UNIVERSIDADE FEDERAL DO OESTE DO PARÁ – UFOPA
INSTITUTO DE ENGENHARIA E GEOCIÊNCIAS – IEG
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA

THAIS CRISTINA DOS ANJOS GOMES

ANÁLISE DE FREQUÊNCIAS EM SINAIS DE PRESENÇA: APLICAÇÃO DA TRANSFORMADA DE FOURIER E DFT

SANTARÉM
2023

THAIS CRISTINA DOS ANJOS GOMES

ANÁLISE DE FREQUÊNCIAS EM SINAIS DE PRESENÇA: APLICAÇÃO DA TRANSFORMADA DE FOURIER E DFT

Trabalho de Conclusão de Curso apresentado ao Bacharelado Interdisciplinar em Ciência e Tecnologia para obtenção do grau de Bacharel em Ciência e Tecnologia na Universidade Federal do Oeste do Pará, Instituto de Engenharia e Geociências.

Orientador: Manoel Roberval Pimentel Santos

SANTARÉM

2023

ANÁLISE DE FREQUÊNCIAS EM SINAIS DE PRESENÇA: APLICAÇÃO DATRANSFORMADA DE FOURIER E DFT

Thais Cristina dos Anjos Gomes [1], João Victor Figueira dos Santos[2], Manoel Roberval Pimentel [3]
[1]*Universidade Federal do Oeste do Pará, thaiscristinaanjos18@gmail.com*
[2]*Universidade Federal do Oeste do Pará, victor.s.tm20@gmail.com*
[3]*Universidade Federal do Oeste do Pará, proroberval@gmail.com*

Resumo

A transformada de Fourier, uma técnica matemática essencial, desagrega um sinal em suas componentes de frequência, sendo a Transformada Discreta de Fourier (DFT) uma aplicação valiosa para analisar a presença ou ausência de objetos em intervalos temporais específicos. A representação do sinal de presença como um array binário, onde o valor 1 indica presença e 0 indica ausência, oferece clareza na identificação da existência de objetos em cada ponto de amostragem. A DFT emerge como uma ferramenta útil para examinar as distintas frequências que constituem o sinal, revelando as oscilações temporais presentes. Essa análise de sinais permite a extração de informações significativas, tais como amplitude, frequência e padrões de variação. A transformada de Fourier e a DFT, aplicadas em diversos domínios, encontram particular relevância em sistemas de detecção e monitoramento que se baseiam em sensores de presença. Em ambientes nos quais é crucial discernir a presença ou ausência de objetos, essas técnicas matemáticas desempenham um papel fundamental. A capacidade de analisar e interpretar as características intrínsecas dos sinais de presença é essencial para sistemas de vigilância, automação e áreas correlatas à segurança.

Palavras-chave: Transformada, Decomposição, DFT, Sinais.

INTRODUÇÃO

A análise de sinais, com destaque para a decomposição de suas componentes de frequência, é uma aplicação vital de uma ferramenta matemática fundamental. Nesse contexto, os sensores de presença, concebidos para perceber mudanças no ambiente, operam monitorando as variações na intensidade ou frequência de um sinal específico, exemplificado pelo sensor de presença infravermelho que detecta alterações na emissão de calor em um local determinado.

Ao empregar a mencionada técnica em um sensor de presença, o procedimento inicial consiste na aquisição dos dados temporais do sinal. Posteriormente, aplica-se a referida técnica a esses dados, facilitando a identificação das componentes de frequência presentes no sinal capturado. A transformação resultante traduz-se em uma conversão do sinal do domínio do tempo para o domínio da frequência, revelando informações essenciais sobre as frequências presentes e suas respectivas amplitudes no sinal sob análise.

Essa abordagem é inestimável na detecção de padrões ou características específicas presentes no sinal do sensor de presença. A análise subsequente das componentes de frequência obtidas possibilita a identificação de características proeminentes, como frequências predominantes e picos de amplitude. Essa análise refinada desempenha um papel crucial na detecção de presença e na identificação de particularidades do ambiente monitorado.

Em resumo, a aplicação desta técnica, que converte sinais temporais em representações espectrais, é um componente essencial na análise de sinais em sensores de presença. Essa abordagem oferece uma perspectiva mais refinada das propriedades do sinal, viabilizando uma detecção mais precisa de mudanças no ambiente e proporcionando uma compreensão mais aprofundada das características dos sinais capturados pelos sensores de presença.

METODOLOGIA

A metodologia para a aplicação da Transformada de Fourier, enquanto uma etapa do processo de análise de sinais de presença é essencialmente uma abordagem sistemática que pode ser complementada por outras técnicas, algoritmos ou processamentos adicionais, visando a obtenção de resultados mais precisos e confiáveis na detecção de presença ou para outros objetivos específicos.

Para ilustrar um exemplo passo a passo da aplicação da transformada de Fourier a um sinal de um sensor de presença discreto no domínio do tempo, consideremos um cenário simplificado com um sinal de presença em 8 instantes de tempo, representado por $x=[0,0,1,1,0,0,0,1]$.

Passo 1:Determinar o número de pontos no sinal discreto (N). Neste exemplo, $N=8$, uma vez que temos 8 amostras de presença.

Passo 2: Aplicar a fórmula da Transformada Discreta de Fourier (DFT) para cada componente de frequência. Utilizaremos a seguinte fórmula para calcular as componentes de frequência do sinal para cada *k* no intervalo de 0 a $N-1$:

$$X[k]=\sum_{n=0}^{N-1} x[n]\cdot\exp\left(-\frac{2\pi i n k}{N}\right)$$

Para o exemplo, vamos calcular as componentes de frequência $X[k]$ para $k=0,1,2,3,4,5,6,7$.

Para *k* = 0

$$X[0]=0+0+1+1+0+0+0+1=3$$

Para *k* = 1

$$X[1]=0+0-0.707-0.707+0+0+0-0.707\approx-0.707$$

Calculamos os valores para *k* = 2, 3, 4, 5, 6, 7 da mesma forma.

Passo 3: Obter as componentes de frequência. As componentes de frequência do sinal transformado são derivadas dos valores calculados, resultando em $X=[3,-0.707,1,-0.707,0,-0.707,1,-0.707]$.

É crucial destacar que este exemplo é simplificado e destinado a fins ilustrativos. Em situações reais, é possível deparar-se com sinais de presença mais complexos, sendo necessário empregar técnicas adicionais para analisar e interpretar as componentes de frequência de maneira mais sofisticada.

Um exemplo prático de código em Python que utiliza a transformada de Fourier paraanalisar um sinal proveniente de um sensor de presença:

```
```

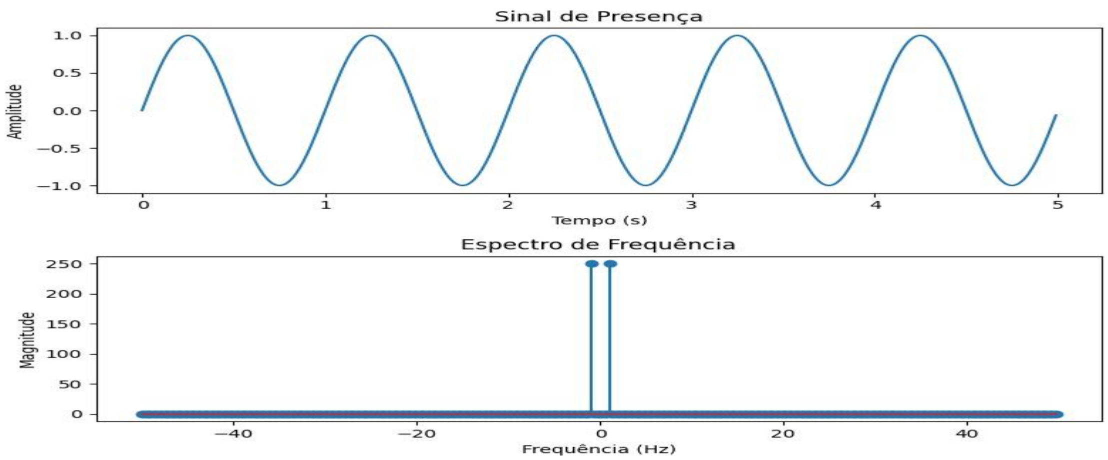

Gráfico 01: Espectro de frequência e Sinal de Presença. ApendiceA

Neste cenário exemplar, o sinal de presença é modelado como uma onda senoidal caracterizada por uma frequência e amplitude específicas. O processo inicia-se com a amostragem do sinal a uma taxa de amostragem determinada, resultando em uma versão discreta do sinal.

Posteriormente, a transformada de Fourier do sinal é calculada empregando a função `np.fft.fft()`. As frequências correspondentes às diversas componentes de frequência são então adquiridas mediante o uso da função `np.fft.fftfreq()`.

Finalizando o procedimento, o código realiza a representação gráfica do sinal de presença no domínio do tempo e do espectro de frequência. Essa visualização é concretizada por meio das funções `plot()` e `stem()` disponíveis no matplotlib, proporcionando uma compreensão intuitiva das características temporais e espectrais do sinal de presença em consideração.

Lembrando-se de importar as bibliotecas numpy e matplotlib para executar esse código.

```
Sinal de presença original: [0 1 1 0 1 0 0 1]

Transformada de Fourier: [ 4.00000000e+00+0.00000000e+00j  4.14213562e-01-1.00000000e+00j
 -4.44089210e-16+1.11022302e-16j -2.41421356e+00+1.00000000e+00j
  0.00000000e+00-2.44929360e-16j -2.41421356e+00-1.00000000e+00j
 -6.66133815e-16+2.22044605e-16j  4.14213562e-01+1.00000000e+00j]
```

Gráfico 02: Sinal de presença e Transformada de Fourier . Apendice A

Neste exemplo, a função `fourier_transform_sinal_presenca` é concebida para processar um sinal de presença representado por um array binário denominado `signal`. O procedimento consiste no cálculo da Transformada Discreta de Fourier (DFT) do sinal, utilizando a equação fundamental da transformada de Fourier discreta.

Dentro do laço `for`, cada componente de frequência `k` é determinada ao somar o produto do sinal de presença pelo fator exponencial correspondente. O resultado de cada iteração é acumulado no array `dft`, que, por sua vez, é retornado como o resultado da função.

No exemplo específico, o sinal de presença binário `[0, 1, 1, 0, 1, 0, 0, 1]` é fornecido como entrada para a função, e o resultado da transformada de Fourier é exibido por meio da instrução `print`. A biblioteca numpy é utilizada para executar esse código de transformada de Fourier.

Importante notar que a Transformada de Fourier, por ser uma técnica destinada à análise do espectro de frequências de um sinal, não se presta diretamente à criação de gráficos 3D. Contudo, é possível gerar um gráfico 3D que representa o espectrograma de um sinal de presença ao longo do tempo, empregando a Transformada de Fourier de Curto Período (STFT) em combinação com um gráfico tridimensional.

Segue abaixo um exemplo de código em Python, utilizando a biblioteca Matplotlib, para a criação de um gráfico 3D do espectrograma de um sinal de presença.

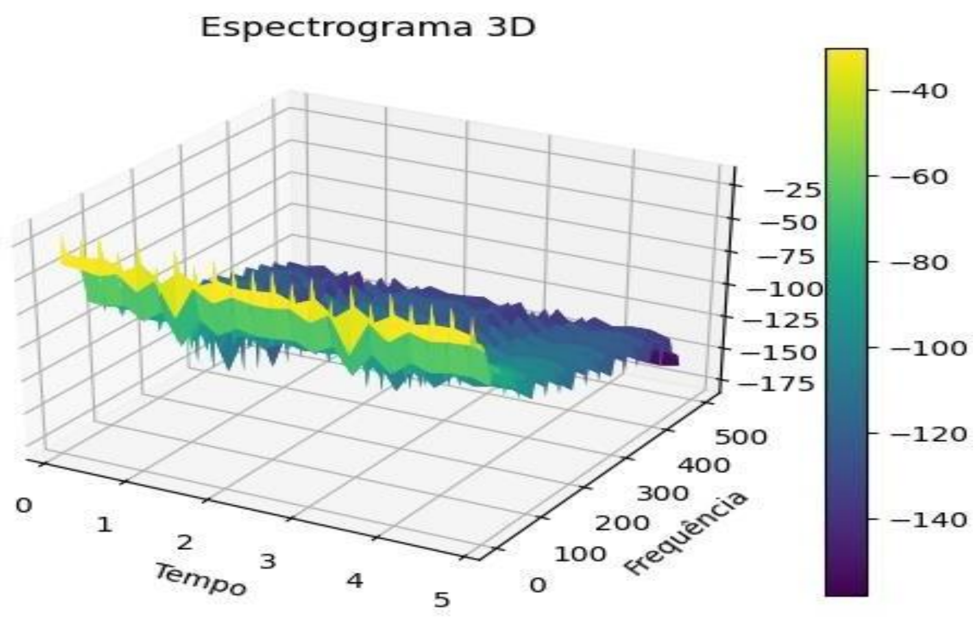

Gráfico 03: Espectrograma do sinal. Apêndice B

Neste exemplo, geramos um sinal de presença usando uma onda senoidal, configuramosa análise do espectrograma usando a STFT e, em seguida, plotamos o gráfico 3D doespectrograma utilizando `plot_surface()` da biblioteca Matplotlib. O eixo x representa otempo, o eixo y representa a frequência e o eixo z representa a magnitude em dB.

Considerações Finais

No âmbito dos sensores de presença, a profundidade oferecida pela análise de Fourier é inquestionável. Essa análise não apenas nos permite discernir entre múltiplos padrões e eventos com precisão, mas também nos proporciona uma visão holística do sinal através dos domínios temporal e espectral. Ao incorporar uma visualização tridimensional, comoo espectrograma, conseguimos apreender nuances da evolução da frequência com o tempo, nuances estas que, por vezes, podem se perder em representações mais tradicionais.

Este artigo enfatiza, portanto, o papel fundamental da análise de Fourier no contexto contemporâneo de sensores. Tanto para profissionais estabelecidos quanto para aqueles que estão ingressando na área, a familiaridade e aplicação destas técnicas são vitais para aprimorar a eficiência e precisão dos modernos sistemas de monitoramento e detecção. Ao visualizarmos o espectrograma em uma perspectiva 3D, ganhamos uma compreensão mais instintiva sobre como as frequências do sinal mudam ao longo do tempo. Esse tipo de representação enriquece nossa compreensão, trazendo insights que poderiam ser menos evidentes em análises puramente bidimensionais.

Resumindo, o estudo reitera o valor da análise de Fourier no contexto prático, em particular na área de sensores. Para aqueles que atuam na área ou se interessam pelo tema,dominar tais técnicas se mostra crucial para incrementar a acurácia e eficiência de sistemas dedicados à detecção e monitoramento.

Referências

Oppenheim, A. V., & Schafer, R. W. (2010). Discrete-Time Signal Processing. Prentice Hall.

Proakis, J. G., & Manolakis, D. G. (2006). Digital Signal Processing: Principles, Algorithms, and Applications. Pearson Education.

Orfanidis, S. J. (2010). Introduction to Signal Processing. Prentice Hall.

Bracewell, R. N. (2000). The Fourier Transform and Its Applications. McGraw-Hill Education.

Apêndice A

```
import numpy as np

import matplotlib.pyplot  as
plt from scipy.signal import
spectrogram
```

```python
# Parâmetros do sinal
frequencia = 1 # Frequência do sinal (em Hz)
amplitude = 1  # Amplitude do sinal
tempo_total = 5 # Tempo total de amostragem
(emsegundos)
taxa_amostragem = 100  # Taxa de amostragem (em Hz)

# Geração do sinal de presença
tempo = np.linspace(0, tempo_total, int(tempo_total
* taxa_amostragem), endpoint=False)
sinal_presenca = amplitude * np.sin(2 *
np.pi *frequencia * tempo)

# Cálculo da Transformada de Fourier
transformada =
np.fft.fft(sinal_presenca) frequencias =
np.fft.fftfreq(len(tempo), 1 /

taxa_amostragem)
# Plot do sinal de presença e do
espectro defrequência
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(8,
6))

ax1.plot(tempo,
sinal_presenca)
ax1.set_xlabel('Tempo (s)')
ax1.set_ylabel('Amplitude')
ax1.set_title('Sinal de
Presença')

ax2.stem(frequencias,
np.abs(transformada))
ax2.set_xlabel('Frequência (Hz)')
ax2.set_ylabel('Magnitude')
ax2.set_title('Espectro de
Frequência')
```

```
plt.tight_l
ayout()
plt.show()

def
    fourier_transform_sinal_presenca(s
    ignal):N = len(signal) #
    Comprimento do sinal frequencies =
    np.arange(N)  # Frequências
correspondentes às componentes de
    frequênciadft = np.zeros(N,
    dtype=np.complex)  #
Inicializar o array da transformada de
    Fourierfor k in range(N):
        dft[k] = np.sum(signal * np.exp(-2j * np.pi
* k *
    frequencies
    / N))return
    dft

signal = np.array([0, 1, 1, 0, 1, 0, 0, 1]) #
Sinalde presença binário
transform =
fourier_transform_sinal_presenca(signal)#
Print dos resultados
print("Sinal de presença original:",
signal)print("Transformada de
Fourier:", transform)
```

Apêndice B

```
# Gera um sinal de exemplo
Fs = 5000
T = 1.0 / Fs
x = np.linspace(0.0, 1.0, Fs)
```

```
# Um sinal com frequências variáveis
y = np.sin(50.0 * 2.0*np.pi*x) + 0.5*np.sin(80.0 *
2.0*np.pi*x)
frequencia, tempo, Sxx = spectrogram(y, Fs)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

T, F = np.meshgrid(tempo, frequencia)
surf = ax.plot_surface(T, F, 10 * np.log10(Sxx),
rstride=1, cstride=1, cmap='viridis',
edgecolor='none')
ax.set_xlabel('Tempo')
ax.set_ylabel('Frequencia')
ax.set_zlabel('Intensidade (dB)')
ax.set_title('3D espectograma')
fig.colorbar(surf)
plt.show()
```