

Universidade de São Paulo



Relatório 1

Processamento de Imagens por Ressonância
Magnética Utilizando Transformada Wavelet
Iniciação Científica

Aluno: Thiago Oliveira dos Santos

Orientador: Fernando Fernandes Paiva

São Carlos

2025

1 Resumo

Este relatório apresenta o desenvolvimento inicial de um projeto de Iniciação Científica cujo objetivo principal é a aplicação da Transformada Wavelet no processamento de imagens de Ressonância Magnética (RM). Inicialmente, o projeto concentrou-se no estudo dos fundamentos do processamento de sinais, abrangendo conceitos de sinais analógicos e digitais, sistemas, e a digitalização de sinais contínuos segundo o teorema de Nyquist. A partir desses fundamentos, foi proposto e implementado um problema inicial envolvendo a Transformada de Fourier (FT) aplicada a um sinal de RM unidimensional simulado. O código desenvolvido, utilizando as bibliotecas `numpy` e `matplotlib`, está organizado em dois arquivos: `useful_lib.py` – que implementa funções para a simulação do sinal, aplicação da FT e plotagem dos resultados –, e `main.py` – que fornece uma interface interativa para a manipulação dos parâmetros e visualização dos resultados. Adicionalmente, o relatório apresenta uma discussão acerca da filtragem digital de sinais, destacando a operação de convolução e os filtros passa-altas, passa-baixas, passa-banda e rejeita-banda. Para superar a limitação da FT na preservação da informação temporal, é apresentada a Transformada de Fourier de Curto Tempo (STFT) e, em seguida, a Transformada Wavelet, que, por meio da Transformada Wavelet Discreta no Tempo (DTWT) e do Algoritmo de Mallat, permite uma análise tempo-frequência mais adequada para sinais com características transitórias. As primeiras implementações com relação à abertura e preparação das imagens de RM também já foram feitas e se encontram preparadas para as próximas etapas do projeto. Desta forma, as etapas propostas para o primeiro semestre de projeto estão completas e estabelecem uma base teórica e computacional sólida para a continuação do trabalho. Como próximos passos, o projeto avançará para a implementação completa da Transformada Wavelet no processamento de imagens de RM e a validação dos resultados com dados experimentais. Todo o código desenvolvido está disponível em um repositório online, facilitando a reprodução dos resultados e a colaboração em futuras etapas do projeto.

2 Objetivos

O objetivo principal deste projeto é aplicar a Transformada Wavelet no processamento de imagens de Ressonância Magnética. Para isso, ao longo dos últimos seis meses, foram se-

guidos alguns objetivos específicos: estudar os conceitos fundamentais do processamento de sinais e das transformadas matemáticas relevantes (Transformada de Fourier e Transformada Wavelet); implementar e analisar a Transformada de Fourier em um sinal unidimensional; comparar as representações nos domínios do tempo e da frequência, visando compreender como os parâmetros escolhidos para o sinal influenciam sua visualização no domínio da frequência; e preparar a base teórica e computacional para a futura implementação da Transformada Wavelet.

3 Introdução

O processamento de sinais e imagens consiste na manipulação de dados, com o objetivo de extrair informações relevantes, melhorar sua qualidade ou transformá-los para facilitar análises posteriores. Para isso, faz-se necessário o uso de ferramentas matemáticas que permitam interpretar e modificar esses sinais em diferentes domínios. Neste relatório, serão abordados os conceitos teóricos que servem como base para o entendimento dos métodos abordados neste trabalho, sendo esses as transformadas matemáticas relevantes e os princípios que regem a manipulação de sinais. Além disso, ao final deste relatório, está disponibilizado um link que direciona ao GitHub (um repositório de códigos e arquivos online) deste projeto, onde o leitor pode encontrar os códigos que estão sendo usados para o desenvolvimento da pesquisa.

4 Desenvolvimento do Projeto

Nesta seção serão abordados todos os passos tomados até o presente momento, e o estado atual do projeto. Como previsto no cronograma inicial, essa primeira etapa do projeto se concentrou no aprendizado, por parte do aluno, dos conceitos necessários para o prosseguimento do projeto, para depois passarmos para a implementação dos códigos relevantes para o estudo.

4.1 Considerações Sobre Sinais Digitais

Antes que se passe para a discussão de tópicos mais avançados, torna-se necessário esclarecer alguns conceitos fundamentais a respeito do processamento de sinais. Esses conceitos formam

a base para compreender as transformadas matemáticas e técnicas utilizadas ao longo deste trabalho.

Entende-se por sinal qualquer grandeza física que varia no tempo, no espaço ou em ambas, e que pode ser utilizada para transmitir informações. Exemplos de sinais são som, potencial elétrico, luz e até gestos com intenção comunicativa. O processo de transmissão de um sinal geralmente envolve a produção de informação por uma fonte, que então é codificada no sinal e transmitida através de um meio, sendo esse possivelmente processado; e, por fim, o sinal é recebido por um sistema que o decodifica em seu conteúdo inicial. Como exemplo, uma lâmpada (fonte) irradia luz (informação), que é codificada em termos das componentes elétrica e magnética de sua radiação (sinal), e viaja através do ar (meio) até o olho de um observador, que é responsável por interpretar essa luz e formar uma imagem na córnea (sistema).

Um sinal pode ser classificado como analógico ou digital. Um sinal analógico é todo aquele que está codificado em grandezas físicas contínuas, tais como pressão, temperatura, potencial, entre outros. Já um sinal digital é discreto e está codificado em bits. Por exemplo, um conjunto de números:

$$S_d(n) = \{1, 2, 3, 4, 5\}$$

pode ser considerado um sinal digital, enquanto a curva contínua:

$$S_a(x) = x, \quad 1 \leq x \leq 5$$

pode ser usada para representar um sinal analógico, como, por exemplo, a intensidade da luz (S_a) em função do tempo (x). Na Figura 1 tem-se um exemplo de sinal analógico.

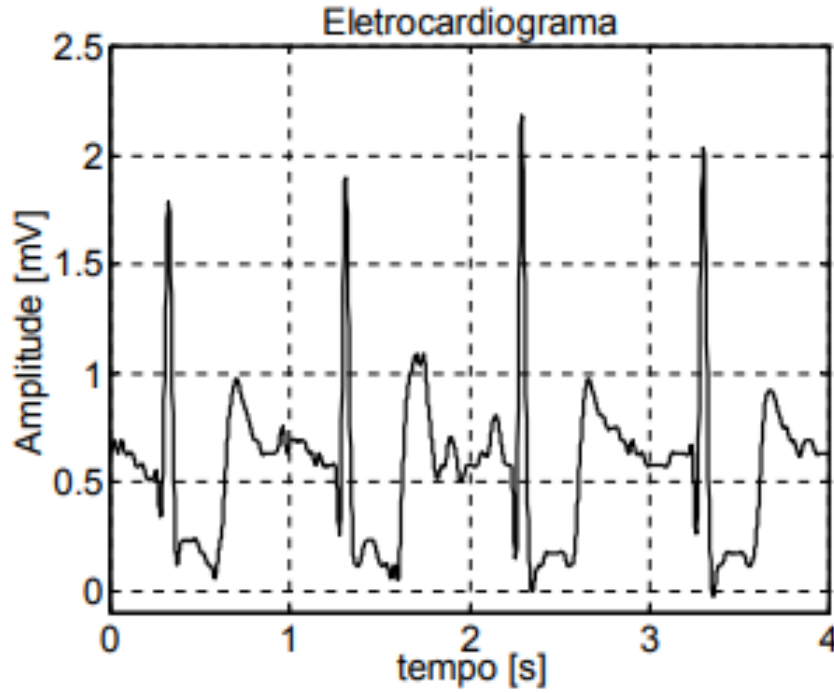


Figura 1: Exemplo de sinal analógico

Fonte: Higuti & Kitano, 2003

Os sinais desempenham um papel fundamental na comunicação e no processamento de informações. No entanto, seu verdadeiro impacto se dá quando são manipulados por sistemas, que extraem, transformam ou transmitem informações contidas nesses sinais. Define-se sistema como qualquer mecanismo que recebe um sinal $S_a(t)$ e retorna outro sinal $S_b(t)$. Os sinais de entrada e saída de um sistema, porém, não necessariamente devem ser trabalhados no mesmo domínio, podendo estar definidos em diferentes domínios, como tempo, frequência ou espaço, dependendo do tipo de processamento realizado. Por exemplo, um sistema $S\{.\}$ pode receber um sinal $x(t)$ com domínio no tempo, e retornar um sinal $y(\omega)$ com domínio na frequência:

$$X(\omega) = \mathcal{F}(x(t)) \quad (1)$$

Um exemplo de sistema que opera entre diferentes domínios é a Transformada de Fourier de um sinal unidimensional, que recebe de um sinal $x(t)$ que depende do tempo e retorna um sinal $X(\omega)$ no domínio da frequência. Essa operação pode ser realizada digitalmente, sendo então, pelo restante deste relatório, a Transformada de Fourier considerada um sistema digital. Existem vários exemplos de sistemas físicos (analógicos) e digitais. Como

visto, a aplicação de uma Transformada de Fourier em um sinal pode ser vista como um processo realizado por um sistema que converte sinais do domínio do tempo para o domínio da frequência, sendo considerado como um sistema digital, como também ocorre com os filtros passa-altas e passa-baixas. Como exemplo de sistemas analógicos, pode-se citar um rádio, que converte um sinal de ondas eletromagnéticas em um sinal sonoro. A Figura 2 ilustra um exemplo esquemático do funcionamento de um sistema, em que um sinal de entrada $x(t)$ é processado, resultando em um sinal de saída $y(t)$.

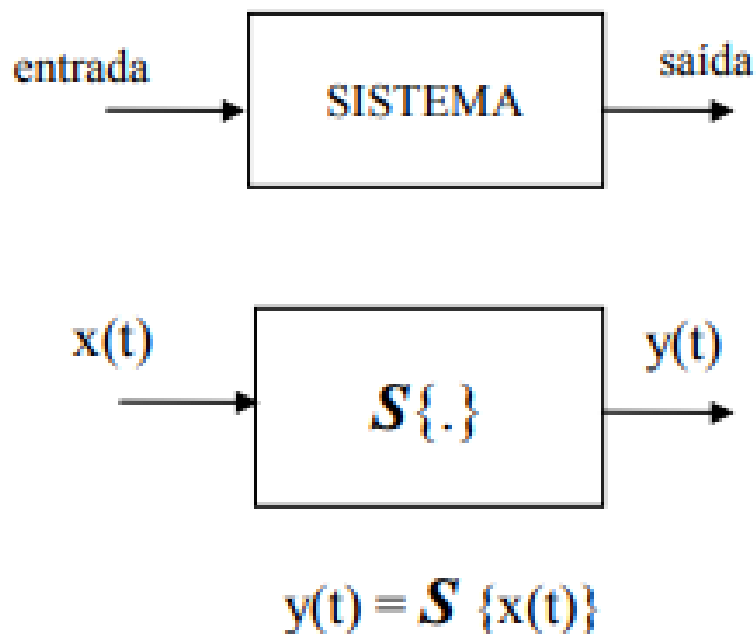


Figura 2: Esquema do funcionamento de um sistema

Fonte: Higuti & Kitano, 2003

Sinais contínuos, definidos em todo o domínio do tempo, são comuns no cotidiano e oferecem grande potencial para a transmissão de informações. Contudo, ao se processar um sinal por meio de um computador, torna-se inviável analisar um sinal contínuo em seu domínio completo, já que isso implicaria em um número infinito de pontos e, consequentemente, um tempo de processamento infinito. Para superar esse desafio, o sinal contínuo é convertido em um sinal digital, ou seja, é discretizado em um número finito de pontos igualmente espaçados.

Essa conversão possibilita a representação precisa do sinal original, desde que a taxa de amostragem seja adequada. Esse princípio é fundamentado no teorema de Nyquist (ou te-

orema da amostragem), que estabelece que, para representar um sinal digital sem aliasing – a sobreposição de frequências – a taxa de amostragem deve ser, no mínimo, o dobro da frequência máxima presente no sinal. Por exemplo, para um sinal com frequência máxima de 1000 Hz, é necessária amostragem em intervalos de 0,5 ms, ou seja, uma frequência de 2000 Hz.

A Figura 3 traz um esquema que exemplifica a aplicação do teorema de Nyquist a um sinal $x_a(t)$. Note que, quando um sinal é discretizado, ele deixa de depender de uma variável contínua (nesse caso, o tempo), e passa a depender de uma variável n discreta, sendo então um $x_b(n)$, com $n \geq 0$.

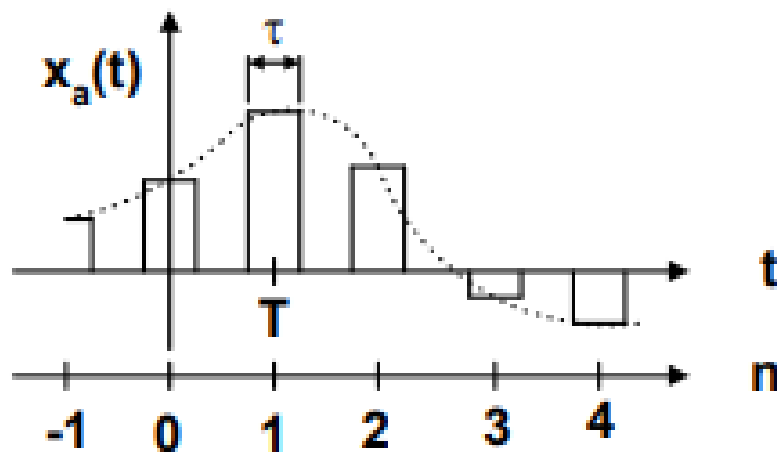


Figura 3: Aplicação do teorema de Nyquist

Fonte: Higuti & Kitano, 2003

4.2 Filtragem Digital

É fato que um sinal carrega informações codificadas, e que este pode ser representado digitalmente, com tanta precisão quanto se queira. Porém, o sinal pode estar imerso em um ruído que atrapalha a obtenção da informação desejada. Ou então, tem-se um sinal que é emitido por várias fontes, mas quer-se atentar à informação advinda de uma única fonte. Por exemplo, ao analisar a gravação de uma música, quer-se analisar somente o som proveniente do cantor.

O processamento de sinais é um processo cujo objetivo principal é analisar as informações contidas nestes; seja em sua totalidade, ou em apenas parte dessas informações. Um sinal digital pode então ser processado para facilitar a visualização da parte de interesse, através

de sistemas conhecidos como filtros digitais.

Um filtro digital é, em sua essência, um processo computacional pelo qual uma sequência discreta de números, o sinal, é transformada em outra sequência discreta de números, denominada sinal de saída. Estamos interessados em filtros digitais que podem ser representados como um conjunto de coeficientes que interagem com o sinal digital via convolução. A discussão a respeito da montagem destes coeficientes que compõem o filtro digital está fora do escopo deste projeto.

Matematicamente, a convolução de um sinal contínuo é definida por:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau = x(t) * h(t) \quad (2)$$

enquanto a convolução de um sinal discreto é definida como:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k] = x[n] * h[n] \quad (3)$$

No entanto, existem certos truques para a realização da convolução em sinais digitais unidimensionais e bidimensionais. Para a convolução unidimensional, imagine que tenhamos um sinal $s[\cdot] = \{1, 2, 3\}$ e um filtro $h[\cdot] = \{4, 5\}$. Alinhe as componentes do sinal e, logo abaixo, as componentes do filtro, de modo a realizar a multiplicação elemento a elemento. Em vez de realizar o “acarregamento” dos dígitos (como na soma convencional), escreve-se apenas o resultado da multiplicação em cada posição, sem levar em conta o transporte entre colunas. Assim, o sinal resultante seria:

$$y[\cdot] = \{4, 13, 22, 15\}.$$

Um esquema ilustrativo desse procedimento está apresentado na Figura 4.

	1	2	3	M-sample long signal
		4	5	N-sample long filter
	5	10	15	
4	8	12	+	
4	13	22	15	resulting signal with M+N-1 samples

Figura 4: Esquema para a aplicação de convolução unidimensional

Fonte: Guido, 2022

Para aplicar uma convolução em um sinal bidimensional (exemplificado na Figura 5), faz-se:

1. **Deslize o filtro sobre a imagem:** Posicione o filtro sobre o sinal de modo que cada elemento do filtro corresponda a um elemento do sinal.
2. **Computar o valor da convolução:** Para cada posição do filtro no sinal, calcule a soma ponderada dos elementos do filtro multiplicados pelos elementos correspondentes do sinal.
3. **Armazenar o resultado:** Substitua o valor original do valor no sinal pela soma calculada.

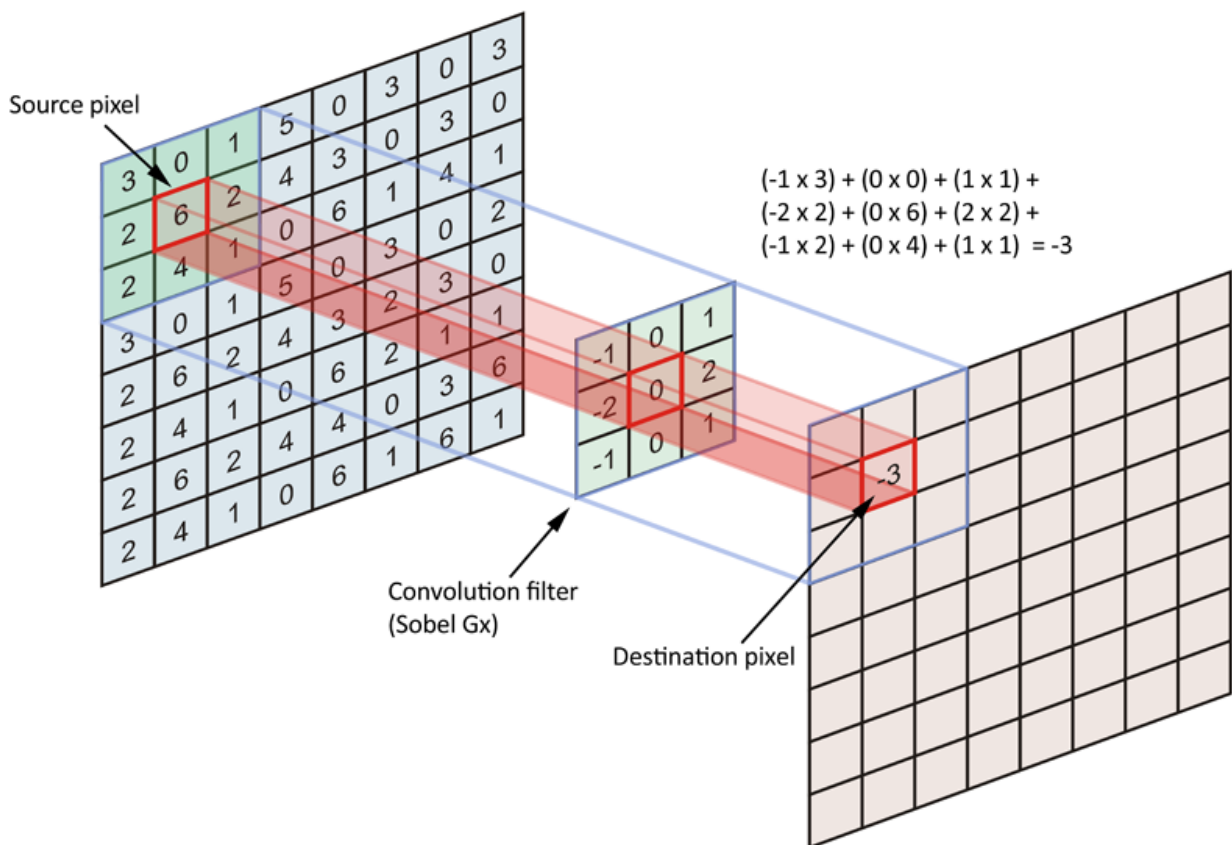


Figura 5: Esquema para a aplicação de convolução bidimensional

Fonte: Slides da disciplina de Processamento e Análise de Imagens Digitais oferecida no IFSC em 2024

Este processo é repetido para cada elemento do sinal, resultando em um novo sinal filtrado. O uso de filtros digitais é essencial quando se tem interesse em suprimir determinadas componentes espectrais ou, alternativamente, ressaltar outras, de modo a extrair a informação

desejada. Dessa forma, após a discretização do sinal, torna-se possível manipular e separar as faixas de frequência de interesse. Assim, um filtro com essas características pode ser classificado como:

1. **Passa-altas:** Permite a passagem das frequências acima de um certo corte e atenua as frequências inferiores.
2. **Passa-baixas:** Permite a passagem das frequências abaixo de um certo corte e atenua as frequências superiores.
3. **Passa-banda:** Permite a passagem de frequências dentro de um intervalo determinado e atenua as demais.
4. **Rejeita-banda:** Atenua as frequências dentro de um determinado intervalo, permitindo a passagem das demais.

Nas Figuras 6 e 7 tem-se exemplos, respectivamente, de um sinal não filtrado e de outro filtrado por um filtro passa-banda.

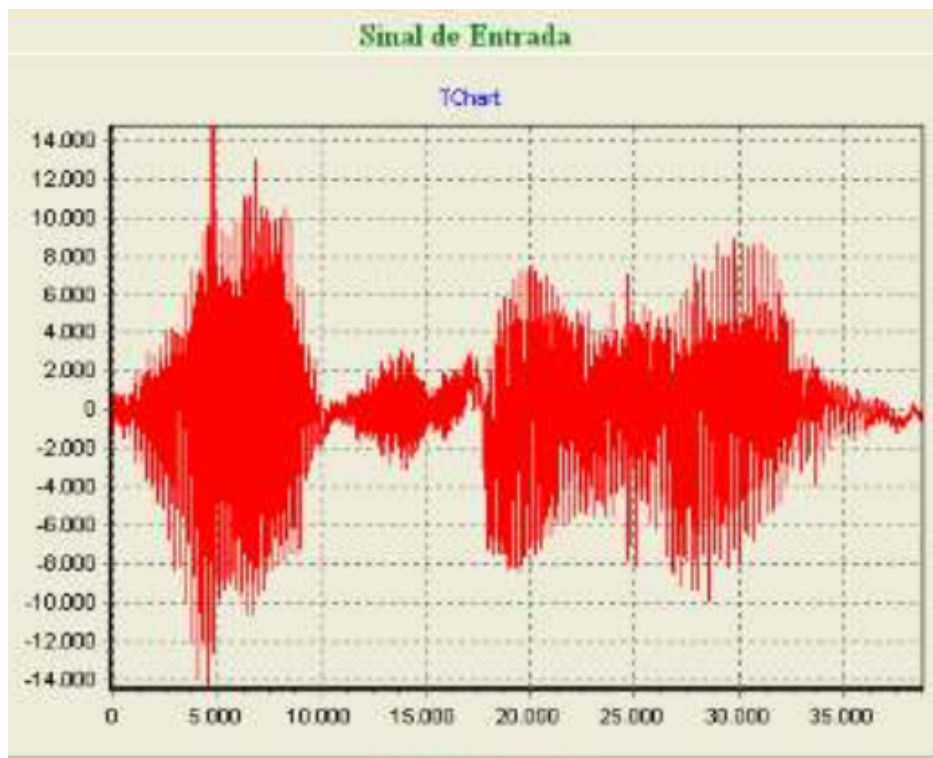


Figura 6: Sinal sonoro não filtrado

Fonte: Marinus, 2011

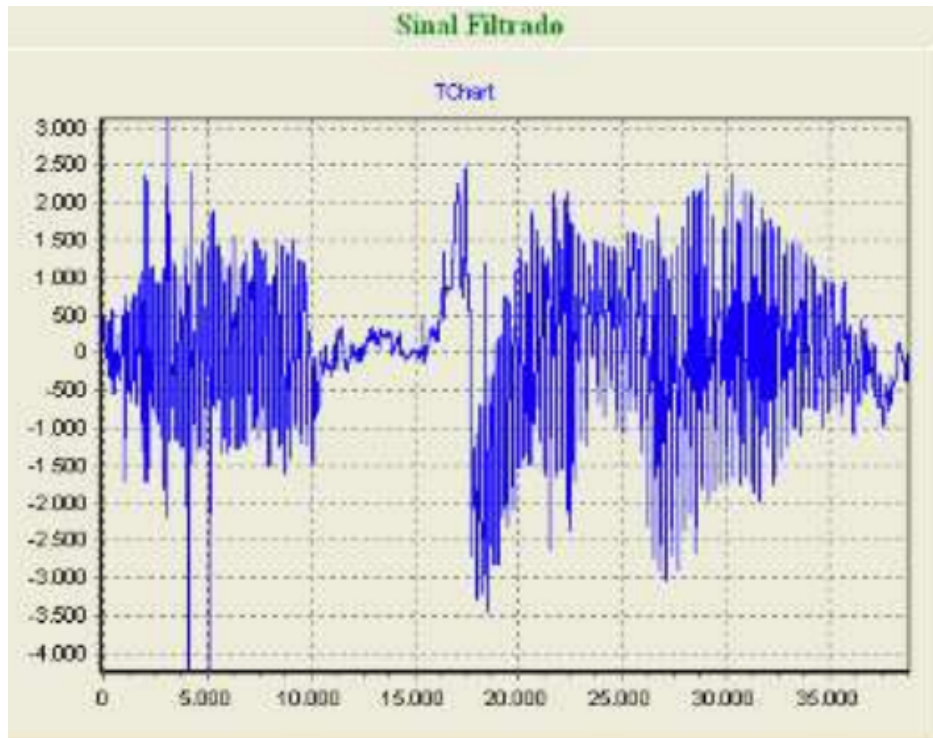


Figura 7: Sinal sonoro filtrado com filtro passa-banda

Fonte: Marinus, 2011

4.3 Transformadas

Frequentemente, na análise de um sinal, torna-se conveniente estudá-lo em um domínio diferente daquele em que ele foi originalmente definido. Para isso, utiliza-se um processo matemático conhecido como transformada integral. Essa transformação é uma operação linear, definida genericamente na forma:

$$T\{f(t)\} = \int_{t_1}^{t_2} f(t)K(t,u)dt \quad (4)$$

onde $K(t,u)$ é o kernel da transformada e $f(t)$ é a função a ser transformada. Ou seja, uma transformada é uma ferramenta matemática que, ao receber uma função de entrada, retorna outra função definida em um domínio distinto. Ressalta-se que não é necessário que a quantidade de variáveis independentes das funções de entrada e saída seja igual, sendo a própria Transformada Wavelet um exemplo de transformada que “acrescenta” variáveis para o domínio da função de saída.

4.3.1 Transformada de Fourier (FT)

Um tipo específico de transformada amplamente utilizada no processamento de sinais é a FT, cujo Kernel é definido como:

$$K(t, u) = K(t, \omega) = e^{-i\omega t} \implies F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (5)$$

Note que a FT retorna um sinal com componentes complexas. Para facilitar a visualização e a manipulação dos dados, frequentemente, em aplicações clínicas relacionadas à RM, consideramos apenas o módulo do sinal transformado, que corresponde ao espectro de amplitudes, deixando a informação de fase em segundo plano.

A FT é usada em processamento de sinais para analisar as componentes espectrais de um sinal. Isso é possível graças às definições da própria transformada e da série de Fourier, que nos dizem que qualquer função pode ser representada a partir de uma série infinita de senos e cossenos. Em outras palavras, a FT identifica quais frequências de senos e cossenos estão presentes no sinal. Por exemplo, considere um sinal da forma:

$$f(t) = e^{628it} \quad (6)$$

Sabendo da relação:

$$\omega = 2\pi f \quad (7)$$

onde ω indica a frequência angular em rad/s e f indica a frequência em Hz, tem-se então que a FT de $f(t)$ consiste em um único pico perto da posição 100 Hz. Na Figura 8 tem-se um exemplo da FT aplicada a um sinal do tipo box.

Embora a Transformada de Fourier (FT) seja uma ferramenta poderosa para a análise e manipulação de sinais contínuos, o processamento digital exige que o sinal seja representado de forma discreta. Para isso, utiliza-se a Transformada Discreta de Fourier (DFT), que é definida por:

$$X[k] = \mathcal{F}\{x[n]\} = \sum_{n=0}^{N-1} x[n]e^{-\frac{i2\pi kn}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (8)$$

A DFT é amplamente utilizada para a manipulação do espectro de sinais digitais; no entanto, sua representação usual — baseada na amplitude (ou módulo) do sinal transformado

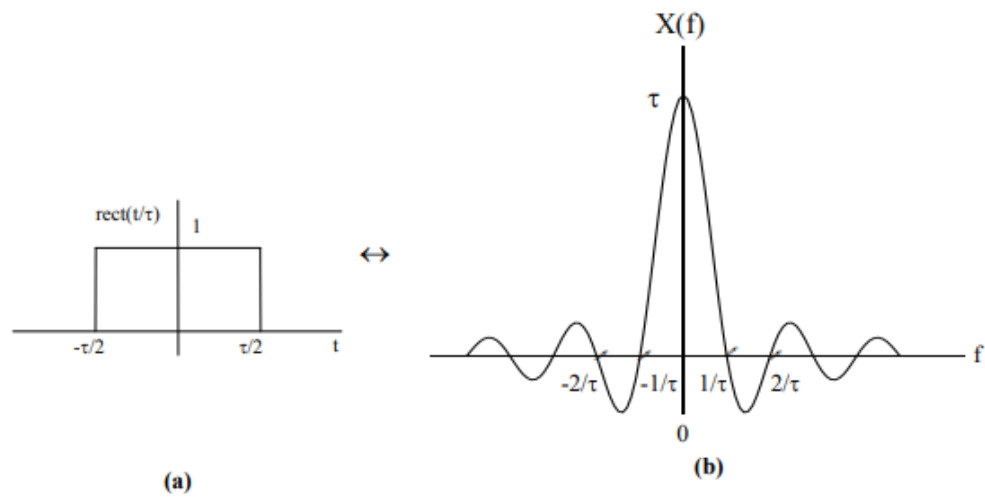


Figura 8: a) Exemplo de sinal b) Transformada de Fourier do sinal

Fonte: Higuity & Kitano, 2003

— resulta na perda da informação temporal. Em outras palavras, a DFT não permite identificar “quando” determinadas componentes espectrais estão presentes no sinal. A Figura 9 ilustra essa limitação, mostrando (na parte superior) um sinal típico de ECG e (na parte inferior) sua FT.

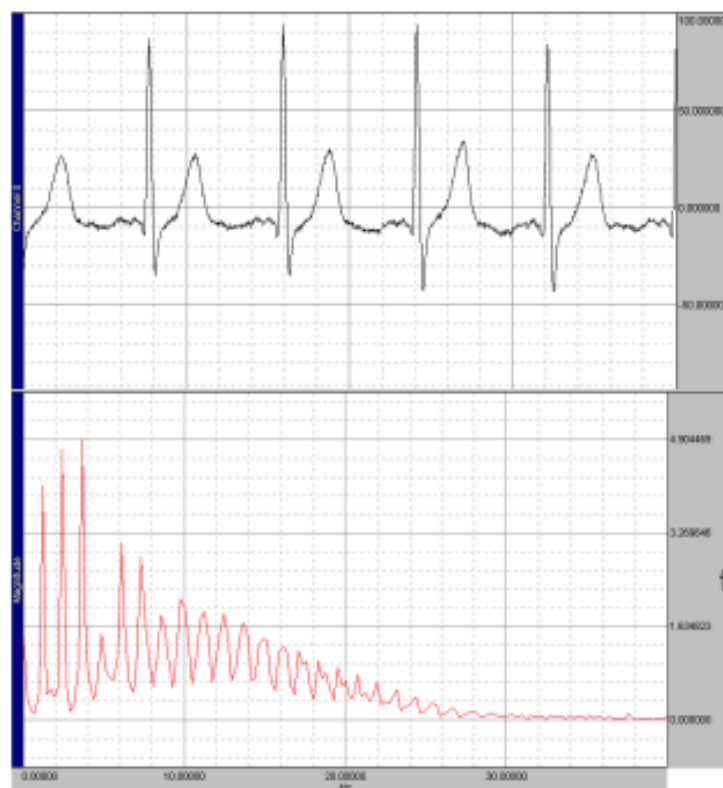


Figura 9: Exemplo de um sinal de ECG, e sua FT

Fonte: Fechine, 2010

Uma solução para essa perda de informação temporal é aplicar a transformada de Fourier de forma localizada, utilizando uma “janela” para calcular a transformada aos poucos. Esse é exatamente o papel da Transformada de Fourier de Curto Tempo (STFT). Contudo, a STFT apresenta uma limitação inerente: o comprimento da janela é fixo. Uma janela muito pequena pode não capturar uma oscilação completa de uma frequência específica, enquanto uma janela muito grande pode englobar múltiplas frequências, afetando a resolução temporal. Esse trade-off entre resolução temporal e espectral é ilustrado na Figura 10.

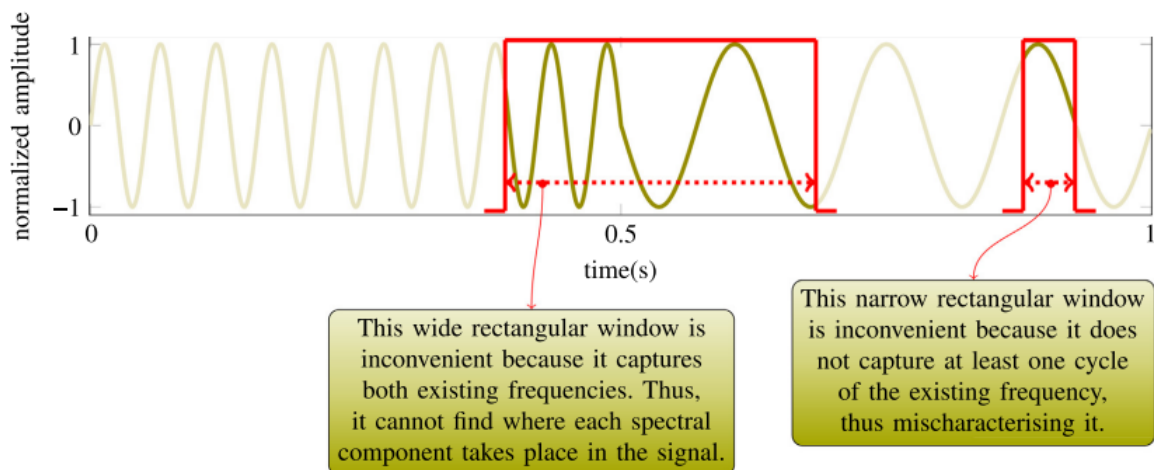


Figura 10: Esquema que ilustra o problema da STFT

Fonte: Guido, 2022

4.3.2 Transformada Wavelet (WT)

Mesmo que a FT seja amplamente utilizada no processamento de sinais, esta não é a melhor opção nos casos em que se quer analisar o domínio espectral juntamente com o espaço original do sinal. A Figura 11 traz uma representação desta transformada. Note que o gráfico da CWT requer a inclusão de uma outra dimensionalidade para a análise dos resultados: a cor em cada ponto do gráfico. A WT, contudo, soluciona o problema da STFT, ao introduzir uma nova base de funções, que não são apenas senos e cossenos, como na FT, mas sim uma com a janela já embutida em si.

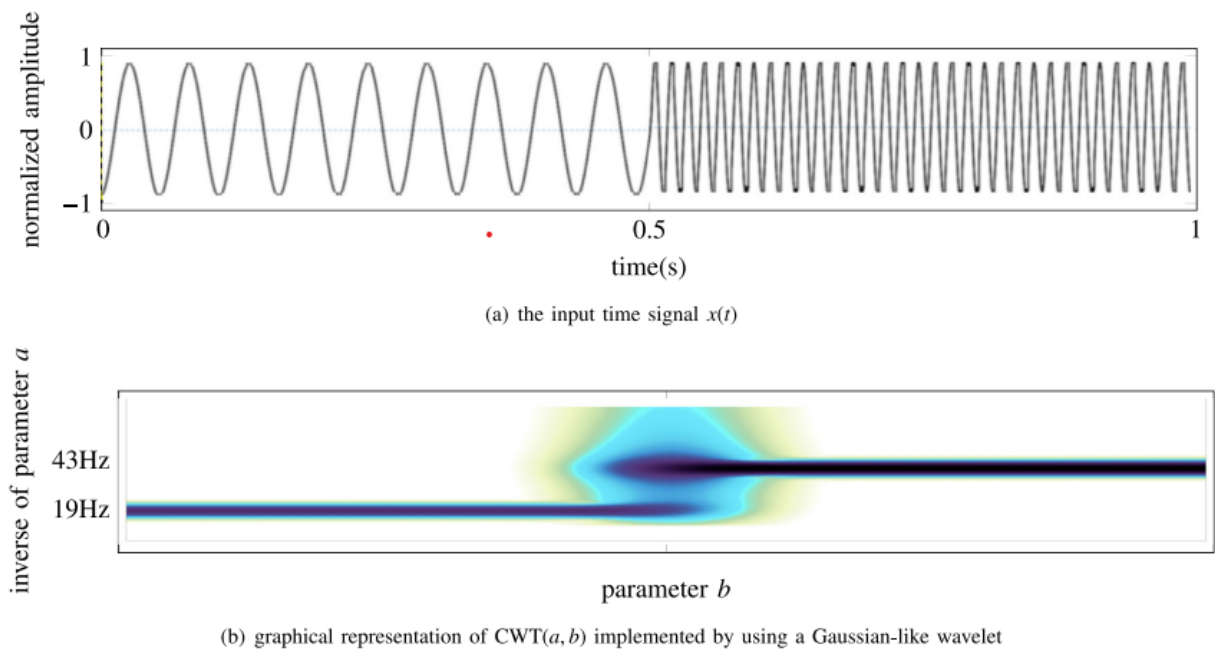


Figura 11: a) Sinal de entrada $x(t)$ b) Representação gráfica da CWT do sinal

Fonte: Guido, 2022

Essa nova base de funções, usada para representar qualquer outra função, introduz então a função wavelet, que molda a janela em uma onda que pode ser deslocada e dilatada como se queira. Na Figura 12 tem-se um exemplo de uma dessas funções.

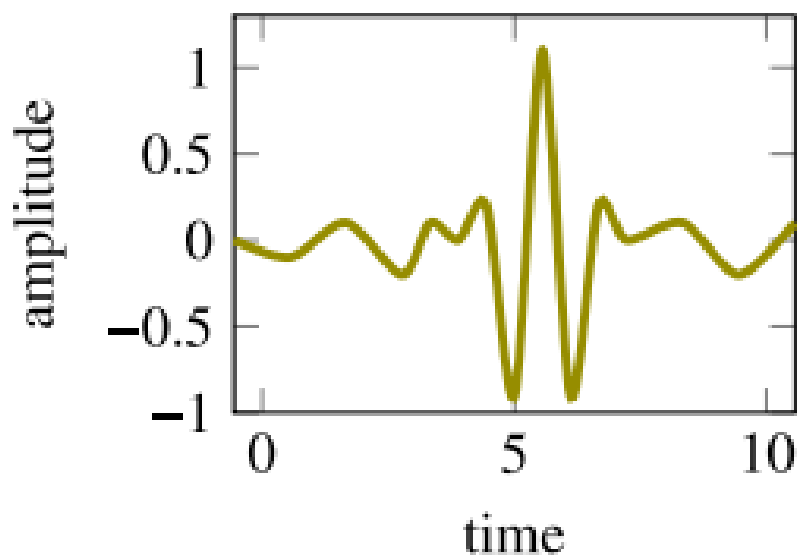


Figura 12: Exemplo de função wavelet Meyer

Fonte: Guido, 2022

Logo, dada uma função wavelet $\psi(t)$, sua janela pode ser modulada através da inserção

de um novo parâmetro b , que indica seu deslocamento, e um parâmetro a , que indica sua dilatação. A Figura 13 ilustra o funcionamento dessa janela. Dessa forma, a Transformada Wavelet Contínua (CWT) é definida como:

$$\psi(t) = \psi(t, a, b) = \psi\left(\frac{t-b}{a}\right) \Leftrightarrow CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \quad (9)$$

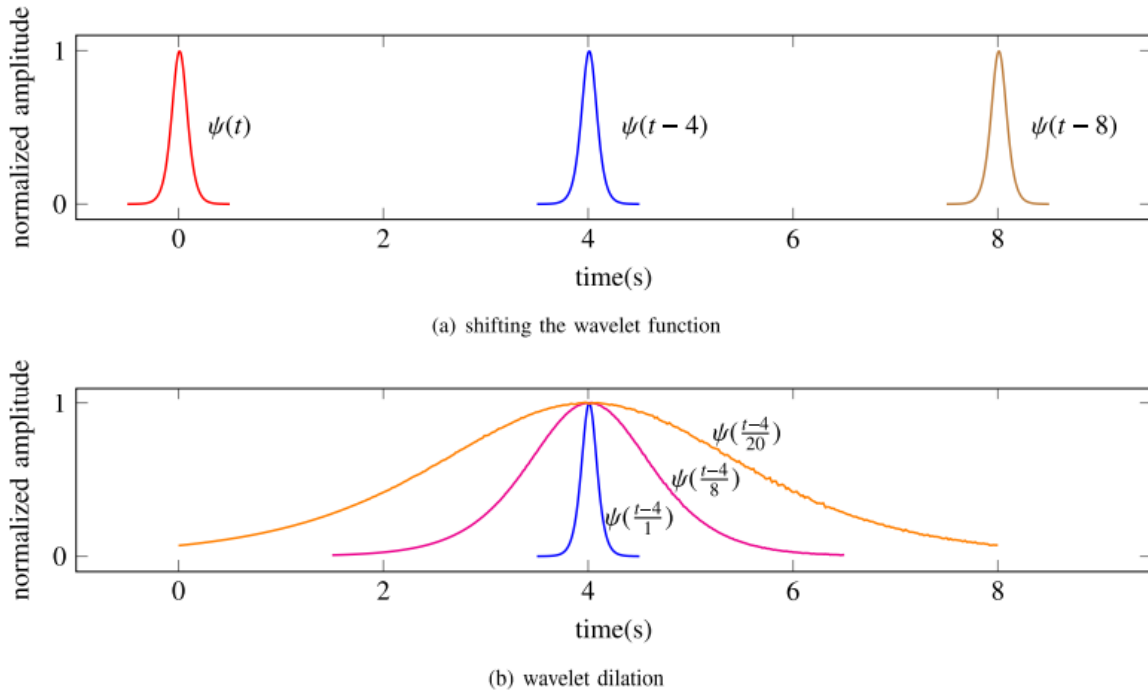


Figura 13: a) Influência do parâmetro a na função ψ b) Influência do parâmetro b na função ψ

Fonte: Guido, 2022

A capacidade de dilatar e transladar a janela (ou wavelet) permite uma melhor resolução em diferentes escalas, o que é essencial em muitos sinais com características transitórias. Novamente, entretanto, deve-se atentar que não é possível realizar computacionalmente a CWT para todos os valores possíveis de a e b . A solução seria discretizar tais valores, o que é prontamente realizado pela Transformada Wavelet Discreta (DWT). Esta é feita continuamente dividindo o parâmetro a por um fator de 2, o que implica em larguras de banda que diminuem progressivamente pela metade.

A DWT, contudo, não é capaz de cobrir o início da faixa espectral, requerendo a definição de uma segunda função ϕ , denominada função scaling, responsável por fechar a base de funções. A Figura 14 ilustra a necessidade da função scaling.

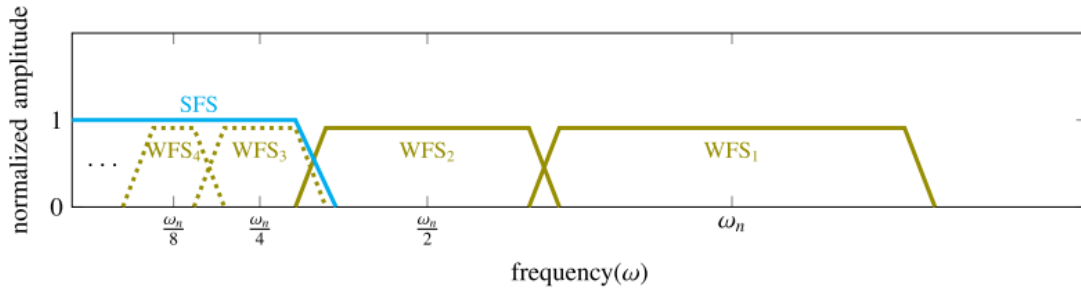


Figura 14: Em ciano, representação da função scaling $\phi(t)$. Em amarelo, função wavelet $\psi(t)$ com sua largura diminuindo pela metade. Note que para a wavelet chegar na função scaling, seria necessário um número infinito de interações

Fonte: Guido, 2022

Além disso, vale notar que, na DWT, a discretização ocorre apenas no domínio do sinal de saída, mantendo o sinal de entrada contínuo. Para os processos computacionais que se pretende implementar, o ideal é que todos os elementos sejam discretos – isto é, o sinal de entrada, as funções scaling e wavelet (respectivamente, também chamadas de wavelet-pai e wavelet-mãe) e o sinal de saída. Essa é a proposta da Transformada Wavelet Discreta no Tempo (DTWT).

O processo de discretização do sinal é relativamente simples, sendo realizado com base no teorema de Nyquist. No entanto, discretizar as funções wavelet não segue o mesmo procedimento. Em vez de utilizar as funções ϕ e ψ , como é feito na CWT e na DWT, a DTWT emprega um conjunto de filtros para análise, composto por um filtro passa-baixas $h[\cdot]$ e um filtro passa-altas $g[\cdot]$. Embora seja possível aplicar a DTWT por meio de conceitos de álgebra linear aplicados às funções wavelet, para o escopo deste projeto, optamos pela abordagem de sucessivas convoluções com filtros.

De forma análoga, a Transformada Wavelet Inversa Discreta no Tempo (IDTWT), que reconstrói um sinal do domínio tempo-frequência para o domínio do tempo, também pode ser moldada em termos de sinais de entrada e saída discretos, e usa-se um conjunto de filtros para síntese, denominados $\bar{h}[\cdot]$ e $\bar{g}[\cdot]$. A união dos filtros de análise e de síntese constitui o banco de filtros wavelet; e, se esse banco for ortogonal (isto é, se $h[\cdot]$ e $g[\cdot]$ forem ortogonais), ele é classificado como um banco de filtros espelhados em quadratura (QMF).

O uso de filtros QMF é particularmente relevante devido a uma propriedade única: caso um banco de filtros ortogonal seja um banco de filtros wavelet, então a resposta nas frequências

de $g[\cdot]$ é nula na frequência $\omega = 0$. Essa propriedade é chamada de momento nulo dos filtros wavelet. Uma outra propriedade importante de bancos de filtros wavelet ortogonais é que a definição de qualquer um dos filtros implica na definição de todos os outros. Isso é possível devido às relações:

$$g[k] = (-1)^k h[N - k - 1] \quad 0 \leq k \leq N - 1 \quad (10)$$

$$\bar{h}[k] = h[N - k - 1] \quad 0 \leq k \leq N - 1 \quad (11)$$

$$\bar{g}[k] = (-1)^{k+1} h[k] \quad 0 \leq k \leq N - 1 \quad (12)$$

Para realizar a filtragem no domínio espectraltemporal, o procedimento usual consiste em aplicar a WT, realizar a filtragem por meio da multiplicação com uma função de filtragem e, em seguida, executar a Transformada Wavelet Inversa (IWT). Embora esse processo seja computacionalmente custoso, é importante lembrar que, para determinados filtros, a filtragem pode ser simplificada à convolução do sinal original com o kernel do filtro. Para os fins deste projeto, concentraremos nossa atenção em filtros wavelet que atendam a essa propriedade.

Contudo, a operação de convolução geralmente resulta em um sinal com um número de amostras maior que o sinal original. Essa discrepância pode ser corrigida por meio do downsampling — o descarte de algumas amostras — que, no caso particular da Transformada Wavelet, é realizado com um fator de 2. Assim, a filtragem baseada em convolução seguida por downsampling constitui a essência da DTWT, e essa abordagem pode ser implementada de forma eficiente utilizando o Algoritmo de Mallat.

O Algoritmo de Mallat utiliza os filtros $h[\cdot]$ (passa-baixas) e $g[\cdot]$ (passa-altas), os quais são projetados como filtros de meia-banda, garantindo a análise com a resolução requerida. Por exemplo, suponha um sinal $s[\cdot]$ com M amostras e frequência máxima T . A convolução do sinal com o filtro passa-baixas $h[\cdot]$ gera um sinal com $M/2$ amostras que cobre as frequências de 0 até $T/2$ Hz; enquanto a convolução com o filtro passa-altas $g[\cdot]$ gera outro sinal com $M/2$ amostras, contendo as frequências de $T/2$ até T Hz. A DTWT resulta, então, da concatenação (ou combinação) desses sinais.

O Algoritmo de Mallat impõe a condição de que o sinal de entrada possua um número de amostras M que seja uma potência de 2. Além disso, o algoritmo simplifica o procedimento

de convolução e downsampling por meio de uma multiplicação matricial. Assim, pode-se representar o processo como:

$$A[\cdot][\cdot]S[\cdot] = R[\cdot] \quad (13)$$

onde $A[\cdot][\cdot]$ representa a matriz formada pelos filtros de análise, $S[\cdot]$ representa o sinal de entrada, e $R[\cdot]$ representa a saída da transformação. A Figura 15 traz um esquema que ilustra esse procedimento. Para montar a matriz $A[\cdot][\cdot]$ deve-se:

1. **Escrever os filtros:** Inicialmente, os filtros $h[\cdot]$ e $g[\cdot]$ são dispostos em linhas consecutivas da matriz, nessa ordem, completadas com zeros.
2. **Shift dos filtros:** A cada novo par de linhas, os filtros são deslocados (shift) para a direita, adicionando-se, de forma cumulativa, um par de zeros antes dos filtros.
3. **Wraparound:** Se os índices excedentes do filtro ultrapassarem o tamanho da matriz, realiza-se o wraparound, escrevendo esses índices no início da linha.
4. **Filtro maior que o sinal:** Caso o tamanho do filtro seja superior ao do sinal, o sinal pode ser repetido ou preenchido de forma adequada para que a multiplicação matricial seja viável.

$$\underbrace{\begin{pmatrix} h_0 & h_1 & h_2 & \dots & \dots & \dots & h_{N-1} & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ g_0 & g_1 & g_2 & \dots & \dots & \dots & g_{N-1} & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & \dots & \dots & \dots & h_{N-1} & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & \dots & g_{N-1} & 0 & 0 & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_{N-2} & h_{N-1} & 0 & 0 & \dots & \dots & \dots & 0 & 0 & h_0 & h_1 & \dots & \dots & h_{N-4} & h_{N-3} \\ g_{N-2} & g_{N-1} & 0 & 0 & \dots & \dots & \dots & 0 & 0 & g_0 & g_1 & \dots & \dots & g_{N-4} & g_{N-3} \end{pmatrix}}_{M \times M \text{ matrix } A[\cdot][\cdot]} \cdot \underbrace{\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ \dots \\ \dots \\ \dots \\ s_{M-2} \\ s_{M-1} \end{pmatrix}}_{\text{input}(S[\cdot])} = \underbrace{\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \dots \\ \dots \\ \dots \\ r_{M-2} \\ r_{M-1} \end{pmatrix}}_{\text{output}(R[\cdot])} .$$

Figura 15: Ilustração da utilização do Algoritmo de Mallat

Fonte: Guido, 2022

A partir da saída da primeira multiplicação matricial, pode-se começar a montar a árvore de decomposição da DTWT, como ilustrado na Figura 16. Ao aplicar a DTWT em determinado nível de resolução j , a folha da esquerda, associada ao filtro $h[\cdot]$, é formada pelos índices pares de $R[\cdot]$, enquanto a folha da direita, associada ao filtro $g[\cdot]$, é formada pelos

índices ímpares. Ao fim da aplicação da DTWT em certo nível, o resultado final é obtido ao se concatenar as folhas da árvore, da esquerda para a direita e de baixo para cima.

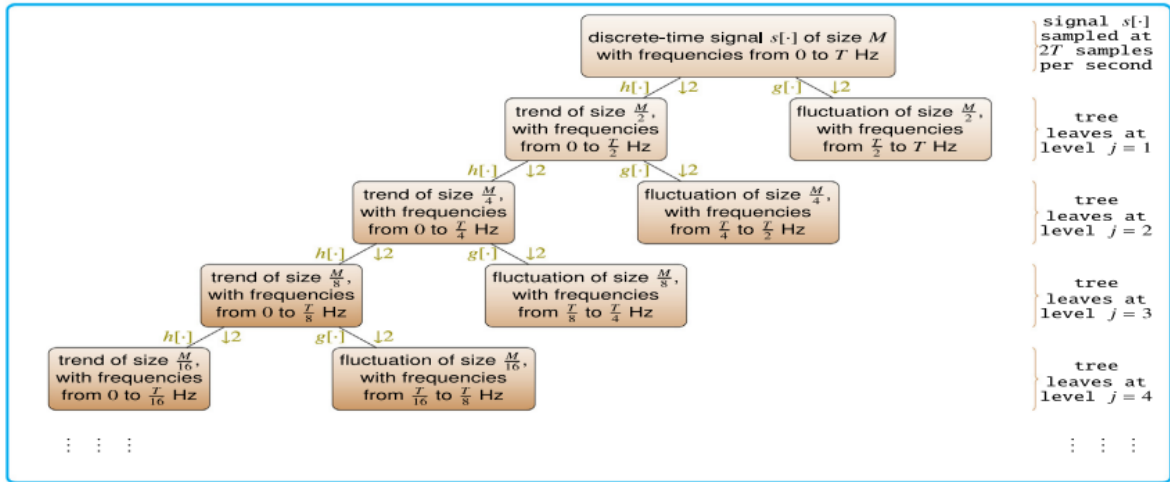


Figura 16: Árvore de decomposição da DTWT ilustrada até o nível de resolução $j = 4$. As setas viradas para baixo seguidas pelo número 2 representam downsamplings por 2

Fonte: Guido, 2022

O Algoritmo de Mallat também pode ser estendido para o cálculo da IDTWT. Isso ocorre porque, se representarmos o processo de decomposição da DTWT por meio da multiplicação matricial, tem-se a Equação 13. Então, para reconstruir o sinal original, temos

$$S[\cdot] = A^{-1}[\cdot][\cdot] R[\cdot]. \quad (14)$$

No caso de interesse, onde os bancos de filtros wavelet são bancos de filtros QMF (ortogonais), vale que $A^{-1} = A^T$. Dessa forma, o cálculo da IDTWT pode ser realizado aplicando o procedimento inverso da DTWT, utilizando a transposta da matriz de filtros. Vale ressaltar que, como $R[\cdot]$ representa o sinal transformado, seus índices estão alternados entre as folhas dos dois filtros de análise na árvore de decomposição da DTWT.

4.4 FT de Sinal Unidimensional de Ressonância Magnética (RM)

Em conjunto com o estudo dos conceitos relevantes, foi proposto um problema inicial da FT de um sinal unidimensional de RM, com o intuito de colocar em prática os conhecimentos obtidos e preparar a base computacional para a futura implementação dos códigos relevantes de processamento de imagens usando WT.

O sinal de RM pode ser simulado a partir da fórmula:

$$S(t) = S_0 e^{-\frac{t}{T}} e^{i\omega t} \quad (15)$$

onde T indica o tempo de relaxação, ω a frequência angular em rad/s, e S_0 a amplitude inicial. Com base nos valores obtidos na literatura considerando as aplicações clínicas em campos magnéticos típicos destes sistemas, esse sinal deve ser gerado com 2048 pontos, igualmente espaçados entre si por um $\delta t = 5 * 10^{-4}$ segundos. A partir daí, deve-se aplicar a FT deste sinal, e analisar o que ocorre com o sinal transformado conforme os parâmetros ω , T e S_0 são variados.

Para isso, criamos um código simples capaz de alterar os parâmetros do sinal, plotá-lo e plotar sua FT. As bibliotecas utilizadas foram `numpy` e `matplotlib`. Os códigos estão divididos em 2 arquivos, cujos nomes são `main.py` e `useful_lib.py`, organizados da seguinte forma:

- `useful_lib.py`: Abre um arquivo chamado `first_parameters.txt` que contém os parâmetros iniciais do sinal simulado, e os coloca em variáveis dentro do código. Implementa funções para simulação do sinal, aplicação da FT sobre ele, e plotagem de ambos o sinal e sua FT.
- `main.py`: Menu que permite ao usuário visualizar o sinal, alterar seus parâmetros, e visualizar sua FT.

O arquivo com código a ser executado pelo usuário é o `main.py`, enquanto a `useful_lib.py` é utilizada como biblioteca de suporte pelo `main`. A execução do programa começa com a exibição de um menu, mostrado na Figura 17. Caso o usuário execute a opção "S" sem antes modificar os parâmetros do sinal, o sinal é gerado utilizando os parâmetros contidos no arquivo `first_parameters.txt`.

```
Bem-vindo à resolução do Problema 1!
Escolha sua opção:
    S: Mostra o sinal gerado.      P: Modifica os parâmetros do sinal.
    F: Mostra a FT do sinal.
    M: Mostra este menu novamente. X: Termina o programa.

Digite sua escolha: █
```

Figura 17: Menu inicial do programa

Caso o usuário execute a opção "P", ele é levado a outro menu, mostrado na Figura 18, onde pode alterar os parâmetros de simulação do sinal.

```
Você escolheu mudar os parâmetros do sinal.
Os parâmetros atuais são:
  (1) S_0 = 1.0   (3) omega = 0.0 (5) n_pontos = 2048
  (2) T = 0.92   (4) dt = 0.0005

Escolha sua opção:
  *: Digite um número de 1 a 5 para mudar o parâmetro correspondente.
  H: Obter ajuda a respeito do que é cada parâmetro.      R: Reseta os parâmetros para os originais.
  M: Mostra este menu novamente.                        X: Sair deste menu.

Digite sua escolha: █
```

Figura 18: Menu para manipulação dos parâmetros

Os parâmetros que podem ser mudados são aqueles presentes na Equação 15, assim como número de pontos simulados, e o espaçamento de tempo entre eles. Não entraremos em muitos detalhes a respeito destes dois últimos parâmetros, pois o foco desta seção é na FT. Caso o leitor se interesse, ele pode manipular esses parâmetros para testar o Teorema de Nyquist. Neste segundo menu, caso o usuário execute a opção "H", informações a respeito de cada parâmetro serão escritas na tela, como na Figura 19.

```
Estes parâmetros controlam o comportamento do sinal simulado.
  S_0: Indica a amplitude máxima do sinal.
  T: Indica o tempo de relaxação.
  omega: Indica a frequência de oscilação do sinal em rad/s.
  dt: Indica o espaçamento em tempo entre os pontos usados pra gerar o sinal.
  n_pontos: Indica quantos pontos serão usados para simular o sinal.
O sinal é gerado pela função:  $S(t) = S_0 * \exp(-t/T) * \exp(i * \omega * t)$ 

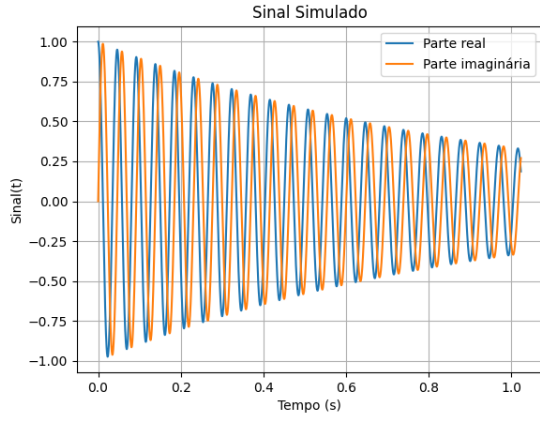
Digite sua escolha: █
```

Figura 19: Exibição das informações de cada parâmetro

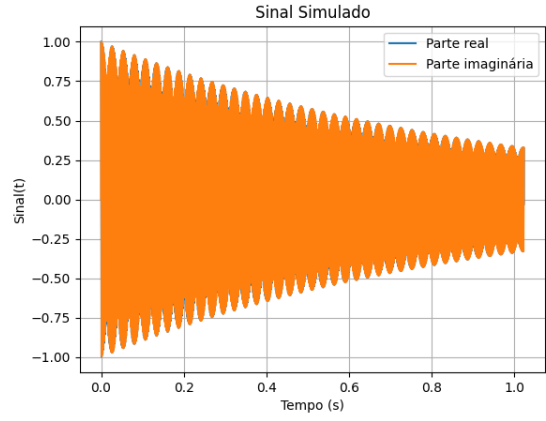
Após demonstrar a interface do programa, passamos agora à análise de como a alteração dos parâmetros influencia tanto o sinal gerado quanto sua FT. A Figura 20 apresenta gráficos de sinais gerados pelo programa com diferentes configurações. A partir desses gráficos, pode-se inferir como cada parâmetro influencia o sinal:

- ω determina a frequência de oscilação do sinal, fazendo com que o sinal oscile mais rapidamente à medida que ω aumenta.

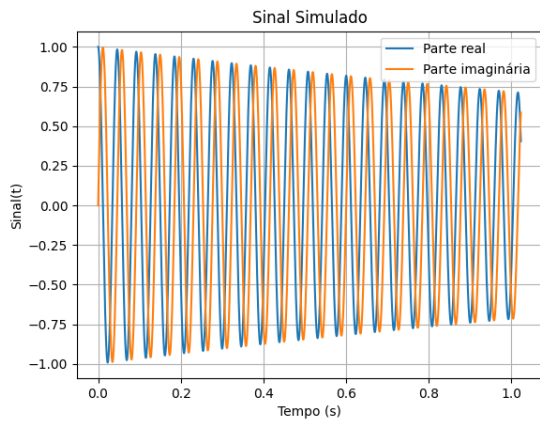
- T , que representa o tempo de relaxação, indica quanto tempo o sinal permanece oscilando antes de atingir o equilíbrio; um valor maior de T faz com que o sinal oscile por mais tempo.
- S_0 define a amplitude inicial do sinal, e seu aumento resulta em um sinal de maior amplitude durante toda a sua duração.



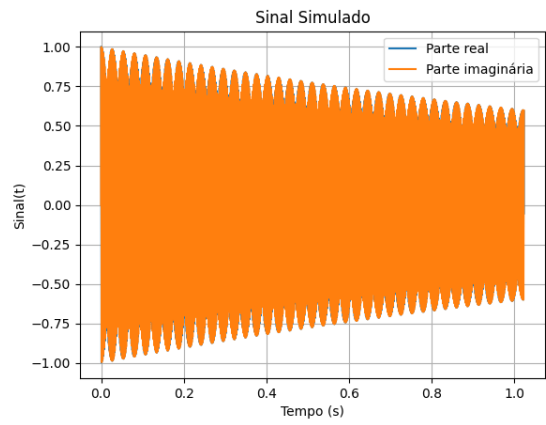
(a) Sinal com $\omega = 136$



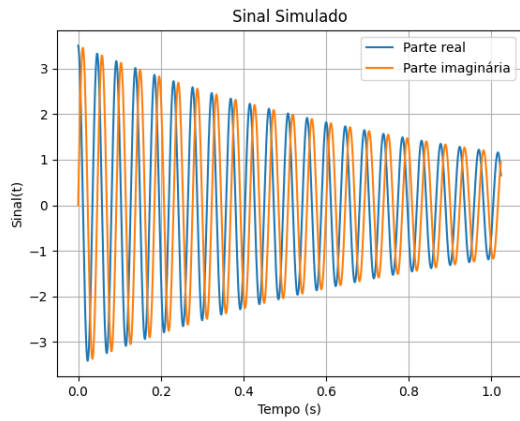
(b) Sinal com $\omega = 3200$



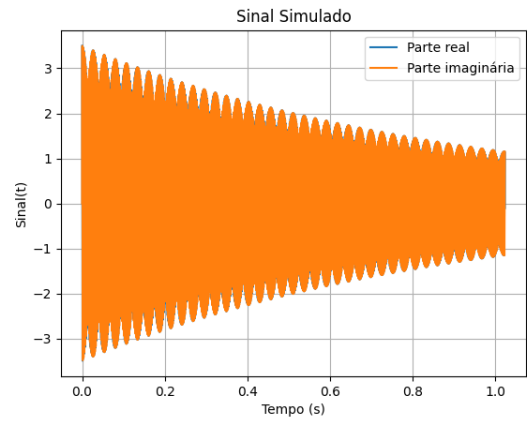
(c) Sinal com $\omega = 136$ e $T = 3.0$



(d) Sinal com $\omega = 3200$ e $T = 3.0$



(e) Sinal com $\omega = 136$ e $S_0 = 3.5$

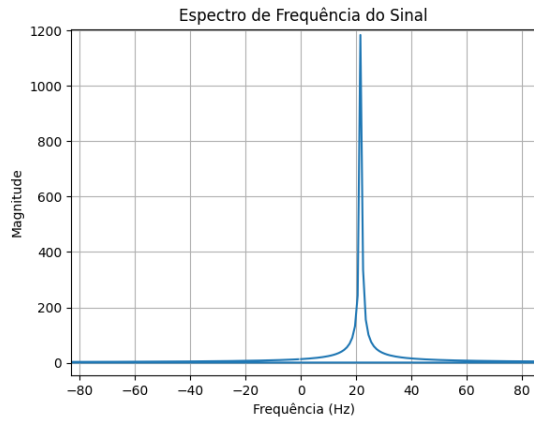


(f) Sinal com $\omega = 3200$ e $S_0 = 3.5$

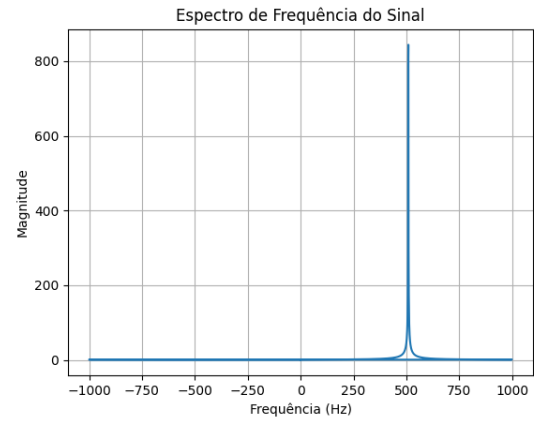
Figura 20: Sinais com diferentes parâmetros

A Figura 21 exibe os gráficos das FTs dos sinais apresentados na Figura 20. As principais inferências que podem ser extraídas são as seguintes:

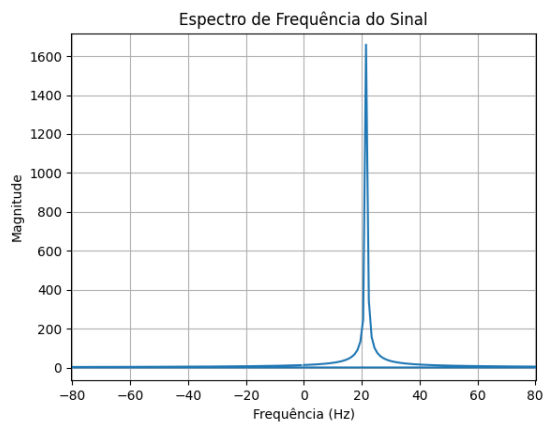
- O aumento de T e S_0 eleva a amplitude do pico da transformada, pois o sinal possui uma amplitude maior (ou permanece oscilando por mais tempo, no caso de T).
- A variação de ω desloca o pico para a posição correspondente, conforme definido na Equação 7.



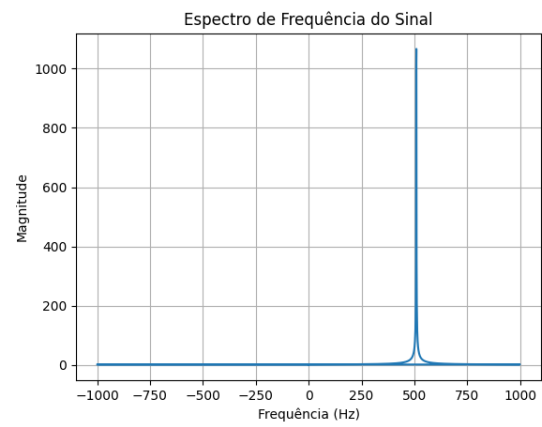
(a) FT de sinal com $\omega = 136$



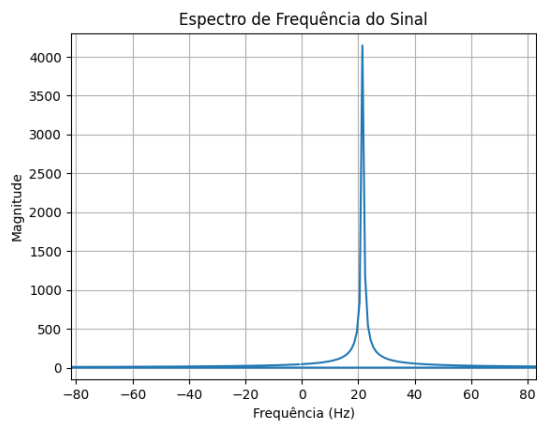
(b) FT de sinal com $\omega = 3200$



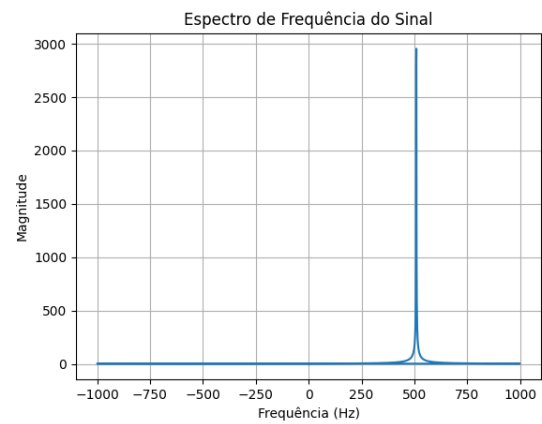
(c) FT de sinal com $\omega = 136$ e $T = 3.0$



(d) FT de sinal com $\omega = 3200$ e $T = 3.0$




(e) FT de sinal com $\omega = 136$ e $S_0 = 3.5$



(f) FT de sinal com $\omega = 3200$ e $S_0 = 3.5$

Figura 21: FT de sinais com diferentes parâmetros

4.5 Leitura e FT de Imagens de RM

Já foram implementadas também funções para a leitura e aplicação da FT em imagens digitais. Segue em anexo a biblioteca implementada e utilizada pelo aluno: 

Essa biblioteca, que faz uso das bibliotecas `matplotlib`, `math`, `numpy`, `copy` e `random`, representa imagens digitais através da classe `Image`, cuja inicialização é dada por `__init__(self, MeuArquivo, from_fourier = False, matriz_complex = None)`, onde

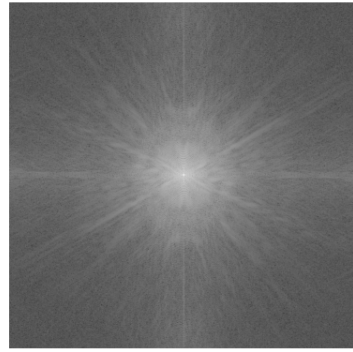
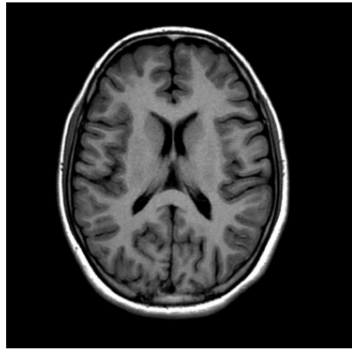
`MeuArquivo` indica o nome do arquivo, `from_fourier` indica se a imagem está no domínio da frequência, e `matriz_complex` é uma variável que guarda o sinal com as componentes complexas, caso a imagem seja fruto de uma IFT. Além disso, é definida também uma classe especial, `Fourier_Image`, cuja inicialização é dada por `__init__(self, imagem)`, onde `imagem` é uma instância da classe `Image`.

A classe `Fourier_Image` possui um método da seguinte forma: `representacao(self)`. Esse retorna uma instância da classe `Image`, que pode ser usada para visualizar o resultado da transformada de Fourier. A Figura 22 mostra um exemplo do uso da biblioteca.

```
1  import my_image_lib_0_5 as img
2
3  def main():
4      imagem = img.Image("Input\\RM.pgm")
5      img.print_grayscale_image(imagem)
6
7      imagem_fourier = img.Fourier_Image(imagem)
8      representa1 = imagem_fourier.representacao()
9      img.print_grayscale_image(representa1)
10
11  main()
```

Figura 22: Exemplo de uso da biblioteca

A biblioteca também possui uma função chamada `print_grayscale_image(imagem)`, onde `imagem` novamente se refere a uma instância da classe `Image`. Ela é responsável por imprimir a imagem referenciada na tela. A Figura 23 contém as figuras abertas pelo código presente na Figura 22.



(a) Imagem de RM aberta pela biblioteca

(b) FT de a)

Figura 23: Leitura e FT de imagem de RM

4.6 Próximos Passos

A partir dos resultados obtidos na análise da FT aplicada ao sinal unidimensional de RM, e da implementação de uma biblioteca que pode abrir, manipular e aplicar a FT em imagens digitais, os próximos passos do projeto se concentrarão em:

1. **Estudo guiado:** Dar prosseguimento ao estudo da WT, focando nas diferentes famílias wavelets e suas aplicações.
2. **Integração da Transformada Wavelet (WT):** Desenvolver e implementar a Transformada Wavelet, explorando diferentes famílias de wavelets, de forma a comparar os resultados com os obtidos pela FT. Essa etapa incluirá a análise do desempenho dos filtros e a adaptação dos parâmetros para maximizar a resolução tanto temporal quanto espectral.
3. **Processamento de Imagens de RM:** Estender a abordagem dos sinais unidimensionais para a análise e filtragem de imagens de RM. Isso envolverá a implementação da WT em sinais bidimensionais, permitindo a segmentação e realce de características de interesse nas imagens.

4.7 Repositório Online

Todos os códigos, scripts e documentos utilizados para o desenvolvimento deste projeto estão organizados em um repositório online, que pode ser acessado por meio do seguinte link:

Neste repositório, o leitor encontrará:

- O arquivo `main.py`, que contém o menu interativo e a integração das funcionalidades.
- A biblioteca `useful_lib.py`, que implementa as funções para simulação do sinal, aplicação da FT e visualização dos resultados.
- O arquivo `first_parameters.txt`, com os parâmetros iniciais da simulação.
- Documentação adicional e exemplos de execução para facilitar a reprodução dos resultados.
- Materiais para estudo que detalham a Transformada Wavelet e o processo de Imagens por Ressonância Magnética Funcional.
- Biblioteca para processamento de imagens digitais.

5 Conclusão

Neste relatório, foram apresentados os resultados iniciais de um trabalho desenvolvido no âmbito de um Projeto de Iniciação Científica, cujo principal objetivo é a aplicação da Transformada Wavelet ao processamento de imagens de Ressonância Magnética (RM). O relatório iniciou-se com noções de processamento de sinais, incluindo a distinção entre sinais analógicos e digitais, o papel dos sistemas na manipulação de informações e os princípios da digitalização com base no teorema de Nyquist.

Partindo destes conceitos, foi realizada uma proposta para a análise espectral de um sinal unidimensional de RM utilizando a FT. Com base nos resultados, é possível discorrer sobre a influência dos parâmetros S_0 , T e ω tanto no sinal no domínio do tempo quanto na sua representação no domínio da frequência. Checamos que S_0 e T influenciam a amplitude do sinal transformado, e a mudança de ω acarreta no deslocamento do pico observado.

Na continuação, o relatório apresenta conceitos relacionados com a filtragem digital de sinais, discorrendo sobre a operação de convolução e sobre o Algoritmo de Mallat para a DTWT. Esta técnica, que faz uso de convoluções e de downsampling, simplifica o processamento computacional de sinais e será um subsídio essencial para a implementação futura da Transformada Wavelet no processamento de Imagens de RM.

Enfatiza-se que o projeto, no futuro, irá centrar-se na implementação da Transformada Wavelet no âmbito de imagem, na otimização dos seus algoritmos de processamento e na validação dos resultados com base em dados experimentais. Serão disponibilizadas na Web as ferramentas desenvolvidas pelo aluno, que permitirão replicar os resultados obtidos e incentivar a sua utilização pela comunidade científica.

Pode concluir-se que os trabalhos desenvolvidos até ao momento permitem estabelecer um conjunto de bases teóricas e computacionais que podem viabilizar a aplicação de técnicas mais avançadas de processamento de sinais e imagens em RM.

6 Referências

- HIGUTI, Ricardo Tokio; KITANO, Cláudio. Sinais e sistemas, 2003. Disponível em https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/optoeletronica/sinais_e_sistemas.pdf. Acesso em 12 de março de 2025.
- DE LIMA, Manoel Eusébio. Conversão Digital Analógico e Analógico Digital. Disponível em https://www.cin.ufpe.br/~es238/arquivos/aulas/aula17_conversores_adda.pdf. Acesso em 12 de março de 2025.
- ALBUQUERQUE, Márcio Portes; ALBUQUERQUE, Marcelo Portes. Processamento Digital de Sinais, 2002.
- DE LIMA, Márcio Henrique T.; DOMINGUES, Moisés Ferreira S.; BATISTA, Rogério Menezes. Conversor: um dispositivo que recebe um sinal analógico transformando-o em digital, 2010.
- ALMEIDA, Maria Aparecida Fernandes. Filtragem Digital de Sinais Biomédicos, 1997.
- MARINUS, Nustenil Segundo de Moraes Lima. Processamento Digital de Sinais Aplicados a Efeitos de Áudio.
- TONIDANDEL, Danny Augusto Vieira. Entre o Real e o Complexo: Uma Visão Unificada do Conceito de Transformada, 2011.
- FECHINE, Joseana Macêdo. A Transformada de Fourier e Suas Aplicações, 2010.
- GUIDO, Rodrigo Capobianco. Wavelets behind the scenes: Practical aspects, insights, and perspectives, 2022.