita do espectro. Já no espectro de fase, os valores ficaram variando entre dois valores 0 e π com um padrão de oscilação mais lento na parte esquerda do espectro e um padrão de oscilação mais rápido na parte direita do espectro.

```
# Importando as bibliotecas
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import loadmat

data = loadmat('trabalho4-2023-1.mat') # Carregar os dados do arquivo
u = np.array(data['u'])
y = np.array(data['y'])
t = np.array(data['t'])

# Frequencia de amostragem (fs) e período de amostragem (T)
fs = 2 # Hz
T = 1 / fs

U = np.fft.fft(u) # Calcula o espectro do sinal de entrada u(t) usando a FFT
frequencias = np.fft.fftfreq(len(U), d=T) # Vetor de frequencias para o eixo x
modulo_U = np.abs(U) # Modulo do espectro do sinal de entrada
fase_U = np.angle(U) # Fase do espectro do sinal de entrada

# Plot do espectro do sinal de entrada (módulo e fase)
plt.figure(figsize=(25, 12))

# Plot do modulo com estrelas e linhas para o eixo x
plt.subplot(2, 1, 1)
plt.scatter(frequencias, modulo_U, marker='*', s=100, color='b')
plt.vlines(frequencias, ymin=0, ymax=modulo_U, colors='b')
plt.xlabel('Frequencia (Hz)')
plt.ylabel('Módulo')
plt.title('Espectro da magnitude')
plt.grid()

# Plot da fase com estrelas e linhas para o eixo x
plt.subplot(2, 1, 2)
plt.scatter(frequencias, fase_U, marker='*', s=100, color='r')
plt.vlines(frequencias, ymin=0, ymax=fase_U, colors='r')
plt.xlabel('Frequencia (Hz)')
plt.ylabel('Fase (rad)')
plt.title('Espectro da fase')
plt.grid()

plt.tight_layout()
plt.show()
```

2 Resposta em frequência do sistema $G(j\omega)$

Estimando a resposta em frequência do sistema $G(j\omega)$ utilizando os espectros dos sinais de entrada $U(j\omega)$ e de saída $Y(j\omega)$:

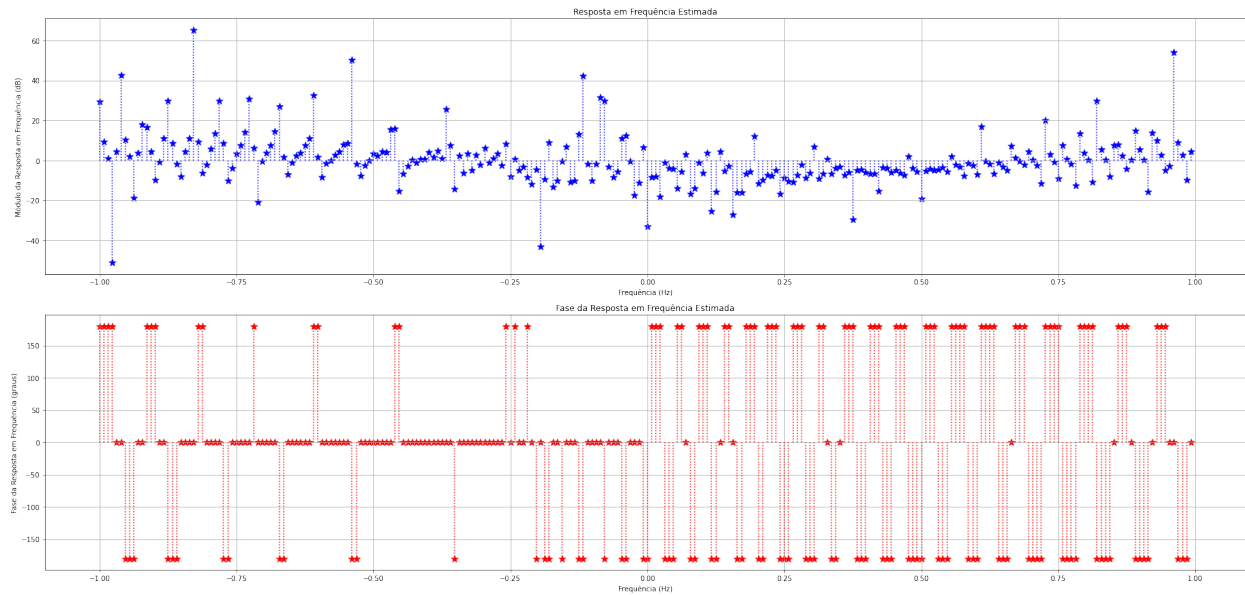


Figura 2: Resposta em frequência

```
Y = np.fft.fft(y) # Calcula o espectro do sinal de saída y(t) usando a FFT
G = Y / U # Calcula a resposta em frequência estimada G(jw)
modulo_G_dB = 20 * np.log10(np.abs(G)) # Calcula o modulo em dB da resposta em frequência estimada
fase_G_graus = np.angle(G) * (180 / np.pi) # Calcula a fase em graus da resposta em frequência estimada
plt.figure(figsize=(25, 12)) # Plot dos graficos usando subplot
plt.subplot(2, 1, 1) # Plot do modulo com estrelas e linhas para o eixo x (resposta em frequência estimada - modulo)
plt.scatter(frequencies, modulo_G_dB, marker='*', s=100, color='b')
plt.vlines(frequencies, ymin=0, ymax=modulo_G_dB, colors='b', linestyle='dotted')
plt.xlabel('Frequencia (Hz)')
plt.ylabel('Modulo da Resposta em Frequencia (dB)')
plt.title('Resposta em Frequencia Estimada')
plt.grid()
plt.subplot(2, 1, 2) # Plot da fase com estrelas e linhas para o eixo x (resposta em frequência estimada - fase)
plt.scatter(frequencies, fase_G_graus, marker='*', s=100, color='r')
plt.vlines(frequencies, ymin=0, ymax=fase_G_graus, colors='r', linestyle='dotted')
plt.xlabel('Frequencia (Hz)')
plt.ylabel('Fase da Resposta em Frequencia (graus)')
plt.title('Fase da Resposta em Frequencia Estimada')
plt.grid()
plt.tight_layout()
plt.show()
```