

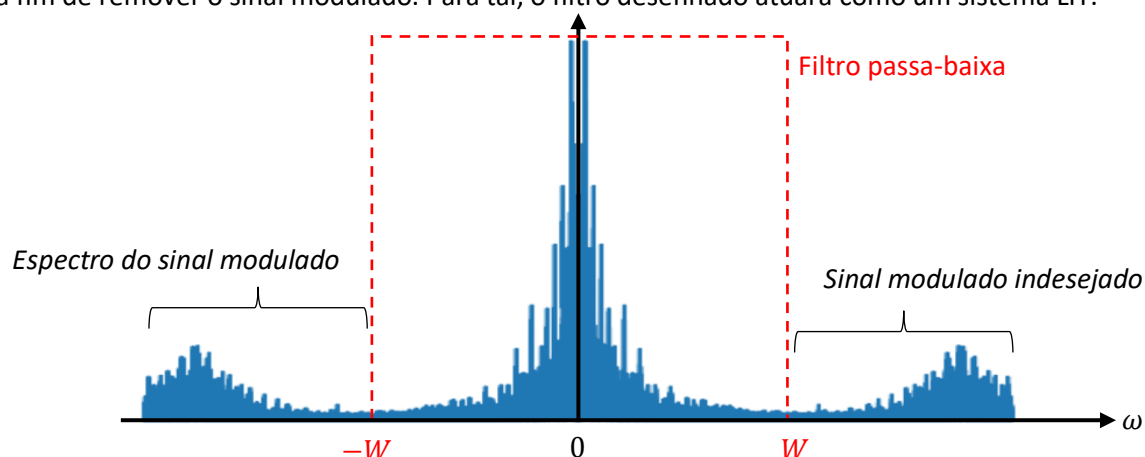
Projeto computacional de filtro – Transformada de Fourier de tempo discreto

Objetivo

Aplicar os conceitos estudados em sala e descritos no livro base da disciplina para o desenho de um filtro passa-baixa em ambiente computacional.

Descrição da atividade

A atividade se resume em filtrar um sinal de áudio composto por uma música somada por um sinal modulado (deslocado na frequência). O filtro passa-baixa deverá ser parametrizado através da escolha do parâmetro W , largura do filtro, a fim de remover o sinal modulado. Para tal, o filtro desenhado atuará como um sistema LIT.



Conceitos utilizados

- Transformada de Fourier de tempo discreto
- Aplicação da transformada em sinais periódicos e sua relação com a Série de Fourier
- Propriedades do deslocamento na frequência, da convolução e da multiplicação
- Sinais discretos no tempo
- Sistemas LIT discretos no tempo
- Resposta em frequência discreta
- Convolução linear entre sinais

Ferramentas a serem utilizadas

- Ambiente computacional para implementação de código (e.g., Python ou MATLAB)
- Algoritmo *Fast Fourier Transform* `fft()` para cálculo da resposta em frequência
- Algoritmo *Inverse Fast Fourier Transform* `ifft()` para cálculo do sinal no tempo
- Algoritmo de convolução linear
- Algoritmo de leitura/escrita de arquivos de áudio

Tarefas

Considere o arquivo de áudio disponibilizado no SIGAA:

- `ImperialPlusCantina.wav`

No ambiente computacional escolhido, leia o arquivo `.wav` para obter o sinal $x[n]$ em forma de vetor/arranjo.

- No Python, pode-se usar a função `read()` do módulo `scipy.io.wavfile`. Há outras possibilidades.
- No Python, $x[n]$ pode ser representado por um Numpy Array

- O arquivo de áudio, uma vez lido e representado por um vetor/arranjo, tem uma quantidade de pontos por segundo (taxa de amostragem). Pode ser necessário limitar o processamento para apenas alguns segundos iniciais.

Plote o gráfico do sinal $x[n]$. Depois calcule a Transformada de Fourier de $x[n]$ e plote o gráfico do espectro (magnitude). Observa-se que na região central do espectro há as componentes do sinal de áudio desejado, enquanto o sinal modulado tem suas componentes nas bordas devido ao deslocamento em frequência (modulação).

- OBS: É possível que o algoritmo `fft()` usado retorne o espectro na frequência com indexação espelhada. A sugestão é testar a utilização da função com um sinal conhecido antes.

Descreva a resposta ao impulso $h[n]$ cuja resposta em frequência seja um pulso retangular com largura W que atuará como um filtro passa-baixa.

Implemente $h[n]$ no ambiente computacional e plote o seu gráfico no tempo. Depois calcule a Transformada de Fourier de $h[n]$ e plote o gráfico do espectro (magnitude). Discorra sobre as diferenças observadas entre a descrição teórica e a implementação prática do filtro $h[n]$.

Calcule a resposta do filtro $y[n]$ para o sinal de áudio $x[n]$. Aqui, você deve usar:

- Método da convolução no tempo
- Método da multiplicação na frequência

Para o **método da convolução no tempo**, calcule a Transformada de Fourier de $y[n]$. Compare com a resposta do **método da multiplicação na frequência**. Comente.

Depois, calcule a Transformada Inversa de Fourier da resposta do **método da multiplicação na frequência**. Compare com a resposta do **método da convolução no tempo**. Comente.

- OBS: É possível que o algoritmo `ifft()` usado retorne o sinal no tempo com indexação espelhada. A sugestão é testar a utilização da função com um sinal conhecido antes.

Por fim, escreva o vetor/arranjo do sinal filtrado $y[n]$ em um arquivo `.wav`.

- No Python, pode-se usar a função `write()` do módulo `scipy.io.wavfile`. Há outras possibilidades.
- Para escrever o sinal processado em um arquivo `.wav`, é necessário passar como entrada a taxa de amostragem.

A **qualidade da filtragem** deve ser mensurada de forma subjetiva ao escutar o áudio filtrado. Tal qualidade está diretamente relacionada ao valor escolhido para a largura W do filtro. Comente os resultados para alguns valores W de sua escolha e indique qual o melhor valor obtido. A avaliação de qualidade pode ser feita da seguinte forma:

- Compare o sinal antes de ser processado (o mesmo que foi lido, ou a versão cortada) com os sinais filtrados
- É esperado o uso dos dois métodos (convolução e multiplicação), gerando então dois arquivos com áudio filtrado. Caso somente um método seja utilizado, haverá uma penalização na nota.

Os gráficos plotados devem ter os eixos X e Y com os valores esperados, conforme a teoria. Por exemplo, frequência deve ser ω (com amostras em múltiplos de $2\pi/N$) e tempo deve ser n (tempo discreto).

Prazo de entrega (pela SIGAA): 01/12/2023

Trabalho individual ou em dupla. Deve-se mostrar os detalhes do desenvolvimento das implementações. A entrega deve ser em formato ZIP contendo um documento estilo relatório + código, podendo ser um Notebook (e.g., Jupyter), e o arquivo WAV do sinal filtrado.