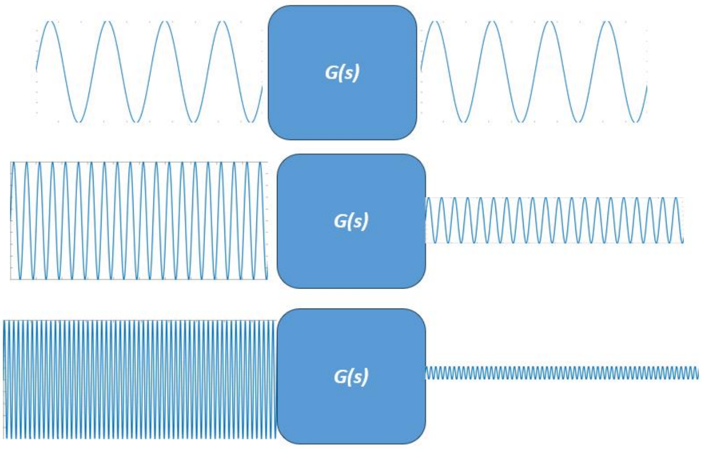
PROJETO 7 – Filtros Digitais (equalizador digital de áudio)

Um filtro digital é um sistema que processa sinais digitais com o objetivo de modificar ou extrair certas características do sinal, como remover ruído, destacar uma faixa de frequências ou limitar a largura de banda. Ele atua sobre uma sequência de amostras (valores discretos no tempo do sinal digitalizado) e produz uma nova sequência como resultado. Sabemos que um sinal pode ser decomposto em muitas senoides, de diversas frequências, certo? Um filtro digital pode então ser visto como uma "peneira" que deixa passar apenas certas frequências que compõem o sinal (por exemplo, graves ou agudos) e bloqueia o resto. Adotando como critério as frequências que um filtro bloqueia e as que mantém ou amplifica, categorizamos os filtros digitais em quatro tipos:

- Filtros passa-baixa (low pass filter) – Elimina as frequências que compõem o sinal que estejam acima de um dado limite denominado frequência de corte f_{corte}.
- Filtros passa-alta (high pass filter) – Elimina as frequências qie compõem o sinal que estejam abaixo de um dado limite denominado frequência de corte f_{corte}.
- Filtros passa-faixa (band pass filter) – Elimina todas as frequências que estejam fora de uma faixa (intervalo) de frequências.
- Filtros rejeita-faixa (notch filter) – Elimina todas as frequências que estejam dentro de uma faixa (intervalo) de frequências.

- Filtro de picos – Amplifica ou atenua uma certa faixa de frequências em uma quantidade desejada.

Um filtro, por exemplo, passa-baixa, atenua sinais de altas frequências, tendo o comportamento de sistemas dinâmicos cujas saídas são atenuadas para entradas de frequências acima de um certo valor, denominada frequência de corte. A figura abaixo ilustra o comportamento do filtro.



Considere que temos um sinal $x(t)$ que foi digitalizado e salvo em uma variável como uma lista de valores $X[k]$ $(k = 0,1,2,3 \ldots, tamanho\ da\ lista)$. Como podemos aplicar um filtro a esse sinal a fim de eliminarmos ou amplificarmos determinadas faixas de frequências que o compõem? A resposta é: Criamos uma nova lista, denominada lista de saída $Y[k]$, que será o sinal original filtrado. O sinal original é considerado como a entrada, enquanto o sinal filtrado é considerado a saída do filtro. Cada valor da lista de saída é obtido através de uma fórmula que modifica os valores da lista referente ao sinal original. Essa fórmula determina que cada posição da lista de saída é na verdade uma combinação ponderada de valores da lista original. Um exemplo de filtro atuando no sinal original $X[k]$ e produzindo o sinal filtrado $Y[k]$ é:

$$Y[k] = 1{,}8\, Y[k-1] - 0{,}7\, Y[k-2] + 0{,}2\, X[k] + 0{,}3X[k-1] + 0{,}2\, X[k-2]$$

Repare que o sinal filtrado é uma combinação de valores anteriores de si próprio mais uma combinação de valores do sinal original. Dependendo dos coeficientes que escolhemos para multiplicar esses valores, o filtro atuará de um modo diferente. Veja como seria a implementação desse filtro:

```
def filtro_IIR_ordem2(x):
    y = [0]*len(x)
    for n in range(2, len(x)):
        y[n] = (1.5 * y[n-1]) - (0.7 * y[n-2]) + (0.2 * x[n]) + (0.3 * x[n-1]) +
    return y
```

Ainda considerando o exemplo acima, podemos reescrever a equação avançando 2 posições em todos os termos. Se a equação vale pra k, também vale pra k+2:

$$Y[k+2] = 1{,}8\, Y[k+1] - 0{,}7\, Y[k] + 0{,}2\, X[k+2] + 0{,}3X[k+1] + 0{,}2\, X[k]$$

Iremos agora usar um operador bastante presente em sistemas digitais: o operado Z. Esse operador significa “avanço” na posição de uma lista. Ou seja, $Z \cdot Y[k] = y[k+1]$. Ou ainda $Z \cdot Z \cdot Y[k] = Z^2 Y[K] = y[k+2]$.

Utilizando o operador Z, nosso filtro exemplo ficaria:

$$Z^2 Y[K] = 1{,}8\, Z\, Y[K] - 0{,}7\, Y[k] + 0{,}2\, Z^2\, X[K] + 0{,}3\, Z\, X[K] + 0{,}2\, X[k]$$

Manipulando-se a equação:

$$Y[K]\,(Z^2 - 1{,}8\, Z + 0{,}7) = X[K]\,(0{,}2\, Z^2 + 0{,}3\, Z + 0{,}2)$$

E finalmente

$$\frac{Y[K]}{X[K]} = \frac{0{,}2\, Z^2 + 0{,}3\, Z + 0{,}2}{Z^2 - 1{,}8\, Z + 0{,}7}$$

Temos então uma maneira de escrever nosso filtro como uma relação entre entrada e saída. Muito parecido com as funções de transferências que você estuda em controle, certo? Sim! O filtro digital nada mais é que uma função de transferência discreta.

$$G(Z) = \frac{Y[K]}{X[K]} = \frac{0{,}2\, Z^2 + 0{,}3\, Z + 0{,}2}{Z^2 - 1{,}8\, Z + 0{,}7}$$

Na verdade, existe uma relação entre uma função de transferência contínua, no domínio de Laplace e uma função discreta, no domínio de Z. Basta lembrarmos de que s indica uma derivada, que de maneira discreta tem como aproximação $\frac{X[k+1]-X[k]}{T}$, ou seja, $\frac{Z\,X[K]-X[k]}{T}$.

As teorias utilizadas para se encontrar os coeficientes corretos para que o filtro funcione como desejado são vastas e estão fora dos nossos objetivos do curso. Felizmente, podemos recorrer a bibliotecas Python, Matlab, ou mesmo ao chat gpt para nos dar funções de transferências de filtros desejados.

Veja como o Python pode nos fornecer, por exemplo, os coeficientes de um filtro que elimina as frequências superiores a 500 Hz presentes num sinal de entrada, ou seja, um filtro passa baixas:

```
from scipy.signal import butter

# Parâmetros desejados
fs = 44100          # Taxa de amostragem (Hz)
fc = 500            # Frequência de corte (Hz)
ordem = 2           # Ordem do filtro

# Filtro passa-baixas Butterworth
b_lp, a_lp = butter(N=ordem, Wn=fc / (fs / 2), btype='low')

b_lp, a_lp
```

```
(array([0.00120741, 0.00241481, 0.00120741]),
 array([ 1.        , -1.89933342,  0.90416304]))
```

O Filtro fornecido pelo Python é do tipo Butterworth. Esse nome é dado em homenagem a seu inventor, Stephen Butterworth. Em termos de função de transferência, o filtro seria:

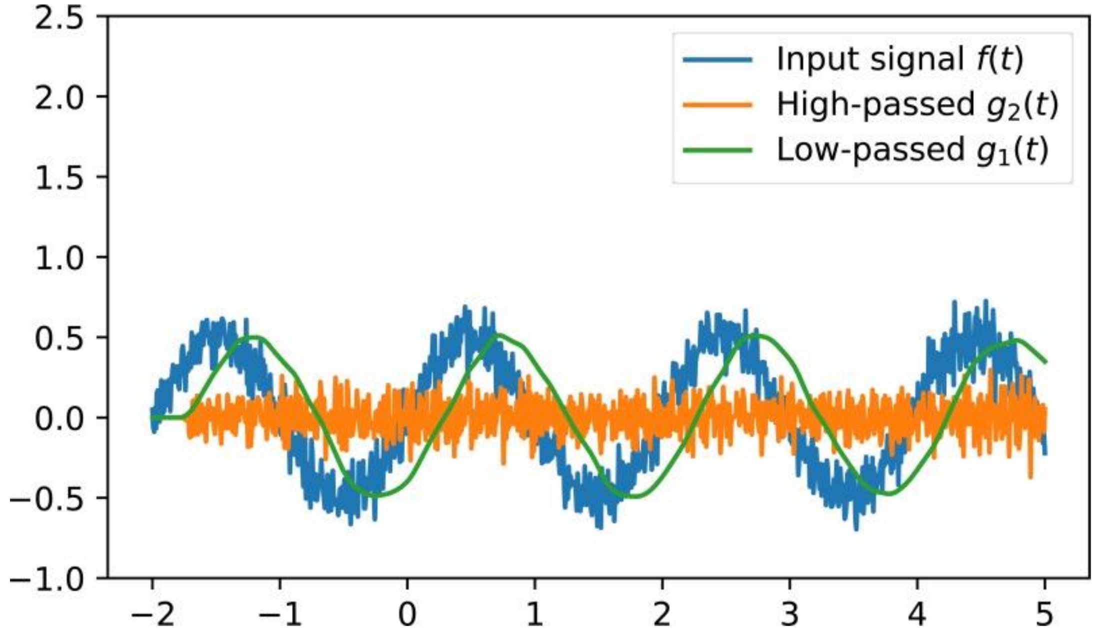
$$G(Z) = \frac{Y[K]}{X[K]} = \frac{0{,}00120741\, Z^2 + 0{,}00241481\, Z \; + 0{,}00120741}{Z^2 - 1{,}8993342\, Z + 0{,}90416304}$$

Lembre-se de que Z representa o operador avanço, e assim, com a função de transferência, podemos implementar o filtro para uma sequência de valores representando um sinal.

Exercício 1

Grave ou gere matematicamente com frequência de amostragem de 44100 Hz um sinal com alguns poucos segundos de duração que possua componentes (harmônicos) acima de 500Hz. Mostre o FFT desse sinal, evidenciando que ele possui componentes acima de 500Hz. Em seguida, implemente o filtro do exemplo anterior e faça a filtragem do sinal original. Mostre o FFT do sinal filtrado verificando que as frequências acima de 500 Hz foram atenuadas. Reproduza o áudio original e o filtrado. Consegue perceber a diferença? O que notou de diferente?

Os filtros passa-baixa são muito utilizados para eliminar ruídos de alta frequência contaminando um sinal de frequências mais baixas. Já os filtros passa-alta, teriam o efeito oposto, eliminando as baixas frequências de um sinal. Repare na figura a seguir o sinal original, filtrado por passa-baixa e filtrado por passa-alta:



Decibel - dB

O decibel (dB) é uma unidade logarítmica usada para expressar razões entre grandezas — principalmente em contextos de potência, amplitude, tensão, sinal de áudio, e resposta de sistemas. Apesar do nome, *decibel* não é uma unidade absoluta como "metros" ou "segundos". Ele representa uma razão relativa, geralmente comparando um valor com uma referência. A vantagem, por ser logarítmica, é que essa escala pode representar intervalos extensos. No seu dia a dia, certamente você ouve essa unidade quando se trata de intensidade sonora. Vamos entender o porquê ela é usada e como:

Para o ouvido humano produzir sensação auditiva, é necessária uma intensidade acústica de $10^{-12}\,\frac{W}{m^2}$. Em contrapartida, se a intensidade chegar a $1\,\frac{W}{m^2}$, começamos a sentir dores nos ouvidos e riscos de lesões. Repare como temos um intervalo extenso de intensidades! Ao invés de expressarmos uma dada intensidade acústica I em W/m^2 (que seria uma escala bastante inconveniente: de $10^{-12}\ a\ 1$) usamos uma relação chamada $Decibel$:

$$I_{dB} = 10 \cdot log\ \left(\frac{I}{10^{-12}}\right)$$

Ou seja, usamos o logaritmo da razão entre a intensidade e o limiar de audição multiplicado por 10. Assim a intensidade expressa em Decibel I_{dB} varia dentro do range 0 (limiar da audição) a 120 (limiar da dor). Por exemplo, vamos supor que que temos uma onda sonora com intensidade $10^{-4}\,\frac{W}{m^2}$. Em dB essa intensidade seria dada por

$$I_{dB} = 10 \cdot log\left(\frac{10^{-4}}{10^{-12}}\right) = 10 \cdot log\ (10^{8}) = 80dB$$

Exercício 2

Expresse em dB a intensidade limiar da audição $10^{-12}W/m^2$ e a intensidade limiar da dor $10^{0}W/m^2$

Decibel para sinais elétricos.

Quando estamos falando de sinais elétricos provenientes da transdução e assim podem ser digitalizados, a intensidade do sinal é proporcional ao quadrado da amplitude da tensão. Lembre-se que $P = UI = \frac{U^2}{R}$. Além disso, é bastante comum em processamento de sinais ou técnicas de controle, expressarmos a razão entre a intensidade de um sinal depois de ser processado (sinal de saída) e a intensidade do sinal antes de ser processado (sinal de entrada). Assim, a razão entre as intensidades de saída e a de entrada é denominada de *ganho* do processo e dada em dB na seguinte forma:

$$Ganho = 10 \cdot log\left(\frac{I_{saida}}{I_{entrada}}\right) = 10 \cdot log\left(\frac{V_{saida}^2}{V_{entrada}^2}\right) = 10 \cdot log\left(\frac{V_{saida}}{V_{entrada}}\right)^2 = 20 \cdot log\left(\frac{V_{saida}}{V_{entrada}}\right)$$

Assim, quando falamos de sinais de entrada e saída de um sistema, podemos expressar o ganho do sistema (ganho da função de transferência) da seguinte maneira:

$$G_{dB} = 20 \cdot log\left(\frac{V_{saida}}{V_{entrada}}\right)$$

A diferença em relação aos sinais acústicos, é apenas o fator 20 ao invés de 10.

Exercício 3

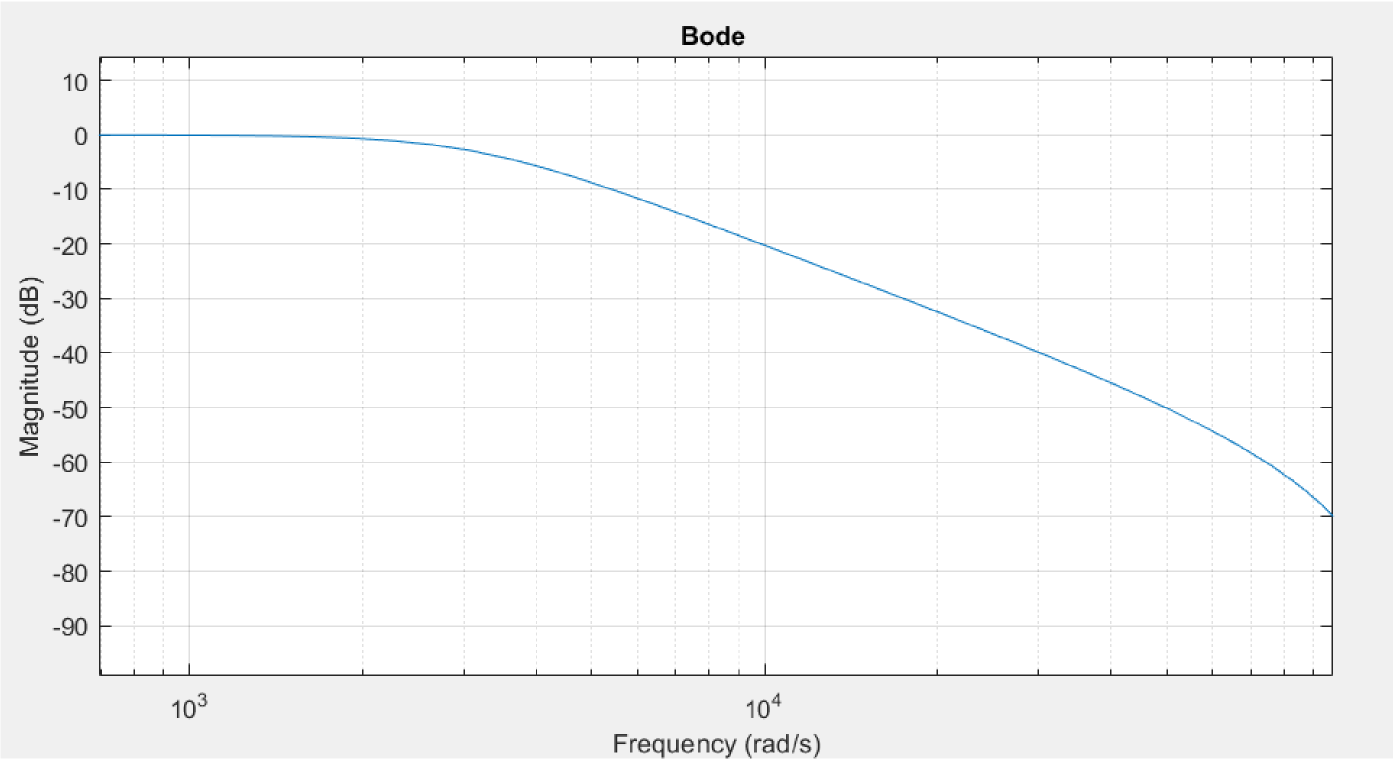
Um filtro tem como entrada um sinal de entrada dado por $U(t) = 20\ sen(\ 1000\pi\ t)$, sendo que o sinal de saída foi de $Y(t) = 10\ sen(\ 1000\pi\ t)$. Qual o ganho em Db desse filtro para esse sinal de entrada? (Obs.: Pensando em sinais senoidais, as tensões de entrada e saída são as amplitudes dos respectivos sinais!)

O Diagrama de Bode – resposta em frequência

Normalmente, o ganho de filtros ou outros sistemas lineares variam de acordo com a frequência do sinal de entrada. Ou seja, a relação entre as amplitudes do sinal de saída e de entrada varia de acordo com a frequência do sinal de entrada (a frequência do sinal de saída é sempre a mesma que o de entrada, pois estamos tratando de sistemas lineares). Assim, um gráfico muito útil é o denominado diagrama de Bode. Esse gráfico mostra o ganho em dB de um filtro (ou qualquer outra função de transferência representando um sistema linear) em função da frequência do sinal de entrada! Existe também uma defasagem entre os sinais de entrada e saída, mas por ora não será importante nesse curso. Repara no exemplo abaixo que mostra do diagrama de Bode do filtro utilizado no exercício 1:

$$G(Z) = \frac{Y[K]}{X[K]} = \frac{0{,}00120741\, Z^2 + 0{,}00241481\, Z\ + 0{,}00120741}{Z^2 - 1{,}8993342\, Z + 0{,}90416304}$$

O ganho em dB desse filtro varia com a frequência do sinal e entrada $U[K]$ de acordo com o diagrama de Bode abaixo:



Exercício 4a

Considere o filtro utilizado no exercício 1 e seu diagrama de Bode mostrado acima. Caso o sinal de entrada aplicado a esse filtro seja $U(t) = 50\ sen(\ 10^4\ t)$*, qual seria a amplitude do sinal de saída? E caso o sinal de entrada fosse* $U(t) = 50\ sen(\ 10^3\ t)$*, qual seria a amplitude o sinal de saída?*

Exercício 4b

Em caso de filtros passa-baixa, define-se a frequência de corte do filtro como a frequência em que se tem uma atenuação de -6dB. Estime a frequência de corte do filtro passa-baixo do exemplo. Qual seria a relação entre a amplitude de saída sobre amplitude de entrada para um sinal com frequência igual à frequência de corte?

Exercício 5

O código abaixo (filtro passa faixa.ipynb) fornecido via BB fornece a função de transferência discreta (G_Z) *de um filtro digital passa faixa. O filtro foi calculado para passar apenas frequências próximas de 2 kHz (frequência central). Além disso o código também faz diagrama de Bode do filtro. Rode o código e altere o valor do parâmetro Q (chamado fator de qualidade). Descreva como esse parâmetro atua.*

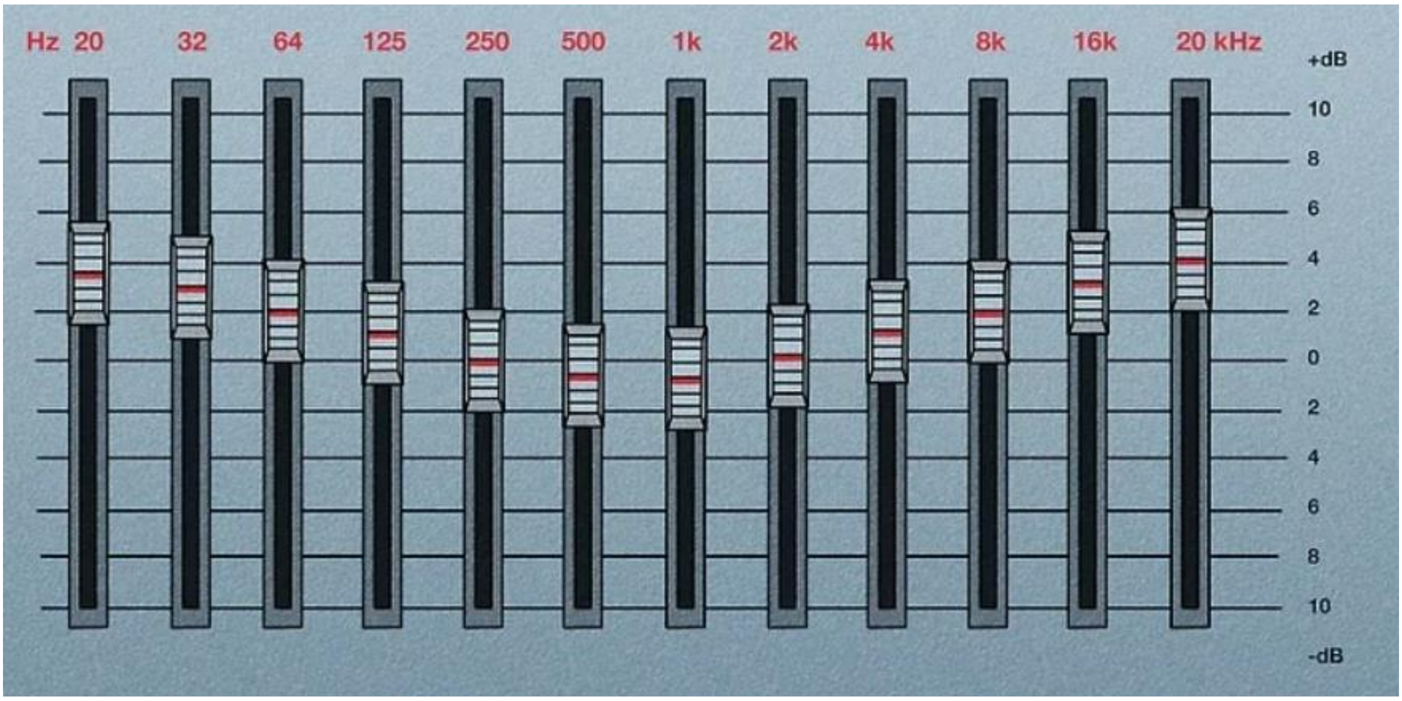
```python
import numpy as np
from scipy.signal import iirpeak, freqz, TransferFunction
import matplotlib.pyplot as plt

# Parameters
fs = 44100             # Sampling rate (Hz)
f0 = 2000             # Center frequency (Hz)
Q = 10                 # Quality factor

# Design band-pass filter using iirpeak (biquad)
b, a = iirpeak(w0=f0/(fs/2), Q=Q)  # Normalized frequency (f0 / Nyquist)

# Optional: Create TransferFunction object (discrete system)
# Note: 'dt=1/fs' makes it a discrete-time system
tf = TransferFunction(b, a, dt=1/fs)

print(tf)

# Frequency response plot  ... Bode
w, h = freqz(b, a, fs=fs)
plt.semilogx(w, 20 * np.log10(abs(h)))
plt.title('Band-Pass Filter Frequency Response')
plt.xlabel('Frequency [Hz]')
plt.ylabel('Magnitude [dB]')
plt.grid(which='both', linestyle='--', linewidth=0.5)
plt.axvline(f0, color='red', linestyle=':', label='Center Frequency')
plt.legend()
plt.tight_layout()
plt.show()

```

Exercício 6

O código abaixo (filtr0_notch.ipynb) fornecido via BB fornece a função de transferência discreta (G_Z) de um filtro digital rejeita faixa. O filtro foi calculado para rejeitar frequências próximas de 2 kHz (frequência central). Além disso o código também faz diagrama de Bode do filtro. Rode o código e altere o valor do parâmetro Q (chamado fator de qualidade). Descreva como esse parâmetro atua. Altere a frequência central. Você consegue pensar em alguma utilidade para o filtro?

Projeto 7 – Equalizador de áudio

Após a resolução dos exercícios propostos, voce deverá construir um equalizador de áudio capaz e atuar em 12 bandas de frequências (ou mais, se desejar). O usuário deverá poder configurar o filtro de modo a atenuar ou amplificar cada uma das frequências em diferentes níveis! A atenuação e a amplificação de cada uma das faixas deve estar entre -10B a +10dB e as frequências amplificadas ou atenuadas são as da figura abaixo.



Para obter a função de transferência discreta dos filtros utilizados, você poderá usar o código fornecido via BB filtro_peak_EQ.

Mantenha um curto áudio gravado que será usado para testes. Ao iniciar a aplicação, o usuário irá configurar o equalizador, definindo a amplificação ou atenuação de cada uma das bandas.

Seu código deve então de obter as funções de transferência coerentes com a configuração do equalizador. Após isso, deve aplicar os filtros um a um ao sinal.

O sinal resultante deve ser executado após o sinal original, a fim de observarmos a diferença entre os áudios!

O código deverá plotar o diagrama de Bode do filtro como um todo! Para isso você deverá pensar em como obter uma função de transferência que equivale a todas as funções individuais de cada banda!

Rubrica – entrega para 25/04

- Filtros implementados e atuando no sinal: C
- Interface com usuário bem construída, execução dos áudios: B
- Diagrama de Bode do filtro completo: A
- As notas intermediárias, como C+, B+ e A+ serão atribuídas mediante um contexto geral, considerando capricho e resolução dos exercícios.