

Afinador de Ukulele

Valdomiro Botelho Junior, Samuel Fernandes

Resumo— O seguinte trabalho possui por objetivo a implementação prática dos conceitos de Transformada Rápida de Fourier e filtragem digital, e para isso estudará uma solução em python para afinar o instrumento musical ukulele. Para a solução, foi escolhido implementar um algoritmo de Cooley-Tukey do qual é efetuado recursivamente o cálculo da FFT, e a partir dos conceitos de filtro passa faixa, selecionar as frequências interessantes para a implementação do afinador.

I. INTRODUÇÃO

Tendo em vista a abordagem dos conceitos ministrados na disciplina, esse trabalho possui a proposta de utilizar a transformada rápida de Fourier para obter os espectros de cada onda de sinal captada pelo microfone de um notebook e selecionar frequências uteis para afinação de um instrumento musical.

II. TRANSFORMADA RAPIDA DE FOURIER

Uma FFT, ou Transformada Rápida de Fourier, é um método de calcular uma transformada de Fourier discreta a partir de um algoritmo de computador, pensada de modo a minimizar o esforço computacional. Os princípios de resolução de uma transformada de Fourier discreta têm originalmente um número de multiplicações de númerodepontos^2 , o que claramente causa um aumento de operações cada vez maior com o aumento do número de pontos, de forma quadrática. A transformada rápida separa o sinal discreto nos seus índices pares e ímpares, fazendo com que o índice do cálculo do somatório da DFT vá até $N/2 - 1$ ao invés de $N - 1$, onde $N = \text{númerodepontos}$, considerando então a simetria relacionada à periodicidade da FFT em $N/2$ se reduz ainda mais o custo computacional ao não fazer operações desnecessárias. Outra manipulação é a manutenção do númerodepontos em base 2, onde se completa o sinal com “zeros” caso o número de pontos não tenha base dois. Isso é feito para possibilitar a computação da FFT de forma recursiva, a função pode então chamar ela mesma várias vezes, buscando calcular duas FFTs com $N/2$ pontos a cada chamada onde, no fim da recursão, são calculadas N DFTs de

1 ponto apenas. Desta forma o algoritmo cresce de forma logarítmica conforme o número de pontos aumenta, sendo que o número de multiplicações (operação que mais importa no contexto) é $N * \log_2(N)$, o que é muito mais vantajoso em relação ao cálculo original de uma DFT.

O algoritmo de Cooley Tukey é o mais utilizado para efetuar o cálculo de transformada rápida de Fourier e utiliza o conceito de recursão acima descrito para transformar sinais no domínio do tempo para o domínio do espaço.

III. IMPLEMENTAÇÃO DA SOLUÇÃO

Para conseguir afinar o ukulele, foi utilizada a estratégia de captar o áudio ambiente através do microfone de um notebook durante o intervalo de 5 segundos, e a partir dos quadros por segundo (frames) obtidos, efetuar uma transformada rápida de Fourier para se fazer a análise no domínio da frequência desses dados e assim, obter as frequências que correspondem ao som emitido pelas cordas do instrumento, tratar essas frequências de modo que seja informada a frequência que mais se destaca entre as que compõem o som ambiente e informar para o usuário a afinação atual de seu instrumento musical.

Para uma melhor análise do que foi implementado, são geradas figuras que mostram o áudio capturado tanto no domínio do tempo quanto no domínio da frequência. Apenas para fim de comparação, foi efetuada uma plotagem do mesmo sinal no domínio da frequência utilizando o algoritmo FFT disponível na biblioteca numpy.

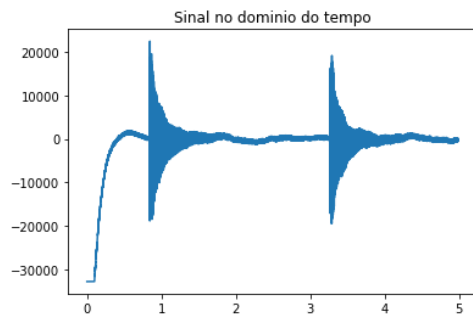


Fig. 1. Curva do sinal obtido no domínio do tempo

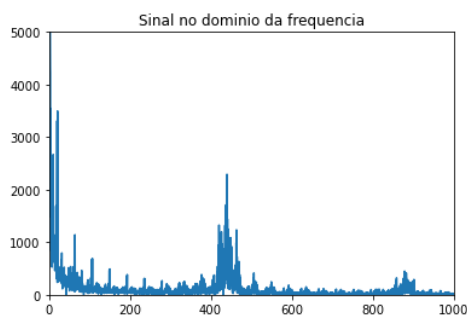


Fig. 2. Curva do sinal obtido no domínio da frequência pelo algoritmo implementado

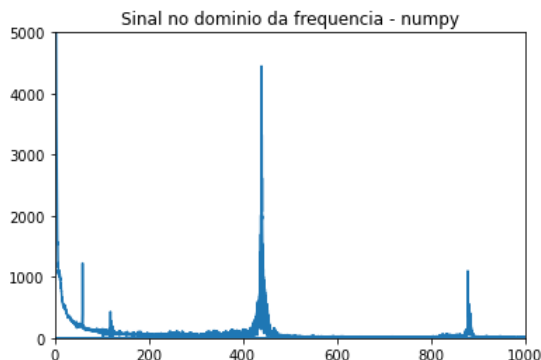


Fig. 3. Mesmo sinal obtido no domínio da frequência pela biblioteca numpy

Com os dados no domínio da frequência obtidos, é efetuada uma seleção de quais informações obtidas serão necessárias para a solução do afinador. Logo, foi implementado um algoritmo que se assemelha ao passa faixa, zerando quaisquer frequências irrelevantes que foram obtidas, deixando apenas disponível o intervalo suficiente para que cada corda do instrumento ukulele seja afinada de acordo com a afinação padrão do instrumento.

A figura que demonstra a filtragem efetuada pode ser visto a seguir:

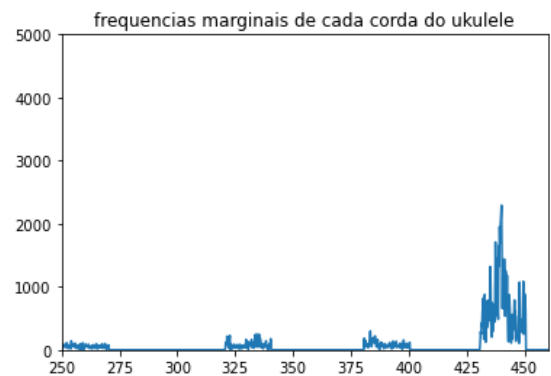


Fig. 4. Curva do sinal obtido no domínio da frequência, mantendo apenas frequências relevantes para a afinação do instrumento musical.

O algoritmo disponibiliza apenas as frequências de 250 Hz a 270 Hz (representam acorde em dó), 320 Hz a 340 Hz (representam acorde em Mi), 380 Hz a 400 Hz (representam acorde em Sol) e 430 Hz e 450 Hz (representam acorde em Lá).

Com a filtragem concluída, é efetuada a busca pela maior frequência captada pelo microfone, e então é comparada com a afinação padrão de cada corda e dessa forma, é obtida a corda que o músico tocou e quanto desafinada essa corda está. Por fim, é informado para o usuário a maior frequência captada, a nota musical representada por essa frequência e a frequência esperada para que o instrumento esteja afinado.

CONCLUSÕES

Pode-se concluir que o afinador funcionou bem para o instrumento ao qual foi projetado, indicando com precisão o quanto cada corda deveria ser afrouxada ou apertada para se obter a afinação adequada para o instrumento. O algoritmo FFT implementado não está tão perfeito quanto o disponível na biblioteca numpy, visto que existe uma pequena demora para transformar grandes quantidades de dados para o domínio da frequência, se comparado ao disponível na numpy. Mesmo tendo eficiência menor, o mesmo demonstrou total capacidade de afinar o instrumento para qual o trabalho se propôs.

REFERÊNCIAS

[1] Understanding the FFT Algorithm.

Disponível em:

<https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/>

[2] REDUCIBLE. The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever?

Disponível em:

https://www.youtube.com/watch?v=h7apO7q16V0&ab_channel=Reducible

[3] CADERNO DE LABORATÓRIO.

Acessando o microfone com python.

Disponível em:

<https://cadernodelaboratorio.com.br/acessando-o-microfone-com-o-python-biblioteca-pyaudio/>