

Equalizador de música

Relatório da disciplina de Processamento Digital de Sinais

Gabriel de Carvalho Ferreira – 17/0181111

Departamento de Ciência da Computação

Universidade de Brasília

Brasília, Brasil

gabrielcarvfer@gmail.com

I. INTRODUÇÃO

O aumento da capacidade de processamento digital aliado a transformada rápida de Fourier [2] permitiu diversos tipos de aplicações que antes eram consideradas impossíveis tanto por dificuldade de implementação em aplicações analógicas quanto por complexidade computacional em aplicações digitais.

Uma das principais aplicações do processamento digital de sinais é a filtragem destes, seja no domínio do tempo através da convolução com filtros ou no domínio da frequência, através do cálculo da transformada de Fourier, multiplicação pela transformada do filtro, e a transformada inversa de Fourier.

Uma das aplicações mais comuns de técnicas de filtragem de sinais, tanto analógica quanto digital, são os equalizadores, que selecionam bandas de frequência através de seus filtros, e permitem o ajuste de ganho em cada uma dessas faixas, permitindo que um usuário final controle parcialmente o sinal de saída conforme sua vontade.

Neste trabalho implementamos um equalizador com filtros digitais para processamento e execução em tempo real de blocos de amostras PCM importadas de arquivos “.wav”, além da exibição de maneira animada numa interface gráfica.

II. CONCEITOS

A. Convolução circular

A transformada inversa discreta de Fourier (IDFT) requer que o sinal seja periódico, e este pode ser obtido através da multiplicação da DFT de dois sinais, sendo um não periódico e outro periódico no tempo [1].

O resultado da IDFT é a convolução circular dos dois sinais de entrada, que difere da convolução linear, que é geralmente o objetivo no processamento digital de sinais [1].

B. Overlap-and-add

Uma das formas de se obter a convolução linear de dois sinais de tamanho arbitrário a partir de convoluções circulares de tamanho fixo é o método *overlap-and-add* [1].

O método consiste em calcular a convolução circular de um sinal de tamanho N com o outro sinal de tamanho K, geralmente um filtro, de maneira a obter o sinal convoluído circularmente de tamanho N+K-1.

Das N+K-1 amostras, onde de 0 a N-1 temos a primeira parte do sinal do bloco, e de N a N+K-1 temos a segunda parte do sinal do bloco. A segunda parte do sinal de um bloco é somado à primeira parte do sinal do próximo bloco, até o fim de todos os blocos, resultando na convolução linear [1].

C. Filtros

Filtros tem como objetivo separar diferentes partes de um dado sinal, de maneira a extrair informações do sinal ou permitir algum tipo de processamento naquela parte do sinal, para futura recomposição e exibição do sinal.

Tipos mais comuns de filtros são o passa baixas, passa faixa e passa altas, que fazem o que diz seus nomes, permitindo passagem de frequências abaixo de uma frequência de corte, ou uma faixa entre duas frequências de corte, ou a passagem de frequências acima de uma frequência de corte.

A resposta ao impulso de um passa baixas ideal é dado por

$$\frac{\omega_c}{\pi}, \text{ para } n = 0$$
$$\frac{1}{\pi \times n} \times \sin \omega_c n, \text{ para } n \neq 0$$

A resposta ao impulso do passa faixa ideal é dado por

$$\frac{\omega_{c2} - \omega_{c1}}{\pi}, \text{ para } n = 0$$
$$\frac{1}{\pi n} [\sin \omega_{c2} n - \sin \omega_{c1} n], \text{ para } n \neq 0$$

Já a resposta ao impulso do passa altas ideal é dado por

$$\frac{1 - \omega_c}{\pi}, \text{ para } n = 0$$
$$-\frac{1}{\pi \times n} \times \sin \omega_c n, \text{ para } n \neq 0$$

Filtros ideais requerem que o filtro tenha tamanho infinito, porém como isto não é factível, o resultado das equações desses filtros limitados na frequência se aproximam de uma função sinc, resultado do efeito de Gibbs [3]. Um dos meios de se atenuar o sinc e fazê-lo mais parecido com o filtro original, é usando janelas no filtro, de maneira a descartar parte do filtro e ficar com seu trecho mais relevante para a aplicação ao qual se destina [3].

Uma dessas janelas para filtros é o de Hann ou Hanning [2, 3], que tem uma razão grande entre o lobo principal do filtro e os lobos secundários, e dependendo do ajuste de um parâmetro

α , o lobo principal pode conter aproximadamente 99.96% da energia total da janela. A janela, dados M o número de amostras da janela, α o parâmetro de ajuste, e n o número da amostra, é dada pelas equações

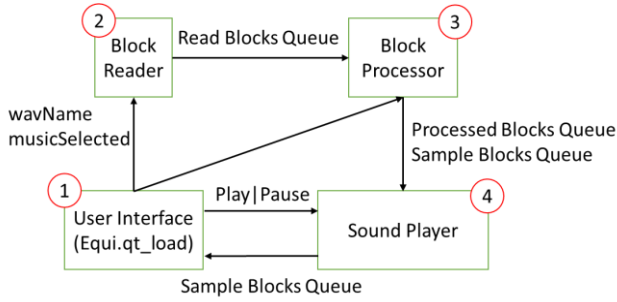
$$\alpha - (1 - \alpha) \cos \frac{2\pi n}{M}, \text{ para } |n| \leq \frac{M}{2}$$

$$0, \text{ para } |n| > \frac{M}{2}$$

III. IMPLEMENTAÇÃO

A implementação do equalizador foi estruturada em 4 processos, um responsável pela leitura de arquivos, outro pelo processamento de blocos, outro pela execução do áudio e gravação do arquivo de áudio processado, e outro responsável pela interface com o usuário. Cada uma das partes é detalhada a seguir.

Fig. 1. Arquitetura do equalizador



A. Interface com usuário

A interface com usuário possui diversos seletores de ganho de -12dB até +12dB, para cada uma das bandas determinadas pelas frequências centrais 32, 64, 125, 250, 500, 1000, 2000, 4000, 8000, 16000. Além dos seletores, existem barras que mostram a amostras de maior intensidade num dado bloco para cada uma das bandas filtradas.

Na parte de baixo da tela, existem três botões, o primeiro de executar música, segundo de parar a execução, e o último que permite selecionar o endereço do arquivo .wav a ser tocado.

Quando o arquivo “.wav” é selecionado, o processo de interface do usuário altera valores do nome de arquivo “.wav” e um booleano, que indica se o arquivo foi selecionado, num espaço de memória compartilhada entre os processos, permitindo que o processo leitor de blocos (bloco 2 na Fig. 1) inicie a leitura dos blocos.

O processo de interface com usuário também recebe do tocador de música (bloco 4 na Fig. 1), via troca de mensagens em *SampleBlocksQueue*, as amostras de intensidade que deve mostrar nas barras.

B. Leitor de blocos

O processo leitor de blocos (bloco 2 na Fig. 1) abre o arquivo de entrada “wavName”, lê as amostras de Modulação por

Código de Pulso (PCM) e despacha estas amostras em blocos através de troca de mensagens na fila *ReadBlocksQueue* para o processador de blocos (bloco 3 na Fig. 1), que fará o processamento do sinal.

C. Processador de blocos

O processador de blocos (bloco 3 na Fig. 1) é responsável pelo processamento do sinal recebido do processo leitor de blocos e pelo envio dos blocos processados para o tocador de música.

O processamento é feito sobre cada bloco de cada canal (tipicamente esquerdo e direito), sobre cada uma das bandas, convoluindo circularmente o sinal de entrada somado do resto da convolução circular anterior, com o filtro adequado a cada banda, que pode ser passa baixa, para a primeira banda (banda 0), passa faixas, para bandas intermediárias (bandas 1 até L-2), e passa alta, para a última banda (banda L-1), para L bandas filtradas.

O filtro aplicado as bandas são filtros ideais passa baixa, passa faixa e passa alta, multiplicados pelo filtro janela Hamming. A largura do filtro usada foi de 101 amostras.

Para o passa baixas, a banda passante é de 0Hz até 140% a frequência de corte do filtro (nesta implementação, a frequência de corte do passa baixas é de 32Hz).

Para os passa bandas, as bandas passantes dos filtros são 70% da frequência central até 140% da frequência central da banda.

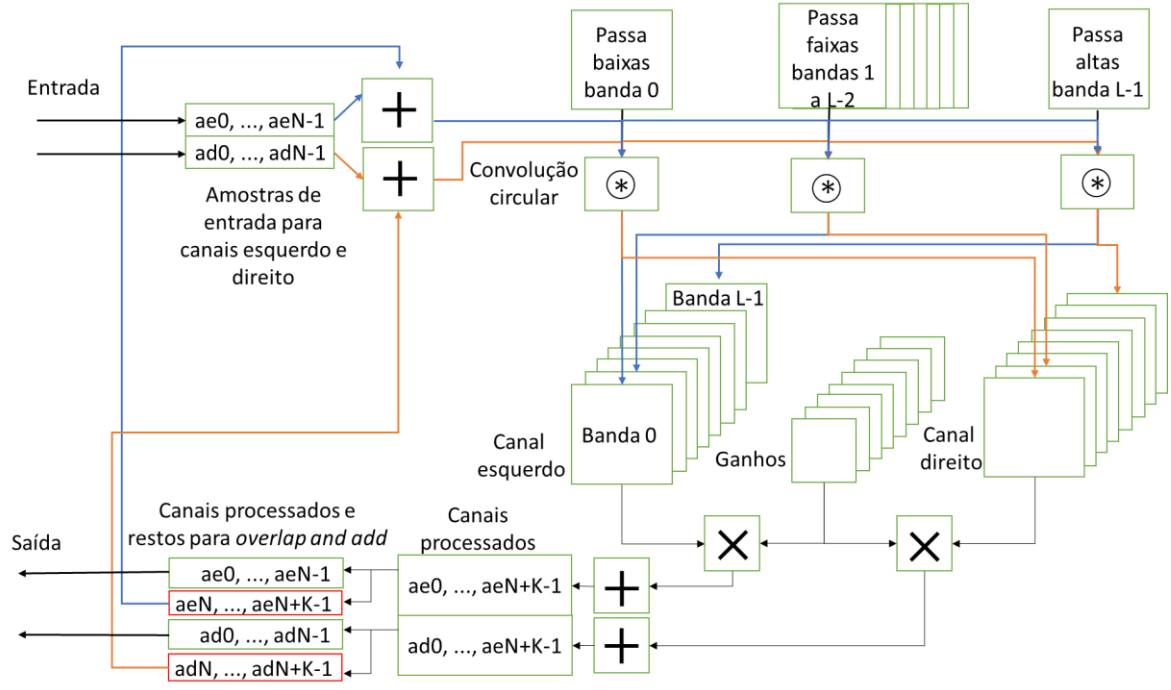
Após cada banda filtrada, o sinal filtrado de cada uma das bandas é multiplicado pelo ganho inserido pelo usuário, e os sinais com ganho ajustados são somados no sinal de saída.

O sinal de saída tem comprimento maior que o de entrada, em razão da convolução entre um sinal de tamanho N e um filtro de tamanho K , que resultam num sinal de tamanho $N+K-1$. Este sinal resultante é separado em duas partes: das amostras 0 a $N-1$, temos o sinal processado de saída, e das amostras N a $N+K-1$ temos o resto que será somado no próximo sinal de entrada, fazendo com que a convolução circular resulte na convolução linear via *overlap-and-add*.

O funcionamento descrito anteriormente do processador de blocos é mostrado na Fig. 2.

Além disto, o processador de blocos, após cada aplicação do ganho em cada uma das bandas de cada canal, busca a amostra de maior intensidade, que é enviada para a interface gráfica do equalizador para exibição na forma de barras. Antes de iniciar o processamento do bloco seguinte, o ainda é feito um ajuste *anti-clipping*, que verifica se alguma das amostras em um dos canais é maior que o limite do PCM (por exemplo, inteiro de 16 bits), e caso seja, abaixa o valor dessa amostra para o valor máximo, e faz o ajuste em todas as outras amostras para que uma frequência não seja prejudicada e o efeito da alteração dos ganhos fique mais visível mesmo com ganhos altos.

Fig. 2. Processamento de sinais por blocos



D. Tocador de música

O tocador de música recebe um bloco processado pelo processador de blocos (bloco 3 da Fig.2) por troca de mensagens via fila *ProcessedBlocksQueue*, e despacha o bloco para que o sistema operacional toque o áudio. Também recebe as maiores amostras de cada uma das bandas filtradas e as envia para a interface com usuário (bloco 1 da Fig. 2), ambas via troca de mensagens.

Além disso, o tocador de música guarda as amostras recebidas, e ao final da execução da música guarda o áudio processado num arquivo “.wav” de saída.

IV. RESULTADOS

As respostas em frequência obtidas para os filtros passam baixa e passa alta, além de um dos filtros passa baixas é colocado a seguir, para efeito de demonstração.

Fig. 3. Resposta em frequência do filtro passa-baixas, com frequência de corte em 140% da frequência central 32Hz, ou 44.8Hz.

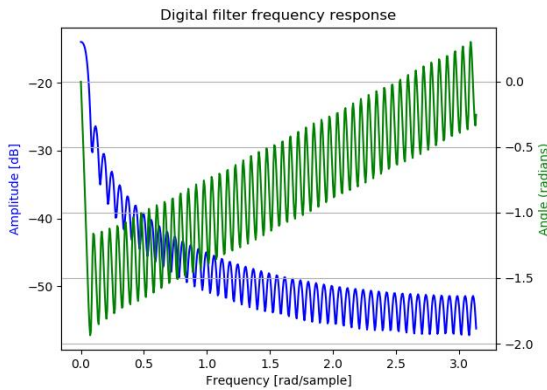


Fig. 4. Resposta em frequência do filtro passa-faixa, com frequência central de 4kHz, ou frequências de corte 2.8kHz e 5.6kHz

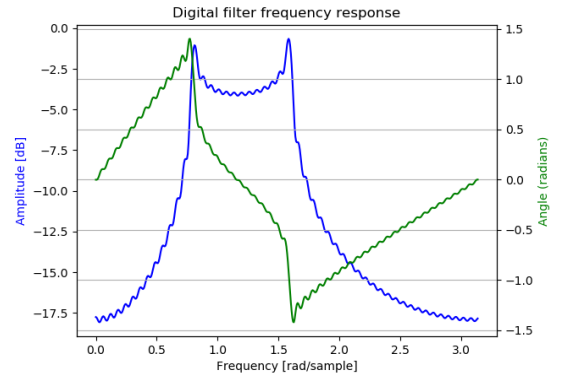
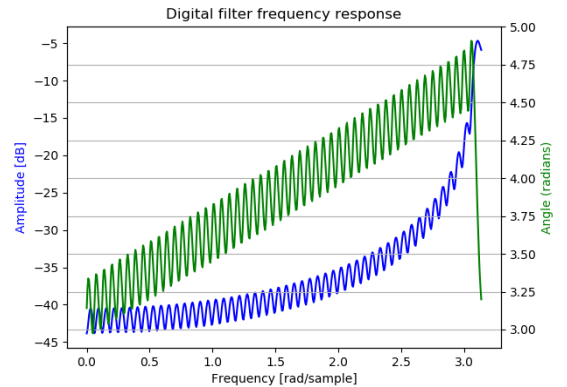


Fig. 5. Resposta em frequência do filtro passa-alta, com frequência central de 16kHz, ou frequência de corte 11kHz.



A largura dos filtros, vulgo a determinação da(s) frequência(s) de corte deles, baseados na frequência central do filtro foi feita experimentalmente, alterando os valores de 50%~90% para frequências abaixo da central, e 120% até 150% para frequências acima da central, até se encontrar o valor de 70%~140% da frequência central, cujo o sinal de saída teve a menor diferença quando comparado com o sinal de entrada (-45dB quando o sinal de saída era invertido e somado ao sinal de entrada).

REFERÊNCIAS

- [1] P. Diniz, E. Silva and S. Netto, "Digital Signal Processing: System Analysis and Design", Cambridge University Press, 2010.
- [2] J. Cooley and J. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, Mathematics of Computation, Vol. 19, Issue 90, 1965, pp.297-301.
- [3] A. Oppenheim and R. Schafer, "Discrete-Time Signal Processing", 3rd edition, Pearson Education Limited, 2014.