



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Código	Disciplina	Professor
ELE042	Processamento de Sinais	Hilton de Oliveira Mota

TRABALHO PRÁTICO

- Condições:
- Grupos de 3 alunos.
- Avaliação:
- Implementação básica – compressor de áudio: 15 pontos.
 - Bônus – escalonador de tempo com correção de tons: 5 pontos *extras*.
- Entregar:
- Documento em .pdf contendo identificação dos membros do grupo, descrição da solução, apresentação das imagens, resultados de simulações e análises de desempenho.
 - Scripts, códigos fontes e/ou executáveis implementados em Matlab, Python ou C/C++ que possam ser inspecionados e executados localmente.

Introdução

Um sinal de áudio digital é a versão amostrada de uma onda de pressão do ar que é medida e transformada em uma tensão ou corrente elétrica por um sensor de áudio (genericamente, um microfone). Como em outras situações, a digitalização propicia o uso de ferramentas de processamento muito mais elaboradas do que em domínio analógico, tais como compressão, filtragem, equalização, extração de características e reconhecimento da fala. O processamento digital possibilita, também, a geração artificial de sons, abrindo um grande leque de aplicações em síntese de música, fala, ruídos, efeitos, etc.

Um sistema digital de processamento de áudio possui uma estrutura similar à mostrada na Figura 1. O sinal é modelado como uma função de tempo contínuo $x_c(t)$ que é digitalizada por um conversor analógico-digital (ADC) com período de amostragem $T=1/f$, gerando o sinal discreto $x[n]$. Este é processado pelo sistema, gerando a saída $y[n]$ que é reconstruída por um conversor digital-analógico (DAC), resultando no sinal de tempo contínuo $y_c(t)$. O processamento é feito por meio de algoritmos implementados em algum sistema computacional tal como microcontroladores, processadores de sinais digitais (DSPs), FPGAs¹, ASICs², etc.

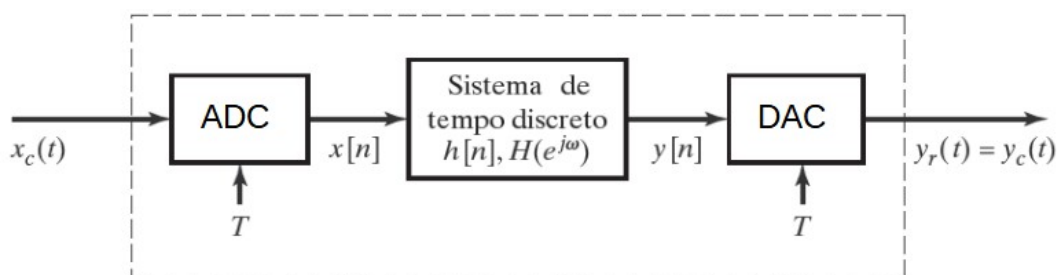


Figura 1: sistema de processamento de áudio.

Se o algoritmo não implicar alterações na frequência de amostragem, o DAC trabalhará com o mesmo período de amostragem $T=1/f_s$ do ADC, tal como mostrado na figura. Se, por qualquer razão, surgir a necessidade de utilizar frequências diferentes, o sinal sofrerá um efeito conhecido como “efeito Chimpunk”³,

¹ “Field-programmable Gate Arrays”.

² “Application-specific Integrated Circuits”.

³ Wikipedia. “Recording technique”. In: *Alvin and the Chipmunks*. Disponível em: https://en.wikipedia.org/wiki/Alvin_and_the_Chipmunks#Recording_technique.

em que os tons (“pitches”) são alterados para mais agudos ou mais graves dependendo se há aumento ou diminuição da frequência de amostragem, respectivamente⁴.

O mesmo efeito é observado se o processamento envolver uma compressão do sinal simplesmente descartando-se amostras, como em:

$$y[n] = x[n \cdot M]$$

em que M é um número inteiro positivo. Se o DAC utilizar a mesma frequência de amostragem do ADC, o áudio reconstruído terá componentes de frequência não presentes no sinal original. A equação expressa uma operação (variante no tempo!) em que, a cada M amostras de $x[n]$, 1 é preservada e $(M - 1)$ são descartadas, como mostrado na Figura 2 para $M = 3$.

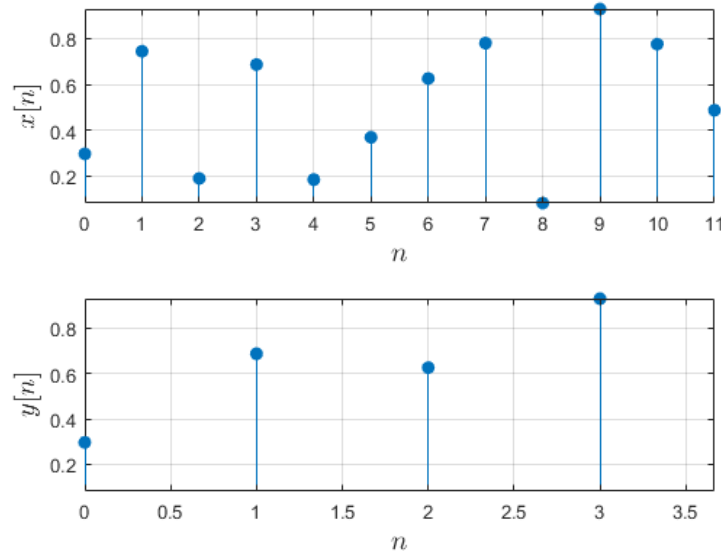


Figura 2: exemplo de descarte de amostras.

Tanto a variação na frequência de amostragem quanto o descarte de amostras para uma mesma f_s causam alterações no espectro de frequências⁵ de acordo com a propriedade da compressão/dilatação da Transformada de Fourier:

$$x(at) \stackrel{\mathcal{F}}{\Leftrightarrow} \frac{1}{a} X\left(j\frac{\omega}{a}\right).$$

O problema pode ser resolvido, sob algumas restrições, fazendo-se uma “compensação” do espectro do sinal de tempo discreto antes de reconstruí-lo⁶. Em tempo discreto, o descarte de amostras tem o seguinte efeito: suponha um sinal $x[n]$ com espectro de frequências

$$x[n] \stackrel{\mathcal{F}}{\Leftrightarrow} X(e^{j\omega}).$$

Ao descartar amostras por um fator M , obtém-se no domínio da frequência:

4 No Matlab, experimente executar o código:

```
[x, fs] = audioread('audio.wav');
sound(x, fs);
sound(x, 2*fs);
sound(x, fs/2);
```

5 OPPENHEIM, Alan V.; WILLSKI, Alan S. Tabelas de propriedades de Fourier e de pares básicos da transformada de Fourier. Em: Sinais e sistemas. 2ª ed. São Paulo: Pearson Education do Brasil, 2010. Seção 4.6, págs. 189 – 191.

6 OPPENHEIM, Alan V.; SCHAFER, Ronald W. Mudança da taxa de amostragem usando processamento em tempo discreto. Em: Processamento em tempo discreto de sinais. 3ª ed. São Paulo: Pearson Education do Brasil, 2012. Seção 4.6, págs. 107 – 116.

$$y[n] = x[n \cdot M] \Leftrightarrow Y(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X\left[e^{j\left(\frac{\omega}{M} - k \cdot \frac{2\pi}{M}\right)}\right].$$

O descarte causa uma expansão do espectro de frequências por M bem como o aparecimento de réplicas a cada $k \cdot \frac{2\pi}{M} \text{ rad}$, $0 \leq k \leq (M-1)$. Por exemplo, para $M = 2$ o espectro seria dado por

$$y[n] = x[n \cdot 2] \Leftrightarrow Y(e^{j\omega}) = \frac{1}{2} \sum_{k=0}^{2-1} X\left[e^{j\left(\frac{\omega}{2} - k \cdot \frac{2\pi}{2}\right)}\right] = \frac{1}{2} \left\{ X\left(e^{j\frac{\omega}{2}}\right) + X\left[e^{j\left(\frac{\omega}{2} - \pi\right)}\right] \right\}$$

e seria como o mostrado na Figura 3. Percebe-se que o espectro principal sofreu uma expansão por $M = 2$ enquanto a amplitude foi reduzida por 2. A expansão reflete-se no aparecimento de altas frequências, o que causa a sensação de aumento de tons no áudio reconstruído (sinal mais agudo).

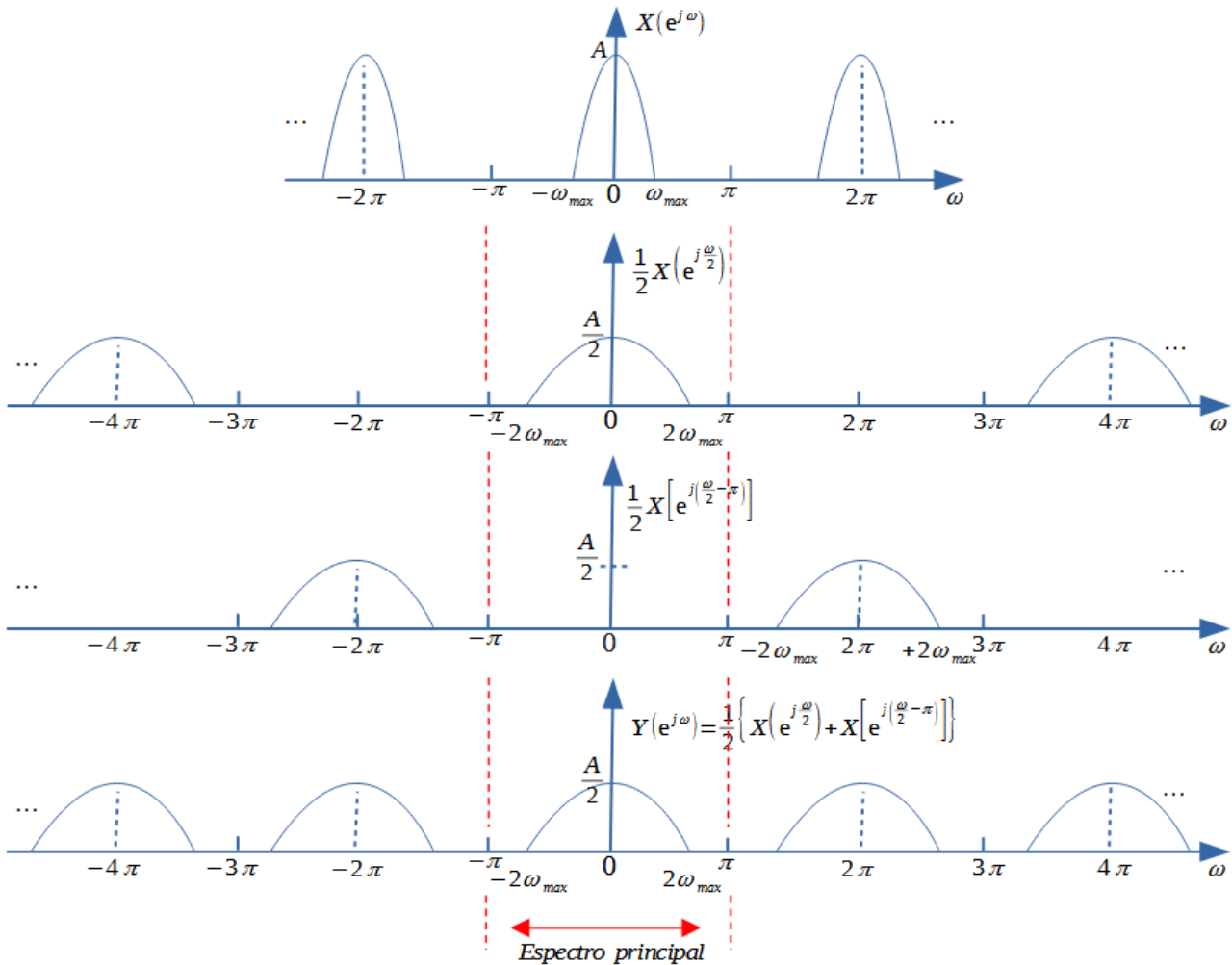


Figura 3: representação da dizimação por $M = 2$ no domínio da frequência.

Percebe-se que o descarte não pode ser realizado para qualquer valor de M uma vez que valores muito elevados causariam a violação do limite de Nyquist (aliasing). Para evitar o aliasing, deve-se garantir que

$$M \cdot \omega_{\max} \leq \pi \quad \text{ou} \quad \omega_{\max} \leq \frac{\pi}{M}.$$

Se não for possível garantir que $x[n]$ possui espectro limitado pode-se forçar esta limitação por meio de um filtro anti-aliasing *digital* com frequência de borda

$$\omega_p = \frac{\pi}{M}.$$

No exemplo anterior, considerando $M = 4$ e supondo um filtro ideal, teríamos a situação mostrada na Figura 4 e o espectro da saída seria dado por

$$Y(e^{j\omega}) = \frac{1}{4} \sum_{k=0}^{4-1} X_{PB} \left[e^{j\left(\frac{\omega}{4} - k \cdot \frac{2\pi}{4}\right)} \right] = \frac{1}{4} \left\{ X_{PB} \left(e^{j\frac{\omega}{4}} \right) + X_{PB} \left[e^{j\left(\frac{\omega}{4} - \frac{\pi}{2}\right)} \right] + X_{PB} \left[e^{j\left(\frac{\omega}{4} - \pi\right)} \right] + X_{PB} \left[e^{j\left(\frac{\omega}{4} - 3 \cdot \frac{\pi}{2}\right)} \right] \right\}.$$

Neste caso, não seria possível recuperar o espectro do sinal de tempo contínuo e o áudio reconstruído seria uma versão de baixas frequências daquele (som “abafado”).

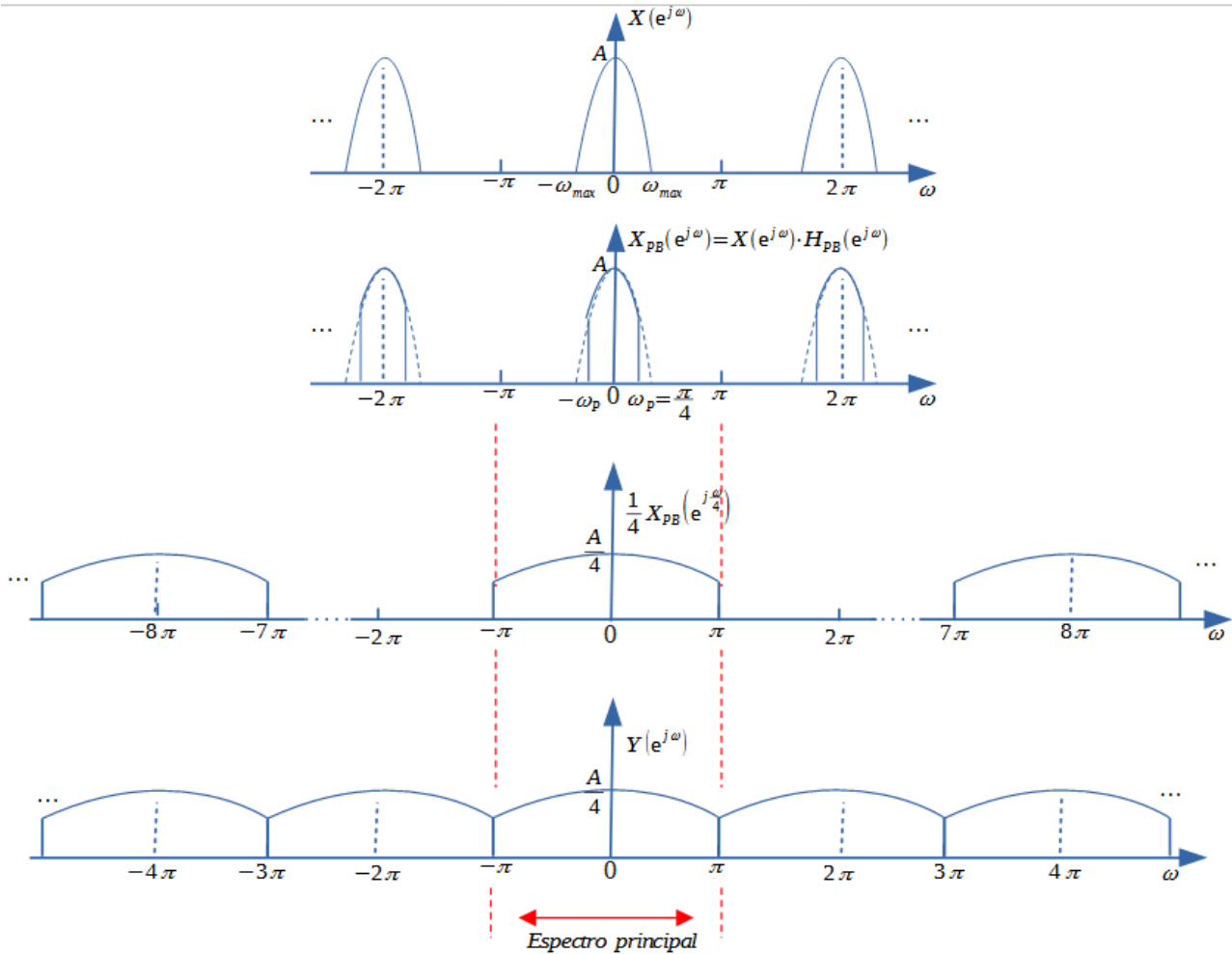


Figura 4: Espectro de um sinal limitado em $\omega_p = \frac{\pi}{4}$ e dizimado por $M = 4$.

Segundo o que foi observado, percebe-se que o algoritmo de um *dizimador de ordem M* genérico é composto por um filtro anti-aliasing com frequência de borda $\omega_p = \frac{\pi}{M}$ seguido de um descarte de M amostras, $y[n] = x[n \cdot M]$, como representado na Figura 5.

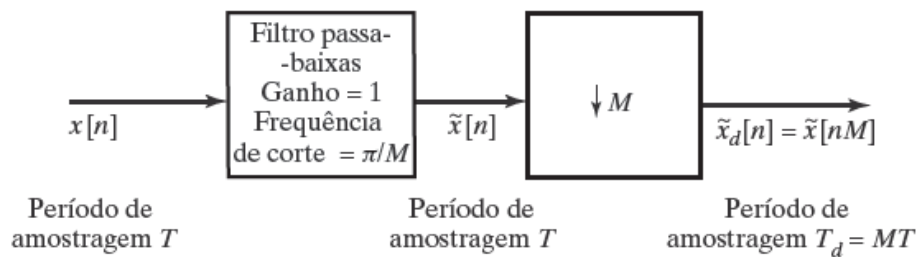


Figura 5: dizimador genérico de ordem M .

Uma situação análoga acontece para um sistema *interpolador de ordem L*. A operação visa aumentar a frequência de amostragem de um sinal por um fator L , ou seja, para cada par de amostras originais, 1 é preservada e $(L - 1)$ são criadas. Isto é feito por meio de duas operações em sequência. Na primeira, insere-se $(L - 1)$ zeros entre cada par de amostras do sinal,

$$y[n] = \begin{cases} x\left[\frac{n}{L}\right], & n = k \cdot L, k \in \mathbb{Z}; \\ 0, & \text{caso contrário.} \end{cases}$$

como mostrado na Figura 6 para $L = 4$.

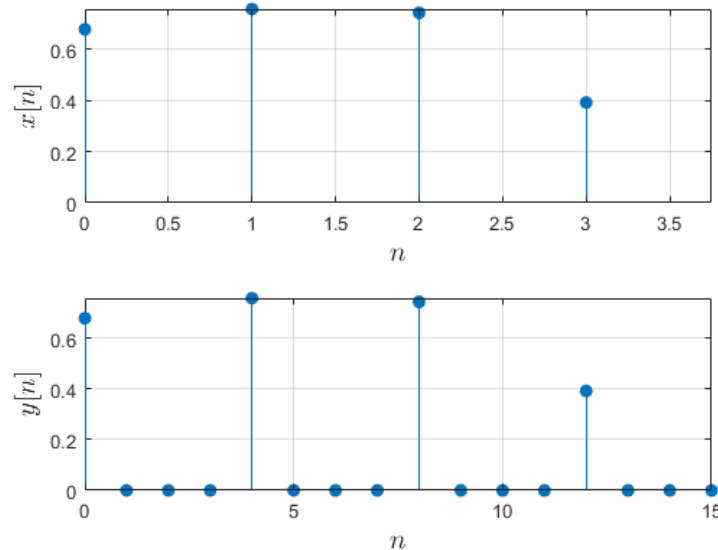


Figura 6: exemplo de enchimento com zeros.

Posteriormente, utiliza-se um algoritmo de interpolação para “suavizar” a forma de onda (interpolador de ordem zero, linear, splines, Lagrange, etc.⁸).

O efeito da interpolação é o mesmo de uma filtragem passa-baixas, ou seja, um interpolador de ordem L possui estrutura complementar à do dizimador de ordem M , como mostrado na Figura 7.

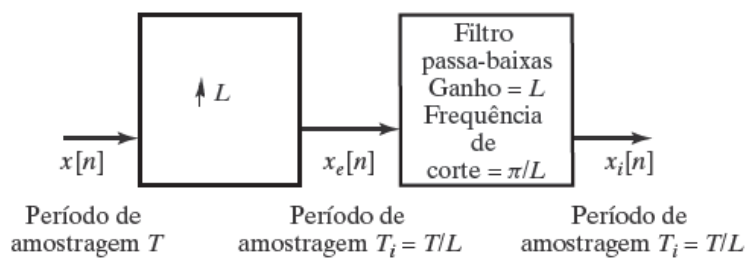


Figura 7: interpolador genérico de ordem L ⁹.

No domínio da frequência, o enchimento com zeros causa uma compressão do espectro do sinal:

$$x[n] \xleftrightarrow{\mathcal{F}} X(e^{j\omega}) \rightarrow y[n] = x\left[\frac{n}{L}\right] \xleftrightarrow{\mathcal{F}} Y(e^{j\omega}) = X(e^{jL\omega}).$$

Isto é mostrado na Figura 8 para $L = 2$. Após a compressão, o espectro é filtrado pelo filtro passa-baixas com frequência de borda

8 OPPENHEIM, Alan V.; SCHAFER, Ronald W. Filtros de interpolação simples e práticos. Em: Processamento em tempo discreto de sinais. 3ª ed. São Paulo: Pearson Education do Brasil, 2012. Seção 4.6.3, págs. 111 – 115.

9 OPPENHEIM, Alan V.; SCHAFER, Ronald W. Figura 4.23. Em: Processamento em tempo discreto de sinais. 3ª ed. São Paulo: Pearson Education do Brasil, 2012. Pag. 111.

$$\omega_p = \frac{\pi}{L},$$

resultando em uma única cópia da versão comprimida na faixa principal. Se a frequência de amostragem do DAC for mantida igual à do ADC, a sensação causada será a de um sinal de áudio de baixas frequências (grave) e que está sendo reproduzido de forma mais lenta do que o áudio original.

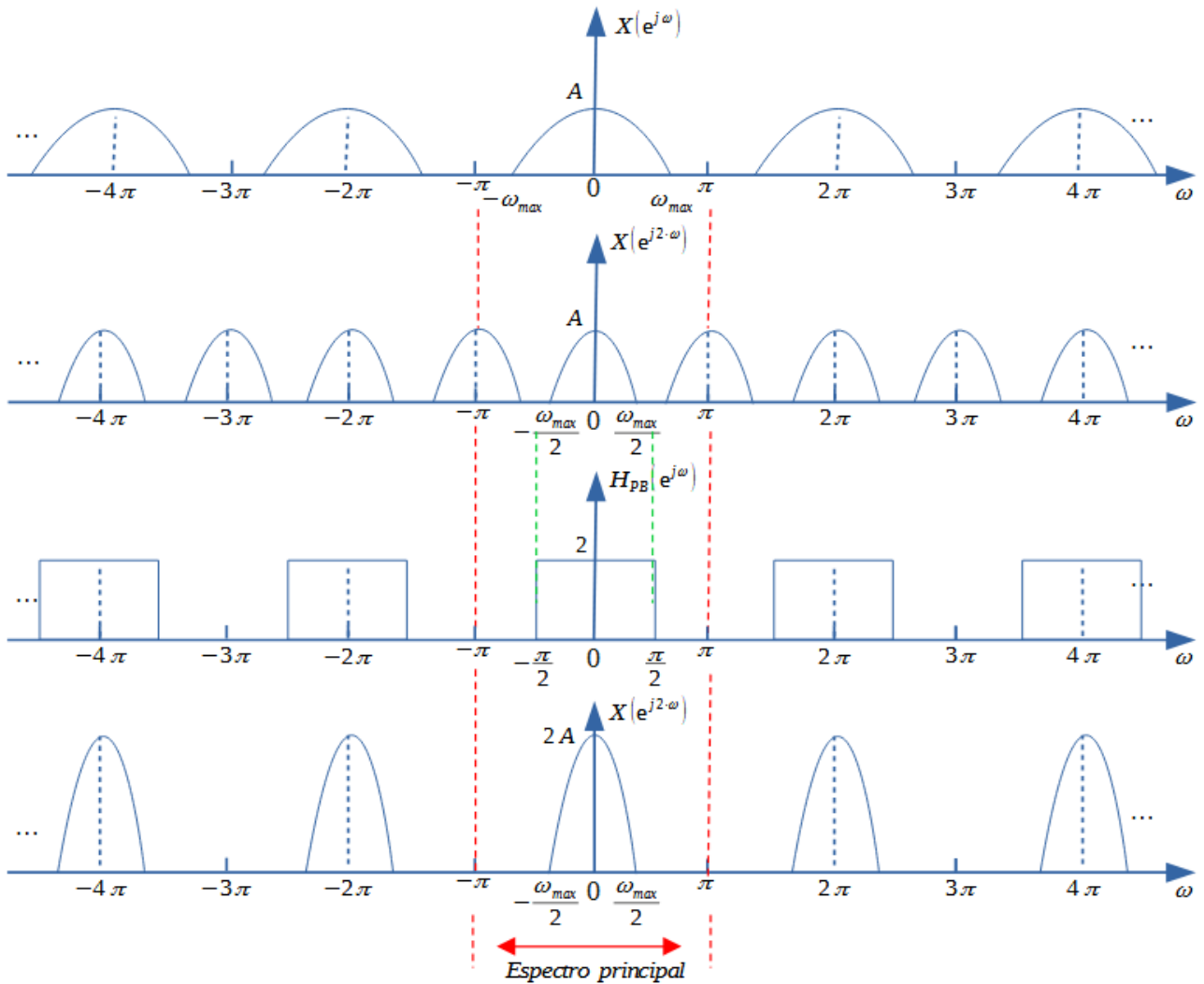


Figura 8: representação da interpolação por $L = 2$ no domínio da frequência.

Especificação de requisitos

A figura abaixo apresenta a forma de onda e o espectro de frequências de um sinal de áudio contido no arquivo “EuGostoDeDSP.wav”, fornecido em anexo¹⁰.

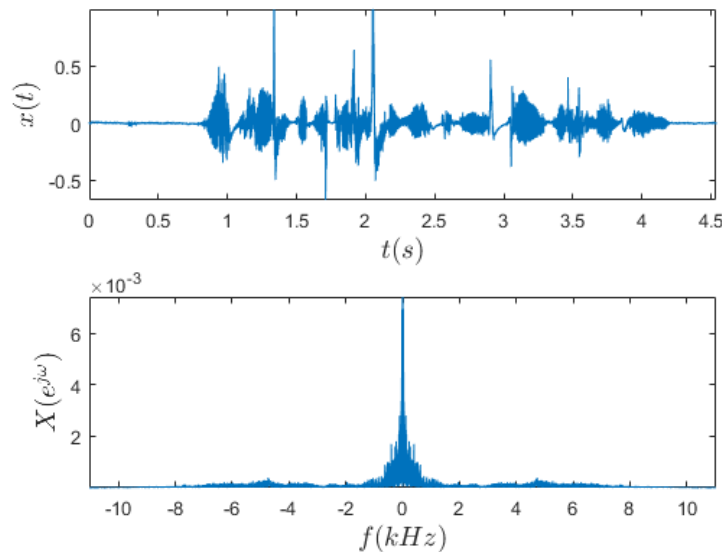


Figura 9: sinal nos domínios do tempo e da frequência.

O propósito deste trabalho é desenvolver um sistema para dizimar ou interpolar o sinal de áudio em diferentes taxas M e L , à escolha do usuário, visando aplicações em compressão de dados (redução de armazenamento) e modificação da velocidade de reprodução sem alteração de tons (“pitches”).

1. Implementação básica: compressor de áudio.

O sistema desenvolvido deverá realizar, no mínimo, as seguintes tarefas:

- 1.1. Carregar o arquivo “EuGostoDeDSP .wav” e gerar os gráficos da forma de onda em função do tempo, em s, e do espectro de frequência, em kHz, similares aos da Figura 9.
- 1.2. Executar o sinal no sistema de áudio do computador.
- 1.3. Implementar os algoritmos do dizimador de ordem M e o interpolador de ordem L , como mostrados nas figuras 5 e 7. Os algoritmos devem ser implementados como funções que receberão o sinal e a ordem de dizimação/interpolação desejados e retornarão o sinal processado, tal como em:

```
resultado = dizimarSinal(sinal, M);  
resultado = interpolarSinal(sinal, L);
```

- 1.4. Tanto a dizimação quanto a interpolação deverão utilizar filtros¹¹ passa-baixas projetados dinamicamente segundo os critérios discutidos na introdução. Utilize filtros FIR e faça o projeto pelo método da janela de Kaiser¹² com os seguintes parâmetros:

a) Frequências de bordas das faixas de passagem: $\omega_{P,diz} = \frac{\pi}{M}$, $\omega_{P,int} = \frac{\pi}{L}$.

¹⁰ No Matlab, o arquivo de áudio pode ser aberto e tocado por meio dos comandos

```
[x, fs] = audioread('EuGostoDeDSP.wav');  
sound(x, fs);
```

em que x representa o vetor de dados e fs a frequência de amostragem.

¹¹ No Matlab, veja a função $y = \text{filter}(b, a, x)$.

¹² OPPENHEIM, Alan V.; SCHAFER, Ronald W. *Método de projeto de filtros utilizando a janela de Kaiser et Exemplos de projetos de filtros FIR pelo método da janela de Kaiser..* 3ª ed. São Paulo: Pearson Education do Brasil, 2012. Seções 7.5.3 e 7.6, págs. 320 e 322.

- b) Frequências de bordas das faixas de rejeição: $\omega_{R,diz} = \frac{5}{4} \cdot \frac{\pi}{M}$ $\omega_{P,int} = \frac{5}{4} \cdot \frac{\pi}{L}$.
- c) Máxima distorção nas faixas de passagem e rejeição: $\delta = 1\%$
- d) Apresente os gráficos das janelas $w[n] \times n$ e reporte as ordens N e os parâmetro β encontrados para cada filtro.
- e) Apresente a resposta ao impulso $h[n] \times n$ e as respostas de magnitude $|H(e^{j\omega})| \times f(kHz)$ e fase $\theta(\omega) \times f(kHz)$ na faixa de frequências apresentada na Figura 9.
- f) Apresente os espectros de frequências dos sinais após a dizimação/interpolação, de forma similar às Figuras 4 e 8.
- 1.5. Implementar um algoritmo que receba do usuário um sinal, uma taxa de compressão M , faça a dizimação e armazene o resultado em um arquivo com as informações da frequência de amostragem original f_s e da taxa de compressão M .
- 1.6. Implementar um algoritmo que receba do usuário um arquivo comprimido, faça a leitura dos dados e realize a reconstrução do sinal de forma automática.
- a) Gere os gráficos da forma de onda e do espectro de frequências e execute o sinal reconstruído no sistema de áudio do computador.
- 1.7. Faça uma análise dos efeitos da taxa de compressão, da filtragem linear, qualidade do resultado, distorções, desempenho computacional, etc.
2. **Bônus:** modificação da velocidade de reprodução sem alteração de tons.

Desenvolva um sistema para alterar a velocidade de reprodução de um sinal de áudio sem alterar a frequência de amostragem e os tons do sinal¹³. O sistema deverá realizar, no mínimo, as seguintes tarefas:

- 2.1. Receber um sinal de áudio e a taxa de redução/ampliação da velocidade de reprodução desejada.
- 2.2. Realizar a compressão/descompressão por meio dos algoritmos de dizimação/interpolação.
- 2.3. Realizar a “formatação” do espectro de frequências de forma a minimizar o “efeito Chipmunk” discutido na introdução. Segundo o que foi apresentado nas figuras 4 e 8, proponha uma forma de minimizar a distorção do espectro do sinal original sem alterar a frequência de amostragem.
- 2.4. Gerar os gráficos da forma de onda e espectro de frequências.
- 2.5. Executar o sinal processado no sistema de áudio do computador.
- 2.6. Faça uma análise dos efeitos da taxa de compressão, da formatação do espectro, qualidade do resultado, distorções, desempenho computacional, etc.

Referências bibliográficas

1. OPPENHEIM, Alan V.; SCHAFER, Ronald W. *Processamento em tempo discreto de sinais*. 3ª ed. São Paulo: Pearson Education do Brasil, 2012.
2. MITRA, Sanjit K. *Digital signal processing: a computer-based approach*. 4th ed., New York: McGraw-Hill, 2011.

¹³ Wikipedia. *Audio time stretching and pitch scaling*. Disponível em: https://en.wikipedia.org/wiki/Audio_time_stretching_and_pitch_scaling.