

Processamento digital de sinais de áudio utilizando a Transformada de Fourier e Filtro Passa Baixa

Emanuel Angelim Pereira, Paulo Armando Cavalcante Aguilar

¹Universidade Federal do Ceará (UFC) - Campus Quixadá
Av. Jose de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará 63902-580

emanuel_ec@alu.ufc.br, cavalcante.aguilar@gmail.com

Abstract. *This paper reports procedures and results obtained through the use of MATLAB software tools in the digital processing of audio signals.*

Resumo. *Este artigo relata procedimentos e resultados obtidos através da utilização de ferramentas do software MATLAB no processamento digital de sinais de áudio.*

1. Introdução

A Transformada Rápida de Fourier (FFT), é um dos algoritmos mais eficientes para se calcular a Transformada discreta de Fourier. Esse algoritmo tem uma série de aplicações na área de sinais e sistemas, com destaque no processamento digital de sinais, resolução de equações diferenciais e parciais, e multiplicação de grandes números inteiros.

Alguns problemas são muito difíceis de solucionar diretamente, a análise do espectro da frequência de áudios é um desses problemas. Neste trabalho é mostrado uma exemplo de utilização de ferramentas do MATLAB, especialmente a Transformada de Fourier, para facilitar o processamento digital de áudio.

2. Análise de sinais de áudio utilizando FFT no MATLAB

Em um sinal de áudio, há milhares, ou até mesmo milhões de frequências de sinais misturadas, essas frequências passam por um amplificador de áudio e excitam um "speaker", gerando um sinal sonoro. No sinal sonoro, pode predominar sinais com altas frequências, que são áudios mais agudos, ou sinais com baixas frequências, que são áudios mais graves.

2.1. Áudio no MATLAB

Na Figura 1, podemos observar um exemplo de um sinal de áudio, que foi capturado no momento da fala de um ser humano, e gerado o gráfico no software MATLAB [?]. Para gerar essa forma de onda, utilizamos a função `audioread(filename)`, que retorna a forma de onda de um sinal de áudio juntamente com sua frequência.

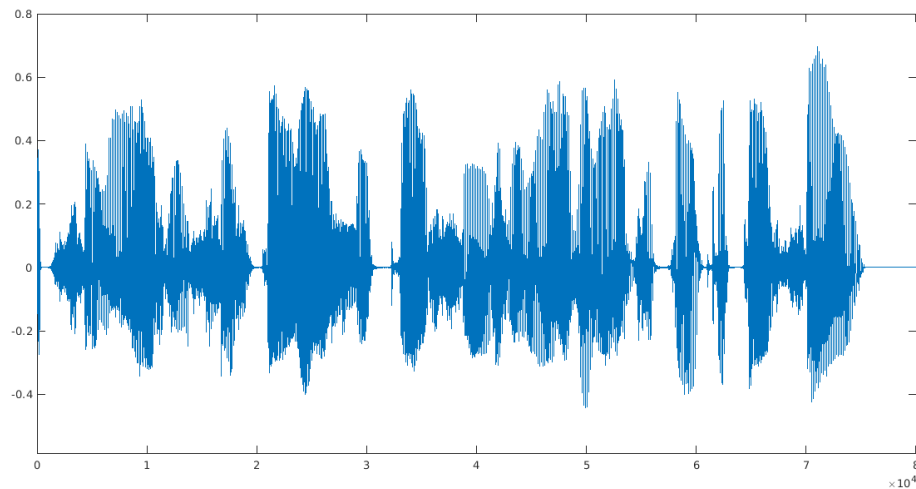


Figura 1. Formato de onda de um sinal de áudio

2.2. Transformada Rápida de Fourier

É importante ter em mente que um sinal de áudio pode ser representado tanto no domínio do tempo como também no domínio da frequência. Uma forma de fazer essa transição de domínios é utilizando a transformada de Fourier que é definida como:

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

Essa equação matemática pode ser utilizada na transição do domínio tempo para domínio frequência, já para fazer a transição contrária, pode ser utilizado a transformada inversa de Fourier que é definida como:

$$f(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

Nas bibliotecas do software MATLAB, existem funções implementadas que realizam esse cálculo e retornam o resultado, são elas: `fft()` e `ifft()`. Utilizando a `fft()` para calcular a Transformada de Fourier do sinal, é obtido o sinal da Figura 2.

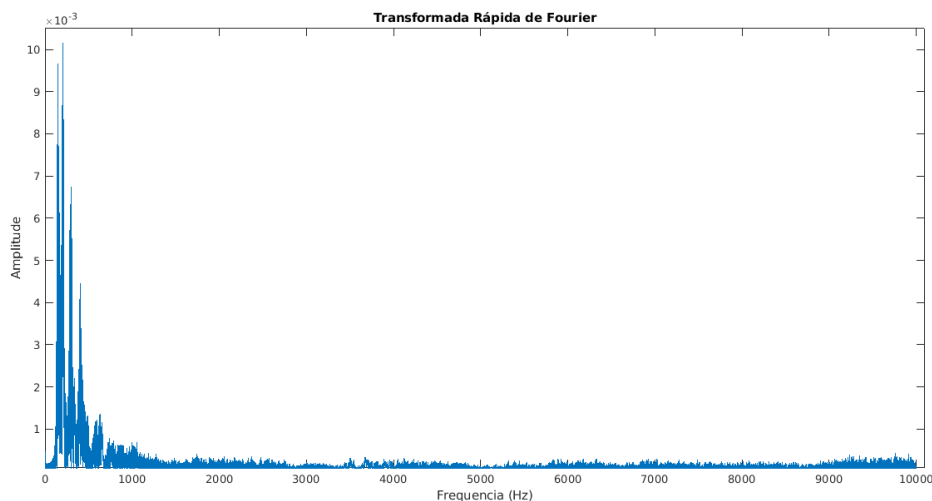


Figura 2. Análise do espectro da frequência do sinal

Após aplicar a Transformada Rápida de Fourier no sinal de áudio, o sinal passa do domínio tempo para o domínio frequência. Ao analisar o gráfico da Figura 2, podemos perceber que esse áudio possui frequências que variam de 1 Hertz, até 10000 Hertz, contudo os sinais que predominam e que tem maior amplitude no sinal estão na faixa de 120 - 700 Hertz.

3. Filtro Passa Baixa

Os Filtro de frequência, a grosso modo, tem a função de eliminar qualquer tipo de frequência que seja acima ou abaixo (dependendo do filtro) da frequência de corte. Contudo, no mundo real, não existe nenhum tipo de filtro que tenha 100% de aproveitamento, porém há modelos que tem um desempenho muito bom, e próximo de um filtro ideal.

No caso do filtro passa baixa, a intenção é eliminar qualquer tipo de frequência que esteja acima da frequência de corte. Neste exemplo iremos adotar 3050 como frequência de corte, isso significa que frequências acima dessa, será eliminada.

3.1. Algoritmo

O primeiro cuidado a ser tomado na utilização desse tipo de filtro, é respeitar o Teorema de Nyquist que afirma que a frequência de amostragem deve ser pelo menos o dobro da maior frequência do sinal, como a frequência de amostragem do exemplo é 22050 Hertz e a frequência máxima do sinal é 10010 Hertz, podemos seguir adiante.

Para a execução da filtragem do sinal, precisamos definir os parâmetros frequência de passagem e frequência de corte, a média simples dessas frequências resultará na frequência de transição intermediária. Na Figura 3, pode ser observado que a frequência de passagem foi definida como 2300 Hertz e a frequência de corte foi definida como 3800 Hertz, logo a frequência de transição, que será onde o sinal será cortado, resulta em 3050 Hertz, fazendo a média simples.

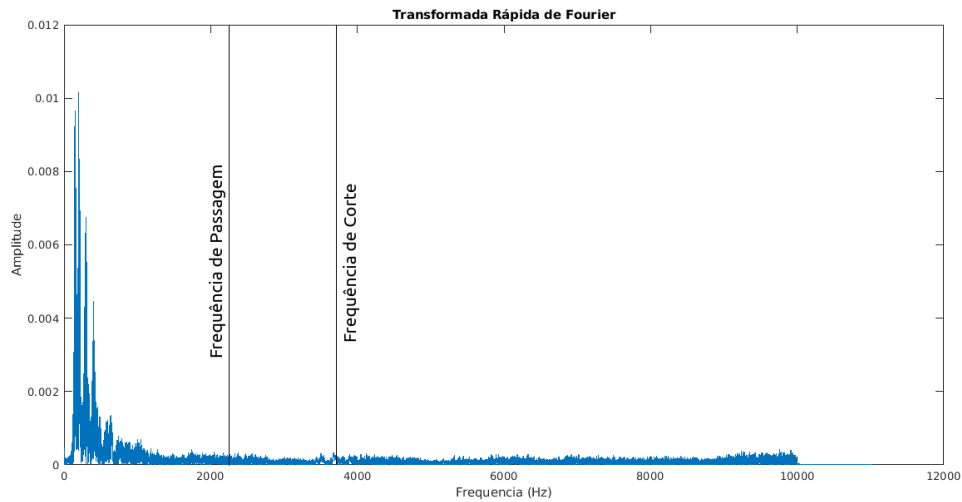


Figura 3. Frequências de passagem e de corte

Este filtro FIR utiliza a função sinc que é definida como:

$$f(x) = \frac{\sin(x\omega)}{x\omega}$$

Como é notável no gráfico da Figura 4, esta função tem um lóbulo principal e outros lóbulos secundários, para a nossa aplicação o que interessa é apenas o lóbulo principal. A maneira mais fácil de eliminar esses lóbulos secundários é utilizar algum tipo de janela. Entre os vários tipos de janela, a mais eficiente nesse caso é a janela de Hamming que é definida como:

$$f(x) = 0.54 + 0.56 \cos(2\pi \frac{x}{y}), 0 < y < x$$

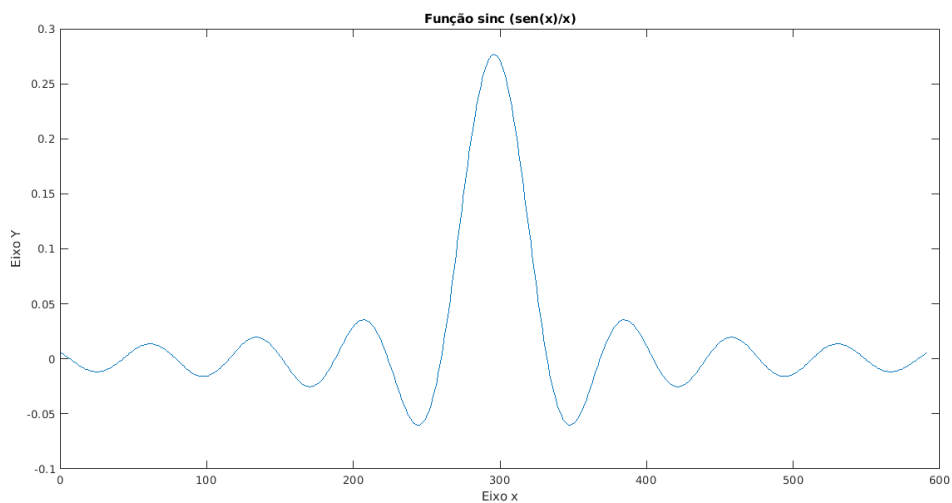


Figura 4. Função sinc(x)

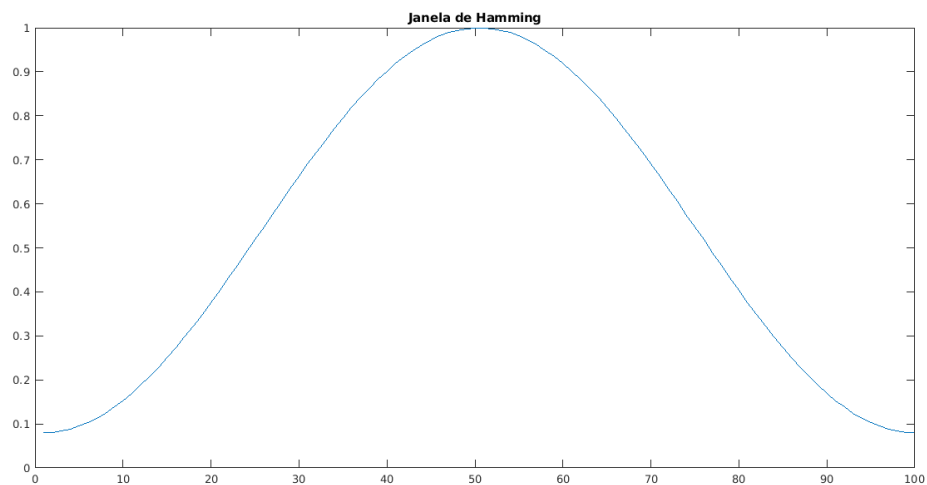


Figura 5. Janela de hamming

Após eliminar os lóbulos secundários da função $\text{sinc}(x)$, o próximo passo é onde ocorre a filtragem propriamente dita do sinal. O processo utilizado é a convolução entre o lóbulo principal da função $\text{sinc}(x)$ e o sinal original. No MATLAB também há a função que realiza a convolução de dois sinais e retorna o resultado. Após realizar a convolução, obtemos o sinal filtrado, para analisar melhor, podemos processar esse sinal com a Transformada Rápida de Fourier e analisar o espectro da frequência do sinal obtido.

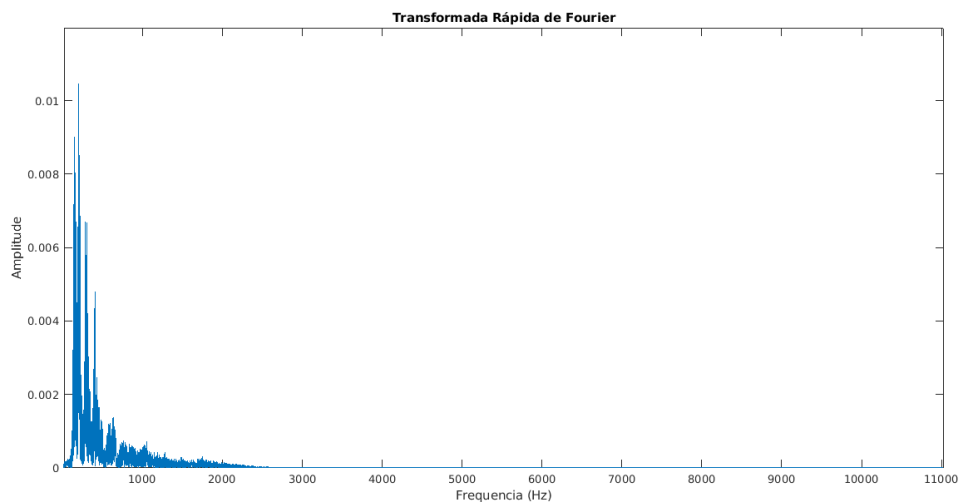


Figura 6. Análise do espectro da frequência do sinal filtrado

Fica bem claro no gráfico da Figura 6, que não existem mais frequências acima da frequência de corte estabelecida. Na prática esse processamento neste sinal, deixou o seu tom acústico bem mais grave, pois tons agudos são formados por altas frequências, que foram eliminadas do sinal.

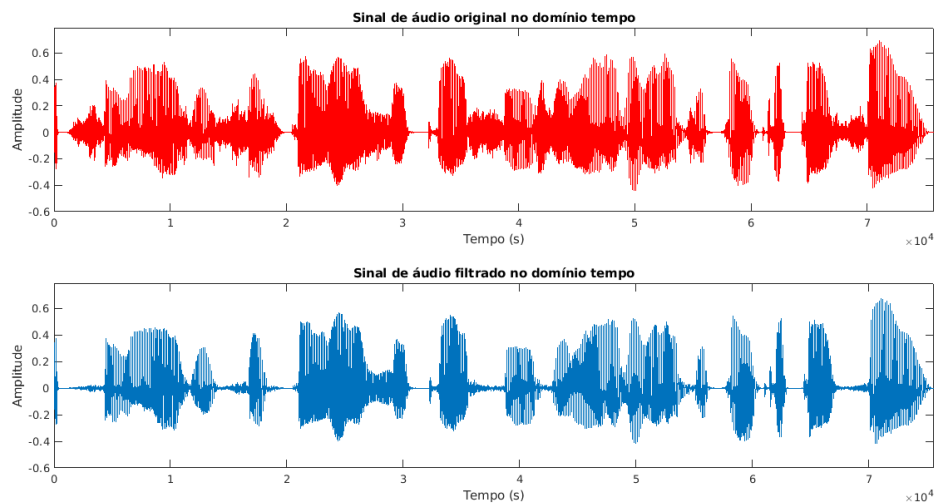


Figura 7. Comparação entre o sinal original e o sinal filtrado

Para uma análise mais completa, foi feito um comparativo entre o sinal original e o sinal filtrado. Na Figura 7, pode ser percebido que o sinal do gráfico em azul tem algumas diferenças em relação ao gráfico em vermelho, principalmente nas faixas onde apresentam frequências mais altas. Já na Figura 8, há o comparativo entre a amplitude das frequências do sinal original, e do sinal filtrado, é notável que no gráfico em azul, a amplitude das frequências mais altas, são bem menores.

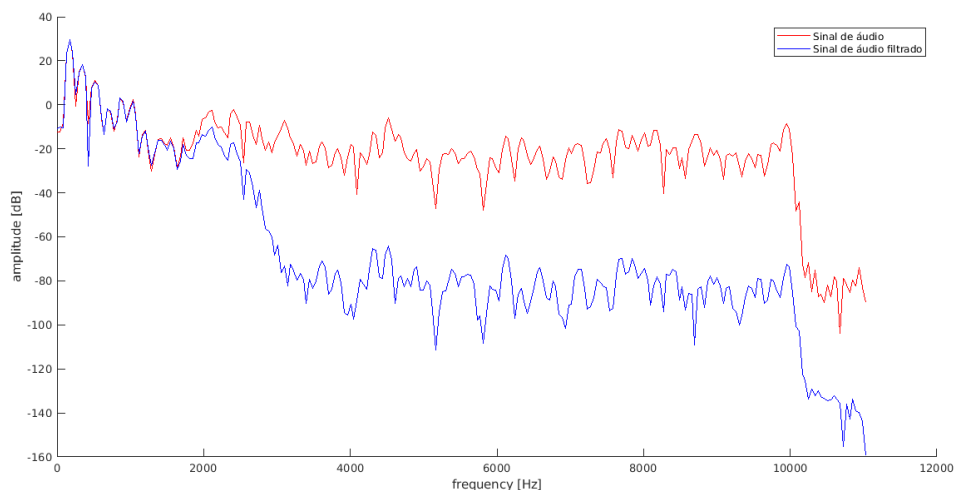


Figura 8. Comparação entre a amplitude das frequências dos sinais

4. Conclusão

Como foi demonstrado, os filtros digitais podem ser bem eficientes na eliminação de frequências não desejadas no sinal. Neste trabalho o filtro foi utilizado para deixar um áudio mais grave, mas pode ser utilizado para eliminar ruídos por exemplo. Uma boa continuação para o trabalho seria incluir outros tipos de filtros, como passa alta, passa baixa e rejeita faixa por exemplo.

5. Referências

- 1 plotly, (2017), Fast Fourier Transform in MATLAB.
<http://plot.ly/matlab/fft>
- 2 Jarno Seppänen, (1999), Audio Signal Processing Basics.
<http://www.cs.tut.fi/sgn/arg/intro/basics.html>
- 3 NTI Áudio, (2015), FFT.
<http://www.nti-audio.com/en/functions/fast-fourier-transform-fft.aspx>
- 4 Marco Aurélio Gonçalves da Silva, Filtros Digitais Aplicados em Sinais de Áudio.
<http://www.gcg.ufjf.br/pub/doc49.pdf>
- 5 UNESP, (2012), A Transformada de Fourier.
<http://www.feis.unesp.br/Home/departamentos/mcap05.pdf>