PSI-3531 Processamento de Sinais Aplicado
02/05/2022 - Magno T. M. Silva

Experiência 3
Como o som é processado em um MP3 player

Observações:

1. Nesta experiência, vamos estudar a codificação de áudio MPEG-1 Layer-I.
2. Anexar no Moodle os programas usados na resolução do exercício. O relatório deve conter todos os gráficos e comentários solicitados.

Dados necessários para a realização do exercício

Antes de dar início à resolução do exercício, extraia todos os arquivos de Dados4.zip no seu diretório. Arquivos .mat poderão ser lidos com o comando load.

1 Banco de filtros com dois canais

Vamos começar examinando um banco de filtros de dois canais usando filtros convencionais e filtros QMF e com isso verificar o efeito da dizimação e interpolação em cada ramo.

a) Para isso, vamos considerar um sinal cuja frequência varia com o tempo. Um sinal *chirp* linear de tempo contínuo é definido como

$$x_c(t) = \cos(A_0 t^2),$$

em que A_0 tem unidade de radianos/s^2. Esses sinais são chamados de *chirps* (gorjeios em português) porque na faixa de frequências audíveis, pulsos curtos têm um som semelhante a gorjeios de pássaros. O sinal $x_c(t)$ é um membro da classe mais geral dos sinais de frequência modulada (FM) para os quais a frequência instantânea é definida como a derivada em relação ao tempo do argumento do cosseno, ou seja,

$$\Omega_i(t) = \frac{d(A_0 t^2)}{dt} = 2A_0 t,$$

que, por sua vez, varia proporcionalmente com o tempo. Por isso, a designação como sinal *chirp* linear. Amostrando $x_c(t)$ com período de amostragem $T_a = 1/f_a$ obtermos o sinal *chirp* linear de tempo discreto

$$x[n] = x_c(nT_a) = \cos(A_0 T_a^2 n^2) = \cos(\alpha_0 n^2),$$

em que $\alpha_0 = A_0 T_a^2$ tem unidade de radianos. A frequência instantânea do *chirp* amostrado é normalizada em frequência, ou seja,

$$\omega_i[n] = \Omega_i(nT_a) \cdot T_a = 2A_0 T_a^2 n = 2\alpha_0 n,$$

que mostra o mesmo aumento proporcional com n, com α_0 controlando a taxa de aumento. [Texto extraído de Oppenheim e Schafer: *Processamento em tempo discreto de sinais*, Pearson, 3a edição, 2013].

Gere amostras de um sinal *chirp* linear de tempo contínuo com 4 segundos de duração. A frequência deve variar 0 a 4 kHz. Use a frequência de amostragem $f_a = 8$ kHz. Gere um gráfico de seu espectrograma. Para isso use a função spectrogram.m do Matlab (specgram.m em versões mais antigas) e considere, por exemplo, uma FFT de comprimento 1024, uma janela de Hanning de comprimento 512 e uma sobreposição de 256 amostras.

b) Considere uma redução de taxa de amostragem por um fator $M = 2$. Em seguida, aumente a taxa de amostragem por um fator $L = 2$ inserindo zeros entre as amostras do sinal subamostrado. Multiplique o sinal superamostrado (com taxa maior) por 2 para manter a potência do sinal original do item a). Gere um gráfico do espectrograma do sinal resultante e compare com o espectrograma do sinal original obtido no item a). Ouça os dois sinais. O que você observa? Explique.

c) Projete um filtro passa-baixas FIR de fase linear por trechos com frequência de corte aproximadamente igual a 2000 Hz (considere que a faixa de passagem termina em 1960 Hz e a faixa de rejeição começa em 2040 Hz). O ripple na banda de passagem deve ser 2×10^{-4} (74 dB) e na faixa de rejeição 10^{-4} (80 dB). Um dos métodos mais utilizados para projetos de filtros FIR é o que utiliza o algoritmo de Parks–McClellan. Se optar por usar esse método, poderá usar as funções firpmord.m e firpm.m do Matlab (remezord.m e remez.m em versões mais antigas). Apresente um gráfico do módulo (em dB) e outro da fase do filtro projetado (você pode usar a função freqz.m do Matlab).

d) Filtre o sinal superamostrado do item b) pelo filtro projetado do item c). Gere um gráfico do espectrograma do sinal filtrado e compare com o espectrograma do sinal superamostrado obtido no item b). O que você observa? Faça um gráfico do sinal filtrado no tempo e ouça esse sinal. Explique o que acontece durante a primeira metade do intervalo de tempo e durante a segunda metade.

e) O que deve ser feito antes da subamostragem para evitar o *aliasing* observado no item d)? Proponha uma solução e a implemente. Gere os gráficos necessários para verificar sua solução. Neste item, você deve obter um sinal, cujo espectro resultante contém a banda de baixas frequências do sinal original.

f) Como obter um sinal cujo espectro corresponde à banda de altas frequências? Implemente a solução proposta correspondente e gere os gráficos para verificar.

g) Some o sinal do item e) ao do item f) e verifique se ocorre reconstrução perfeita. Não esqueça de descontar o atraso dos filtros FIR. Lembre que um filtro FIR simétrico de ordem N, comprimento $N+1$, tem um atraso de $N/2$ amostras.

2 Banco de filtros QMF com dois canais

a) Considere o filtro passa-baixas cuja resposta impulsiva $h_0[n]$ é fornecida no arquivo h0_QMF.mat. Gere os filtros $h_1[n]$, $g_0[n]$ e $g_1[n]$ para que ocorra reconstrução *quase* perfeita. Obtenha gráficos da resposta em frequência desses filtros. Compare com os filtros da Seção 1 quanto à seletividade em frequência e à reconstrução perfeita.

b) Filtre o chirp gerado na Seção 1 pelo banco de filtros formado com os filtros QMF do item a) e verifique a partir do espectrograma e do erro de reconstrução se de fato ocorre reconstrução perfeita.

3 Banco de filtros pseudo-QMF com 32 canais

Vamos estender o esquema de banco de filtros anterior para M canais. Na abordagem pseudo-QMF, os filtros de análise $H_i(z)$ são obtidos modulando-se a resposta impulsiva $h[n]$ de um filtro passa-baixas protótipo com M frequências de portadora. Essas frequências são uniformemente distribuídas no intervalo de frequências normalizadas [0 1]. Vamos considerar o banco de filtros usado no codificador MPEG-1 Layer-I que possui 32 canais. Esse banco de filtros satisfaz a condição de reconstrução perfeita e os coeficientes das respostas impulsivas dos filtros de análise e síntese estão no arquivo Analise_Sintese32.mat. Note que cada linha das matrizes fornecidas corresponde à resposta impulsiva de um filtro. Assim, os coeficientes da linha 1 da matriz PQMF32_Hfilters correspondem à resposta impulsiva do filtro de análise do Canal 0 e os coeficientes da linha 32, a resposta impulsiva do filtro do Canal 31. O mesmo ocorre com a matriz PQMF32_Gfilters.

a) Considerando frequência de amostragem igual a $f_a = 44100$ Hz, gere um gráfico do módulo da resposta em frequência dos 32 filtros de análise. A partir do

gráfico, observe que dada uma certa banda ocorre superposição apenas com as bandas adjacentes. Note também que o ganho do filtro é igual a 15 dB, i.e., $20\log_{10}(32)/2$. Assim, filtrando o sinal duas vezes pelo filtro resulta um ganho de 32, o que compensa a dizimação pelo fator 32.

b) Estenda os programas que você fez nas seções anteriores, considerando agora o banco de filtros pseudo-QMF com 32 canais do item a). Gere novamente amostras de um sinal *chirp* linear de tempo contínuo, mas agora com 20 segundos de duração. A frequência deve variar 0 a 22050 Hz. Use a frequência de amostragem $f_a = 44100$ Hz. Gere um gráfico do espectrograma do *chirp* original e do *chirp* reconstruído após análise e síntese. Otenha também um gráfico do erro de reconstrução.

c) A função snr.m fornecida calcula a razão sinal-ruído (SNR – *signal-to-noise ratio*) do sinal reconstruído em dB. Ela estima a potência do ruído de reconstrução a partir do sinal original. Use essa função para calcular a SNR do sinal reconstruído no item b), ou seja,

```
SNR_PQMF=snr(sinal original, sinal reconstruido,0)
```

Não esqueça de levar em conta o atraso causado no sinal reconstruído.

d) Considere o arquivo de áudio fornecido violino.wav e substitua o chirp usado no item b) por esse sinal de áudio. Repita o item b) para esse arquivo. Ouça e gere gráficos dos sinais de saída do filtros de síntese dos canais 0, 5 e 10. O que você observa? Novamente calcule a SNR como no item c).

4 Codificação de audio perceptual

O processo de filtragem em sub-bandas desenvolvido nas seções anteriores transforma o fluxo original de amostras com frequência de amostragem f_a em 32 sub-bandas paralelas amostradas com $f_a/32$. Ainda são necessários os passos de quantização e codificação para se conseguir a taxa de compressão de bits global. Vamos agora verificar o efeito da quantização.

a) Considere novamente o sinal de áudio violino.wav. Em cada sub-banda, quantize o sinal de entrada dos filtros de síntese considerando 4 bits. Para isso, use um quantizador do tipo *mid-tread* em $[-1, +1]$ (função midtreadQ.m fornecida no Moodle). Para cada caso, ouça o sinal de saída do banco de filtros e calcule a SNR. Os sinais ficaram distorcidos? A SNR variou muito em relação ao valor obtido na Seção anterior?

b) A forma como você implementou o banco de filtros usando a função filter.m do Matlab não é a mais otimizada. Na prática, se utiliza a transformada *Lapped*, que é um tipo de transformação linear em bloco com sobreposição nas bordas que implementa a análise e a síntese. Utilize a função lappedQ.m fornecida do Moodle e repita o item a) usando essa função. Compare o sinal reconstruído (saída da função lappedQ.m) com o sinal original, desprezando do sinal original as $N_h - 1$ amostras iniciais e as $N_h - 1$ amostras finais, sendo N_h o número de coeficientes de cada subfiltro. Você obteve o mesmo resultado?

c) Uma forma de melhorar a qualidade do som quantizado é aplicar um fator de escala ao quantizador de cada sub-banda. No MPEG-1 Layer-I, se calcula um novo fator de escala a cada $32 \times 12 = 384$ amostras do sinal original (amostrado com f_a) ou a cada 12 amostras (do sinal subamostrado com $f_a/32$). Considere novamente o sinal de áudio do arquivo violino.wav. Complete a linha 32 do código lappedQadap.m fornecido no Moodle. Ouça o sinal gerado com esse quantizador adaptativo e calcule novamente a SNR. Para calcular a SNR e o erro de reconstrução neste item, compare o sinal reconstruído (saída da função lappedQadap.m) com o sinal original, desprezando do sinal original as $N_h - 1$ amostras iniciais e as N_h amostras finais, sendo N_h o número de coeficientes de cada subfiltro. Você consegue observar melhoria na qualidade do som?

d) Para melhorar ainda mais a qualidade, deve-se usar o modelo psicoacústico. Para isso, use as seguintes funções fornecidas no Moodle: lappedPsico.m, MPEG1_psycho_acoustic_model1.m e MPEG1_bit_allocation.m. A função lappedPsico.m consiste em uma modificação da função usada no item c) para incorporar as outras duas funções. A função MPEG1_psycho_acoustic_model1.m calcula os valores da razão sinal-máscara (SMR) levando em conta o modelo psicoacútico e a função MPEG1_bit_allocation.m implementa um algoritmo que calcula o número de bits em função dos valores de SMR. Considere novamente o sinal de áudio do arquivo violino.wav e use a taxa de bits igual a 192000 bits/s. Calcule a SNR e faça o gráfico do espectrograma do sinal processado pela função lappedPsico.m. Novamente, para calcular a SNR e o erro de reconstrução neste item, compare o sinal reconstruído (saída da função lappedPsico.m) com o sinal original, desprezando do sinal original as $N_h - 1$ amostras iniciais e as N_h amostras finais, sendo N_h o número de coeficientes de cada subfiltro. Houve melhoria em relação ao caso anterior?

e) Modifique a função lappedPsico.m para obter gráficos 3D da SMR, SNR e número de bits alocados em função da banda de frequência e do bloco (frame). Note que esses valores são calculados a cada 12 amostras na taxa mais baixa. Isso significa que o número de bits do bloco # 1 é o mesmo dos blocos # 2, # 3, $\cdots$ # 12. A

SMR do bloco # 13 é a mesma dos blocos # 14, # 15, $\cdots$ # 24. Comente os resultados obtidos.

A Codificação de áudio perceptual com mascaramento

O ouvido humano é um "dispositivo" muito complexo. Sua sensibilidade depende da frequência, ou seja, para ser ouvido, um estímulo de banda estreita deve ter um nível de pressão sonora superior a um limiar, chamado de limiar auditivo absoluto, que é uma função da frequência. Para uma dada frequência, a percepção do volume não está linearmente relacionada à pressão sonora. Além disso, a sensibilidade do ouvido a um determinado estímulo não é a mesma se este estímulo estiver sozinho ou se estiver misturado com algum outro estímulo de mascaramento. Ainda pior, esse efeito depende do tipo de estímulo de mascaramento. Por exemplo, tons puros e ruído de banda estreita não têm o mesmo efeito de mascaramento. Por último, a sensação acústica depende do ouvinte e varia com o tempo. Na Figura 1, são mostrados os limiares de audição e de dor (figura à esquerda) e o efeito do mascaramento devido a um tom puro em 1 kHz com diferentes intensidades (figura à direita). Sons que estão abaixo do limiar de audição modificado ficam "escondidos" pelo tom do mascaramento. Um tom puro em 1 kHz com uma intensidade de 80 dB SPL (*sound pressure level*), por exemplo, faz com que um outro tom puro em 2 kHz e 40 dB SPL se torne inaudível.

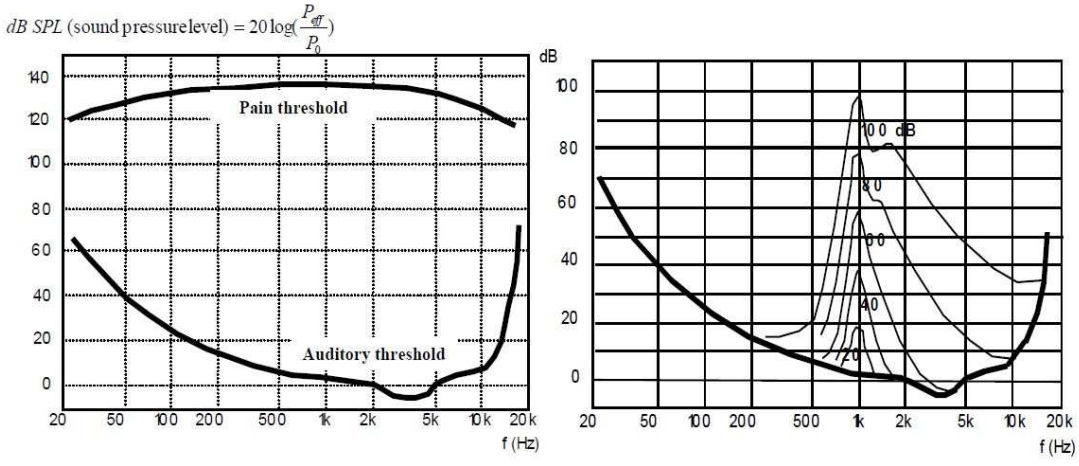

Figura 1: Esquerda: limiares de audição e de dor em função da frequência; Direita: efeito do mascaramento devido a um tom puro em 1 kHz para várias intensidades. Fonte: [Dutoit e Marques: *Applied Signal Processing: a Matlab proof of concept*, Springer, 2009].

O interesse em codificação por sub-bandas é estabelecer uma relação mestre-escravo em cada sub-banda entre (i) o limiar de mascaramento local calculado pelo modelo psicoacústico a partir de uma estimação dos estímulos de mascaramento presentes no espectro de audição e (ii) o ruído de quantização, ou seja, o número de bits usados para quantizar o sinal na correspondente sub-banda. A Figura 2 esquematiza o diagrama de blocos do codificador MPEG-1 Layer-I.

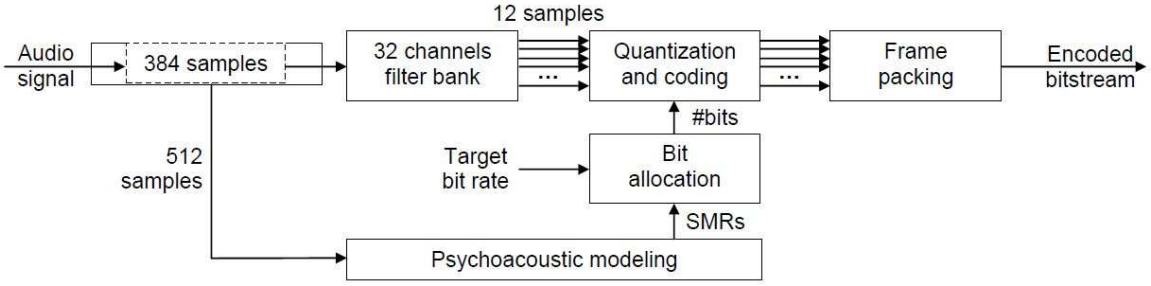

Figura 2: Diagrama de blocos do codificador de áudio do MPEG-1 Layer-I. Fonte: [Dutoit e Marques: *Applied Signal Processing: a Matlab proof of concept*, Springer, 2009].

Embora os codificadores em sub-bandas sejam baseados nas características do processo de audição, eles não o refletem fielmente. Por exemplo, as bordas das bandas críticas não são uniformemente espaçadas em frequência, mas a maioria dos codificadores em sub-bandas usa canais com a mesma largura de banda. O efeito do mascaramento depende do volume atual do sinal. Enquanto o codificador não sabe o nível de volume atual do ouvinte, o modelo psicoacústico utilizado na codificação de sub-bandas considera que o usuário define o nível de volume de tal forma que o bit menos significativo do código PCM (*pulse code modulation*) de 16 bits original possa ser ouvido.

A norma de áudio MPEG-1, desenvolvido pelo grupo MPEG (*Moving Picture Experts Group*), define três camadas de complexidade de processamento, cada uma com seu próprio esquema de codificação em sub-bandas, modelo pscicoacústico e quantizador. Começando com 1,4 Mbits/s para um sinal de áudio estéreo, a Camada-I (*Layer-I*) atinge um nível de compressão de 1:4 (abaixo de 384 kbits/s). Com a adição da Camada-II atinge-se uma taxa de compressão de 1:6-8 (entre 256 e 192 kbits/s). Esse codificador é muito usado em TV digital. O MPEG-1 Layer-III, comercialmente conhecido como MP3, comprime ainda mais o fluxo de dados até 128 a 112 kbits/s (isto é, uma taxa de compressão em 1:10-12). Nesta experiência, vamos considerar somente a Camada-I.

O codificador MPEG-1 Layer-I considera 32 canais. As amostras são processadas por blocos de 384 amostras, ou seja, 12 amostras na taxa mais baixa. O modelo

psicoacústico é baseado na estimativa com 512 amostras da densidade espectral de potência local do sinal, denotada por $S_{xx}(f)$. Máximos locais com picos dominantes são detectados como máscaras tonais e extraídas da FFT. Uma única máscara de ruído é então computada da energia restante em cada banda crítica. As máscaras são então combinadas, representando novamente a existência de bandas críticas (tipicamente, várias máscaras em uma única banda são mescladas na mais forte delas). Limiares individuais de mascaramento são finalmente estimados para cada máscara tonal e de ruído e somados para produzir o limiar global de mascaramento, $\Phi(f)$. Isso leva à determinação de uma razão sinal-máscara (SMR – *signal-to-mask ratio*) em cada sub-banda dada por

$$\mathrm{SMR}(k) = 10 \log_{10} \left(\frac{\max_{\text{banda } k} S_{xx}(f)}{\min_{\text{banda } k} \Phi(f)} \right).$$

Sinais das sub-bandas são quantizados uniformemente e o número de bits para cada sub-banda é escolhido para que a SNR fique superior à SMR para que o ruído de quantização se torne inaudível, como mostrado na Figura 3. Sub-bandas de frequência mais alta tipicamente requerem menos bits. Em particular, são atribuídos 0 bits às sub-bandas 27 a 32. A quantização é feita adaptativa, normalizando-se os sinais de sub-banda antes de quantizá-los. A normalização é realizada por blocos de 12 amostras de sub-bandas (blocos e 384 amostras na frequência de amostragem original), estimando-se um fator de escala local e dividindo as amostras por esse valor.

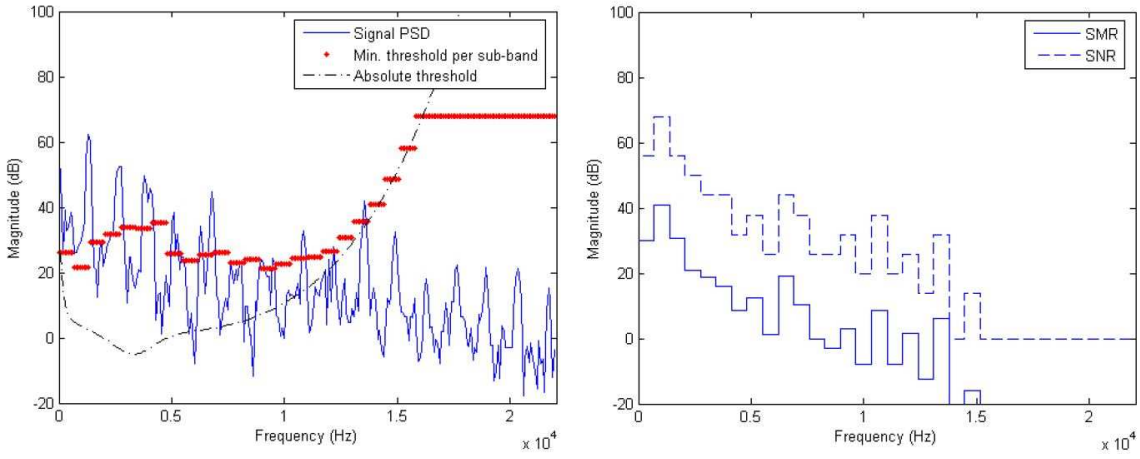

Figura 3: Modelo pscicoacústico do MPEG-1. Figura à esquerda: Limiar de mascaramento por sub-banda. Figura à direita: SMR e máxima SNR obtida após a alocação de bits. Fonte: [Dutoit e Marques: *Applied Signal Processing: a Matlab proof of concept*, Springer, 2009].

B Quantizadores uniformes

Existem dois tipos de quantização: a uniforme e não uniforme. Quando níveis de quantização são uniformemente espaçados, define-se a quantização como uniforme. Neste caso, há também dois tipos de quantização: a *mid-tread* e a *mid-rise*. Na quantização *mid-tread* existe o nível zero de modo que há um número ímpar de níveis, enquanto na quantização *mid-rise* o nível zero não aparece e há um número par de níveis, como mostrado na Figura 4.

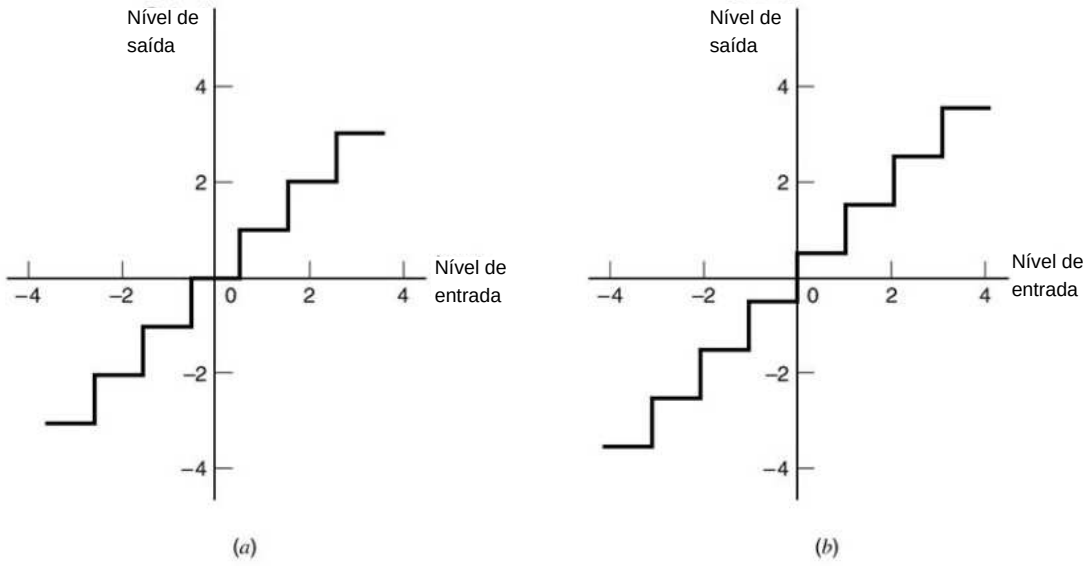

Figura 4: (a) Quantização *mid-tread* e (b) Quantização *mid-rise*.

Nessa experiência, para efeito de simplificação, vamos considerar apenas o quantizador *mid-tread*. Considerando que a amplitude do sinal está no intervalo $[-\rho,\ \rho]$, a equação desse quantizador é dada por

$$Q(x) = \frac{1}{\alpha}\left\lfloor \alpha x + \frac{1}{2} \right\rfloor,$$

em que $\lfloor x \rfloor$ é a função que retorna o maior número inteiro que é menor ou igual a x (função floor.m no Matlab), $\alpha = 2^{N-1}/\rho$, sendo N o número de bits que se deseja usar na quantização. Note que ρ é um fator importante na quantização e já está embutido neste quantizador. Na prática, o sinal é normalizado por ρ, o quantizador é mantido fixo e o sinal quantizado deve ser multiplicado por ρ. Para exemplificar, vamos considerar a quantização com $N = 4$ bits de um sinal senoidal de 10 Hz amostrado com $f_a = 1000$ Hz, ou seja,

$$x[n] = 0{,}5\sin(2\pi 10n/1000).$$

Esse sinal foi dividido blocos de 10 amostras sem sobreposição. Em cada bloco, se calculou o valor de $\rho_k = \max_{\text{bloco}\,k}\{|x[n]|\}$. Os resultados da quantização considerando $\rho_k = 1$ para todos os blocos e ρ_k variável estão mostrados na Figura 5. Note que um

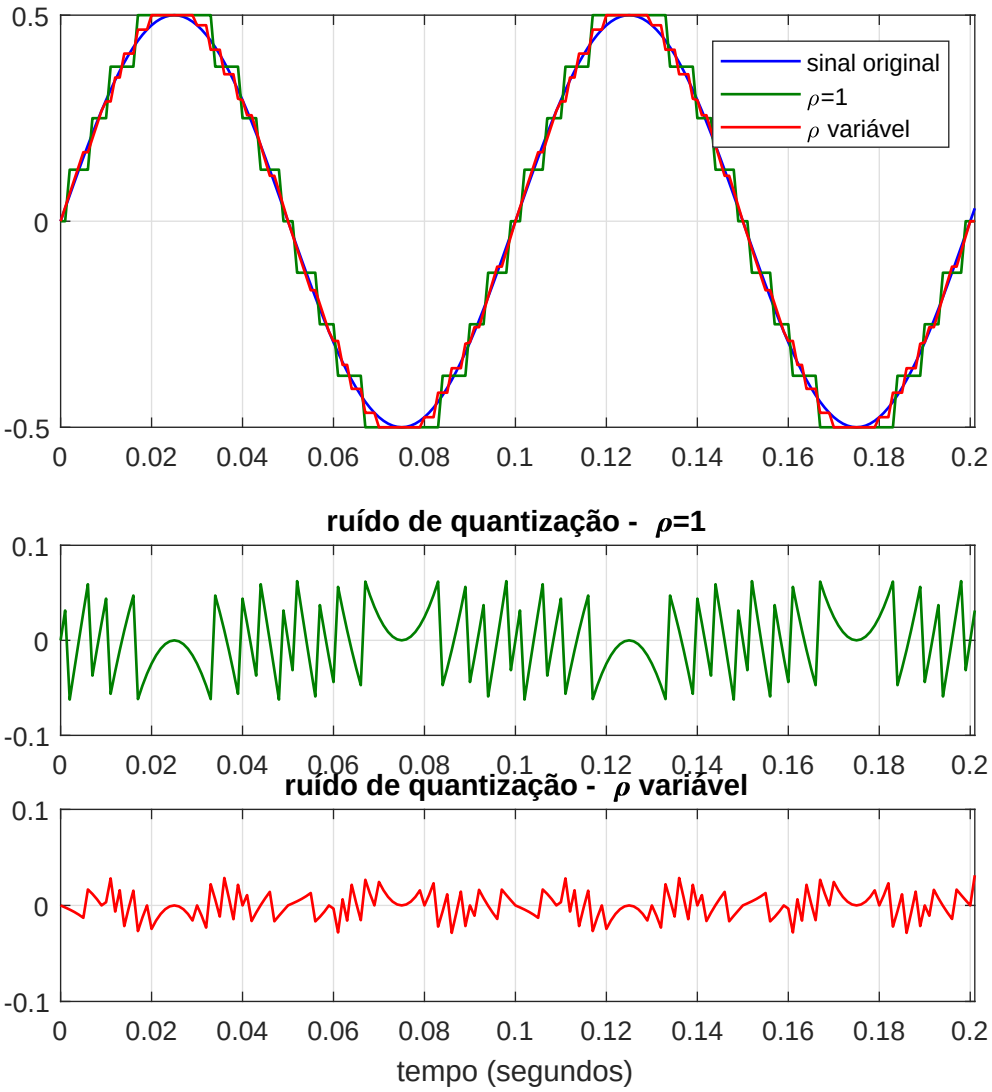

Figura 5: Quantização de um sinal senoidal com quantizador *mid-tread* com $N = 4$ bits; fator unitário e variável.