PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

FACULDADE DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

SISTEMA DE ESTEGANOGRAFIA EM ÁUDIO DIGITAL QUE UTILIZA TÉCNICAS EFICIENTES DE INSERÇÃO DE DADOS

CRISTIANO AUGUSTO SCHÜTZ

PROF. DR. RUBEM DUTRA RIBEIRO FAGUNDES

Porto Alegre, março de 2009.

CRISTIANO AUGUSTO SCHÜTZ

SISTEMA DE ESTEGANOGRAFIA EM ÁUDIO DIGITAL QUE UTILIZA TÉCNICAS EFICIENTES DE INSERÇÃO DE DADOS

Dissertação apresentada como requisito para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rubem Dutra Ribeiro Fagundes

Porto Alegre, março de 2009.

CRISTIANO AUGUSTO SCHÜTZ

SISTEMA DE ESTEGANOGRAFIA EM ÁUDIO DIGITAL QUE UTILIZA TÉCNICAS EFICIENTES DE INSERÇÃO DE DADOS

Dissertação apresentada como requisito para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovada em 30 de março de 2009.

BANCA EXAMINADORA:

_____

Rubem Dutra Ribeiro Fagundes, Dr.
Presidente – PUCRS

_____

Fernando César Comparsi de Castro, Dr.
PUCRS

_____

Vinicius Licks, Dr.
PUCRS

Dedico este trabalho a todos aqueles que, ainda vivos ou já falecidos, dedicaram ou ainda dedicam a maior parte de suas vidas ao desenvolvimento da ciência e da tecnologia, mas sem deixar de lado os princípios morais e éticos que regem uma boa conduta na sociedade.

Agradecimentos

À minha família, e em especial aos meus pais, Carlos e Rosana, pelo apoio moral, afetivo, educacional e financeiro e pelo incentivo constante na minha busca por aprimoramento.

À minha namorada Patricia Alves, por ter aturado meu mau-humor oriundo do estresse causado por esta dissertação, pelo incentivo e pela força nas horas difíceis e por todo o amor a mim dado.

À CAPES, pelo auxílio financeiro que possibilitou meu ingresso no PPGEE.

“Versuch nicht, ein Mann des Erfolgs zu werden. Werde lieber ein Mann von Wert!”

(“Não procure se tornar um homem de sucesso. Procure, antes, se tornar um homem de valor.”)

Albert Einstein

RESUMO

Esta dissertação apresenta uma melhoria feita em uma técnica de esteganografia para áudio que emprega espalhamento espectral (*spread spectrum*), o que permite que se insira muito mais informação no sinal hospedeiro, sem que com isso haja degradação da qualidade perceptual, utilizando três métodos diferentes de compressão de dados sem perdas nas fontes de dados. Um código corretor de erros (ECC) também é utilizado a fim de permitir taxas de bits de dados mais altas sem afetar a probabilidade de erros de detecção. O modelo psicoacústico, assim como a técnica de espalhamento espectral, são explicados em detalhes, e o comportamento do sistema auditivo humano (HAS) sob o efeito de estímulos auditivos é descrito. Além disso, testes de medição da capacidade de inserção e de avaliação da qualidade perceptual do áudio – mais especificamente a versão básica do algoritmo *Perceptual Evaluation of Audio Quality* (PEAQ) (ITU, 1998-2001) – são realizados com seis diferente excertos de áudio e diferentes arquivos de texto comprimidos sendo utilizados como fontes de dados. Finalmente, é apresentada uma comparação dos algoritmos de compressão utilizados nesta dissertação, provando que a compressão dos dados não afeta a qualidade perceptual do áudio e, ao mesmo tempo, pode aumentar a capacidade de inserção do sinal hospedeiro em mais de 100%.

ABSTRACT

This dissertation presents an improvement to a spread spectrum audio steganography technique that allows us to embed much more information into the host signal, with no perceptual quality degradation, by using three different lossless data compression methods on the data sources. An error-correcting code (ECC) is also used in order to allow higher data bit rates without affecting the detection error probability. The psychoacoustic model, as well as the spread spectrum technique, are explained in details, and the behavior of the human auditory system (HAS) under the effect of auditory stimuli is described. Furthermore, embedding capacity measurement tests and audio perceptual quality evaluation tests – more specifically the basic version of the Perceptual Evaluation of Audio Quality (PEAQ) algorithm (ITU, 1998-2001) – are performed over six different audio excerpts, with different compressed text files as data sources. Finally, a comparison of the compression algorithms used in this dissertation is presented, proving that data compression do not affect the perceptual audio quality and, at the same time, it can enhance the host signal embedding capacity in over 100%.

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ATH – absolute threshold of hearing (limiar absoluto de audição)

AWGN – additive white Gaussian noise (ruído branco gaussiano aditivo)

BCH – Bose-Chadhuri-Hocquenghem

BPSK – binary phase-shift keying

BWT – Burrows-Wheeler transform (transformada de Burrows-Wheeler)

CD – compact disc (disco compacto)

DCT – discrete cosine transform (transformada discreta do cosseno)

DFT – discrete Fourier transform (transformada discreta de Fourier)

DHT – discrete Hartley transform (transformada discreta de Hartley)

DS – direct sequence

DSP – digital signal processing (processamento digital de sinais)

DSSS – direct sequence spread spectrum

DVD – digital video disc (disco digital de vídeo)

DWT – discrete wavelet transform (transformada wavelet discreta)

ECC – error correcting code (código corretor de erros)

FFT – fast Fourier transform (transformada rápida de Fourier)

FH – frequancy hopping

GCC – GNU Compiler Collection

HAS – human auditory system (sistema auditivo humano)

HIR – host-to-information ratio (relação sinal hospedeiro/quantidade de informação)

ISO – International Organization for Standardization (Organização Internacional para Padronização)

ITU – International Telecommunication Union (União Internacional de Telecomunicações)

JND – just noticeable distortion (mínima distorção perceptível)

JPEG – Joint Photographic Experts Group

LFSR – linear feedback shift register (registrador de deslocamento com realimentação linear)

LZW – Lempel-Ziv-Welch

MDCT – modified discrete cosine transform (transformada discreta do cosseno modificada)

ME – módulo de extração

MI – módulo de inserção

MOV – model output variable (variável de saída do modelo)

MPEG – Moving Pictures Experts Group

MTF – Move-to-Front transform (transformada Move-to-Front)

NMN – noise-masking-noise (ruído mascarando ruído)

NMT – noise-masking-tone (ruído mascarando tom)

ODG – objective difference grade

PCM – pulse code modulation

PEAQ – perceptual evaluation of audio quality (avaliação perceptual da qualidade do áudio)

PN – pseudonoise (pseudo-ruído)

SFM – spectral flatness measure (medida do achatamento espectral)

SMR – signal-to-mask ratio (relação sinal-mascaramento)

SNR – signal-to-noise ratio (relação sinal-ruído)

SPL – sound pressure level (nível de pressão sonora)

SS – spread spectrum (espalhamento espectral)

TH – time hopping

TNM – tone-masking-noise (tom mascarando ruído)

Sumário

1. Introdução

A ocultação de informações vem sendo utilizada durante séculos. Há registros escritos por Heródoto no século V A.C. que revelam como Histieu, o tirano de Mileto, por volta de 440 A.C., raspou a cabeça de um mensageiro e tatuou nela uma mensagem, que desapareceu depois que o cabelo voltou a crescer, a fim de instigar uma revolta contra os Persas (Petitcolas, 2000).

Até meados dos anos 1950, a ocultação de informações era utilizada somente com propósitos militares, principalmente em comunicações altamente sigilosas. Com o advento dos equipamentos e meios de gravação de áudio, o interesse por técnicas de ocultação de informações ganhou um novo foco, o de proteger os direitos autorais e de propriedade das obras musicais. Em 1954, nos Estados Unidos da América, foi registrada uma patente descrevendo um método de inserir um código de identificação em músicas de maneira imperceptível para fins de prova de propriedade (Cox e Miller, 2001).

Atualmente, conteúdos multimídia são facilmente copiados e distribuídos através da Internet, fazendo-se necessário o desenvolvimento de novas técnicas para monitorar, controlar e restringir o acesso e a distribuição desse tipo de material. Outra aplicação da ocultação de informações em meios digitais é a de agregar mais valor ao conteúdo multimídia, inserindo material adicional no próprio conteúdo já existente sem, no entanto, aumentar o espaço de armazenamento. Esse será o foco principal deste trabalho.

1.1. Motivação

Nos dias de hoje, a venda de conteúdo acabou se tornando mais rentável do que a venda dos meios de armazenamento em si. Para que se entenda melhor essa afirmação, analise-se a situação atual da indústria fonográfica mundial. É mais lucrativo vender ingressos para shows do que vender CDs. Isso dá ao público acesso ao conteúdo produzido por determinado artista, mas esse conteúdo, em geral, não será armazenado em nenhum meio

para ser desfrutado mais tarde, a não ser os casos de shows gravados ao vivo e vendidos em DVDs posteriormente.

Além do mais, a venda de músicas através da Internet vem crescendo. O que está sendo comercializado nesse caso é a música, o conteúdo em si, e não o meio de armazenamento dessa música, como é o caso de um CD de áudio. As pessoas podem escolher que músicas elas querem adquirir individualmente, sem ter a necessidade de comprar um CD inteiro quando gostam de apenas uma música. Idealmente, tem-se por objetivo viabilizar a compra de músicas pela rede mundial possibilitando o download de um arquivo de áudio do mesmo tamanho de antes, mas com mais conteúdo. Tal conteúdo permanece oculto em meio ao áudio, sendo imperceptível para quem quer apenas escutar música, porém acessível àqueles que desejam obtê-lo. Ele pode ser qualquer tipo de informação em formato digital, como texto, imagens exclusivas, material de áudio adicional etc. É possível acrescentar letras de músicas, fotos do artista, alguma declaração do próprio artista gravada em áudio, entre outros.

Esse tipo de abordagem da ocultação de informações ainda vem sendo pouco estudado na literatura. Por esta razão, este trabalho apresenta uma técnica que permite inserir em arquivos de áudio uma quantidade maior de informações do que já foi mostrado em boa parte dos sistemas propostos anteriormente. Além disso, é inútil aumentar a capacidade de inserção de informações se com isso houver degradação da qualidade do sinal hospedeiro.

1.2. Objetivos

Este trabalho tem por objetivo o desenvolvimento de uma técnica de esteganografia capaz de aumentar a capacidade de inserção de informações de um sistema de ocultação de informações para áudio. Isso será realizado utilizando diferentes métodos de compressão de dados sem perdas, que serão comparados para que se verifique qual o mais eficiente dentre eles. O sistema proposto é baseado em uma técnica derivada da área de comunicação digital conhecida como espalhamento espectral (*spread spectrum*).

A imperceptibilidade das informações inseridas no sinal de áudio será garantida pela aplicação de um modelo psicoacústico, que é um modelo matemático que representa a maneira como o sistema auditivo humano percebe os sons. Um código corretor de erros também será utilizado para previnir erros de detecção da informação devido às taxas de bits maiores com que o sistema de esteganografia trabalha.

Testes para medir a capacidade de inserção de informações e testes para avaliar a qualidade perceptual das amostras de áudio serão realizados a fim de verificar se o sistema implementado realmente funciona da maneira proposta. Este trabalho toma como base o sistema de ocultação de informações apresentado por Garcia (1999), porém introduz uma série de modificações com o intuito de aumentar a capacidade de inserção sem, com isso, comprometer a imperceptibilidade e a probabilidade de ocorrência de erros de detecção.

1.3. Organização da Dissertação

Esta dissertação está organizada em seis capítulos e um apêndice, da maneira mostrada a seguir. No Capítulo 1, faz-se uma breve introdução ao tema desenvolvido ao longo deste trabalho, relatando os motivos que levaram ao seu desenvolvimento e os objetivos traçados.

O Capítulo 2 descreve em detalhes o modelo psicoacústico utilizado, bem como explica como o sistema auditivo humano percebe os sons.

A técnica de espalhamento espectral é apresentada no Capítulo 3, onde ela é tratada com detalhes. Também nesse capítulo são feitas algumas considerações básicas a respeito de processamento digital de sinais.

No Capítulo 4, são abordados os métodos de compressão de dados sem perdas aplicados no sistema proposto. Exemplos de uso desses métodos também são mostrados.

O Capítulo 5 revela a proposta de trabalho e a metodologia seguida durante o desenvolvimento desta dissertação.

Os resultados e conclusões são discutidos no Capítulo 6, que também traz sugestões para trabalhos futuros.

No Apêndice A, o código corretor de erros utilizado é brevemente descrito.

2. Modelagem Perceptual

A modelagem perceptual consiste em utilizar um modelo matemático que representa a maneira como o sistema auditivo humano (em inglês, *human auditory system* ou HAS) percebe os sons a fim de eliminar componentes sonoras irrelevantes nos algoritmos de compressão de áudio. No caso de esteganografia, a modelagem perceptual é utilizada para tornar o sinal portador da informação transparente, isto é, imperceptível ao HAS, quando inserido no sinal hospedeiro.

Existem basicamente duas abordagens diferentes no que tange à modelagem perceptual: a fisiológica e a psicoacústica. A primeira deriva-se do conhecimento fisiológico e anatômico do HAS; a segunda diz respeito a como o HAS percebe os sons e é baseada em testes subjetivos de audição (Jehan, 2005; Pohjalainen, 2007). Este trabalho compreende apenas a abordagem psicoacústica, uma vez que é a mais utilizada na literatura por ser menos complexa que a abordagem fisiológica. Antes de discutir alguns conceitos psicoacústicos importantes e de apresentar o modelo psicoacústico utilizado, porém, será feita uma breve descrição de como funciona a percepção dos sons no HAS.

2.1. Sistema Auditivo Humano

O sistema auditivo humano (HAS) é formado basicamente por duas regiões de processamento: a periférica e a neural. O processamento dos sons começa na região periférica, isto é, o ouvido, e segue para a região neural através da cóclea, no ouvido interno, até o córtex auditivo, no cérebro (Pohjalainen, 2007).

O ouvido divide-se em três partes fundamentais, conforme ilustra a Figura 2.1. O ouvido externo é composto pelo canal auditivo e pelo tímpano, membrana que o separa do ouvido médio e que transmite a ele as oscilações do ar. O ouvido médio funciona como um transdutor que converte essas oscilações do ar em oscilações apropriadas para o ouvido interno por meio de três ossículos (conhecidos popularmente como martelo, bigorna e

estribo) que transmitem essas vibrações através da membrana da janela oval. O ouvido interno inicia o processamento neural dos sons na cóclea, que é uma estrutura em forma de caracol preenchida por fluidos, ao longo da qual se encontra a membrana basilar. A membrana basilar possui inúmeras células ciliadas que vibram com as oscilações que chegam através da janela oval. Essas vibrações produzem pulsos elétricos que são transmitidos ao cérebro por meio do nervo auditivo (Jehan, 2005; Pohjalainen, 2007).

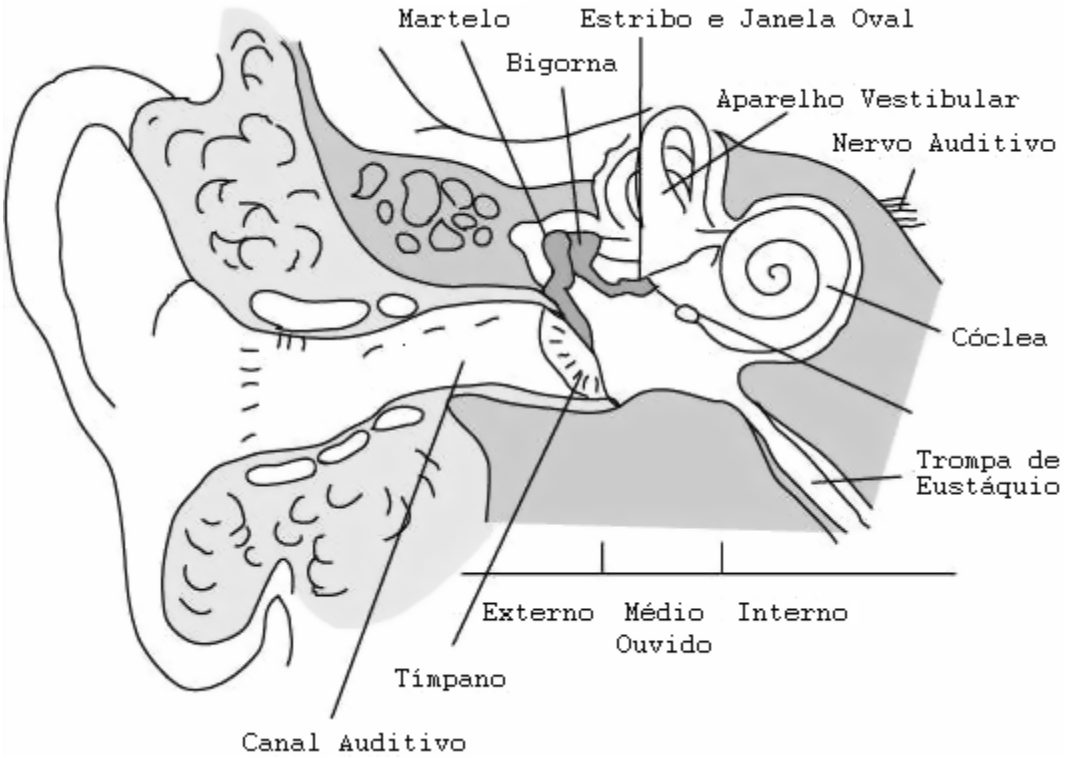


Figura 2.1 – Estrutura do sistema auditivo periférico (Pohjalainen, 2007).

Comparado ao processamento auditivo periférico, o processamento auditivo neural ainda é pouco compreendido (Pohjalainen, 2007). Visto que o tratamento detalhado do funcionamento do HAS não faz parte do escopo deste trabalho, apenas algumas características do processamento no nível neural serão mostradas. Segundo um grupo de psicólogos alemães (Pohjalainen, 2007), existem quatro princípios básicos que governam o processamento auditivo neural:

- Princípio da proximidade – a proximidade dos elementos sonoros nos domínios tempo e/ou freqüência favorece seu agrupamento em um mesmo fluxo auditivo;
- Princípio da similaridade – sons com timbres similares tendem a ser agrupados no mesmo fluxo auditivo;

- Princípio do encerramento – mecanismo que completa a percepção de um som temporariamente sobreposto por outro para deduzir sua continuação durante os períodos de sobreposição;
- Princípio do destino comum – diferentes partes do espectro sonoro que se alteram da mesma maneira e ao mesmo tempo, seja em freqüência ou em amplitude, tendem a ser agrupados no mesmo fluxo auditivo.

O HAS possui a capacidade de perceber separadamente os sons que compõem um determinado estímulo auditivo. Músicos, e principalmente maestros, têm essa capacidade extremamente desenvolvida. Quando um maestro está regendo uma orquestra, ele consegue distinguir, enquanto a música é tocada, um ou mais instrumentos individualmente. O som de cada um desses instrumento representa um fluxo auditivo. Maiores detalhes sobre o assunto podem ser encontrados em Sussman et al.[1] (1999).

2.2. Princípios Psicoacústicos

Psicoacústica é a ciência que estuda a maneira como o sistema auditivo humano (HAS) percebe os sons, bem como a modelagem matemática que caracteriza o funcionamento da percepção auditiva humana. Essa área do conhecimento vem sendo estudada há décadas, mas foi com o desenvolvimento dos codificadores de áudio que ela teve um crescimento mais significativo e ainda continua crescendo.

A maioria dos codificadores de áudio atuais utiliza-se da psicoacústica para alcançar maiores taxas de compressão sem comprometer a qualidade do áudio (Painter e Spanias, 2000). Quando um codificador realiza uma análise psicoacústica em determinado sinal de áudio, ele busca por componentes desse sinal que não são relevantes, isto é, aquelas que o HAS não consegue perceber. Isso está associado a uma série de fenômenos que serão discutidos ao longo deste capítulo.

O primeiro conceito a ser definido é o de nível de intensidade sonora, também conhecido como nível de pressão sonora (em inglês, *sound pressure level* ou SPL). O SPL é uma métrica internacional padronizada que quantifica a intensidade de um estímulo sonoro. Sua unidade de medida é o decibel (dB), normalmente notado como dB_{SPL}. O SPL é definido pela Equação (2.1), onde p é a pressão sonora do estímulo em Pascals (Pa) – equivalente a

[1] SUSSMAN, Elyse; RITTER, Walter; VAUGHAN Jr., Herbert G. An investigation of the auditory streaming effect using event-related brain potentials. **Psychophysiology**, Cambridge, n. 36, pp. 22-34, Jul. 1999.

Newtons por metro quadrado (N/m^2) – e p_0 é o nível padrão de referência 20 μPa ou $2x10^{-5}$ N/m^2 (Painter e Spanias, 2000).

$$20\log_{10}\left(p / p_0\right) \tag{2.1}$$

Seguindo o conceito de SPL, tem-se a definição de outro conceito importante, que é o de limiar absoluto de audição (em inglês, *absolute threshold of hearing* ou ATH). Esse limiar caracteriza a mínima quantidade de energia que um tom puro, isto é, um sinal de áudio composto por uma só freqüência, deve conter para que ele seja percebido por um ouvinte em um ambiente silencioso (Painter e Spanias, 2000). Também pode ser definido como o menor SPL em que um tom puro é audível em cada freqüência do espectro (Pohjalainen, 2007). Assume-se que o ATH pertença a um ouvinte jovem e com audição perfeita (Painter e Spanias, 2000; Pohjalainen, 2007). O ATH depende da freqüência e é conseqüência das respostas dos ouvidos externo e médio (Jehan, 2005). Sua relação com a freqüência foi quantificada por Harvey Fletcher[2] (1940 apud Painter e Spanias, 2000). Essa relação foi aproximada por uma função não-linear por Ernst Terhardt[3] (1979 apud Painter e Spanias, 2000), como mostra a Equação (2.2).

$$T_q\left(f\right) = 3,64\left(10^{-3} f\right)^{-0,8} - 6,5e^{-0,6\left(10^{-3} f - 3,3\right)^2} + 10^{-3}\left(10^{-3} f\right)^4 \tag{2.2}$$

Atualmente, existe o padrão ISO-226, de 2003, que define um conjunto de curvas (Pohjalainen, 2007), chamadas isoaudíveis, que se assemelham às curvas de Fletcher-Munson, publicadas por esses dois cientistas dos Laboratórios Bell em 1933 no artigo intitulado "*Loudness, its definition, measurement and calculation*"[4]. A Figura 2.2 mostra as curvas isoaudíveis do padrão ISO-226 para alguns valores de SPL juntamente com a curva do ATH gerada com a Equação (2.2) (Pohjalainen, 2007).

Conforme foi dito no início desta seção, existem outros fenômenos, além do ATH, associados ao HAS. Eles são conhecidos como bandas críticas, mascaramento em freqüência e mascaramento no tempo e serão explicados a seguir.

---

[2] FLETCHER, Harvey. Auditory Patterns. **Reviews of Modern Physics**, [S.l.], v. 12, n. 1, pp. 47-55, May 1940.
[3] TERHARDT, Ernst. Calculating Virtual Pitch. **Hearing Research**, [S.l.], v. 1, n. 2, pp. 155-182, Mar. 1979.
[4] FLETCHER, H.; MUNSON, W. A. Loudness, its definition, measurement and calculation. **Journal of the Acoustical Society of America**, [S.l.], v. 5, n. 2, pp. 82-108, Oct. 1933.

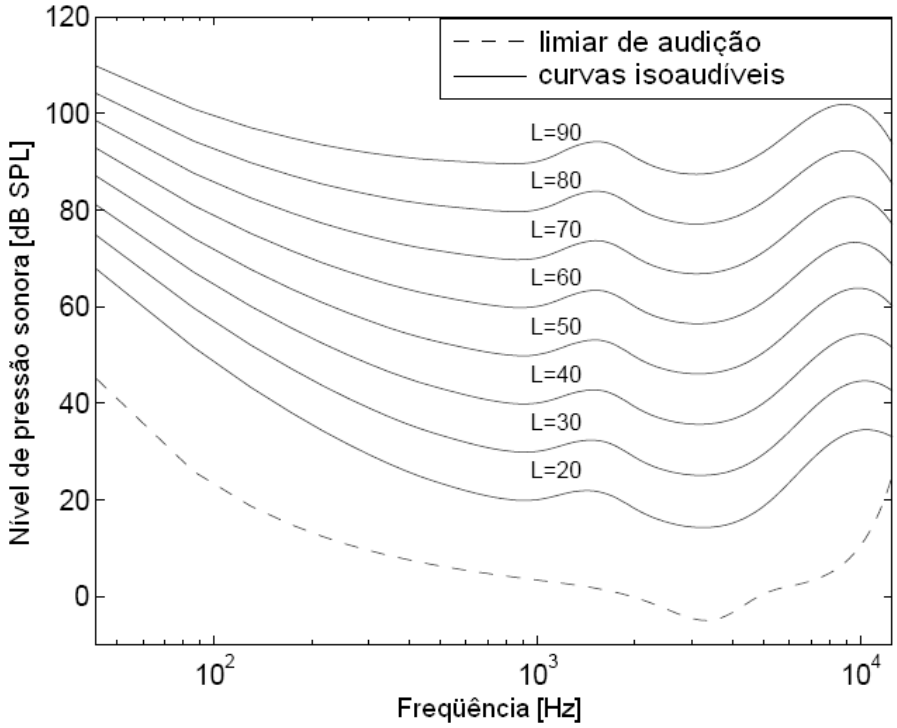


Figura 2.2 – Curvas isoaudíveis e limiar absoluto de audição (Pohjalainen, 2007).

2.2.1. Escala de Bandas Críticas

A cóclea, devido às suas características mecânicas (larga e rígida na base, mais estreita e menos rígida no ápice), funciona como um banco de filtros (Jehan, 2005) passa-faixa com alto grau de superposição (Painter e Spanias, 2000). Além disso, a largura de banda desses filtros não é uniforme, uma vez que ela aumenta à medida que a freqüência cresce (Painter e Spanias, 2000).

Quando dois tons puros de freqüências diferentes soam suavemente, significa que eles se encontram em uma mesma banda crítica (Jehan, 2005). Abaixo dos 500 Hz, a largura das bandas críticas é aproximadamente constante, em torno de 100 Hz. Para freqüências mais altas, seu aumento é proporcional à freqüência central (cerca de 20% da mesma (Jehan, 2005)), chegando a 1300 Hz para uma freqüência central de 7 kHz, conforme a Tabela 2.1.

A escala de bandas críticas, também conhecida como escala Bark, foi criada para facilitar a identificação e a manipulação dessas bandas de freqüências. A unidade de medida da escala é o Bark, que corresponde à distância de uma banda crítica (Painter e Spanias, 2000). A aproximação mais utilizada para realizar a conversão de Hz para Barks é dada pela Equação (2.3).

$$z(f) = 13\tan^{-1}\left(\frac{0,76f}{1000}\right) + 3,5\tan^{-1}\left(\left(\frac{f}{7500}\right)^2\right) \quad (2.3)$$

Tabela 2.1 – As primeiras 24 bandas críticas da escala (Painter e Spanias, 2000).

Banda Crítica (Barks)	Freqüência Central (Hz)	Freqüências Limites (Hz)	Largura de Banda (Hz)
1	50	20-100	80
2	150	100-200	100
3	250	200-300	100
4	350	300-400	100
5	450	400-510	110
6	570	510-630	120
7	700	630-770	140
8	840	770-920	150
9	1000	920-1080	160
10	1175	1080-1270	190
11	1370	1270-1480	210
12	1600	1480-1720	240
13	1850	1720-2000	280
14	2150	2000-2320	320
15	2500	2320-2700	380
16	2900	2700-3150	450
17	3400	3150-3700	550
18	4000	3700-4400	700
19	4800	4400-5300	900
20	5800	5300-6400	1100
21	7000	6400-7700	1300
22	8500	7700-9500	1800
23	10500	9500-12000	2500
24	13500	12000-15500	3500

2.2.2. Mascaramento em Freqüência

O mascaramento em freqüência, também chamado de mascaramento simultâneo, ocorre quando dois ou mais estímulos sonoros chegam ao sistema auditivo humano (HAS) e pelo menos um deles não é percebido, ou é fracamente percebido. A diferença de nível de intensidade sonora (SPL) entre o som mascarador e o som mascarado é conhecido por nível de mascaramento (Garcia, 1999) ou relação sinal-mascaramento (em inglês *signal-to-mask ratio* ou SMR) (Cvejic, 2004). Uma explicação mais precisa diz que a presença de um ruído ou de um tom com intensidade suficiente para criar uma forte excitação na membrana basilar,

em uma região de banda crítica, bloqueia a percepção de um estímulo mais fraco (Painter e Spanias, 2000).

Embora situações de mascaramento simultâneo arbitrárias possam ser complexas, considera-se apenas três tipos de mascaramento simultâneo no tocante à conformação de distorções causadas pela codificação do áudio: ruído mascarando um tom (em inglês, *noise-masking-tone* ou NMT), um tom mascarando ruído (em inglês, *tone-masking-noise* ou TNM) ou ruído mascarando ruído (em inglês, *noise-masking-noise* ou NMN) (Painter e Spanias, 2000). No primeiro caso, um ruído de banda estreita (tendo um Bark de largura, por exemplo) mascara um tom que se encontra na mesma banda crítica, dado que a intensidade desse tom esteja abaixo de um determinado limiar relacionado à intensidade e à freqüência central daquele ruído. No segundo caso, um tom puro cuja freqüência é a freqüência central de uma banda crítica mascara um ruído com largura de banda menor ou igual à dessa banda e que esteja situado entre os limites inferior e superior da mesma, dado que o espectro do ruído não ultrapasse um determinado limiar associado à intensidade daquele tom e à freqüência central da banda crítica (Painter e Spanias, 2000). Esses fenômenos podem ser observados nas Figuras 2.3a e 2.3b, respectivamente.

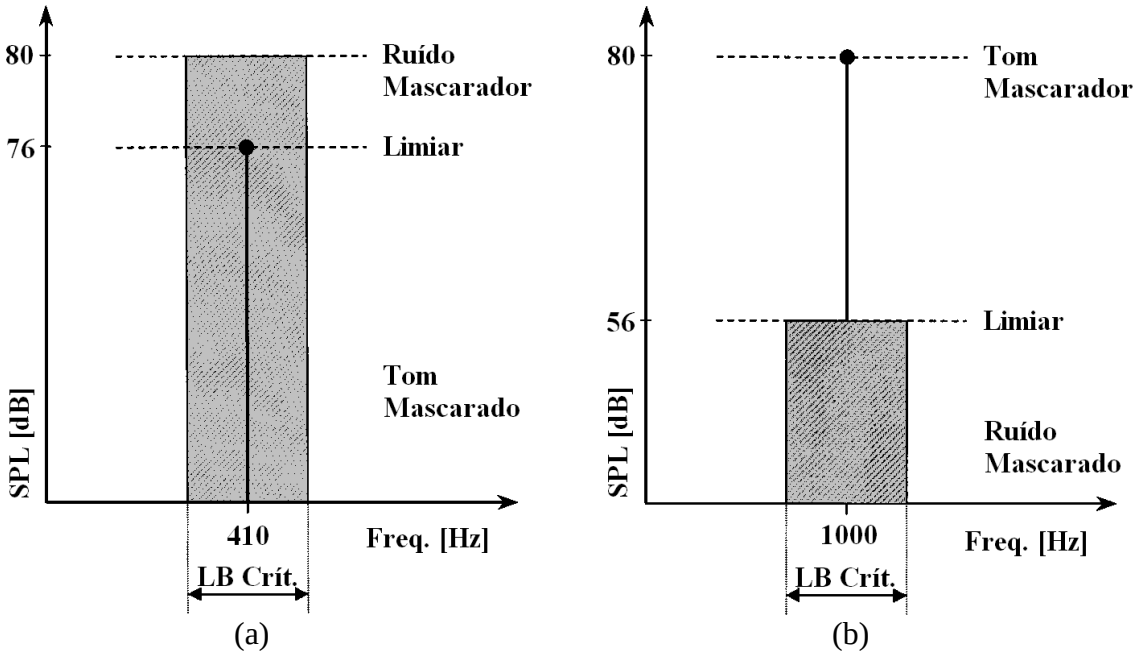


Figura 2.3 – Exemplos que ilustram os fenômenos de (a) NMT e (b) TNM (Painter e Spanias, 2000).

Observa-se nos exemplos da Figura 2.3 que para (a) um tom puro de 410 Hz a 76 dB_{SPL} ser mascarado por um ruído (NMT), apenas 4 dB_{SPL} de diferença entre eles é

necessária, enquanto que para (b) um ruído de 56 dB_{SPL} ser mascarado por um tom puro de 1 kHz (TMN), é necessária uma diferença de 24 dB_{SPL}. Isso mostra que o nível de mascaramento associado a um tom mascarador é significativamente maior do que o nível de mascaramento associado a um ruído mascarador, o que ocorre devido à sensibilidade do HAS em relação a ruído aditivo (Cvejic, 2004).

No terceiro caso, isto é, na situação NMN, um ruído de banda estreita mascara um outro ruído de banda estreita. Essa situação é muito mais difícil de ser caracterizada do que NMT ou TNM, uma vez que as relações de fase entre o mascarador e o mascarado acabam confundindo o levantamento da influência que o primeiro exerce sobre o último (Painter e Spanias, 2000).

Apesar de normalmente se utilizarem as simplificações discutidas anteriormente, os efeitos do mascaramento em freqüência não ocorrem apenas dentro de uma banda crítica. Há um espalhamento desses efeitos para as bandas vizinhas (Jehan, 2005; Painter e Spanias, 2000). Tal espalhamento, que ocorre ao longo da membrana basilar, pode ser modelado calculando o espectro de potência do sinal sonoro, convertendo-o para a escala de bandas críticas e, então, convoluindo o resultado com uma função de espalhamento (Pohjalainen, 2007). Esse procedimento será explicado em maiores detalhes mais adiante, porém um modelo popular de função de espalhamento é dado pela Equação (2.4), onde *z* é o valor da banda crítica em Barks.

$$B_{dB}(z) = 15{,}81 + 7{,}5(z + 0{,}474) - 17{,}5\sqrt{1 + (z + 0{,}474)^2} \qquad (2.4)$$

A curva gerada por (2.4) assemelha-se a uma parábola com concavidade para baixo, mas com declividades diferentes antes (aproximadamente 25 dB/Bark) e depois (aproximadamente -10 dB/Bark) do ponto de inflexão. A Figura 2.4 mostra a função de espalhamento $B_{dB}(z)$ utilizando-se nove pontos.

2.2.3. Mascaramento no Tempo

O fenômeno do mascaramento no tempo, mascaramento temporal ou ainda mascaramento não-simultâneo acontece normalmente antes e depois da ocorrência de um som mascarador. O primeiro fenômeno denomina-se pré-mascaramento e o segundo, pós-mascaramento. Ainda há uma parcela do mascaramento temporal que é simultânea, assim

como acontece com o mascaramento em freqüência. Essa parcela simultânea pode ocorrer devido às relações de fase entre os dois sinais sonoros (Painter e Spanias, 2000), ou seja, entre o estímulo mascarador e o som mascarado.

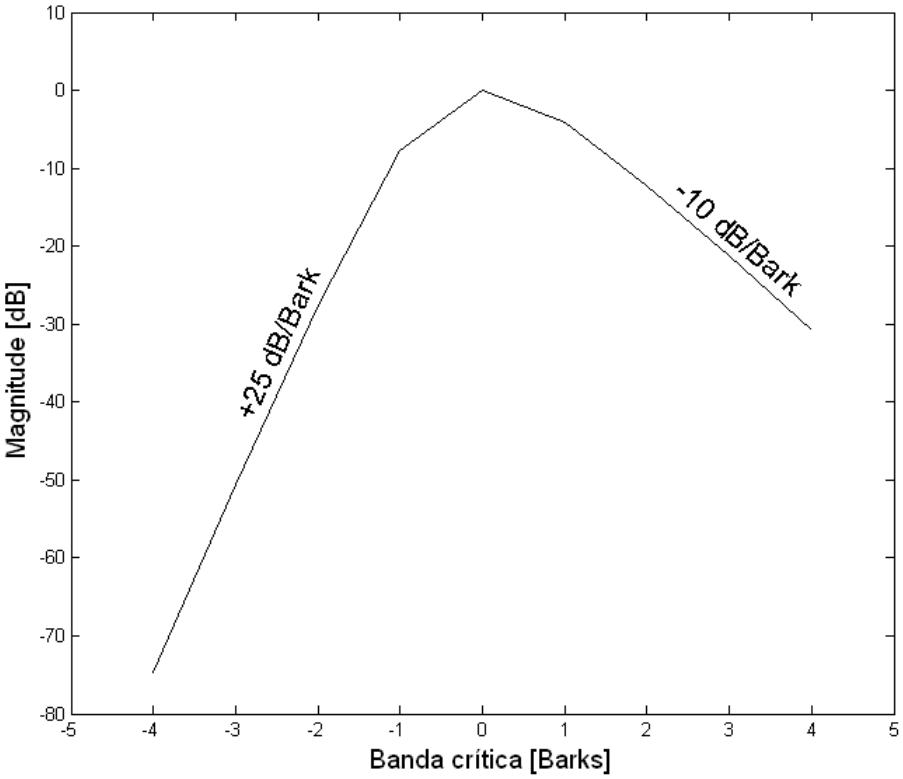


Figura 2.4 – Função de espalhamento da membrana basilar.

O efeito do pré-mascaramento ainda não foi pesquisado a ponto de se ter uma conclusão definitiva à respeito de sua ocorrência (Jehan, 2005) ou do tempo de duração (Painter e Spanias, 2000). Existem autores[5] (Zwicker e Fastl, 1999 apud Jehan, 2005) que, por meio de testes com rajadas de ruídos, revelaram que o fenômeno do pré-mascaramento dura cerca de 20 ms. Outro autor[6] (Moore, 1995 apud Pohjalainen, 2007) afirma que esse efeito depende muito do treinamento do ouvinte, isto é, indivíduos treinados apresentam um período de pré-mascaramento pequeno ou até mesmo nulo.

Por outro lado, a respeito do pós-mascaramento têm-se informações mais precisas e em maior quantidade. O tempo aproximado de duração desse fenômeno é discutido na literatura como sendo de no mínimo 50 ms (Painter e Spanias, 2000; Cvejic, 2004) e de no máximo 200 ms (Cvejic, 2004; Jehan, 2005; Pohjalainen, 2007). Esse tempo depende de

[5] ZWICKER, E.; FASTL, H. **Psychoacoustics:** Facts and Models**.** Berlin: Springer Verlag, 1999. 428 pp.
[6] MOORE, Brian C.J. (Ed.). **Hearing.** San Diego: Academic Press, 1995. 468 pp.

alguns fatores, incluindo a freqüência, a intensidade e a duração do estímulo mascarador (Painter e Spanias, 2000). O efeito do pós-mascaramento pode ser considerado um tipo de "efeito *ringing*" e ele contribui de maneira importante na percepção de ritmo (Jehan, 2005). A Figura 2.5 ilustra, esquematicamente, os efeitos do mascaramento temporal.

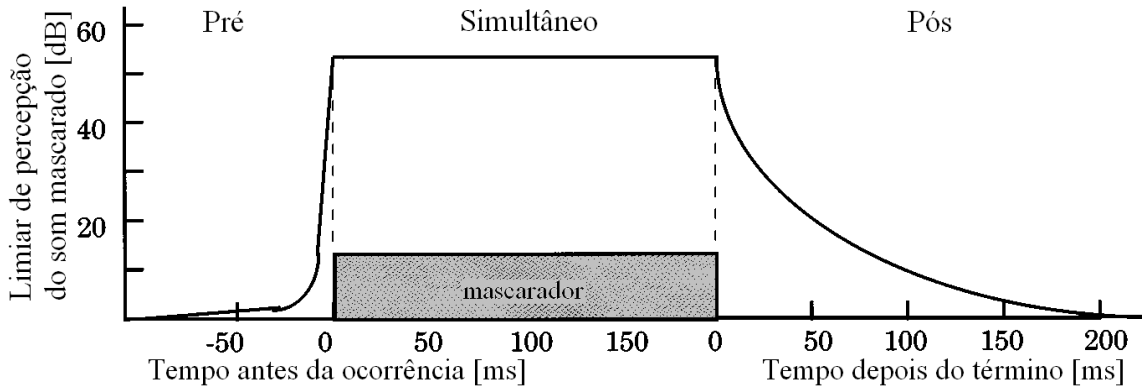


Figura 2.5 – Efeitos do mascaramento temporal no limiar de percepção de um estímulo sonoro mascarado (Painter e Spanias, 2000).

2.3. Modelo Perceptual Utilizado

O modelo perceptual utilizado na implementação do sistema de esteganografia deste trabalho é baseado em modelagem psicoacústica. Tal modelo foi utilizado em (Garcia, 1999). Há uma série de conceitos atrelados a esse modelo psicoacústico, os quais serão apresentados à medida que ele será desenvolvido.

2.3.1. Espectro de Potência

O espectro de potência de um determinado sinal mostra a potência que está associada a cada uma das freqüências que compõem o espectro desse sinal. Dependendo da transformada utilizada para convertê-lo ao domínio freqüência, o cálculo do espectro de potência será diferente. Por esse motivo, o espectro de potência será representado aqui apenas por $P(\omega)$.

O primeiro passo, depois de se obter $P(\omega)$, é calcular o efeito do espalhamento que o mascaramento simultâneo causa nas bandas críticas vizinhas. Como já foi dito anteriormente, o espectro de potência deve ser convertido para a escala de bandas críticas e, então, o resultado deve ser convoluído com uma função de espalhamento (Pohjalainen, 2007). A

conversão do espectro de potência é feita com a Equação (2.5), que não é nada mais do que a energia por banda crítica. O índice *z* representa as bandas críticas. O número de bandas críticas (z_t) depende da freqüência de amostragem (f_s) do sinal. Para áudio com qualidade de CD, isto é, f_s = 44100 Hz, z_t é igual a 25 (Painter e Spanias, 2000). Os limites do somatório LF_z e UF_z representam as freqüências inferior e superior da banda crítica *z*, respectivamente.

$$E_z = \sum_{\omega=LF_z}^{UF_z} P(\omega), \qquad \text{para } 1 \le z \le z_t \tag{2.5}$$

A Figura 2.6 mostra o espectro de potência $P(\omega)$ e a respectiva energia por banda crítica E_z de um pequeno trecho de um sinal de áudio. A freqüência no gráfico foi limitada a 5500 Hz para facilitar a visualização das primeiras bandas críticas.

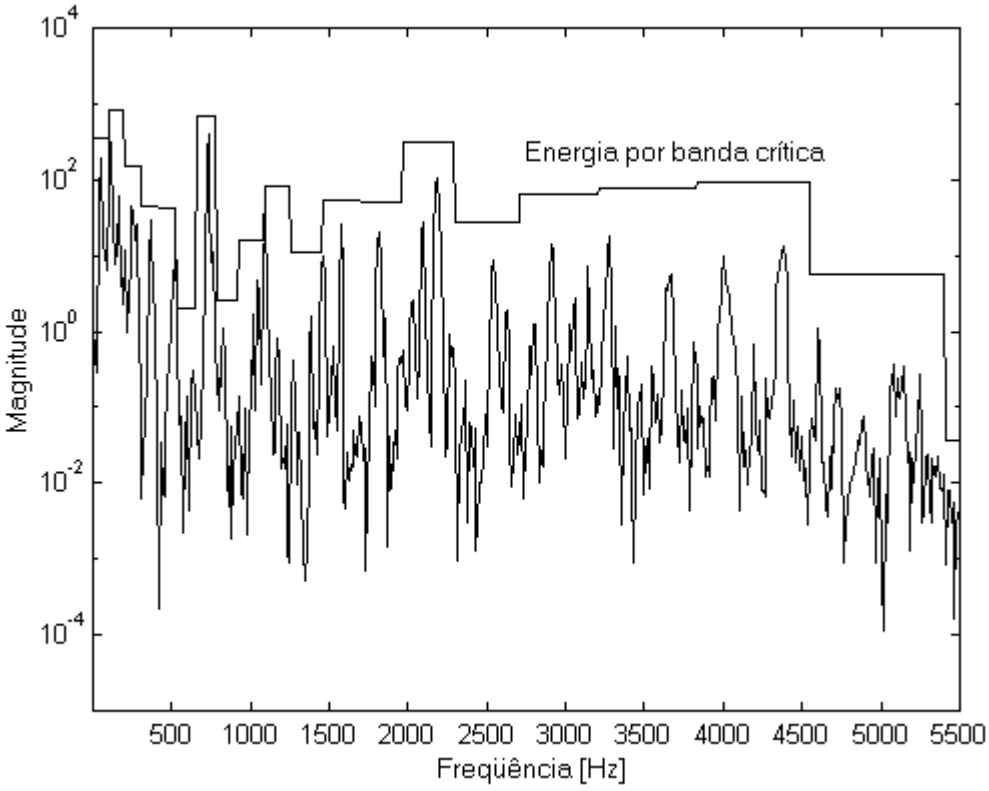


Figura 2.6 – Espectro de potência e energia por banda crítica de um sinal de áudio (Garcia, 1999).

Por fim, calcula-se a convolução da Equação (2.4) com a Equação (2.5) para se obter o espalhamento do mascaramento através das bandas críticas, conforme a Equação (2.6).

Assim como em (2.5), o índice z representa as bandas críticas e z_t o número de bandas críticas presentes no sinal.

$$SM_z = B_z * E_z, \quad \text{para } 1 \leq z \leq z_t \tag{2.6}$$

O cálculo de um verdadeiro espalhamento deveria ser feito com cada componente das bandas críticas, mas o uso de E_z é uma boa aproximação dado o propósito desse modelo psicoacústico. A Equação (2.6) pode ser interpretada como sendo a energia por banda crítica, considerando o espalhamento através das bandas (Garcia, 1999). A Figura 2.7 mostra o espectro de potência $P(\omega)$ e o respectivo espalhamento SM_z de um pequeno trecho de um sinal de áudio. Novamente, a freqüência no gráfico foi limitada a 5500 Hz para facilitar a visualização das primeiras bandas críticas.

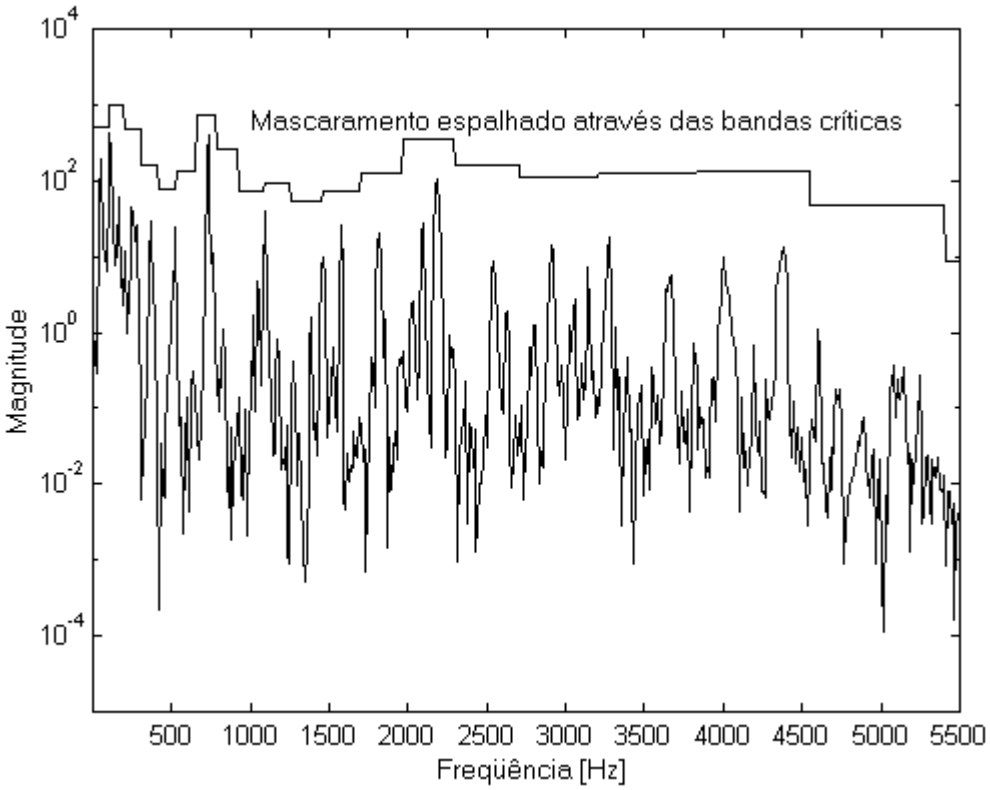


Figura 2.7 – Espectro de potência e mascaramento espalhado através das bandas críticas de um sinal de áudio (Garcia, 1999).

2.3.2. Cálculo do Limiar de Mascaramento

O limiar de mascaramento é uma medida semelhante ao limiar de audição, porém levando em consideração os efeitos do mascaramento. Também conhecido por mínima distorção perceptível (em inglês, *just noticeable distortion* ou JND) (Painter e Spanias, 2000; Cvejic, 2004), o limiar de mascaramento pode ser calculado seguindo-se alguns passos.

O primeiro consiste em classificar os sinais mascaradores como sendo ruído ou tom, a fim de determinar entre as situações de ruído mascarando um tom (NMT) ou de um tom mascarando ruído (TMN). Isso pode ser feito com as Equações (2.7) e (2.8), que representam a medida do achatamento espectral (em inglês, *spectral flatness measure* ou SFM), dada em dB, e o fator de tonalidade, respectivamente. A SFM é usada para determinar se um trecho de áudio sendo analisado se parece mais com ruído ou com um tom puro (Garcia, 1999). O fator de tonalidade, ou coeficiente de tonalidade, varia de zero a um e é utilizado na escolha de um índice de mascaramento apropriado. Quando α se aproxima de zero, considera-se que aquele trecho de áudio se parece mais com ruído. Quando α se aproxima de um, considera-se o trecho mais parecido com um tom puro. Na Equação (2.7), μ_g e μ_a são, respectivamente, as médias geométrica e aritmética da energia por banda crítica E_z.

$$SFM_{dB} = 10\log_{10}\frac{\mu_g}{\mu_a} \tag{2.7}$$

$$\alpha = \min\left(\frac{SFM_{dB}}{-60}, 1\right) \tag{2.8}$$

O índice de mascaramento é definido como sendo (14,5 + *z*) dB abaixo do mascaramento espalhado através das bandas críticas SM_z, no caso de TMN, onde *z* é a freqüência do tom puro dada em Barks, e como sendo 5,5 dB abaixo do SM_z, no caso de NMT, independentemente da freqüência (Garcia, 1999). A Figura 2.8 mostra uma situação em que quatro tons puros (linhas contínuas) de 70, 250, 1000 e 4000 Hz, alteram a curva de percepção (linha tracejada) do sistema auditivo humano (HAS).

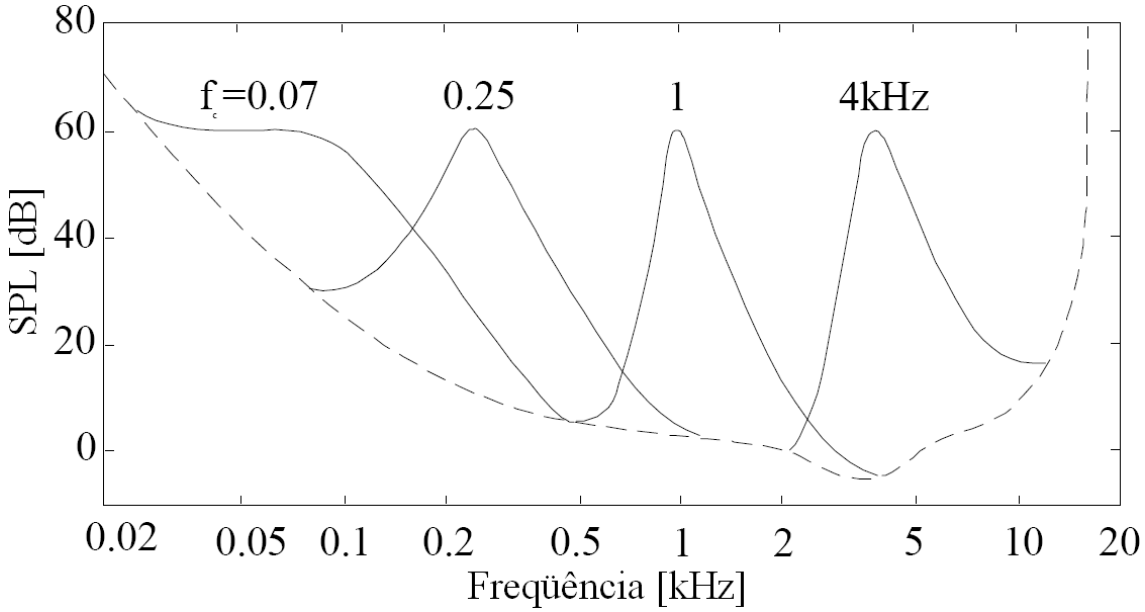


Figura 2.8 – Mascaramento simultâneo por quatro tons puros (linhas contínuas) e limiar absoluto de audição (linha tracejada) (Garcia, 1999).

A escolha do índice de mascaramento é feita com a Equação (2.9). É importante notar que o índice *z*, assim como nas equações anteriores, representa as bandas críticas e que z_t é o número de bandas críticas. Observando (2.9) atentamente, percebe-se que ela representa perfeitamente a definição de índice de mascaramento.

$$O_z = \alpha(14{,}5 + z) + (1-\alpha)5{,}5, \quad \text{para } 1 \le z \le z_t \tag{2.9}$$

O resultado de (2.9), que é dado em dB, é escalonado por um fator de correção para simular a desconvolução da função de espalhamento (Painter e Spanias, 2000) da Equação (2.4) e, então, é subtraído do mascaramento espalhado através das bandas críticas para se obter o limiar de mascaramento inicial, conforme a Equação (2.10).

$$T_z = 10^{\log_{10} SM_z - \frac{O_z}{10}}, \quad \text{para } 1 \le z \le z_t \tag{2.10}$$

A Figura 2.9 mostra o espectro de potência $P(\omega)$ e o respectivo limiar de mascaramento T_z de um pequeno trecho de um sinal de áudio. Assim como antes, a visualização da freqüência foi limitada a 5500 Hz para facilitar a visualização das primeiras bandas críticas.

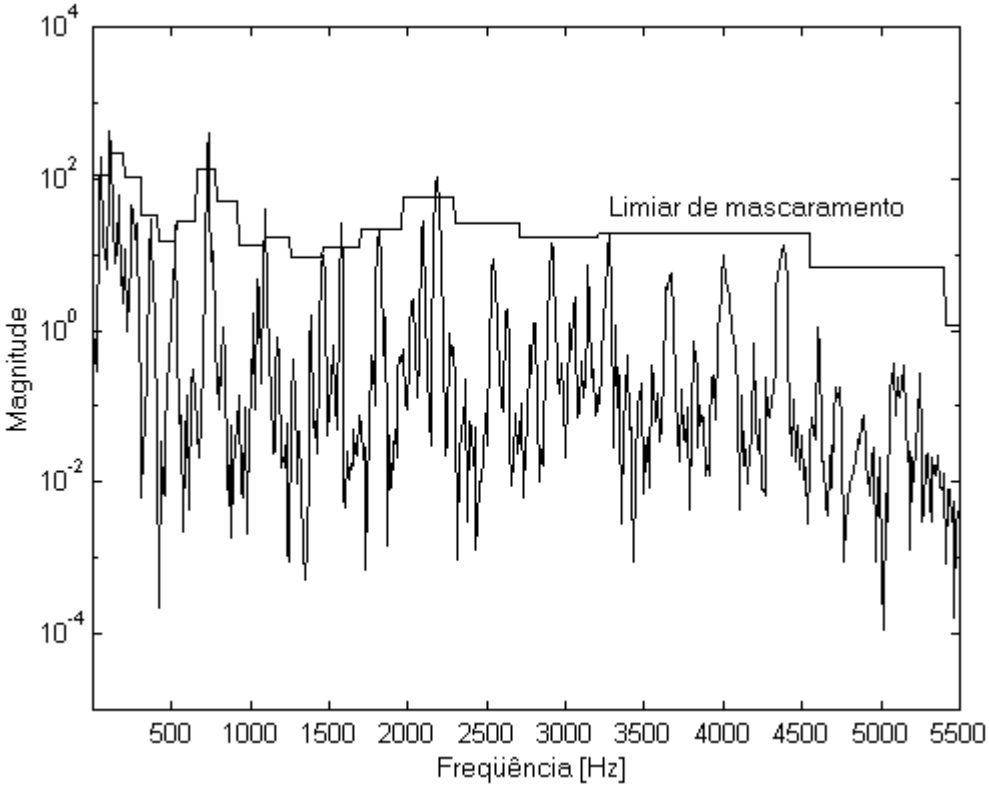


Figura 2.9 – Espectro de potência e limiar de mascaramento de um sinal de áudio (Garcia, 1999).

Esse limiar de mascaramento, porém, não pode ser utilizado como definitivo. O uso da função de espalhamento da Equação (2.4) aumenta o nível de energia em cada banda crítica (Garcia, 1999). Para que tal efeito seja removido, deve-se normalizar o limiar T_z. As bandas críticas mais altas têm mais componentes do que as mais baixas, isto é, são mais largas que as últimas. Isso é facilmente constatado quando se observam as Figuras 2.6, 2.7 e 2.9. A normalização, mostrada na Equação (2.11), é realizada dividindo-se o limiar de mascaramento de cada banda crítica z pelo número de componentes Q_z presentes em cada uma dessas bandas. O resultado pode ser observado na Figura 2.10.

$$T_{Nz} = \frac{T_z}{Q_z}, \quad \text{para } 1 \le z \le z_t \tag{2.11}$$

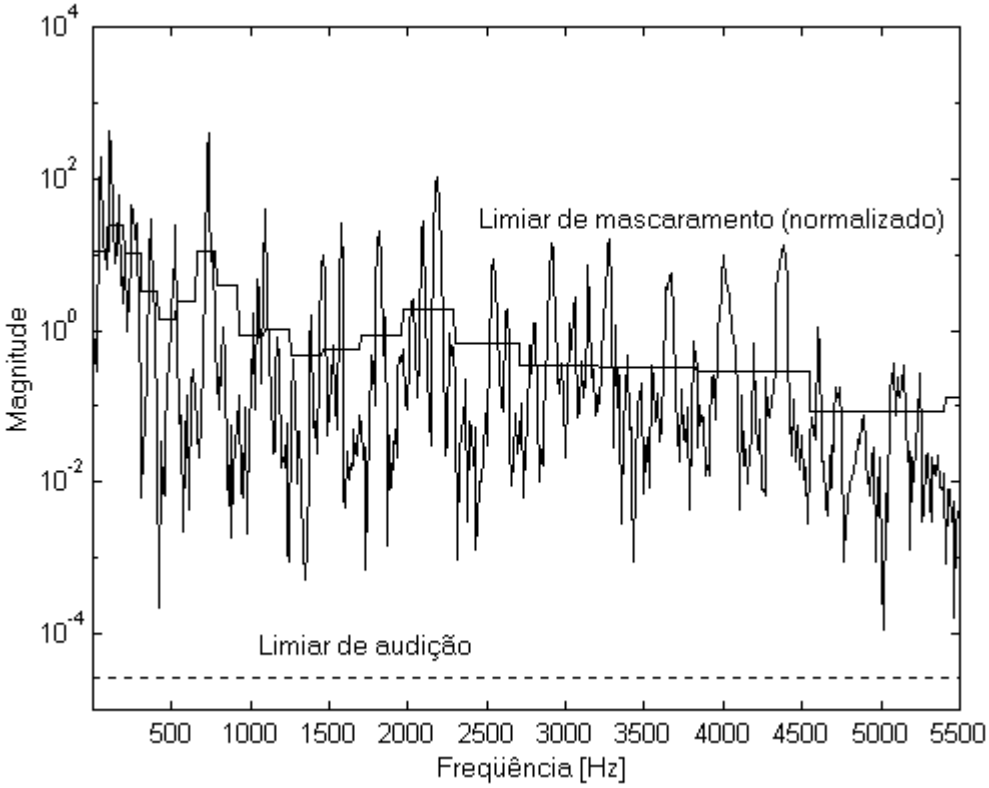


Figura 2.10 – Espectro de potência, limiar de mascaramento normalizado (linhas contínuas) e limiar de audição aproximado para o pior caso (linha tracejada) de um sinal de áudio (Garcia, 1999).

Além do limiar de mascaramento normalizado, também é possível observar na Figura 2.10 uma linha tracejada assinalada como limiar de audição. Esse limiar é uma aproximação de pior caso do limiar absoluto de audição (ATH) mostrado na Figura 2.2, sendo considerado a máxima potência do espectro de potência de um tom puro de 4 kHz. A aproximação de pior caso é utilizada pois é impossível saber em que nível de intensidade (SPL) o áudio será reproduzido (Painter e Spanias, 2000), e assumir o pior caso implica determinar que o SPL de reprodução será o mínimo perceptível ao ouvido humano. A freqüência de 4 kHz foi escolhida baseada em estudos empíricos realizados por Zwicker e Zwicker[7] (1991 apud Garcia, 1999) que mostram que a faixa mais sensível do HAS é de 2,5 a 4,5 kHz. A amplitude desse sinal senoidal deve ser de um nível de quantização, isto é, depende do número de bits *b* das amostras de áudio, conforme a Equação (2.12). Essa é a menor amplitude possível em um sinal de áudio digital.

[7] ZWICKER, E.; ZWICKER, U. T. Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System. **Journal of Audio Engineering Society**, [S.l.], v. 39, n. 3, pp. 115-126, Mar. 1991.

$$y(t) = 2^{-b} \operatorname{sen}(2\pi 4000t) \tag{2.12}$$

Finalmente, para se obter o limiar de mascaramento definitivo *T*, calcula-se a potência máxima do espectro de potência de *y*(*t*), como foi dito anteriormente. Depois disso, verifica-se, para cada banda crítica *z*, o valor máximo entre o limiar da Equação (2.11) e esse recém calculado (T_H), conforme mostra a Equação (2.13).

$$T = \max(T_{Nz}, T_H), \quad \text{para } 1 \le z \le z_t \tag{2.13}$$

2.3.3. Conformação de Ruído

O sinal portador da informação não pode ser inserido no áudio sem tratamento prévio, visto que isso acarretaria distorções perceptíveis no sinal hospedeiro. Com o intuito de reduzir, ou até mesmo de eliminar, as distorções audíveis, deve-se realizar um processo chamado conformação de ruído. Tal processo consiste em modelar o sinal portador da informação a fim de torná-lo transparente ao ouvinte comum.

Tomando como parâmetro de referência o limiar de mascaramento da Equação (2.13), compara-se com ele o sinal de áudio que receberá a informação adicional. As componentes de freqüência do áudio que ficarem abaixo do limiar podem ser consideradas irrelevantes e, conseqüentemente, eliminadas sem qualquer prejuízo na qualidade sonora. As componentes do áudio que estiverem acima do limiar de mascaramento serão mantidas, enquanto que aquelas que se encontrarem abaixo serão substituídas pelas componentes do sinal de dados. Esse procedimento ainda ajuda a não mudar consideravelmente a energia média do sinal de áudio. As Equações (2.14) e (2.15) mostram como isso pode ser feito. $X(\omega)$ e $W(\omega)$ representam os sinais de áudio e de dados no domínio da freqüência, respectivamente.

$$X_s(\omega) = \begin{cases} X(\omega) & \text{se } P(\omega) \ge T(z) \\ 0 & \text{se } P(\omega) < T(z) \end{cases} \tag{2.14}$$

$$W_s(\omega) = \begin{cases} W(\omega) & \text{se } P(\omega) < T(z) \\ 0 & \text{se } P(\omega) \ge T(z) \end{cases} \tag{2.15}$$

A conformação de ruído é feita aplicando-se um fator F_z ao sinal $W_s(\omega)$. Esse fator pode ser calculado com a expressão da Equação (2.16). Vale lembrar que cada banda crítica *z*

possui limites inferior (LF_z) e superior (UF_z) de freqüências, tal que $LF_z \leq \omega \leq UF_z$. O ganho $A \in (0;1]$ serve apenas para atenuar o sinal de dados.

$$F_z = A\frac{\sqrt{T(z)}}{\max\left(\left|W_s(\omega)\right|\right)}, \quad \text{para } 1 \leq z \leq z_t \tag{2.16}$$

O denominador da Equação (2.16) denota o valor máximo do espectro de magnitude do sinal $W_s(\omega)$ em cada uma das bandas críticas. Por último, a informação é finalmente inserida no sinal de áudio da maneira mostrada na Equação (2.17).

$$X_m(\omega) = X_s(\omega) + F_z W_s(\omega) \tag{2.17}$$

3. Técnica de Espalhamento Espectral

As técnicas de espalhamento espectral (em inglês, *spread spectrum* ou SS) foram inicialmente aplicadas em sistemas de comunicação militares (Sklar, 2001). O uso de SS oferece maiores robustez e segurança à comunicação, pois permite a redução ou a supressão de interferências, sejam elas naturais ou intencionais (*jamming*). Além disso, a informação pode ser transmitida através de um sinal de baixa potência, ocultando-a em meio ao ruído no canal e dificultando sua detecção por pessoas não autorizadas (Proakis, 1995).

Para que um sistema de comunicação se caracterize como sendo SS, ele deve atender a algumas especificações (Sklar, 2001):

- O sinal ocupa uma banda muito maior do que a mínima necessária para a transmissão da informação;
- O espalhamento é feito por meio de um sinal de espalhamento, que deve ser independente dos dados;
- A recuperação da informação original é realizada calculando-se a correlação do sinal recebido com uma cópia sincronizada do sinal de espalhamento utilizado na codificação.

Há três técnicas básicas de SS (Sklar, 2001): *direct sequence* (DS), *frequency hopping* (FH) e *time hopping* (TH). Existem também técnicas híbridas, que utilizam duas ou mais técnicas básicas em conjunto. Visto que neste trabalho será utilizada a técnica DS, as outras técnicas citadas anteriormente não serão abordadas. Antes, porém, de se entrar no cerne deste capítulo, que é justamente a técnica *Direct Sequence Spread Spectrum* (DSSS), serão introduzidos alguns conceitos básicos relativos a processamento digital de sinais (em inglês, *digital signal processing* ou DSP) e a seqüências pseudoaleatórias.

3.1. Conceitos preliminares

A fim de se compreender o que será exposto mais adiante neste trabalho, alguns conceitos preliminares sobre DSP serão apresentados, como as questões de segmentação e de enjanelamento de um sinal, além de algumas transformadas. Também serão discutidas o que são, como são geradas e porque utilizar seqüências pseudoaleatórias como sinais de espalhamento em um sistema SS.

3.1.1. Segmentação e Janelas

Um sinal digital, seja ele qual for (áudio, imagens etc.), em geral é muito extenso do ponto de vista do número de amostras. Por exemplo, um sinal de áudio amostrado com uma freqüência de 48 kHz possui 48000 amostras para cada segundo de duração. A análise dessa quantidade imensa de dados torna-se impraticável se for feita de uma vez só. A fim de resolver esse problema, utiliza-se uma técnica conhecida como segmentação, que trata de dividir o sinal em segmentos (também chamados de quadros) menores. Na prática, isso é feito multiplicando-se o sinal a ser analisado, no domínio do tempo, por sucessivas janelas retangulares, definidas na Equação (3.1) (Proakis e Manolakis, 1996; Oppenheim e Willsky, 1997), onde M é o tamanho da janela em número de amostras.

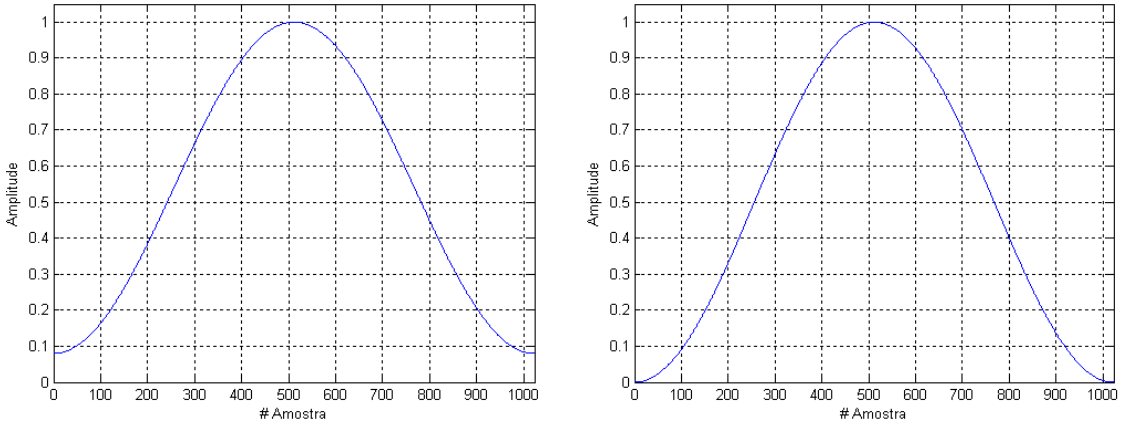
$$w(n) = \begin{cases} 1, & \text{para } 0 \le n < M \\ 0, & \text{caso contrário} \end{cases} \tag{3.1}$$

O problema da janela retangular da Equação (3.1) é que no domínio da freqüência ela se transforma em uma função $\text{sen}(x)/x$, a conhecida função $\text{sinc}(x)$, que possui infinitas componentes. Como é impossível representar digitalmente um número infinito de componentes, acaba-se perdendo as componentes de freqüências maiores, implicando uma distorção temporal do sinal. Para diminuir a perda de componentes, chamada de vazamento espectral, foram criadas outros tipos de janelas. As mais comuns utilizadas em processamento de áudio são a de Hamming e a de Hann, respectivamente definidas nas Equações (3.2) e (3.3) (Proakis e Manolakis, 1996).

$$w(n) = \begin{cases} 0,54 - 0,46\cos\left(\dfrac{2\pi n}{M-1}\right), & \text{para } 0 \le n < M \\ 0, & \text{caso contrário} \end{cases} \tag{3.2}$$

$$w(n) = \begin{cases} 0,5\left[1 - \cos\left(\dfrac{2\pi n}{M-1}\right)\right], & \text{para } 0 \le n < M \\ 0, & \text{caso contrário} \end{cases} \tag{3.3}$$

A forma das janelas de (3.2) e de (3.3) podem ser vistas na Figura 3.1, para um *M* de 1024 amostras. A Figura 3.1a mostra a janela de Hamming e a 3.1b a janela de Hann.



(a) (b)

Figura 3.1 – Janelas de (a) Hamming e de (b) Hann.

Como é possível observar na Figura 3.1, ambas as janelas atenuam as extremidades de cada quadro do sinal. Isso, porém, também acarreta perda de informação, mais até do que o vazamento espectral. A fim de reduzir essa perda, deve-se utilizar a sobreposição de quadros, isto é, cada um dos quadros deve conter um certo percentual de amostras do quadro anterior. Fatores comuns de sobreposição são 2/3 (~ 67%) e 3/4 (75%). O cálculo do número de quadros N_F no qual um sinal será dividido, dada a superposição, pode ser feito usando a Equação (3.4), onde *L* é o tamanho total do sinal, *M* é o tamanho da janela e *O* é o fator de sobreposição, tudo dado em número de amostras.

$$N_F = \frac{L - O}{M - O} \tag{3.4}$$

3.1.2. Transformadas

Com o intuito de realizar a análise espectral de um sinal, é necessário passá-lo do domínio tempo para o domínio da freqüência. Isso é feito por meio de uma transformada, que mapeia as amostras do domínio tempo para componentes espectrais no domínio da freqüência. A transformada mais conhecida e mais utilizada em processamento digital de sinais (DSP) é a transformada de Fourier. Com ela é possível se obter uma série de informações a respeito do sinal que está sendo processado, como os espectros de magnitude, de fase e de potência. Ela é definida como sendo uma soma infinita de senóides e cossenóides, que são representadas no plano complexo ou na forma polar como magnitude e fase. A transformada discreta de Fourier (em inglês, *discrete Fourier transform* ou DFT) é mostrada na Equação (3.5) e sua inversa na Equação (3.6) (Frigo e Johnson, 2006), onde *M* é o número de amostras do quadro sendo analisado, $j = \sqrt{-1}$ e $e^{j\theta}$ é a identidade de Euler, isto é, $e^{j\theta} = \cos\theta + j\,\mathrm{sen}\,\theta$.

$$X(k) = \sum_{n=0}^{M-1} x(n) e^{-\frac{2\pi j}{M} kn} \tag{3.5}$$

$$x(n) = \frac{1}{M} \sum_{n=0}^{M-1} X(k) e^{\frac{2\pi j}{M} kn} \tag{3.6}$$

Além disso, o espectro de potência de um sinal pode ser calculado, dada sua transformada de Fourier *X*(*k*), com a Equação (3.7) (Garcia, 1999).

$$P(k) = \left|X(k)\right|^2 = \mathrm{Re}\left\{X(k)\right\}^2 + \mathrm{Im}\left\{X(k)\right\}^2 \tag{3.7}$$

Em 1942, Ralph Hartley criou uma transformada (Mintchev et al., 1995) com propriedades semelhantes à transformada de Fourier (Ullmann, 1984). A transformada de Hartley, como foi chamada, é na verdade uma simplificação da transformada de Fourier. Ao contrário dessa, que pode ser aplicada em sinais reais ou complexos, aquela aplica-se somente em sinais reais. Para se calcular a transformada discreta de Hartley (em inglês, *discrete Hartley transform* ou DHT), utiliza-se a Equação (3.8) (Ullmann, 1984; Frigo e Johnson, 2006).

$$X_H(k) = \sum_{n=0}^{M-1} x(n)\left[\cos\left(\frac{2\pi}{M}kn\right) + \text{sen}\left(\frac{2\pi}{M}kn\right)\right] \tag{3.8}$$

Como se pode observar em (3.8), a resposta da transformada de Hartley também é real. A inversa da DHT é ela própria multiplicada por $1/M$. Para se calcular o espectro de potência de um sinal, dada sua transformada de Hartley $X_H(k)$, utiliza-se a Equação (3.9) (Mintchev et al., 1995).

$$P_H(k) = \frac{X_H(k)^2 - X_H(-k)^2}{2} \tag{3.9}$$

Apesar de a resposta da DHT ser real, não há perda de informação em relação à DFT, visto que as duas transformadas se relacionam pelas Equações (3.10), (3.11) e (3.12) (Ullmann, 1984).

$$X_H(k) = \text{Re}\{X(k)\} - \text{Im}\{X(k)\} \tag{3.10}$$

$$\text{Re}\{X(k)\} = \frac{X_H(k) + X_H(-k)}{2} \tag{3.11}$$

$$\text{Im}\{X(k)\} = \frac{X_H(k) - X_H(-k)}{2} \tag{3.12}$$

Outra transformada comumente utilizada em DSP, principalmente para fins de compressão de dados, é a transformada do cosseno. Na verdade, a transformada discreta do cosseno (em inglês, *discrete cosine transform* ou DCT). A DCT compreende um grupo de oito transformadas, conhecidas como DCT-I a DCT-VIII (Frigo e Johnson, 2006). A DCT-II é a forma mais comum e pode ser definida conforme a Equação (3.13) (Khayam, 2003). A Equação (3.14) mostra a definição de $w(k)$ (Khayam, 2003).

$$X(k) = w(k)\sum_{n=0}^{M-1} x(n)\cos\left(\frac{\pi(2n+1)k}{2M}\right) \tag{3.13}$$

$$w(k) = \begin{cases} \dfrac{1}{\sqrt{M}}, & \text{para } k = 0 \\ \sqrt{\dfrac{2}{M}}, & \text{para } k > 0 \end{cases} \tag{3.14}$$

Existe, ainda, uma outra transformada derivada da DCT, conhecida como transformada discreta do cosseno modificada (em inglês, *modified discrete cosine transform* ou MDCT). Essa transformada foi proposta por Princen et al.[8] (1987 apud Brandenburg, 2002) e ela consiste basicamente de uma DCT-IV com sobreposição de amostras e uso de uma janela já embutida na própria transformada (Brandenburg, 2002). A MDCT não será discutida em maior profundidade, pois não é o intuito deste trabalho permanecer discutindo transformadas.

A DCT de modo geral, incluindo suas oito variantes e também a MDCT, foi largamente utilizada em compressão de áudio, imagens e vídeo ao longo das duas últimas décadas. Devido à sua característica de concentrar a maior parte da energia de um sinal nos coeficientes das baixas freqüências, ela acaba proporcionando taxas de compressão muito boas, e por isso está presente em diversos padrões como JPEG, MPEG-1, MPEG-2, MPEG-4, H.261 e H.263 (Khayam, 2003). Além disso, foi descoberto na prática que as transformadas do cosseno produzem melhores resultados do que a DFT para aplicações de codificação e restauração de áudio e de imagens (Yaroslavsky e Wang, 2000).

Porém para que a DCT e a MDCT sejam usadas de forma apropriada, às vezes é necessário que se estabeleçam relações entre o espectro de um sinal calculado por meio da DFT e o calculado por meio daquelas transformadas. Dois autores (Yaroslavsky e Wang, 2000) realizaram estudos a fim de fazer comparações no que diz respeito à resolução espectral e à capacidade de compactação da energia, isto é, a capacidade de redistribuir a energia de um sinal em um número pequeno de coeficientes espectrais. Eles concluíram que todas essas transformadas são adequadas para a realização de análise espectral. A DFT apresenta resolução espectral e capacidade de distribuição de energia praticamente uniformes ao longo de todo o espectro de um sinal. Com a DCT, a resolução espectral é maior nas altas freqüências e sua capacidade de compactação de energia é muito boa (cerca de 95% da energia do sinal concentra-se em 10% dos coeficientes espectrais). Em geral, os coeficientes com maior concentração de energia são os das baixas freqüências, como foi mencionado anteriormente. No caso da MDCT, a resolução espectral é um pouco mais regular do que a da DCT, mas sua capacidade de compactação de energia é semelhante. A Figura 3.2 ilustra de maneira mnemônica como fica a representação de um sinal com a DFT, a DCT e a MDCT.

[8] PRINCEN, J.; JOHNSON, A.; BRADLEY, A. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTIC, SPEECH AND SIGNAL PROCESSING, v. 12, 1987, Dallas, TX, USA. **Proceedings**… [S.l. : s.n], 1987. pp. 2161–2164.

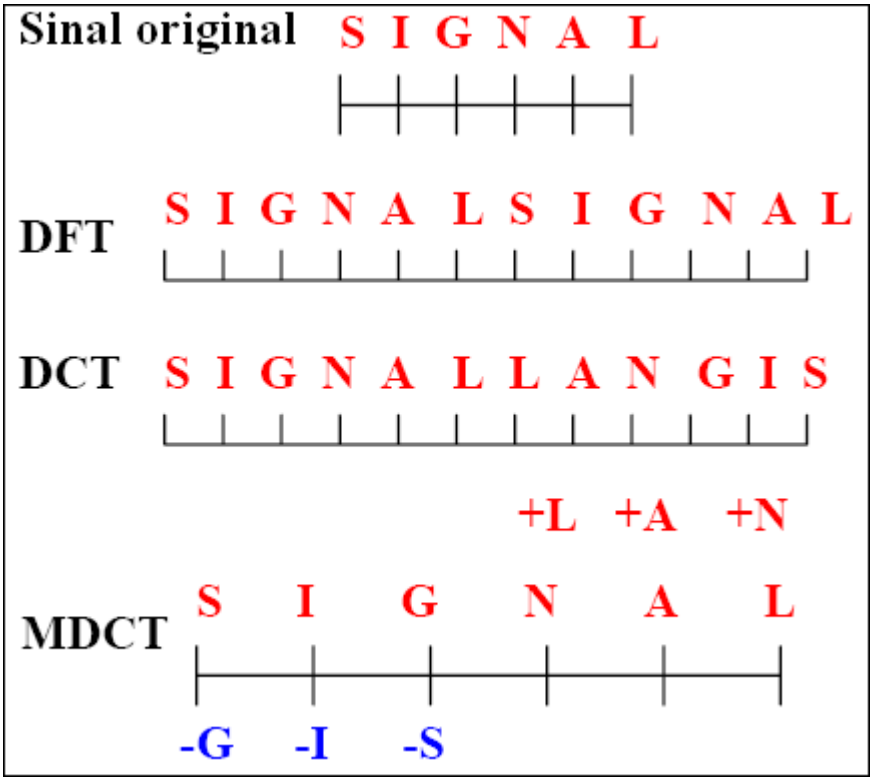


Figura 3.2 – Representação mnemônica de um sinal e de suas transformadas, adaptada de (Yaroslavsky e Wang, 2000).

3.1.3. Seqüências Pseudoaleatórias

Seqüências pseudoaleatórias são comumente utilizadas em aplicações *spread spectrum* (SS), pois elas não precisam ser transmitidas juntamente com os dados como aconteceria com seqüências verdadeiramente aleatórias. Essas seqüências, também conhecidas como seqüências de pseudo-ruído (em inglês, *pseudonoise* ou PN), receberam tal denominação por terem propriedades estatísticas semelhantes àquelas do ruído branco, isto é, aleatoriedade e espectro plano ao longo de toda a faixa de freqüências, e também por serem determinísticas e não verdadeiramente aleatórias (Sklar, 2001).

A geração de seqüências PN para aplicações SS foi consideravelmente pesquisada na literatura (Proakis, 1995). De longe, a seqüência PN binária mais conhecida é a de comprimento máximo (em inglês, *maximum length sequence* ou *m-sequence*), que é gerada por um registrador de deslocamento com realimentação linear (em inglês, *linear feedback shift register* ou LFSR). Uma *m-sequence* gerada por um LFSR de m estágios é periódica, com período $n = 2^m - 1$ (Proakis, 1995; Sklar, 2001). Além disso, o número de bits "1" é 2^{m-1}

por período e o número de bits “0” é $2^{m-1}-1$ por período (Proakis, 1995; Sklar, 2001). A Figura 3.3 mostra um LFSR de cinco estágios, isto é, $m=5$, feito com *flip-flops* tipo D.

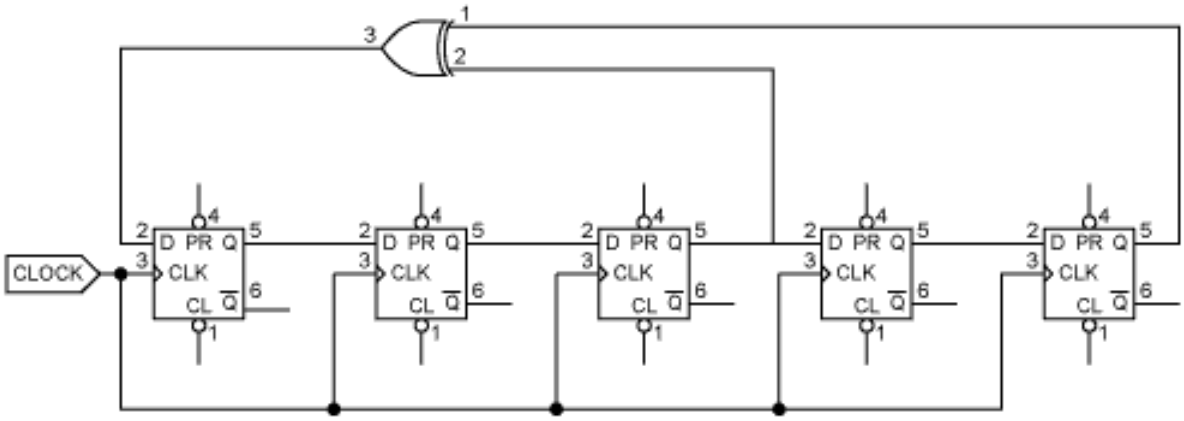


Figura 3.3 – Representação de um LFSR de cinco estágios ($m=5$) feito com *flip-flops* tipo D.

Uma importante propriedade das seqüências PN periódicas é ter uma função de autocorrelação também periódica (Proakis, 1995). Para *m-sequences*, a função de autocorrelação R_a, dada em função de um número inteiro *k*, é mostrada na Equação (3.15) (Proakis, 1995).

$$R_a(k)=\begin{cases} n, & \text{para } k=0 \\ -1, & \text{para } 0<k\le n-1 \end{cases} \tag{3.15}$$

Além das *m-sequences*, existem outras seqüências PN. Por exemplo, as seqüências geradas pelos códigos de Gold e de Kasami são baseadas em combinações de certos pares específicos de *m-sequences*. No código de Gold, os pares de seqüências devem ter o mesmo comprimento *n* e a correlação cruzada entre elas deve permitir apenas um dos três valores $\{-1,-t(m),t(m)-2\}$, onde *t*(*m*) é definido na Equação (3.16) (Proakis, 1995). Tais pares são chamados de pares preferenciais e eles geram *n* seqüências de Gold com o mesmo período das *m-sequences* originais. A geração é feita realizando-se a operação lógica “ou-exclusivo” dessas seqüências, sendo que uma delas deve ser rotacionada *n* vezes a fim de gerar *n* seqüências de Gold diferentes.

$$t(m) = \begin{cases} 2^{(m+1)/2} + 1, & \text{para } m \text{ ímpar} \\ 2^{(m+2)/2} + 1, & \text{para } m \text{ par} \end{cases} \tag{3.16}$$

No código de Kasami, o par de seqüências geratrizes é formado por uma *m-sequence* padrão e por uma seqüência que é uma versão decimada por um fator $q = 2^{m/2} + 1$ da primeira, onde *m* é par (Proakis, 1995), e replicada *q* vezes, para que as duas fiquem com o mesmo tamanho *n*. Da mesma forma que são geradas as seqüências de Gold, então, geram-se $2^{m/2}$ seqüências de Kasami (contando-se as duas originais), sendo que aquela gerada a partir da *m-sequence* padrão deve ser rotacionada $2^{m/2} - 2$ vezes. A correlação cruzada entre as seqüências geratrizes no código de Kasami assume apenas os valores contidos na trinca $\left\{-1, -\left(2^{m/2}+1\right), 2^{m/2}-1\right\}$ (Proakis, 1995).

3.2. *Direct Sequence Spread Spectrum*

"*Direct sequence* é o nome dado à técnica de espalhamento espectral na qual uma onda portadora é primeiramente modulada com um sinal de dados *d*(*t*), então esse sinal é novamente modulado com um sinal de espalhamento *c*(*t*) de alta velocidade (banda larga)" (Sklar, 2001, p. 732).

3.2.1. Visão Geral

A técnica *Direct Sequence Spread Spectrum* (DSSS), como foi esclarecido anteriormente, realiza duas modulações consecutivas: a primeira consiste em modular um sinal, normalmente binário e bipolar, isto é, que assume os valores ±1, com uma portadora senoidal (modulação *binary phase-shift keying* ou BPSK); a segunda consiste em realizar o espalhamento do sinal por meio de uma seqüência PN.

A recuperação do sinal de dados no receptor é feita calculando a correlação do sinal recebido com uma cópia sincronizada da seqüência PN utilizada na transmissão (Sklar, 2001) e, então, fazendo a demodulação. Isso parece simples, porém quando o sinal recebido é distorcido por ruído, a detecção correta dos dados enviados pode ser um problema. Esse ruído pode ser decorrente da transmissão do sinal através de um canal de comunicação, do processamento do sinal antes que ele chegue ao seu destino final ou da adição proposital de

ruído (*jamming*) a fim de impedir, ou pelo menos de dificultar, a transmissão desse sinal. As próximas seções discutirão com mais detalhes modulação BPSK, *jamming* e sincronização da seqüência PN.

3.2.2. Modulação BPSK

O método *binary phase-shift keying* (BPSK) realiza a modulação em fase de um sinal binário. Devido ao sinal modulado ser binário, sua fase pode assumir apenas os valores 0 ou π (180°). A modulação BPSK pode ser expressa de duas maneiras, conforme as Equações (3.17) e (3.18) (Garcia, 1999). No primeiro caso, que é a representação típica, é fácil visualizar que a modulação é feita em fase; no segundo, observa-se que a modulação é em amplitude. Ambas equações descrevem exatamente a mesma modulação, uma vez que somar 180° à fase de um sinal senoidal é o mesmo que multiplicá-lo por –1. Tem-se que E_b é a energia de um bit, T_b é o tempo que um bit leva para ser transmitido, ω_0 é a freqüência da portadora e d é o bit de dados sendo transmitido, sendo que $d \in \{-1, +1\}$.

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left(\omega_0 t + \frac{\pi(d+1)}{2}\right), \quad \text{para } 0 \le t \le T_b \tag{3.17}$$

$$s(t) = d\sqrt{\frac{2E_b}{T_b}} \cos(\omega_0 t), \quad \text{para } 0 \le t \le T_b \tag{3.18}$$

3.2.3. **Jamming**

O processo de adicionar, intencionalmente, ruído a um canal de comunicação ou a um sinal é chamado de *jamming* (Garcia, 1999). O objetivo disso é degradar a transmissão, utilizando um conhecimento prévio do sistema de comunicação, como as bandas utilizadas, o sistema de temporização etc., com um custo mínimo (Garcia, 1999; Sklar, 2001). A meta do projetista é desenvolver um sistema de comunicação resistente a *jamming* (resistente, mas não imune, visto que isso é impossível), sabendo que o único conhecimento a respeito desse sistema que seu adversário não possui é a seqüência PN utilizada no espalhamento do sinal (Sklar, 2001).

Para se entender melhor as diversas formas possíveis de *jamming*, deve-se introduzir alguns conceitos básicos, que são, na verdade, parâmetros do sistema de comunicação. O primeiro deles é a largura de banda W_{ss} disponibilizada pelo canal. Ela deve ser pelo menos o dobro da freqüência máxima existente no sinal, segundo o Teorema da Amostragem de Nyquist (Proakis e Manolakis, 1996; Oppenheim e Willsky, 1997). Outro parâmetro é a potência S do sinal na entrada do receptor, que é descrito na Equação (3.19). E_b e T_b são os mesmos das Equações (3.17) e (3.18), e R_b é a taxa de transmissão do sistema dada em bits por segundo.

$$S = \frac{E_b}{T_b} = E_b R_b \tag{3.19}$$

O ganho de processamento G do sistema é descrito pela Equação (3.20) (Garcia, 1999; Sklar, 2001). A relação entre as potências do ruído de *jamming* e do sinal é J/S, ambas medidas na entrada do receptor. Finalmente, a grandeza mais importante é a relação sinal-ruído (em inglês, *signal-to-noise ratio* ou SNR), que em comunicação digital é mais comumente denotada por E_b/N_0 e significa a relação entre a potência média do sinal e a potência média do ruído (Sklar, 2001). Aqui será utilizada a notação E_b/J_0 para ficar claro de que o ruído em questão é proveniente de *jamming*. Essa relação pode ser desdobrada em função dos parâmetros já referidos, conforme mostra a Equação (3.21) (Sklar, 2001).

$$G = \frac{W_{ss}}{R_b} \tag{3.20}$$

$$\frac{E_b}{J_0} = \frac{S/R_b}{J/W_{ss}} = \frac{W_{ss}/R_b}{J/S} = \frac{G}{J/S} \tag{3.21}$$

A relação E_b/J_0 é uma figura de mérito em sistemas de comunicação *spread spectrum* (SS) e ela representa a mínima energia por densidade espectral de potência do ruído de *jamming* que um bit deve ter para que o sistema de comunicação continue operando com a probabilidade de erros exigida pela aplicação (Sklar, 2001).

Como já foi mencionado, existem diversas formas de *jamming*. Este trabalho abordará as três principais: *jamming* de banda larga, no qual o ruído é igualmente distribuído ao longo de toda a banda de comunicação W_{ss}; *jamming* de banda parcial, no qual o ruído é igualmente

distribuído ao longo de uma fração ρ da banda de comunicação; *jamming* por pulsos, que é semelhante ao *jamming* de banda parcial, porém o ruído ocorre apenas em frações τ de tempo.

Em *jamming* de banda larga, o ruído é modelado como um processo Gaussiano estacionário com média zero e com densidade espectral de potência distribuída igualmente em toda a faixa de freqüências (Sklar, 2001), nesse caso W_{ss}, como pode ser observado na Figura 3.4, onde J é a potência total do ruído.

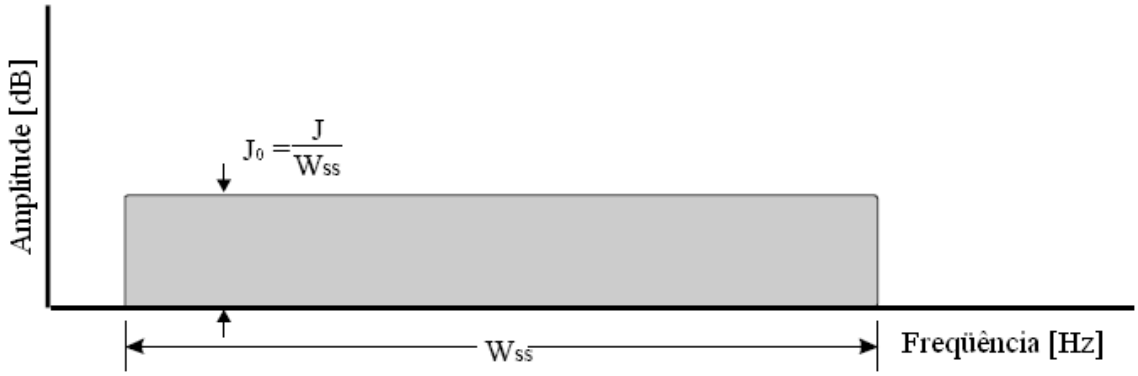


Figura 3.4 – Ruído de *jamming* de banda larga (Garcia, 1999).

A probabilidade média de erros de bits para um sistema SS utilizando modulação BPSK, quando submetido a *jamming* de banda larga pode ser calculada com a Equação (3.22) (Sklar, 2001). Todos os valores envolvidos nessa equação já foram discutido anteriormente. A função $Q(\cdot)$, conhecida como função erro complementar, está definida na Equação (3.23) (Garcia, 1999; Sklar, 2001).

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0 + J_0}}\right) = Q\left(\sqrt{\frac{2E_b/N_0}{1+(E_b/N_0)(J/S)/G}}\right) \tag{3.22}$$

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{u^2}{2}\right)du \tag{3.23}$$

No caso de *jamming* de banda parcial, a largura de banda W_J do ruído é menor do que a largura de banda total W_{ss} do sistema de comunicação. Isso faz com que a densidade espectral de potência do ruído seja mais concentrada em uma banda menor, havendo maior sucesso na interferência caso aquela banda esteja sendo utilizada na transmissão da

informação. Isso pode ser observado na Figura 3.5. O fator ρ deve ser menor que um para que seja *jamming* de banda parcial.

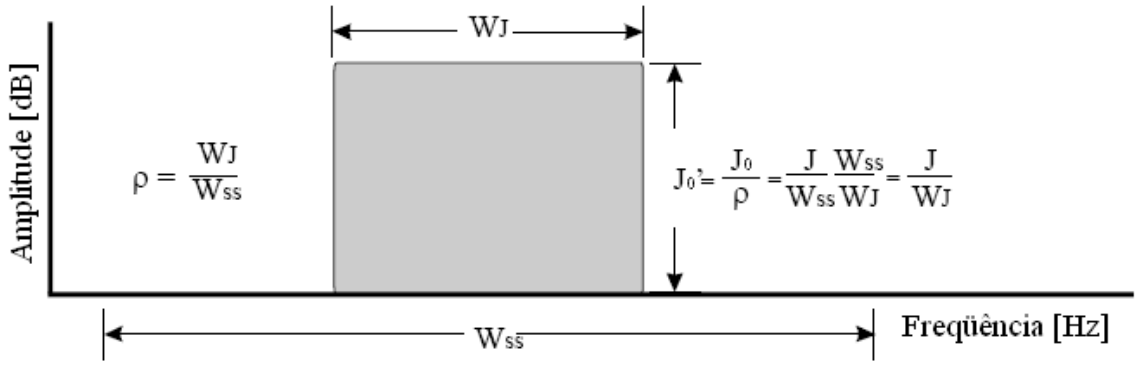


Figura 3.5 – Ruído de *jamming* de banda parcial (Garcia, 1999).

O terceiro caso é o de *jamming* por pulsos. Nessa situação, o ruído não é constante, ou seja, ocorre durante um intervalo τ finito. Os pulsos são de ruído Gaussiano limitado em banda cuja potência média é *J* ao longo de todo o período, embora ela seja maior durante o intervalo τ (Sklar, 2001). A probabilidade média de erros de bits para um sistema SS utilizando modulação BPSK, quando submetido a *jamming* por pulsos pode ser calculada com a Equação (3.24) (Sklar, 2001).

$$P_B = (1-\tau)Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \tau Q\left(\sqrt{\frac{2E_b}{N_0 + J_0/\tau}}\right) \tag{3.24}$$

Como $N_0 \ll J_0$, o fator N_0 pode ser ignorado, resultando na Equação (3.25) (Sklar, 2001). O período τ_0 que maximiza P_B é dado na Equação (3.26) (Garcia, 1999; Sklar, 2001). Conseqüentemente, a probabilidade de erros máxima pode ser calculada com a Equação (3.27) (Garcia, 1999; Sklar, 2001).

$$P_B \approx \tau Q\left(\sqrt{\frac{2E_b\tau}{J_0}}\right) \tag{3.25}$$

$$\tau_0 = \begin{cases} \dfrac{0,709}{E_b/J_0}, & \text{para } \dfrac{E_b}{J_0} > 0,709 \\ 1, & \text{para } \dfrac{E_b}{J_0} \le 0,709 \end{cases} \tag{3.26}$$

$$P_{B_{\max}} = \begin{cases} \dfrac{0,083}{E_b/J_0}, & \text{para } \dfrac{E_b}{J_0} > 0,709 \\ Q\left(\sqrt{\dfrac{2E_b}{J_0}}\right), & \text{para } \dfrac{E_b}{J_0} \le 0,709 \end{cases} \tag{3.27}$$

A diferença na probabilidade de erros de bits para diversos valores de τ pode ser vista na Figura 3.6a e uma comparação entre um ruído com duração constante ($\tau = 1$) e um ruído por pulsos considerando o pior caso ($\tau = \tau_0$) pode ser observada na Figura 3.6b. Nessa situação, é possível verificar que para uma probabilidade de erros igual a 10^{-5} a diferença em E_b/J_0 entre o ruído constante e o ruído com $\tau = \tau_0$ é de quase 30 dB (Garcia, 1999).

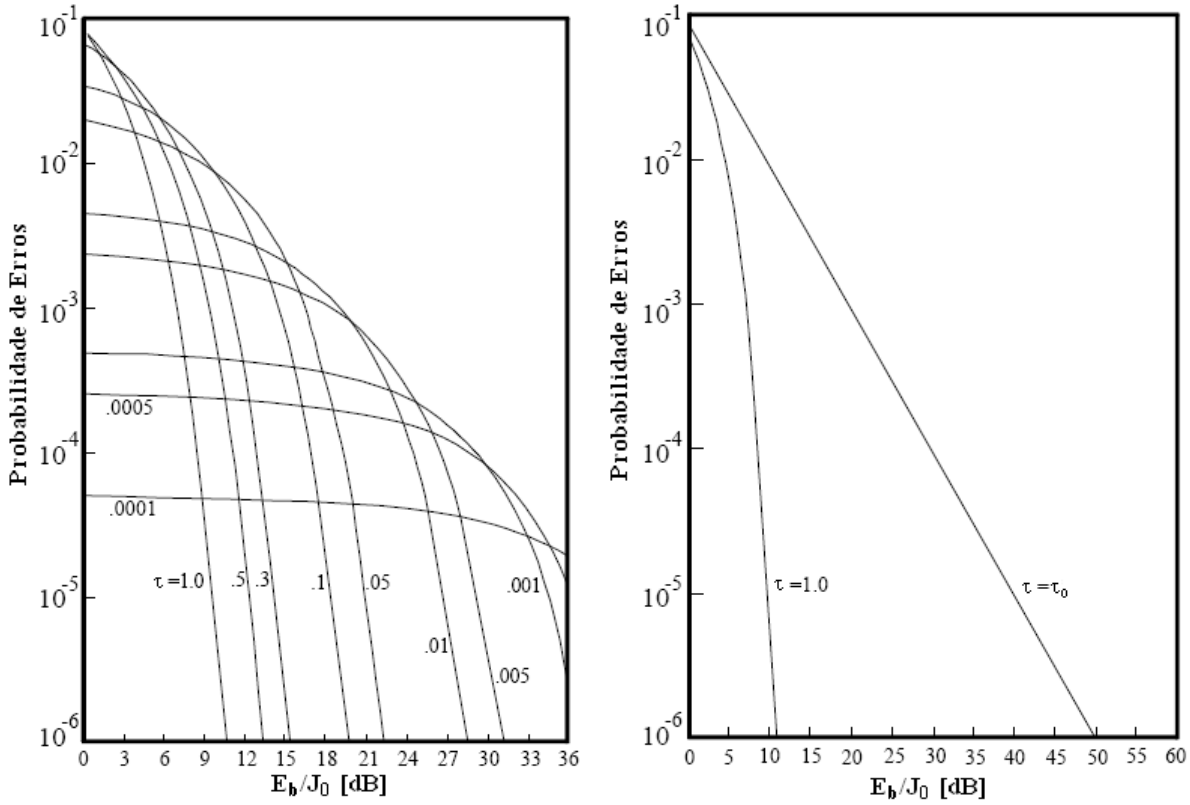


Figura 3.6 – Probabilidades de erros de bits para (a) valores variados de τ e (b) em comparação entre ruído constante e o pior caso de ruído por pulsos (Simon et al.[9], 1994 apud Garcia, 1999).

[9] SIMON, Marvin K.; OMURA, Jim K.; SCHOLTZ, Robert A.; LEVITT, Barry K. **Spread Spectrum Communications Handbook.** New York: McGraw-Hill, 1994. 1228 pp.

3.2.4. Sincronização e Detecção

Como diz a citação no início da seção 3.2, na técnica de DSSS o sinal de dados *d*(*t*) é modulado primeiramente com uma onda portadora e depois com o sinal de espalhamento *c*(*t*), que é uma seqüência PN. Nas Equações (3.17) e (3.18) é mostrado como é a modulação do sinal de dados com a onda portadora. As Equações (3.28) e (3.29) demonstram, também, a modulação com o sinal de espalhamento. As variáveis e as constante envolvidas são as mesmas das equações referidas anteriormente.

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left[\omega_0 t + \frac{\pi\left[d(t)c(t)+1\right]}{2}\right], \quad \text{para } 0 \le t \le T_b \tag{3.28}$$

$$s(t) = d(t)c(t)\sqrt{\frac{2E_b}{T_b}} \cos\left[\omega_0 t\right], \quad \text{para } 0 \le t \le T_b \tag{3.29}$$

É importante ressaltar que a seqüência PN *c*(*t*) deve ser *N* vezes mais rápida do que a seqüência de dados *d*(*t*), para que ocorra o espalhamento espectral. Logo, tem-se que o intervalo de duração de um bit T_c da seqüência PN é igual a T_b / N. Esse intervalo é chamado período de *chip* ou, simplesmente, *chip*.

Para que um sinal transmitido via DSSS seja corretamente detectado no receptor, esse deve conter um réplica sincronizada da seqüência PN utilizada no transmissor (Proakis, 1995; Garcia, 1999; Sklar, 2001). Essa sincronização é feita em duas fases, chamadas aquisição e rastreamento (Proakis, 1995; Garcia, 1999; Sklar, 2001). Na fase da aquisição, deve-se realizar uma busca através de uma região de incertezas em tempo e em freqüência a fim de sincronizar o sinal recebido com a cópia local do sinal de espalhamento (Sklar, 2001). O alinhamento dos sinais dá-se no domínio tempo, e a diferença aceitável na aquisição deve ser menor do que um *chip* (Proakis, 1995; Garcia, 1999).

Na fase de rastreamento, um alinhamento mais fino é feito. Isso acontece devido a um sistema realimentado que fica constantemente regulando a fase do sinal de espalhamento no receptor para manter a sincronia perfeita, possibilitando, assim, a recuperação do sinal original sem espalhamento (*de-spreading*). No entanto, aquisição e rastreamento podem ser realizados juntos, utilizando uma estrutura de filtros casados (em inglês, *matched filters*) ou de correlatores para pesquisar, com alta resolução, o sinal recebido e compará-lo com a

réplica local da seqüência PN (Garcia, 1999). Neste trabalho, será utilizada filtragem adaptativa para executar tal tarefa.

Os filtros adaptativos têm a característica de "aprender" à medida que um sinal passa por eles, isto é, seus coeficientes são periodicamente atualizados com valores obtidos a partir desse sinal (Garcia, 1999). Eles podem ser implementados tanto no domínio do tempo como no domínio da freqüência e suas funções de transferência são funções do sinal e/ou do ruído (Rivera-Colon et al., 1992). Esses filtros classificam-se, em ordem crescente de dificuldade baseada no conhecimento que se tem a priori a respeito do sinal e do ruído, em três classes distintas (Rivera-Colon et al., 1992):

- Classe 1: os modelos dos espectros do sinal e do ruído são conhecidos;
- Classe 2: o modelo do espectro do sinal ou do espectro do ruído são conhecidos, mas não ambos simultaneamente;
- Classe 3: os modelos dos espectros do sinal e do ruído são desconhecidos.

Um exemplo de filtro adaptativo de detecção de alta resolução classe 1/3 é dado na Equação (3.30) (Lindquist[10], 1989 apud Garcia, 1999), onde $S^*(m)$ é o complexo conjugado do espectro do sinal que se deseja detectar e $\left\langle \left| R(m) \right|^2 \right\rangle$ é uma versão suavizada do espectro de potência do sinal de entrada. Essa suavização é feita no domínio da freqüência por uma janela, que pode ser a de Hamming ou a de Hann, citadas anteriormente, ou qualquer outra. Sua finalidade é atenuar as componentes espectrais de alta freqüência, preservando o espectro a ser analisado (Rivera-Colon et al., 1993a). A aplicação da janela $B(m)$ no domínio da freqüência é mostrada na Equação (3.31).

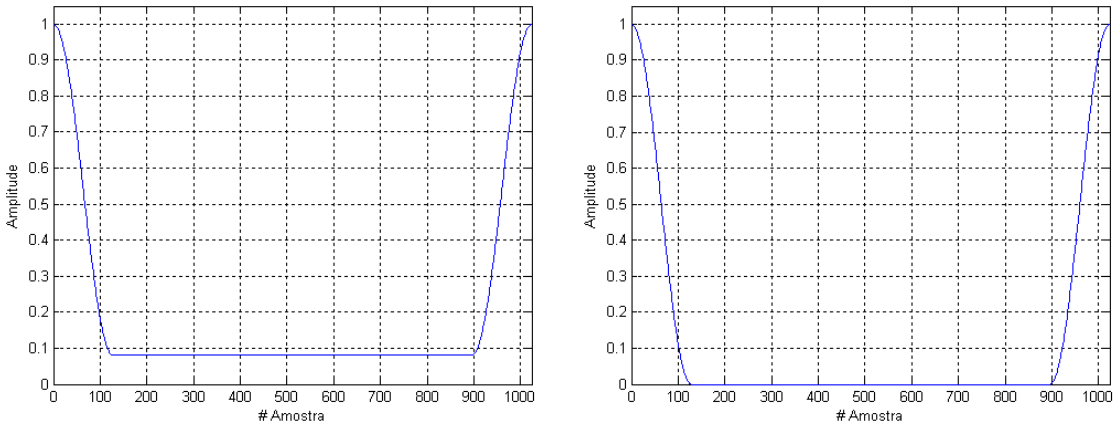
$$H(m) = \frac{S^*(m)}{\left\langle \left| R(m) \right|^2 \right\rangle} \tag{3.30}$$

$$\left\langle R(m) \right\rangle = R(m) * B(m) \tag{3.31}$$

Outra maneira de explicar a finalidade da suavização é que ela serve para estimar a média do espectro do sinal mais ruído a partir do sinal de entrada do sistema (Garcia, 1999). O nível de suavização é um percentual do número de *bins* do espectro do sinal a ser

[10] LINDQUIST, Claude S. **Adaptive & Digital Signal Processing with Digital Filtering Applications.** Miami: Steward & Sons, 1989. 847 pp.

suavizado (Rivera-Colon et al., 1993b). As mesmas janelas de 1024 amostras da Figura 3.1 podem ser vistas na Figura 3.7 com uma suavização de 25%.



(a) (b)

Figura 3.7 – Janelas de (a) Hamming e de (b) Hann com suavização de 25%.

Um artigo de Lindquist e Rivera-Colon[11] (1995) traz maiores detalhes a respeito das classes dos filtros adaptativos, bem como diversos exemplos de filtros das três classes. O artigo ainda mostra uma série de gráficos que demonstram como funcionam alguns dos filtros dados como exemplos.

A detecção é feita seguindo a ordem inversa das modulações realizadas no processo de transmissão. Em primeiro lugar, desfaz-se o espalhamento (*de-spreading*) multiplicando o sinal recebido pela réplica sincronizada da seqüência PN utilizada; após, demodula-se o sinal a fim de obter apenas a seqüência de bits transmitida. O *de-spreading* só é possível devido a uma importante propriedade das seqüências PN, que pode ser vista na Equação (3.32) (Garcia, 1999).

$$c^2(t) = 1, \quad \forall t \tag{3.32}$$

Observando a Equação (3.32) fica claro porque a seqüência PN no receptor deve estar perfeitamente sincronizada com o sinal recebido. Caso contrário, não haveria como desfazer o espalhamento espectral. A demodulação é feita multiplicando o sinal resultante pela mesma onda utilizada na modulação. A Equação (3.33) (Garcia, 1999; Sklar, 2001) mostra como

[11] LINDQUIST, Claude S.; RIVERA-COLON, Ramfis. Selection of Transforms and the Use of Smoothing in Class 2 and 3 Adaptive Filters. In: 29th ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, v. 1, 1995, Pacific Grove, CA, USA. **Conference**… [S.l. : s.n.], 1995. pp. 755-759.

ocorrem o *de-spreading* e a demodulação, onde $n(t)$ é um ruído Gaussiano com média zero, proveniente de *jamming*.

$$r(t) = \left[d(t) + n(t) \right] c(t) \sqrt{\frac{2}{T_b}} \cos\left[\omega_0 t\right], \quad \text{para } 0 \le t \le T_b \tag{3.33}$$

Por fim, a decisão final é tomada calculando-se a integral da Equação (3.34) e seguindo o critério definido na Equação (3.35). É importante observar que o resultado da integral é simplesmente $d\sqrt{E_b}$, uma vez que, como foi dito anteriormente, $n(t)$ possui média zero.

$$z(t) = \int_0^{T_b} r(t)\, dt \tag{3.34}$$

$$\hat{d} = \begin{cases} 1, & \text{se } z > 0 \\ -1, & \text{se } z \le 0 \end{cases} \tag{3.35}$$

4. Técnicas de Compressão de Dados

As técnicas de compressão de dados classificam-se basicamente em dois tipos: com perdas (em inglês, *lossy*) e sem perdas (em inglês, *lossless*). A compressão com perdas caracteriza-se, evidentemente, por haver perdas de informação nos dados comprimidos em relação aos dados originais. Ela aplica-se a tipos específicos de dados que permitem que a redução na quantidade de informação não seja percebida, como é o caso de imagens, de áudio e de vídeo. A compressão sem perdas, por outro lado, realiza uma representação mais eficiente dos dados (Sklar, 2001), reduzindo seu volume sem alterar a quantidade de informação.

O termo informação, nesse contexto, traz consigo o sentido de imprevisibilidade contida em determinado conjunto de dados (Wells, 1999). Por exemplo, um texto contendo apenas a letra ‘A’ repetida centenas de vezes não contém praticamente informação nenhuma, apesar de compor um grande conjunto de dados. Da mesma forma, uma imagem totalmente em branco ou um arquivo de som contendo apenas silêncio não trazem novidades a quem está olhando ou escutando.

Em 1948, Claude Shannon escreveu um artigo que foi publicado em duas partes[12,13] no Bell System Technical Journal tratando da modelagem matemática de sistemas de comunicação. Nesse artigo, Shannon deu uma definição matemática precisa da quantidade média de informação contida em cada elemento de uma fonte de dados (Wells, 1999). Essa medida é chamada de entropia e será definida mais adiante.

[12] SHANNON, Claude E. A Mathematical Theory of Communication. **Bell System Technical Journal**, Hoboken, v. 27, pp. 379-423, Jul. 1948.
[13] SHANNON, Claude E. A Mathematical Theory of Communication. **Bell System Technical Journal**, Hoboken, v. 27, pp. 623-656, Oct. 1948.

4.1. Fontes de Dados e Entropia

As fontes de dados podem ser contínuas ou discretas. Como os dados tratados neste trabalho são digitais, serão abordadas apenas as fontes discretas, que são formadas por alfabetos. Um alfabeto pode ser representado na forma de um conjunto finito de elementos, conforme a Equação (4.1), onde a_i, para $0 \leq i < M$, são os elementos de A, também chamados de símbolos.

$$A = \{a_0, a_1, \ldots, a_{M-1}\} \tag{4.1}$$

O número de símbolos M do alfabeto A é chamado de cardinalidade (Wells, 1999) e sua representação matemática é $M = |A|$. A probabilidade de que cada símbolo a_i seja emitido pela fonte de dados cujo alfabeto é A é denotada por $p_i = \Pr(a_i)$, formando o conjunto de probabilidades de A mostrado na Equação (4.2).

$$P_A = \{p_0, p_1, \ldots, p_{M-1}\} \tag{4.2}$$

Visto que uma fonte de dados emite apenas símbolos pertencentes ao seu alfabeto, tem-se que o somatório dos elementos de P_A deve ser sempre 1, como é mostrado na Equação (4.3).

$$\sum_{i=0}^{M-1} p_i = 1 \tag{4.3}$$

Uma vez que alguns conceitos básicos foram descritos, pode-se agora definir o que é entropia. Conforme Wells (1999, p. 4), entropia é “a quantidade média de informação transmitida por símbolo da fonte”. Matematicamente, a entropia é descrita segundo a Equação (4.4).

$$H(A) = \sum_{i=0}^{M-1} p_i \log_2(1/p_i) \tag{4.4}$$

O logaritmo utilizado na Equação (4.4) é normalmente o de base 2, resultando numa entropia dada em bits, que é a unidade básica utilizada em comunicação digital. Dado o

conjunto de probabilidades $P_A = \{0{,}5;\ 0{,}3;\ 0{,}15;\ 0{,}05\}$ de uma fonte de dados cujo alfabeto possui quatro símbolos (cada um representado por dois bits), pode-se calcular a entropia dessa fonte, que é de 1,6477 bit por símbolo (Wells, 1999). Nota-se, com isso, que a quantidade de informação da fonte é menor do que os dois bits utilizados para representar cada símbolo, o que acarreta um uso desnecessário de aproximadamente 17,6% dos bits emitidos (Wells, 1999).

4.2. Codificação de Fonte

Codificação de fonte nada mais é do que mapear um alfabeto *A* em um outro alfabeto *B* utilizando uma função de mapeamento com o intuito de aproximar o número de bits por símbolo emitidos pela fonte de sua entropia. Evidentemente, essa função de mapeamento deve ser invertível para que *B* possa ser mapeado de volta para *A*. A notação matemática do primeiro mapeamento é mostrada na Equação (4.5) (Wells, 1999).

$$C : A \rightarrow B \tag{4.5}$$

Usando o conjunto de probabilidades do exemplo anterior, pode-se construir o código apresentado na Tabela 4.1. Com esse código, o número médio de bits por símbolo emitido pela fonte é 1,7 (Wells, 1999), acarretando um uso desnecessário de aproximadamente 3,1% dos bits emitidos (Wells, 1999), o que é muito menor do que os 17,6% sem a codificação.

Tabela 4.1 – Distribuição de probabilidades de um alfabeto de quatro símbolos *A* e sua respectiva versão codificada *B*.

i	p_i	a_i	b_i
0	0,50	00	0
1	0,30	01	10
2	0,15	10	110
3	0,05	11	111

Os números do parágrafo acima revelam que a codificação de fonte é, na realidade, uma forma de compressão de dados. No entanto, o que pode parecer simples, como é o caso do exemplo da Tabela 4.1 no qual todos os símbolos do alfabeto são independentes, em uma fonte real isso quase nunca ocorre. Em um texto escrito em inglês, por exemplo, as letras que

o compõem, formando palavras e frases, são estatisticamente dependentes de suas predecessoras (Proakis, 1995). Já em um programa escrito em Fortran, ou em qualquer outra linguagem de programação, espera-se que a dependência entre as letras que o compõem seja significativamente menor (Proakis, 1995).

Dependendo da natureza da fonte, diferentes métodos de compressão podem apresentar resultados mais ou menos eficazes. A eficácia da compressão utilizada na codificação da fonte influencia diretamente a capacidade de transmissão do canal de comunicação. O conceito de capacidade de canal também foi definido por Shannon em seu trabalho de 1948. Esse assunto será tratado a seguir.

4.3. Capacidade de Canal

Para a Teoria da Informação, capacidade de canal é a máxima quantidade de informação que pode ser transmitida através de um canal de comunicação dado um certo grau de confiabilidade, isto é, uma probabilidade de erros aceitável. Shannon, em 1948, mostrou que a capacidade C_c de um canal perturbado por ruído branco gaussiano aditivo (em inglês, *additive white Gaussian noise* ou AWGN) está associada à relação sinal-ruído *SNR* do sinal transmitido e com a largura de banda *W* do canal de comunicação (Sklar, 2001). Tal relação, mostrada na Equação (4.6), é conhecida como Teorema de Shannon-Hartley (Sklar, 2001). A unidade de medida da capacidade de canal é normalmente dada em bits por segundo (observar uso do logaritmo de base 2).

$$C_c = W \log_2 \left(1 + SNR\right) \tag{4.6}$$

Visto que em um sistema de ocultação de informações que utiliza espalhamento espectral o sinal hospedeiro é considerado ruído (que evidentemente não possui espectro plano como o AWGN por se tratar de ruído colorido) e que sua energia é muito maior do que a energia do sinal que carrega a informação (ao contrário do que ocorre na maioria dos sistemas de comunicação), a Equação (4.6) não se aplica no cálculo da capacidade de canal nesse tipo de sistema. Além disso, é muito difícil determinar a largura de banda desse tipo de canal. Alguns textos (Shimizu, 2002; Cvejic e Seppänen, 2005; Sedghi et al., 2006) apresentaram soluções para o cálculo da capacidade de canal em sistemas com tais

características, mas a solução apresentada neste trabalho foi a avaliação empírica, que será discutida no próximo capítulo.

4.4. Código de Huffman

Em 1952, Huffman[14] desenvolveu um algoritmo de codificação ótimo e de comprimento variável baseado na probabilidade de ocorrência das letras (ou símbolos) provenientes da fonte (Proakis, 1995). Isso significa que o número médio de bits por símbolo emitido pela fonte aproxima-se da entropia da mesma quando esta for codificada. No entanto, tal característica depende do alfabeto gerado por essa fonte (Sklar, 2001). Certos alfabetos podem apresentar taxas de compressão menores do que outros. Para se conseguir taxas de compressão mais altas, é possível transformar o alfabeto antes de se aplicar a codificação (Sklar, 2001), o que será mostrado mais adiante neste capítulo.

Códigos de Huffman são formados usando um algoritmo de construção de árvores (Wells, 1999). O primeiro passo do algoritmo é listar os símbolos do alfabeto em ordem decrescente de probabilidades (Wells, 1999; Sklar, 2001). A fim de tornar mais clara a explanação, será utilizado o alfabeto $X = \{\text{A, B, C, D, E, F}\}$ com o respectivo conjunto de probabilidades (não normalizado) $P_X = \{6;5;4;3;2;1\}$. O passo seguinte é agrupar os dois símbolos com menores probabilidades em um único símbolo composto cuja probabilidade (ou freqüência relativa) é a soma daquelas dos símbolos formadores. Isso é ilustrado na Figura 4.1.

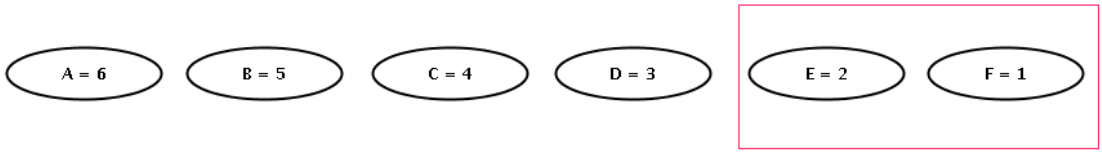


Figura 4.1 – Agrupamento dos símbolos que possuem as freqüências relativas mais baixas, neste caso ‘E’ e ‘F’ (Wikipédia, 2009)[15].

[14] HUFFMAN, David A. A Method for the Construction of Minimum Redundancy Codes. In: INSTITUTE OF RADIO ENGINEERS (IRE), v. 40, n. 9, 1952, New York, NY, USA. **Proceedings**… [S.l. : s.n.], 1952. pp. 1098-1101.

[15] WIKIPÉDIA. Codificação de Huffman. Disponível em: http://pt.wikipedia.org/wiki/Codificação_de_Huffman. Acesso em: 13 fev. 2009.

Conforme se observa na Figura 4.2, os símbolos ‘E’ e ‘F’ recebem um bit “0” ou um bit “1”, de acordo com suas probabilidades iniciais (o de maior probabilidade recebe o bit “0” e o outro o bit “1”). Então o símbolo composto ‘E+F’ é agrupado com o próximo símbolo cuja probabilidade é a mais baixa, formando um novo símbolo composto. Tal processo se repete até que a árvore binária esteja completa (vide Figuras 4.3 até 4.6).

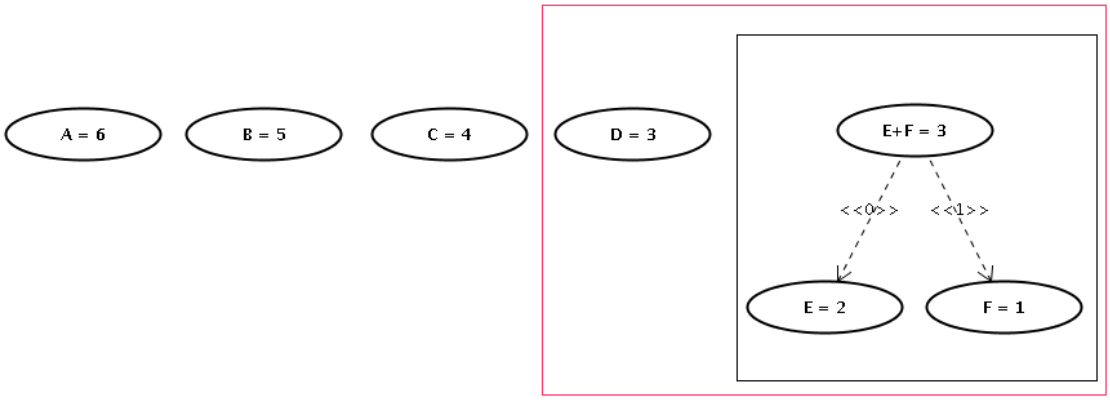


Figura 4.2 – Agrupamento dos símbolos que possuem as freqüências relativas mais baixas, neste caso ‘E+F’ e ‘D’ (Wikipédia, 2009).

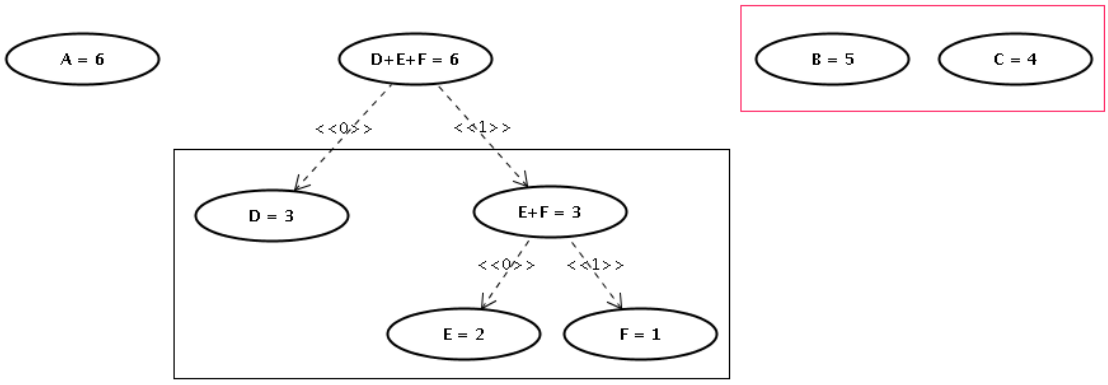


Figura 4.3 – Agrupamento dos símbolos que possuem as freqüências relativas mais baixas, neste caso ‘B’ e ‘C’ (Wikipédia, 2009).

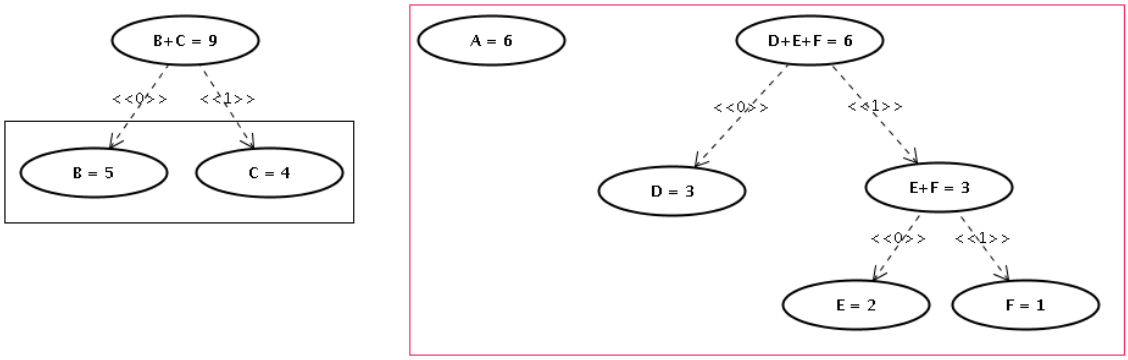


Figura 4.4 – Agrupamento dos símbolos que possuem as freqüências relativas mais baixas, neste caso ‘D+E+F’ e ‘A’ (Wikipédia, 2009).

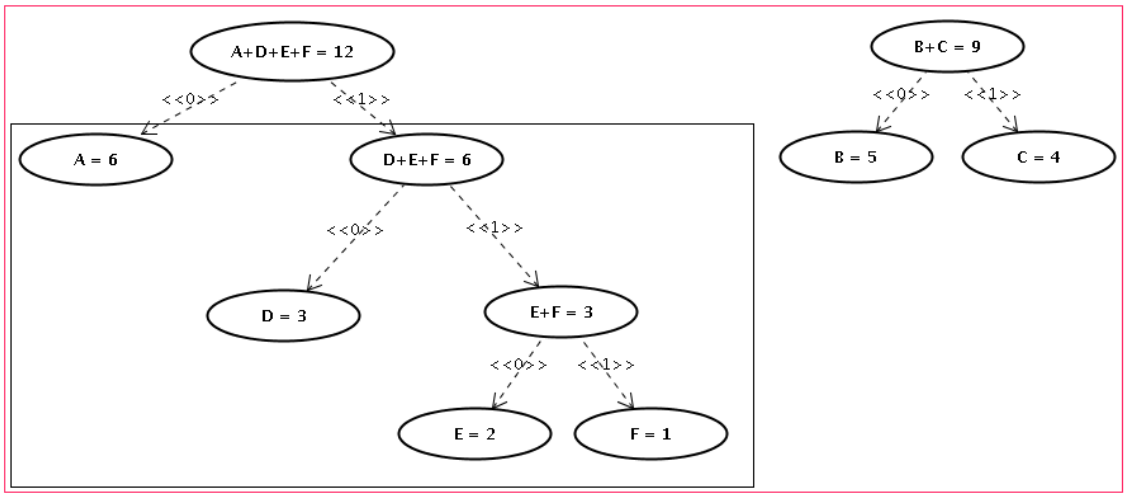


Figura 4.5 – Agrupamento de todos os símbolos (Wikipédia, 2009).

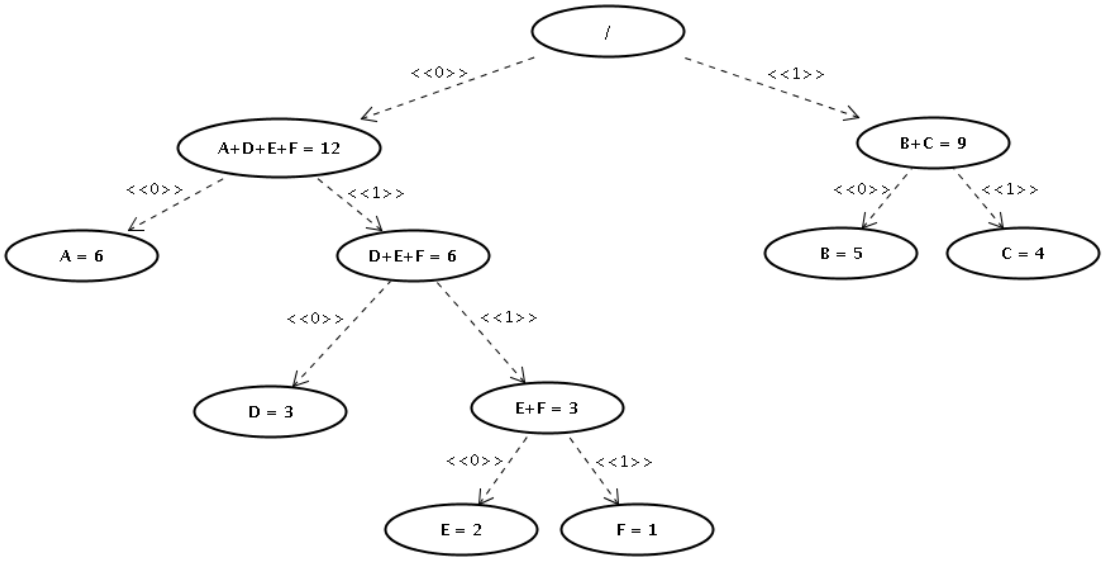


Figura 4.6 – Árvore binária de Huffman completa (Wikipédia, 2009).

A seguir, basta percorrer a árvore da Figura 4.6 a partir da raiz em direção aos nós extremos e ir concatenando os bits “0” ou “1” para formar as palavras-código relativas a cada símbolo do alfabeto. O resultado pode ser visto na Tabela 4.2. O mapeamento realizado foi $Z : X \rightarrow Y$.

Tabela 4.2 – Possível representação binária do alfabeto *X*, sua respectiva distribuição de probabilidades (não normalizada e normalizada) e sua versão codificada *Y*.

i	p_i	$p_{inorm.}$	x_i	y_i
0	6	0,286	000	00
1	5	0,238	001	10
2	4	0,190	010	11
3	3	0,143	011	010
4	2	0,095	100	0110
5	1	0,048	101	0111

Utilizando a Equação (4.4) podemos calcular $H(X) \approx 2{,}3983$, e utilizando a Equação (4.7) podemos calcular o número médio de bits $\bar{L} \approx 2{,}4286$ por símbolo emitido pela fonte, dado que L_i é o número de bits da palavra-código relativa ao símbolo de índice *i* do alfabeto original (Wells, 1999).

$$\bar{L} = \sum_{i=0}^{M-1} p_i L_i \tag{4.7}$$

No exemplo anterior, a codificação de Huffman permitiu que apenas 1,25% dos bits emitidos pela fonte fossem redundantes. Frente aos 20% que seriam desperdiçados sem codificação, esse resultado mostra como o código de Huffman pode ser eficiente em determinados casos.

4.5. Algoritmo de Lempel-Ziv-Welch

Uma grande dificuldade em utilizar o código de Huffman é que as probabilidades dos símbolos deve ser conhecida ou estimada (Sklar, 2001). Existe uma variedade de aplicações em que isso é possível, mas também há aquelas em que isso se torna impraticável (Wells, 1999). Além do mais, com o código de Huffman a árvore de codificação deve ser conhecida pelo codificador e pelo decodificador. Se as probabilidades são obtidas durante a codificação, elas devem ser transmitidas juntamente com os dados, acarretando um *overhead* que reduz os ganhos obtidos com a compressão (Sklar, 2001).

O algoritmo de Lempel-Ziv-Welch (LZW) é um algoritmo de compressão de dados baseado em dicionário, isto é, as tabelas de codificação e de decodificação são construídas durante sua execução (Wells, 1999). Isso é feito analisando-se os próprios dados, e não é

necessário se ter conhecimento das probabilidades dos símbolos de antemão (Wells, 1999). Esse algoritmo foi desenvolvido por Welch[16] em 1984, que baseou-se nos trabalhos de Jacob Ziv e Abraham Lempel (1977[17] e 1978), implementando algumas modificações.

O LZW é relativamente simples, embora sua implementação prática possa ser um pouco trabalhosa devido à manipulação de *strings* e das tabelas de codificação e de decodificação. O processo de codificação pode ser resumido em adicionar novos *strings* na tabela e devolver índices quando um *string* já existir na tabela. Um pseudocódigo de compressão utilizando o LZW é mostrado na Figura 4.7.

```
S = ""
inicializa a tabela com os caracteres da entrada
ENQUANTO existirem caracteres para ser lidos FAÇA
     C = próximo caractere da entrada
     SE S+C já estiver na tabela ENTÃO
          S = S+C
     SENÃO
          transmite o índice do string S
          adiciona S+C à tabela
          S = C
     FIM
FIM
```

Figura 4.7 – Pseudocódigo do algoritmo de codificação de Lempel-Ziv-Welch.

Para que o algoritmo de codificação fique mais claro, a Tabela 4.3 mostra um exemplo simples, utilizando o *string* "NOS_SOMOS_NOS_SOMOS_NOS#", que possui muitos caracteres repetidos apenas para efeito de demonstração. O caractere '#' representa o fim do *string*.

[16] WELCH, Terry A. A Technique for High-Performance Data Compression. **Computer**, Los Alamitos, v. 17, n. 6, pp. 8-19, Jun. 1984.
[17] ZIV, Jacob; LEMPEL, Abraham. A Universal Algorithm for Sequential Data Compression. **IEEE Transactions on Information Theory**, [S.l.], v. IT-23, n. 3, pp. 337-343, May 1977.

Tabela 4.3 – Demonstração do algoritmo de codificação de Lempel-Ziv-Welch.

entrada	*string*	índice	saída	bin. (dec.)
N	N			
O	O	06 (NO)	N	001 (01)
S	S	07 (OS)	O	010 (02)
_	_	08 (S_)	S	011 (03)
S	S	09 (_S)	_	100 (04)
O	O	10 (SO)	S	011 (03)
M	M	11 (OM)	O	010 (02)
O	O	12 (MO)	M	101 (05)
S	OS			
_	_	13 (OS_)	07 (OS)	111 (07)
N	N	14 (_N)	_	100 (04)
O	NO			
S	S	15 (NOS)	06 (NO)	110 (06)
_	S_			
S	S	16 (S_S)	08 (S_)	1000 (08)
O	SO			
M	M	17 (SOM)	10 (SO)	1010 (10)
O	MO			
S	S	18 (MOS)	12 (MO)	1110 (12)
_	S_			
N	N	19 (S_N)	08 (S_)	1000 (08)
O	NO			
S	NOS			
#	#	20 (NOS#)	15 (NOS)	1111 (15)
			00 (#)	0000 (00)

Considerando que o *string* escolhido possui 24 caracteres (contando com o símbolo de fim ‘#’) e assumindo que cada caractere é representados por 3 bits, tem-se que para transmitir o referido *string* sem codificação são necessários 72 bits. Com a versão codificada, são necessários apenas 54 bits, o que representa uma taxa de compressão de 25%. Para *strings* de entrada mais longos, como ocorre na realidade, as taxas de compressão atingidas podem ser muito maiores. Também é importante ressaltar que o caractere que representa o fim do *string* sempre receberá o índice zero.

O processo de decodificação consiste em reconstruir o mesmo dicionário criado durante a codificação a partir dos índices recebidos e em traduzi-los (decodificá-los) para que se obtenha o *string* original. Um pseudocódigo de descompressão utilizando o LZW é mostrado na Figura 4.8.

```
lê IND_ANT
transmite IND_ANT
ENQUANTO existirem caracteres para ser lidos FAÇA
      lê IND_NOVO
      S = tradução do IND_NOVO
      transmite S
      C = primeiro caractere de S
      adiciona IND_ANT+C à tabela de tradução
      IND_ANT = IND_NOVO
FIM
```

Figura 4.8 – Pseudocódigo do algoritmo de decodificação de Lempel-Ziv-Welch.

A fim de tornar o algoritmo de decodificação mais claro, a Tabela 4.4 mostra um exemplo para decodificar o *string* codificado no exemplo anterior. A entrada do algoritmo é a seqüência comprimida de 54 bits.

Tabela 4.4 – Demonstração do algoritmo de decodificação de Lempel-Ziv-Welch.

entrada	*string*	índice	saída
N	N		N
O	O	06 (NO)	O
S	S	07 (OS)	S
_	_	08 (S_)	_
S	S	09 (_S)	S
O	O	10 (SO)	O
M	M	11 (OM)	M
		12 (MO)	
07 (OS)	OS		OS
_	_	13 (OS_)	_
		14 (_N)	
06 (NO)	NO		NO
		15 (NOS)	
08 (S_)	S_		S_
		16 (S_S)	
10 (SO)	SO		SO
		17 (SOM)	
12 (MO)	MO		MO
		18 (MOS)	

entrada	*string*	índice	saída
08 (S_)	S_		S_
		19 (S_N)	
15 (NOS)	NOS		NOS
00 (#)	#	20 (NOS#)	#

O LZW, no entanto, apresenta um caso de exceção, que ocorre quando há no dicionário uma seqüência do tipo "*string* + caractere" e é lida da entrada uma seqüência do tipo "*string* + caractere + *string* + caractere + *string*". Para tornar a explicação mais clara, a Tabela 4.5 mostra como ficaria a tabela de codificação para o *string* "ABCABCABCABCABCABC#". Novamente o símbolo '#' representa o fim do *string*.

Tabela 4.5 – Demonstração do algoritmo de codificação de Lempel-Ziv-Welch para uma situação em que ocorre exceção.

entrada	*string*	índice	saída	bin. (dec.)
A	A			
B	B	04 (AB)	A	01 (01)
C	C	05 (BC)	B	10 (02)
A	A	06 (CA)	C	11 (03)
B	AB			
C	C	07 (ABC)	04 (AB)	100 (04)
A	CA			
B	B	08 (CAB)	06 (CA)	110 (06)
C	BC			
A	A	09 (BCA)	05 (BC)	101 (05)
B	AB			
C	ABC			
A	A	10 (ABCA)	07 (ABC)	111 (07)
B	AB			
C	ABC			
A	ABCA			
B	B	11 (ABCAB)	10 (ABCA)	1010 (10)
C	BC			
#	#	12 (BC#)	05 (BC)	0101 (05)
			00 (#)	0000 (00)

Visto que tal exceção só é detectada no processo de decodificação, tem-se na Tabela 4.6 como ficaria a decodificação do *string* "ABCABCABCABCABCABC#" codificado com o LZW.

Tabela 4.6 – Demonstração do algoritmo de decodificação de Lempel-Ziv-Welch para uma situação em que ocorre exceção.

entrada	*string*	índice	saída
A	A		A
B	B	04 (AB)	B
C	C	05 (BC)	C
		06 (CA)	
04 (AB)	AB		AB
		07 (ABC)	
06 (CA)	CA		CA
		08 (CAB)	
05 (BC)	BC		BC
		09 (BCA)	
07 (ABC)	ABC		ABC
10 (ABCA)	não existe	10 (ABCA)	ABCA
05 (BC)	BC	11 (ABCAB)	BC
00 (#)	#	12 (BC#)	#

Verifica-se na Tabela 4.6 que quando o índice 10 é lido na entrada, ele ainda não existe no dicionário, não havendo como traduzi-lo para o *string* “ABCA”. O que o algoritmo modificado faz para resolver esse problema é acrescentar ao final da entrada anterior, nesse caso “ABC”, o primeiro caractere dessa seqüência, finalmente adicionando o novo índice ao dicionário. Um pseudocódigo do algoritmo de descompressão de LZW modificado é mostrado na Figura 4.9.

4.6. Transformada de Burrows-Wheeler

A transformada de Burrows-Wheeler (em inglês, *Burrows-Wheeler Transform* ou BWT) é uma transformada de ordenamento de blocos reversível que auxilia na compressão sem perdas de dados utilizando permutações entre os símbolos desses blocos (Effros et al., 2002). O algoritmo em si não realiza a compressão dos dados, mas por meio de reordenamento ele a torna mais fácil, quando utilizado em conjunto com outros algoritmos, como a transformada Move-to-Front (MTF) (Burrows e Wheeler, 1994), que será descrita mais adiante.

A BWT é considerada uma abordagem completamente nova na compressão de dados (Manzini, 1999) e ela é a base para os melhores e mais modernos compressores sem perdas existentes (Manzini, 1999, 2001). Além disso, o algoritmo da BWT é tão rápido quanto as

técnicas de Lempel e Ziv, mas consegue atingir taxas de compressão comparáveis aos melhores compressores baseados em estatística (Burrows e Wheeler, 1994). Outro relato (Manzini, 1999) afirma que estudos feitos revelaram que o uso de recursos computacionais por parte do algoritmo (tempo de processamento e espaço em memória) é relativamente baixo comparado às excelentes taxas de compressão obtidas.

```
lê IND_ANT
transmite IND_ANT
C = IND_ANT
ENQUANTO existirem caracteres para ser lidos FAÇA
      lê IND_NOVO
      SE IND_NOVO não estiver na tabela ENTÃO
            S = tradução do IND_ANT
            S = S+C
      SENÃO
            S = tradução do IND_NOVO
      FIM
      transmite S
      C = primeiro caractere de S
      adiciona IND_ANT+C à tabela de tradução
      IND_ANT = IND_NOVO
FIM
```

Figura 4.9 – Pseudocódigo do algoritmo de decodificação de Lempel-Ziv-Welch modificado para tratar a exceção.

O algoritmo de codificação consiste em construir uma tabela indexada contendo todas as rotações possíveis do bloco de entrada e, a seguir, ordená-la lexicograficamente. As saídas do algoritmo são a última coluna da tabela e o índice da tabela onde se encontra o bloco na sua ordem original. A Figura 4.10 ilustra o processo de codificação da BWT utilizando o *string* “pessoa” como bloco de entrada.

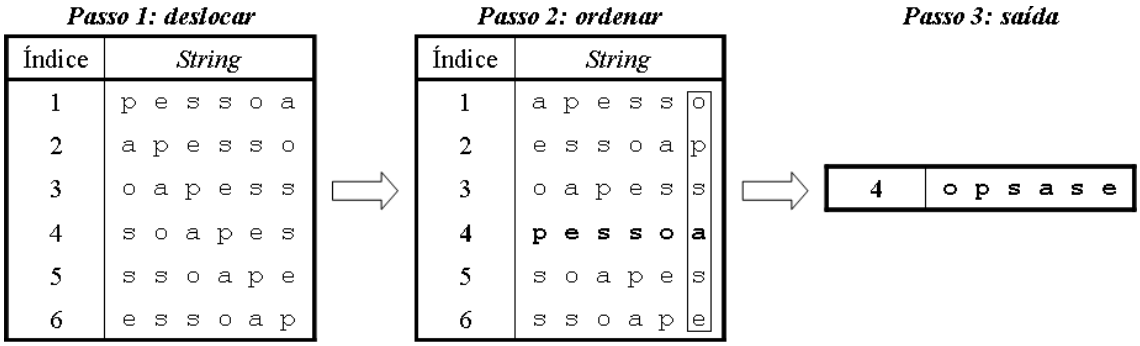


Figura 4.10 – Processo de codificação da palavra “pessoa” utilizando a transformada de Burrows-Wheeler.

Deve-se observar na Figura 4.10 que a palavra original e seu respectivo índice estão destacados em negrito no passo 2 e que a última coluna da tabela foi contornada por um retângulo para deixar claro que ela é uma das saídas do algoritmo, conforme é mostrado no passo 3. O algoritmo de decodificação é um pouco mais complexo do ponto de vista computacional. Nele são realizados $N-1$ ordenamentos lexicográficos, onde N é o número de símbolos contidos no bloco de entrada. A Figura 4.11 ilustra a primeira, a segunda e a última iterações do processo de decodificação da palavra “pessoa” codificada anteriormente.

O passo 1 consiste em pôr na primeira coluna da tabela o bloco codificado; o passo 2 é onde se realiza o ordenamento lexicográfico dos elementos desse bloco; no passo 3, a lista de elementos ordenados é copiada para as primeiras colunas da tabela. A cada iteração os três passos se repetem até que o bloco original esteja completamente reconstruído. Quando isso ocorre, utiliza-se o índice obtido no processo de codificação para encontrar qual das linhas da tabela corresponde ao bloco original. Na Figura 4.11, isso é mostrado na quinta iteração, em que a linha contendo a palavra “pessoa” está destacada em negrito.

4.7. Transformada Move-to-Front

A transformada Move-to-Front (MTF) foi desenvolvida para melhorar a eficiência de técnicas de compressão baseadas em entropia, como é o caso do código de Huffman. Tal transformada assume que os símbolos de um alfabeto são representados pela sua posição dentro de uma lista ordenada (em geral lexicograficamente). Cada símbolo da seqüência de entrada ao ser codificado é movido para a frente daquela lista, cuja primeira posição é representada pelo índice zero.

Iteração 1			
Índice	Passo 1	Passo 2	Passo 3
1	o	a	a····o
2	p	e	e····p
3	s	o	o····s
4	a	p	p····a
5	s	s	s····s
6	e	s	s····e

Iteração 2			
Índice	Passo 1	Passo 2	Passo 3
1	oa	ap	ap···o
2	pe	es	es···p
3	so	oa	oa···s
4	ap	pe	pe···a
5	ss	so	so···s
6	es	ss	ss···e

Iteração 5			
Índice	Passo 1	Passo 2	Passo 3
1	oapes	apess	apesso
2	pesso	essoa	essoap
3	soape	oapes	oapess
4	apess	pesso	**pessoa**
5	ssoap	soape	soapes
6	essoa	ssoap	ssoape

Figura 4.11 – Processo de decodificação da palavra “pessoa” utilizando a transformada de Burrows-Wheeler.

O primeiro passo do processo de codificação da MTF é criar a lista contendo os símbolos pertencentes à seqüência de entrada ordenados de alguma maneira e definir um vetor de inteiros que armazenará as posições dos símbolos à medida que aquela seqüência é codificada (Burrows e Wheeler, 1994). Após, deve-se percorrer a seqüência de entrada do início ao fim, guardando a posição do símbolo atual no vetor de inteiros e, então, movendo-o para o início da lista (Burrows e Wheeler, 1994).

O processo de decodificação é tão simples quanto o de codificação. Primeiro deve-se criar a lista contendo os símbolos da seqüência original e um vetor de símbolos que armazenará aqueles que forem sendo decodificados. A seguir, percorre-se o vetor dos índices, guardando no vetor de símbolos o símbolo correspondente ao índice atual e movendo esse símbolo para a frente da lista ordenada. Os processos de codificação e de decodificação da MTF podem ser visualizados na Figura 4.12.

S = “pessoa”	
L = [‘a’, ‘e’, ‘o’, ‘p’, ‘s’]	V = [3]
L = [‘p’, ‘a’, ‘e’, ‘o’, ‘s’]	V = [3, 2]
L = [‘e’, ‘p’, ‘a’, ‘o’, ‘s’]	V = [3, 2, 4]
L = [‘s’, ‘e’, ‘p’, ‘a’, ‘o’]	V = [3, 2, 4, 0]
L = [‘s’, ‘e’, ‘p’, ‘a’, ‘o’]	V = [3, 2, 4, 0, 4]
L = [‘o’, ‘s’, ‘e’, ‘p’, ‘a’]	V = [3, 2, 4, 0, 4, 4]

V = [3, 2, 4, 0, 4, 4]	
L = [‘a’, ‘e’, ‘o’, ‘p’, ‘s’]	S = “p · · · · ·”
L = [‘p’, ‘a’, ‘e’, ‘o’, ‘s’]	S = “pe · · · ·”
L = [‘e’, ‘p’, ‘a’, ‘o’, ‘s’]	S = “pes · · ·”
L = [‘s’, ‘e’, ‘p’, ‘a’, ‘o’]	S = “pess · ·”
L = [‘s’, ‘e’, ‘p’, ‘a’, ‘o’]	S = “pesso ·”
L = [‘o’, ‘s’, ‘e’, ‘p’, ‘a’]	S = “pessoa”

Figura 4.12 – Processos de codificação (em cima) e de decodificação (em baixo) da palavra “pessoa” usando a transformada Move-to-Front.

4.8. Exemplo do uso da BWT+MTF

Para justificar o uso da MTF depois de aplicar a BWT a fim de melhorar a taxa de compressão de um sistema, será utilizado um bloco maior do que os seis bytes da palavra “pessoa”. Primeiramente a frase “O elefante elegante subiu na estante.”, que possui 37 bytes, será transformada com a BTW, depois com a MTF e por fim com BWT seguida da MTF. O resultado pode ser visto na Tabela 4.7.

Tabela 4.7 – Resultado das transformadas BWT, MTF e BWT+MTF sobre um bloco de 37 bytes.

Frase Original	`O elefante elegante subiu na estante.`
Com a BWT	`Oeaueenfgtutttll  eebee aaa.e snnnis`
Com a MTF	2, 1, 5, 9, 1, 7, 6, 10, 12, 4, 6, 1, 6, 1, 10, 6, 6, 6, 4, 6, 12, 13, 12, 13, 2, 4, 7, 8, 2, 7, 7, 8, 4, 5, 2, 4, 13
Com BWT+MTF	2, 5, 4, 13, 2, 0, 11, 8, 9, 13, 5, 1, 0, 0, 12, 0, 9, 0, 0, 7, 0, 11, 1, 0, 2, 9, 0, 0, 11, 3, 3, 13, 11, 0, 0, 13, 2

Como se observa na Tabela 4.7, a BWT tende a reunir símbolos iguais, o que favorece a utilização da MTF, uma vez que essa transformada sempre atribui o valor zero a partir do segundo símbolo repetido consecutivamente. Isso acaba aumentando a redundância dos dados e facilitando sua compressão. A Tabela 4.8 traz a distribuição de probabilidades da seqüência transformada somente com a MTF e com a BWT seguida da MTF.

Tabela 4.8 – Distribuições de probabilidades do bloco de 37 bytes codificado só com a MTF e com a BWT seguida da MTF.

i	p_i (só MTF)	p_i (BWT+MTF)
0	0,000	0,325
1	0,108	0,054
2	0,108	0,108
3	0,000	0,054
4	0,136	0,027
5	0,054	0,054
6	0,189	0,000
7	0,108	0,027
8	0,054	0,027
9	0,027	0,081
10	0,054	0,000
11	0,000	0,108
12	0,081	0,027
13	0,081	0,108

Se as probabilidades da Tabela 4.8 fossem de fontes de dados sem codificação, suas entropias, calculadas com a Equação (4.4), seriam 3,2963 e 3,1060 bits por símbolo, respectivamente. A diferença pode parecer pouca (aproximadamente 6%), mas se o tamanho do bloco for de alguns quilobytes em vez de apenas 37 bytes, o ganho na taxa de compressão pode ser muito mais significativo (Burrows e Wheeler, 1994).

5. Proposta de Trabalho e Metodologia

Este trabalho tem como objetivo o desenvolvimento de um sistema de esteganografia em áudio capaz de armazenar dados com alta taxa de bits, sem degradar perceptualmente sua qualidade. O sistema utiliza uma técnica de espalhamento espectral (em inglês, *spread spectrum* ou SS), originalmente aplicada em sistemas de comunicação militares (Sklar, 2001), e técnicas de compressão de dados sem perdas, como o código de Huffman e o algoritmo de Lempel-Ziv-Welch. A Figura 5.1 mostra o esquema de um sistema de esteganografia genérico para áudio baseado em SS.

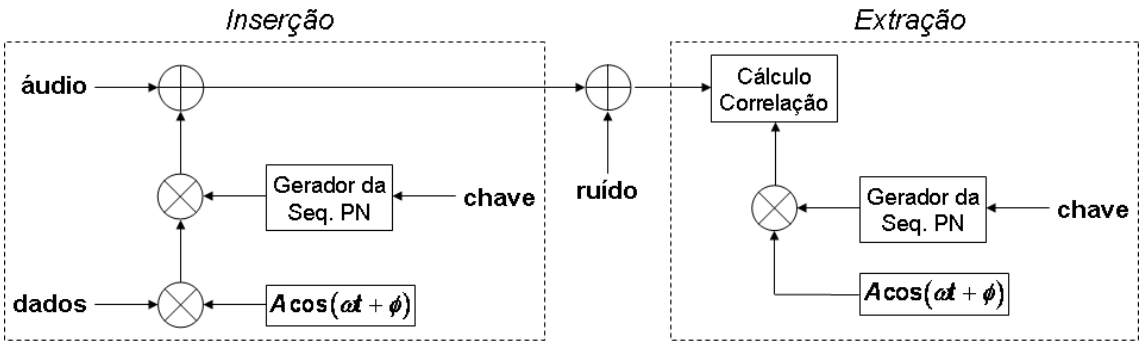


Figura 5.1 – Esquema de um sistema de esteganografia genérico em áudio baseado em uma técnica de espalhamento espectral.

No esquema da Figura 5.1, supõe-se que depois que os dados são inseridos no sinal de áudio, esse sinal é transmitido através de um canal que adiciona ruído a ele. Essa não é a situação que será mostrada neste trabalho, pois os arquivos de áudio utilizados nos testes foram apenas armazenados em um disco rígido. A seguir, o sistema será descrito em maiores detalhes.

5.1. Descrição do Sistema Proposto

O sistema que será apresentado a seguir é composto basicamente por dois grandes módulos: o de inserção e o de extração. O primeiro módulo é onde o sinal hospedeiro (áudio) e o sinal que carrega os dados são tratados por um modelo psicoacústico e, por fim, adicionados um ao outro. O segundo, é onde os dados inseridos no sinal de áudio são recuperados a fim de extrair a informação neles contida.

5.1.1. Módulo de Inserção

O módulo de inserção (MI) é formado por diversos blocos, conforme mostra a Figura 5.2. O bloco mais importante e complexo de todo o esquema é do modelo psicoacústico (ou modelo perceptual), o qual permite que não ocorra degradação na qualidade perceptual do sinal hospedeiro após a inserção dos dados. Esse bloco foi explicado em detalhes no Capítulo 2.

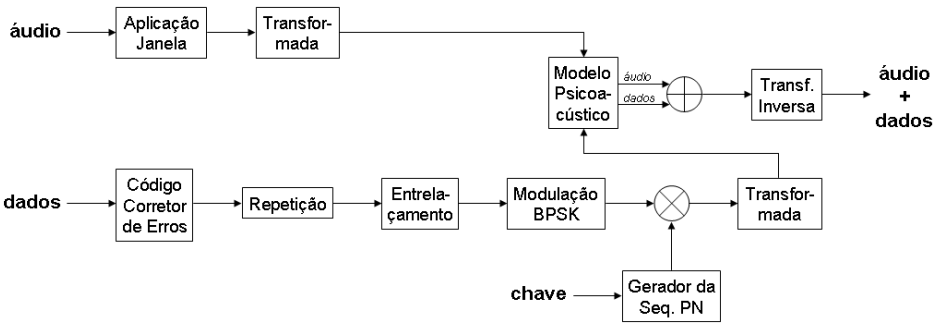


Figura 5.2 – Diagrama esquemático do módulo de inserção.

O MI possui basicamente três entradas (sinal de áudio, sinal de dados e chave ou semente da seqüência PN) e uma saída (sinal de áudio com os dados inseridos), além de alguns parâmetros que são passados a certos blocos intermediários. O sinal de áudio passa primeiramente por uma janela de Hamming de aproximadamente 93 ms e é, então, transformado para o domínio da freqüência utilizando-se a transformada rápida de Fourier (em inglês, *fast Fourier transform* ou FFT). O fator de sobreposição das janelas é de 75%.

Os dados são inicialmente lidos de um arquivo e convertidos em um *stream* de bits bipolar, isto é, que assume os valores “+1” ou “–1”. Esse *stream* de bits é dividido em blocos de aproximadamente 50 bits (esse valor pode sofrer variações, como será explicado mais

adiante). Ao início de cada um desses blocos concatena-se, em valores binários, um número seqüencial que indica a ordem exata em que os blocos foram inseridos no sinal hospedeiro, a fim de facilitar sua extração posterior. Esses passos iniciais não foram incluídos no esquema da Figura 5.2 por razões de simplificação e de clareza do desenho. O bloco de dados, assim como mostrado no esquema, deve conter exatamente 57 bits ao chegar ao código corretor de erros (em inglês, *error-correcting code* ou ECC). Isso deve-se ao fato de que o ECC utilizado é o BCH (63, 57). Uma vez que o foco deste trabalho não são ECCs, que foram utilizados apenas como uma ferramenta, far-se-á uma breve explanação a respeito do BCH no Apêndice A.

Depois de passar pelo ECC, cada bit do bloco de dados é repetido de acordo com um fator *m* de vezes (explicado mais adiante) e, então, o bloco sofre um entrelaçamento. O motivo do entrelaçamento é de fazer com que a interferência de *jamming* por pulsos em cada bit de dados repetido seja independente, resultando, assim, em uma probabilidade de erros de detecção muito mais baixa (Garcia, 1999). A probabilidade de erros para cada um dos *m* bits é dada na Equação (5.1) e a probabilidade de erros para cada bit de dados é mostrada na Equação (5.2). Tem-se que E_b/N_0 é a relação sinal-ruído do sistema, ρ é a fração de tempo durante a qual um pulso interfere no sistema e $Q(\cdot)$ é a função erro complementar definida na Equação (3.23).

$$\varepsilon = \rho Q\left(\sqrt{\frac{2E_b}{mN_0}\rho}\right) \tag{5.1}$$

$$P_B = \sum_{k=\frac{m+1}{2}}^{m} \binom{m}{k} \varepsilon^k (1-\varepsilon)^{m-k} \tag{5.2}$$

Dado que *m* deve ser um valor ímpar, a Equação (5.2) pode ser entendida como a probabilidade de ocorrer um erro na estimação de um bit de dados é igual à probabilidade de que $(m+1)/2$ ou mais dos *m* bits repetidos sejam detectados erroneamente.

O entrelaçamento de um bloco de dados dá-se por meio de uma matriz. A matriz de entrelaçamento possui *H* linhas e *I* colunas. O *stream* de bits, depois de ter cada bit repetido, é escrito na matriz coluna por coluna e depois é lido linha por linha. Para efeito de ilustração, suponha-se que o bloco de dados depois de passar pelo ECC seja {+1, +1, –1, +1}. Considerando um fator de repetição $m = 3$, tem-se {+1, +1, +1, +1, +1, +1, –1, –1, –1, +1,

+1, +1}. Aplicando uma matriz de entrelaçamento com $H = 4$ e $I = 3$, obtém-se {+1, +1, –1, +1, +1, +1, +1, –1, +1, +1, –1, +1}. Se o tamanho do bloco for menor do que a dimensão da matriz, os espaços que sobram são preenchidos com "+1". Ao final do entrelaçamento, um cabeçalho contendo 24 valores "+1" é concatenado ao início do bloco para fins de sincronização durante a extração.

O bloco de dados entrelaçado é modulado utilizando-se modulação BPSK. Após, é feito o espalhamento espectral multiplicando o sinal modulado pela seqüência PN gerada a partir de uma chave ou semente. O sinal resultante passa, então, por uma janela de Hamming de aproximadamente 93 ms (também omitida da Figura 5.2 por razões de simplificação e de clareza) e é transformado para o domínio da freqüência por meio da FFT.

Ambos no domínio da freqüência, os sinais de áudio e de dados são moldados de acordo com o modelo psicoacústico, somados, e o resultado é transformado de volta ao domínio do tempo. O novo sinal de áudio é gravado em um arquivo PCM Wave com o mesmo número de bits por amostra e mesma taxa de amostragem do arquivo de áudio original.

5.1.2. Módulo de Extração

O módulo de extração (ME), assim como o MI, também é composto por diversos blocos. A Figura 5.3 mostra um diagrama esquemático daquele módulo. Nesse caso, o bloco mais importante e mais complexo do esquema é o de sincronização. Esse bloco foi explicado em detalhes no Capítulo 3.

O ME possui duas entradas (sinal de áudio com dados e chave ou semente da seqüência PN) e uma saída (estimativa dos dados originalmente inseridos), além de alguns parâmetros que são passados a certos blocos intermediários. O sinal de áudio com dados, que pode ou não ter sofrido interferência de ruído aditivo, passa primeiramente por uma janela de Hamming de 93 ms aproximadamente para depois ser transformado para o domínio da freqüência por meio da FFT. O fator de sobreposição das janelas também é de 75%.

O mesmo cabeçalho de 24 valores "+1" é modulado utilizando-se modulação BPSK e depois é multiplicado pela seqüência PN gerada a partir da chave para realização do espalhamento espectral. Esse sinal é, então, transformado para o domínio da freqüência por meio da FFT.

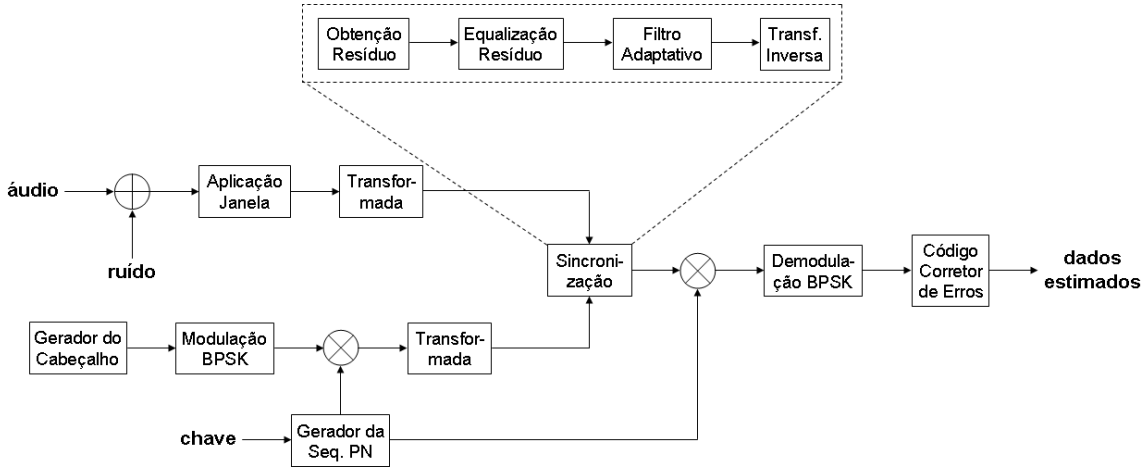


Figura 5.3 – Diagrama esquemático do módulo de extração.

No bloco de sincronização, obtém-se um sinal residual a partir do sinal de áudio com dados. Esse sinal residual é formado pelas componentes em freqüência do sinal de áudio que se encontram abaixo do limiar de mascaramento calculado com o modelo psicoacústico, ou seja, são aquelas que efetivamente contêm os dados ocultados. O sinal residual deve ser equalizado para que seu espectro tenha todas as componentes presentes com mesma magnitude (Garcia, 1999). O fator de equalização, dado na Equação (5.3), é semelhante ao fator de conformação do ruído apresentado na Equação (2.16). O denominador da equação denota o valor máximo do espectro de magnitude do sinal residual $R(\omega)$ em cada uma das bandas críticas z da escala Bark.

$$G_z = \frac{1}{\max\left(\left|R(\omega)\right|\right)}, \quad \text{para } 1 \le z \le z_t \tag{5.3}$$

Para a sincronização propriamente dita, utiliza-se um filtro adaptativo de detecção de alta resolução classe 1/3. Esse filtro foi apresentado na Equação (3.30). O cabeçalho gerado no ME deve estar perfeitamente alinhado com o cabeçalho do sinal residual para que a extração dos dados possa ser realizada corretamente. Depois que o início de um bloco de dados é encontrado, isto é, a posição onde o pico da resposta do filtro no domínio do tempo é descoberta, o sinal residual deve ser multiplicado pela mesma seqüência PN utilizada anteriormente para que o espalhamento espectral seja desfeito. A seguir, realiza-se a demodulação BPSK e remove-se o cabeçalho do bloco de dados. A detecção dos bits repetidos é feita com a Equação (3.35). Deve-se, então, fazer um desentrelaçamento, desta vez escrevendo o *stream* de bits na matriz linha por linha e depois lendo-o coluna por coluna.

Os bits de dados originais são obtidos com a Equação (5.4), onde *m* é o fator de repetição, T_b é o tempo de duração de um bit e $\hat{d}(t)$ é a estimação obtida com a Equação (3.35).

$$u(t) = \begin{cases} 1, & \text{se } \int_0^{(m-1)T_b} \hat{d}(t)\,dt > 0 \\ -1, & \text{se } \int_0^{(m-1)T_b} \hat{d}(t)\,dt \le 0 \end{cases} \tag{5.4}$$

Por fim, o *stream* de bits obtido passa pelo ECC para verificar se houve algum erro de decisão e, caso haja no máximo um erro, o BCH (63, 57) é capaz de corrigi-lo.

5.2. Implementação do Sistema Proposto

O sistema descrito na seção anterior foi desenvolvido totalmente em MATLAB®, versão 7.1. A escolha dessa ferramenta deu-se porque as bibliotecas que acompanham o programa são confiáveis e fáceis de se usar. A manipulação de arquivos de áudio PCM Wave resume-se ao uso de duas funções: uma de leitura e outra de escrita. As bibliotecas que contêm as transformadas também são de utilização simplificada.

O ponto principal do programa, porém, são as funções e operações de manipulação de vetores e matrizes. Caso o sistema tivesse sido implementado em uma linguagem de programação comum, por exemplo C, uma biblioteca teria que ser desenvolvida do zero para manipular essas estruturas, ou então seria necessário utilizar bibliotecas de terceiros, nem sempre confiáveis e fáceis. A linguagem C++ já vem com uma classe para manipulação de vetores, mas sua utilização não é tão simples quanto à do MATLAB®.

Outro aspecto vantajoso desse programa são as funções para geração e manipulação de gráficos. Muitas vezes, acaba-se descobrindo um erro no código desenvolvido apenas observando um gráfico gerado a partir dos dados sendo trabalhados. Algo que demoraria muito tempo para ser encontrado em um código em C pode ser rapidamente resolvido com o MATLAB®.

Há, no entanto, uma grande desvantagem no uso do MATLAB®. Por se tratar de um programa que utiliza linguagem interpretada e não compilada, o tempo de execução de um código em MATLAB® é muito maior do que o de um código escrito em C e compilado. Além do mais, esse programa custa várias centenas de dólares enquanto que compiladores C e C++ são gratuitos e existem diversas opções.

5.3. Ferramentas Utilizadas

Além do MATLAB®, que foi utilizado para implementar o sistema desenvolvido, mais cinco ferramentas de apoio foram utilizadas. Um conjunto de ferramentas de compressão de dados escritas em linguagem C por Michael Dipperstein[18], de código-fonte aberto e disponível gratuitamente para download, foi utilizado para gerar os arquivos de dados comprimidos. Para compilar os arquivos-fonte dos programas de compressão, foi utilizado o ambiente de programação Bloodshed Dev-C++[19] 4.9.9.2, que é baseado no GCC, compilador C/C++ de código-fonte aberto e gratuito.

Os testes de qualidade do áudio, que serão comentados mais adiante, foram realizados com o PQevalAudio[20]. Essa ferramenta foi totalmente desenvolvida em MATLAB®. Ela é descrita em (Kabal, 2002) e pode ser obtida gratuitamente. A extração dos arquivos de testes a partir de CDs de áudio foi feita com o Windows® Media Player 11 e sua edição foi realizada com o Audacity[21] 1.2.6, uma ferramenta para edição de áudio com código-fonte aberto e gratuita para download.

5.4. Amostras para Testes

Foram selecionadas para a realização dos testes de capacidade e de qualidade perceptual duas amostras de texto do conhecido corpus de compressão Calgary[22] e seis amostras de áudio, obtidas a partir de CDs de áudio, com estilos musicais variados. Os arquivos de texto foram usados como fontes de dados para o sistema de esteganografia. Tais arquivos possuem características completamente distintas em termos de distribuição de probabilidades de caracteres. Cada um deles gerou três arquivos comprimidos, que foram inseridos nos excertos de áudio testados.

A escolha dessas amostras de texto deu-se devido a sua natureza diferente e também porque ambos os arquivos possuem tamanhos semelhantes. A Tabela 5.1 resume as informações sobre esses arquivos bem como traz dados relativos à compressão dos mesmos

---

[18] Disponível em: <http://michael.dipperstein.com>. Acesso em: 3 dez. 2008.
[19] Disponível em: <http://www.bloodshed.net/dev/devcpp.html>. Acesso em: 10 abr. 2008.
[20] Disponível em: <http://www-mmsp.ece.mcgill.ca/Documents/Downloads/PQevalAudio>. Acesso em: 30 out. 2008.
[21] Disponível em: <http://audacity.sourceforge.net>. Acesso em: 12 fev. 2008.
[22] WITTEN, Ian H.; BELL, Timothy C.; CLEARY, John G. Calgary Compression Corpus. 1987. Disponível em: <ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus>. Acesso em: 2 dez. 2008.

com os diferentes métodos apresentados no Capítulo 4. Os valores entre parênteses são os respectivos percentuais de compressão.

Tabela 5.1 – Descrição e tamanhos dos arquivos de texto originais e comprimidos.

Nome do arquivo	paper3	progp
Descrição	texto com estilo de artigo	código-fonte em Pascal
Tamanho original (bytes)	46526	49379
Tamanho comprimido (bytes)		
Código de Huffman	27702 (40,46%)	30666 (37,90%)
Lempel-Ziv-Welch	22149 (52,39%)	19195 (61,13%)
BWT+MTF+Huffman	22428 (51,79%)	17077 (65,42%)

As amostras de áudio, como já foi mencionado anteriormente, foram extraídas de CDs de áudio. Elas têm 16 bits por amostra e sua freqüência de amostragem é de 44100 Hz. A duração dos trechos é de 100 segundos, obtidos a partir do começo de cada uma das respectivas faixas, incluindo o silêncio inicial. Apenas um canal de cada música foi utilizado nos testes. Os estilos musicais avaliados são jazz, rock, blues e música clássica. Na Tabela 5.2 podem-se verificar os detalhes das amostras.

Tabela 5.2 – Detalhes das amostras de áudio selecionadas.

Nome do arquivo	Nº da faixa/Título/Álbum/Artista	Características
de04r	04/ Take the “A” Train/The History of Jazz/Duke Ellington	Jazz, canal direito
dt01l	01/Overture/Six Degrees of Inner Turbulence (CD2)/Dream Theater	Rock orquestrado (instrumental), canal esquerdo
ec02r	02/Before You Accuse Me/Unplugged/Eric Clapton	Blues, apresentação ao vivo, violão com cordas de aço, ruído da platéia (palmas), canal direito
gr10l	10/November Rain/Use Your Illusion I/Guns ’n’ Roses	Balada de rock, solo de piano com acompanha-mento de teclado eletrônico e vocal, canal esquerdo
pu01l	01/IX Sinfonia de Beethoven – 4º Movimento/Concertos Comunitários Zaffari Vol. 1/Coral e Orquestra da PUCRS	Música Clássica, orquestra completa e coral, final do 4º Movimento, canal esquerdo

Nome do arquivo	Nº da faixa/Título/Álbum/Artista	Características
qu01r	01/We Will Rock You/Queen Collection/Queen	Rock, percussão, palmas e vocal, guitarra distorcida no final, canal direito

5.5. Testes Iniciais

Os primeiros testes começaram a ser realizados quando o sistema ainda estava em desenvolvimento. O intuito desses testes era avaliar que influência alguns parâmetros do módulo de inserção (MI) tinham sobre a qualidade perceptual do áudio e sobre a probabilidade de ocorrência de erros de detecção a fim de se escolher os parâmetros mais adequados para os testes consecutivos. Aqueles testes também serviram para avaliar o tamanho ideal de um bloco de dados a fim de que houvesse a melhor relação entre a quantidade de dados inserida e a capacidade de recuperação desses dados. A Tabela 5.3 lista todos os parâmetros do sistema.

Tabela 5.3 – Listagem de parâmetros do sistema de esteganografia.

Símbolo	Parâmetro	Unidade	Descrição
b	taxa de bits de dados	bps	É a taxa de bits com que os dados são inseridos no sinal hospedeiro.
A	atenuação do sinal de dados	dB	É o nível de atenuação do sinal de dados.
f_0	freqüência da portadora do sinal de dados	Hz	É a freqüência da portadora na modulação BPSK.
m	fator de repetição de bits	–	É o número de vezes que os bits do bloco de dados original são repetidos.
H	número de linhas da matriz	–	É o número de linhas da matriz de entrelaçamento.

Símbolo	Parâmetro	Unidade	Descrição
I	número de colunas da matriz	–	É o número de colunas da matriz de entrelaça-mento.
N	fator de espalhamento	–	É o número de *chips* da seqüência PN com mesma duração de um bit de dados.
a	fator de repetição dos dados	–	É o número de vezes que um conjunto completo de dados é inserido no sinal hospedeiro.
n	tamanho da palavra-código	bit	É o tamanho da palavra-código do BCH.
k	tamanho da palavra	bit	É o tamanho da palavra codifi-cada com o código BCH.
γ	tamanho da palavra do número seqüencial	bit	É o tamanho da palavra binária que representa o número seqüencial concatenado ao início de cada bloco de dados.

Os parâmetros do sistema avaliados nos testes iniciais foram *b*, *A*, *m*, *H* e *I*. Uma vez que o conjunto de dados em todos os testes era composto somente por um bloco de 9 bytes, aumentos no valor de *b* acarretaram aumentos no valor de *a*. Dois conjuntos de dados foram utilizados e, para cada um, cinco baterias de testes foram realizadas. Cada bateria consistia em manter fixo o valor de *b* (e conseqüentemente o de *a*) e ir variando todos os outros parâmetros considerados. A Tabela 5.4 resume os valores utilizados nos testes.

Tabela 5.4 – Listagem de valores dos parâmetros avaliados.

Parâmetro	Valores utilizados
b	{100, 150, 200, 300, 500}
a	{59, 87, 113, 170, 272}
m	{3, 5, 7, 9}
H	{34, 36, 40, 42}
I	{7, 11, 14, 17}
A	{0, 2, 4, 6, 8, ..., 22}*

*Somente com $b = 100$ utilizou-se A até 22 dB. Nas demais baterias só se testou A até 8 dB.

Foram realizados ao todo 248 testes. O arquivo de áudio escolhido foi o “dt01l” e os conjuntos de dados utilizados foram os *strings* “Cristiano” e “Wenceslau”. O limite máximo do parâmetro *A* foi fixado em 22 dB, pois foi o máximo de atenuação do sinal de dados que se conseguiu atingir, sendo ainda possível detectar um *string* sem erros. Com ambos os *strings*, isso foi obtido com os valores $b = 100$, $m = 5$, $H = 36$ e $I = 11$.

5.6. Testes de Capacidade de Inserção

A capacidade de inserção é o foco principal deste trabalho. Os testes realizados para averiguar a capacidade de inserção máxima do sistema partiram dos resultados obtidos nos testes iniciais. Nesses testes, verificou-se que as únicas taxas de bits de dados que apresentaram percentuais de erros de detecção aceitáveis foram $b = 100$ e $b = 150$. Além disso, os outros parâmetros avaliados naqueles testes também influenciaram a detecção correta dos dados e a qualidade perceptual das amostras de áudio. Baseando-se nos resultados dos testes iniciais, o conjunto de valores que melhor atendeu aos requisitos de baixa taxa de erros e de transparência dos dados foi $A = 4$ dB, $m = 9$, $H = 42$ e $I = 17$.

Nos testes definitivos, o tamanho dos blocos de dados utilizados foi de 63 bits, devido a uma limitação do BCH. Por esse motivo, os valores dos parâmetros *H* e *I* foram alterados para 41 e 14, respectivamente. A partir daí, os testes de capacidade de inserção finalmente começaram a ser realizados. Partindo de 100 bps, a taxa de bits de dados foi sendo reduzida em passos de 5 bps, até que se chegou a 70 bps. Visto que a qualidade do áudio não melhorou significativamente abaixo dos 80 bps, manteve-se esse valor como limite mínimo para os testes seguintes. Para obter o limite máximo, partiu-se de 150 bps e foi-se incrementado em passos de 5 bps até chegar em 175 bps. A partir desse valor, o percentual de erros de detecção elevou-se além do limite mínimo aceitável, impossibilitando a detecção correta de um conjunto inteiro de dados. Escolheu-se, então, a taxa de 170 bps para o limite máximo utilizado nos testes seguintes.

Esse número, porém, não representa a taxa de bits efetiva do sistema, uma vez que há informação adicional inserida juntamente com os dados. O cabeçalho de 24 valores “+1”, o número seqüencial binário concatenado ao início do bloco de dados e os bits redundantes do código BCH somam-se, reduzindo a quantidade de dados que pode ser inserida no sinal hospedeiro. A Equação (5.5) mostra a maneira de se calcular a taxa de bits de dados efetiva do sistema. A maioria das variáveis são parâmetros do MI descritos na Tabela 5.3, porém

ainda há um fator de compensação $\xi \approx 0,93$ devido ao cabeçalho e também a razão de compressão $0 < q \leq 1$.

$$b_{ef} = \xi \frac{b(k-\gamma)}{naq} \tag{5.5}$$

5.7. Testes de Qualidade Perceptual

Como foi mencionado anteriormente, a avaliação da qualidade perceptual das amostras de áudio foi feita com a ferramenta PQevalAudio. A implementação desse programa é descrita em (Kabal, 2002), onde também se encontram uma análise crítica da recomendação ITU-R BS.1387 (ITU, 1998-2001) da International Telecommunication Union, também conhecida como *Perceptual Evaluation of Audio Quality* (PEAQ), e algumas sugestões de implementação em certos pontos em que a norma é muito vaga ou omissa.

O método PEAQ possui duas versões: uma básica, cujo modelo do ouvido é baseado apenas na FFT, e uma avançada, que utiliza um banco de filtros para modelar o ouvido humano, além de também utilizar o modelo baseado na FFT (ITU, 1998-2001; Kabal, 2002). Em ambos os casos, variáveis de saída do modelo (em inglês, *model output variables* ou MOVs) são ponderadas e combinadas por uma rede neural treinada a fim de gerar uma nota única para o nível de degradação das amostras de áudio (Kabal, 2002). Essa nota é chamada de *Objective Difference Grade* (ODG) e sua faixa de valores varia de –4,0, o que significa uma distorção muito desagradável, até 0,0, que seria ausência total de distorções.

O PQevalAudio é uma implementação da versão básica do PEAQ. Essa versão possui 11 MOVs, que são calculadas para cada bloco de áudio analisado. Por ter sido desenvolvida no MATLAB®, a ferramenta é lenta, mas a simplicidade do uso compensa a lentidão. O programa teve que ser levemente modificado, pois sua versão original funcionava somente com arquivos PCM Wave amostrados à 48 kHz.

O PEAQ funciona da seguinte maneira: o sinal de áudio que está sendo avaliado é comparado com uma referência, que em geral é o áudio original sem codificação. Todas as MOVs são calculadas em relação ao sinal de referência. Ambos os sinais são processados pelo modelo psicoacústico apresentado na recomendação antes de serem comparados por meio das MOVs.

Além de se utilizar o PQevalAudio para avaliar a qualidade perceptual das amostras testadas, calculou-se a relação entre as densidades espectrais médias de potência dos sinais hospedeiro e de dados após serem moldados pelo modelo psicoacústico a fim de averiguar sua influência na qualidade do áudio. Essa relação informa basicamente o quanto da energia do sinal de áudio é composta pelo sinal de dados.

6. Resultados e Conclusões

Os resultados constantes deste capítulo referem-se aos testes de capacidade e de qualidade perceptual do áudio. Os resultados dos testes iniciais não são mostrados porque eles refletem apenas a escolha dos valores dos parâmetros utilizados nos outros testes, além de constituírem uma grande massa de dados sem maior utilidade ao leitor. Além disso, os testes iniciais foram realizados enquanto o sistema de esteganografia ainda estava sendo desenvolvido e, por isso, seus resultados não têm a mesma representatividade dos resultados finais.

A listagem completa dos parâmetros do sistema está na Tabela 5.3. A Tabela 6.1 mostra apenas os valores que foram utilizados durante os testes definitivos. À exceção da freqüência de modulação, que foi utilizada a mesma mencionada em (Garcia, 1999), todos os outros valores foram escolhidos para extrair a maior eficiência possível do sistema.

Tabela 6.1 – Listagem dos valores dos parâmetros utilizados nos testes de capacidade e de qualidade perceptual do áudio.

Parâmetro	Valor(es) utilizado(s)
b	80 e 170 bps
A	4 dB
f_0	3500 Hz
m	9
H	41
I	14
N	5
a	3
n	63 bits
k	57 bits
γ	5 e 6 bits

6.1. Resultados dos Testes de Capacidade de Inserção

Os testes de capacidade têm o intuito de averiguar qual a máxima quantidade de dados que se consegue inserir no sinal hospedeiro de maneira que não sejam comprometidas a detecção correta de um conjunto inteiro de dados nem a qualidade perceptual do sinal de áudio. Os valores das taxas de bits de dados obtidas foram calculados com a Equação (5.5) a partir das informações contidas na Tabela 5.1. Os resultados são apresentados na Tabela 6.2, para $b = 80$ e $\gamma = 5$, e na Tabela 6.3, para $b = 170$ e $\gamma = 6$. Os valores da razão de compressão *q* são calculados com a Equação (6.1), onde *P%* é o percentual de compressão.

$$q = \frac{100 - P\%}{100} \tag{6.1}$$

Tabela 6.2 – Taxas de bits de dados efetivas para $b = 80$ e $\gamma = 5$.

Método de Compressão	Taxa de Bits Efetiva (bps)	
	paper3	progp
Código de Huffman	34,379	32,961
Lempel-Ziv-Welch	42,999	52,659
BWT+MTF+Huffman	42,464	59,190

Tabela 6.3 – Taxas de bits de dados efetivas para $b = 170$ e $\gamma = 6$.

Método de Compressão	Taxa de Bits Efetiva (bps)	
	paper3	progp
Código de Huffman	71,651	68,695
Lempel-Ziv-Welch	89,615	109,747
BWT+MTF+Huffman	88,500	123,359

Observa-se nas Tabelas 6.2 e 6.3 que a fonte de dados influencia significativamente a taxa de bits efetiva do sistema. Isso se deve ao fato de que o método de compressão é influenciado pelas características das fontes de dados. Além disso, a escolha do método de compressão também é muito importante para que se atinjam taxas de bits efetivas elevadas. Comparando as taxas obtidas com o terceiro método com aquelas obtidas com a codificação de Huffman simples, verifica-se que elas foram 23,5% maiores para o arquivo “paper3” e 79,6% maiores para o arquivo “progp”.

6.2. Resultados dos Testes de Qualidade Perceptual

A qualidade perceptual das amostras de áudio foi avaliada com a ferramenta PQevalAudio (Kabal, 2002). Também se fez uma análise da influência que a quantidade de informação inserida no sinal hospedeiro tem sobre sua transparência, calculando a relação sinal hospedeiro/quantidade de informação (HIR). Os resultados da avaliação do nível de degradação devem apresentar ODGs maiores do que –1,5 para serem considerados aceitáveis. ODGs maiores do que –1,0 são consideradas muito boas.

As Tabelas 6.4 e 6.5 mostram as ODGs das 72 situações avaliadas, sendo que os testes cujos resultados estão na primeira tabela foram realizados com $b = 80$ e $\gamma = 5$, e os outros com $b = 170$ e $\gamma = 6$. As Tabelas 6.6 e 6.7 mostram, respectivamente, os valores de HIR para as mesmas condições mencionadas anteriormente. É importante ressaltar que a HIR é dada em dB. Uma descrição das músicas encontra-se na Tabela 5.2.

Tabela 6.4 – ODGs para $b = 80$ e $\gamma = 5$.

	Métodos de Compressão	Músicas de04r	dt01l	ec02r	gr10l	pu01l	qu01r
paper3	Código de Huffman	-0,700	-0,263	-0,233	-1,580	-0,672	-1,746
	Lempel-Ziv-Welch	-0,692	-0,269	-0,232	-1,578	-0,674	-1,714
	BWT+MTF+Huffman	-0,699	-0,268	-0,234	-1,593	-0,671	-1,738
progp	Código de Huffman	-0,697	-0,268	-0,233	-1,591	-0,664	-1,730
	Lempel-Ziv-Welch	-0,697	-0,267	-0,234	-1,584	-0,672	-1,717
	BWT+MTF+Huffman	-0,699	-0,266	-0,236	-1,585	-0,663	-1,731
	Média	**-0,697**	**-0,267**	**-0,234**	**-1,585**	**-0,669**	**-1,729**
	Variância	**8,27 ×10^{-6}**	**4,57 ×10^{-6}**	**1,87 ×10^{-6}**	**3,50 ×10^{-5}**	**2,15 ×10^{-5}**	**1,49 ×10^{-4}**

Comparando as Tabelas 6.4 e 6.5, pode-se comprovar o que já era esperado: um aumento da taxa de bits acarreta um nível maior de degradação do áudio. Outra constatação facilmente perceptível é que os métodos de compressão não afetam a qualidade perceptual. Isso pode ser verificado observando-se as variâncias nas ODGs para cada uma das músicas testadas. A natureza das fontes de dados também não influenciam na qualidade do áudio.

Em ambos os casos, isto é, tanto para $b = 80$ quanto para $b = 170$, as músicas “gr10l” e “qu01r” apresentaram ODGs abaixo do limite aceitável. Isso certamente ocorreu devido a características particulares dessas faixas. A primeira é um sinal de áudio com energia muito baixa e a segunda possui vários períodos de silêncio ao longo do trecho utilizado. O modelo

psicoacústico descrito neste trabalho não foi capaz de contornar tais peculiaridades, o que levou a um grau de degradação maior do que o máximo aceitável.

Tabela 6.5 – ODGs para $b = 170$ e $\gamma = 6$.

	Métodos de Compressão	Músicas					
		de04r	dt01l	ec02r	gr10l	pu01l	qu01r
paper3	Código de Huffman	-0,856	-0,402	-0,331	-1,927	-0,858	-1,961
	Lempel-Ziv-Welch	-0,859	-0,393	-0,325	-1,911	-0,844	-1,942
	BWT+MTF+Huffman	-0,858	-0,407	-0,329	-1,927	-0,850	-1,940
progp	Código de Huffman	-0,851	-0,399	-0,332	-1,909	-0,850	-1,960
	Lempel-Ziv-Welch	-0,855	-0,396	-0,327	-1,916	-0,839	-1,921
	BWT+MTF+Huffman	-0,850	-0,402	-0,325	-1,909	-0,839	-1,942
	Média	**-0,855**	**-0,400**	**-0,328**	**-1,917**	**-0,847**	**-1,944**
	Variância	**1,34 ×10^{-5}**	**2,46 ×10^{-5}**	**8,97 ×10^{-6}**	**7,27 ×10^{-5}**	**5,51 ×10^{-5}**	**2,19 ×10^{-4}**

Por outro lado, as músicas “dt01l” e “ec02r” apresentaram ODGs excelentes, todas abaixo de –0,5. Isso mostra que as características de cada amostra influenciam fortemente a qualidade perceptual do áudio codificado. As músicas “de04r” e “pu01l” também receberam todas as ODGs abaixo de –1,0, provando que o sistema cumpriu o que foi proposto.

Tabela 6.6 – HIR para $b = 80$ e $\gamma = 5$.

	Métodos de Compressão	Músicas					
		de04r	dt01l	ec02r	gr10l	pu01l	qu01r
paper3	Código de Huffman	14,989	15,213	15,864	16,034	16,519	16,538
	Lempel-Ziv-Welch	15,032	15,236	15,919	16,067	16,535	16,566
	BWT+MTF+Huffman	14,979	15,220	15,885	16,028	16,499	16,535
progp	Código de Huffman	14,992	15,198	15,875	16,032	16,490	16,530
	Lempel-Ziv-Welch	15,039	15,250	15,956	16,076	16,533	16,568
	BWT+MTF+Huffman	14,996	15,216	15,900	16,022	16,520	16,539
	Média	**15,005**	**15,222**	**15,900**	**16,043**	**16,516**	**16,546**
	Variância	**6,13 ×10^{-4}**	**3,35 ×10^{-4}**	**1,13 ×10^{-3}**	**5,07 ×10^{-4}**	**3,28 ×10^{-4}**	**2,75 ×10^{-4}**

Analisando as Tabelas 6.6 e 6.7, verifica-se que o aumento da taxa de bits em mais de 100% não mudou muito a HIR. Se as médias de cada música forem comparadas entre 80 bps e 170 bps, pode-se observar que a maior variação foi de 0,78% na faixa “pu01l”. Em geral,

houve um aumento da HIR devido ao aumento da taxa de bits, exceto na faixa “gr10l”, onde ocorreu uma diminuição de 0,30%.

Tabela 6.7 – HIR para $b = 170$ e $\gamma = 6$.

	Métodos de Compressão	Músicas					
		de04r	dt01l	ec02r	gr10l	pu01l	qu01r
paper3	Código de Huffman	15,085	15,265	15,928	15,992	16,629	16,664
	Lempel-Ziv-Welch	15,042	15,247	15,908	15,955	16,622	16,625
	BWT+MTF+Huffman	15,121	15,307	15,968	16,045	16,687	16,693
progp	Código de Huffman	15,079	15,257	15,930	15,999	16,645	16,652
	Lempel-Ziv-Welch	15,037	15,244	15,902	15,986	16,631	16,632
	BWT+MTF+Huffman	15,091	15,278	15,943	15,996	16,654	16,677
	Média	**15,076**	**15,266**	**15,930**	**15,996**	**16,645**	**16,657**
	Variância	**1,00 ×10^{-3}**	**5,50 ×10^{-4}**	**5,77 ×10^{-4}**	**8,41 ×10^{-4}**	**5,65 ×10^{-4}**	**6,84 ×10^{-4}**

É possível se concluir também que, mantendo a atenuação do sinal de dados em um nível fixo, 4 dB no caso desses testes, a HIR não tem grande influência sobre a qualidade perceptual das amostras de áudio. Músicas com valores médios de HIR muito próximos, como “pu01l” e “qu01r”, obtiveram ODGs completamente diferentes. Da mesma forma, músicas com valores médios de ODG muito próximos, como “de04r” e “pu01l”, obtiveram valores de HIR completamente diferentes.

6.3. Conclusões

Existem muitos estudos na área de ocultação de informações em áudio digital, mas grande parte deles é voltada para marcas d’água usadas em proteção de propriedade, autenticação e detecção de alterações, rastreamento de cópias etc. Este trabalho tratou de abordar um aspecto pouco estudado na literatura, a capacidade de inserção de dados.

Com o uso de técnicas de compressão de dados sem perdas, foi possível aumentar consideravelmente a capacidade de inserção de dados de um sistema de esteganografia baseado na teoria de espalhamento espectral derivada da área de comunicação digital. Além disso, foi utilizado um código corretor de erros, possibilitando atingir taxas de bits ainda maiores, sem acarretar aumento na probabilidade de erros de detecção.

Os testes realizados revelaram que o modelo psicoacústico utilizado para tornar os dados inseridos transparentes ao ouvinte comum não funcionou perfeitamente com todas as amostras de áudio testadas, mas tratou de cumprir seu papel na maioria da situações. Ficou constatado que, dado um fator de atenuação do sinal de dados constante, a relação entre as densidades espectrais de potência dos sinais hospedeiro e de dados não influencia significativamente a qualidade perceptual das faixas de áudio testadas. Tais faixas foram obtidas a partir de CDs de áudio. Elas têm a mesma duração, o mesmo número de bits por amostra e a mesma taxa de amostragem, mas são de estilos musicais diferentes, que incluem jazz, rock, blues e música clássica.

As fontes de dados utilizadas nos testes foram dois arquivos de texto de um conhecido corpus de dados, o Calgary Compression Corpus. Cada um desses arquivos gerou três arquivos comprimidos por métodos diferentes, os quais foram inseridos nos excertos de áudio por meio do método descrito neste trabalho usando-se duas taxas de bits distintas.

O sistema de esteganografia proposto mostrou-se eficiente quanto à capacidade de inserção de dados e quanto à transparência desses dados. Os resultados obtidos tornam sua aplicação interessante em qualquer nicho de mercado em que a venda de conteúdo de áudio possa ser enriquecida por material adicional, seja texto, imagem ou mesmo mais áudio, sem a necessidade de utilizar espaço extra de armazenamento.

6.4. Trabalhos Futuros

Apesar de o sistema apresentado satisfazer o que foi proposto, é possível melhorá-lo ainda mais. A primeira sugestão seria sua implementação para sinais de áudio estereofônicos. O modelo psicoacústico também pode ser aprimorado, ou até substituído, para atender às situações em que ele se mostrou insatisfatório.

Os testes de avaliação da qualidade perceptual podem ser estendidos ao modelo avançado da ITU-R BS.1387 ou ainda fazer uso de outros métodos mais precisos, como fizeram Vanam e Creusere (2005), além de utilizar mais amostras variadas. Segundo Cvejic e Seppänen (2005), outras transformadas como a transformada discreta do cosseno (DCT) ou a transformada *wavelet* discreta (DWT) são mais adequadas para aplicações que trabalham com altas taxas de bits de dados.

Gordy e Bruton (2000) realizaram testes de desempenho implementando diferentes sistemas de esteganografia em MATLAB® e em linguagem C, comprovando que linguagens

compiladas são muito mais ágeis que linguagens interpretadas. Portanto o sistema proposto poderia ser implementado em C para se tornar mais rápido, ou até mesmo ser implementado em hardware. Por fim, alguma técnica de criptografia poderia ser adicionada ao sistema a fim de prover certo grau de segurança às informações ocultadas no sinal de áudio, se assim desejado ou necessário fosse.

Referências Bibliográficas

BRANDENBURG, Karlheinz. Perceptual Coding of High Quality Digital Audio. In: KAHRS, Mark (Ed.); BRANDENBURG, Karlheinz (Ed.). **Applications of Digital Signal Processing to Audio and Acoustics.** Boston, MA: Kluwer Academic Publishers, 2002. 560 pp.

BURROWS, M.; WHEELER, D. J. A Block-Sorting Lossless Data Compression Algorithm. In: Digital Equipment Corporation (SRC Research Report 124), Palo Alto, CA, USA, 1994. Disponível em: <http://www.eecs.harvard.edu/~michaelm/CS222/burrows-wheeler.pdf>. Acesso em: 22 nov. 2008.

COX, Ingemar J.; MILLER, Matt L. Electronic Watermarking: The First 50 Years. In: 4th IEEE WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING, 2001, Cannes, France. **Proceedings**... Cannes: [s.n.], 2001.pp. 225-230. Disponível em: <http://www.ee.ucl.ac.uk/~icox/papers/2001/mmsp01.pdf>. Acesso em: 27 fev. 2008.

CVEJIC, Nedeljko. **Algorithms for Audio Watermarking and Steganography.** 2004. 112 fls. Ph.D. Dissertation. Department of Electrical and Information Engineering, University of Oulu, Oulu, Finland, 2004. Disponível em: <http://herkules.oulu.fi/isbn9514273842/isbn9514273842.pdf>. Acesso em: 1 out. 2007.

CVEJIC, Nedeljko; SEPPÄNEN, Tapio. Watermark Bit Rate in Diverse Signal Domains. **Proceedings of World Academy of Science, Engineering and Technology**, [S.l.], v. 2, pp. 30-33, Jan. 2005. Disponível em: <http://www.waset.org/pwaset/v2/v2-8.pdf>. Acesso em: 22 nov. 2008.

EFFROS, Michelle; VISWESWARIAH, Karthik; KULKARNI, Sanjeev R.; VERDÚ, Sergio. Universal Lossless Source Coding With the Burrows Wheeler Transform. **IEEE Transactions on Information Theory**, [S.l.], v. 48, n. 5, pp. 1061-1081, May 2002. Disponível em: <http://www.ee.princeton.edu/~verdu/reprints/effrosmay02.pdf>. Acesso em: 22 nov. 2008.

FRIGO, Matteo; JOHNSON, Steven G. **FFTW User Manual for version 3.1.2.** Cambridge: MIT, 2006. Disponível em: <http://www.fftw.org/fftw3.pdf>. Acesso em: 22 abr. 2008.

GARCIA, Ricardo A. **Digital Watermarking of Audio Signals Using a Psychoacoustic Auditory Model and Spread Spectrum Theory.** 1999. 116 fls. M.Sc. Thesis. School of Music, University of Miami, Coral Gables, FL, USA, 1999. Disponível em: <http://www.ragomusic.com/publications/ragothesismiami_nocode.pdf>. Acesso em: 22 abr. 2008.

GORDY, J. D.; BRUTON, L. T. Performance Evaluation of Digital Audio Watermarking Algorithms. In: 43rd MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, v.1, 2000, Lansing, MI. **Proceedings**... Lansing, MI, USA: [s.n.], 2000. pp. 456-459. Disponível em: <http://www-mddsp.enel.ucalgary.ca/People/gordy/MWSCAS.PDF>. Acesso em: 28 set. 2007.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU-R BS.1387:** Method for Objective Measurements of Perceived Audio Quality. Geneva, 1998-2001. 101 fls.

JEHAN, Tristan. Music Listening. In: JEHAN, Tristan. **Creating Music by Listening.** 2005. 137 fls. Ph.D. Dissertation. School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005. Disponível em: <http://web.media.mit.edu/~tristan/Papers/PhD_Tristan.pdf>. Acesso em: 15 ago. 2008.

KABAL, P. **An Examination and Interpretation of ITU-R BS.1387**: Perceptual Evaluation of Audio Quality**.** Montreal, 2002. 93 pp. Disponível em: <http://www.mp3-tech.org/programmer/docs/kabalr2002.pdf>. Acesso em: 29 out. 2008.

KHAYAM, Syed Ali. **The Discrete Cosine Transform (DCT):** Theory and Application. [Lecture Notes]. East Lansing, 2003. 32 pp. Disponível em: <http://www.egr.msu.edu/waves/people/Ali_files/DCT_TR802.pdf>. Acesso em: 12 jun. 2008.

MANZINI, Giovani. The Burrows-Wheeler Transform: Theory and Practice. In: KUTYŁOWSKI, Mirosław (Ed.); PACHOLSKI, Leszek (Ed.); WIERZBICKI, Tomasz (Ed.). **Mathematical Foundations of Computer Science 1999.** Berlin: Springer, 1999. v. 1672, pp. 34-47. Disponível em: <http://www.mfn.unipmn.it/~manzini/papers/mfcs99x.pdf>. Acesso em: 22 nov. 2008.

_____. An Analysis of the Burrows-Wheeler Transform. **Journal of the Association for Computing Machinery**, [S.l.], v. 48, n. 3, pp. 407-430, May 2001. Disponível em: <http://www.mfn.unipmn.it/~manzini/papers/bwjacm2.pdf>. Acesso em: 22 nov. 2008.

MINTCHEV, Martin; KINGMA, Jack; BOWES, Kenneth. Methods to Assess Gastric Electrical Activity. **Cyberzine on Electrogastrography,** Calgary, 1995. Disponível em: <http://enel.ucalgary.ca/People/Mintchev/methods.htm>. Acesso em: 12 jun. 2008.

OPPENHEIM, Alan V. (Ed.); WILLSKY, Allan S. (Ed.). **Signals and Systems.** Upper Saddle River, NJ, USA: Prentice-Hall, 1997. 796 pp.

PAINTER, Ted; SPANIAS, Andreas. Perceptual Coding of Digital Audio. **Proceedings of the IEEE**, [S.l. : s.n.], v. 88, n. 4, pp. 451-515, 2000. Disponível em: <http://ieeexplore.ieee.org/iel5/5/18261/00842996.pdf>. Acesso em: 22 jan. 2008.

PETITCOLAS, Fabien A. P. Introduction to Information Hiding. In: KATZENBEISSER, Stefan (Ed.); PETICOLAS, Fabien A. P. (Ed.). **Information Hiding Techniques for Steganography and Digital Watermarking.** Norwood, MA, USA: Artech House, 2000. pp. 1-14.

POHJALAINEN, Jouni. Auditory Perception. In: POHJALAINEN, Jouni. **Methods of Automatic Audio Content Classification.** 2007. 123 fls. Lic.Sc. Thesis. Department of Electrical and Communications Engineering, Helsinki University of Technology, Helsinki, Finland, 2007. Disponível em: <http://lib.tkk.fi/Lic/2007/urn010219.pdf>. Acesso em: 15 ago. 2008.

PROAKIS, John G. **Digital Communications.** New York, NY, USA: McGraw-Hill, 1995. 928 pp.

PROAKIS, John G.; MANOLAKIS, Dimitris G. **Digital Signal Processing:** Principles, Algorithms and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, 1996. 1016 pp.

RIVERA-COLON, Ramfis; LINDQUIST, Claude S. ; REDDY, Sridhar P. Adaptive Filter Design for Estimation and Detection of Biological Signals (An Application to Electroencephalography). In: 26th ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, v. 1, 1992, Pacific Grove, CA, USA. **Conference…** [S.l. : s.n.], 1992. pp. 178-181. Disponível em: <http://ieeexplore.ieee.org/iel2/445/6707/00269282.pdf>. Acesso em: 15 ago. 2008.

_____. Class 3 Adaptive Filters using Time Domain Smoothing. In: 27th ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, v. 2, 1993, Pacific Grove, CA, USA. **Conference**… [S.l. : s.n.], 1993. pp. 1538-1542. Disponível em: <http://ieeexplore.ieee.org/iel5/922/8008/00342355.pdf>. Acesso em: 15 ago. 2008.

_____. Frequency Domain Windowing Analysis for Class 3 Adaptive Filters. In: 15th ANNUAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 1993, San Diego, CA, USA. **Proceedings**… [S.l. : s.n.], 1993. pp. 420-421. Disponível em: <http://ieeexplore.ieee.org/iel5/8466/26669/00978616.pdf>. Acesso em: 7 nov. 2008.

SEDGHI, S.; KHADEMI, M.; CVEJIC, N. Channel Capacity Analysis of Spread Spectrum Audio Watermarking for Noisy Environments. In: CONFERENCE ON INTELLIGENT INFORMATION HIDING AND MULTIMEDIA SIGNAL PROCESSING, 2006, Pasadena, CA, USA. **Proceedings**… [S.l. : s.n.], 2006. pp. 33-36. Disponível em: <http://ieeexplore.ieee.org/iel5/4041645/4041646/04041660.pdf>. Acesso em: 9 set. 2008.

SHIMIZU, Shuichi. Performance Analysis of Information Hiding. In: SECURITY AND WATERMARKING OF MULTIMEDIA CONTENTS IV, v. 4675, 2002, San Jose, CA, USA. **Proceedings**... [S.l.]: SPIE, 2002. pp. 421-432. Disponível em: <http://www.trl.ibm.com/projects/RightsManagement/datahiding/paper/Shu_EI02Paper.pdf>. Acesso em: 22 nov. 2008.

SKLAR, Bernard. **Digital Communications: Fundamentals and Applications.** Upper Saddle River, NJ, USA: Prentice-Hall, 2001. 1079 pp.

ULLMANN, Ronald F. An Algorithm for the Fast Hartley Transform. Stanford Exploration Project, Report 38, pp. 325-350, 1984. Disponível em: <http://sepwww.stanford.edu/oldreports/oldreports/sep38/38_29.pdf>. Acesso em: 27 fev. 2008.

VANAN, Rahul; CREUSERE, Charles D. Evaluating low bitrate scalable audio quality using advanced version of PEAQ and energy equalization approach. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, v. 3, 2005 Philadelphia, PA, USA. **Proceedings**… [S.l. : s.n.], 2005. pp. 189-192. Disponível em: <http://ieeexplore.ieee.org/iel5/8466/26669/01415678.pdf>. Acesso em: 7 nov. 2008.

WELLS, Richard B. Applied Coding and Information Theory for Engineers. Upper Saddle River, NJ, USA: Prentice-Hall, 1999. 305 pp.

YAROSLAVSKY, L.; WANG, Ye. DFT, DCT, MDCT, DST and Signal Fourier Spectrum Analysis. In: 10th EUROPEAN SIGNAL PROCESSING CONFERENCE, 2000, Tampere, Finland. **Proceedings**… [S.l. : s.n.], 2000. pp. 1065-1068. Disponível em: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2000/sessions/WedPm/SS2/cr1207.pdf >. Acesso em: 12 jun. 2008.

Apêndice A – Códigos BCH

"Os códigos BCH abrangem uma grande classe de códigos cíclicos que incluem alfabetos binários e não-binários" (Proakis, 1995, p. 435). BCH são as iniciais dos sobrenomes dos inventores – Bose, Chadhuri e Hocquenghem – desses códigos. Os códigos BCH permitem uma "grande seleção de tamanhos de blocos, taxas de bits de código, tamanhos de alfabetos e capacidade de correção de erros" (Sklar, 2001, p. 370).

A construção desses códigos é feita considerando-se três parâmetros: o tamanho *n* da palavra-código, o tamanho *k* da palavra a ser codificada e o número *t* de bits que podem ser corrigidos dentro de cada palavra-código. As relações entre esses parâmetros podem ser vistas nas Equações (I.1), (I.2) e (I.3) (Proakis, 1995). É importante notar que $m \geq 3$ e que $k < n$ em qualquer situação. A distância de Hamming mínima do código é $d_{\min}$.

$$n = 2^m - 1 \tag{I.1}$$

$$n - k \leq mt \tag{I.2}$$

$$d_{\min} = 2t + 1 \tag{I.3}$$

Os polinômios geradores para códigos BCH devem ser primitivos. Os coeficientes desses polinômios são representados por números octais arranjados de tal maneira que, quando são convertidos em valores binários, o dígito mais à direita corresponde ao termo independente do polinômio (Sklar, 2001). A representação dos códigos BCH é feita pelo par ordenado (*n*, *k*). Um código BCH dito (15, 5), por exemplo, tem como polinômio gerador o número octal 2467, que na forma binária fica 10 100 110 111. Assim sendo, seu polinômio gerador na forma clássica de representação fica $g(p) = p^{10} + p^8 + p^5 + p^4 + p^2 + p + 1$ (Proakis, 1995).