

# FILTRO EQUALIZADOR GRÁFICO PERSONALIZADO

Carlos Rafael Gimenes das Neves

Faculdade de Tecnologia  
Bandeirantes

## RESUMO

O uso de filtros equalizadores é muito comum em aplicações profissionais de áudio, por uma série de razões, como por exemplo, para compensar pequenas discrepâncias entre diferentes sistemas de reprodução, com diversas respostas em frequência. Filtros equalizadores também são encontrados em aplicações domésticas para permitir que os usuários alterem a resposta original do áudio a seu gosto, dando-lhe mais brilho ou mais graves. Grande parte dos equalizadores encontrados nessas aplicações são equalizadores paramétricos, que permitem ao usuário controlar a resposta de algumas bandas de frequências pré-determinadas. Em alguns poucos casos, é possível especificar o fator Q para controlar o ganho entre as bandas. O presente trabalho propõe uma técnica para criar filtros equalizadores personalizados, onde o usuário pode literalmente desenhar a resposta em frequência desejada, em vez de controlar o ganho apenas de algumas frequências. A técnica utiliza a transformada rápida de Fourier para permitir que tanto a criação como a aplicação do filtro sejam realizadas de forma rápida, em tempo real.

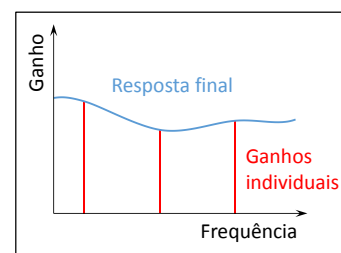
## 1. INTRODUÇÃO

O uso de filtros equalizadores é uma prática muito comum para aplicar correções e realizar ajustes a um áudio durante sua reprodução. Esses filtros são utilizados tanto em ambientes profissionais como em ambientes domésticos.

Em ambientes profissionais os equalizadores são utilizados para tentar se obter uma reprodução da maneira mais fidedigna possível. Para isso, deve-se alterar o ganho de determinadas frequências, compensando pela resposta em frequência do meio por onde o áudio é reproduzido. Por exemplo, caixas acústicas de diferentes materiais e de diferentes tecnologias costumam apresentar diferentes respostas em frequência, por vezes atenuando algumas regiões do espectro de frequências, enquanto favorecendo outras regiões.

Já em ambientes domésticos, os equalizadores normalmente são utilizados para se obter uma experiência sonora mais prazerosa, acrescentando ou removendo brilho do áudio, atenuando a região do espectro que compreende a voz humana, favorecendo os instrumentos, acrescentando mais grave, dentre outros inúmeros cenários.

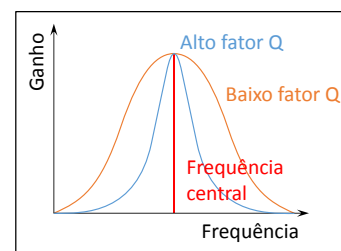
Grande parte dos filtros equalizadores disponíveis em aplicações de uso doméstico são equalizadores paramétricos, onde o usuário controla o ganho individual de um conjunto pré-determinado de frequências. A resposta real obtida fica a cargo da aplicação, sendo que por vezes, é desconhecida pelo usuário. A Figura 1 ilustra esse cenário.



**Figura 1.** Exemplo de resposta em frequência obtida por filtros paramétricos.

Na Figura 1, as linhas vermelhas representam o ganho desejado, fornecido pelo usuário, para uma dada frequência e a curva azul representa a resposta final obtida. Nesses casos, o usuário não é capaz de especificar o comportamento da resposta final entre as frequências para as quais ele definiu o ganho. Além disso, essa curva nem sempre é exibida para o usuário.

Em algumas aplicações, é possível definir o fator Q do filtro, seu fator de qualidade [1], de modo a ajustar a largura de banda afetada pelo controle, conforme mostrado na Figura 2. Nela, a curva azul representa o ganho real aplicado através de um filtro com fator Q alto, e a curva laranja representa o ganho real produzido por um fator Q baixo. Quanto maior o fator Q, menor é a largura de banda ao redor da frequência central, e quanto menor o fator Q, maior é a largura de banda.



**Figura 2.** Exemplo de resposta com diferentes fatores Q.

Mesmo especificando o fator  $Q$ , ainda não é possível personalizar efetivamente a resposta em frequência de maneira arbitrária, para um conjunto maior de frequências. Por exemplo, supondo uma frequência central de 1000 Hz, com um fator  $Q$  qualquer, o ganho será aplicado de modo equivalente para frequências entre menores que 1000 Hz, como para frequências maiores que 1000 Hz.

Contudo, o usuário poderia desejar aplicar ganhos distintos para cada uma das metades ao redor da frequência central. Ou ainda, o usuário poderia desejar aplicar um ganho com curvas diferente das obtidas automaticamente, sem que cada parte curva fosse regida pela mesma equação, apenas com coeficientes trocados.

Para esse tipo de cenário, o trabalho propõe o uso de um filtro equalizador gráfico criado a partir de uma curva inteiramente desenhada pelo usuário. Através da curva gera-se um filtro FIR [2], que depois é aplicado ao sinal de áudio através do processo de convolução [2]. Para acelerar a aplicação do filtro, o sinal do áudio é transportado para o domínio da frequência, através da transformada rápida de Fourier para dados reais, FFT Real [3], de modo que a convolução possa ser realizada como uma sequência linear de multiplicações.

## 2. CRIAÇÃO DO FILTRO EQUALIZADOR

O processo de criação do equalizador personalizado inicia com a curva de resposta em frequência desejada, fornecida pelo próprio usuário. Cada ponto da curva representa o ganho desejado, em dB, para a frequência daquele ponto.

Como a criação do filtro envolverá o uso de uma FFT, as frequências disponíveis para configuração dependem da taxa de amostragem do áudio e do comprimento da FFT utilizada. Por exemplo, para uma FFT com 2048 pontos, e uma taxa de amostragem de 44100 Hz, cada ponto do filtro corresponderia a  $44100 \text{ Hz} / 2048 = 21,5332 \text{ Hz}$ . Com uma taxa de amostragem de 48000 Hz, cada ponto passaria a corresponder a  $23,4375 \text{ Hz}$  [2].

Para evitar essas diferenças, e prover uma experiência uniforme ao usuário, a curva editada pelo usuário trabalha com um conjunto pré-estabelecido de frequências. Durante a criação do filtro, os ganhos em dB de cada frequência da curva são mapeados linearmente para as frequências do filtro. Caso um ponto do filtro englobe mais de um ponto da curva, realiza-se uma média aritmética entre os ganhos da curva. No caso oposto, realiza-se uma interpolação linear com dois pontos da curva para preencher vários pontos do filtro. Quanto maior a quantidade de pontos da FFT, maior será a fidelidade à curva original.

Depois de determinado o ganho em dB de uma determinada frequência, ele é convertido para um valor de magnitude, utilizando a Equação 1 [1].

$$\text{Magnitude} = 10^{dB/20} \quad (1)$$

Os valores da magnitude de cada ponto devem ser completados com os respectivos valores da fase de cada ponto, de modo a obter um número complexo na notação polar, como mostra a Equação 2.

$$\text{Polar: Magnitude } e^{-k \cdot j} \quad (2)$$

Já que o usuário não fornece os valores da fase de cada ponto, assume-se uma fase linear,  $k$ , como mostram as Equações 3 e 4. A constante  $L$  representa o comprimento da FFT utilizada, como 2048 ou 1024.

A variável  $i$  representa o índice do ponto atual do filtro, variando de 0 à  $M$ . A constante  $M$ , por sua vez, representa o índice do último ponto do filtro, que contém  $M + 1$  amostras.

Isso ocorre porque uma FFT que trabalha com  $L$  amostras reais no domínio do tempo, precisa só de  $(L / 2) + 1$  amostras complexas no domínio da frequência [2]. Dentre essas amostras, duas são sempre números reais: a amostra de índice 0, que é a componente DC, e a amostra de índice  $(L / 2)$  que é a componente da frequência de Nyquist [2, 3].

$$k = \frac{i \pi (M/2)}{(L/2)} = \frac{i \pi}{2} \quad (3)$$

$$M = (L/2) \quad (4)$$

O próximo passo na criação do filtro é aplicar a FFT inversa sobre os valores complexos, obtendo a resposta do filtro no domínio do tempo. Para isso, os pontos devem ser convertidos da notação polar para a notação retangular, onde cada ponto passa a possuir uma componente real e outra componente imaginária, e não mais magnitude e fase, como mostrado nas Equações 5 e 6 [1].

$$\text{Real} = \text{Magnitude} \cos(-k) \quad (5)$$

$$\text{Imaginária} = \text{Magnitude} \sin(-k) \quad (6)$$

Já no domínio do tempo, aplica-se uma janela ao filtro de modo a melhorar sua resposta nesse domínio [2]. Foram testadas, empiricamente, as janelas de Hanning, Hamming e Blackman, sendo que a janela de Blackman, mostrada na Equação 7, obteve mais sucesso. Na Equação 7,  $W(i)$  é a função janela, e a variável  $i$  representa o índice da amostra no domínio do tempo, variando de 0 à  $M$ , já definido na Equação 4.

$$W(i) = \frac{21}{50} + \frac{1}{2} \cos\left(\frac{2 i \pi}{M}\right) + \frac{2}{25} \cos\left(\frac{4 i \pi}{M}\right) \quad (7)$$

Depois de aplicar a janela aos  $(M + 1)$  primeiros pontos, ou amostras, do filtro no domínio do tempo, as amostras restantes, cujos índices variam de  $(M + 1)$  à  $(L - 1)$ , são

zeradas. Em seguida, aplica-se a FFT sobre os dados no domínio do tempo, obtendo novamente o filtro no domínio da frequência.

Todo esse processo de converter o filtro para o domínio do tempo, aplicar uma janela e então retornar o filtro para o domínio da frequência é necessário para melhorar os resultados da aplicação do filtro ao áudio [2], que é um sinal originalmente obtido no domínio do tempo.

Apesar de necessário, esse processo introduz alterações à curva da resposta em frequência original, suavizando seu formato. Contudo, a curva real da resposta em frequência pode ser facilmente obtida, e exibida ao usuário.

Para isso, basta obter as magnitudes a partir dos pontos da versão final do filtro no domínio da frequência. Como cada ponto do filtro é na verdade um número complexo na notação retangular, com exceção de DC e de Nyquist que são números reais, a magnitude pode ser obtida com a Equação 8, onde  $R$  é a componente real do ponto, e  $I$  é a componente imaginária do ponto. Por fim, antes de ser exibido para o usuário, o valor de magnitude deve ser convertido para um valor em dB, utilizando o inverso da Equação 1.

$$Magnitude = \sqrt{R^2 + I^2} \quad (8)$$

### 3. APLICAÇÃO DO FILTRO EQUALIZADOR

Com o filtro criado, e com seus pontos no domínio da frequência, é possível realizar a aplicação do filtro através de uma sequência linear de multiplicações, uma vez que a convolução no domínio do tempo é representada por multiplicações no domínio da frequência [2].

Para a aplicação do filtro,  $(L / 2)$  amostras do áudio são processadas por vez, uma vez que, apesar de estarem sendo realizadas transformações de Fourier com  $L$  amostras, as últimas  $(L / 2)$  amostras do sinal precisam ser zeradas, de modo a se aplicar a técnica de *Overlap-Add* [2], utilizada na aplicação de filtros a sinais cujo comprimento excede o comprimento do filtro.

O algoritmo completo de aplicação do filtro é descrito nos oito passos a seguir:

1. Preenche-se um vetor, *data*, com  $(L / 2)$  amostras de áudio, seguidas por  $(L / 2)$  zeros.
2. Aplica-se uma FFT de comprimento  $L$  em *data*, o que resulta em  $(L / 2) + 1$  amostras no domínio da frequência, *dataF*.
3. As amostras em *dataF*, exceto DC e Nyquist [2], são multiplicadas pelos valores do filtro, utilizando uma multiplicação comum de números complexos na forma retangular.
4. As amostras DC e Nyquist [2] de *dataF* são multiplicadas através de uma multiplicação simples, uma vez que elas são números reais, sem a componente imaginária.

5. Aplica-se a FFT inversa sobre *dataF*, obtendo  $L$  amostras filtradas, já no domínio do tempo, *dataT*.
6. Soma-se as amostras em *lastData* às primeiras  $(L / 2)$  amostras de *dataT*. O vetor *lastData* é um vetor de comprimento  $(L / 2)$ , que deve ser completamente preenchido com zeros antes de se aplicar o filtro ao primeiro subconjunto de amostras.
7. Copia-se as últimas  $(L / 2)$  amostras de *dataT* para *lastData*. As primeiras  $(L / 2)$  amostras de *dataT* já podem ser utilizadas, e representam as  $(L / 2)$  amostras do passo 1, só que filtradas.
8. Repete-se o processo inteiro, desde o passo 1, para as próximas  $(L / 2)$  amostras de áudio.

Todos os passos do algoritmo devem ser repetidos para cada canal do áudio, individualmente.

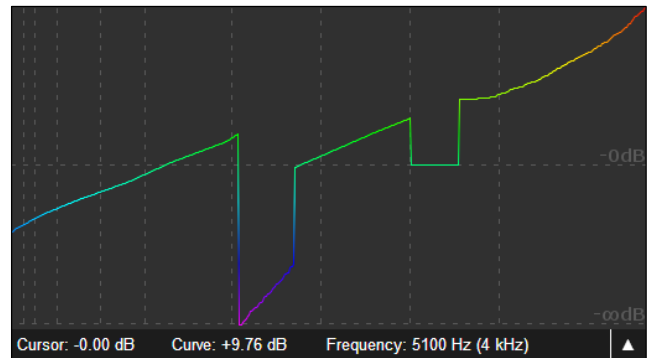
Caso não seja necessário preservar as amostras originais do áudio, é possível economizar memória criando o vetor *data* com comprimento suficientemente grande para armazenar  $(L / 2) + 1$  amostras complexas, e então utilizar *data* no lugar de *dataF* e de *dataT*.

### 4. DETALHES DA IMPLEMENTAÇÃO

Toda a ideia apresentada no trabalho foi originalmente implementada em um software em linguagem C++, para plataforma Windows. Contudo, decidiu-se levar a implementação para o ambiente Web, para que funcionasse independentemente da plataforma.

A implementação da aplicação para Web utilizou o elemento *canvas* do HTML5, para permitir que o usuário desenhasse a resposta desejada, como mostra a Figura 3.

Além de poder desenhar a resposta desejada, o usuário pode também visualizar a curva da resposta real, pode configurar os canais direito e esquerdo do áudio separadamente e pode optar por alterar os ganhos em grupo. Ele tem também a opção de normalizar a curva para que o ganho real nunca ultrapasse 0 dB.

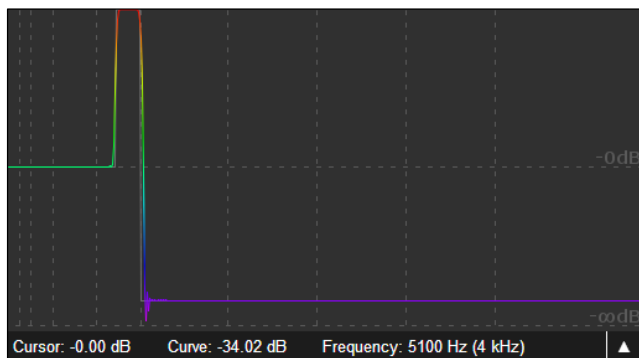


**Figura 3.** Aparência do equalizador gráfico implementado com JavaScript e HTML5.

A implementação permite que o usuário trabalhe com 512 frequências, em uma distribuição quase logarítmica, e inicialmente faz uso de um filtro com 2048 pontos. Apesar

desse valor, é oferecida ao usuário uma opção para ajustar o tamanho do filtro de 64 até 8192 pontos. Essa opção, aliada à opção de exibir a resposta real, permite mostrar o impacto da quantidade de pontos sobre o formato da curva final, além dos efeitos causados pelo janelamento, como o aparecimento da ondulação (ripple) quando há alterações muito acentuadas no ganho entre frequências próximas.

Por exemplo, a Figura 4 mostra a ondulação que surge quando se altera o ganho de 40 dB, em 710 Hz, para -34 dB, em 720 Hz.



**Figura 4.** Ondulação da resposta em frequência devido à alterações bruscas de ganho.

Para efetivamente aplicar o filtro ao áudio, foi utilizada a, ainda experimental, Web Audio API [4]. Com ela é possível carregar e reproduzir sons direto no browser, via JavaScript. Ela é baseada em nós, que são interligados, formando um caminho por onde trafega o áudio.

O caminho é iniciado em nós do tipo *AudioSourceNode*, responsáveis por prover o áudio, e termina em nós do tipo *AudioDestinationNode*, responsáveis por reproduzi-lo.

É possível acrescentar nós para alterar o áudio ao longo do caminho. Na implementação foram utilizados dois nós do tipo *ConvolverNode*, um para o canal direito e um para o canal esquerdo. Os nós do tipo *ConvolverNode* realizam a convolução do áudio com uma dada resposta ao impulso no domínio do tempo. No caso, essa resposta é o filtro criado no domínio do tempo, antes da aplicação da janela.

Não é necessário aplicar a janela ao filtro pois a aplicação inteira do filtro fica a cargo do browser, que internamente implementa o código de *ConvolverNode*. Por esse mesmo motivo, não é necessário implementar o algoritmo de aplicação do filtro, descrito anteriormente no trabalho.

O código-fonte da implementação é aberto, e pode ser encontrado no GitHub, no endereço:

<https://github.com/carlosrafaelgn/GraphicalFilterEditor/>

A implementação pode ser experimentada em qualquer browser que suporte a Web Audio API (até o dia da escrita do trabalho, apenas o Google Chrome suportava a Web Audio API), no endereço:

<http://carlosrafaelgn.com.br/GraphicalFilterEditor/>

## 5. TRABALHOS FUTUROS

Apesar da implementação funcionar corretamente, e filtrar o áudio da maneira esperada, há ainda uma série de melhorias que podem ser realizadas, tanto no algoritmo como na implementação da aplicação Web.

Poderia ser testada a aplicação de outras janelas ao filtro no domínio do tempo, além das três já testadas no trabalho, procurando por janelas que ofereçam melhor resposta em frequência, ou ainda, realizando uma análise prévia sobre a curva, para selecionar em tempo real a melhor janela para cada tipo de curva.

Além disso, é possível realizar um estudo mais profundo sobre qual fase utilizar em conjunto com a magnitude fornecida pelo usuário, substituindo  $k$  da Equação 3.

Com relação à implementação da interface com o usuário, poderiam ser criados meios para que ele pudesse desenhar linhas retas, ou outras formas geométricas e curvas, de maneira automatizada. Também poderiam ser criados meios para que o usuário pudesse salvar as configurações, para depois recuperá-las.

## 6. CONCLUSÃO

Em ambientes domésticos, os filtros equalizadores são muito utilizados por aqueles que gostam de realmente experimentar o som, com todas as sensações que ele pode proporcionar.

Este trabalho propõe uma técnica para criar filtros equalizadores, com a esperança de atender a alguns dos desejos desse tipo de usuário, que nem sempre se satisfaz com filtros equalizadores convencionais.

A aplicação Web se mostrou funcional, e a Web Audio API mostrou ser bastante promissora. Contudo, como esta API ainda está sendo desenvolvida e não foi implementada por grande parte dos browsers, não é possível utilizar a aplicação Web em todas as plataformas, como se desejava inicialmente.

## 7. REFERÊNCIAS

- [1] Irwin, J. D. *Análise de Circuitos em Engenharia 4ª Edição*. São Paulo: Makron Books, 2000.
- [2] Smith, S. W. *The Scientist & Engineer's Guide to Digital Signal Processing*. California: California Technical Publishing, 1997.
- [3] Press, W. H. et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 2007.
- [4] Rogers, C. *Web Audio API: W3C Working Draft*. W3C, 2012.