Transformada de Fourier no domínio discreto

Implementações e exemplos

Matheus Rodrigues de Souza

UFRJ

4 de julho de 2023

Sumário

Sumário

Transformada de Fourier no domínio discreto

- A transformada de Fourier discreta é uma ferramenta matemática usada para analisar sinais no domínio da frequência.
- Ela nos permite decompor um sinal discreto em suas componentes de frequência constituintes e obter as respectivas magnitude e fase.
- Ademais, essa ferramenta é muito usada em áreas como robótica, telecomunicações e processamento de imagem.
- De forma simplificada, é uma ferramenta que recebe uma informação e te devolve as frequências contidas naquela informação.

Transformada de Fourier no domínio discreto

- Seja x_n o sinal discreto a ser analizado na amostra n.
- n Amostra atual.
- k A frequência atual onde $k \in [0, N-1]$.
- N O número de amostras.
- X_k A transformada, que contêm as informações de amplitude e fase.

Transformada de Fourier no domínio discreto

- Podemos obter a transformada através da seguinte fórmula

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-j2\pi kn}{N}} \tag{1}$$

- A saída da transformada de fourier discreta, X_k, é um vetor de números complexos que armazenam as informações de frequência, amplitude e fase das senoides com compõe o sinal de entrada.

Transformada de Fourier no domínio discreto

- A primeira metade da do vetor, é composta pelos termos de frequência positiva, a segunda metade contêm as componentes de frequência negativa
- para sinais reais, a primeira metade é o conjugado da segunda metade. Sendo assim, normalmente focamos na primeira metade.

Transformada de Fourier no domínio discreto

- Podemos obter o espectro de amplitude da transformada através da seguinte equação:

$$A = \frac{|X_k|}{N} = \frac{\sqrt{Re^2(X_k) + Im^2(X_k)}}{N} \tag{2}$$

- Essa equação nos dá um espectro que aponta quais frequências contribuem mais para amplitude de X_k

Transformada de Fourier no domínio discreto

- Além disso, podemos obter o espectro de fase da transformada X_k através da seguinte equação:

$$\phi = \text{arctan}^2(Im(X_k), Re(X_k)) \tag{3}$$

- Essa equação aponta quais componentes de fase estão presentes no sinal analizado.

Transformada de Fourier no domínio discreto

- Além disso a transformada discreta de Fourier é regida pelas seguintes propriedades:
- Linearidade: Essa propriedade permite que analizemos sinais compostos através das frequências dos sinais separados.
- Podemos pensar na transformada discreta de Fourier como uma ferramenta que separa as frequências que constituem um sinal, a propriedade da linearidade é o que assegura que essa separação é preservada mesmo quando os sinais são combinados

$$ax_{n1} + bx_{n2} \xleftrightarrow{DFT} aX_{k1} + bX_{k2} \tag{4}$$

Transformada de Fourier no domínio discreto

- Deslocamento no tempo: Essa propriedade nos diz que se atrasarmos ou adiantarmos um sinal no tempo, as frequências correspondentes são deslocadas na mesma quantidade.

$$x_{n-n_0} \xleftrightarrow{DFT} e^{jkn_0} \cdot X_k \tag{5}$$

Transformada de Fourier no domínio discreto

- Deslocamento na frequência: De maneira análoga, essa propriedade descreve como um deslocamento de uma certa quantidade na frequência afeta a transformada.

$$x_n \cdot e^{jk_0 n} \xleftrightarrow{DFT} X_k(k - k_0) \tag{6}$$

Transformada de Fourier no domínio discreto

- Reversão no tempo: Podemos obter a transformada discreta de Fourier revertida no tempo de um sinal negando-o.

$$x_{-n} \overset{DFT}{\longleftrightarrow} X_k(-k) \tag{7}$$

Transformada de Fourier no domínio discreto

- Derivada na frequência: Podemos obter a representação da derivada de um sinal no domínio da frequência da seguinte forma:

$$n \cdot x_n \overset{DFT}{\longleftrightarrow} j \cdot \frac{d}{dk} X_k \tag{8}$$

Transformada de Fourier no domínio discreto

- Convolução no tempo: A convolução de dois sinais no tempo pode ser obtida tirando o produto dos mesmos entre si
- Essa propriedade é muito usada para implementar, computacionalmente, uma convolução, já que, fazendo um produto, a complexidade de tempo e espaço é significativamente menor que a de uma operação de integração

$$x_{n1} * x_{n2} \overset{DFT}{\longleftrightarrow} X_{k1} \cdot X_{k2} \tag{9}$$

Transformada de Fourier no domínio discreto

- Convolução na frequência: De forma análoga, também podemos definir a convolução na frequência:

$$x_{n1} \cdot x_{n2} \overset{DFT}{\longleftrightarrow} X_{k1} * X_{k2} \tag{10}$$

Implementações na biblioteca scipy.signal

- Agora vamos ver como a transformada discreta de Fourier é implementada.
- Faremos alguns exemplos usando a biblioteca scipy.signal
- Mas antes, precisamos falar sobre como a transformada é implementada computacionalmente

Implementações na biblioteca scipy.signal

- A Transformada rápida de Fourier é o algoritmo usado para implementar computacionalmente a DFT
- A FFT tem complexidade computacional de $O(Nlog(n))$, é significativamente mais rápida que a forma tradicional que vimos da DFT que tem complexidade de $O(N^2)$
- A FFT consegue essa performance aplicando uma estratégia de dividir para conquistar.

Implementações na biblioteca scipy.signal

- A Transformada rápida de Fourier é o algoritmo usado para implementar computacionalmente a DFT
- A FFT tem complexidade computacional de $O(Nlog(n))$, é significativamente mais rápida que a forma tradicional que vimos da DFT que tem complexidade de $O(N^2)$
- A FFT consegue essa performance aplicando uma estratégia de dividir para conquistar.

Implementações na biblioteca scipy.signal

- Ela explora a inerente simetria e periodicidade da DFT.
- Além disso, divide o problema em problemas menores, calculando DFTs de sequências menores e combinando para gerar o resultado final.

Implementações na biblioteca scipy.signal

- Vamos começar calculando a FFT de um sinal de teste com 50Hz,
- Nossa saída deve ser um gráfico que mostra a frequência que compõe esse sinal, no nosso caso, 50Hz
- Segue o nosso sinal de exemplo:

Implementações na biblioteca scipy.signal

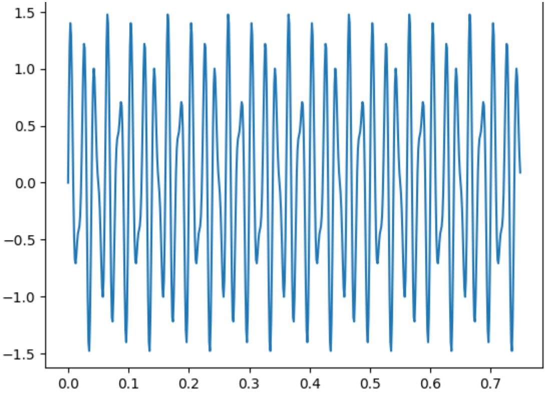

Figura: Sinal de teste de 50Hz

Implementações na biblioteca scipy.signal

- Nossa resposta de sáida:

Implementações na biblioteca scipy.signal

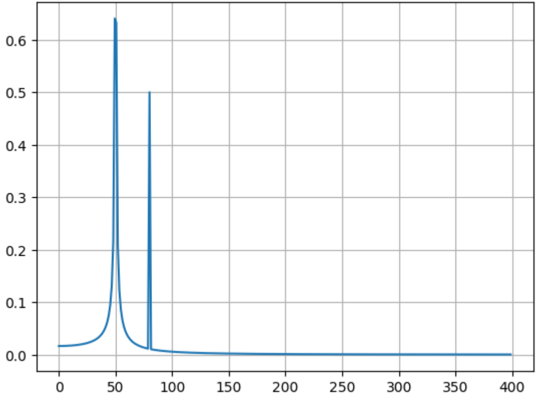

Figura: Resposta da transformada

Implementações na biblioteca scipy.signal

- Agora vamos obter o sinal de volta usando outra implementação da transformada de Fourier, a ifft
- Essa é uma implementação da transformada inversa, ela é capaz de, a partir do conjunto de frequências de um sinal nos devolver o sinal original reconstruído
- Nesse caso, nossa entrada agora, será:

Implementações na biblioteca scipy.signal

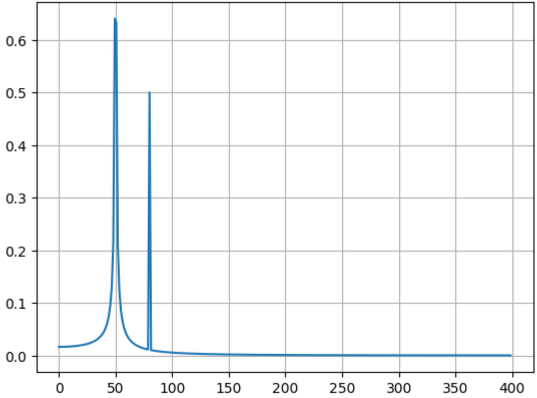

Figura: Entrada da IFFT

Implementações na biblioteca scipy.signal

- E a nossa resposta será o sinal original, como esperávamos:

Implementações na biblioteca scipy.signal

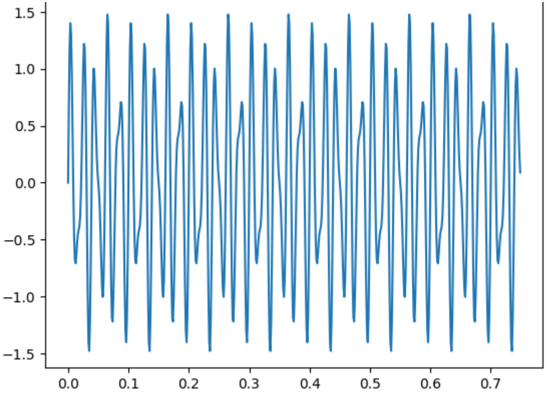

Figura: Sinal reconstruído

Sumário

Transformada de Fourier de tempo curto

- Essa implementação divide o sinal em pedaços e calcula a FFT de cada pedaço, sendo que cada pedaço tem o mesmo tamanho
- A FFT, nesse caso, é calculada separada e posta em função do tempo
- Assim, podemos avaliar como o sinal muda ao longo do tempo no domínio da frequência
- Normalmente, geramos um espectograma do sinal ao longo do tempo.

Transformada de Fourier de tempo curto

- Um desafio da implementação desse algoritmo é escolher o tamanho de janela ideal.
- Janelas muito grandes geram melhor resolução no domínio da frequência, mas pior resolução no domínio do tempo e vice-versa
- Essa técnica é muito utilizada para analizar sinais dinâmicos, não estacionários.

Bibliografia

- DFT
- Espectros de fase e amplitude
- Propriedades da DFT
- STFT