Marco Aurélio Gonçalves da Silva

Filtros Digitais Aplicados em Sinais de Áudio

Orientador:
Augusto Santiago Cerqueira

Co-orientador:
Marcelo Bernardes Vieira

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Juiz de Fora

Monografia submetida ao corpo docente do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como parte integrante dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação

Prof. Augusto Santiago Cerqueira, D. Sc.
Orientador

Prof. Marcelo Bernardes Vieira, D. Sc.
Co-Orientador

Prof. Rubens de Oliveira, D. Sc.

Prof. Marcelo Lobosco, D. Sc.

Sumário

Resumo

Este trabalho trata sobre Filtros Digitais, assunto da área de Processamento Digital de Sinais, e como podemos utilizá-los para a construção de um Equalizador de áudio. Para o processo de projeto de filtros, uma discussão sobre Funções Janela é introduzida para proporcionar melhorias em relação à ação dos filtros.

Uma implementação de um equalizador é realilzada para ilustrar os resultados. Com um equalizador, podemos claramente perceber as alterações realizados no domínio da freqüência de um sinal de áudio digital.

1 Introdução

A percepção auditiva sempre foi muito importante para o ser humano em vários aspectos e há muito tempo a música tem grande importância em todo o mundo. A música é responsável por proporcionar boas sensações ao ouvinte graças ao seu ritmo e harmonia.

Uma das formas mais comuns de processamento do sinal de áudio consiste em sua filtragem em relação as freqüências do sinal. Muitas vezes a filtragem tem por objetivo tornar o sinal de áudio (uma música, por exemplo) mais agradável ao ouvinte. A esse processo, muitas vezes chamamos de equalização.

1.1 Equalização de Sinais de Áudio

O termo equalização diz respeito ao ajuste dos graves, médios e agudos no contexto do espectro de freqüências de um determinado sinal de áudio. Existem várias maneiras de realizar uma equalização de um sinal de áudio, tanto para um sinal de áudio analógico, quanto para um sinal de áudio digital. Uma equalização pode ser feita, por exemplo, por uma mesa de som, pelo controle de equipamentos como equalizadores gráficos ou paramétricos que ajustam as faixas de freqüências do sinal através de ganhos que devem ser dados a diferentes bandas, ou seja, as diferentes faixas de freqüência.

Um fator importante que deve ser levado em consideração é a resposta em freqüência de sistemas de captura e sistemas de reprodução. Mesmo uma música ou outro sinal de áudio qualquer bem gravado e devidamente equalizador durante o processo de gravação pode ser mal executado por um sistema de reprodução. Assim o sinal gerado terá características diferentes no domínio da freqüência, quando comparado com o sinal de áudio original executado em um sistema de reprodução com resposta em freqüência ideal, ou seja, sem acrescentar ou retirar energia de alguma faixa de freqüência. Um exemplo de sistema de reprodução é uma caixa amplificadora de um aparelho de som [Ser02].

Sistemas de captura com uma resposta em freqüência diferente da ideal também podem distorcer o sinal de áudio que recebem e, assim, gerar uma gravação com um espectro de freqüências diferente do sinal original emitido. O exemplo clássico de um sistema de captura é o microfone.

A equalização também é importante em shows musicais e varia em função do local em que eles são realizados. Nesses eventos, normalmente existe um controlador de mesa de som que trata da equalização do som. Em locais abertos e amplos, esse controlador deve retirar mais energia das altas freqüências do que em ambientes fechados e pequenos. Se não houvesse um sistema de equalização seria insuportável o nível das freqüências médias e altas, pois essas são mais absorvidas pelo ar do que as baixas, portanto as pessoas próximas ao palco teriam uma sensação muito desagradável em relação a música, enquanto que as mais distantes escutariam o som melhor equalizado.

Então, podemos ver que a equalização de um sinal de áudio pode ser necessária por vários motivos, pois mesmo um sinal que passa por um sistema de captura e um sistema de reprodução, ambos ideais, pode ser equalizado por simples questão de gosto do ouvinte.

1.2 Sinais

Sinais possuem importantes papéis em nosso cotidiano. Alguns exemplos de sinais são a fala, a música, imagens e vídeos. Todo sinal é uma função de uma variável independente como tempo, espaço, temperatura, pressão. A fala, por exemplo, é a pressão do ar em função do tempo em determinado ponto do espaço. De forma primitiva, os sinais são originalmente emitidos da natureza, porém atualmente existem vários meios de produzir sinais sinteticamente.

O som é uma onda, uma perturbação da matéria que transporta energia de um lugar para outro. Quando um objeto vibra, ele empurra as moléculas de ar e cria uma compressão, e durante o intervalo de tempo em que o objeto volta, ele cria rarefações no ar. Esses eventos ocorrendo repetidamente caracterizam uma onda longitudinal. Uma onda longitudinal se caracteriza por seus pontos se moverem em direção paralela à própria onda [Pie89].

A vibração do ar causa pequenas e rápidas vibrações na pressão do interior do ouvido humano, estimulando certos nervos a transmitirem impulsos elétricos que serão interpretados pelo cérebro como o som da maneira que percebemos.

Os sons naturais são, na sua maior parte, combinações de sinais, mas um som puro (senóide) possui uma única velocidade de oscilação ou freqüência que se mede em hertz (Hz) e uma amplitude ou energia que se mede em decibéis. Os sons audíveis pelo ouvido humano têm uma freqüência entre 20 Hz e 20 kHz [Ang01].

Por questões de simplicidade, usamos ondas transversais para ilustrar o som. Um microfone, por exemplo, possui um transdutor, que é a parte do equipamento que converte os sinais captados na membrana em energia elétrica. A variação da pressão do ar faz com que a membrana do microfone vibre e assim ele pode converter essa variação em sinal elétrico. A geração do som nada mais é que o processo inverso do microfone. Um amplificador gera pressões no ar através da vibração de seu alto-falante.

Podemos dizer que um som que escutamos é mais complexo que outro pelas suas características como timbre e faixa de freqüências existente. Uma flauta, por exemplo, produz um som muito simples, porém um saxofone possui um timbre bem mais complexo e interessante. Entende-se por timbre a capacidade de distinguir dois sons na mesma nota e mesmo volume. Quando escutamos uma música, por exemplo, recebemos uma combinação de sons de diferentes instrumentos e conseqüentemente diferentes timbres e diferentes faixas de freqüência.

Quando falamos ou tocamos algum instrumento musical, não emitimos apenas uma freqüência específica, mas sim uma faixa de freqüência. Este fenômeno acontece por causa da existência dos harmônicos [Ang01]. A corda de um violão, por exemplo, quando vibra, não emite apenas a freqüência do modo fundamental de uma nota, ela também vibrará em freqüências múltiplas inteiras dessa freqüência que são chamadas de modos harmônicos.

Esta característica do som possui grande importância na teoria de escalas musicais e na formação de timbre dos instrumentos musicais. Muitas vezes torna-se interessante para os músicos a filtragem de determinadas freqüências que seus instrumentos possam emitir, querendo desse modo diminuir a faixa de ocorrência de harmônicos de uma determinada nota ou conjunto de notas.

1.3 Proposta de Trabalho

O campo do Processamento de Sinais Digitais apresentou enorme crescimento durante as últimas décadas. A vasta gama de aplicações e a evolução tecnológica foram fatores decisivos para tal acontecimento.

Há alguns anos atrás o áudio era processado apenas por meios analógicos, assim era necessário a construção de circuitos eletrônicos para realizar tal operação. Hoje o áudio é mais comumente encontrado armazenado em meios digitais, desta maneira é natural que o processamento de sinais digitais tenha tornado uma área de grande importância em relação ao processamento de áudio. O meio digital possui as vantagens de compactabilidade física, fácil edição de dados e cópia destes dados sem perdas, o que possibilita que inúmeras cópias de um arquivo de áudio sejam sempre idênticas, o que não ocorre no meio analógico.

Esse trabalho tem como objetivo promover um estudo sobre processamento de sinais digitais, mais especificamente sobre filtros digitais e como projetá-los. Uma vez com um filtro projetado, podemos juntar esse e mais outros filtros e construir um equalizador para sinais de áudio, que é o objetivo da aplicação realizada nesse trabalho.

1.4 Visita Guiada

O Capítulo 2 deste trabalho apresenta a matemática necessária para o entendimento do objetivo do trabalho, que é criar filtros digitais. Para isso o assunto é dividido inicialmente em relação à representação do sinal.

O modelo computacional, visto no Capítulo 3, indica os métodos utilizados para uma solução computacional do problema e como eles foram distribuídos no desenvolvimento.

No Capítulo 4, os resultados da aplicação podem ser observados, ilustrando o conhecimento aplicado e fazendo um pequeno comparativo dos diferentes filtros usados.

Finalmente, no Capítulo 5, algumas conlusões sobre o trabalho são apresentadas juntamente com as possibilidades de trabalhos futuros.

2 Modelo Matemático

Neste capítulo, os conceitos matemáticos importantes na área de processamento de sinais são introduzidos. Será comentado como são feitas as representações do sinal de áudio no espaço contínuo e no espaço discreto.

2.1 Representações do Sinal de Áudio Contínuo

Um sinal de áudio emitido pela natureza é um sinal contínuo, ou seja, ele está definido para qualquer instante do tempo. Existem várias formas de representar um sinal contínuo, porém apenas duas são importantes para a formulação deste trabalho, que é a representação do sinal no domínio do tempo e a representação do sinal no domínio da freqüência.

Matematicamente sinais de áudio no domínio do tempo são definido por $x(t)$, onde t representa o tempo e $x(t)$ representa a amplitude no instante de tempo t. Podemos representar $x(t)$ por um valor que pode ser real ou complexo. Um sinal de áudio pode ser decomposto por várias senóides. Senóides são sinais periódicos que possuem fase, amplitude e freqüências determinadas. Na Figura 1, podemos ver graficamente como o sinal pode ser representado no domínio do tempo.

Figura 1: Domínio do tempo

Normalmente quando pensamos em sinais de áudio, agregamos estes à sua representação no domínio do tempo. Porém podemos representar sinais no domínio da freqüência. Um sinal contínuo no domínio da freqüência é definido por $X(\omega)$, onde $\omega = 2\pi f$ representa a freqüência angular em radianos por segundo e $X(\omega)$ representa a amplitude do sinal na freqüência ω [Lat04]. Na Figura 2, podemos ver graficamente como o sinal pode ser representado no domínio da freqüência.

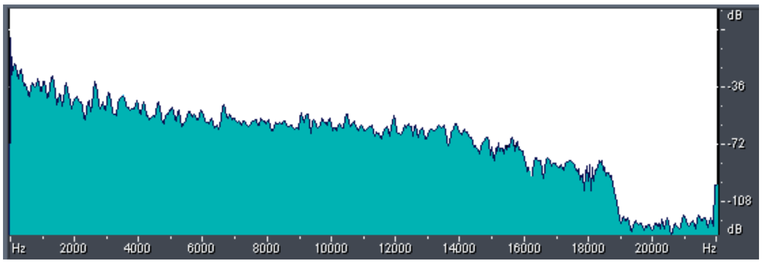

Figura 2: Domínio da freqüência

Nesta seção, vamos estudar noções para representação do sinal de áudio contínuo no domínio do tempo e no domínio da freqüência e como transformar a representação de um sinal em um domínio para o outro.

2.1.1 Senóides Reais

Uma senóide real é uma função que descreve uma onda pela variação do tempo e que pode ser expressa pela forma

$$x(t) = A \text{ sen}\,(\omega t + \phi) \tag{2.1}$$

onde t é a variável independente de valor real e A, ω, e ϕ são constantes reais e correspondem a amplitude, freqüência angular e fase inicial respectivamente.

A amplitude pode ser vista como a magnitude da senóide em um determinado instante de tempo e satisfaz a relação $|x(t)| \leq A$. A freqüência angular é dada por $\omega = 2\pi f$, onde f é dado em Hz (Hertz). A fase inicial representa o deslocamento da onda em relação ao eixo que corresponde a amplitude no instante $t = 0$. A fase instantânea é definida por $\omega t + \phi$. Note que se derivarmos a fase instantânea em função do tempo obteremos a freqüência angular

$$\frac{d}{dt}\,(\omega t + \phi) = \omega$$

Neste caso, como a fase instantânea varia com o tempo, a sua derivada resulta na freqüência instantânea. Na Figura 3, podemos observar o gráfico de uma senóide.

A cada T segundos, a função correspondente a senóide completa um ciclo. Uma observação importante é que a função seno é periódica com período igual a 2π, ou seja,

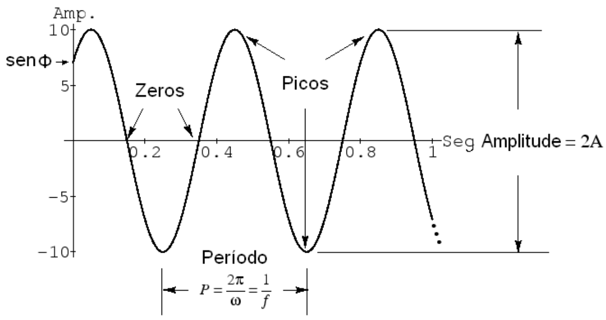

Figura 3: Exemplo de uma senóide $x(t) = 5sin\left(\omega t + \frac{\pi}{2}\right)$

$\phi = \pm 2\pi$ não proporciona um resultado diferente. Podemos então trabalhar considerando o intervalo

$$-\pi \leq \phi < \pi.$$

Se adicionarmos $\pi/2$ à fase inicial da senóide, obteremos uma cossenóide de mesma freqüência, ou seja,

$$\text{sen}\left(\omega t + \frac{\pi}{2}\right) = \cos(\omega t).$$

Podemos observar essa diferença de fase na Figura 4.

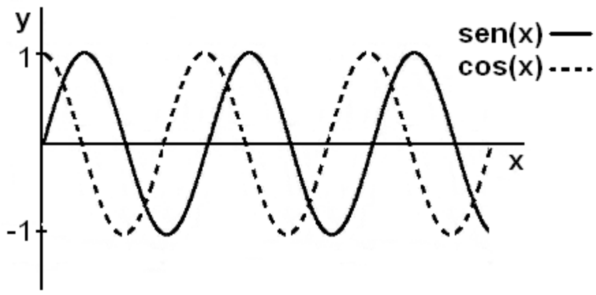

Figura 4: Defasamento entre seno e cosseno

Com observação neste resultado, dizemos que a função seno, que possui fase zero, é chamada de componente em fase e a função cosseno, que possui fase $\pi/2$,é chamada componente em quadratura [Smi03]. Nota-se portanto que as duas funções possuem uma diferença de fase de 90°.

Também é interessante observar o gráfico de uma senóide no domínio da freqüência. Na Figura 5, temos a senóide $x(t) = \text{sen } 50t$ e sua visão espectral. Note a existência de duas componentes de freqüência, uma positiva com valor de 25HZ e outra negativa com valor de -25Hz, e cada uma possuindo metade da amplitude da senóide. O gráfico não apresenta a fase inicial da senóide que no caso é zero.

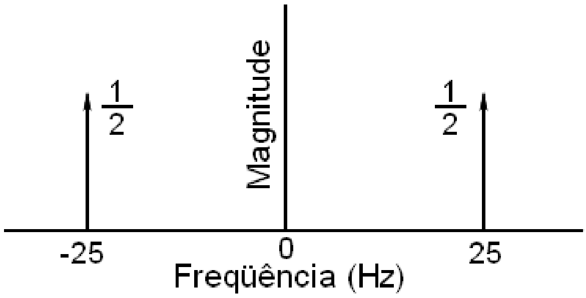

Figura 5: Visão espectral de uma senóide real

2.1.2 Funções Exponenciais

A forma canônica de uma função exponencial, da forma que é comumente usada em processamento de sinais é:

$$a(t) = Ae^{-t/r} \tag{2.2}$$

onde r é chamado de constante de tempo e A é a amplitude. Podemos notar que

$$\frac{a\,(t)}{a\,(0)} = \frac{1}{e},$$

onde

$$a(t) = \frac{Ae^{-t/r}}{Ae^{-0/r}},$$

representa o decaimento da função em relação ao tempo. A Figura 6 mostra uma exponencial normalizada decaindo.

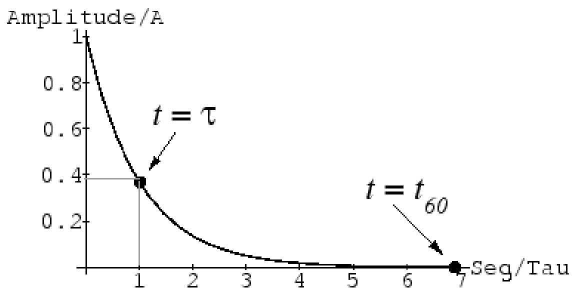

Figura 6: Função exponencial decaindo [Smi03]

A importância de comentar sobre exponenciais é que o decaimento é um acontecimento que ocorre naturalmente, como um instrumento de cordas que exibe um decaimento exponencial em resposta a uma excitação momentânea [Smi03]. Existe também o crescimento exponencial, porém este crescimento é instável.

2.1.3 Senóides Complexas

Uma senóide complexa é definida por [Smi03]

$$Ae^{i(\omega t+\phi)} = A\cos(\omega t+\phi) + iA\,\mathrm{sen}(\omega t+\phi) \tag{2.3}$$

onde, como para a senóide real, t é a variável independente de valor real e A, ω, e ϕ são constantes reais e correspondem a amplitude, freqüência angular e fase inicial respectivamente. Uma introdução sobre os números complexos pode ser vista no Apêndice A.

Se compararmos a equação definida para uma senóide complexa com a identidade de Euller, que é dada pela Equação A.3, concluiremos que a senóide complexa é uma extensão dessa definição dada pelo simples acréscimo da amplitude A e fazendo $\theta = \omega t + \phi$ [Smi03].

A senóide complexa é formada por um componente em fase (seno) para sua parte real e um componente em quadratura (cosseno) para sua parte imaginária. Sabendo que $\mathrm{sen}^2\theta + \cos^2\theta = 1$, conclui-se, observando a expressão abaixo, que a senóide complexa possui módulo constante.

$$|s(t)| = \sqrt{\mathrm{re}^2\{s(t)\} + \mathrm{im}^2\{s(t)\}} = A$$

A característica de possuir módulo constante leva a uma propriedade muito importante, pois sendo $\theta = \omega t$, a senóide complexa $x(t) = e^{i\omega t}$ descreverá um circulo no plano complexo traçado no sentido anti-horário e $x(t) = e^{-i\omega t}$ descreverá o mesmo circulo, porém no sentido horário. Portanto diz-se que uma senóide complexa $x(t) = e^{i\omega t}$, com $\omega > 0$ possui freqüência positiva e sua conjugada $x(t) = e^{-i\omega t}$ possui freqüência negativa.

Somando uma senóide complexa $e^{i\theta}$ com sua conjugada $e^{-i\theta}$ obtém-se $2\cos\theta$. Se operarmos com a subtração nessas duas senóides complexas obtemos $-2i\ \mathrm{sen}\theta$. Dessa forma, podemos expressar seno e cosseno em termos de senóides complexas

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \tag{2.4}$$

$$\mathrm{sen}\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i} \tag{2.5}$$

Conseqüentemente, fica claro que senóides reais possuem um termo correspondente a um movimento circular de freqüência positiva e um termo correspondente a um movimento circular de freqüência negativa. Isso explica o aparecimento de dois componentes de freqüência na análise do espectro de uma senóide real mostrado na Figura 5. Senóides complexas são mais simples, pois possuem apenas um componente de freqüência, como pode ser visto na Figura 7, e além disso, possuem módulo constante. Podemos ainda acres-

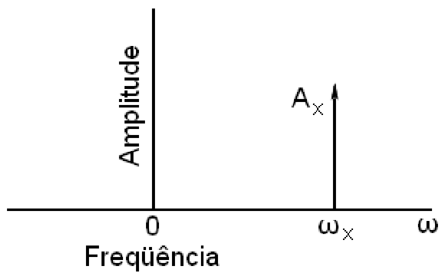

Figura 7: Visão espectral de uma senóide complexa

centar um envelope de amplitude a uma senóide complexa com uma função exponencial dada da seguinte forma

$$y(t) = \alpha e^{st}$$

onde $\alpha = Ae^{i\theta}$ e $s = \sigma + i\theta$, assim obtemos

$$y(t) = Ae^{i\phi}e^{\sigma + i\omega t} = Ae^{\sigma t}\left[\cos(\omega t + \phi) + iA \text{ sen}(\omega t + \phi)\right]$$

A função de envelope de amplitude é de $Ae^{\sigma t}$ que multiplica toda a senoíde complexa. Podemos mudar a notação fazendo $r = -t/r$, assim teremos

$$Ae^{-t/r}\left[\cos(\omega t + \phi) + iA \text{ sen}(\omega t + \phi)\right], \tag{2.6}$$

de maneira que agora o envelope de amplitude ficará da forma de função exponencial que foi definida na Equação 2.2.

Com o envelope de amplitude a senóide complexa não mais terá uma amplitude constante, ela decairá ou crescerá.

2.1.4 Transformada 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 freqüência como já foi mostrado em alguns exemplos anteriores. Uma forma de fazer essa transição de domínios é utilizando a transformada de Fourier que é definida como [Smi03]

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{2.7}$$

onde $f(t)$ e $F(\omega)$ são funções continuas.

A transformada de Fourier é uma transformada integral que expressa uma função em termos de funções de base senoidal. Dentro da integral, além da função $f(t)$, temos a senóide complexa $e^{-i\omega t}$ que representa a função de base senoidal. A transformada de Fourier expressa o produto interno entre essas duas funções. Para melhor entendimento observe a definição de produto interno para o espaço de funções continuas e complexas para um $[a, b]$

$$\langle f, g\rangle = \int_a^b f(t)\,\overline{g(t)}.$$

Isso explica porque na definição da transformada de Fourier temos a senóide complexa na sua forma conjugada.

Podemos interpretar que a transformada é uma operação de produto interno que calcula o coeficiente de projeção da função $f(t)$ sobre a senóide complexa [Smi03]. $F(\omega)$ é a medida da amplitude e da fase da senóide complexa que representa o sinal de entrada na freqüência ω.

A condição suficiente para a existência da transformada de Fourier de uma função $f(x)$ qualquer é que essa função seja integrável e finita [Mit98], ou seja

$$\int_{-\infty}^{\infty} |f(x)|\, dx < \infty.$$

Podemos observar na Tabela 1 algumas propriedades da transformada de Fourier que são importantes diante do contexto deste trabalho [Mit98]:

Tabela 1: Propriedades da Transformada de Fourier [Mit98]

Propriedade	Domínio do tempo	Domínio da freqüêcnia
Linearidade	$\alpha f(t) + \beta g(t)$	$\alpha F(\omega) + \beta G(\omega)$
Atraso no tempo	$f(t-\tau)$	$e^{-i\omega\tau}F(\omega)$
Atraso na Freqüência	$e^{i\omega_0 t}f(t)$	$F(\omega-\omega_0)$
Convolução	$f(t) * g(t)$	$F(\omega)G(\omega)$
Conjugado Simétrico	$\overline{f}(t)$	$\overline{F}(-\omega)$
Simetria par	$f(-t)$	$F(-\omega)$
Simetria ímpar	$-f(t)$	$F(\omega)$

Existe uma forma de fazer uma transição de domínios inversa, ou seja, passar do domínio da freqüência e ir para o domínio do tempo. Isso ocorre através da transformada de Fourier inversa que é definida como [Smi03]

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t} d\omega \qquad (2.8)$$

A transformada inversa é a reconstrução de um sinal a partir da sua projeção em N vetores diferentes que pertencem também aos complexos de tamanho N. Podemos interpretar a transformada de Fourier inversa como uma operação de projeção entre duas funções, que é definida por

$$P_f(g) = \frac{\langle f, g\rangle}{\|f\|^2} f.$$

2.2 Representações Discretas do Sinal de Áudio

Em uma máquina digital podemos apenas trabalhar com valores discretos, ou seja, a máquina é incapaz de obter uma represetação para todos os valores de um domínio contínuo. Em razão deste fato, devemos então discretizar o sinal amostrando seu domínio e quantizando seu contra-domínio (quantização será explicada na seção 3.1). Portanto, nessa seção, o sinal de áudio passará a ser representado como uma seqüência de números com a variável independente representada como um inteiro.

2.2.1 O Processo de Amostragem

Em uma seqüência, o valor de uma amostra é denotado como $x[n]$, com n sendo um inteiro entre $-\infty$ e ∞. No caso de uma seqüência $x[n]$ gerada por amostragem de um sinal contínuo temos

$$x[n] = x_a(t)|t = nT = x_a(nT). \tag{2.9}$$

O espaço T entre duas amostras consecutivas é chamado período de amostragem. F_T é chamado de freqüência de amostragem, dada por

$$F_T = \frac{1}{T} \quad \text{(Hz)}.$$

A unidade da freqüência de amostragem é ciclos por segundo, ou Hertz, e o período de amostragem é definido em segundos.

Pela relação da variável de tempo do sinal contínuo t com a variável do sinal discreto n, teremos t_n, definido apenas para instantes discretos dados por

$$t_n = nT = \frac{n}{f_T} = \frac{2\pi n}{\omega_T}, \tag{2.10}$$

onde $\omega_T = 2\pi f_T$ denota a freqüência angular de amostragem. Podemos definir uma freqüência angular Ω_0 de um sinal discretizado no tempo $x[n]$ pela relação com a freqüência analógica ω_0 por

$$\Omega_0 = \frac{2\pi\omega_0}{\omega_T} = \omega_0 T, \tag{2.11}$$

onde Ω_0 é a freqüência angular digital normalizada do sinal discreto $x[n]$. A unidade de Ω_0 é radianos por amostra e a unidade de ω_0 é radianos por segundo.

Por exemplo, para obtermos um sinal digital com 3 Hz a partir de uma senóide e utilizando uma taxa de amostragem de 10 Hz fazemos

$$x[n] = \cos\left(\Omega_0 n + \phi\right)$$

$$\Omega_0 = \frac{2\pi \times 3}{10} = 0.6\pi$$

$$x[n] = \cos\left(0.6\pi n\right).$$

Para um sinal digital com 7 Hz teríamos $x[n] = \cos\left(1.4\pi n\right)$ porém

$$\cos\left(1.4\pi n\right) = \cos\left(\left(2\pi - 0.6\pi\right) n\right) = \cos(0.6\pi n).$$

Podemos notar assim uma ambigüidade na representação desses dois sinais de freqüência diferentes para esta taxa de amostragem. A este fenômeno, de um sinal continuo no tempo representado por uma senóide de alta freqüência adquirir a identidade de uma senóide de freqüência mais baixa depois da amostragem, damos o nome de, *aliasing* [Mit98]. Para que o processo de amostragem ocorra de forma consistente devemos ter $\omega_T > 2\omega_{max}$, onde ω_{max} é a maior freqüência do sinal a ser amostrado.

2.2.2 Transformada de Fourier Discreta

A transformada de Fourier discreta de uma seqüência $x[n]$ é definida por [Mit98]

$$X[\omega_k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi nk/N}, \quad k = 0, 1, 2, \ldots N-1, \tag{2.12}$$

onde $X[\omega_k]$ é uma seqüência finita no domínio da freqüência e possui N amostras.

A senóide complexa que aparece na trasformada de Fourier discreta é chamada de *kernel*. O *kernel* consiste de amostras de uma senóide complexa em N freqüências discretas ω_k uniformemente espaçadas entre 0 e a taxa de amostragem $\omega_T = 2\pi f_T$.

Podemos interpretar que a transformada é uma operação de produto interno que calcula o coeficiente de projeção da seqüência $x[n]$ sobre a senóide complexa $\cos\left(\omega_k n\right) + i\ \mathrm{sen}(\omega_k n)$. $X(\omega_k)$ representa a medida da amplitude e da fase da senóide complexa que representa a seqüência de entrada na freqüência ω_k.

A transformada de Fourier discreta inversa de uma seqüência de amostras de freqüência é definida por [Mit98]

$$x[n] = \frac{1}{2\pi} \sum_{k=0}^{N-1} X[\omega_k] e^{i2\pi nk/N}, \quad k = 0, 1, 2, \ldots N-1, \tag{2.13}$$

onde $x[n]$ é uma seqüência finita no domínio do tempo e possui N amostras.

Esta transformada inversa é a reconstrução de um sinal $x \in \mathrm{C}^N$ da sua projeção em N vetores diferentes que pertencem também aos complexos de tamanho N. Um extenso estudo sobre a matemática envolvida pela transformada de Fourier discreta pode ser encontrado em [Smi03].

2.2.3 Transformada z

Podemos entender a transformada z como uma generalização da transformada de Fourier, que pode existir para seqüências em que a transformada de Fourier não existe [Mit98]. Além disso, o uso da transformada z permite manipulações algébricas mais simples, sendo muito utilizada no estudo de sistemas discretos. A transformada z de uma seqüência $g[n]$ é definida por [Mit98]

$$G(z) = \mathcal{Z}\{g[n]\} = \sum_{n=-\infty}^{\infty} g[n] z^{-n}, \tag{2.14}$$

onde z é uma variável complexa. Fazendo $z = re^{i\omega}$, teremos

$$G(re^{i\omega}) = \sum_{n=-\infty}^{\infty} g[n] r^{-n} e^{-i\omega n}$$

que pode ser considerado a transformada de Fourier da seqüência $g[n]r^{-r}$. Para $r = 1$ a transformada z de $g[n]$ se reduz a transformada de Fourier. O contorno $|z| = 1$ é um círculo no plano z de raio unitário e é chamado círculo unitário que pode ser visto na Figura 8. Para mais detalhes sobre o círculo unitário e o plano dos complexos visite o Apêndice B.

Podemos ver o resultado da transformada z de um sinal como um polinômio em z^{-1}, com coeficientes dados pelas amostras do sinal. Por exemplo, para o seguinte sinal

$$x(n) = \begin{cases} n+1, & 0 \le n \le 2 \\ 0, & \text{senão} \end{cases}$$

sua transformada z será

$$X(z) = 1 + 2z^{-1} + 3z^{-2} = 1 + 2z^{-1} + 3(z^{-1})^2$$

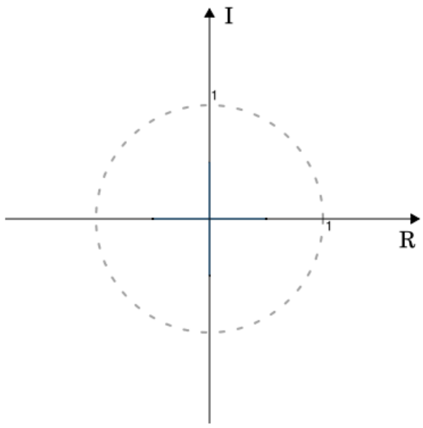

Figura 8: Círculo Unitário

Em processamento de sinais, normalmente usa-se a variável s da transformada de Laplace para análise no domínio contínuo. Para um domínio discreto, usamos a variável z relativo a transformada z [Lat04]. Portanto, para sistemas contínuos no tempo, o domínio da freqüência é o plano s, enquanto para sistemas discretos, o domínio da freqüência é o plano z.

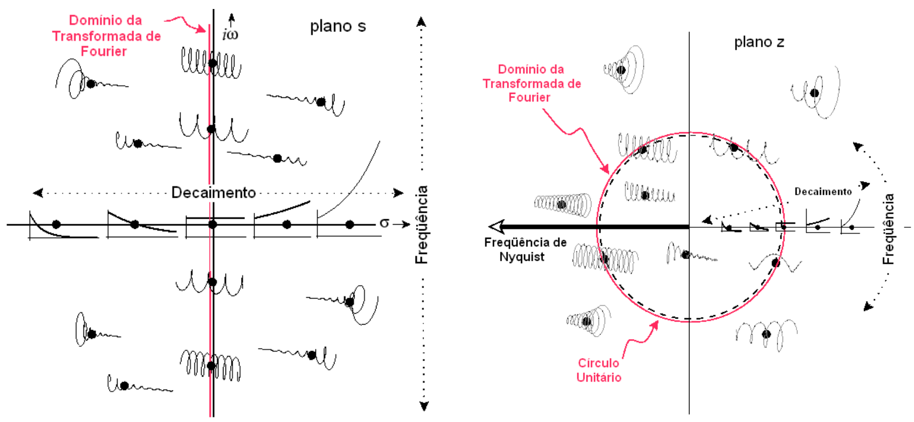

Figura 9: (a) Plano s (Domínio da Transformada de Laplace) e (b) Plano z (Domínio da Transformada z) [Smi03].

Como podemos ver na Figura 9a, no plano s, o eixo de freqüência é $s = i\omega$ e os pontos sobre ele correspondem a senóides complexas. Na Figura 9b, temos o plano z, seu eixo de freqüência é o círculo unitário e os pontos sobre ele correspondem a senóides complexas amostradas [Smi03].

2.3 Operações em Sinais

Para obter sinais derivados de algum processamento necessitamos usar algumas operações nos sinais de entrada a fim de alterar alguma das propriedades do sinal que estamos manipulando.

2.3.1 Convolução

Da transformada de Fourier, encontramos uma importante propriedade que nos leva a necessidade de entender a convolução. Tal propriedade é a seguinte

$$f(t) * g(t) = F(\omega)G(\omega),$$

onde $f(t)$ e $g(t)$ são funções contínuas em função do tempo.

O produto de convolução é definido da seguinte forma [Lat04]

$$f(t) * g(t) = \int_0^t f(\tau)g(t-\tau)d\tau. \tag{2.15}$$

A convolução é definida como a integral do produto de uma das funções com uma cópia revertida e deslocada da outra. Para se adquirir uma boa visão intuitiva da convolução, é preciso entender que diversas cópias transladadas e tomadas de trás para frente de uma das funções são ponderadas pelo valor da outra função, e somadas, produzem o resultado. A convolução é uma operação matemática que possui três propriedades, comutatividade, associatividade e distributividade.

2.3.2 Operações em Seqüências

Um sistema discreto no tempo opera sobre uma seqüência de entrada e produz uma seqüência de saída. A seqüência produzida, muitas vezes, possui propriedades mais interessantes. Uma entrada de um sistema, por exemplo, pode ser um sinal com ruído e a saída seria um sinal sem esse ruído, que seria retirado pelo sistema por meio de operações realizadas. Um sistema pode ter um número de seqüências de entradas diferente do número de seqüências de saída. Para saber como as seqüências são classificadas veja o Apêndice C.

Uma operação possível em relação a seqüências é o produto. Podemos formar uma nova seqüência aplicando esta operação deste modo

$$w_1[n] = x[n]y[n].$$

Este produto é conhecido como modulação. Uma aplicação desta operação é transformar uma seqüência com componentes senoidais de baixa freqüência em uma seqüência com altas freqüências. Outra aplicação desta operação é a formação de seqüências de tamanho finito de uma seqüência que antes era infinita. Isto ocorre simplesmente pegando-se uma seqüência finita, denominada como seqüência janela, e fazendo o produto desta com a seqüência infinita.

Outra operação básica é a adição que é o simples processo de somar o valor das amostras de duas seqüências:

$$w_2[n] = x[n] + y[n].$$

A multiplicação por um escalar, que é o produto de um número por todas amostras da seqüência, é dada por

$$w_3[n] = Ax[n].$$

A operação de reversão no tempo é dada por

$$w_4[n] = x[-n],$$

e finalmente a operação de atraso, que faz com que as amostras desloquem para a direita, quando N é positivo, é definida por

$$w_5[n] = x[n - N].$$

No caso da operação de atraso, se tivermos um valor negativo para N, o deslocamento das amostras será para a esquerda e teremos uma adiantamento ao invés do atraso.

Tendo o conhecimento de seqüências básicas (Apêndice C) e das operações básicas, podemos agora representar determinados tipos de seqüência. A representação de uma seqüência arbitrária no domínio do tempo pode ser dada pela soma de pesos das seqüências básicas e também de suas versões com algum atraso.

A seqüência mais comumente usada é a seqüência de amostra pontual, definida em C.2. Um exemplo de seqüência $x[n]$ usando amostras pontuais pode ser visto como

$$x[n] = \delta[n+3] + \delta[n+2] + \delta[n-1]$$

2.3.3 Sistemas Discretos no Tempo

Um sistema discreto no tempo processa uma ou mais seqüências de entrada e produz uma ou mais seqüências de saída. A seqüência de saída é gerada seqüencialmente de acordo com um índice relacionado a variável de tempo inicial. Por exemplo, para uma seqüência de entrada $x[n]$, é gerada uma seqüência de saída $y[n]$, supondo n_0 sendo o início do índice, relacionado ao tempo, primeiramente $y[n_0]$ será gerado, posteriormente $y[n_0+1]$ e assim sucessivamente.

Um exemplo de sistema discreto no tempo é dado por uma expressão que determina a seqüência de saída

$$y[n] = a_1x[n] + a_2x[n-1] + a_3x[n-2].$$

O diagrama desse sistema pode ser visto na figura abaixo. Existem outros exemplos de

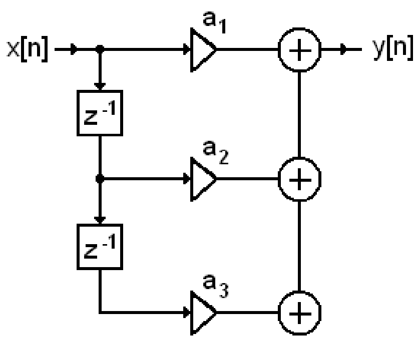

Figura 10: Sistema representado por um diagrama

sistemas comumente usados para processar sinais. Um deles é o acumulador que soma todos os valores de amostras de uma seqüência de entrada de -∞ até n.

$$y[n] = \sum_{k=-\infty}^{n} x[k].$$

Podemos classificar esses sistemas baseando-se em suas propriedades existentes nas relações de entrada e saída. O tipo de sistema discreto no tempo mais usado em processamento de sinais é o Linear. Um sistema ser classificado como linear significa que o princípio da superposição é assegurado. Para as seqüências de entrada $x_1[n]$ e $x_2[n]$ um sistema discreto no tempo linear produz $y_1[n]$ e $y_2[n]$, temos a entrada

$$x[n] = \alpha x_1[n] + \beta x_2[n],$$

a saída produzida será

$$y[n] = \alpha y_1[n] + \beta y_2[n].$$

Portanto pelo princípio da superposição a propriedade de multiplicação por um escalar é conferida para qualquer número racional. Se a saída de $x[n]$ for $y[n]$, então a saída de $cx[n]$ será $cy[n]$.

Outra propriedade muito importante é a da invariância no tempo [Lat04], propriedade presente na maioria dos sistemas discretos no tempo de filtros digitais. Invariância no tempo significa que se aplicarmos uma entrada no sistema em um tempo t_0 ou alguns segundos depois, a saída será idêntica em ambos os casos, exceto por um atraso de T segundos. Formalmente temos que a resposta a uma entrada

$$x[n] = x_1[n - n_0]$$

será:

$$y[n] = y_1[n - n_0]$$

com n_0 sendo um inteiro positivo ou negativo.

Um sistema linear invariante no tempo satisfaz a propriedade de linearidade e a propriedade de invariância no tempo. Um sistema que satisfaz ambas as propriedades é matematicamente fácil de trabalhar e pode ser caracterizado inteiramente por uma única função chamada de *resposta ao impulso* do sistema. Também são sistemas muito usados em algoritmos de processamento de sinais.

Outra propriedade de sistemas é a causalidade onde a amostra n_0 de uma seqüência de saída de um sistema discreto no tempo $y[n]$ depende apenas das amostras de $x[n]$ em que $n \leq n_0$. Existem também a propriedade de estabilidade e passividade que estão relacionadas respectivamente ao limite do valor das amostras e energia existente nas seqüências.

2.3.4 Filtros

A resposta de um filtro digital para uma seqüência de amostra pontual $\delta[n]$ é chamada resposta à amostra pontual, ou resposta ao impulso, e é denotada por $h[n]$. Desta maneira, podemos observar que um sistema discreto no tempo, linear e invariante no tempo, como por exemplo, um filtro digital, poderá ser caracterizado no domínio do tempo por sua resposta ao impulso.

Para um sistema definido pela seguinte equação à diferenças

$$y[n] = a_1 x[n] + a_2 x[n-1] + a_3 x[n-2],$$

a resposta ao impulso $h[n]$ fazendo $x[n] = \delta[h]$ será

$$h[n] = a_1 \delta[n] + a_2 \delta[n-1] + a_3 \delta[n-2]$$

Podemos usar também a resposta do sistema discreto no tempo em relação a seqüência de degrau unitário, definida na Equação C.3. Como um sistema discreto no tempo, linear e invariante no tempo pode ser especificado por sua resposta ao impulso, se conhecermos esta resposta ao impulso, poderemos computar a saída do sistema para qualquer entrada.

Fazendo $h[n]$ representar a resposta ao impulso para uma entrada $\delta[n]$ do sistema dado no exemplo, como o sistema é invariante no tempo a resposta para $\delta[n-1]$ será $h[n-1]$, e também pela linearidade podemos escrever toda a resposta do sistema acima como

$$y[n] = a_1 h[n] + a_2 h[n-1] + a_3 h[n-2].$$

Assim podemos generalizar esta relação para qualquer seqüência de entrada $x[n]$ expressando da forma

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

A resposta do sistema para a seqüência $x[n]\delta[n-k]$ será $x[n]h[n-k]$, desta maneira a resposta do sistema para $x[n]$ será

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]. \tag{2.16}$$

O somatório apresentado acima é chamada de produto de convolução e pode também ser escrita em uma representação mais compacta dada por

$$y[n] = x[n] * h[n].$$

O produto de convolução foi definido para funções contínuas em 2.3.1. Podemos notar que o produto de convolução é formado de operações aritméticas simples. Portanto, pela propriedade de convolução, dada na Tabela 1 podemos usar a convolução para alterarmos o domínio da freqüência de um sinal. Desta maneira, podemos calcular a resposta de um sistema discreto no tempo, linear e invariante no tempo de maneira rápida com o auxilio do computador.

2.3.4.1 Função de Transferência

Sabemos então que $h[n]$ representa a resposta ao impulso de um sistema, porém esta seqüência é representada no domínio do tempo, transformando-a para o domínio da freqüência teremos então

$$H[\omega_k] = \sum_{n=-\infty}^{\infty} h[n] e^{-i2\pi nk/N}. \tag{2.17}$$

Esta nova seqüência é chamada de resposta em freqüência do sistema $h[n]$.

Uma generalização da função de resposta em freqüência, $H[\omega_k]$, nos leva ao conceito da função transferência. Entretanto, usar uma função complexa de uma variável de freqüência ω torna difícil a sua manipulação para a realização do filtro digital. Portanto, usamos a transformada z.

Considerando um sistema discreto no tempo, linear e invariante no tempo, a relação entrada-saída do sistema é dado por

$$y[n] = h[n] * x[n].$$

A transformada z possui a mesma propriedade de convolução da transformada de Fourier (Tabela 1), assim podemos ter

$$Y(z) = H(z)X(z).$$

Da relação anterior, chegamos em

$$H(z) = \frac{Y(z)}{X(z)}.$$

Onde a quantidade $H(z)$ que é a transformada z da resposta ao impulso $h[n]$ do filtro, é mais comumente chamada de *função de transferência*.

Podemos aplicar a transformada z em equações à diferenças. Por exemplo

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) - a_1 y(n-1) - \cdots - a_N y(n-N)$$

$$\mathcal{Z}\left\{y[n]\right\} = \mathcal{Z}\left\{b_0x[n] + b_1x[n-1] + \cdots + b_Mx[n-M] - a_1y[n-1] - \cdots - a_Ny[n-N]\right\}$$

distribuindo

$$\mathcal{Z}\left\{y[n]\right\} = \mathcal{Z}\left\{b_0x[n]\right\} + \mathcal{Z}\left\{b_1x[n-1]\right\} + \cdots + \mathcal{Z}\left\{b_Mx[n-M]\right\} - \cdots$$

$$-\mathcal{Z}\left\{a_1y[n-1]\right\} \cdots \mathcal{Z}\left\{a_Ny[n-N]\right\}$$

$$Y(z) = b_0X[z] + b_1z^{-1}X[z] + \cdots + b_Mz^{-M}X[z] - a_1z^{-1}Y[z] - \cdots - a_Nz^{-N}Y[z]$$

Agrupando os termos com Y(z)

$$Y(z) + a_1z^{-1}Y(z) + \cdots + a_Nz^{-N}Y(z) = b_0X(z) + b_1z^{-1}X(z) + \cdots + b_Mz^{-M}X(z)$$

Fatorando os termos comuns

$$Y(z)\left[1 + a_1z^{-1} + \cdots + a_Nz^{-N}\right] = X(z)\left[b_0 + b_1z^{-1} + \cdots + b_Mz^{-M}\right]$$

Pela definição de função transferência, fazendo $Y(z)/X(z)$ temos

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + \cdots + b_Mz^{-M}}{1 + a_1z^{-1} + \cdots + a_Nz^{-N}}$$

e assim, para obter a resposta ao impulso, pode-se operar com a transformada z inversa.

Olhando para a forma da função transferência, podemos notar que o objetivo em um projeto de construção de um filtro para sinais de áudio é escolher os coeficientes que projetam uma resposta em freqüência que filtre as freqüências desejadas.

2.3.4.2 Projeto de Filtros

Antes de construirmos um filtro, é importante planejá-lo de acordo com os parâmetros desejados. Veremos que, como não é possível obter um filtro ideal, devemos escolher o que melhor se aproxima e obtém um melhor resultado para a solução desejada.

Podemos classificar uma função de transferência no domínio do tempo baseando em seu tamanho. Desta maneira, teremos uma resposta ao impulso finita ou uma resposta ao impulso infinita. Outra forma de classificar é de acordo com o formato da função magnitude $|H(z)|$ ou a forma de fase $\theta(\omega)$

Existem quatro tipos de filtros ideais que são designados a passar certas componentes de freqüências do sinal sem nenhuma distorção. Esses filtros possuem uma resposta em freqüência de valor igual a um para a faixa de freqüência que deve passar e resposta em freqüência de valor zero para todas as outras freqüências, bloqueando totalmente componentes do sinal com essas freqüências [Ant93].

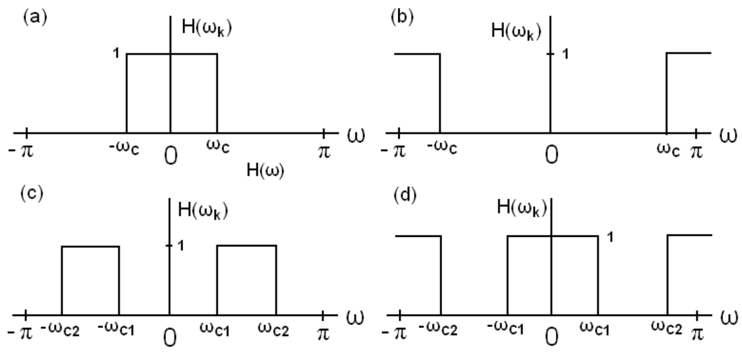

Figura 11: Resposta em freqüência de Filtros Ideais. (a)Passa-Baixa, (b)Passa-Alta, (c)Passa-faixa e (d)Rejeita-faixa

Na Figura 11, o eixo horizontal é a freqüência angular e o eixo vertical representa a fração de cada componente de freqüência que passará no processo de filtragem do sinal. No passa-baixa, por exemplo, os componentes de freqüência entre $-\omega_c$ e ω_c, onde ω_c é a freqüência de corte, não serão afetados, enquanto que os outros componentes de freqüência serão removidos completamente. A freqüência angular do gráfico está normalizada.

Para obtermos a resposta ao impulso de um filtro ideal devemos então aplicar a transformada de Fourier discreta inversa. Para o passa-baixa temos

$$h_{ideal}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{ideal}[\omega]e^{i\omega n}d\omega = \frac{1}{2\pi}\int_{-\omega}^{\omega} e^{i\omega n}\ d\omega =$$

$$\frac{1}{2\pi}\int_{-\omega}^{\omega} \cos n\omega + i \text{ sen } n\omega\ d\omega = \frac{1}{2\pi}\int_{-\omega}^{\omega} \cos n\omega + i \text{ sen } n\omega\ d\omega$$

$$h_{ideal}[n] = \frac{\text{sen } n\omega_c}{n\omega_n} \tag{2.18}$$

para $-\infty < n < \infty$, onde $\omega = 2\pi k/N$. Essa função é conhecida como função *sinc*.

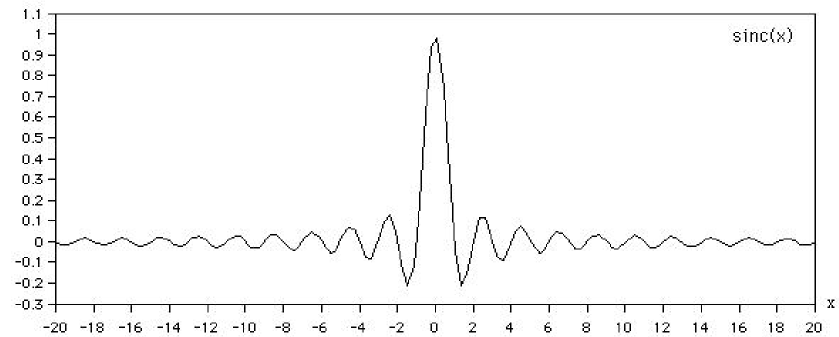

Figura 12: Função *sinc*

Desta forma, podemos ver que na prática é impossível construir um filtro linear e invariante no tempo com resposta em freqüência ideal, pois teríamos que trabalhar com uma função de tamanho infinito. Um modo de tornar possível o uso da função *sinc* como resposta ao impulso é truncando alguns de seus valores. Para tal ação, geralmente usa-se uma função que provoca um efeito conhecido como "janelamento", que torna a função *sinc* finita e produz uma resposta em freqüência aproximada para um filtro ideal. Existem também outros tipos de aproximações para a resposta em freqüência que são apresentadas na próxima seção.

2.4 Solução para o Problema da Equalização

Neste ponto, já é possível entender como um filtro é construído e quais são suas propriedades. Porém um simples filtro não é o objetivo deste trabalho como também a construção de um equalizador.

Um equalizador controla ganhos de faixas de freqüências de um sinal de áudio. Sendo assim não é difícil de notar que podemos construir um equalizador pela soma da resposta em freqüência de n filtros, cada um projetado para passar sinal em uma faixa de freqüência determinada.

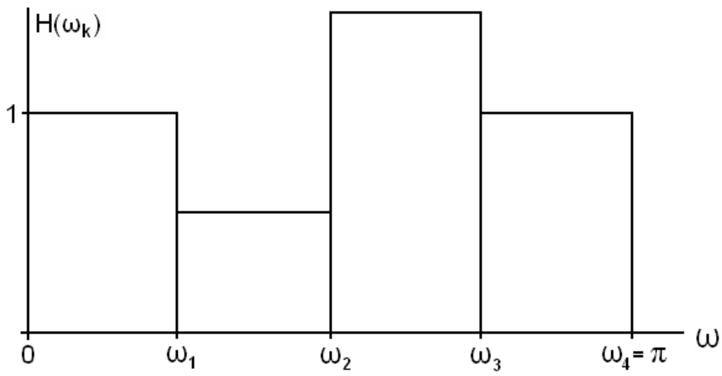

Figura 13: Esquema para um equalizador ideal com 4 bandas

Na Figura 13, temos a representação gráfica de um equalizador ideal de 4 bandas. Podemos notar então que a solução seria usar 4 filtros, um passa-baixa com faixa de passagem de freqüências menores que ω_1, dois passa-faixa, o primeiro para o intervalo de freqüências entre ω_1 e ω_2 e o segundo para o intervalo ω_2 e ω_3 e por fim, um passa-alta com faixa de passagem para freqüências maiores que ω_3.

Também podemos observar na Figura 13 que o equalizador exemplificado já possui algumas faixas atuando com ganhos e atenuações. Fazendo α_i denotar o ganho de uma faixa, com $i = 1, 2, 3, 4$, teremos que para a primeira faixa do equalizador $\alpha_1 = 1$ para $\omega < \omega_1$, portanto não há alteração do domínio da freqüência nesta faixa. Em $\omega_1 \leq \omega < \omega_2$ temos $\alpha_2 < 1$ o que indica uma atenuação da faixa, porém em $\omega_2 \leq \omega < \omega_3$ temos $\alpha_3 > 1$ indicando um ganho nesta faixa. Por fim, $\omega_3 \leq \omega < \omega_4$ possui $\alpha_4 = 1$ não existindo alteração nesta faixa.

Veremos nas próximas seções como obter os filtros para cada faixa de freqüência para esse exemplo de equalizador.

2.4.1 Equalização no Domínio da Freqüência

Uma das formas de se aplicar ganhos a determinadas faixas de freqüências é com a Transformada de Fourier Discreta.

Seja $H_i(\omega_k)$ as funções de transferência para os l intervalos $f_i = \{\omega \mid \omega \in [\omega_i, \omega_{i+1})\}$ que formam uma partição do espectro. Um equalizador é a aplicação $EQ(x[n]) = TFD^{-1}(X(\omega_k) \cdot \prod_{i=1}^{l} H_i(\omega_k))$.

$$H_i(\omega_k) = \begin{cases} \alpha_i & \text{se } \omega_k \in f_i, \\ 0 & \text{se } \omega_k \notin f_i, \text{ para } i = 1, 2, ..., l. \end{cases} \tag{2.19}$$

Note que a transformação $EQ(x[n])$ não impõe restrições às faixas f_i e seus pesos α_i. Entretanto, truncagens na série de Fourier produzem imperfeições na resposta em freqüência produzidas pelas funções de transferência $H_i(\omega_k)$. Veremos na seção seguinte como obter aproximações para estas respostas em freqüências.

2.4.2 Aproximações de um Filtro Digital

Existem diferentes maneiras de se obter aproximações de respostas em freqüência e cada uma delas possui suas propriedades. Então, torna-se interessante estudar tais aproximações e escolher a que melhor se adapta a solução do problema.

A aproximação pelo truncamento da função *sinc* (Equação 2.18) produz algumas ondulações na faixa de passagem e na faixa de rejeição de uma resposta em freqüência. Essas ondulações podem ser vistas como imperfeições causadas pela aproximação provocada pela truncamento. Outro resultado indesejado é a banda de transição que passa a existir, tirando a possibilidade de filtrar apenas uma faixa exata de freqüências.

Na Figura 14, podemos observar as ondulações, onde δ representa a variação da ondulação.

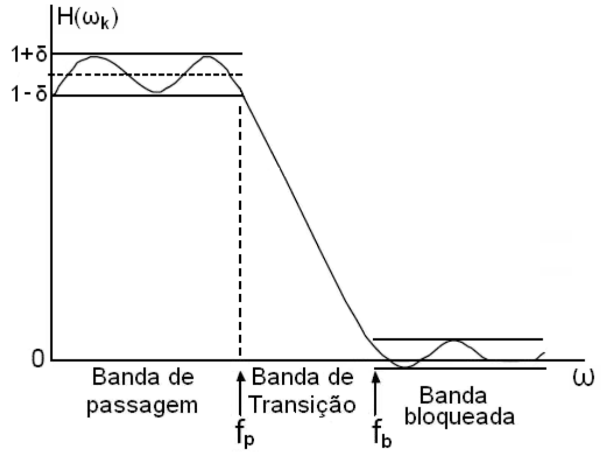

Figura 14: Resposta em Freqüência aproximada pela função *sinc*

Uma aproximação muito usada é a de Butterworth [Dar76]. A resposta em freqüência de um filtro Butterworth é bem plana na faixa de passagem, ou seja, não possui ondulações e também na faixa de rejeição, se aproximando do zero. Esse é um aspecto positivo do Butterworth, porém a banda de transição gerada por este tipo de filtro é grande.

A função de transferência tem uma resposta em magnitude dada por [Dar76]

$$G_n(\omega) = |H(\omega)| = \frac{1}{\sqrt{1 + (\omega/\omega_c)^{2n}}}$$

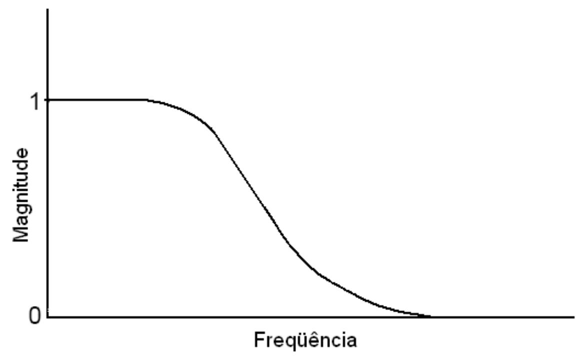

Figura 15: Reposta de um filtro Butterworth

Outra aproximação muito conhecida é a de Chebyshev. Podemos dizer que as características da resposta em freqüência produzida por um filtro utilizando Chebyshev é o contrário do Butterworth. Um filtro Chebyshev exibe ondulações na banda passante ou na banda rejeitada, porém possui uma banda de transição menor que a do filtro Butterworth.

Existem dois tipos de funções de transferência de Chebyshev. A aproximação do tipo 1 possui ondulações na banda de passagem, mas não possui ondulações na banda rejeitada. A aproximação do tipo 2 inverte essa situação. Um estudo mais detalhado deste assunto pode ser visto em [Dar76] e [Ant93].

Para o tipo 1 temos [Dar76]:

$$G_n(\omega) = |H(\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

Para o tipo 2 temos [Dar76]:

$$G_n(\omega) = |H(\omega)|^2 = \frac{1}{\sqrt{1 + \frac{1}{\epsilon^2 T_n^2(\omega_0/\omega)}}} \gg$$

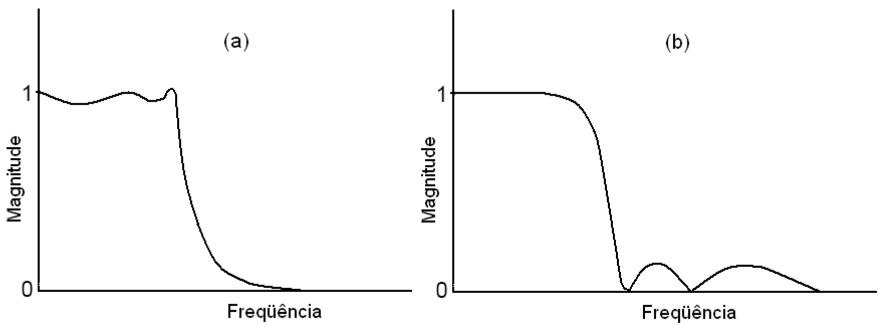

Figura 16: Reposta de um filtro Chebyshev (a) Tipo 1 e (b) Tipo 2

A existência da banda de transição passa a ser um problema na construção do equalizador, pois quando somamos a resposta em freqüência das aproximações acontece como na Figura 17. Podemos notar que as bandas de transição são somadas e assim haverá

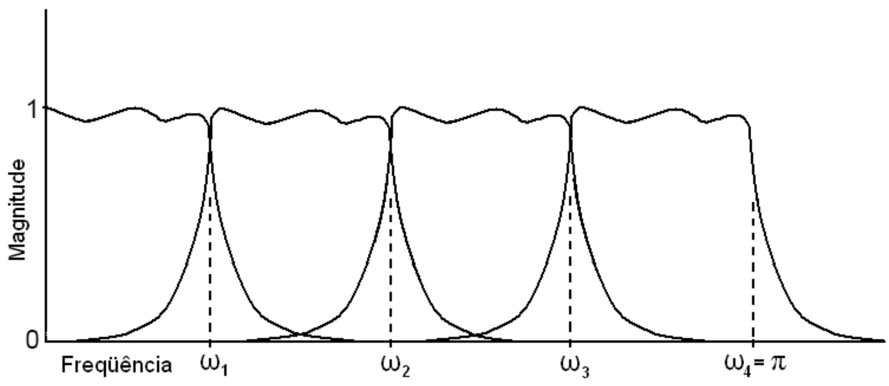

Figura 17: Reposta em freqüência de um equalizador usando aproximações

um ganho maior que o esperado para tais freqüências. Portanto, devemos minimizar este fato.

Outra consideração a ser feita é sobre as ondulações. Também é interessante minimizá-las para se obter uma resposta em freqüência constante na banda de passagem.

3 Modelo Computacional

Até aqui observamos a matemática necessária para que possamos processar um sinal. Vimos que podemos discretizar um sinal de áudio analógico obtendo assim uma representação digital para o mesmo. A intenção deste trabalho é usar o computador como o hardware para processar o sinal, portanto veremos agora como o computador pode ser útil na solução do problema em questão.

3.1 Áudio Digital

Há muito tempo o homem é capaz de armazenar sinais de áudio encontrados na natureza. O uso de áudio digital é relativamente recente, antes desta forma de armazenamento ser usada, as fitas de plástico criadas pela BASF eram gravadas de forma magnética. O Magnetofone, inventado em 1936, foi o primeiro gravador magnético que usava este tipo de fita. Outro modo de gravação analógica, porém mecânica, é a gravação utilizada nos discos de vinil [Ser02].

Apesar de todos avanços introduzidos pela tecnologia digital, ainda hoje são usadas fitas de rolo para gravação profissional de forma analógica, e segundo alguns músicos ou profissionais da área, pode-se obter resultados até melhores do que com gravações digitais.

Um som gravado de forma digital é convertido para valores discretos, já a gravação por processo analógico possibilita a obtenção de valores contínuos para a representação do sinal de entrada. Deste modo é claro perceber que no meio digital existem arredondamentos dos valores das amostras obtidas do sinal de entrada, enquanto no meio analógico são usados valores reais já que o valor do pulso elétrico gerado pelo sinal de entrada pode ser armazenado diretamente na mídia de destino.

A popularização do formato de armazenamento digital do áudio se iniciou na década de 80, quando a Phillips anunciou o lançamento do CD (*Compact Disc*). O armazenamento de áudio digital não está restrito ao CD, ele é apenas uma forma portátil de armazenar o sinal, mas este tipo de armazenamento se desenvolveu obviamente nos computadores. O que a gravação digital faz é a essência de qualquer informação armazenada em um computador, ou seja, converter esses dados em uma seqüência de dígitos binários.

No processo de gravação digital, os impulsos elétricos enviados pelo microfone são transformados em uma seqüência de números que são armazenados. Este arquivo de áudio gerado pode ser processado ou simplesmente executado, que seria a conversão dessa seqüência de números em impulsos elétricos que seriam enviados para uma caixa de som por exemplo. Em um computador comum, o equipamento que realiza este processo de captação dos impulsos elétricos e sua subseqüente conversão em seqüência de números é a placa de som. O processo inverso (seqüência de números para impulso elétrico) também é função desta parte do hardware.

Aprofundando mais no processo de gravação digital, podemos entrar em detalhes importantes a serem observados, como, por exemplo, a quantização dos valores da seqüência de números que formam o sinal digital. Outro fator importante é em relação a amostragem do sinal, pois o som é uma grandeza que varia no tempo. Assim o número de amostras por unidade de tempo e a quantidade de dados usados para a representação de cada amostra tornará uma variável influente na qualidade e tamanho físico do arquivo que representará o sinal de áudio.

A taxa de amostragem define com que freqüência o áudio será capturado e discretizado. Assim, cada amostra terá a informação do sinal de entrada em um instante de tempo que dependerá da taxa de amostragem definida. Quanto maior a taxa de amostragem, melhor será a qualidade do sinal discretizado, pois deste modo teremos um maior número de informações sobre o sinal de entrada por tempo. Entretanto, nossa sensibilidade auditiva tem um limite aproximado de 20000 Hz, portanto, não nos interessa amostrar componentes de freqüência acima deste valor.

Como estamos levando em conta a placa de som como o conversor analógico/digital, podemos então esperar por um pulso elétrico representado por uma onda longitudinal, e não uma onda transversal que é o tipo de onda que o som realmente representa. Portanto, a placa de som fará amostragem desse pulso elétrico e assim podemos entender que o valor de cada número que forma a seqüência discretizada representará a amplitude do sinal de entrada em um determinado instante de tempo.

No processo de reconstrução da onda, os valores obtidos na amostragem serão usados. Levaremos em conta que a placa de som fará o papel de conversor digital/analógico, ou seja, a placa de som se encarregará de representar uma amostra como um valor para formar o pulso elétrico que chegará até uma caixa de som, por exemplo.

A quantização definirá o tamanho reservado para cada célula de armazenamento de uma amostra. Como estamos trabalhando com armazenamento digital, esse tamanho é a quantidade de bits que irá compor o número binário que representa cada amostra. A quantidade geralmente usada é de 8 ou 16 bits. Esse parâmetro representa a faixa de números que podem ser usados para fazer a aproximação da amplitude do sinal de entrada. Dessa maneira, quanto maior o número de bits usado para a quantização, melhor será a aproximação obtida, e maior será o espaço necessário para o armazenamento do arquivo de áudio gerado.

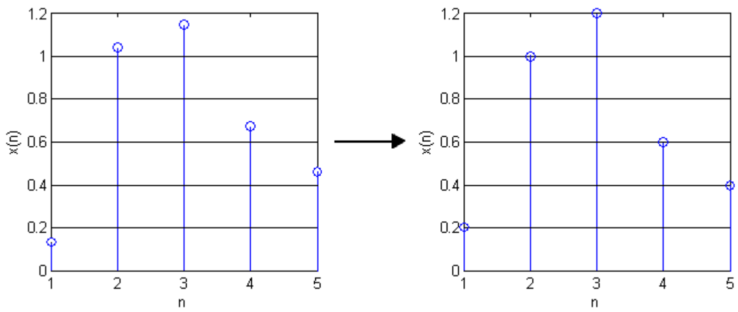

Figura 18: Arredondamento dos valores da seqüência original

A Figura 18 mostra os arredondamentos causados pela quantização. Utilizando, por exemplo, 16 bits para representação de uma amostra, teremos uma faixa de 65.536 valores diferentes. Esses valores representam a posição de cada amostra em relação ao eixo vertical da onda longitudinal que representa o sinal de áudio, ou seja, representa a amplitude.

3.2 Filtro Digital com Resposta ao Impulso Finita

Como vimos na Seção 2.3.4, podemos projetar um filtro de acordo com nossas necessidades. Esta seção apresentará formas de gerar um filtro computacionalmente.

3.2.1 Estimativa da Ordem do Filtro

Estimar a ordem do filtro significa calcular uma aproximação para o tamanho da resposta ao impulso usada para filtragem. Para um projeto de um filtro, muitos autores têm fórmulas avançadas para a estimativa do valor mínimo de um filtro de tamanho N pelas especificações do filtro digital: a banda de passagem com freqüência angular, a banda bloqueada com freqüência angular, a variação da ondulação na banda passante e a variação da ondulação na banda bloqueada.

Uma fórmula simples desenvolvida por *Kaiser* é dada por [Mit98]

$$N \cong \frac{-20\log 10(\sqrt{\delta_p\delta_b}) - 13}{14.6(\omega_p - \omega_b)/2\pi}. \tag{3.1}$$

Podemos notar por esta fórmula que a ordem de um filtro é inversamente proporcional ao tamamho da banda de transição. Desta maneira, quanto menor a banda de transição, maior será o tamanho do filtro, e melhor será a aproximação desse filtro.

Entretanto, a fórmula 3.1 não nos dá uma boa estimativa no caso de filtros com uma banda de passagem moderadamente grande, ou muito pequena. Para uma estimativa de um filtro com uma banda de passagem grande usamos [Mit98]

$$N \cong \frac{-20\log 10(\sqrt{\delta_p}) + 0.22}{(\omega_p - \omega_b)/2\pi}. \tag{3.2}$$

E por outro lado, em um filtro com banda de passagem pequena, a ondulação da banda bloqueada predomina, portanto usamos [Mit98]

$$N \cong \frac{-20\log 10(\sqrt{\delta_b}) + 5.94}{27(\omega_p - \omega_b)/2\pi}. \tag{3.3}$$

3.2.2 Aproximação pela Função *Sinc*

Vimos em 2.3.4 que a resposta ao impulso de um filtro ideal possui tamanho infinito. Desta maneira, devemos utilizar uma resposta ao impulso aproximada para projetar o filtro desejado. Uma forma simples de se fazer isso é utilizar a função *sinc* com termos truncados e posteriormente usar alguma função *janela* para otimizar a resposta do filtro.

Como podemos ver na Equação 2.18, a função *sinc* possui suporte infinito e não é absolutamente somável, desta maneira, essa resposta ao impulso não pode ser implementada. Simplesmente zerando os coeficientes dessa resposta ao impulso que estão fora do intervalo $-M \leq n \leq M$, obtemos uma aproximação de tamanho $N = 2M+1$, que quando deslocada para a direita nos dá os coeficientes de um passa-baixa

$$h_{PB}[n] = \begin{cases} \frac{\omega_c}{\pi}, & \text{para } n = 0, \\ \frac{\text{sen}(\omega_c\cdot(n-M))}{\pi\cdot(n-M)}, & 0 \leq n \leq N-1, \\ 0, & \text{senão} \end{cases} \tag{3.4}$$

Os coeficientes da resposta ao impulso de uma aproximação de um passa-alta ideal é dado por

$$h_{PA}[n] = \begin{cases} 1 - \frac{\omega_c}{\pi}, & \text{para } n = 0, \\ -\frac{\text{sen}(\omega_c(n-M))}{\pi(n-M)}, & \text{para } 0 \leq n \leq N-1, \\ 0, & \text{senão} \end{cases} \tag{3.5}$$

E finalmente, os coeficientes da resposta ao impulso de uma aproximação de um passa-faixa ideal é dado por

$$h_{PF}[n] = \begin{cases} \frac{\omega_{c2}}{\pi} - \frac{\omega_{c1}}{\pi}, & \text{para } n = 0 \\ \frac{\text{sen}(\omega_{c2}\cdot(n-M)) - \text{sen}(\omega_{c1}\cdot(n-M))}{\pi\cdot(n-M)}, & \text{para } 0 \leq n \leq N-1, \\ 0, & \text{senão} \end{cases} \tag{3.6}$$

Esses coeficientes gerados por trucamento resultam em ondulações na resposta em magnitude do filtro gerado. Esta conseqüência é chamada de Fenômeno de Gibbs [Che79]. Aumentando a ordem do filtro podemos minimizar os efeitos deste fenômeno.

3.2.3 Melhoramentos por Funções Janela Fixas

Quando geramos uma resposta ao impulso com base na função *sinc*, obtendo apenas $N+1$ termos, onde N é a ordem do filtro, estamos implicitamente utilizando a janela retangular. A idéia da janela retangular é exatamente truncar os termos da função *sinc*, assim, o filtro de resposta ao impulso finita obtido por truncamento pode ser alternativamente expresso como

$$h_t[n] = h_d[n] \cdot w[n],$$

onde h_d é a resposta ao impulso ideal e h_t é a resposta ao impulso truncada. Podemos definir a janela retangular como

$$w_R[n] = \begin{cases} 1 & \text{se } -M \leq n \leq M, \\ 0 & \text{senão.} \end{cases} \tag{3.7}$$

A janela retangular causa uma transição acentuada entre aos valores que estão fora do intervalo $-M \leq n \leq M$ e o que estão dentro do intervalo. Esta é a razão do aparecimento do Fenômeno de Gibbs na resposta em magnitude do filtro que usa este tipo de janela. Para a redução dos efeitos do Fenômeno de Gibbs podemos utilizar outros tipos de janelas. Algumas janelas utilizadas são

$$w_{Hann} = \frac{1}{2}\left[1 + \left(\frac{2\pi n}{2M+1}\right)\right], \quad -M \leq n \leq M, \tag{3.8}$$

$$w_{Hamming} = 0.54 + 0.46\left(\frac{2\pi n}{2M+1}\right), \quad -M \leq n \leq M, \tag{3.9}$$

$$w_{Blackman} = 0.42 + 0.5\left(\frac{2\pi n}{2M+1}\right) + 0.08\left(\frac{4\pi n}{2M+1}\right), \quad -M \leq n \leq M. \tag{3.10}$$

O espectro de magnitude de cada janela é caracterizado por um grande lóbulo principal centrado em $\omega = 0$ seguido de séries de lóbulos laterais com amplitute decrescentes. Os dois parâmetros que demostram a performance de uma janela em um filtro de resposta ao impulso finita é a largura do lóbulo principal e o nível do lóbulo lateral. Podemos observar a resposta em amplitude dessas janelas na Figura 19.

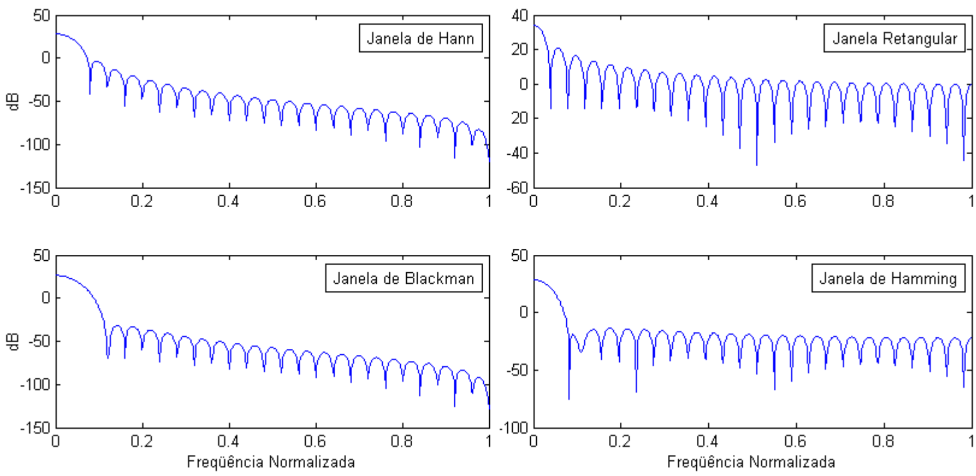

Figura 19: Resposta em amplitude das funções janela fixas

3.2.4 Melhoramentos por Funções Janela Ajustáveis

A ondulação, que varia por δ, como visto na Figura 14, dos filtros projetados usando alguma das funções janela fixas possui δ fixo. Algumas janelas foram desenvolvidas para prover controle sobre δ. Para isso, é adicionado um novo parâmetro à janela [Mit98].

A janela de Dolph-Chebyshev de tamanho $2M+1$ é definida por

$$w_{DC} = \frac{1}{2M+1}\left[\frac{1}{\gamma} + 2\sum_{k=1}^{M} T_k\left(\beta \cos\frac{k\pi}{2M+1}\right)\cos\frac{2nk\pi}{2M+1}\right], \quad -M \leq n \leq M, \tag{3.11}$$

onde γ é a amplitude do lóbulo lateral relativo como uma fração,

$$\gamma = \frac{\text{amplitude do lóbulo lateral}}{\text{amplitude do lóbulo principal}}, \tag{3.12}$$

$$\beta = \cosh\left(\frac{1}{2M}\cosh^{-1}\frac{1}{\gamma}\right), \tag{3.13}$$

e T_l é o polinômio de Chebyshev de ordem l que é definido por

$$T_l(x) = \begin{cases} \cos(l\cos^{-1}x), & \text{para } |x| \leq 1, \\ \cosh(l\cosh^{-1}x), & \text{para } |x| > 1, \end{cases} \tag{3.14}$$

A janela acima pode ser projetada com qualquer nível de lóbulo lateral relativo específico, e como no caso de outras janelas, a largura de seu lóbulo principal pode ser ajustada escolhendo o tamanho apropriado.

A janela ajustável mais usada é a janela de *Kaiser* dada por

$$w_{Kaiser} = \frac{I_0\beta\sqrt{1-(n/M)^2}}{I_0(\beta)}, \quad -M \leq n \leq M, \tag{3.15}$$

onde β é um parâmetro ajustável e $I_0(u)$ é a função de Bessel de ordem zero modificada que pode ser expressa em uma forma de série de potências

$$I_0 = 1 + \sum_{r=1}^{\infty}\left[\frac{(u/2)^r}{r!}\right]^2 \tag{3.16}$$

que é positiva para todos valores reais de u. Na prática, é suficiente pegar apenas os primeiros 20 termos do somatório da Equação 3.16 para obter uma aproximação razoável de $I_0(u)$.

O parâmetro β controla a atenuação mínima α_s, como por exemplo, o pico da ondulação δ, na banda rejeitada da resposta do filtro que usa janela.

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7), & \text{para } \alpha_s > 21, \\ 0.5842(\alpha_s - 21)^0.4 + 0.07886(\alpha_s - 21), & \text{para } 21 \leq \alpha_s \leq 50, \\ 0, & \text{para } \alpha_s < 21. \end{cases} \tag{3.17}$$

O tamanho do filtro é estimado pela fórmula

$$N \approx \begin{cases} \frac{\alpha_s - 7.95}{14.36\Delta f} + 1, & \text{para } \alpha_s > 21, \\ \frac{0.9222}{\Delta f} + 1 & \text{para } \alpha_s \leq 21. \end{cases} \tag{3.18}$$

Devemos notar que a janela de Kaiser não provê controle independente sobre δ_p da banda de passagem. Entretanto, na prática, δ_p é aproximadamente igual a δ_s. Podemos observar a resposta em ganho dessas janelas na Figura 20, onde para a janela de Dolph-Chebyshev o nível do lóbulo lateral relativo é de 50 dB.

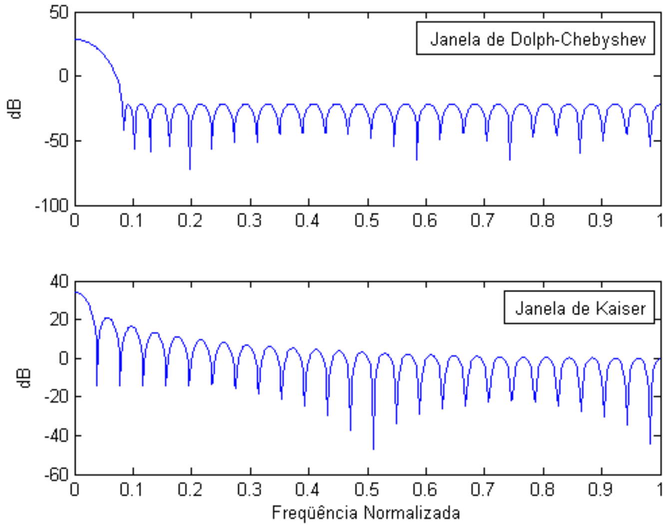

Figura 20: Resposta em Ganho das funções janela ajustáveis

3.3 Desenvolvimento do Equalizador

Para o desenvolvimento da aplicação foram criadas algumas classes utilizando do paradigma da Orientação a Objetos. Cada classe é uma unidade que define os objetos através de métodos e atributos.

Existe uma classe chamada `Equalizador` que interage diretamente com a classe `Filtro`. Alguns dos métodos da classe que se referem ao gerenciamento de arquivos de áudio foram extraidos do projeto que pode ser visto em [Gag05]. Quando um objeto da classe `Equalizador` é criado, ainda no construtor da classe, um número filtros também é criado, de acordo com o número de bandas que o equalizador possuirá. Cada filtro é uma instância da classe `Filtro`.

Os atributos da classe `Filtro` são:

- `freqMin`: relativo a freqüência mínima da faixa de corte do filtro.
- `freqMax`: relativo a freqüência máxima da faixa de corte do filtro.
- `tamanho`: tamanho estimado do filtro, como visto na Seção 3.2.1.
- `respImpulso`: vetor que armazena os coeficientes da resposta ao impulso de cada filtro.

- `origem`: relativo a posição que será usada como centro do vetor da resposta ao impulso.
- `tAmostragem`: informa a taxa de amostragem usada pelo arquivo de áudio usado. Necessário para o cálculo da freqüência de corte.
- `janela`: vetor que armazena os coeficientes gerados pelas Funções Janelas.

Os métodos da classe `Filtro` são:

- `calculaOrdem()`: calcula a ordem do filtro por estimativa.
- `geraSinc()`: calcula os coeficientes da função *sinc* e retorna $n+1$ coeficientes, onde n é a ordem do filtro.
- `convolucao()`: realiza a convolução, ou seja, filtra o sinal de entrada.

A classe `Janela` simplesmente encapsula os métodos envolvidos no cálculo dos coeficientes das funções janela. A decisão de criar uma classe para esta estrutura acontece pela necessidade de usar métodos para manipulações matemáticas.

Resumindo, a classe `Equalizador` permite que o número de bandas do equalizador seja escolhido, assim como a faixa de freqüência que cada banda deverá atuar. Para obter o resultado final da equalização basta somar a saída de cada filtro.

Durante o processo de convolução o ganho referente a cada faixa de freqüência é aplicado, ou seja, logo após o cálculo do valor de cada amostra o ganho é aplicado sobre aquele valor e só depois disso é que ele é atribuido à estrutura de saída.

3.4 Manipulação dos Dados

Como o objetivo da aplicação é realizar a equalização em tempo real devemos dividir as amostras do arquivo de áudio em blocos, chamados *buffers*. Basicamente devemos ter um fluxo de *buffers* para que possamos aplicar a convolução nesses *buffers* e aplicar o ganho de cada faixa de freqüência.

3.4.1 *Buffers*

Se carregassemos todo arquivo Wave na memória e fizessemos o computador executá-lo por inteiro escutariamos o sinal de áudio contido nele normalmente. Porém não poderiamos modificar algumas de suas características durante sua execução e portanto seu processamento sempre seria *off-line*. A importância de alterarmos o sinal *on-line* existe pela vantagem de ajuste e resposta instantânea. Pensando nisso é que toda a aplicação é voltada para *buffers*.

Para obter os dados, precisamos primeiro checar se o arquivo em questão está no formato Wave válido comparando os dados do cabeçalho. Se temos um arquivo Wave válido, então podemos carregar os dados das amostras na memória.

A medida que um *buffer* é formado, devemos ter guardado na memória um índice que indica até que amostra do arquivo já foi consumida pelo processo. Desta maneira, quando um novo *buffer* for solicitado para o processamento, as amostras que serão copiadas para esse novo *buffer* serão de $kn + 1$, até $(kn + 1) + n$, onde n é o tamanho dos *buffers* e k o número de *buffers* já consumidos. A Figura 21 ilustra a divisão das amostras em *buffers*.

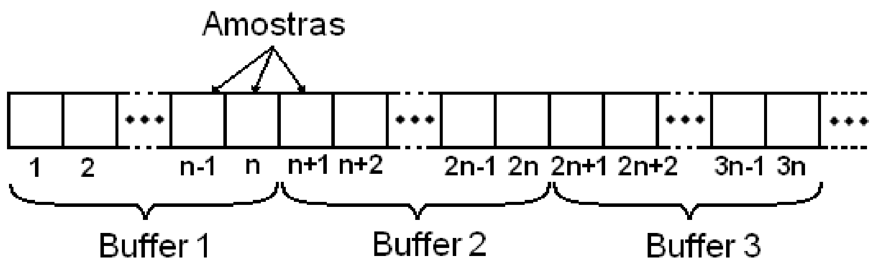

Figura 21: Divisão das amostras do arquivo de áudio

Como temos *buffers* de tamanho pré-determinados, não levando em conta o número total de amostras do arquivo, muito provavelmente o último bloco não terá n amostras. Para previnir que possíveis erros possam ocorrer com esta situação, basta termos acesso ao tamanho total do arquivo em relação ao número de amostras. Assim, quando o número de amostras restantes for menor que n, basta criar um *buffer* com o tamanho de amostras restantes.

A classe `Equalizador` carrega o arquivo de áudio, cria os *buffers*, executa o áudio, pausa a execução, e para a execução. A classe conta com um conjunto de atributos que guardam as informações necessárias para executar tais métodos. Para que o áudio seja executado, as amostras são enviadas à placa de som através de chamadas de procedimento de uma biblioteca multimídia do sistema.

3.4.2 Fluxo de *Buffers*

O fluxo de *buffers* deve ser gerenciado para que estes sejam executado na ordem e sem interrupções na execução do áudio final. Para isso *threads* são criadas através da API do *Windows* para melhor desempenho do fluxo de dados até a execução das amostras. Cada *thread* contém um conjunto de *buffers* que devem ser executados em ordem, pois o arquivo de áudio é contínuo e ordenado. Podemos observar o processo na Figura 22.

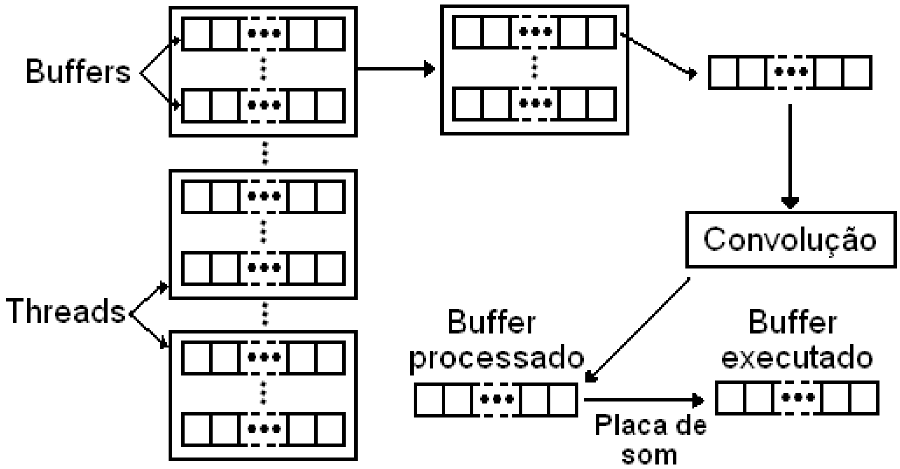

Figura 22: Fluxo de *buffers*

Porém, antes dos *buffers* serem executados, eles precisam ser processados, ou seja, precisamos aplicar o filtro sobre aquele trecho do arquivo de áudio. Neste momento já possuimos os filtros projetados e suas repectivas resposta ao impulso geradas.

Como para aplicar a convolução precisamos de amostras passadas e futuras, precisamos nos preocupar com os *buffers* passado e futuro. A solução adotada pela aplicação é gerar um *buffer* com amostras passadas ao *buffer* atual e um outro *buffer* para as amostras futuras. Para o caso do ínicio e final do arquivo podemos imaginar que as amostras que não existem, nesse caso, têm valor zero.

Os *buffers* filtrados ainda não estão preparados para serem executados. O projeto da aplicação permite que o equalizador tenha m bandas. Portanto, cada *buffer* necessita ser filtrado m vezes, cada vez correnpondendo a uma banda, ou seja, um intervalo de freqüência. No final, teremos os m *buffers* processados, somados em um novo *buffer* que contém o resultado final e este é executado.

4 *Aplicação*

Este capítulo trata da implementação do Equalizador Digital e de sua aplicação para um sinal de áudio, especificamente em um arquivo no formato Wave. Também será apresentado de que forma os dados do arquivo de áudio digital são enviados a placa de som para serem executados.

4.1 O Formato Wave

O Wave é um formato padrão da Microsoft e IBM para armazenamento de áudio em computadores e é uma variação do método de formatação de fluxo de bits RIFF (*Resource Interchange File Format*), padrão definido pela empresa Eletronic Arts, para armazenar dados em blocos chamados *chunks*. O Wave é o formato padrão utilizado para a gravação de CD's de áudio, porém mascarado sob a forma "cda". Os CDs de áudio que estão no mercado utilizam 44,1 KHz para a taxa de amostragem e 16 bits de quantização como padrão.

O arquivo Wave em sua forma mais simples contem áudio não comprimido no formado PCM (*Pulse-code modulation*) que nada mais é que o sinal de áudio discretizado, porém também pode ser encontrado áudio com compressão. O Wave contendo PCM é muito usado para edição de áudio por softwares devido a fácil manipulação que ele provê. É por este motivo que é importante descrever alguns dos detalhes deste formato, pois ele foi utilizado neste trabalho como o sinal de áudio digital de entrada.

Como já dito, o arquivo Wave é dividido em blocos, chamados de *chunks*, e os primeiros 8 bytes do arquivo são seu cabeçalho e identificam o arquivo como sendo do tipo RIFF e mostra qual o tamanho do arquivo total, tirando o cabeçalho. Esta identificação é feita por um ID contendo a seqüência de caracteres que correspondem à RIFF. Os próximos 4 bytes indicam que o arquivo é do tipo "WAVE", como podemos ver na Tabela 2. A partir disso temos outros tipos de *chunks*.

Tabela 2: Estrutura de um arquivo do tipo RIFF encapsulando Wave

Tamanho	Descrição	Valor
4 bytes	ID do Chunk	"RIFF"
4 bytes	Tamanho dos dados do Chunk	(tamanho do arquivo) - 8
4 bytes	Tipo de RIFF	"WAVE"
Chunks Wave		

Tabela 3: Estrutura do Format Chunk

Tamanho	Descrição	Valor
4 bytes	ID do Chunk	"fmt"
4 bytes	Tamanho dos Dados do Chunk	16 + Bytes de Formato Extra
2 bytes	Código de Compressão	1 (PCM)
2 bytes	Número de Canais	1,2 ou mais
4 bytes	Freqüência de amostragem	amostras por segundo
4 bytes	Média de Bytes por segundo	Freq. de Amostragem x Bloco Alinhado
2 bytes	Bloco Alinhado	Bits por amostra / 8 x N. de Canais
2 bytes	Bits Significantes por Amostra	Normalmente 8 ou 16
2 bytes	Bytes de Formato Extra	varia
Bytes de Formato Extra		

Para um arquivo RIFF encapsulando Wave, existem diferentes tipos de *chunks*, porém dois deles são obrigatórios, são o *Format Chunk* e o *Data Chunk*. O *Format Chunk* contém informação sobre como o áudio é armazenado e como ele deve ser tocado. O *Format Chunk* contém informação sobre como o áudio é armazenado e como ele deve ser tocado. Contém dados como tipo de compressão, número de canais, taxa de amostragem, tamanho das amostras e outras características. A tabela 3 mostra a estrutura do *Format Chunk*

As amostras ficam armazenadas no *Data Chunk* como mostra a Tabela 4. Em arquivos que usam amostras com 8 bits são usados apenas valores positivos para a representação da amplitude, porém, inexplicavelmente, para arquivos que usam amostras de 16 bits esses valores variam entre positivos e negativos. Além disso, as amostras são armazenadas usando o padrão *Little Endian*, onde primeiro temos o byte menos significativo mais à esquerda e em seguida os mais significativos à direita. Esse padrão é utilizado para otimizar o processamento em processadores que seguem o padrão Intel.

Para um trecho de áudio com apenas um canal, as amostras são armazenadas seqüencialmente. No caso de multi-canais as amostras são armazenadas de forma intercalada, ou seja, para um sinal de áudio com dois canais (estéreo), a primeira amostra corresponde ao

Tabela 4: Estrutura de um Data Chunk

Tamanho	Tipo	Descrição	Valor
4 bytes	char[4]	ID do Chunk	“DATA”
4 bytes	dword	tamanho do Chunk	varia
dados do áudio			

canal esquerdo e a seguinte corresponde ao canal direito, e assim por diante. A Figura 23 mostra toda a divisão em blocos de um arquivo Wave. Mais detalhes sobre o formato Wave e outros formatos pode ser encontrado em [Ser02].

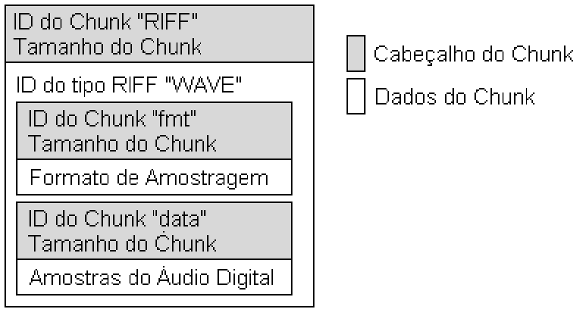

Figura 23: Blocos do Arquivo Wave

4.2 Interface

A interface gráfica da aplicação simula os equalizadores gráficos que os aparelhos de áudio possuiam antigamente, como a maioria dos *softwares* fazem hoje. Desta maneira, cada faixa de freqüência, ou seja, cada banda, possui um *slider* que controla o ganho dado àquela banda. A Figura 24 apresenta a interface. Esta interface é uma adaptação de uma existente em [Gag05].

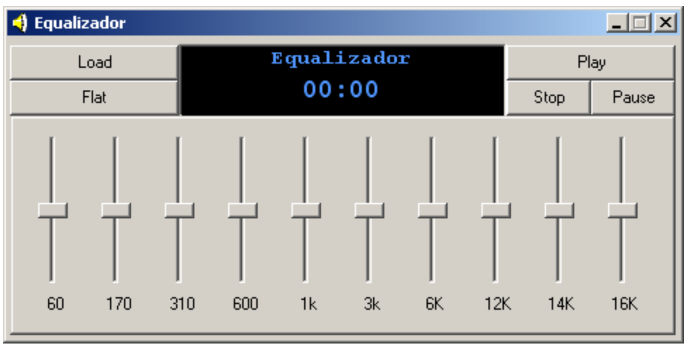

Figura 24: Interface Gráfica da aplicação

O botão *Load* tem como função carregar o arquivo de áudio no formato *Wave.* Com o arquivo já carregado, basta apertar o botão *Play* e o arquivo será executado. O botão *Stop* pára a música e o botão *Pause* pausa a música que pode voltar a ser executada a partir do ponto em que parou. O botão *Flat* volta com todos os *sliders* modificados para a posição padrão, que é no centro.

Deslizando algum *slider* para cima, aplica-se um ganho de energia na faixa de freqüência centrada na freqüência específica mostrada abaixo do slider. Deslizando o *slider* para baixo promovemos perda de energia naquela faixa.

As alterações promovidas no domínio da freqüência do sinal de áudio digital podem ser percebidas on-line, o que permite uma facilidade para o usuário que procura um ajuste que atenda ao seu gosto.

4.3 Detalhes da Aplicação

Para a implementação deste trabalho foi utilizado a lingüagem de programação C++. Esta lingüagem permite um acesso eficiente a cada amostra armazenada no *buffer* pela utilização de ponteiros. Utilizando ponteiros atribuidos à diferentes tipos de dados podemos acessar amostras que possuem tamanhos diferentes de célula de quantização, como os 8 ou 16 bits comumente usado pelo formato Wave.

Após a implementação da função que realizava a convolução, observou-se uma possibilidade de otimização para a redução de uso do processador. Essa otimização levaria à possibilidade de utilizar respostas ao impulso maiores, pois com a redução do uso do processador, podemos agora realizar maior quantidade de operações para obter um resultado melhor.

Tal otimização consistia em utilizar os valores para os coeficientes da resposta ao impulso calculada representados com ponto fixo. Utilizando as equações mostradas na Seção 3.2.2, obteríamos valores entre 0 e 1, ou seja, valores que normalmente seriam representados no computador por ponto flutuante. Um valor representado por ponto flutuante normalmente necessita de um espaço de memória maior, ou seja, um maior número de bits para sua representação, e assim, para operarmos com um valor representado por ponto flutuante necessitamos de maior esforço computacional.

Portanto, para utilizar ponto fixo, durante o cálculo dos coeficientes da resposta ao impulso, foi utilizado o deslocamento de bits provido pelo C++ que tem o efeito de uma multiplicação por um inteiro. Desta forma podemos representar o coeficiente da resposta ao impulso por um número inteiro e as multiplicações realizadas pela convolução são realizadas mais rapidamente. No final do processo, para cada amostra resultante da convolução, basta utilizar o deslocamento de bits novamente, porém, desta vez, com efeito de divisão.

Outro detalhe importante que deve ser comentado é sobre o valor máximo de cada amostra. Para um arquivo Wave de 8 bits, os valores possíveis para uma amostra variam entre 0 e 255, desta forma, devemos tomar o valor 128 como o zero para esta representação. Para um arquivo Wave utilizando 16 bits de quantização os valores das amostras variam entre -32768 e 32767, e assim, ao obtermos as amostras resultantes da convolução, devemos sempre limitá-las a este valores para evitar problemas de *overflow*.

4.4 Resultados

Nesta seção, podemos ver os resultados obtidos com a aplicação implementada. Inicialmente os testes focavam em verificar se a aplicação funcionava com apenas um filtro. Sendo o filtro a base da aplicação, era necessário que um filtro funcionasse para que depois houvesse progressão no desenvolvimento da aplicação.

Utilizando o *software Cool Edit Pro 2.0* para capturar os resultados obtidos, podemos claramente ver os resultados da aplicação de um filtro no sinal de áudio. A Figura 25 demostra uma visão espectral no tempo de uma amostra de áudio estéreo sem nenhuma alteração. Os dois espectros são decorrentes da utilização de um arquivo de áudio estéreo (dois canais). A Figura 26 demostra os resultados obtidos para 4 filtros diferentes com a mesma faixa de passagem, entre 5300 Hz e 9200 Hz, aplicados na amostra da Figura 25. As Figuras 27 e 28 mostram uma análise do espectro para os mesmos filtros, porém no mesmo instante do tempo relativo ao mesmo sinal de áudio.

Já a Figura 29 demostra os resultados obtidos para as janelas de Hann e Blackman, também com faixa de passagem entre 5300 Hz e 9200 Hz.

Como era de se esperar, pela Figura 26, observamos que a janela retangular permite que freqüências que estão fora da faixa de passagem desejada passem com um ganho maior do que quando se usa a janela de Hamming. Em contrapartida, quando utilizamos um filtro de ordem menor, a janela de Hamming cria uma larga banda de transição, deixando que freqüências vizinhas a faixa de passagem apareçam no resultado final.

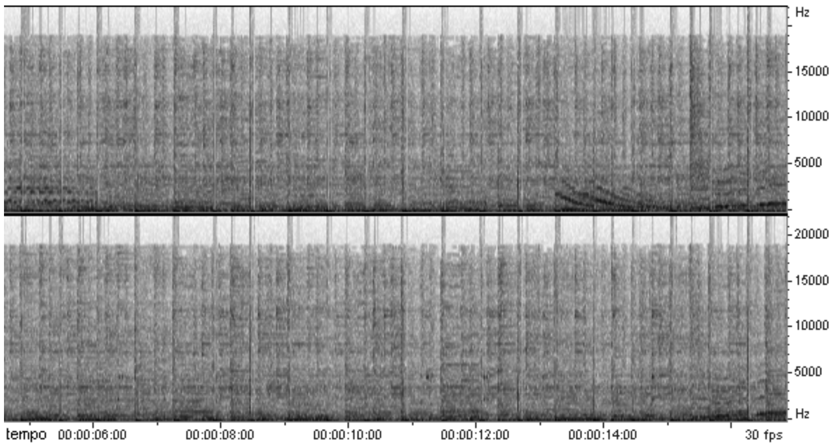

Figura 25: Espectro de amostra de áudio digital sem alterações

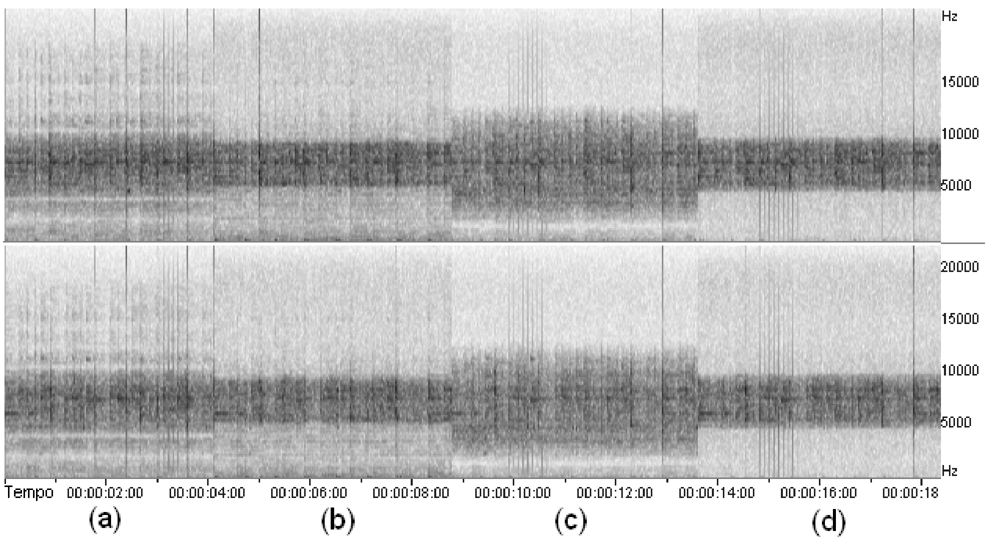

Figura 26: Filtros utilizando (a) janela Retangular com tamanho 21, (b) janela Retangular com tamanho 101, (c) janela de Hamming com tamanho 21 e (d)Filtro com janela de Hamming com tamanho 101

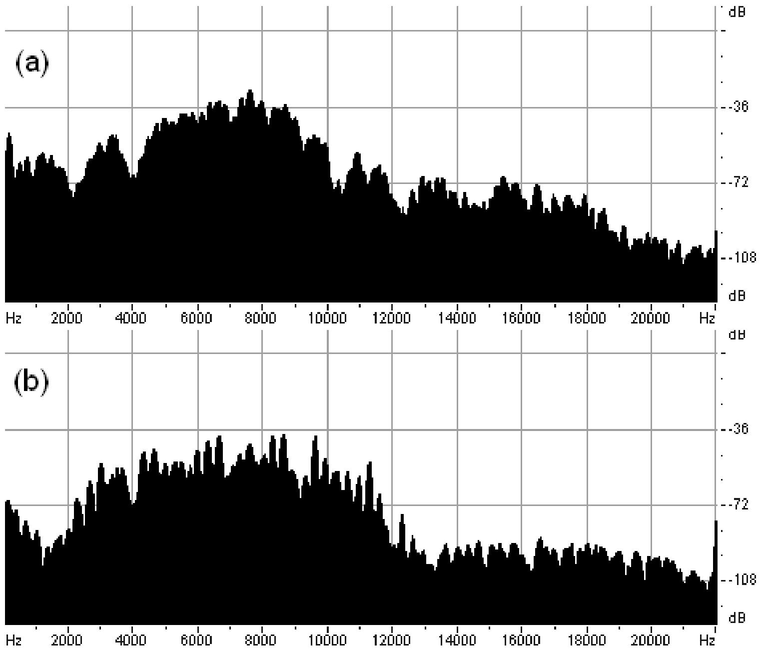

Figura 27: (a)Janela Retangular com tamanho 21 (b) Janela de Hamming com tamanho 21

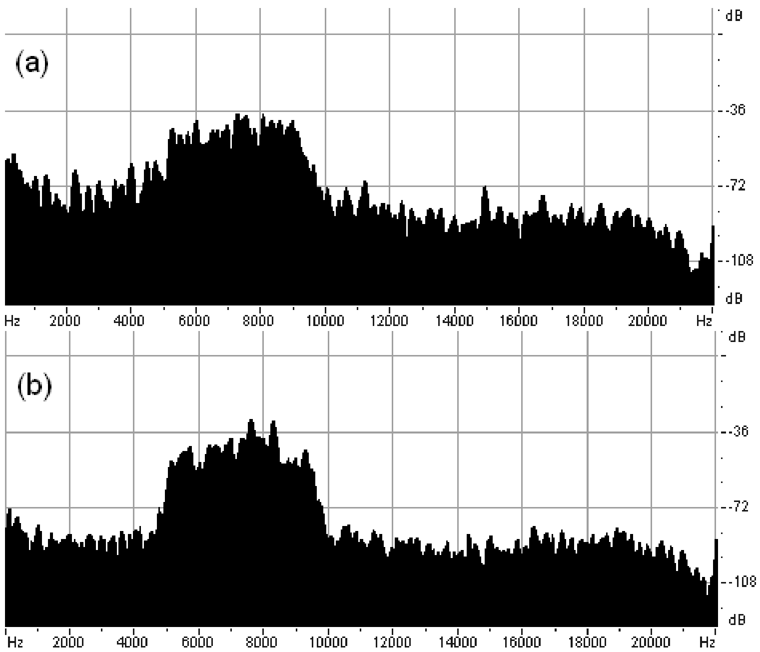

Figura 28: (a) Janela Retângular com tamanho 101 (b) Janela de Hamming com tamanho 101

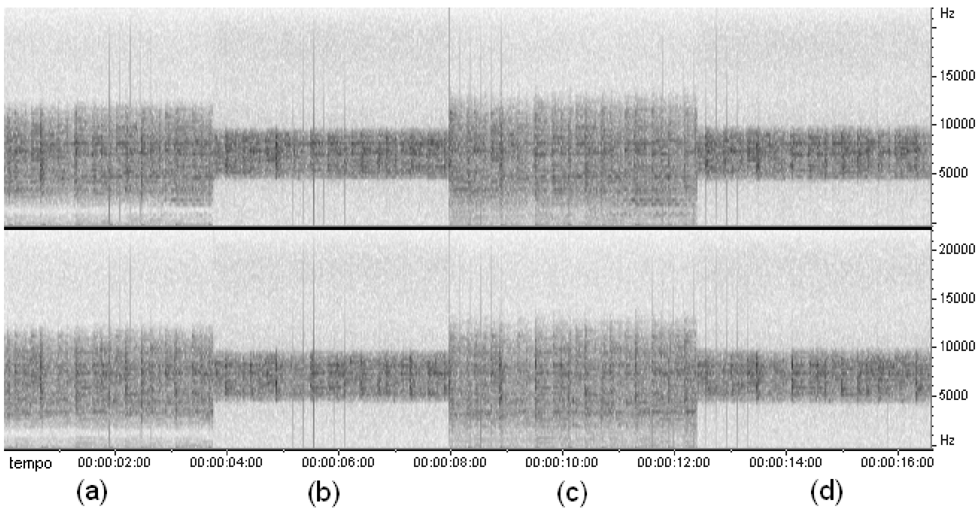

Figura 29: Filtros utilizando (a) janela de Hann com tamanho 21, (b) janela de Hann com tamanho 101, (c) janela de Blackman com tamanho 21 e (d)Filtro com janela de Blackman com tamanho 101

5 Conclusão

Através deste trabalho, podemos ver como o meio digital pode trazer tantos benefícios. Hoje, podemos projetar filtros em computadores por meio de *softwares*, não mais necessitando o uso de *hardwares* dedicados. O sinal discretizado pode ser facilmente processado, não só pelo uso de filtros digitais, e existem vários *softwares* que realizam esta tarefa de forma eficiente.

Observamos que podemos utilizar técnicas para obtenção de respostas ao impulso finitas para construir filtros. Pelo uso de funções janela obtemos um filtro que age mais eficientemente para as faixas de freqüência em que ele deve atenuar.

Uma observação importante a ser feita é sobre nossa percepção em relação a sinais de áudio. Como as alterações realizadas no domínio da freqüência de um sinal de áudio podem ser bem perceptíveis para nós, podemos imaginar então o ouvido como um analizador de freqüências dos sinais de áudio que recebemos.

Pelo estudo feito até aqui, podemos observar outras aplicações que têm por base parte dos conceitos introduzidos nesse trabalho, como por exemplo um analisador de espectro de um sinal, que pode ser implementado usando-se da transformada de Fourier. Outro exemplo vem da observação dos resultados. Podemos notar pela simples audição de uma música utilizando um equalizador que temos menos sensibilidade para determinadas faixas de freqüência. Esse tipo de observação é levada em conta no desenvolvimento de algoritmos de compactação de sinais de áudio. Estes são exemplos de possíveis trabalhos ainda nesta área.

APÊNDICE A – Números Complexos

É interessante passar por conceitos introduzidos pelo conjunto dos números complexos antes de entrar em detalhes sobre senóides e principalmente sobre senóides complexas. Para extrair as raízes do polinômio $p(x) = x^2 - 5x + 6$ é usada a seguinte fórmula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

e neste caso obtém-se $x_1 = 2$ e $x_2 = 3$. Porém, na tentativa de obter as raízes de $p(x) = x^2 + 4$ obtém-se $x = \pm\sqrt{-4} = \pm 2\sqrt{-1}$, ou seja, não existe raiz real para este polinômio, o que significa que o gráfico da função não corta o eixo x. Portanto chamamos este tipo de raiz de raiz imaginária.

Podemos escrever a raiz de qualquer número negativo como $\sqrt{|c|}\sqrt{-1}$ e chamamos $\sqrt{-1}$ de i. Desta forma as raízes do polinómio anterior seriam $\pm 2i$ e poderia ser escrito na forma $p(x) = (x - 2i)(x + 2i)$. Portanto os números complexos são escritos na forma

$$z = x + iy \tag{A.1}$$

onde x e y são reais e $i^2 = -1$.

Todas as operações da álgebra que se aplicam aos reais, também são aplicáveis aos complexos. Além disso, podemos dividir um número complexo $z = x + iy$ em uma componente real e uma imaginária da seguinte forma

$$\mathrm{re}\{z\} = x$$

$$\mathrm{im}\{z\} = y$$

Podemos notar que os números reais formam um subconjunto do conjunto dos números complexos.

Um número complexo pode ser colocado em um plano levando sua parte real ao eixo x e a parte imaginária ao eixo y, assim representamos um par ordenado (x,y) e chamamos este plano de plano complexo. Na figura 30, podemos observar o plano complexo. É interessante comentar que um conjugado de um número complexo $z = x + iy$ é dado por $z = x - iy$ e esse conjugado representa, no plano complexo, uma reflexão no eixo dos reais.

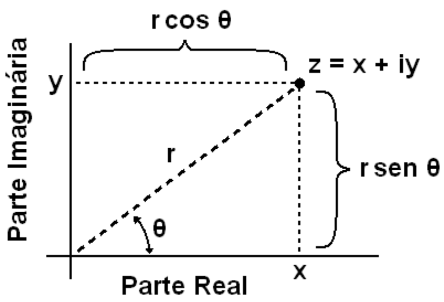

Figura 30: Plano Complexo

Existe outra forma interessante de expressar os números complexos que é em termos de suas coordenadas polares, como um par ordenado (r,θ), onde r é a distância da origem (0,0) ao ponto z e θ é o ângulo do número relativo ao eixo dos reais positivos. Do plano complexo tiramos que $x = r\cos\theta$ e $y = r\ \text{sen}\theta$, desta maneira a forma polar dos números complexos é dada por

$$z = r\cos\theta + i\ \text{sen}\theta \tag{A.2}$$

Usando a identidade de Euler que é definida por

$$e^{i\theta} = \cos\theta + i\ \text{sen}\theta \tag{A.3}$$

podemos facilmente obter uma outra forma de expressar um número complexo multiplicando a identidade de Euler por r e assim obtendo $re^{i\theta}$. Esta outra maneira de expressar a forma polar de um número complexo é bem interessante pela facilidade que ela proporciona em manipulações algébricas. O conjugado de $z = re^{i\theta}$ nesta forma será $z = re^{-i\theta}$.

APÊNDICE B – Raízes da Unidade

Utilizando a identidade de Euler definida na Equação (A.3) e fazendo $\theta = 2\pi$ teremos

$$e^{i2\pi} = \cos 2\pi + i \sin 2\pi = 1 + 0 = 1,$$

e sabemos que o mesmo acontece para $e^{i2\pi k}$, para todo k inteiro.

Fazendo $1^{1/N}$ tiramos a enésima raiz de 1. Desde que $e^{i2\pi k} = 1$, para todo k inteiro podemos escrever

$$1^{k/N} = e^{i2\pi k/N}, \quad k = 0, 1, 2, \ldots N - 1 \tag{B.1}$$

e assim obtemos as N raízes da unidade.

Para $N = 4$, obtemos o gráfico da solução apenas dividindo o círculo unitário em quatro partes iguais, a partir do primeiro ponto que será $z = 1$. Quando N é par, existirá um ponto $z = -1$, porém, se N for ímpar, não existirá esse ponto.

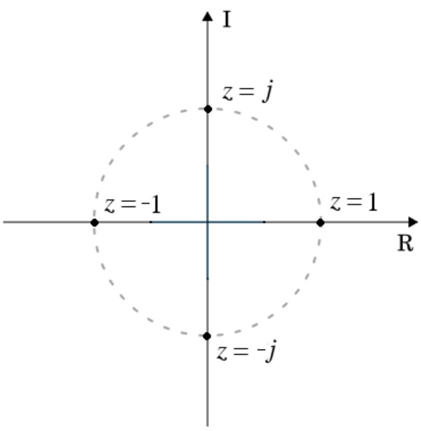

Figura 31: N raízes

APÊNDICE C – Classificação de Seqüências

Se todos os valores de $x[n]$ são reais, então $x[n]$ é uma seqüência real. Se algum valor de $x[n]$ for complexo, então $x[n]$ é uma seqüência complexa. Podemos separar a parte real e a parte imaginária da seqüência $x[n]$, assim podemos escrever a seqüência complexa como

$$x[n] = x\ \mathrm{re}[n] + x\ \mathrm{im}[n] \tag{C.1}$$

onde $x\ \mathrm{re}[n]$ é a parte real de $x[n]$ e $x\ \mathrm{im}[n]$ é a parte imaginaria de $x[n]$.

Podemos classificar seqüências de diferentes maneiras. Uma delas é baseada no tamanho da seqüência. Um sinal discreto no tempo pode ser representado por uma seqüência de tamanho finito ou infinito. Uma seqüência de tamanho finito é definida em um intervalo de tempo

$$N_1 \leq n \leq N_2$$

onde $N_1 > -\infty$ e $N_2 < +\infty$ $N_2 \geq N_1$. A tamanho da seqüência será $N = N_2 - N_1 + 1$. Uma seqüência discreta no tempo de tamanho N possui N amostras. Uma seqüência de tamanho finito pode ser considerada como uma seqüência de tamanho infinito atribuindo o valor zero às amostras fora da faixa definida pela seqüência.

Uma seqüência pode também ser definida por sua simetria. Uma seqüência $x[n]$ é conjugada simétrica se $x[n] = \overline{x}[n]$. Uma seqüência real conjugada simétrica é chamada seqüência par. Uma seqüência $x[n]$ é conjugada anti-simétrica se $x[n] = -\overline{x}[-n]$. Uma seqüência real conjugada anti-simétrica é chamada seqüência ímpar.

Existem outros tipo de classificação como a baseada na periodicidade que leva em conta sua periodicidade. Uma seqüência que satisfaz essa classificação é da forma:

$$x[n] = x[n + kN]$$

para todo n, onde N será o período que define a repetição da seqüência.

Uma seqüência é dita limitada se todas as amostras forem menor que um limite estabelecido e esse limite é menor que infinito. Uma seqüência $x[n]$ é absolutamente somável se

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty$$

Uma seqüência é dita ser quadrado somável se

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$$

Também é importante ressaltar algumas seqüências básicas que desempenham papéis importantes quando usamos a representação do sinal discreto no tempo. Geralmente expressamos uma seqüência em termos de alguma dessas seqüências básicas. Outra importância é a representação de uma classe de sistemas discretos no tempo em termos da resposta do sistema a certas seqüências básicas. Essa representação permite a computação da resposta do sistema discreto no tempo para arbitrários sinais discretos no tempo se esses podem ser expressados em termos de seqüências básicas.

A seqüência mais simples é a seqüência de amostra pontual, também chamada de impulso discreto ou impulso único. Essa seqüência possui grande importância e é uma das mais usadas. Ela é denotada por $\delta\,[n]$ e definida por

$$\delta[n] = \begin{cases} 1 & \text{se } n = 0, \\ 0 & \text{se } n \neq 0. \end{cases} \tag{C.2}$$

Outra seqüência básica importante é a seqüência de degrau unitário denotada por $\mu[n]$ e definida por:

$$\mu[n] = \begin{cases} 1 & \text{se } n \geq 0, \\ 0 & \text{se } n < 0. \end{cases} \tag{C.3}$$

Existe também um conjunto de seqüências básicas formado quando elevamos a amostra n de uma seqüência a potência também n de uma constante real ou complexa. Essas funções já foram discutidas na seção anterior, porém tratadas no espaço contínuo. No espaço discreto temos seqüências exponenciais que são da forma geral

$$x[n] = \alpha e^{sn}$$

que podemos expressar como

$$x[n] = [Ae^{\sigma n}][\cos(\omega n + \phi) + i \sin(\omega n + \phi)],$$

desta maneira obtemos uma seqüência senoidal complexa, onde σ, ω e ϕ são números reais.

Se escrevermos $x[n] = x\ \mathrm{re}^2[n] + ix\ \mathrm{im}^2[n]$ então:

$$x_{re}[n] = [Ae^{\sigma n}][\cos(\omega n + \phi)]$$

$$x_{im}[n] = [Ae^{\sigma n}][\sin(\omega n + \phi)].$$

Portanto, a parte real e imaginária de uma seqüência exponencial complexa são seqüências senoidais reais com amplitude constante, crescente ou decrescente , dependendo do valor de σ.

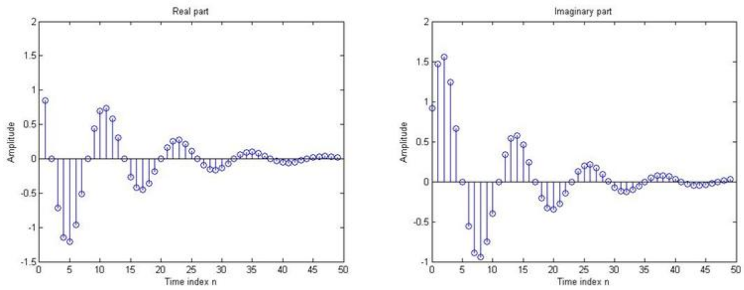

Figura 32: Parte real e imaginária de uma seqüência senoidal complexa

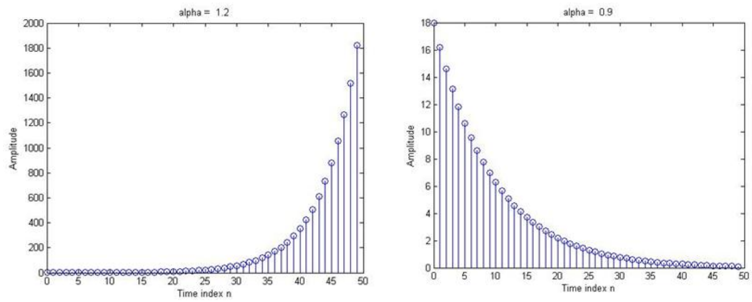

Figura 33: Seqüências Exponenciais decrescente e crescente

Como citado antes quando dividimos a senoide complexa em suas partes real e imaginária, existem também seqüências senoidais reais, que possuem amplitude constante e são da forma

$$A \sin(\omega n + \phi)$$

onde A é a amplitude, ω é a Freqüência Angular e ϕ é a fase inicial da senoide real, todos sendo reais. A seqüência senoidal real pode ser representada como

$$x[n] = x_f[n] + x_q[n],$$

onde $x_f[n]$ e $x_q[n]$ são respectivamente, os componentes em fase e em quadratura de $x[n]$.

Referências

[Ang01] David Howard Jamie Angus. *Acoustics and Psychoacoustics*. Focal Press, 2 edition, 2001.

[Ant93] Andreas Antoniou. *Digital Filters: Analysis, Design and Applications*. McGraw-Hill Companies, 1993.

[Che79] Chi-Tsong Chen. *One-Dimensional Digital Signal Processing*. Marcel Dekker, Inc., 1979.

[Dar76] Gobind Daryanani. *Principles of Active Network Synthesis and Design*. John Wiley and Sons Inc, 1976.

[Gag05] Gagah. *Equalizer (WIN32)*. http://www.cppfrance.com/codes/EQUALIZER-WIN32_34436.aspx, 2005.

[Lat04] B. P. Lathi. *Linear Systems and Signals*. Oxford University Press, 2 edition, 2004.

[Mit98] Sanjit Kumar Mitra. *Digital Signal Processing: A Computer Based Approach*. McGraw-Hill, 1998.

[Pie89] Allan D. Pierce. *Acoustics: An Introduction to Its Physical Principles and Applications*. Acoustical Society of Amer, 1989.

[Ser02] Fábio Luis Ferreira Serra. *Áudio Digital: a tecnologia aplicada à música e ao tratamento de som*. Ciência Moderna, 2002.

[Smi03] Julius O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, http://www.w3k.org/books/, 2003.