EA614 - Análise de Sinais

Atividade 4 - Amostragem e DFT

Iran Seixas Lopes Neto - RA: 244827
Letícia Lopes Mendes da Silva - RA: 184423

Amostragem e *Aliasing*

Para resolver os itens da atividade, utilizamos códigos em Python com as bibliotecas a seguir, incluindo as rotinas fornecidas em `espectro.py` e `kaiser.py`:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy.io as sio
import IPython.display as ipd
from espectro import espectro
from kaiser import kaiser
```

Item (a)

Com o comando `wavfile.read` do `SciPy`, realizamos a leitura do arquivo de áudio e transformamos o sinal $y[n]$ obtido em um sinal mono, para assim gerar o seu espectro:

```
Fs, y = sio.wavfile.read('creed_overcome.wav')
y = (y[:,0] + y[:,1]) / 2
Y, w = espectro(y)
```

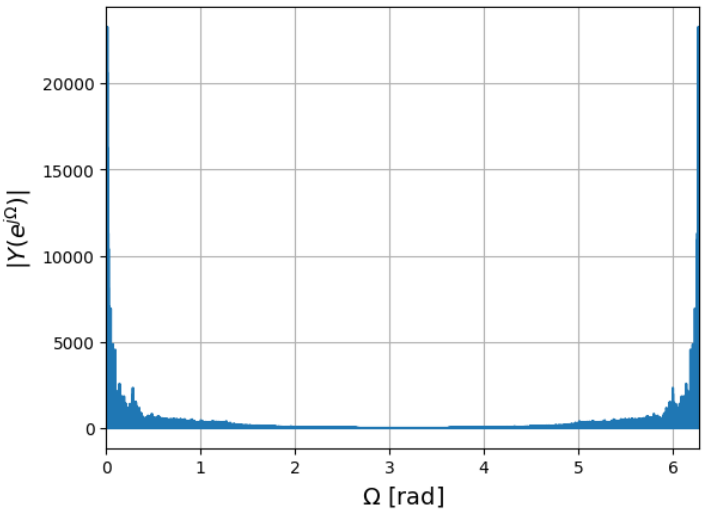


Figura 1: Espectro de frequências do sinal de áudio original ($y[n]$).

Observando a Fig. 1, percebe-se que o sinal tem maiores valores de magnitude em baixas frequências. Podemos notar isso ao relacionar as frequências digitais Ω [rad] com as frequências analógicas f [Hz], usando a taxa de amostragem $F_s = 44.1$ kHz. Sabemos que:

$$\Omega = \omega T_s = 2\pi f \frac{1}{F_s} \Rightarrow \boxed{f = \frac{\Omega F_s}{2\pi}} \tag{1}$$

Nas frequências digitais até $\frac{\pi}{3}$ rad, onde o sinal tem maiores valores, temos que o sinal está mais presente nas frequências analógicas até 7.35 kHz, que, se comparado a F_s, são baixas frequências. Além disso, vale notar que estas frequências estão dentro da faixa de 20 Hz a 20 kHz, que são os limites da percepção auditiva humana.

Item (b)

Em seguida, usamos um fator de $M = 6$ para reduzir a taxa de amostragem (processo de decimação) de $y[n]$. A linha `y_dec = y[::M]` seleciona uma amostra a cada M amostras de $y[n]$, descartando as $M-1$ amostras seguintes. Então, geramos o espectro do sinal decimado.

```
M = 6
y_dec = y[::M]
Y_dec, w_dec = espectro(y_dec)
```

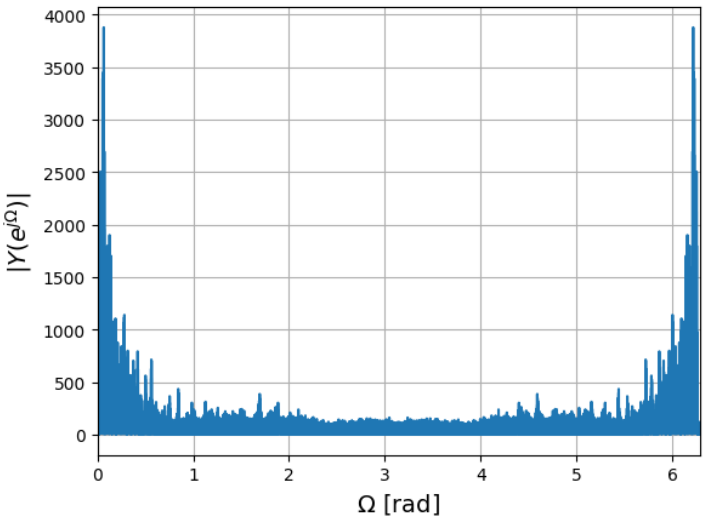


Figura 2: Espectro de frequências do sinal de áudio decimado ($y_{dec}[n]$).

Ao analisar o espectro do sinal decimado (Fig.2), é possível notar que a magnitude do espectro possui um máximo valor de amplitude bem menor, apenas próximo de 4000, se comparado com o sinal original, que possui o máximo maior que 20000. Além disso, observa-se que as magnitudes das frequências altas foram redistribuídas, alterando significativamente o conteúdo espectral do sinal original. Isso pode ser notado ao gerar os áudios usando o comando `Audio`, do `IPython.display`:

```
ipd.Audio(y, rate=Fs)        # Sinal original
ipd.Audio(y_dec, rate=Fs/M) # Sinal decimado
```

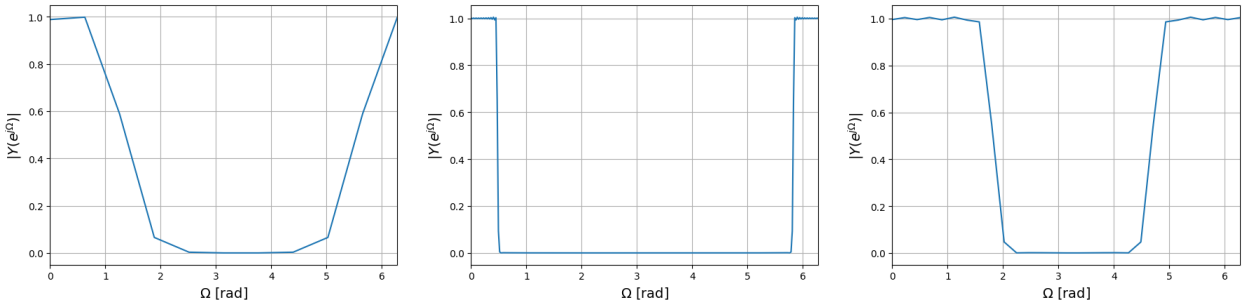
Ao ouvir ambos os áudios, percebemos que há uma clara distorção causada pela decimação do sinal, com perda de detalhes sonoros e distorção, provavelmente causada pela interferência de *aliasing*.

Item (c)

Neste item, utilizamos o método da janela de *Kaiser*, da rotina `kaiser`, para gerar três filtros passa-baixas (FPBs), cada um com frequências de passagem e de rejeição diferentes:

- Caso 1: $\Omega_p = 0.45$ rad e $\Omega_r = 2$ rad
- Caso 2: $\Omega_p = 0.45$ rad e $\Omega_r = 0.5$ rad
- Caso 3: $\Omega_p = 1.5$ rad e $\Omega_r = 2$ rad

```
h1 = kaiser(0.45, 2)
Y1,w1 = espectro(h1)    # Caso 1
h2 = kaiser(0.45, 0.5)
Y2,w2 = espectro(h2) # Caso 2
h3 = kaiser(1.5, 2)
Y3,w3 = espectro(h3)     # Caso 3
```



(a) Caso 1 (b) Caso 2 (c) Caso 3

Figura 3: Resposta em frequência para os casos 1, 2 e 3 analisados, respectivamente.

Analisando os três FPBs da Fig. 3, podemos destacar como principal distinção a faixa de transição de cada um. No filtro 3a, vemos que a diferença entre as frequências de passagem e de rejeição é maior que nos dois casos, e por isso possui uma maior faixa de transição. Por conta disso, esse filtro não está próximo a um FPB ideal. Já o filtro 3c possui uma frequência de passagem bem grande comparado com os outros dois, o que pode não ajudar a resolver o problema de *aliasing*. Por isso, o filtro 3b é o melhor filtro entre os três, tanto por possuir uma menor faixa de transição, como também por possuir uma frequência de passagem pequena.

Item (d)

Utilizando o filtro $h_2[n]$, do caso 2, utilizamos a rotina `convolve` do `Numpy` para convoluir $y[n] * h_2[n]$ e obter o sinal filtrado $z[n]$.

```
z = np.convolve(y, h2)  # z[n] = y[n] * h_2[n]
Z, w = espectro(z)
```

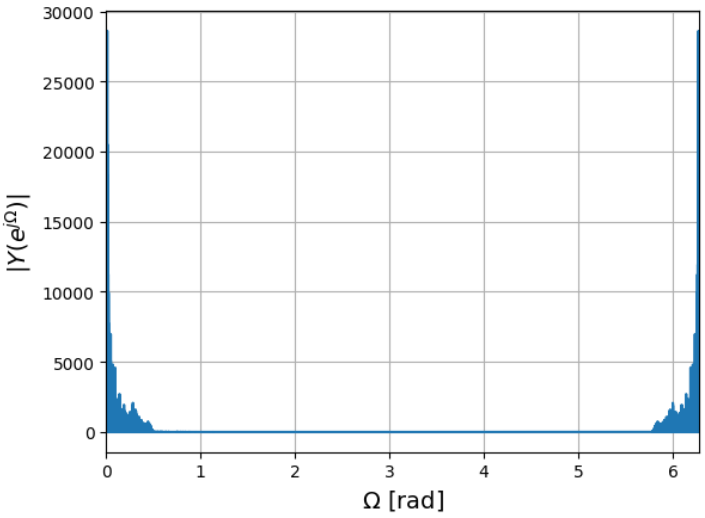


Figura 4: Espectro de frequências do sinal de áudio original filtrado ($z[n]$).

No sinal filtrado (Fig. 4), as faixas de alta frequência foram significativamente atenuadas, enquanto as baixas frequências, dentro da faixa de passagem de interesse (de 0 a 0.45 rad, estabelecida por $h_2[n]$), foram preservadas. Dessa forma, o sinal filtrado mantém a maioria das informações relevantes do áudio, mas elimina harmônicos de altas frequências que poderiam causar *aliasing*. Apesar disso, ainda há uma faixa de frequência perdida pela filtragem, que pode ser notada ao ouvir o áudio gerado:

```
ipd.Audio(z, rate=Fs)    # Sinal filtrado
```

A distorção causada em $z[n]$ provavelmente se deve à perda das frequência entre 0.45 e $\frac{\pi}{3}$ rad.

Item (e)

Após a filtragem do sinal original, novamente subamostramos o sinal por um fator $M = 6$:

```
M = 6
z_dec = z[::M]
Z_dec, w_dec = espectro(z_dec)
```

Da Fig. 5, vemos que o espectro filtrado e decimado $z_{dec}[n]$ apresenta uma representação mais fiel das baixas frequências se comparado ao espectro filtrado diretamente do áudio

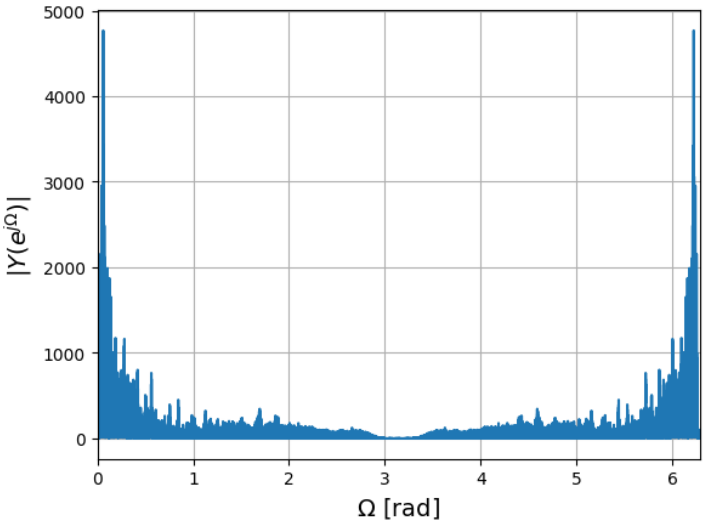


Figura 5: Espectro de frequências do sinal de áudio original filtrado e decimado ($z_{dec}[n]$).

original $y_{dec}[n]$, como mostrado na Fig. 2. Isto é, a filtragem prévia removeu frequências que causariam distorção por aliasing, resultando em um sinal menos degradado. Podemos verificar isso analisando o novo áudio decimado:

```
ipd.Audio(z_dec, rate=Fs/M) # Sinal decimado
```

Ao comparar os áudios $z[n]$ e $z_{dec}[n]$, percebemos que eles são muito semelhantes, sem aparente interferência de *aliasing*, o que confirma o resultado esperado pela filtragem.

DFT e Identificação de Frequências

Item (f)

Nosso objetivo é gerar o espectro de magnitude do EEG para identificar a frequência analógica f_e: a frequência do estímulo visual ao qual o indivíduo estava submetido.

Para isso, primeiro, realizamos a leitura do arquivo `EEG.csv` usando o comando `read_csv` do `pandas`, obtendo o sinal $x[n]$. Em seguida, calculamos a magnitude da DFT do sinal $X[k]$, usando o comando `fft.fft` do `NumPy`, e depois limitamos o sinal apenas à sua primeira metade, dada sua simetria. Além disso, definimos o intervalo de frequências analógicas até a metade da taxa de amostragem (0 a 125 Hz).

Como a DFT retorna valores para índices k, que representam amostras no domínio da frequência digital, podemos usar o comando `argmax` do `NumPy` para encontrar o índice do primeiro pico de maior magnitude de $X[k]$, pois esta também é a posição da frequência f_e. Para encontrar podemos usar a relação entre a frequência digital e o índice ($\Omega = \frac{2\pi}{N}k$). Usando a Eq. 1, podemos associar os valores de frequência analógicos ao índice k:

$$f = \frac{2\pi k}{N} \cdot \frac{f_s}{2\pi} = \frac{k f_s}{N}$$

A partir disso, foi possível descobrir o valor de f_e:

$$f_e = \frac{k_{max} f_s}{N}$$

```
# Leitura dos dados do arquivo EEG.csv
eeg = pd.read_csv('EEG.csv')
x = eeg.iloc[:, 1].values   # x[n]

# Calculo do espectro de magnitude do sinal EEG
X = np.abs(np.fft.fft(x))   # |X[k]| = |DFT{x[n]}|

# Intervalo de frequencias
fs = 250    # Taxa de amostragem (Hz)
N = len(X)  # Numero de amostras
X = X[:N//2]
f = np.linspace(0, fs//2, N//2)

# Frequencia do pico
k_max = np.argmax(X)
fe = (k_max * fs) / N

# Visualizacao do espectro de magnitude do EEG
plt.plot(f, X, label="|X[k]|")
plt.axvline(fe, color="r", ls="--", label=rf"Pico em $f_e$ = {fe:.2f} Hz")
plt.xlabel("f (Hz)", fontsize=12)
plt.ylabel("Magnitude", fontsize=12)
plt.legend()
plt.grid()
plt.show()
```

Analisando o gráfico da Fig. 6, vemos claramente o pico em $f_e = 11.67$ Hz, que é a frequência do estímulo visual. Essa análise demonstra como a DFT pode ser usada para identificar padrões de frequência em sinais biológicos.

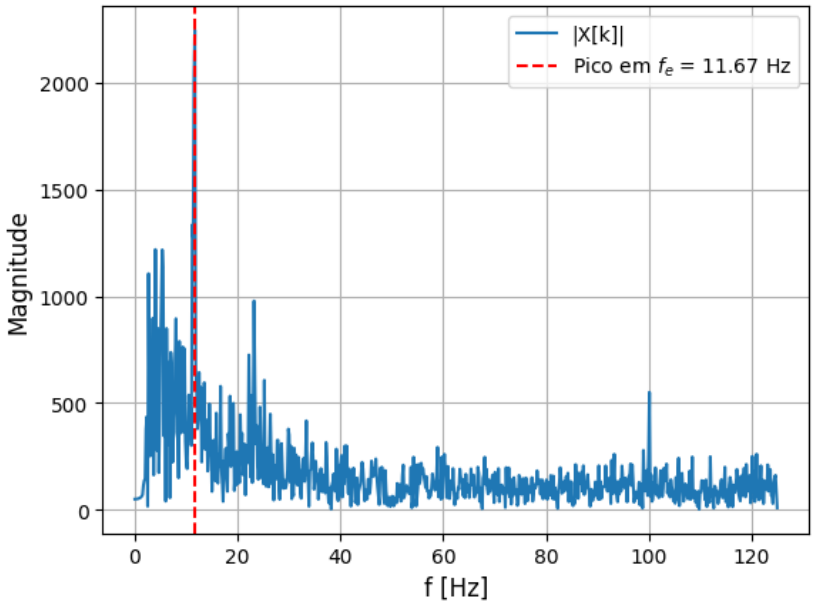


Figura 6: Espectro de magnitude do sinal registrado em EEG em termos de frequências analógicas.