



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

**JOÃO VICTOR FIGUEIRA DOS SANTOS**

**ANÁLISE DE CURTO TEMPO DE FOURIER PARA UMA GRANDE  
QUANTIDADE DE ENERGIA EM FORMA DE CALOR E ONDAS DE  
PRESSÃO INTERNA**

SANTARÉM

2023

**JOÃO VICTOR FIGUEIRA DOS SANTOS**

**ANÁLISE DE CURTO TEMPO DE FOURIER PARA UMA GRANDE  
QUANTIDADE DE ENERGIA EM FORMA DE CALOR E ONDAS DE  
PRESSÃO INTERNA**

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: Paula Renatha Nunes Da Silva

SANTARÉM

2023

# ANÁLISE DE CURTO TEMPO DE FOURIER PARA UMA GRANDE QUANTIDADE DE ENERGIA EM FORMA DE CALOR E ONDAS DE PRESSÃO INTENSA

**João Victor Figueira dos Santos**<sup>1</sup> – victor.s.tm20@gmail.com

**Paula Renatha Nunes da Silva**<sup>1</sup> – paula.silva@ufopa.edu.br

<sup>1</sup> Universidade Federal do Oeste do Pará, Instituto de Engenharia e Geociências – Santarém, PA, Brasil.

**Resumo:** Este estudo propõe a aplicação da Transformada de Fourier de Tempo Curto (STFT) para analisar as ondas de calor e ondas de choque geradas por armas termobáricas. Essas armas emitem ondas de calor e choque intensas, exigindo ferramentas adequadas para lidar com sinais não estacionários, cujo conteúdo de frequência varia ao longo do tempo. Destaca-se como uma ferramenta crucial, permitindo a análise da distribuição de energia em diferentes frequências e intervalos de tempo. Ao dividir o sinal em segmentos temporais curtos e aplicar a transformada de Fourier a cada segmento fornece uma representação da energia em várias frequências ao longo do tempo, identificando padrões espectrais e suas variações temporais. No contexto de ondas de pressão intensa, e facilita a análise da composição frequencial ao longo do tempo, identificando frequências dominantes e rastreando sua evolução à medida que a onda se propaga ou sofre alterações. A técnica incorpora o conceito de janelamento, multiplicando cada segmento do sinal por uma função de janela antes da transformada de Fourier, o que reduz artefatos causados por transições abruptas e assegura uma análise mais precisa e confiável. A escolha cuidadosa do tamanho e tipo de janela é crucial, influenciando a resolução temporal e espectral da análise. Portanto, surge como uma ferramenta inestimável para analisar sinais não estacionários, desempenhando um papel crucial na compreensão de fenômenos como ondas de calor e ondas de choque associadas a armas termobáricas.

**Palavras-chave:** Análise espectral; Armas termobáricas; Ondas de calor; Ondas de choque; Transformada de Fourier de Tempo Curto.

## 1. INTRODUÇÃO

A bomba termobárica também conhecida como bomba ar-combustível ou de vácuo é uma arma explosiva que gera uma grande explosão por meio da combinação de combustível e oxidante. Ao contrário das bombas convencionais, que usam explosivos químicos, as bombas termobáricas utilizam uma mistura altamente combustível de gases ou pós. A operação da bomba envolve a dispersão da mistura na área-alvo, seguida da detonação, resultando em uma oxidação rápida dos componentes e na liberação de energia em forma de calor e onda de pressão intensa.

Após dispersar a mistura, quando há uma fonte de ignição, inicia-se a detonação. A fonte de ignição pode ser uma espoleta ou uma carga explosiva adicional que inflama a mistura de combustível-oxidante, causando uma rápida oxidação dos componentes da mistura, liberando uma grande quantidade de energia na forma de calor e uma onda de pressão intensa. Essa combinação resulta em uma explosão extremamente poderosa com efeitos devastadores na área circundante.

Em ARMAS TERMOBÁRICAS O Estado da Arte das Cargas Explosiva relata que:

A bomba termobárica também é conhecida como bomba ar-combustível ou de vácuo. Essa primeira denominação advém do fato de que ela utiliza o

oxigênio do ar para gerar uma onda de explosão de alta temperatura e duração quando comparada a uma arma convencional. Já a segunda, relaciona-se com a formação de uma zona de baixa pressão, levando a um vácuo parcial com poder suficiente para causar danos a estruturas e pessoas.

Bombas termobáricas tem uma característica única que é sua capacidade de gerar uma grande quantidade de calor, e uma onda de pressão prolongada, decorrente da mistura de combustível e oxidante, que fornece seu próprio oxidante, permitindo que a reação de combustão continue mesmo sem oxigênio. Como resultado, a explosão termobárica é mais destrutiva e letal do que as bombas convencionais e, portanto, são armas altamente destrutivas, e seu uso é regulamentado em situações militares específicas devido ao seu potencial de causar danos maciços.

Para avaliar com maior precisão os danos causados por uma arma termobárica, é necessário extrair parâmetros característicos, visto que o teste de sobrepressão por onda de choque de explosão geralmente ocorre em ambientes complexos, e os dados do teste contêm muito ruído devido à interferência eletromagnética complexa, impacto de fragmentos e vibração do solo causado pela explosão. Desta maneira é de grande importância selecionar um método razoável de remoção de ruído e extração de características para extrair parâmetros efetivos de características de sobrepressão de ondas de choque dos sinais contendo ruído medido.

Dentre várias técnicas presentes na literatura, a Transformada de Fourier de Curto Tempo (STFT) é uma técnica que analisa sinais não estacionários, capturando informações de frequência que mudam ao longo do tempo. É útil para analisar sinais com componentes espectrais em constante mudança, com uma grande quantidade de energia em forma de calor. A STFT divide o sinal em pequenos segmentos de tempo e aplica a transformada de Fourier em cada segmento, revelando a distribuição de energia em diferentes frequências ao longo do tempo. Ao ser aplicado em uma onda de pressão intensa, é possível identificar as frequências dominantes e sua evolução durante a propagação. A técnica é calculada por meio de janelamento, reduzindo artefatos entre janelas. A escolha adequada do tamanho e tipoda janela é essencial para a resolução da análise temporal e espectral.

O objetivo fundamental deste estudo é conduzir uma avaliação precisa dos danos causados por bombas termobáricas, dispositivos explosivos que geram uma explosão substancial por meio da combinação de combustível e oxidante. Dada a complexidade inerente aos ambientes onde os testes de sobrepressão por onda de choque são realizados, somada à presença de ruído nos dados de teste, torna-se imperativo a seleção de um método eficiente para remoção de ruído e extração de características, visando analisar com precisão os parâmetros distintivos. Nesse contexto, o artigo propõe a adoção da Transformada de Fourier de Curto Tempo sendo uma estratégia meticulosamente desenvolvida para analisar sinais não estacionários, destacando-se ao capturar informações de frequência que se alteram ao longo do tempo, sendo especialmente valiosa para a análise de sinais dinâmicos.

## 2. TRANSFORMADA DE FOURIER DE TEMPO CURTO

Sinais não estacionários são aqueles cujas frequências variam com o tempo. Para estes tipos de sinais a teoria clássica de Fourier requer mais do que a transformada e demanda a introdução de uma dependência no tempo na análise de Fourier, se possível, preservando a linearidade.

A Transformada de Fourier de Curto Tempo (STFT) introduz um parâmetro de

frequência local (local no tempo), como se a Transformada de Fourier Local observasse o sinal através de uma curta janela dentro da qual o sinal permanece aproximadamente estacionário. A representação é bidimensional e, portanto, é definida matematicamente da seguinte forma, para um sinal contínuo  $x(t)$ :

Onde:

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-j\omega t} dt \quad (1)$$

- $\tau$  é o deslocamento no tempo, determinando a posição do segmento de análise.
- $\omega$  é a frequência angular.
- $w(t)$  é a função janela, que é aplicada ao sinal antes da transformada de Fourier ser calculada.

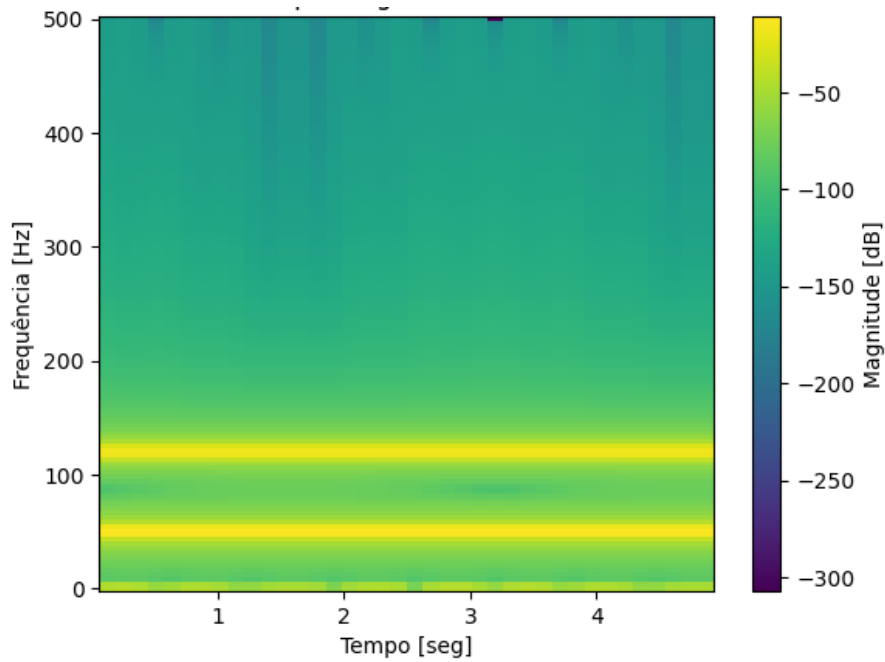
Para um sinal discreto  $x[n]$ , a STFT é calculada como:

$$X[m, k] = \sum_{n=-\infty}^{\infty} x[n] w[n - m] e^{-j\frac{2\pi}{N}kn} \quad (2)$$

- $m$  é o índice do segmento de análise.
- $k$  é o índice da frequência discreta.
- $N$  é o comprimento do sinal  $x[n]$ .
- $w[n]$  é a função janela discreta aplicada ao sinal.

A escolha adequada da função janela é importante para obter bons resultados na análise de sinais usando STFT. Alguns exemplos comuns de funções janela incluem a janela retangular, janela de Hamming, janela de Hann, janela de Blackman, entre outras. Cada tipo de janela tem características diferentes em relação à resolução temporal e de frequência.

Após a realização dos cálculos, torna-se viável adquirir o espectrograma, uma representação visual que reflete o módulo ou magnitude do resultado em relação ao tempo e à frequência. Este gráfico oferece insights acerca da distribuição de energia do sinal em diversas frequências ao longo do tempo, proporcionando a capacidade de analisar variações espectrais em sinais não estacionários. Um código foi desenvolvido para exemplificar um espectrograma relacionado ao tópico abordado, culminando na criação do Gráfico 1.

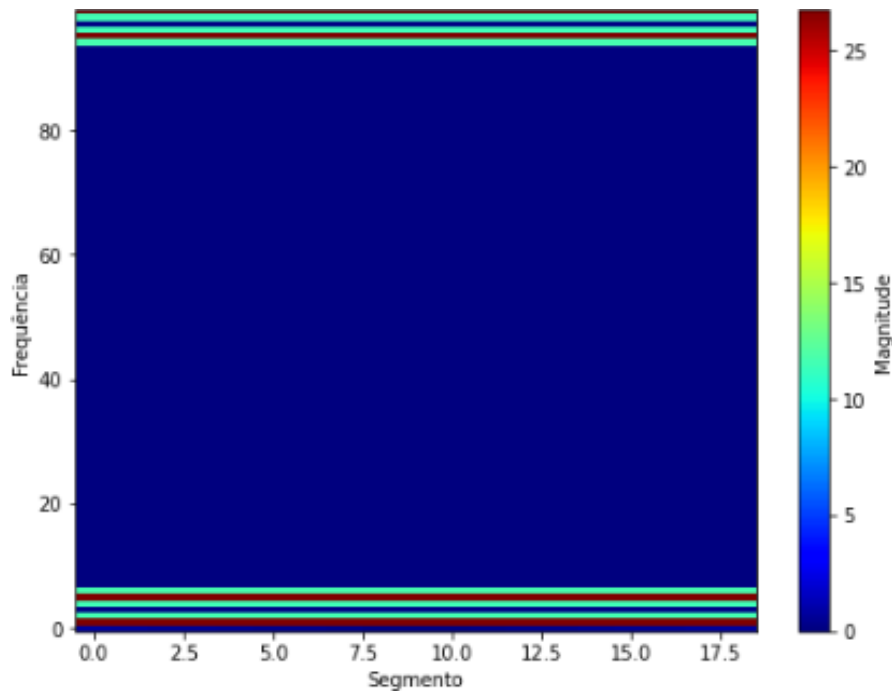


**Gráfico 1:** Espectrograma derivado da abordagem contextual.

O recurso combina o uso da Transformada de Fourier com o janelamento de segmentos de um sinal para analisar suas propriedades espectrais ao longo do tempo, sendo especialmente útil para analisar sinais não estacionários, como uma grande quantidade de energia em forma de calor ou uma onda de pressão intensa.

### 3. APLICAÇÃO PRÁTICA E REPRESENTAÇÃO GRÁFICA

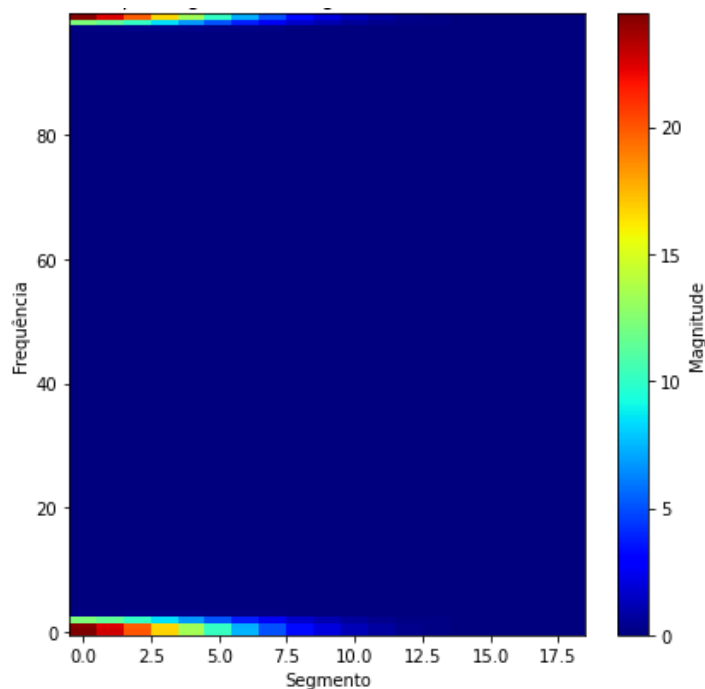
No exemplo, um sinal de exemplo com duas frequências sinusoidais ( $f_1 = 10$  Hz e  $f_2 = 50$  Hz) é criado. Em seguida, é aplicada usando a função 'stft' da biblioteca SciPy, com os parâmetros adequados. O resultado é armazenado nas variáveis 'f', 't' e 'Zxx', representando respectivamente as frequências, os tempos e o espectrograma resultante. Por fim, o espectrograma é plotado usando a função 'pcolormesh' do Matplotlib, podendo-se ajustar os parâmetros de acordo com suas necessidades específicas, como o tamanho da janela, a sobreposição entre as janelas e a função janela utilizada.



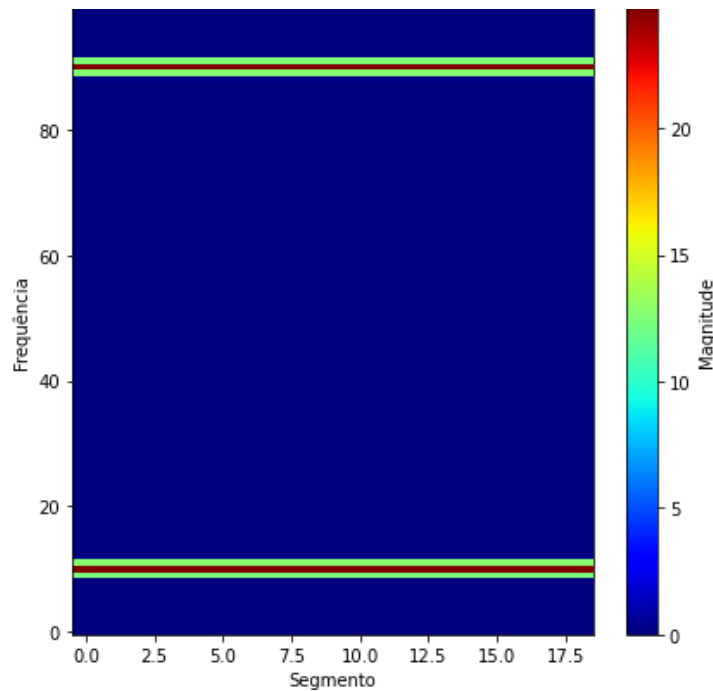
**Gráfico 2:** Espectrograma demonstra a aplicação em um sinal discreto.

Para um sinal contínuo, a integral precisa ser aproximada numericamente utilizando métodos como a quadratura numérica.

No apêndice C está um exemplo de código Python que realiza a Análise de Curto Tempo de Fourier (STFT) em um sinal representando uma grande quantidade de energia em forma de calor e uma onda de pressão intensa, no qual são considerados dois sinais distintos: um sinal representando uma grande quantidade de energia em forma de calor ( $x_1$ ) e um sinal representando uma onda de pressão intensa ( $x_2$ ). A função `stft` é utilizada para calcular a STFT em cada um dos sinais.



**Gráfico 3:** Espectrograma – Energia em Forma de Calor



**Gráfico 4:** Espectrograma – Onda de Pressão Intensa.

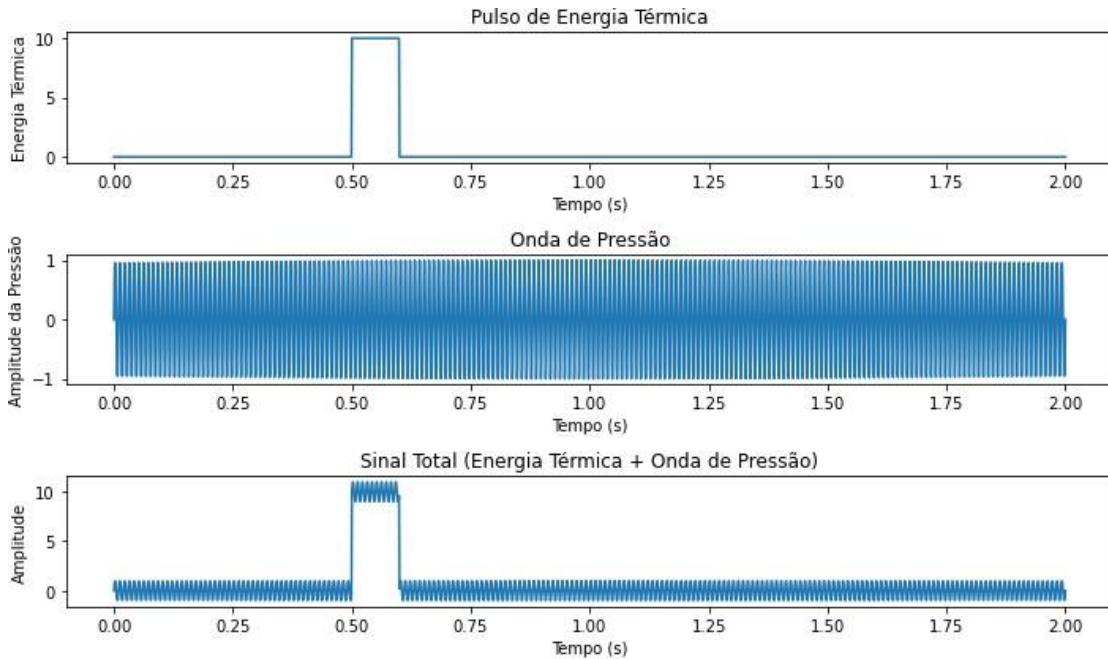
O código aplicado realiza a aplicação da Transformada de Fourier de Tempo Curto (STFT) e a criação de espectrogramas para dois sinais distintos: um relacionado à energia em forma de calor e outro associado a ondas de pressão intensa demonstrados nos gráficos 3 e 4. Os principais pontos são:

- A função ``stft`` processa um sinal, aplicando a STFT com uma janela e número de pontos por segmento especificados.
- São gerados dois sinais de exemplo, um para representar calor e outro para ondas de pressão intensa, com características específicas de frequência, duração e amostragem.
- Definição do tamanho da janela de análise (``nperseg``) e uso da janela de Hanning.
- A função ``stft`` é aplicada a cada sinal, resultando em matrizes ``X1`` e ``X2``.
- Os espectrogramas são visualizados em subplots, mostrando a magnitude da STFT ao longo do tempo e da frequência para os sinais de calor e onda de pressão intensa.
- A figura final contém dois subplots, cada um representando um espectrograma.
- As cores indicam a magnitude dos componentes de frequência.
- O subplot da esquerda representa "Energia em Forma de Calor", enquanto o da direita é associado a "Onda de Pressão Intensa".
- Utilização de bibliotecas como NumPy e Matplotlib.
- Emprego da função de janela de Hanning para melhorar a análise espectral.

Portanto, o código proporciona uma representação eficaz da distribuição de energia em diferentes frequências ao longo do tempo para os sinais fornecidos.



Para uma simulação com grande quantidade de energia em forma de calor e uma onda de pressão intensa foi considerado pulso de energia térmica seguido por uma onda de pressão usando uma abordagem numérica mostrando do gráfico 5 que pode ajudar a ilustrar essa simulação.

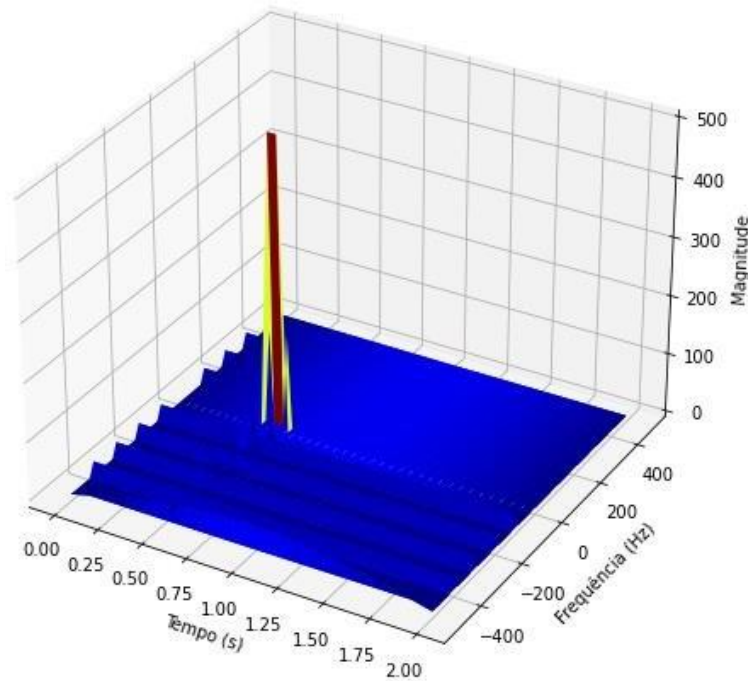


**Gráfico 5:** Pulso de Energia Térmica, Onda de Pressão e Sinal Total (Energia Térmica + Onda de Pressão).

No Gráfico 5 é apresentado um exemplo da simulação, que consiste em um pulso de energia térmica e uma onda de pressão. O pulso de energia térmica é representado por um pulso retangular no início da simulação, enquanto a onda de pressão é uma função senoidal com amplitude e frequência definidas.

Os sinais de energia térmica, onda de pressão e o sinal total (soma dos dois sinais) são plotados utilizando a biblioteca Matplotlib. É importante lembrar que esta é apenas uma simulação simples, e que o comportamento real de uma grande quantidade de energia em forma de calor e uma onda de pressão intensa podem ser muito mais complexos. É fundamental ter conhecimentos avançados, e lidar com cuidado ao realizar simulações ou experimentos relacionados a esses tipos de eventos.

No Gráfico 6 está ilustrado um exemplo como o código que realiza uma análise STFT, em uma grande quantidade de energia em forma de calor e ondas de pressão intensa:



**Gráfico 6:** Análise de Curto Tempo de Fourier – Grande Quantidade de Energia em Forma de Calor e Ondas de Pressão Intensa.

Neste exemplo foi simulado para uma grande quantidade de energia em forma de calor uma sequência de ondas de pressão intensa. Em seguida, aplicamos a análise de curto tempo de Fourier (STFT) no sinal total resultante. O Gráfico 6 3D mostra a magnitude da STFT em função do tempo e da frequência. Vale ressaltar que é importante certificar-se que as bibliotecas ``numpy``, ``matplotlib`` e ``mpl_toolkits.mplot3d`` instaladas em seu ambiente Python antes de executar o código.

## CONSIDERAÇÕES FINAIS

Este estudo concentra-se na avaliação precisa dos danos causados por bombas termobáricas, dispositivos explosivos que desencadeiam explosões significativas ao combinar combustível e oxidante. Diferentemente das bombas convencionais que empregam explosivos químicos, as bombas termobáricas utilizam misturas altamente combustíveis de gases ou pós. A dispersão dessa mistura na área-alvo, seguida da detonação, resulta em rápida oxidação dos componentes, liberando considerável energia na forma de calor e onda de pressão intensa. Essas armas, reconhecidas por sua destrutividade, são sujeitas a regulamentações específicas em contextos militares, dada sua capacidade de causar danos massivos.

A necessidade de avaliação precisa dos danos conduz à extração de parâmetros característicos. Os testes de sobrepessão por onda de choque, frequentemente realizados em ambientes complexos, geram dados ruidosos devido a interferências eletromagnéticas, impacto de fragmentos e vibração do solo. A escolha de um método eficaz para remoção de ruído e extração de características é, portanto, crucial.

Nesse cenário, a Transformada de Fourier de Curto Tempo (STFT) destaca-se como uma ferramenta essencial. Esta técnica analisa sinais não estacionários, capturando informações de frequência em constante mudança ao longo do tempo. Ao dividir o sinal em segmentos temporais e aplicar a transformada de Fourier, a STFT

revela a distribuição de energia em diferentes frequências, permitindo identificar frequências dominantes e sua evolução durante a propagação. A aplicação da STFT em ondas de pressão intensa destaca-se como particularmente valiosa.

A análise por meio de janelamento, uma estratégia de cálculo da STFT, reduz artefatos entre janelas. A escolha cuidadosa do tamanho e tipo da janela é crucial para garantir resolução temporal e espectral adequada. Em resumo, este estudo propõe uma abordagem metódica para avaliar os danos causados por bombas termobáricas, enfatizando a importância crítica da STFT na análise de sinais dinâmicos em ambientes complexos e ruidosos.

## REFERENCIAS

ARMAS TERMOBÁRICAS O Estado da Arte das Cargas Explosivas. Disponível em: <<https://portaldeperiodicos.marinha.mil.br/index.php/passadico/article/download/2313/2297/>>. Acesso em: 18 de novembro de 2023.

## Apendice A

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import spectrogram, get_window

def stft(signal, fs, window='hann', nperseg=256, noverlap=None):
    f, t, Sxx = spectrogram(signal, fs, window=get_window(window, nperseg),
                             nperseg=nperseg, noverlap=noverlap)
    return f, t, Sxx

# Função para calcular a STFT
def calculate_stft(signal, fs, window='hann', nperseg=256, noverlap=None):
    frequencies, times, spectrogram_result = stft(signal, fs, window=window,
                                                    nperseg=nperseg, noverlap=noverlap)
    return frequencies, times, spectrogram_result

# Sinal de exemplo
fs = 1000 # Frequência de amostragem
t = np.linspace(0, 5, 5 * fs, endpoint=False) # Tempo de 0 a 5 segundos
signal = np.sin(2 * np.pi * 50 * t) + 0.5 * np.sin(2 * np.pi * 120 * t)

# Parâmetros da STFT
nperseg = 256
noverlap = nperseg // 2
window = 'hann'

# Calcular a STFT
frequencies, times, spectrogram_result = calculate_stft(signal, fs, window=window,
                                                         nperseg=nperseg, noverlap=noverlap)

# Plotar o espectrograma
```

```

plt.pcolormesh(times, frequencias, 10 * np.log10(spectrogram_result), shading='auto')
plt.ylabel('Frequência [Hz]')
plt.xlabel('Tempo [seg]')
plt.title('Espectrograma - STFT')
plt.colorbar(label='Magnitude [dB]')
plt.show()

```

## Apêndice B

```

import numpy as np
import matplotlib.pyplot as plt

# Função para calcular a STFT def stft(x, window, nperseg):
N = len(x) M = nperseg
H = int(nperseg/2) # Sobreposição entre janelas

num_segments = int(np.ceil((N - M) / H)) + 1 padded_length = (num_segments - 1) * H + M

x_padded = np.pad(x, (0, padded_length - N), 'constant')

# Inicialização do array para armazenar os resultados da STFT X = np.zeros((nperseg,
num_segments), dtype=np.complex128)

# Aplicação da STFT em cada segmento for m in range(num_segments):
start = m * H end = start + M
segment = x_padded[start:end] windowed_segment = segment * window
# Cálculo da transformada de Fourier
X[:, m] = np.fft.fft(windowed_segment, n=nperseg) return X
# Parâmetros do sinal
fs = 1000 # Frequência de amostragem
t = np.linspace(0, 1, fs) # Vetor de tempo fl = 10 # Frequência do sinal
f2 = 50 # Frequência do sinal
x = np.sin(2 * np.pi * fl * t) + np.sin(2 * np.pi * f2 * t) # Sinal de exemplo

# Parâmetros da STFT
nperseg = 100 # Tamanho da janela de análise
window = np.hamming(nperseg) # Janela de Hamming

# Aplicação da STFT
X = stft(x, window, nperseg)

# Plotagem do espectrograma plt.figure(figsize=(8, 6))

plt.imshow(np.abs(X), aspect='auto', origin='lower', cmap='jet') plt.colorbar(label='Magnitude')
plt.xlabel('Segmento') plt.ylabel('Frequência') plt.title('Espectrograma (STFT)')
plt.show()

```

## Apêndice C

```

import numpy as np
import matplotlib.pyplot as plt

# Função para calcular a STFT def stft(x, window, nperseg):
N = len(x) M = nperseg
H = int(nperseg/2) # Sobreposição entre janelas

num_segments = int(np.ceil((N - M) / H)) + 1 padded_length = (num_segments - 1) * H + M

x_padded = np.pad(x, (0, padded_length - N), 'constant')

# Inicialização do array para armazenar os resultados da STFT X = np.zeros((nperseg,
num_segments), dtype=np.complex128)

# Aplicação da STFT em cada segmento for m in range(num_segments):
start = m * H end = start + M
segment = x_padded[start:end]
windowed_segment = segment * window # Cálculo da transformada de Fourier
X[:, m] = np.fft.fft(windowed_segment, n=nperseg) return X

# Parâmetros do sinal de calor
fs1 = 1000 # Frequência de amostragem duration1 = 1.0 # Duração do sinal em segundos
t1 = np.linspace(0, duration1, int(fs1*duration1)) # Vetor de tempo f1 = 10 # Frequência do sinal
x1 = np.exp(-10 * t1**2) * np.cos(2 * np.pi * f1 * t1) # Sinal de exemplo de calor

# Parâmetros do sinal de onda de pressão fs2 = 1000 # Frequência de amostragem
duration2 = 1.0 # Duração do sinal em segundos
t2 = np.linspace(0, duration2, int(fs2*duration2)) # Vetor de tempo f2 = 100 # Frequência do
sinal
x2 = np.sin(2 * np.pi * f2 * t2) # Sinal de exemplo de onda de pressão

# Parâmetros da STFT
nperseg = 100 # Tamanho da janela de análise window = np.hanning(nperseg) # Janela de Hanning

# Aplicação da STFT no sinal de calor X1 = stft(x1, window, nperseg)

# Aplicação da STFT no sinal de onda de pressão X2 = stft(x2, window, nperseg)

# Plotagem dos espectrogramas plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.imshow(np.abs(X1), aspect='auto', origin='lower', cmap='jet')
plt.colorbar(label='Magnitude')
plt.xlabel('Segmento') plt.ylabel('Frequência')
plt.title('Espectrograma - Energia em Forma de Calor')

plt.subplot(1, 2, 2)
plt.imshow(np.abs(X2), aspect='auto', origin='lower', cmap='jet')
plt.colorbar(label='Magnitude')
plt.xlabel('Segmento') plt.ylabel('Frequência')
plt.title('Espectrograma - Onda de Pressão Intensa')

```

```
plt.tight_layout()plt.show()
```

## Apêndice D

```
import numpy as np
import matplotlib.pyplot as plt

# Parâmetros da simulação
duration = 2.0 # Duração da simulação em segundos
fs = 1000 # Frequência de amostragem
N = int(fs * duration) # Número total de amostras

t = np.linspace(0, duration, N) # Vetor de tempo

# Parâmetros da energia térmica
heat_energy = 10.0 # Energia térmica do pulso em J
heat_duration = 0.1 # Duração do pulso térmico em segundos
heat_start_time = 0.5 # Tempo de início do pulso térmico em segundos
heat_end_time = heat_start_time + heat_duration # Tempo de término do pulso térmico em segundos

# Parâmetros da onda de pressão
pressure_amplitude = 1.0 # Amplitude da onda de pressão
pressure_frequency = 100 # Frequência da onda de pressão em Hz
pressure_start_time = 1.0 # Tempo de início da onda de pressão em segundos

# Simulação do sinal de calor
heat_signal = np.zeros(N)
heat_samples = int(heat_duration * fs)
heat_start_index = int(heat_start_time * fs)
heat_end_index = heat_start_index + heat_samples
heat_signal[heat_start_index:heat_end_index] = np.sqrt(heat_energy / heat_duration)

# Simulação da onda de pressão
pressure_signal = pressure_amplitude * np.sin(2 * np.pi * pressure_frequency * (t - pressure_start_time))

# Sinal total (soma do sinal de calor e da onda de pressão)
total_signal = heat_signal + pressure_signal

# Plotagem dos sinais
plt.figure(figsize=(10, 6))
plt.subplot(3, 1, 1)
plt.plot(t, heat_signal)
plt.xlabel('Tempo (s)')
plt.ylabel('Energia Térmica')
plt.title('Pulso de Energia Térmica')

plt.subplot(3, 1, 2)
plt.plot(t, pressure_signal)
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude da Pressão')
plt.title('Onda de Pressão')

plt.subplot(3, 1, 3)
plt.plot(t, total_signal)
plt.xlabel('Tempo (s)')
plt.ylabel('Amplitude')
plt.title('Sinal Total (Energia Térmica + Onda de Pressão)')

plt.tight_layout()plt.show()
```

## Apêndice E

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Parâmetros da simulação
duration = 2.0 # Duração da simulação em segundos fs = 1000 # Frequência de amostragem
N = int(fs * duration) # Número total de amostras t = np.linspace(0, duration, N) # Vetor de tempo

# Parâmetros da energia térmica
heat_energy = 10.0 # Energia térmica do pulso em J heat_duration = 0.1 # Duração do pulso térmico em segundos
heat_start_time = 0.5 # Tempo de início do pulso térmico em segundos
heat_end_time = heat_start_time + heat_duration # Tempo de término do pulso térmico em segundos

# Parâmetros das ondas de pressão num_waves = 5 # Número de ondas de pressão
pressure_amplitude = 1.0 # Amplitude das ondas de pressão
pressure_frequencies = [100, 200, 300, 400, 500] # Frequências das ondas de pressão em Hz
pressure_start_times = [0.7, 0.8, 0.9, 1.0, 1.1] # Tempos de início das ondas de pressão em segundos
pressure_durations = [0.05, 0.06, 0.07, 0.08, 0.09] # Durações das ondas de pressão em segundos

# Simulação do sinal de calor heat_signal = np.zeros(N) heat_samples = int(heat_duration * fs)
heat_start_index = int(heat_start_time * fs) heat_end_index = heat_start_index + heat_samples
heat_signal[heat_start_index:heat_end_index] = np.sqrt(heat_energy / heat_duration)

# Simulação das ondas de pressão pressure_signals = []
for i in range(num_waves):
    pressure_signal = pressure_amplitude * np.sin(2 * np.pi * pressure_frequencies[i] * t)
    pressure_signals.append(pressure_signal)

# Sinal total (soma do sinal de calor e das ondas de pressão) total_signal = heat_signal +
np.sum(pressure_signals, axis=0)

# Função para calcular a STFT
def stft(x, window_size, overlap):
    num_segments = int(np.ceil((N - window_size) / overlap)) + 1 padded_length = (num_segments - 1) *
    overlap + window_size x_padded = np.pad(x, (0, padded_length - N), 'constant')
    X = np.zeros((window_size, num_segments), dtype=np.complex128) for m in
    range(num_segments):
        start = m * overlap
        end = start + window_size segment = x_padded[start:end]
        X[:, m] = np.fft.fft(segment, n=window_size) return X

# Parâmetros da STFT
window_size = int(fs * 0.05) # Tamanho da janela de análise (50 ms) overlap = int(window_size /
2) # Sobreposição entre janelas (50%) frequencies = np.fft.fftfreq(window_size, d=1/fs) # Vetor de
```

frequências

```
# Aplicação da STFT no sinal total
stft_result = stft(total_signal, window_size, overlap)

# Criação da malha para o gráfico 3D
T, F = np.meshgrid(np.linspace(0, duration, stft_result.shape[1]), frequencies)

# Plotagem do gráfico 3D
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d') ax.plot_surface(T, F, np.abs(stft_result), cmap='jet')
ax.set_xlabel('Tempo (s)') ax.set_ylabel('Frequência (Hz)') ax.set_zlabel('Magnitude')
ax.set_title('ANÁLISE DE CURTO TEMPO DE FOURIER\nGrande Quantidade de
Energia em Forma de Calor e Ondas de Pressão Intensa')
plt.show()
```

---

## SHORT-TIME FOURIER ANALYSIS FOR A LARGE AMOUNT OF ENERGY IN THE FORM OF HEAT AND INTENSE PRESSURE WAVES

**Abstract:** *This study proposes the application of the Short-Time Fourier Transform (STFT) to analyze heat waves and shock waves generated by thermobaric weapons. These weapons emit intense heat and shock waves, requiring appropriate tools to deal with non-stationary signals whose frequency content varies over time. It stands out as a crucial tool, allowing for the analysis of energy distribution at different frequencies and time intervals. Dividing the signal into short temporal segments and applying the Fourier transform to each segment provides a representation of the energy at various frequencies over time, identifying spectral patterns and their temporal variations. In the context of intense pressure waves, and facilitates the analysis of the frequency composition over time, identifying dominant frequencies and tracking their evolution as the wave propagates or undergoes changes. The technique incorporates the concept of windowing, multiplying each segment of the signal by a window function before the Fourier transform, which reduces artifacts caused by abrupt transitions and ensures a more accurate and reliable analysis. The careful choice of window size and type is crucial, influencing the temporal and spectral resolution of the analysis. Therefore, it emerges as an invaluable tool for analyzing non-stationary signals, playing a crucial role in understanding phenomena such as heat waves and shock waves associated with thermobaric weapons.*

**Keywords:** *Spectral analysis; Thermobaric weapons; Heat waves; Shock waves; Short-time Fourier transform.*