

▼ CSM QS2020 - Minami

LAB5 - Percepção Auditiva e Básico de Áudio

```
import numpy as np
import math
import matplotlib.pyplot as plt
import IPython.display
```

Mascaramento de Frequências

```
fs = 22050 # Frequência de amostragem
f1 = 2000 # Frequência do sinal 1 em Hz
f2 = 1200 # Frequência do sinal 2 em Hz
tf1 = np.linspace(0, 0.2*fs, num=np.int16(0.2*fs)) # Intervalo de 200ms
tf2 = np.linspace(0, 0.2*fs, num=np.int16(0.2*fs))
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)

```
plt.plot(tf1[1:200]/fs, h1[1:200]); plt.title('Trecho do Sinal 1')
```

```
plt.plot(tf2[1:200]/fs, h2[1:200]); plt.title('Sinal 2')
```

```
# Ouvindo Sinal 1
IPython.display.Audio(data=h1, rate=fs)
```

```
# Ouvindo Sinal 2
IPython.display.Audio(data=h2, rate=fs)
```

```
# criando intervalo de silêncio entre os dois sinais
sil = np.zeros((1,2205)) # 100 ms de silêncio
```

Sinal com frequência f1 mascarando f2

```
import time
# Gradiente, fator de 3dB
fat3dB = math.pow(10,0.3)
fat = 1
Nloop = 8
atenuacao = np.linspace(0, -24, num=Nloop+1)
```

```
# Atenuando tom com f1 = 1.2kHz, que é o tom a ser mascarado
# intervalo de 5s entre um teste e outro
s1 = np.concatenate( (h1, sil), axis=None)
s2 = np.concatenate( (h2, sil), axis=None)
for i in range(0, Nloop+1):
    fmask1 = np.concatenate( (s1, s2), axis=None)
    print("{} Atenuação de tom de {}Hz por {:.1f} dB".format(i+1,f2,atenuacao[i]))
    padrao = IPython.display.Audio(data=fmask1, rate=fs, autoplay=True)
    IPython.display.display(padrao)
    time.sleep(5)
    fat = fat*fat3dB
    s2 = s2/fat
print("==> PODE TOCAR NO PADRÃO DESEJADO, PARA CONFERIR")
```

Sinal com frequência f2 mascarando f1

```
# Invertendo f1 com f2
# Atenuando tom de f2 = 2kHz, que é o tom a ser mascarado
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)

```
fat = 1
Nloop = 8
atenuacao = np.linspace(0,-24,num=Nloop+1)
s1 = np.concatenate( (h2, sil), axis=None)
s2 = np.concatenate( (h1, sil), axis=None)
for i in range(0, Nloop+1):
    fmask1 = np.concatenate( (s1, s2), axis=None)
    print("{} Atenuação de tom de {}Hz por {:.1f} dB".format(i+1,f1,atenuacao[i]))
    padrao = IPython.display.Audio(data=fmask1, rate=fs, autoplay=True)
    IPython.display.display(padrao)
    time.sleep(5)
    fat = fat*fat3dB
    s2 = s2/fat
print("==> PODE TOCAR NO PADRÃO DESEJADO, PARA CONFERIR")
```

FIM

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)