JÚLIA JACOBSEN DORNELLES CHANQUINI

ADAPTAÇÃO DE CODIFICADOR DE ÁUDIO MPEG-4 DE ACORDO COM A NORMA DO SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL

Campinas

2012

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

JÚLIA JACOBSEN DORNELLES CHANQUINI

ADAPTAÇÃO DE CODIFICADOR DE ÁUDIO MPEG-4 DE ACORDO COM A NORMA DO SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestra em Engenharia Elétrica, na área de concentração: Telecomunicações e Telemática.

Orientador: Prof. Dr. Luís Geraldo Pedroso Meloni

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELA ALUNA JÚLIA JACOBSEN DORNELLES CHANQUINI
E ORIENTADO PELO PROF. DR. LUÍS GERALDO PEDROSO MELONI

Assinatura do Orientador

_____

Campinas

2012

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Chanquini, Júlia Jacobsen Dornelles
C363a Adaptação de codificador de áudio MPEG-4 de acordo com a norma do sistema brasileiro de televisão digital / Júlia Jacobsen Dornelles Chanquini. -- Campinas, SP: [s.n.], 2012.

Orientador: Luís Geraldo Pedroso Meloni.
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Televisão digital. 2. Psicocústica. 3. Processamento de sinais - Técnicas digitais. 4. Teoria da codificação. I. Meloni, Luís Geraldo Pedroso. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Modification of a MPEG-4 audio coder to conform to the Brazilian digital television system
Palavras-chave em Inglês: Digital television, Psychoacoustics, Signal processing - Digital techniques, Coding and Information Theory
Área de concentração: Telecomunicações e Telemática
Titulação: Mestra em Engenharia Elétrica
Banca examinadora: Silvio Ernesto Barbin, Yuzo Iano
Data da defesa: 27-07-2012
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Júlia Jacobsen Dornelles Chanquini

Data da Defesa: 27 de julho de 2012

Título da Tese: "Adaptação de Codificador de Áudio MPEG-4 de Acordo com a Norma do Sistema Brasileiro de Televisão Digital."

Prof. Dr. Luís Geraldo Pedroso Meloni (Presidente): _____

Prof. Dr. Silvio Ernesto Barbin: _____

Prof. Dr. Yuzo Iano: _____

Agradecimentos

Gostaria de agradecer às pessoas que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

À minha família, por todo apoio e incentivo sempre me ajudando a superar os desafios e garantindo que nada me faltasse.

Ao meu noivo, que me ajudou muito tanto na parte técnica quanto me apoiando para seguir em frente e não desistir, sempre estando ao meu lado.

Ao meu orientador, por toda paciência para resolver minhas dúvidas e toda a ajuda que foi fundamental para completar este trabalho com sucesso.

E a todos os meus amigos que de uma forma ou outra sempre me incentivaram.

Resumo

Este trabalho apresenta a adequação de um codificador de áudio padrão MPEG-4 AAC para aderência à norma brasileira do SBTVD. Também apresenta um estudo dos conceitos envolvidos em codificadores de áudio perceptuais com enfoque no codificador MPEG-4 AAC e também inclui a parte de multiplexação e sincronia do MPEG-4.

Para o desenvolvimento do projeto foram estudados alguns códigos abertos de codificadores AAC: FAAD, 3GPP e o código de referência do padrão MPEG-4, especialmente a parte referente ao LATM/LOAS. O decodificador de áudio padrão MPEG-4 AAC que foi modificado para suportar a camada LATM/ LOAS foi o FAAD. Foi calculado o tempo adicional que o decodificador modificado leva para decodificar o áudio com a camada LATM/LOAS, sem ser notado um aumento significativo que não permite a decodificação em tempo real do áudio.

Palavras-chave: AAC. Codificação de áudio. SBTVD. TV Digital. MPEG-4.

Abstract

This work presents an adaptation of a standard MPEG-4 AAC audio coder to conform to the Brazilian digital TV standard SBTVD. It also presents a study of the concepts involved in perceptual audio coders focusing on MPEG-4 AAC and also including the multiplexing and synchronization part of the MPEG-4 standard.

To develop this project, open source AAC coders were studied: FAAD, 3GPP and the MPEG-4 reference software code specially the part concerning LATM/LOAS. The AAC audio decoder which was modified to support the LATM / LOAS layer was FAAD. The additional time that the modified decoder needs to decode a sample audio with LATM / LOAS was calculated, and it did not introduce a large enough delay that would restrict real time audio decoding.

Keywords: AAC. Audio coding. SBTVD. Digital TV. MPEG-4.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

3GPP	*Third Generation Partnership Project*
AAC	*Advanced Audio Coding*
ABNT	Associação Brasileira de Normas Técnicas
ADTS	*Audio Data Transport Stream*
ALAC	*Apple Lossless Audio Codec*
ALS	*Audio Lossless Coding*
AU	*Access Unit*
BCC	*Binaural Cue Coding*
CAT	*Conditional Access Table*
CELP	*Code-excited Linear Prediction*
CD	*Compact Disc*
DMIF	*Delivery Multimedia Integration Framework*
DTS	*Digital Theater Systems*
EP	*Error Protection*
ERB	*Equivalent Rectangular Bandwidth*
ES	*Elementary Stream*
FAAC	*Freeware Advanced Audio Coder*
FAAD	*Freeware Advanced Audio Decoder*
FEC	*Forward Error Correction*
FFT	*Fast Fourier Transform*
FLAC	*Free Lossless Audio Codec*
HD	*High-definition*
HE-AAC	*High Efficiency Advanced Audio Coding*
HVXC	*Harmonic Vector Excitation Coding*
IC	*Inter-channel Coherence*
IDD	*Inter-channel Phase Difference*
IEC	*International Electrotechnical Commission*
IID	*Interaural Intensity Difference*

ILD	*Interaural Level Difference*
IP	*Internet Protocol*
IS	*Intensity Stereo*
ISDB-T	*Integrated Services Digital Broadcasting-Terrestrial*
ISO	*International Organization for Standardization*
ITD	*Interaural Time Difference*
ITU	*International Telecommunication Union*
LATM	*Low Overhead MPEG-4 Audio Transport Multiplex*
LC	*Low Complexity*
LFE	*Low Frequency Effects*
LOAS	*Low Overhead Audio Stream*
LPC	*Linear Predictive Coding*
M/S	*Mid/Side*
MDCT	*Modified Discrete Cosine Transform*
MP3	*MPEG Audio Layer III*
MP4	*MPEG-4*
MP4FF	*MPEG-4 File Format*
MPEG	*Moving Picture Experts Group*
NIT	*Network Information Table*
OFDM	*Orthogonal Frequency-Division Multiplexing*
OPD	*Overall Phase Difference*
PAT	*Program Association Table*
PCM	*Pulse-code Modulation*
PCR	*Program Clock Reference*
PE	*Perceptual Entropy*
PES	*Packetized Elementary Stream*
PID	*Packet Identifier*
PMT	*Program Map Table*
PNS	*Perceptual Noise Substitution*
PS	*Parametric Stereo*
PS	*Program Stream*
PSI	*Program Specific Information*

QMF	*Quadrature Mirror Filter*
SBR	*Spectral Band Replication*
SBTVD	Sistema Brasileiro de Televisão Digital
SMR	*Signal to Mask Ratio*
SPL	*Sound Pressure Level*
STC	*System Time Clock*
TDAC	*Time-Domain Aliasing Cancellation*
TNS	*Temporal Noise Shaping*
TS	*Transport Stream*

1 INTRODUÇÃO

Um sinal para ser transmitido necessita se adequar a restrições de banda e complexidade do sistema de transmissão/recepção. A codificação de sinais visa eliminar redundâncias estatísticas e perceptuais (no caso de codificadores com perdas) para ser possível transmitir ou armazenar o sinal usando a menor quantidade de informação que ainda permita ao sinal ser reconstruído no receptor.

As técnicas de codificação de áudio podem ser divididas em com perdas e sem perdas. Nas técnicas sem perdas o sinal reconstruído deve ser idêntico ao que foi codificado, normalmente são usadas apenas redundâncias estatísticas nestes casos. Existem alguns codificadores sem perdas desenvolvidos especificamente para áudio que usam predição linear para estimar o espectro do sinal.

Nas técnicas de codificação para áudio com perdas o sinal reconstruído no receptor não é igual ao original transmitido, ele apenas precisa ser próximo o suficiente para que as diferenças não sejam percebidas ou não sejam incômodas aos ouvintes. Estes codificadores são também conhecidos como perceptuais.

A codificação perceptual de áudio visa reduzir o número de bits necessários para codificar um sinal de áudio eliminando partes do sinal que não serão percebidas pela audição humana. Um sinal de áudio que é perceptualmente igual ao original é denominado transparente. Para saber quais partes do sinal não serão percebidas existem os modelos psicoacústicos, que são modelos matemáticos baseados no estudo do sistema auditivo humano.

Para avaliar se um codificador é transparente normalmente é usada como referência a qualidade do áudio digitalizado no formato de CD, onde o áudio é amostrado em 44,1kHz e com 16 bits por amostra.

O Sistema Brasileiro de Televisão digital (SBTVD) usa o codificador de áudio perceptual padrão MPEG-4 AAC [25]. Este padrão suporta codificar áudio com vários canais e atinge níveis de compressão altos praticamente sem perda de qualidade.

O uso de múltiplos canais melhora a qualidade das transmissões de áudio da televisão digital, pois traz uma sensação melhor de envolvimento no ambiente apresentado no programa, como já é feito em cinemas, para poder oferecer maior qualidade e realismo à parte de áudio das transmissões.

O SBTVD também faz uso de ferramentas existentes no padrão MPEG para transporte e sincronização do áudio para transmissão em pacotes, que é a camada conhecida como LATM/LOAS definida na parte de áudio do padrão MPEG-4 [25].

Para a sincronização dos dados do áudio com vídeo e outros dados necessários para transmissão, é usado o padrão MPEG [22].

O SBTVD foi desenvolvido com base no sistema japonês ISDB-T (*Integrated Services Digital Broadcasting-Terrestrial*), por isso também é conhecido como ISDB-Tb. As principais mudanças feitas para o padrão brasileiro em relação ao japonês foram a criação de um *middleware* próprio e o uso do padrão MPEG-4 ao invés do MPEG-2 para codificação de áudio e vídeo.

Nos capítulos a seguir serão apresentados conceitos de codificação de áudio com enfoque na parte perceptual, com uma introdução do funcionamento básico do sistema auditivo e em seguida os conceitos principais aplicados nos modelos psicoacústicos.

O objetivo desta dissertação é o estudo do codificador padrão MPEG-4 AAC que é o usado pelo Sistema Brasileiro de TV Digital e como que é feito para codificar e transportar o áudio, mantendo a compatibilidade com o padrão brasileiro. Para o desenvolvimento desta dissertação foi feita uma revisão extensa do padrão AAC/HE-AAC e a camada de multiplexação e transporte LATM/LOAS. Foram também realizadas modificações em um codificador AAC de código aberto para se adequar ao SBTVD através da inclusão de suporte à camada LATM/LOAS.

2 TÉCNICAS DE CODIFICAÇÃO DE ÁUDIO

O objetivo da codificação de áudio é reduzir o número de bits necessários para representação do sinal para com isso ser possível reduzir custos com armazenamento e/ou transmissão destes dados.

Existem dois tipos básicos de técnicas de codificação para qualquer tipo de dados: sem perdas e com perdas. Na codificação sem perdas, o sinal deve ser codificado de forma que o decodificador correspondente seja capaz de reconstruir exatamente o mesmo sinal que entrou no codificador, este esquema é também chamado de codificador de entropia. Na codificação com perdas, é permitido eliminar informação do sinal que não serão percebidas pelo ouvinte, de modo que o sinal decodificado será uma aproximação do original.

Os codificadores de áudio também são classificados de acordo com as técnicas de análise e síntese usadas para codificar os sinais de áudio em codificadores no domínio do tempo e no domínio da frequência.

2.1 CODIFICAÇÃO SEM PERDAS

Nas técnicas de codificação sem perdas o sinal decodificado deve ser idêntico ao que foi codificado. Nessas técnicas o objetivo é comprimir utilizando apenas redundâncias estatísticas. Os algoritmos de codificação sem perdas mais usados em codificadores de áudio são Huffman, Lempel-Ziv e aritméticos.

Existem também diversos codificadores sem perdas específicos para áudio como *Monkey's Audio*, *shorten*, FLAC, ALAC (Apple) e também uma extensão do MPEG-4, o *Audio Lossless Coding*, também chamado de MPEG-4 ALS. Os codificadores *Dolby* e DTS também têm suas versões sem perdas para fontes multicanal, que são os codificadores *Dolby TrueHD* e *DTS-HD Master Audio*. Esses codificadores são usados para áudio de alta qualidade gravado em mídia *Blu-Ray*.

A maioria dos algoritmos de compressão sem perdas para áudio usa algum tipo de predição linear para estimar o espectro do sinal e remover redundâncias. Nestes algoritmos aplica-se um preditor linear às amostras do sinal em cada bloco, o que resulta em uma sequência

de amostras de erro de predição. Os parâmetros do preditor, com isso, representam a redundância que é removida do sinal e os parâmetros do preditor codificados sem perdas e o erro de predição juntos representam o sinal em cada bloco.

Um método menos comum também usado para codificadores sem perdas de áudio é um onde se obtém uma representação quantizada de taxa de bits baixa ou uma representação com perdas do sinal, e então a diferença entre a versão com perdas e o sinal original é comprimida com um método sem perdas [18].

Esses codificadores conseguem reduzir o tamanho para metade do sinal sem compressão, dependendo normalmente do sinal a ser comprimido. Eles não alteram a forma de onda do sinal original. São normalmente usados para arquivamento ou para quando o áudio vai passar por várias etapas de edição para evitar distorções em que várias etapas de compressão com perdas provocam.

2.2 CODIFICAÇÃO COM PERDAS

Na codificação com perdas partes do sinal são descartadas, devido à necessidade no sistema de se reduzir a taxa de transmissão ou restrições de capacidade de processamento.

Em codificação de áudio existe a codificação com perdas chamada de codificação perceptual, que usando modelos do sistema auditivo humano, calcula que partes da informação poderão ser descartadas sem serem percebidas pela maioria dos ouvintes. Esses codificadores podem conseguir taxas de compressão de até 1:32.

Um sinal de áudio que é perceptualmente igual ao original é chamado de transparente, isto é, quando por meio do uso de técnicas estatísticas, a maioria dos ouvintes não consegue distinguir o sinal original e o reconstruído no decodificador. Isto ocorre devido a limitações do sistema auditivo humano, que serão estudadas nos capítulos seguintes.

Para avaliar se um codificador é transparente normalmente é usada como referência a qualidade do áudio digitalizado no formato de CD, onde o áudio é amostrado em 44,1kHz e com 16 bits por amostra.

2.3 CODIFICADORES DE ÁUDIO NO DOMÍNIO DO TEMPO

Os codificadores que trabalham no domínio do tempo usam as amostras temporais do sinal para análise. O mais conhecido é PCM (*Pulse Code Modulation*) que é um método para representar amostras de um sinal de áudio analógico de forma digital, onde a amplitude do sinal analógico é amostrada regulamente em intervalos uniformes com cada amostra sendo quantizada para o valor mais próximo dentro de uma faixa de valores digitais. Esse método não realiza compressão do áudio, pois não tem nenhum mecanismo para remover redundâncias, ele apenas faz a quantização das amplitudes da forma de onda com um tamanho de passo fixo. Para um áudio amostrado com uma taxa de 44,1kHz (44100 amostras por segundo) com resolução de 16 bits por amostra (qualidade de CD), multiplicando-se a taxa de amostragem pelo número de bits por amostra isto resulta em uma taxa de 705 Kb/s por canal, isto é, 1,41Mb/s para um par estéreo e 4,23 Mb/s para um sistema com seis canais.

Existem também técnicas de codificação no domínio do tempo que exploram a correlação entre os sinais. Uma destas técnicas é a DPCM (*Differential Pulse Code Modulation*), em um codificador DPCM é quantizada a diferença entre o sinal previsto e o original ao invés de quantizar as amostras temporais diretamente. Esta técnica de codificação assume que os sinais de áudio serão correlacionados o suficiente para que a variância da diferença entre o sinal previsto e o original seja menor que a variância do sinal original. Existem alguns padrões da ITU de codificadores de voz (G.721, G.723, G.726 e G.727) que são baseados em uma versão adaptativa (ADPCM) do DPCM, que usa predição e um passo de quantização variável para reduzir um pouco mais a taxa de bits necessária.

Os codificadores no domínio do tempo possuem algoritmos com menor complexidade, o que possibilita que eles sejam implementados em sistemas com menor poder de processamento, mas costumam precisar de taxas maiores de bits para manter uma qualidade maior se comparados com os codificadores no domínio da frequência.

2.4 CODIFICADORES DE ÁUDIO NO DOMÍNIO DA FREQUÊNCIA

Os codificadores no domínio da frequência apresentam uma melhor qualidade se comparados com os no domínio no tempo, mas costumam ter uma complexidade computacional maior.

Os codificadores de áudio perceptuais usualmente analisam o sinal no domínio da frequência. Os métodos no domínio da frequência podem ser subdivididos em codificadores de sub-bandas e de transformadas.

Na análise por sub-bandas o espectro do sinal é dividido em sub-bandas usando bancos de filtros passa faixa para decompor o sinal em bandas estreitas. Este tipo de codificação imita o mecanismo de análise em frequência do ouvido. Os sinais nessas bandas podem então ser quantizados independentemente com os erros de quantização de cada banda ficando dentro dos limites de frequência da banda correspondente, podendo ser arranjados de maneira a gerar um ruído abaixo do nível de mascaramento calculado pelo modelo psicoacústico.

Na codificação usando transformada, a forma de onda do sinal é convertida para uma representação no para o domínio da frequência através de técnicas como a Transformada Rápida de Fourier (FFT) e Transformada de Cosseno Discreta Modificada (MDCT). A codificação por transformada é baseada no fato de que a compactação do sinal pode ser conseguida graças à representação eficiente dos coeficientes do sinal no domínio transformado, onde a transformada MDCT pode ser considerada uma aproximação da transformada ótima de Karhuen-Loève [28]. O problema com essa abordagem é que ela tende a ter falhas na presença de transientes do sinal, que fazem com que seja preciso atualizar frequentemente os coeficientes, já que estes vão mudar rapidamente nessas condições [7], exigindo o processamento com blocos de duração variáveis conforme as características do sinal.

A MDCT é uma das transformadas mais usadas em codificação de áudio, ela possui apenas coeficientes reais ao contrário da FFT. Os codificadores MPEG-1 camada três (MP3) e MPEG-2/4 AAC usam a MDCT.

3 SISTEMA AUDITIVO HUMANO

O estudo do sistema auditivo humano permite modelar suas limitações, possibilitando uma maior compressão do áudio eliminando informações do sinal de áudio que normalmente não são percebidos pelo sistema auditivo humano. A Figura 3.1 ilustra a estrutura do ouvido humano, que é divido em orelha externa, orelha média e orelha interna.

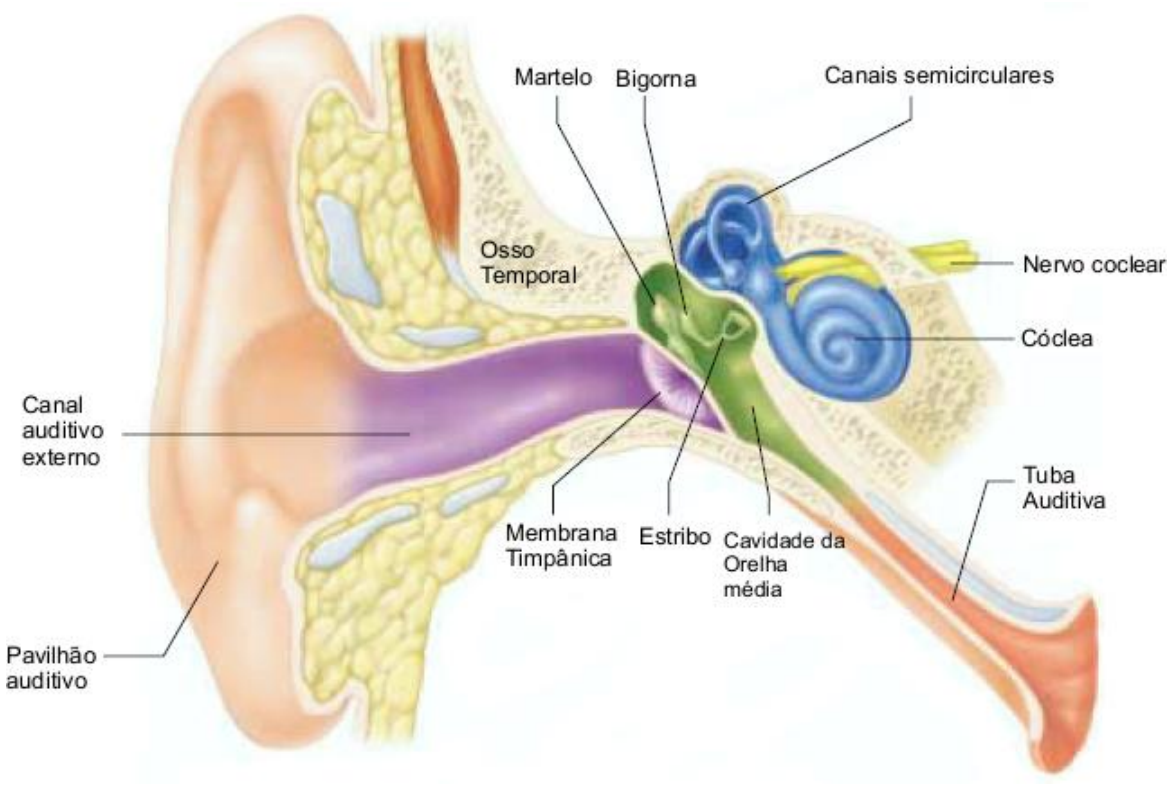


Figura 3.1 – Estrutura do Ouvido Humano (Traduzido de [46]).

3.1 ORELHA EXTERNA

A orelha externa abrange o pavilhão auditivo e o canal auditivo externo ou meato auditivo. A função do pavilhão auditivo é ajudar na localização do som. O pavilhão auditivo coleta as ondas sonoras e direciona para o canal auditivo. Quando as ondas sonoras atingem as

dobras de cartilagem existentes na orelha, elas são atenuadas e refletidas, e essas alterações são usadas pelo cérebro para identificar a direção de onde essas ondas vieram.

O canal auditivo externo estabelece a comunicação entre a orelha média e o meio externo, tem cerca de três centímetros de comprimento e está escavado no osso temporal. O canal auditivo externo é um tubo com um dos lados tampado, o que o faz ter ressonância na faixa de 3 a 5kHz, e com isso melhora a sensibilidade aos sinais de fala. No fim deste canal há uma delicada membrana, o tímpano ou membrana timpânica, que é o começo da orelha média.

3.2 ORELHA MÉDIA

A orelha média começa na membrana timpânica e é composta por esta e por três ossículos (martelo, bigorna e estribo) que são articulados entre si. O cabo do martelo está encostado no tímpano; o estribo apoia-se na janela oval, um dos orifícios dotados de membrana da orelha interna que estabelecem comunicação com a orelha média. O outro orifício é a janela redonda, como pode ser visto na Figura 3.2.

A orelha média comunica-se também com a faringe, através de um canal denominado tuba auditiva. Esse canal permite que o ar penetre no ouvido médio, para que, de um lado e de outro do tímpano, a pressão do ar atmosférico seja igual. Quando essas pressões ficam diferentes, não ouvimos bem, até que o equilíbrio seja restabelecido. Os ossículos fazem a transmissão do som que é recebido do tímpano diretamente à janela oval. A janela redonda permite o movimento do fluído presente no interior da cóclea.

A principal função da orelha média é melhorar a transmissão do som entre o ouvido externo e o interno, reduzindo a reflexão que ocorre quando uma onda sonora incide em uma superfície fluida. Os ossículos e a membrana timpânica podem ser considerados um transformador de impedância que reduz a alta impedância do fluído coclear para uma impedância semelhante a do ar [7].

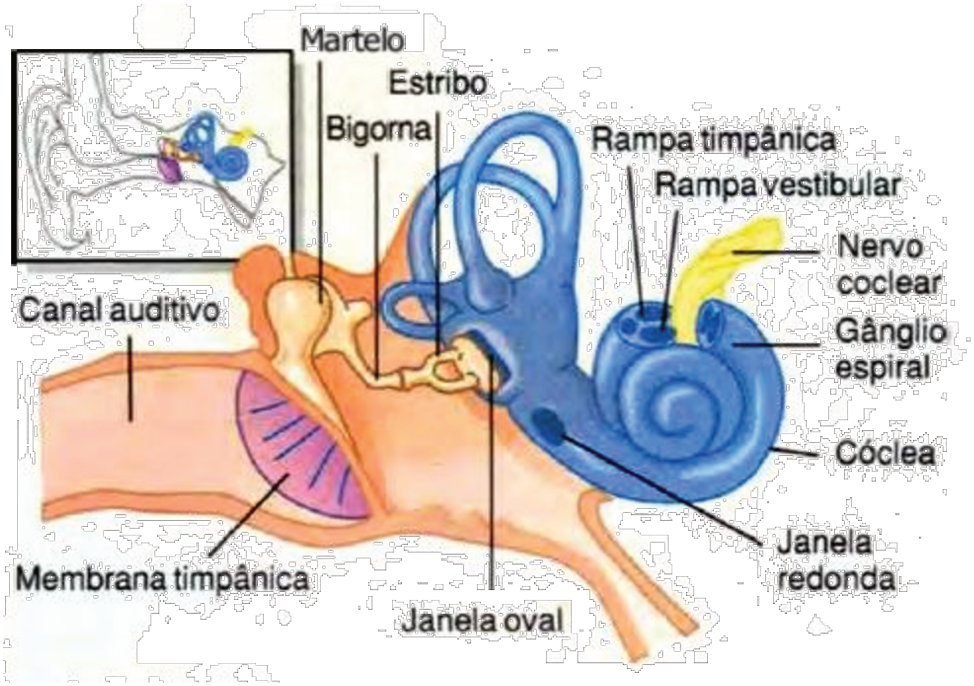


Figura 3.2 – Membrana timpânica, sistema ossicular da orelha média e orelha interna (Reproduzido de [17]).

O tímpano faz a conversão da energia sonora em energia mecânica. Como o líquido é mais difícil de mover que o ar, a pressão sonora transmitida para a orelha interna deve ser amplificada. Os ossículos vibram juntamente com o tímpano para amplificar o som e carregá-lo para o ouvido interno através da janela oval, que é uma abertura coberta por uma membrana que separa a orelha média e interna. O diâmetro da janela oval é de 15 a 30 vezes menor do que o do tímpano, o que amplifica a pressão transmitida para o ouvido interno.

O processo de transformação do sinal acústico nas ondas do líquido coclear conduz a uma função de transferência do ouvido médio. Este processo é equivalente a uma filtragem passa-baixas com corte em 5 kHz, com uma sobre-elevação na faixa entre 2 e 5 kHz e um pico em torno de 3,5 kHz. Como essa filtragem não altera o espectro de forma significativa, ela é, em geral, desconsiderada para sinais com faixa até 5 kHz [7].

Na orelha média o som pode ser atenuado pela atuação dos músculos tensor do tímpano e estapédio (músculo do estribo). Quando sons muito intensos são transmitidos através do sistema ossicular e depois para o sistema nervoso central, ocorre um reflexo depois de um período de latência de apenas 40 a 80 ms, causando contração do músculo estapédio e, em menor grau, do músculo tensor do tímpano. O músculo tensor do tímpano puxa o cabo do martelo para dentro,

enquanto o músculo estapédio puxa o estribo para fora. Estas duas forças se opõem entre si e assim fazem com que o sistema ossicular inteiro desenvolva um aumento de rigidez, reduzindo grandemente a condução ossicular de sons com baixa frequência, principalmente frequências abaixo de 1 kHz.

Este reflexo de atenuação pode reduzir a intensidade de transmissão de sons com frequências baixas em 30 a 40 dB, o que é aproximadamente a mesma diferença entre uma voz intensa e um sussurro. Acredita-se que este mecanismo tenha duas funções:

Proteger a cóclea de vibrações prejudiciais causadas por som excessivamente intenso e mascarar sons com baixa frequência em ambientes com som intenso, para remover uma grande parte do ruído de fundo e permitir que uma pessoa se concentre em sons acima de 1 kHz, onde é transmitida a maior parte da informação de fala.

Outra função dos músculos tensor do tímpano e estapédio é diminuir a sensibilidade auditiva da pessoa à sua própria fala. Este efeito é ativado por sinais nervosos colaterais transmitidos a estes músculos ao mesmo tempo em que o cérebro ativa o mecanismo de produção da voz [17].

3.3 ORELHA INTERNA

A orelha interna, também conhecida como labirinto, é formada por escavações no osso temporal, revestidas por membrana e preenchidas por líquido. Limita-se com a orelha média pelas janelas oval e a redonda. O labirinto apresenta duas partes: uma parte anterior, conhecida como cóclea ou caracol, que é relacionada à audição, e uma parte posterior, relacionada com o equilíbrio, que é constituída pelo vestíbulo e pelos canais semicirculares.

A cóclea é uma cavidade cônica enrolada na forma de um caracol preenchida por um meio aquoso. Nela existem três tubos espiralados lado a lado: a rampa vestibular, a rampa média e a rampa timpânica. A rampa vestibular e a rampa média são separadas entre si pela membrana de Reissner (também chamada de membrana vestibular), a rampa timpânica e a rampa média são separadas entre si pela membrana basilar. Na superfície da membrana basilar, está localizado o órgão de Corti, que contém uma série de células eletromecanicamente sensíveis, as células

ciliadas que funcionam como sensores de vibração ligados a terminações nervosas. Pode-se observar um esquema da orelha média e interna na Figura 3.2.

A cóclea pode ser modelada como um tubo cônico de aproximadamente 30 mm com duas câmaras separadas pela membrana basilar, como pode ser visto na Figura 3.3. Na extremidade oposta à janela oval, existe um orifício sobre a membrana basilar que comunica essas duas câmaras, chamado de helicotrema.

A membrana basilar apresenta uma resistência (mecânica) que varia ao longo de sua extensão: próximo à janela oval ela é mais fina e tensa, ressoando em frequências mais altas, enquanto no seu final, ela é espessa e flácida, ressoando então com frequências mais baixas. As ondas geradas pelo estribo, em resposta a um sinal senoidal, viajam ao longo da cóclea, fazendo com que a membrana basilar vibre de forma mais intensa em uma posição específica da membrana basilar conforme a frequência do sinal de entrada [7].

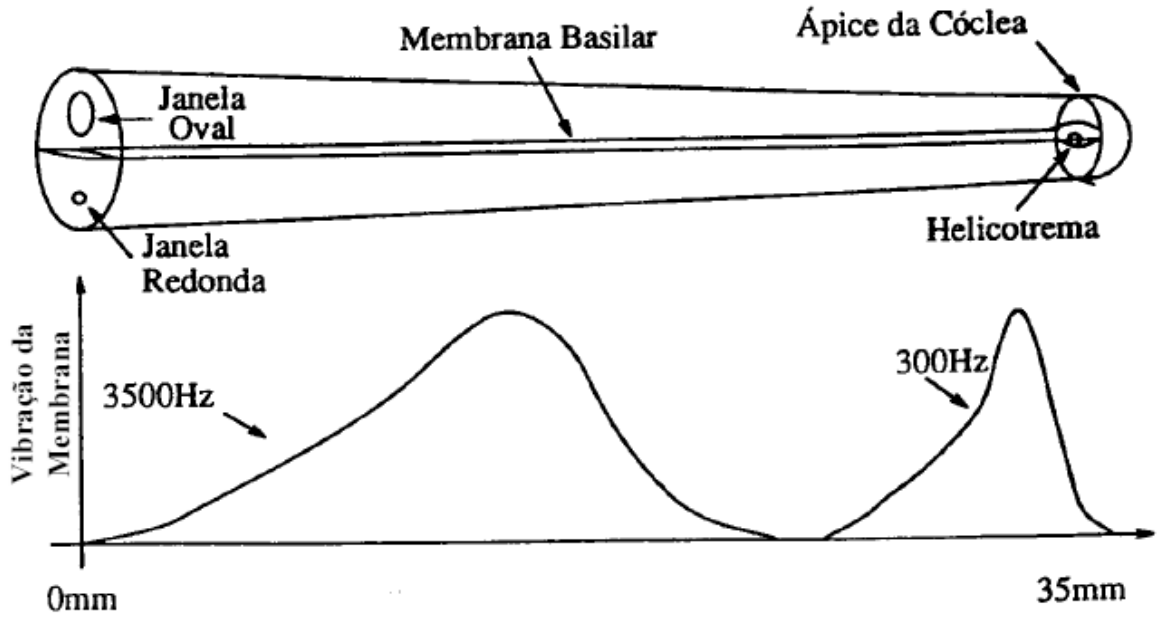


Figura 3.3 – Cóclea, membrana basilar e vibração da membrana em duas frequências diferentes (Reproduzido de [7]).

Sobre a membrana basilar existem ainda duas estruturas: as fibras basilares e o órgão de Corti. A membrana basilar contém cerca de 20.000 a 30.000 fibras basilares. Estas fibras são pequenas estruturas rígidas, elásticas e em forma de palheta, que são livres numa extremidade, podendo vibrar como as palhetas de uma gaita.

Os comprimentos das fibras basilares variam ao longo da membrana, sendo mais curtas perto da janela oval e mais longas no ápice da cóclea. As fibras curtas e rígidas perto da janela oval da cóclea vibram melhor em frequências altas, enquanto as fibras longas e flexíveis perto da extremidade da cóclea vibram melhor em frequências baixas. Com isto, a ressonância de alta frequência da membrana basilar ocorre perto da base, onde as ondas sonoras entram na cóclea através da janela oval. Já a ressonância de baixa frequência ocorre perto do helicotrema, principalmente devido às fibras menos rígidas, mas também devido ao aumento de "carga" com massas extras de líquido que precisam vibrar ao longo dos túbulos cocleares [17].

A vibração das fibras basilares estimula as células ciliadas que compõe o órgão de Corti, que é responsável pelo sensoriamento dos estímulos sonoros recebidos pela orelha, ele faz a conversão da energia mecânica recebida em impulsos elétricos. Quando a membrana basilar se move, ela excita as células ciliadas, que transformam o movimento das fibras basilares em impulsos nervosos, que são então transmitidos pelo nervo coclear para a região específica do córtex cerebral. A Figura 3.4 mostra a anatomia do órgão de Corti.

Na cóclea, então é onde ocorre conversão das vibrações mecânicas em impulsos elétricos. A membrana basilar vibra com a onda sonora e funciona como um analisador de espectro, separando o som em componentes de frequência, com uma associação posição-frequência. Cada ponto da membrana basilar é mais sensível a uma determinada frequência, chamada de frequência característica. Para um ponto específico da membrana basilar, a curva de resposta à frequência de vibração presente na janela oval é equivalente à de um filtro passa-faixa com fator de qualidade aproximadamente constante, resultando numa melhor resolução em frequências baixas. Assim, as fibras basilares localizadas na região de altas frequências características respondem em uma faixa de frequências maior do que as fibras na região de baixas frequências características.

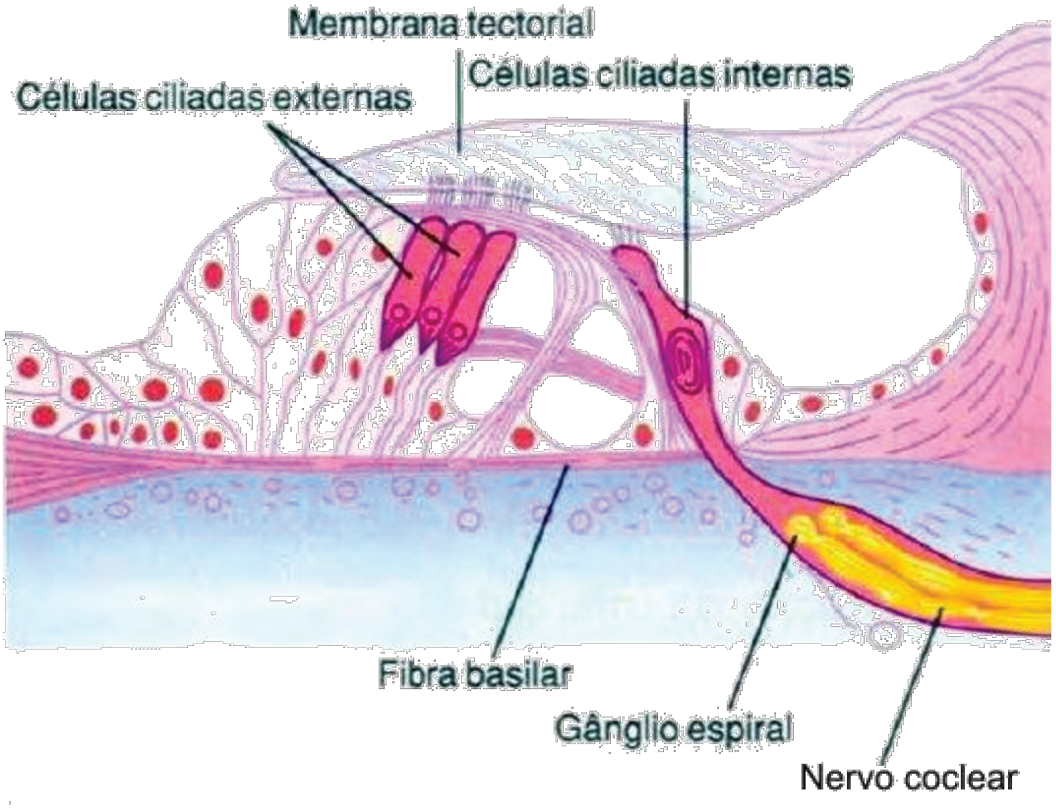


Figura 3.4 – Órgão de Corti (Reproduzido de [17]).

Tal comportamento pode ser observado em traçado da curva de resposta ao longo da membrana basilar para um tom em uma frequência específica, como pode ser observado na Figura 3.3. Para cada frequência, existe um ponto da membrana basilar em que a vibração será máxima. A posição deste ponto, medida a partir do helicotrema, é aproximadamente proporcional ao logaritmo da frequência do som. Ao redor desse ponto haverá uma faixa, de cerda de 1,5 mm onde a vibração estará presente, sendo atenuada à medida que se afasta do ponto [7]. Esta faixa está relacionada ao conceito de bandas críticas que será apresentado no próximo capítulo.

O sistema auditivo humano consegue processar sons com frequências na faixa entre 20 Hz e 20 kHz, mas a resposta para altas frequências diminui com a idade.

4 CODIFICAÇÃO PERCEPTUAL DE ÁUDIO

Como já foi visto no capítulo 2, existem dois tipos básicos de técnicas de codificação de dados: sem perdas e com perdas. Em codificadores de áudio, o codificador com perdas remove partes que não serão percebidas pelo ouvinte, tal codificação com perdas em áudio também é conhecida como codificação perceptual de áudio. Os codificadores perceptuais usam modelos psicoacústicos para determinar quais partes do áudio não serão percebidas.

Alguns dos codificadores de áudio mais conhecidos como MP3, AAC e AC-3 (Dolby Digital) são perceptuais.

A maioria dos codificadores perceptuais de áudio é baseada na arquitetura genérica mostrada na Figura 4.1.

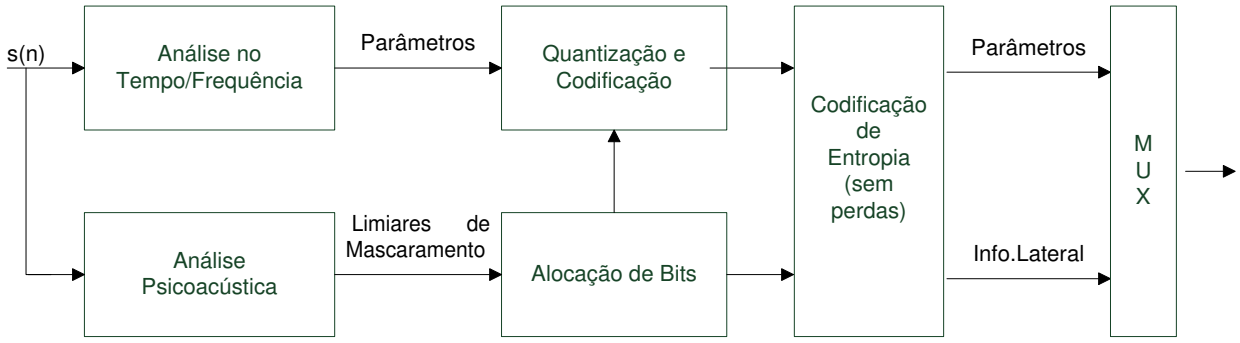


Figura 4.1 – Codificador de Áudio Perceptual Genérico (adaptado de [1]).

Os codificadores perceptuais normalmente dividem o sinal em quadros com durações entre 2 e 50 ms. Em seguida a parte de análise no tempo-frequência estima as componentes temporais e espectrais de cada quadro. É comum que o mapeamento no domínio tempo-frequência seja associado com as propriedades de análise do sistema auditivo humano, apesar de não ser sempre o caso.

O objetivo final dessa parte do codificador é extrair do sinal de áudio de entrada um conjunto de parâmetros no domínio do tempo e frequência que sejam possíveis de ser quantizados e codificados de acordo com uma métrica de distorção perceptual [1].

O controle desta distorção é obtido pela parte de análise psicoacústica do sinal que usa um modelo psicoacústico para estimar o poder de mascaramento do sinal.

4.1 MODELO PSICOACÚSTICO

O modelo psicoacústico é um modelo matemático de como o sistema auditivo humano processa subjetivamente o som. Ele torna possível a compressão com perdas de um sinal mantendo uma alta qualidade perceptual através da descrição de que partes de um dado sinal digital podem ser removidas ou comprimidas com mais intensidade sem perdas significativas na qualidade percebida do som.

A partir desse modelo são calculados limiares de mascaramento, que representam valores limites de energia abaixo do qual um tom ou ruído não será percebido ao ser mascarado por outro som.

Estes limiares podem ser usados para definir limites de ruído de quantização, o ruído deve ficar abaixo do limiar para ser mascarado. Assim, os limiares quantificam a quantidade máxima de distorção em cada ponto no domínio do tempo e frequência onde a quantização dos parâmetros de tempo/frequência não introduzirá artefatos audíveis.

O modelo envolve diversos conceitos que são baseados no estudo do sistema auditivo humano como: limiar absoluto de audibilidade, bandas críticas, mascaramento e seu espalhamento. A combinação destes princípios com propriedades de quantização de sinais também deu origem à teoria da entropia perceptual, proposta por Johnston [29], que é uma estimativa do limite fundamental da compressão transparente de sinais de áudio.

4.1.1 Percepção de volume

A percepção do volume pelo ouvido humano não é linear. O ser humano é mais sensível a variações de pressão da onda sonora para as baixas pressões do que para as altas, por isso as ondas sonoras são normalmente caracterizadas em nível logarítmico.

A unidade mais usada para a o nível de pressão sonora é a SPL (*Sound Pressure Level* - Nível de Pressão Sonora), que expressa em decibéis (dB) SPL o nível de pressão sonora em escala logarítmica em relação a um nível de referência padrão. É dado por:

$$L_{SPL} = 20log_{10}(p/p_0)\ \textbf{(dB SPL)}, \qquad \textbf{(4.1)}$$

onde L_{SPL} é o nível de pressão sonora de um estímulo, p é a pressão sonora do estímulo em Pascal (Pa), e p_0 é o nível de referência padrão de 20μPa que corresponde ao limiar de audibilidade para um tom de 1kHz [1].

Estímulos com níveis em torno de 130 dB SPL ou mais costumam determinar o limiar da dor. Na Tabela 4.1, pode-se observar o nível de pressão sonoro, em dB SPL, para alguns exemplos do cotidiano.

Situação	***Nível de Pressão Sonora (dB SPL)***
Limiar de Audibilidade	0
Estúdio de Gravação	20
Murmúrio	30
Conversação Normal	60
Restaurante Movimentado	70
Trânsito Pesado	80
Britadeira	120
Limiar da dor	130
Motor de Jato	150

Tabela 4.1 - Nível de pressão sonora para exemplos do cotidiano [33].

4.1.2 Limiar Absoluto de Audibilidade em Silêncio

É a quantidade de energia necessária para um ouvinte conseguir detectar um tom puro de certa frequência em um ambiente de silêncio absoluto. É tipicamente expressado em dB SPL, e é aproximado pela equação (**4.2**) [1]:

$$lim = 3,64f^{-0,8} - 6,5e^{-0,6(f-3,3)^2} + 10^{-3}f^4 \quad \textbf{(dB SPL)} \qquad \textbf{(4.2)}$$

onde f é a frequência em kHz.

O limiar varia de acordo com a frequência do tom e é apresentado de forma gráfica na Figura 4.2. Ele representa um ouvinte jovem e com audição apurada.

Esta aproximação é usada em quase todos os métodos perceptuais. Consiste de três termos: o primeiro descreve a frequência de corte para as baixas frequências, o segundo descreve

o aumento de sensibilidade do ouvido para a faixa de frequências em torno de 3 kHz e o último descreve a frequência de corte para altas frequências.

O primeiro termo, ou pelo menos parte dele, é interpretado como um resultado do ruído interno (causado por atividade muscular, fluxo de sangue etc.), ao passo que os dois últimos termos são interpretados como a característica de transferência de ouvido médio para o interno.

Consequentemente, em modelos perceptuais, esta equação é frequentemente dividida em duas partes: uma chamada função de ruído interno e outra chamada função de transferência do ouvido médio.

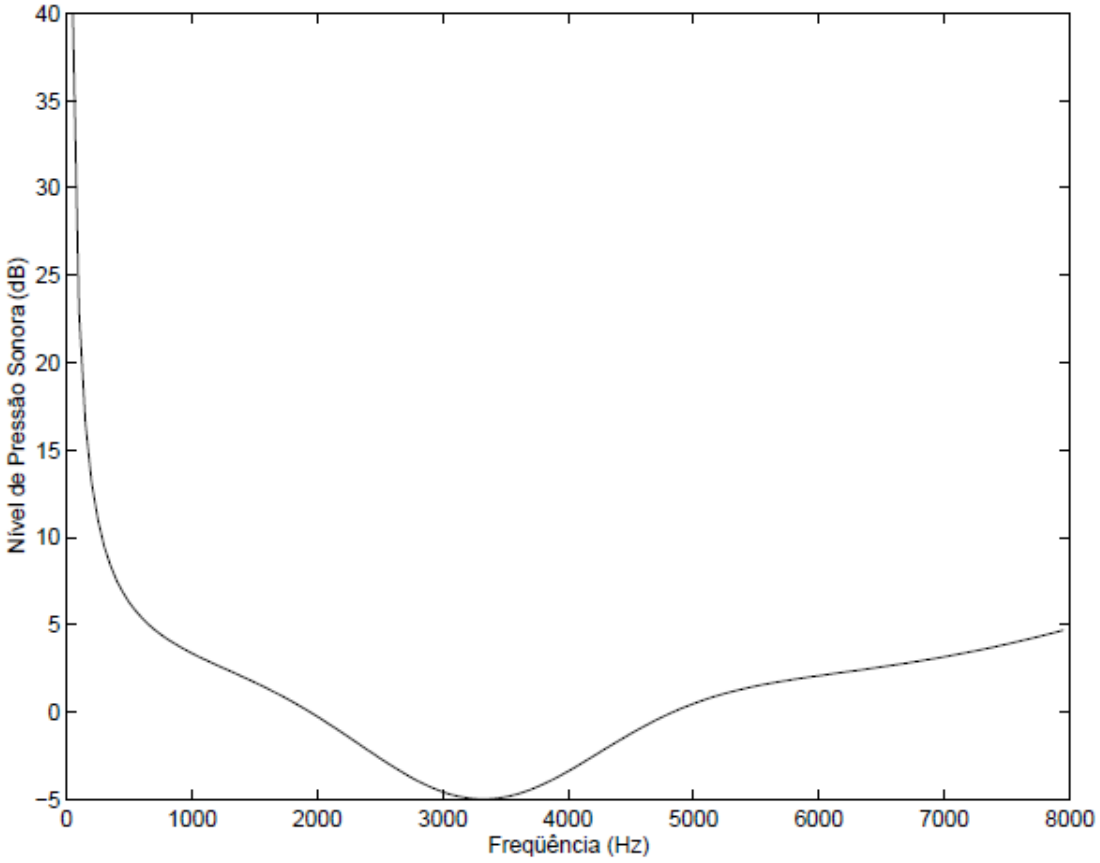


Figura 4.2 – Limiar absoluto de audibilidade [33].

Este limiar apresenta alguns problemas na prática. Um deles é que são considerados estímulos puramente tonais, mas o ruído de quantização nos codificadores perceptuais tende a ser espectralmente complexo ao invés de ser tonal. Além disso, quando é desenvolvido um algoritmo, não é conhecido qual será o nível SPL no qual o som será reproduzido no

decodificador [1]. Por isso normalmente ele é usado em conjunto com outros conceitos do sistema auditivo para formar o modelo psicoacústico a ser aplicado no codificador.

4.1.3 Percepção de frequência

A percepção de frequência no sistema auditivo humano, assim como a de volume, não é linear. Variações em baixas frequências são percebidas melhor do que em altas frequências.

Essa não linearidade ocorre devido à estrutura física da membrana basilar. A variação da largura e da rigidez, em função da distância da base, são os principais fatores que explicam essa não linearidade. A maior parte da membrana responde a sons com frequência inferior a 3 kHz, que é a faixa onde está localizada a maior quantidade de informação necessária para o entendimento da fala [33].

4.1.4 Bandas Críticas

O conceito de bandas críticas é baseado na forma como o ouvido realiza a análise espectral do som. Uma banda crítica define uma faixa em torno de uma frequência central, a qual está associada a um ponto da membrana basilar, que é a responsável pela análise em frequência do som no sistema auditivo humano como visto anteriormente, de modo que a cada ponto é possível definir uma banda crítica, da forma descrita a seguir.

As ondas que se propagam na orelha interna geram respostas de pico em posições específicas da membrana basilar e, portanto, diferentes receptores neurais são efetivamente "sintonizados" para faixas de frequência diferentes de acordo com suas localizações. Para estímulos senoidais, a onda que viaja na membrana basilar se propaga a partir da janela oval até que se aproxima da região com uma frequência de ressonância próxima àquela da frequência do estímulo, onde a sua magnitude aumenta para um valor de pico. A localização deste pico é chamada de “lugar característico” para a frequência do estímulo, e a frequência que melhor excita um lugar específico é chamada de “frequência característica”. Assim, uma transformação de lugar para frequência ocorre como pode ser observado no esquema da Figura 4.3. A figura mostra

uma representação esquemática dos envelopes de vibração da membrana (deslocamento vertical em relação à mesma) que ocorrem em resposta a um tom acústico complexo que contém senóides de 400, 1600 e 6400 Hz. As respostas de pico para cada senóide são localizadas ao longo da superfície da membrana, com cada pico ocorrendo a uma distância específica da janela oval (a "entrada" coclear). Assim, cada componente do estímulo complexo provoca respostas fortes somente a partir dos receptores neurais associados com uma localização de frequência específica [50].

Como resultado da transformação de frequência para lugar, o sistema auditivo pode ser modelado sob uma perspectiva de processamento de sinais, como um banco de filtros passa-faixa, formado por filtros com grande sobreposição. As respostas de magnitude são assimétricas e não lineares (dependentes do nível). Além disso, as faixas de passagem do filtro coclear são de largura de banda não uniforme, e as larguras de banda aumentam conforme a frequência cresce [1].

A largura de banda crítica é uma função de frequência, que quantifica a faixas de passagem do filtro coclear. Trabalhos de medidas acústicas em grupos de ouvintes realizados por vários observadores levaram ao conceito de bandas críticas.

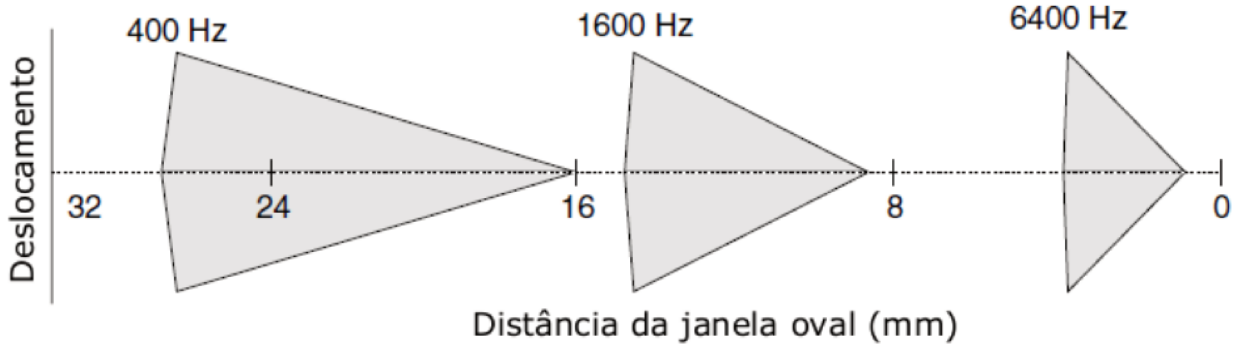


Figura 4.3 – A transformação de frequência para posição ao longo da membrana basilar [43].

Quando dois sinais se situam dentro de uma banda crítica, o de maior energia poderá dominar a percepção e mascarar o outro estímulo sonoro. Portanto, dependendo dos níveis, dois tons distintos só serão distinguidos um do outro quando estiverem em bandas críticas diferentes. Este é o fenômeno responsável pelo mascaramento simultâneo.

As faixas de frequência das bandas foram determinadas através de experimentos psicoacústicos para uma média de um grande número de ouvintes. Uma aproximação da banda crítica é dada por [1]:

$$\mathbf{BW_c}(f) = \mathbf{25 + 75[1 + 1,4}(f/\mathbf{1000})^{\mathbf{2}}]^{\mathbf{0,69}}\ \mathbf{(Hz)} \quad (4.3)$$

Apesar das bandas críticas serem contínuas na frequência, para aplicações práticas é comum ser utilizado um conjunto discreto. O conjunto discreto mais utilizado, apresentado na Tabela 4.2 é denominado escala Bark. Para cobrir todo o espectro de som audível, que vai até 20 kHz, são usadas 25 bandas críticas. A Tabela 4.2 mostra um banco de filtros ideal, com 25 filtros que se sobrepõem com larguras de banda inferiores a 100 Hz para frequências audíveis mais baixas e até 5 kHz para as mais elevadas, e que corresponde aos pontos discretos marcados nas curvas na Figura 4.4(a,b). Uma distância de um Bark corresponde à largura de uma banda crítica. A equação a seguir permite converter frequências em Hertz para a escala Bark [50]:

$$\mathbf{z(f) = 13\ arctan(0,00076f) + 3,5\ arctan[(f/7500)^2]\ (Bark)} \quad (4.4)$$

Na Figura 4.4 (a) pode-se observar a conversão de frequências em Hertz para escala Bark. Também existe a escala mel [44], que é outra escala perceptual muito usada em codificadores de fala.

Banda Crítica	Frequência Inferior (Hz)	Frequência Central (Hz)	Frequência Superior (Hz)	Largura de Banda (Hz)
1	0	50	100	100
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210

12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500
25	15500	19500		

Tabela 4.2 - Bandas Críticas [33].

4.1.5 Mascaramento Auditivo

O mascaramento é um processo que ocorre quando um som torna-se imperceptível para um ouvinte devido à presença de outro som. Quando isso ocorre, o sinal que se torna imperceptível é denominado mascarado e o que provoca o mascaramento é denominado mascarador.

Embora o fenômeno do mascaramento deva ser analisado no plano tempo-frequência, é mais comum serem considerados dois efeitos separados, dependendo se o efeito ocorre no domínio do tempo ou da frequência.

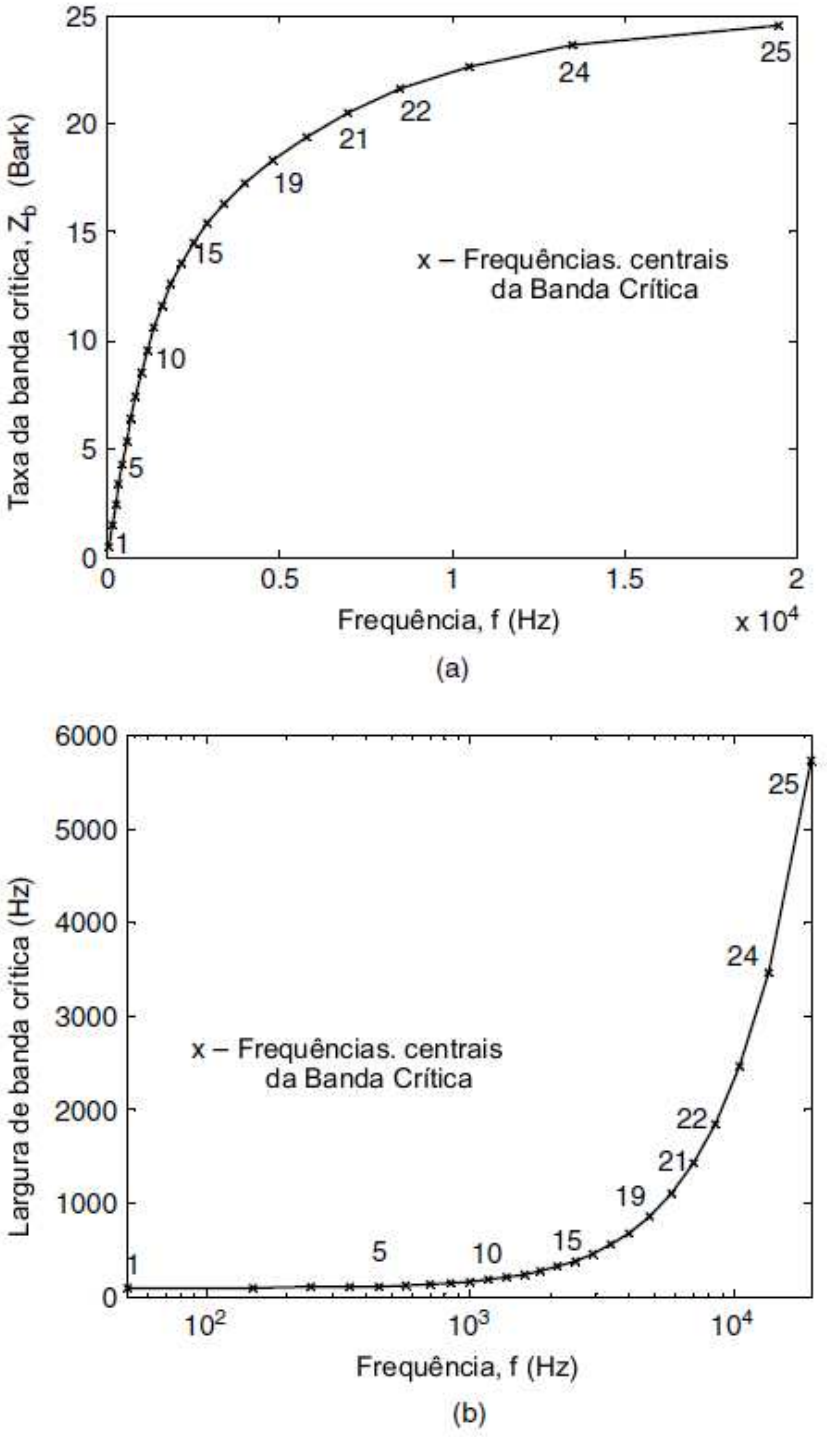


Figura 4.4 – Duas visualizações das larguras de banda críticas. (a) Taxa da banda crítica, Zb(f), faz um mapeamento de Hertz para Barks, e (b) largura de banda crítica, BWc(f) expressa a largura de banda crítica como uma função da frequência central, em Hertz. Os "x" denotam as frequências centrais dos bancos de filtros ideais de banda crítica observados na Tabela 4.2 (Adaptado de [43]).

O nível de energia abaixo do qual um componente do sinal é mascarado por outros componentes é chamado de limiar de mascaramento. O fenômeno do mascaramento auditivo será mais intenso quando os dois sinais estiverem dentro da mesma banda crítica, e menos efetivo se estiverem em bandas adjacentes.

No domínio da frequência, a forma do espectro de intensidade do sinal mascarador em relação ao mascarado é que determina em que medida que a presença de certa energia espectral vai mascarar a presença de outra. No domínio do tempo, as relações de fase entre os estímulos também podem influenciar a ocorrência do mascaramento [43].

O limiar de mascaramento varia com a frequência ao longo da banda crítica. Com isso, pode ser definido um limiar de mascaramento mínimo para uma banda crítica, abaixo da qual um sinal de maior intensidade torna inaudíveis todos os sinais de menor intensidade que se situam dentro dessa banda. A diferença de potência, expressa em dB, entre o mascarador e o limiar de mascaramento mínimo é chamada de relação sinal-máscara (signal-to-mask ratio, SMR) [19]. No limiar de detecção para o tom mascarado, a relação sinal-máscara (SMR) mínima ocorre quando a frequência do tom mascarado está próxima da frequência central do mascarador.

No gráfico da Figura 4.5 é ilustrado o efeito do limiar de audibilidade e de um mascarador em 250 Hz. O sinal com intensidade abaixo da curva do limiar de mascaramento gerada pelo mascarador é mascarado pelo sinal mascarador que tem maior intensidade, assim como qualquer sinal abaixo do limiar de audibilidade não será percebido.

Quando o mascaramento depende somente da localização no domínio da frequência, isto é, os sinais mascarado e mascarador são apresentados no mesmo instante de tempo, tem-se o que é conhecido como mascaramento simultâneo. Se o mascaramento depender principalmente da localização no domínio do tempo, então ele é chamado de mascaramento temporal ou não simultâneo [7].

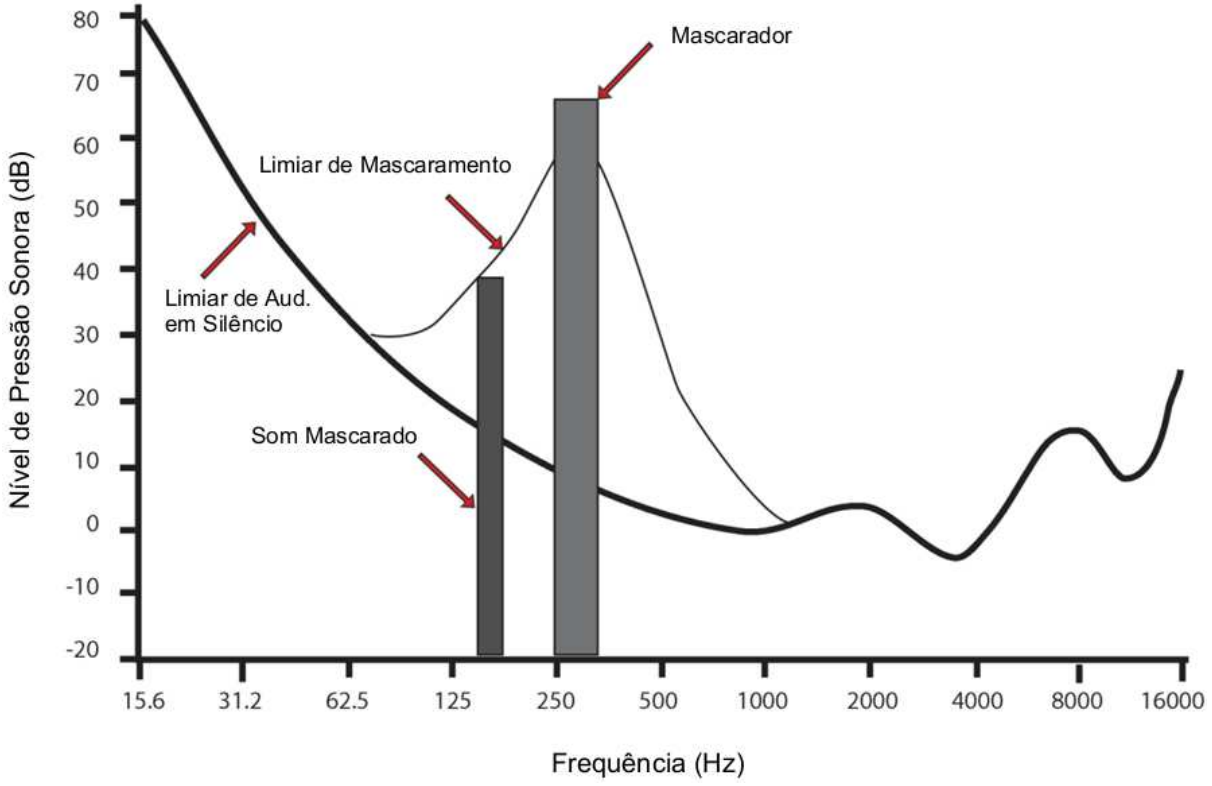


Figura 4.5 – Limiar de mascaramento e de audibilidade (Adaptado de [36]).

4.1.6 Mascaramento Simultâneo

O mascaramento simultâneo ocorre quando os sinais mascarado e mascarador estão presentes ao mesmo tempo no ouvido. Ocorre devido às limitações de resolução em frequência do ouvido humano, basicamente devido à existência das bandas críticas. Quando dois tons estão presentes na mesma banda crítica o de maior intensidade normalmente irá dominar a percepção sonora. A presença de um sinal mascarador cria tamanha excitação na membrana basilar e nas células ciliadas do órgão de Corti, que as oscilações provocadas pelo sinal mascarado não serão percebidas pelo ouvinte.

Considere um exemplo [13] em que exista um ruído com largura de banda de 1 Bark com intensidade de 40 dB. Ao ser adicionado um sinal tonal de 20 dB dentro da banda crítica, será observado um aumento de apenas 0,04 dB no nível de pressão sonora. O mascaramento simultâneo pode ser facilmente observado. Para isso, basta ser realizado um exame de audiometria na presença do mascarador.

A Figura 4.6 mostra o padrão de mascaramento causado por tons em quatro frequências distintas (0,25, 1, 4 e 8 kHz). As curvas mostram o nível mínimo que um sinal deve apresentar para se tornar audível (limiar de audibilidade), em função da frequência. A curva tracejada representa o nível mínimo de audição de tons na ausência de um sinal mascarador, o limiar de audibilidade. As curvas contínuas mostram o nível mínimo que um sinal deve apresentar para se tornar audível na presença de um sinal mascarador. Assim, ao se colocar um tom mascarador em 4 kHz, por exemplo, os tons nas frequências próximas têm que apresentar no mínimo o nível da curva contínua correspondente para que possam ser ouvidos simultaneamente com o mascarador. Note-se que o mascaramento abrange uma largura de faixa menor para baixas frequências do que para altas frequências, o que é uma consequência direta da definição das bandas críticas [7].

O mascaramento simultâneo pode ocorrer em algumas combinações de tipos de mascaradores e sinais mascarados. Os sinais mascarador e mascarado são classificados em tonais e ruído. Apesar do conteúdo espectral de um sinal de áudio na prática conter casos mais complexos de mascaramento simultâneo, para ser possível modelar as distorções na codificação é importante distinguir entre três tipos de mascaramento simultâneo, como ruído mascarando tom, tom mascarando ruído e ruído mascarando ruído.

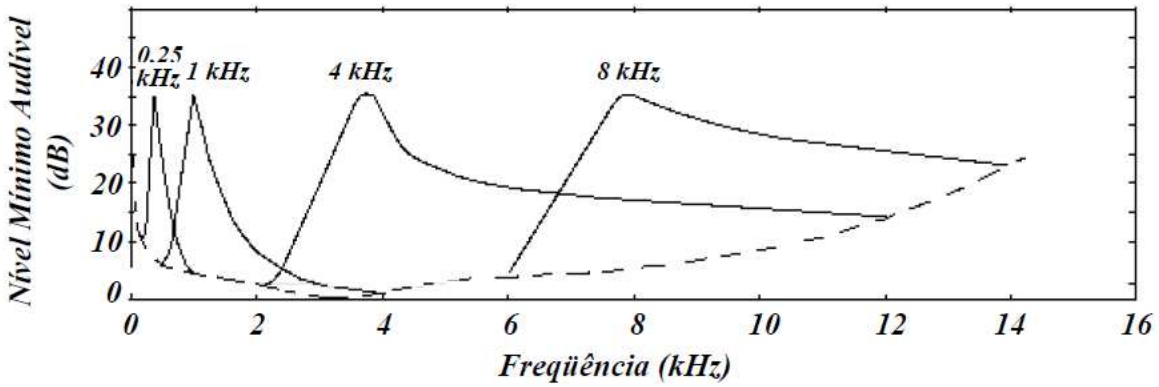


Figura 4.6 – Efeito de mascaramento simultâneo para quatro frequências distintas (reproduzido de [7]).

4.1.6.1 Ruído mascarando tom

Neste caso um ruído de banda estreita (tendo, por exemplo, uma largura de banda de 1 Bark) mascara um tom na mesma banda crítica, desde que a intensidade do tom mascarado esteja abaixo de um limiar diretamente relacionado à intensidade, e com menos importância, à frequência central do ruído mascarador. Existem vários estudos caracterizando o caso de ruído mascarando tom para um ruído aleatório e um estímulo puramente tonal, como os de Fletcher e Munson em 1937 [16] e Egan e Hake em 1950 [14].

Na maioria dos casos, o limiar de mascaramento para esse cenário varia entre -5 e +5 dB. É possível observar que fatores temporais também podem afetar o mascaramento simultâneo. Por exemplo, no caso de ruído mascarando tom, é possível ocorrer um efeito de sobrepassagem (*overshoot*) quando o início do tom de teste ocorre dentro de um intervalo curto imediatamente após o início do mascarador. O *overshoot* pode impulsionar o mascaramento simultâneo, ou seja, diminuir limiar mínimo da SMR, em até 10 dB em um pequeno espaço de tempo [50].

Em alguns casos, um ruído de menor intensidade pode mascarar um tom de maior intensidade [33].

Na Figura 4.7 é mostrado um exemplo de um ruído com largura de banda de 1 Bark, frequência central de 410 Hz e intensidade de 80 dB SPL, mascarando um tom de 76 dB SLP e de mesma frequência central.

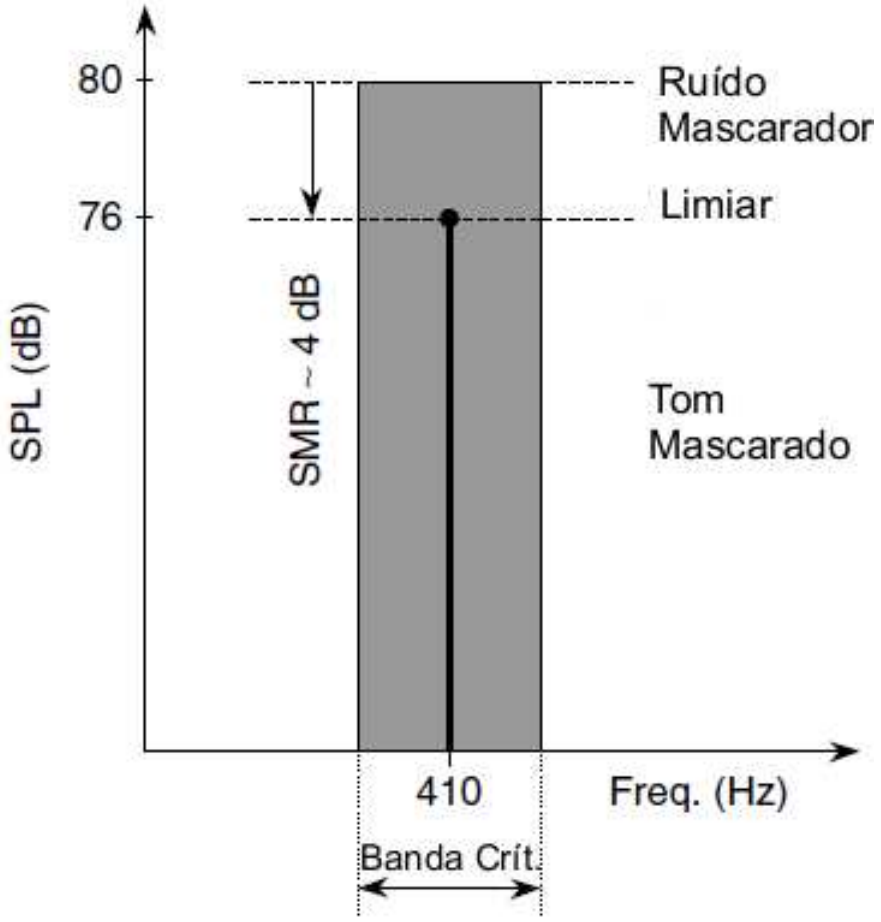


Figura 4.7 – Exemplo de um ruído de banda estreita mascarando um tom (Adaptado de [43]).

4.1.6.2 Tom mascarando ruído

Um tom que esteja no centro de uma banda crítica mascara ruídos contanto que o espectro do ruído esteja abaixo de um limiar diretamente relacionado à intensidade, e com menos influência, à frequência central do tom mascarador. Para esse cenário, o limiar de mascaramento varia entre 21 e 28 dB [33], [41].

Assim como ocorre no caso do ruído mascarando o tom, o limiar de mascaramento possui seu valor máximo quando o tom mascarador está no centro do espectro do ruído mascarado.

Na Figura 4.8, pode ser observado um exemplo de tom mascarando ruído, onde um tom em 1kHz com intensidade de 80 dB SPL mascara um ruído de banda estreita de largura de banda de 1 Bark centrado em 1 kHz com intensidade total de 56 dB SPL.

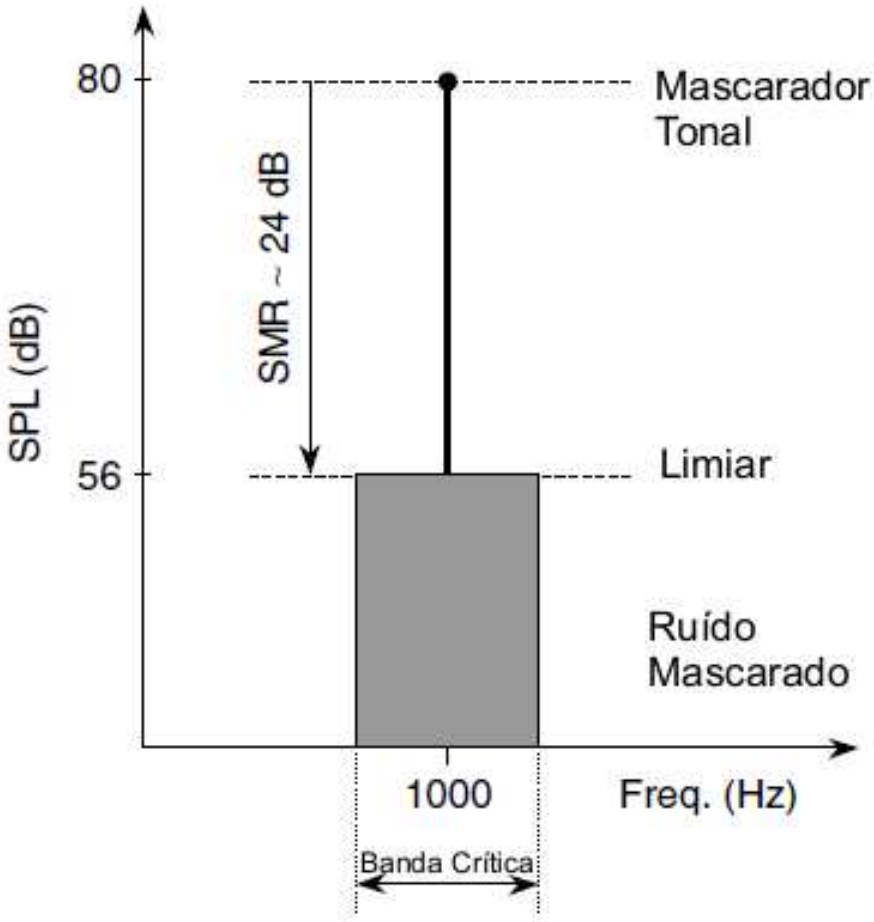


Figura 4.8 – Exemplo de um sinal tonal mascarando um ruído de banda estreita (Adaptado de [43]).

Comparando-se os casos mostrados na Figura 4.7 e na Figura 4.8 Pode-se observar uma assimetria no poder de mascaramento do ruído e do tom, na qual o ruído possui um poder de mascaramento muito maior.

4.1.6.3 Ruído mascarando ruído

O caso de ruído de banda estreita mascarando outro ruído de banda estreita é mais difícil de ser analisado do que o do ruído mascarando tom e vice-versa por causa da influência das

relações de fase entre o ruído mascarador e o mascarado. Essencialmente, diferenças relativas de fase entre os componentes de cada um dos ruídos podem levar a valores de limiar diferentes. Limiares da ordem de 26 dB já foram observados para esse tipo de mascaramento [1].

Os codificadores perceptuais fazem uso, principalmente, dos casos de tom mascarando ruído e ruído mascarando tom.

4.1.7 Mascaramento Temporal

O mascaramento temporal ou não simultâneo é aquele que ocorre na não-simultaneidade do mascarador. Ele pode ser dividido em dois tipos de efeitos: mascaramento progressivo (ou pós-mascaramento) e mascaramento retrógrado (ou pré-mascaramento). No caso do mascaramento progressivo, os componentes do sinal são mascarados após o término do mascarador, e no caso do mascaramento retrógrado, os componentes são mascarados antes do início da execução do mascarador. Um exemplo destes mascaramentos pode ser observado no gráfico da Figura 4.9.

O mascaramento progressivo ocorre devido ao fato do sistema auditivo humano demorar certo tempo para se recuperar após um sinal de alta intensidade, pois quando isso acontece, os neurônios disparados ficam em um estado refratário que pode durar mais de 100 ms. O mascaramento progressivo tem um efeito bem mais significativo do que o pré-mascaramento. Os efeitos deste tipo de mascaramento foram observados em até 200 ms após a presença do mascarador [13].

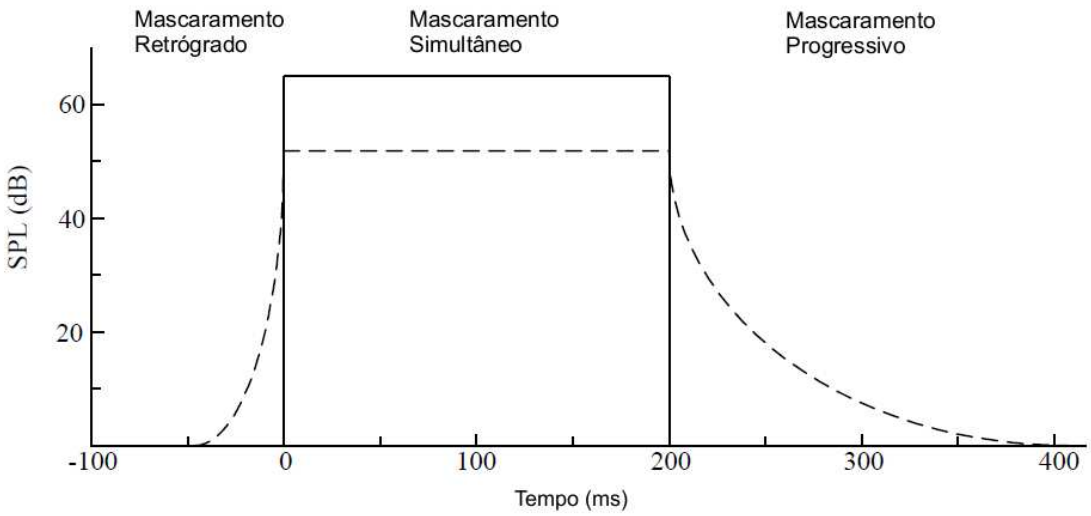


Figura 4.9 – Gráfico ilustrando os principais tipos de mascaramento [13].

O pré-mascaramento ocorre devido a limitações na resolução temporal do sistema auditivo humano e latência do processamento pelo sistema nervoso central, mas seu efeito só é significativo durante 1 ou 2 ms, por causa disso recebe menos atenção que outros tipos de mascaramento. Ele é um fenômeno mais complicado, pois implica em um sinal de grande amplitude mascarar outro sinal antes de o primeiro estar realmente presente. Tal fenômeno é normalmente explicado pela suposição de que o sinal forte é processado mais rapidamente do que o sinal fraco, podendo, com isso, ultrapassar o sinal mascarado durante o processamento dos sinais, ou no nervo auditivo, ou posteriormente, nos níveis mais elevados do sistema auditivo [7].

4.1.8 Espalhamento do Mascaramento

Apesar dos efeitos do mascaramento serem muito maiores dentro da banda crítica, eles propagam-se pelas demais regiões do espectro. Esse efeito é conhecido como espalhamento do mascaramento.

Para os casos de ruído mascarando tom e tom mascarando ruído, foi observado que o mascaramento máximo ocorre quando a frequência central do ruído de faixa estreita coincide com a frequência do sinal tonal.

Devido às características físicas da membrana basilar, para as demais regiões do espectro, o decaimento do nível de mascaramento ocorre de maneira diversa. Para as frequências menores do que a do máximo, o decaimento do nível de espalhamento é muito mais rápido do que para as maiores.

Tipicamente, o espalhamento do mascaramento é aproximado por uma função triangular na escala Bark, independentemente da frequência e do nível do sinal mascarador.

Essa função é conhecida como função de espalhamento. Existem várias funções de espalhamento que já foram propostas na literatura, mas a mais usada é a de Schroeder [1], que possui um decaimento de 25 dB/Bark para as frequências menores que o máximo, e de 10dB/Bark para as maiores. Sua forma analítica é dada por

$$SF_{cB}(z) = 15{,}81 + 7{,}5(z + 0{,}474) - 17{,}5\sqrt{1 + (z + 0{,}474)^2} \quad \text{(dB)} \quad (4.1)$$

onde z é a frequência em Bark.

A Figura 4.10 mostra de forma gráfica a função de espalhamento. Na Figura 4.10(a), pode ser observada a função de espalhamento em Hertz, para um mascarador localizado em 2450 Hz. Na Figura 4.10(b), é mostrada a função com vários mascaradores localizados em frequências múltiplas de 450Hz. Nas figuras (c) e (d), são mostrados os mesmos conjuntos de mascaradores das figuras (a) e (b) respectivamente, só que na escala Bark [33].

4.1.9 Entropia Perceptual

Em 1998, Johnston propôs uma medida para determinar um limite teórico da compressibilidade de sinais de áudio com base na medida do conteúdo que seria perceptualmente relevante. Este limite chamado de entropia perceptual (*perceptual entropy* – PE) representa a quantidade de informação relevante em um determinado sinal de áudio, em bits por amostra (ou bits/s), para atingir-se a codificação transparente. A entropia perceptual é obtida baseando-se tanto na análise psicoacústica do sinal quanto na entropia estatística.

A entropia perceptual representa um limite teórico de compressibilidade para um sinal específico. Medidas da PE publicadas por Johnston em [30] sugerem que uma grande variedade de fontes de áudio com qualidade de CD poderiam ser comprimidas em aproximadamente 2,1 bits por amostra.

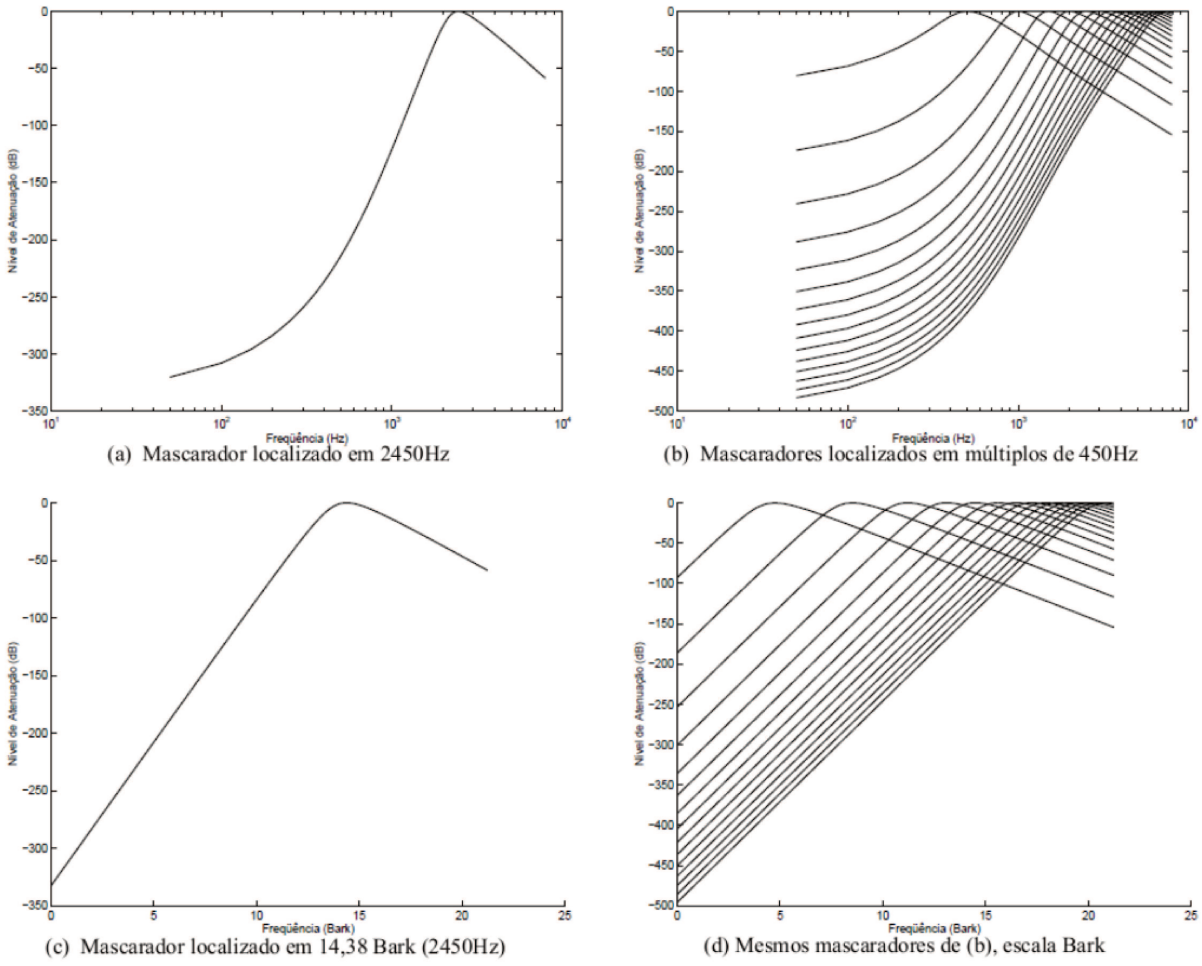


Figura 4.10 – Gráficos da função de espalhamento (Adaptado de [33]).

Para estimar o valor da PE é primeiramente feito um janelamento e transformação do sinal para o domínio da frequência, em seguida é determinado o limiar de mascaramento usando o modelo perceptual, por fim é feito o cálculo do número de bits necessários para quantizar o espectro sem adicionar nenhum ruído perceptível. A medida da PE é obtida através de um histograma da PE ao longo de vários quadros e então é escolhido o valor do pior caso para ser utilizado como valor da PE.

A transformada para o domínio da frequência é feita usando uma janela de Hann e uma FFT de 2048 pontos. Os limiares de mascaramento são obtidos através da análise das bandas críticas (incluindo espalhamento), determinando se o sinal é do tipo tonal ou ruído, aplicando regras de mascaramento de acordo com a qualidade do sinal e então considerando o limiar absoluto de mascaramento.

4.2 O MODELO PSICOACÚSTICO DO PADRÃO MPEG-2 AAC

Um modelo psicoacústico informativo (não normativo, o padrão só normatiza a parte do decodificador) é fornecido no padrão MPEG-2 AAC [23] o qual é praticamente igual ao segundo modelo psicoacústico apresentado no padrão MPEG-1 [21].

O modelo perceptual do AAC foi desenvolvido para suportar várias taxas de amostragem e dois tamanhos de bloco, com 128 ou 1024 amostras, chamados de bloco curto e longo respectivamente. O tamanho do bloco que será utilizado varia de acordo com a parte do sinal de entrada analisada, e tem como objetivo aumentar a resolução temporal para minimizar efeitos como o pré-eco.

O pré-eco ocorre quando um pico de energia de rápida duração temporal (como o de um ataque de bateria) acontece após um período de baixa energia (como um período de silêncio) e quando o início do pico de energia ocorre durante a janela de análise do modelo perceptual. Quando ocorre o pré-eco, um ruído de quantização se torna perceptível ao ouvinte durante a fase de baixa energia, uma vez que o limiar de audibilidade é calculado considerando-se que o sinal esteja estacionário durante toda a janela de análise [33].

O modelo calcula a energia máxima de distorção inaudível para a codificação de um quadro de áudio. As saídas do modelo são um limiar de mascaramento para cada sub-banda do codificador e a relação sinal-máscara (SMR).

4.2.1 Cálculo do espectro complexo do sinal

O primeiro passo no cálculo do limiar é obter o espectro complexo do sinal de entrada, X[*k*]. Para isto a FFT do sinal de entrada é calculada após este ser multiplicado por uma janela de Hann definida por:

$$\boldsymbol{\omega(n) = \frac{1}{2}\left[1 - cos\left(\frac{2\pi n}{N}\right)\right]} \quad \textbf{(4.5)}$$

onde N é o tamanho da janela em número de amostras, são definidas 1024 amostras para bloco longo e 128 para curto.

X[*k*] é representado na forma polar em termos dos seus componentes de magnitude, *r*[*k*] e de fase, ϕ[*k*],

$$\mathbf{X[k] = r[k]e^{j\Phi[k]}} \quad \textbf{(4.6)}$$

A energia em cada partição do codificador é calculada somando-se as energias de cada componente dentro de uma sub-banda,

$$\boldsymbol{e[b] = \sum_{k=k_l}^{k_h} r^2[k]} \quad \textbf{(4.7)}$$

onde $\boldsymbol{k_l}$ e $\boldsymbol{k_h}$ definem os limites inferior e superior respectivamente da sub-banda *b*.

4.2.2 Cálculo do coeficiente de tonalidade

A tonalidade do sinal de entrada, que é usada para estimar a quantidade de mascaramento produzida, é estimada usando um método proposto em [12]. Ao invés de fornecer um valor

global, este método calcula um índice de tonalidade local para cada sub-banda que é estimado através de uma medida de coerência. A medida de coerência corresponde a uma predição de componentes espectrais, em coordenadas polares, a partir do espectro de dois quadros anteriores,

$$\boldsymbol{r_{pred}[k] = r_{t-1}[k] + (r_{t-1}[k] - r_{t-2}[k]),} \tag{4.8}$$

$$\boldsymbol{\phi_{pred}[k] = \phi_{t-1}[k] + (\phi_{t-1}[k] - \phi_{t-2}[k]),} \tag{4.9}$$

onde r_{pred} e ϕ_{pred} representam a magnitude e a fase previstas. Cada par de elementos de predição é transformado numa medida de imprevisibilidade, $c[k]$, comparando com o valor atual, que representa o desvio entre o valor estimado e o valor atual:

$$\boldsymbol{c[k] = \frac{\left|X[k] - X_{pred}[k]\right|}{|r[k]| + \left|r_{pred}[k]\right|}} \tag{4.10}$$

A imprevisibilidade da partição, c[*b*], isto é, o desvio entre o valor estimado e o valor atual é agrupado para cada banda *b*, de modo que cada componente espectral é ponderada pela sua energia, de acordo com a equação:

$$\boldsymbol{c(b) = \sum_{k=li_i}^{ls_i} c(k) r^2(k)} \tag{4.11}$$

onde $\boldsymbol{li_i}$ e $\boldsymbol{ls_i}$ são, respectivamente, os limites inferior e superior da banda *i*.

Em seguida, a energia dos componentes espectrais de cada banda deverá ser somada para ser obtida a energia total da banda, E(*b*), de acordo com a equação:

$$\boldsymbol{E(b) = \sum_{k=li_i}^{ls_i} r^2(k)} \tag{4.12}$$

A energia da banda, E(b), e o desvio dos coeficientes estimados da banda, C(b), deverão passar por uma convolução com a função de espalhamento, SF(i,j), definida a seguir.

Antes de ser feito o cálculo da função de espalhamento, é necessário definir os seguintes coeficientes:

$$\boldsymbol{\alpha} = \begin{cases} \mathbf{3}(\boldsymbol{j} - \boldsymbol{i}), & \boldsymbol{j} \geq \boldsymbol{i} \\ \mathbf{1,5}(\boldsymbol{j} - \boldsymbol{i}), & \boldsymbol{j} < \boldsymbol{i} \end{cases} \quad \textbf{(4.13)}$$

$$\boldsymbol{\beta} = \mathbf{min}\left((\boldsymbol{\alpha} - \mathbf{0,5^2}) - \mathbf{2}(\boldsymbol{\alpha} - \mathbf{0,5}), \mathbf{0}\right) \quad \textbf{(4.14)}$$

$$\boldsymbol{\gamma} = \mathbf{15,811389} + \mathbf{7,5}(\boldsymbol{\beta} + \mathbf{0,474}) - \mathbf{17,5}(\mathbf{1} + (\boldsymbol{\beta} + \mathbf{0,474})^2)^{\mathbf{0,5}} \quad \textbf{(4.15)}$$

onde *i* representa as frequências em Bark do sinal espalhado e *j* é a frequência central da banda em que o sinal será espalhado.

A função de espalhamento para o modelo definido no padrão MPEG-2 AAC é então definida por:

$$\boldsymbol{SF(i,j)} = \begin{cases} \mathbf{0}, \ \boldsymbol{\gamma} < -100 \\ \mathbf{10}^{\frac{(\beta+\gamma)}{\mathbf{10}}} \ \boldsymbol{\gamma} \geq \mathbf{-100} \end{cases} \quad \textbf{(dB SPL).} \quad \textbf{(4.16)}$$

É importante observar que os coeficientes da função de espalhamento da equação (4.16) foram obtidos de maneira empírica, para ser possível atingir uma menor complexidade computacional, prejudicando o mínimo possível a precisão dos valores.

Como resultado da convolução tem-se $\mathrm{E_s(b)}$ como a energia das bandas após o espalhamento e $\mathrm{C_s(b)}$ como os desvios das estimativas.

Devido ao fato de os desvios das estimativas $\mathrm{C_s(b)}$ terem sido ponderados pela energia de cada componente espectral, será necessário calcular os desvios das estimativas normalizados pela energia da banda após o espalhamento, $E_s(b)$, através da seguinte equação:

$$\overline{\boldsymbol{C_s}}(\boldsymbol{b}) = \frac{\boldsymbol{C_s(b)}}{\boldsymbol{E_s(b)}} \quad \textbf{(4.17)}$$

O coeficiente de tonalidade é dado por:

$$\boldsymbol{t(b)} = \mathbf{-0,299 - 0,43ln}\left(\overline{\boldsymbol{C_s}}(\boldsymbol{b})\right) \quad \textbf{(4.18)}$$

4.2.3 Cálculo do Limiar Global de Mascaramento

A energia da banda espalhada $E_s(b)$ deverá ser normalizada, devido à natureza não normalizada da função de espalhamento. Essa normalização é dada por:

$$\overline{E_s}(b) = \frac{E_s(b)}{N(b)} \tag{4.19}$$

onde

$$N(b) = \sum_{i=1}^{b_{max}} SF(i,j) \tag{4.20}$$

O modelo do AAC considera que para o cenário de ruído mascarando tom, um sinal será mascarado se este estiver 6 dB abaixo do sinal mascarador, independentemente da banda em questão. Para o cenário de tom mascarando ruído, o limiar é de 18 dB. Portanto, a relação sinal ruído é calculada por:

$$SNR(b) = 18t(b) + 6(1 - t(b)) \tag{4.21}$$

O limiar de mascaramento é dado por:

$$T(b) = \overline{E_s}(b)10^{-SNR(b)/10} \tag{4.22}$$

Para reduzir os efeitos de pré-eco, o modelo perceptual do AAC compara o limiar de mascaramento atual ao limiar de mascaramento do bloco anterior, Se o limiar de mascaramento do bloco corrente for maior que λ vezes o limiar do bloco anterior, o limiar a ser considerado é limitado a λ vezes o limiar do bloco anterior. Com isso, o limiar global de mascaramento é dado por:

$$T_g(b) = \min\left(T(b), \lambda T_{g_{t-1}}(b)\right) \tag{4.23}$$

onde λ é a relevância da influência do passado e tem valor 1 para janelas de 128 amostras e valor 2 para janelas de 1024 amostras.

4.2.4 Cálculo da relação sinal mascaramento (SMR)

Além do limiar global de mascaramento, a outra saída do modelo perceptual do codificador AAC é a relação sinal mascaramento, SMR (*Signal-to-Mask Ratio*), que tem uma definição análoga a da relação sinal ruído, como foi visto na seção 4.1.5, ela define a relação entre a energia do sinal e o limiar global de mascaramento. Logo, a relação sinal mascaramento é dada por:

$$\boldsymbol{SMR(b) = 10log_{10}\left(\frac{E(b)}{T_g(b)}\right)} \qquad \textbf{(4.24)}$$

5 ÁUDIO MULTICANAL

O termo áudio multicanal refere-se ao áudio gerado e reproduzido em múltiplos canais para criar uma sensação envolvente do som no ouvinte.

A configuração mais comum é a 5.1, que são cinco canais distintos cobrindo toda a faixa de frequências audíveis distribuídos em: um par estéreo frontal, um canal central, outro par estéreo traseiro e mais um canal para efeitos de baixa frequência (até 120Hz), conhecido como canal LFE (*low frequency effect*), a ser reproduzido por um *subwoofer*, que é uma caixa de som otimizada para reprodução de baixas frequências (20 a 200 Hz normalmente).

Os padrões de televisão digital atuais já compreendem a transmissão de áudio em formato multicanal. A norma brasileira usa o formato padrão MPEG-4 AAC [25] para codificar o áudio a ser transmitido e prevê além do estéreo, a configuração de canais 5.1 [4].

O AAC é um codificador perceptual de áudio que possui ferramentas para codificação de áudio incluindo vários canais.

5.1 LOCALIZAÇÃO DO SOM

A reprodução de áudio de modo que seja convincente o suficiente para ser o mais próximo possível da sensação real de presença no ambiente, requer que o sistema que será usado para reproduzir o áudio seja capaz de sintetizar o máximo de indicações perceptuais (*perceptual cues)* necessárias. Essas indicações incluem informações sobre a localização de cada fonte sonora no espaço onde foi gravada e também a interação de cada fonte com elementos acústicos do ambiente [49].

O processo de localização do som pelo ser humano depende da análise dos sinais que chegam às duas orelhas. Diferenças na intensidade, tempo de chegada e mudanças espectrais dependentes da direção causadas pela orelha externa e parte superior do corpo são os principais elementos que provém indicações de localização. Essas diferenças no espectro da onda sonora entre as orelhas são as chamadas diferenças interaurais.

O estudo destas indicações identificou dois mecanismos básicos que são responsáveis pela localização da fonte sonora: diferenças de tempo interaural (*interaural time differences* - ITDs) e

diferenças de intensidade interaural (*interaural level differences* - ILDs). A ITD e a ILD operam cada uma em diferentes comprimentos de onda.

Para baixas frequências, tipicamente abaixo de 1 kHz, com comprimento de onda maior, a cabeça é muito pequena em comparação com o comprimento de onda e a localização é baseada nas diferenças de tempo de chegada às orelhas, isto é, opera o mecanismo ITD, que pode ser observado na Figura 5.1 (a).

Para frequências altas (comprimento de onda menor), opera o mecanismo ILD, que pode ser visto na Figura 5.1 (b). Neste caso, é a diferença de amplitude nos dois ouvidos, provocada pela sombra acústica projetada pela cabeça, que fornece indicação de direção.

Movimentos da cabeça permitem resolver ambiguidades entre sons frontais e posteriores [45].

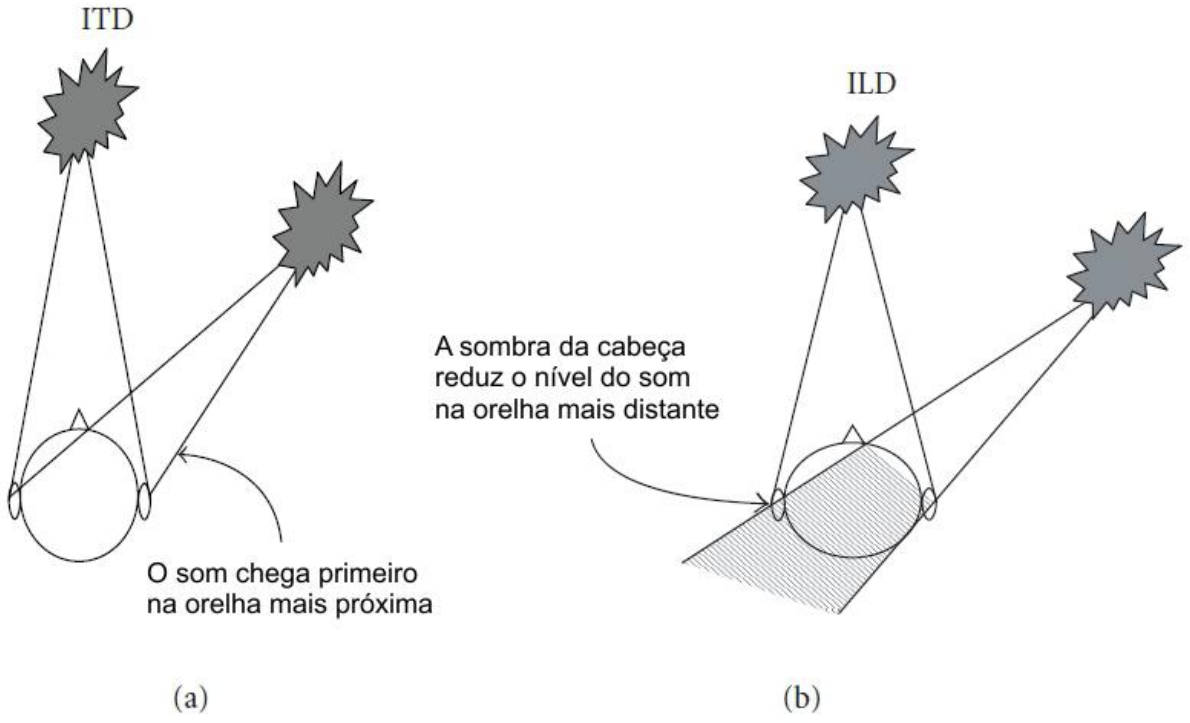


Figura 5.1 – Mecanismos responsáveis pela localização da fonte sonora, ITD e ILD [49] .

Estes dois conceitos de localização do som formaram a base da estereofonia, que começou com o trabalho de Blumlein [10] que foi o primeiro a reconhecer que era possível localizar um som dentro de um faixa de ângulos de azimute através do uso de uma combinação apropriada de diferenças de tempo e intensidade. O trabalho dele focou no desenvolvimento de

técnicas de microfone correspondentes que permitiriam a gravação das diferenças de amplitude e fase necessárias para reprodução estéreo. Fletcher, Snow e Steinbeig dos laboratórios Bell usaram um abordagem diferente. Eles examinaram o seguinte questionamento: “quantos canais de alto-falantes seriam necessários para criar uma representação exata de uma cena sonora?”. Os estudos deles mostraram que se um número infinito de microfones for usado para capturar uma cena sonora, então é possível obter uma reprodução perfeita usando um número infinito de alto-falantes. Apesar de ser um resultado teórico interessante, os pesquisadores sabiam que aplicações práticas iam precisar de um número menor de canais. Eles demonstraram que um sistema composto por três canais, sendo estes, esquerdo, direito e central no plano de azimute poderiam representar a lateralização e a profundidade do campo de som desejado com uma precisão aceitável. Com isso, o estéreo foi criado como um sistema de três canais. O primeiro sistema deste tipo foi demonstrado em 1934 com a orquestra da Filadélfia tocando na Academia de Música da Filadélfia transmitindo o som para uma audiência que estava localizada em Washington através de linhas telefônicas.

5.2 TÉCNICAS DE CODIFICAÇÃO MULTICANAL

Na codificação de áudio existem várias técnicas para explorar redundâncias entre dois ou mais canais de áudio que serão codificados juntos. Estas técnicas visam melhorar a eficiência da codificação eliminando redundâncias que possam existir entre os canais e com isso reduzir a taxa de bits necessária mantendo a qualidade.

5.2.1 Codificação de pares estéreo

O termo *joint stereo* é usado para referir-se a técnicas usadas para codificação conjunta de pares estéreo de canais que tenham com o objetivo explorar as redundâncias que possam existir entre este par de canais.

As técnicas mais usadas são *M/S Stereo* e *Intensity Stereo*. As duas técnicas podem ser aplicadas seletivamente em diferentes regiões de frequência do sinal. No padrão MPEG-2 e 4 são usadas ambas as técnicas no codificador AAC.

5.2.2 Codificação *M/S Stereo*

Também conhecida como codificação de soma e diferença, a técnica *M/S Stereo* consiste em substituir os canais direito e esquerdo pela sua soma e diferença normalizadas que são então chamados de canal meio (*mid* - *M*) e canal lateral (*side* - *S*) respectivamente, de acordo com as equações seguintes:

$$\boldsymbol{M = \frac{L + R}{2}} \tag{5.1}$$

$$\boldsymbol{S = \frac{L - R}{2}} \tag{5.2}$$

A operação de soma e diferença é realizada nos coeficientes espectrais do sinal e por isso pode ser feita seletivamente de acordo com a frequência do sinal.

A matriz de soma/diferença usada na codificação estéreo M/S é inversível. Com exceção da codificação e quantização da matriz de saída, o processamento *joint stereo* é completamente transparente e por isso pode também ser aplicado em altas taxas de bits, mantendo alta qualidade de áudio sem introduzir artefatos.

O ganho de codificação depende fortemente do sinal original. Ele varia de um máximo próximo de 50% no caso onde os sinais dos canais direito e esquerdo são iguais (ou exatamente fora de fase) até situações onde não compensa usar M/S, pois gastaria mais bits do que codificar os canais separadamente [20]. No caso em que os canais são iguais, a diferença entre eles será zero e com isso a informação no canal lateral será toda composta de zeros, que podem ser transmitidos por um único bit que indique que todos estes valores serão zero. Com isto, o canal meio tem o dobro dos bits disponíveis para ser transmitidos, tendo um ganho de codificação de aproximadamente 50%. Apesar de na prática ser raro os canais serem idênticos, a informação do canal lateral normalmente tem um valor menor que no canal esquerdo ou direito, sendo possível a redução no número de bits usando o M/S na maioria dos casos. O codificador pode alternar entre

usar o modo M/S ou codificar os canais esquerdo e direito normalmente, usando um algoritmo que analise o sinal em cada bloco para decidir se haverá ganho, como foi feito por Johnston em [31], onde a decisão é feita a partir de limiares calculados para cada canal onde é feita uma comparação do quanto estes limiares variam entre o canal direito e esquerdo.

O M/S é usado nos codificadores da família ISO/MPEG o MPEG-1/2 camada 3 (MP3) e nos da família MPEG2/4 AAC é usado de maneira melhorada, aplicado individualmente para cada banda de fator de escala. Em fontes de áudio multicanal, a técnica M/S é aplicada a pares de canais que são simétricos para o ouvinte.

5.2.3 Codificação *Intensity Stereo*

A técnica Intensity Stereo (IS) explora o princípio de que as componentes de alta frequência do som são percebidas pelo ouvido através da análise de seus envelopes de energia no tempo ao invés dos sinais em si. Então o codificador pode transmitir apenas um conjunto de valores espectrais e compartilhá-los entre vários canais de áudio mantendo uma boa qualidade, apenas sendo necessário adicionar a energia dos envelopes como dados auxiliares para o decodificador conseguir recuperar o nível correto de energia original do sinal [49].

A técnica IS não oferece uma reconstrução perfeita para sinais de áudio em geral e é mais usada em taxas de bit mais baixas e aplicada apenas nas frequências mais altas do sinal.

Para codificação de áudio multicanal, a técnica pode ser generalizada combinando os coeficientes espectrais de vários canais em um único conjunto de coeficientes espectrais mais informação de dimensionamento para cada canal.

5.2.4 Técnica de Estéreo Paramétrico

A técnica de estéreo paramétrico, chamada na literatura de *parametric stereo* (PS), sintetiza dois canais a partir de um canal mono que é transmitido junto com informação lateral. Ao invés de usar o banco de filtros do próprio codificador, é usado um banco de filtros dedicado para resintetizar os dois canais estéreo a partir do mono transmitido. A partir do sinal de entrada

estéreo ($l[n]$, $r[n]$), os parâmetros estéreo que variam com o tempo são estimados em uma grade de frequência não uniforme, que lembra a grade *Equivalent Rectangular Bandwidth* (ERB). Estes parâmetros descrevem as indicações de localização espacial que são perceptualmente relevantes.

Em seguida, um *downmix* mono, $m[n]$ é gerado. Este sinal mono pode ser codificado por qualquer codificador de áudio. Os parâmetros do sinal estéreo são então quantizados e codificados na parte de dados auxiliares do *bitstream* mono. Um diagrama de blocos do codificador PS pode ser visto na Figura 5.2.

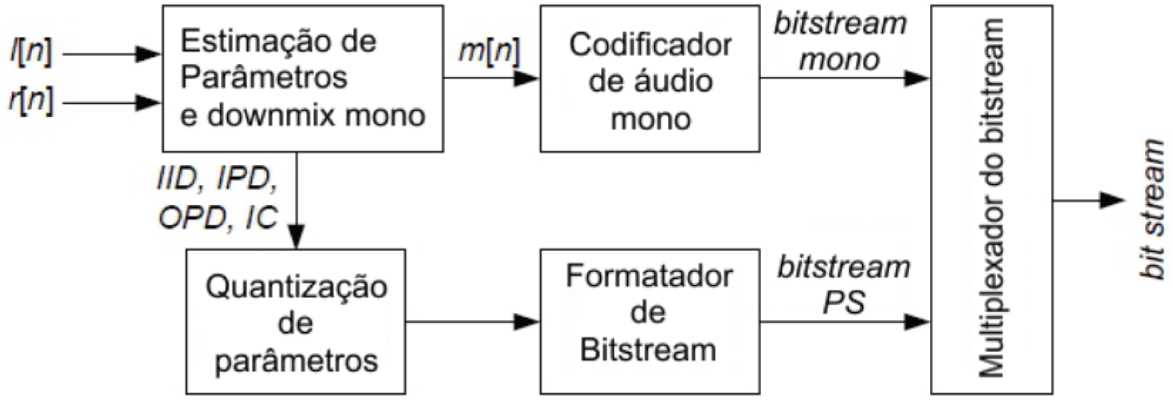


Figura 5.2 – Diagrama de blocos de um codificador PS genérico [42].

Existem alguns parâmetros que podem ser aplicados em um sistema de estéreo paramétrico para descrever a imagem estéreo como: diferença de intensidade entre canais, *Inter-channel Intensity Differences* (IID), correlação cruzada entre canais, *Inter-chanel Coerence* (IC) e diferença de fase entre canais, *Inter-chanel Phase Differences* (IPD).

A IC é medida como o máximo da correlação cruzada em função do tempo ou da fase. Além desses três parâmetros, existe um quarto tipo, que descreve um atraso ou diferença de fase geral, *Overall Phase Difference* (OPD), já que o IPD apenas especifica as diferenças de fase relativas entre os canais estéreo do sinal de entrada e não a distribuição destas diferenças de fase ao longo dos canais esquerdo e direito.

Para reconstruir a imagem estéreo, várias operações são executadas no decodificador PS, que consistem de: escalonamento (IID), rotações de fase (IPD/OPD) e descorrelação (IC). Um diagrama de blocos do decodificador PS pode ser visto na Figura 5.3.

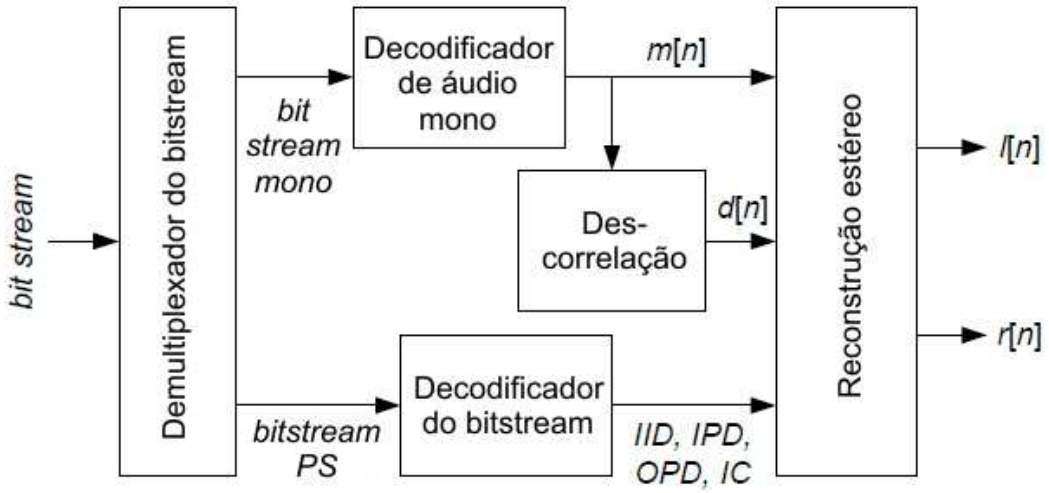


Figura 5.3 – Diagrama de blocos de um decodificador PS genérico [42].

5.2.5 Binaural Cue Coding (BCC)

A técnica *Binaural Cue Coding* (BCC) pode ser considerada uma generalização da ideia da técnica PS, gerando múltiplos canais na saída a partir de um único canal mais alguma informação lateral. O áudio de todos os canais é misturado em um único canal por um processo conhecido como *downmix* e são extraídas informações de localização extras que serão transmitidas em paralelo para permitir a reconstrução de todos os canais por um decodificador compatível. Essas informações podem ser transmitidas com uma taxa bem menor que se fossem transmitidos todos os canais.

Com o BCC é possível transmitir áudio multicanal onde taxas de bits muito baixas são requeridas, como por exemplo, uma transmissão para dispositivos móveis.

Um exemplo de uma codificação de um sinal estéreo usando BCC pode ser visto em [8], este esquema é conhecido como BCC para Renderização Natural, também conhecido como BCC tipo II. No transmissor, um analisador BCC extrai as indicações espaciais binaurais a partir do sinal estéreo original, *L* e *R*. Em seguida é feito um *downmix* do sinal estéreo para mono e este é comprimido por um codificador de áudio adequado. No receptor, o sinal de áudio mono é decodificado e o sintetizador BCC reconstrói a imagem espacial restaurando as indicações de

localização espacial, gerando na saída o sinal estéreo a partir do sinal mono. Este esquema é bem parecido com a codificação *Intensity Stereo*. Mas o BCC é melhor para ser aplicado a todas as frequências do sinal e não só às mais altas como no caso da codificação IS [9].

A análise e síntese BCC são feitas separadamente do codificador de áudio, com isso codificadores de áudio mono ou de voz já existentes podem ser modificados para trabalhar com áudio multicanal adicionando informações do BCC e usando um decodificador capaz de reconstruir todos os canais usando estas informações.

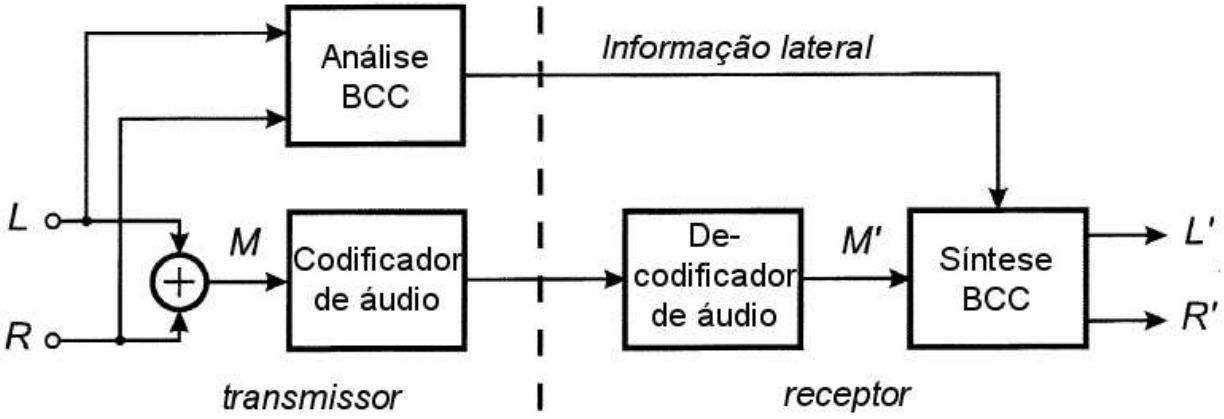


Figura 5.4 – Esquema genérico de codificação BCC para um sinal estéreo [8].

6 O PADRÃO MPEG E O SBTVD

O MPEG-4 é um padrão que foi desenvolvido pelo comitê MPEG (*Moving Pictures Experts Group*). O MPEG é um grupo de trabalho que faz parte das organizações ISO (*International Organization for Standardization*) e IEC (*International Electrotechnical Commission*). A norma que define o MPEG-4 é a ISO/IEC 14496, que foi finalizada em outubro de 1998 e se tornou um padrão internacional no início de 1999.

O objetivo do MPEG é desenvolver padrões para compressão, processamento e representação de imagens em movimento e áudio. Este grupo foi responsável também pelo desenvolvimento dos padrões MPEG-1 (onde foi definido o formato MP3) e mais tarde MPEG-2 (usado na transmissão de vídeo e áudio em alguns sistemas de televisão digital e no vídeo gravado em mídia DVD). O MPEG-4 é o padrão mais recente para codificação audiovisual.

O MPEG-4 divide o conteúdo em objetos, que podem ser unidades de conteúdo visual ou de áudio. O conteúdo pode ser de origem natural, isto é, gravado com uma câmera ou microfone, ou sintético, como áudio gerado por um computador.

O padrão também é divido em perfis e níveis, que nada mais são que grupos de ferramentas disponíveis no padrão, para facilitar a escolha do que será usado de acordo com a aplicação desejada. Dentro de cada perfil é definido um número de níveis, que servem como uma maneira de limitar a complexidade computacional, por exemplo, limitando a taxa de bits, de amostragem, o número de canais de áudio, o número máximo de objetos em uma cena, etc.

O MPEG-4 é composto por várias partes que podem ser implementadas individualmente, como por exemplo, a parte de áudio que pode ser usada por si só, ou combinada com outras partes, como em um filme que usa as partes de áudio e vídeo combinadas em um mesmo fluxo de dados.

A base do padrão é formada pelas partes de sistemas (parte 1), visual (parte 2) e áudio (parte 3). A parte DMIF (*Delivery Multimedia Integration Framework*, parte 6) define uma interface entre aplicação e transmissão/armazenamento. Conformidade (*Conformance,* parte 4) define como testar uma implementação do MPEG-4, e a parte 5 fornece um *software* de referência, que pode ser usado para começar a implementar o padrão e serve como um exemplo de aplicação prática [27].

O SBTVD segue as normas da ABNT 15601 a 15608. A parte de codificação e multiplexação baseia-se principalmente nas normas MPEG-2 e 4.

As normas mais importantes dentro do escopo deste trabalho são as normas ABNT 15602 partes 2 e 3. A norma ABNT 15601 define a parte de transmissão, a 15602 define a parte de codificação e é divida em três partes: a parte 1 trata o vídeo, a parte 2 trata o áudio, a parte 3 trata de sistemas de multiplexação de sinais.

As outras normas utilizadas no SBTVD são a 15603 que trata a multiplexação e serviços de informação (SI), 15604 que trata de receptores, 15605 que trata de segurança, 15606 que trata do *middleware*, 15607 que trata do canal de interatividade e 15608 do guia de operação.

Na transmissão, o SBTVD usa a modulação OFDM, que divide da banda útil do canal em 13 segmentos de 428,5 kHz cada, os quais podem ser agrupados para formar até três camadas diferentes em um processo denominado transmissão hierárquica, onde cada camada pode empregar esquemas de modulação diferentes [48].

O SBTVD assim como o ISDB, tem dois tipos de serviços: *one-seg* e *full-seg*, sendo o primeiro para transmissões de baixa definição destinadas a aparelhos móveis e o segundo para alta definição. A origem dos nomes vem da transmissão divida em 13 segmentos sendo apenas um segmento destes destinado à transmissão com menor qualidade, por isso o nome *one-seg* (um segmento).

6.1 PADRÕES MPEG-2 E MPEG-4 SYSTEMS

Os padrões MPEG-2 e MPEG-4 *Systems* são definidos na parte 1 das normas ISO correspondentes 13818 [22] e 14496 [24]. Nestes padrões são definidas várias ferramentas para agrupar e sincronizar vários sinais de áudio, vídeo e outros dados necessários (como legendas) para formar um fluxo de dados completo para ser armazenado ou como no caso da TV digital, formar um programa completo que será transmitido.

O padrão define o fluxo elementar (*Elementary Stream* - ES), que é o componente básico do MPEG. Juntando vários ES pode ser formado um programa completo com áudio, vídeo, legendas, dados para controle, etc. O ES é usado pelo MPEG para definir a saída de um codificador de áudio ou vídeo. O ES contém apenas um tipo de dados, como por exemplo, só

áudio ou só vídeo. Tais ES podem carregar vários tipos de dados como: áudio codificado, vídeo codificado, dados síncronos ou assíncronos e dados de controle.

Para vídeo e áudio codificados os dados são organizados em unidades de acesso, *access units* (AU) no padrão. Cada AU representa uma unidade fundamental de codificação. Por exemplo, em áudio, uma *access unit* normalmente vai ser um quadro (*frame*) completo codificado.

O padrão MPEG *Systems* oferece uma abordagem em duas camadas de multiplexação:

A primeira camada é usada para garantir a sincronização entre dados de áudio, vídeo e outros dados. Essa camada é chamada de *Packetized Elementary Stream* (PES) e é composta de partes do *Elementary Stream* (ES) divididas em pacotes com um mesmo cabeçalho (*header*), de modo parecido com o modo que dados são transmitidos via Internet pelo protocolo de transporte IP, ou seja, em pacotes cada um com um cabeçalho e dados (*payload)*.

A outra camada é dependente do meio de comunicação que será usado e agrupa os pacotes PES. Para ambientes livres de erros, como no armazenamento de dados localmente, é usado o MPEG-2 *Program* Stream (PS) e para ambientes sujeitos a erros e ruído, como em transmissões de rádio e TV, é usado *Transport Stream* (TS).

Um TS é formado por um ou mais programas. Os ESs de áudio e vídeo são formados por várias AUs. Os dados dos ES são transportados em pacotes PES dentro do TS.

Um pacote PES pode apenas conter um único tipo de ES. São permitidos tanto comprimentos fixos quanto variáveis de pacotes PES. Cada pacote tem um cabeçalho de 6 *bytes*.

O TS foi concebido para comunicação ou armazenamento de um ou mais programas formados por dados codificados de vários tipos em ambientes nos quais erros significantes podem ocorrer. Os erros que podem ocorrer normalmente são devidos a erros em valores de bits ou perdas de pacotes.

Na Figura 6.1 pode-se observar uma representação gráfica do modo como é formado o pacote PS ou TS multiplexando vários pacotes PES.

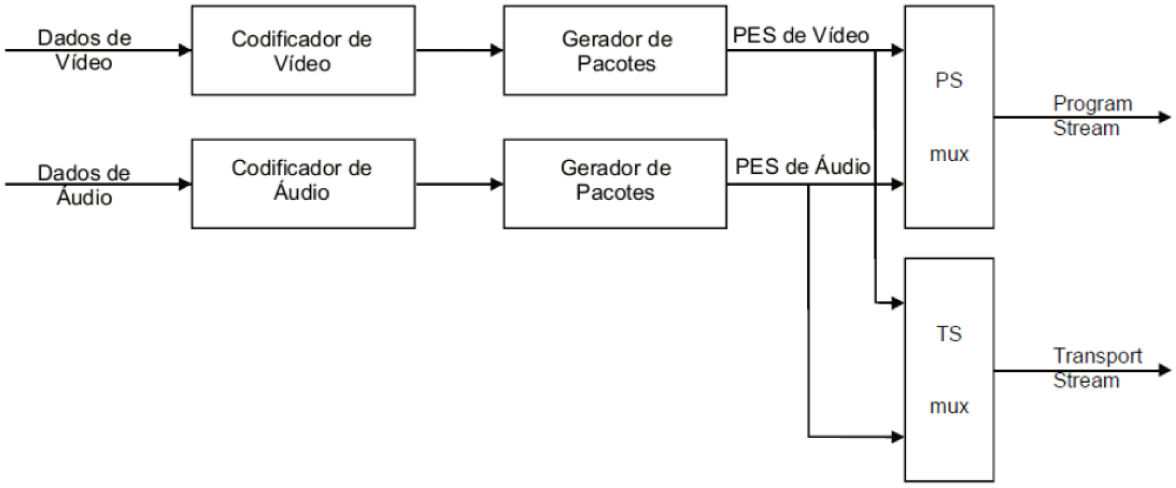


Figura 6.1 – Esquema das camadas de multiplexação do MPEG Systems [22].

O padrão MPEG *systems* não especifica a arquitetura ou implementação dos codificadores, decodificadores, nem dos multiplexadores e demultiplexadores [22]. Apesar disso, as propriedades do *bitstream* acabam impondo alguns requisitos funcionais e de desempenho para os codificadores e multiplexadores. Mesmo assim ainda é mantido um grau de liberdade considerável na hora de se desenvolver um codificador ou multiplexador seguindo o padrão MPEG.

6.1.1 Multiplexação e sincronização dos dados de áudio e vídeo

Para serem transmitidos no sistema de televisão digital os sinais de áudio codificados devem ser agrupados em pacotes e multiplexados com outros sinais de áudio e vídeo codificados, sinais de dados e de informações relacionadas (informações necessárias para a gerência de serviços como autenticação e controle de acesso, segurança de comunicação e registro de uso) que venham a compor o programa.

Todos esses sinais são agrupados em pacotes PES que quando multiplexados irão compor o *Transport Stream* (TS) *MPEG* de acordo com as normas da ABNT [5] e ISO/IEC MPEG-2 e 4 [22][24].

Um pacote TS pode conter diversos tipos de pacotes PES e cada pacote é identificado por um identificador de pacote (PID - *Packet ID*) de 13 bits no cabeçalho do TS. O PID identifica que tipo de dados o pacote TS contém.

Os pacotes TS podem ser de taxa fixa ou variável. Em qualquer um destes casos os *elementary streams* podem ter taxa fixa ou variável. A sintaxe e as limitações semânticas no *stream* são idênticas em cada um destes casos. A taxa do *Transport Stream* é definida pelos valores e localizações dos campos que contém o *clock* de referência do programa (*Program Clock Reference* (PCR)), os quais em geral são campos PCR separados para cada programa.

O PCR é usado para sincronizar o decodificador com o tempo que foi usado para gerar o TS no codificador original. Ele também é usado pelo decodificador para assegurar que o áudio e o vídeo vão estar sincronizados corretamente.

O tamanho dos pacotes TS é de 188 bytes, mas podem ser adicionados bytes extras para correção de erros. O SBTVD usa 204 bytes de tamanho total dos pacotes enviados na transmissão quando é usada correção de erros, 188 bytes do TS mais 16 bytes de dados que podem ser usados para correção de erros e para outras informações que possam ser necessárias para a transmissão [3].

6.1.2 Tabelas de informações específicas de programa (PSI)

Além dos dados de áudio e vídeo o TS também pode carregar sinais para controle da transmissão como as tabelas PSI (*Program Specific Information*). As informações contidas nestas tabelas permitem aos decodificadores demultiplexar corretamente os sinais recebidos. Os programas, os *elementary streams* ou partes destes podem estar embaralhados para acesso condicional, mas a PSI não pode ser embaralhada.

Existem quatro tabelas PSI, que são transportadas dentro do TS:

- Tabela de associação de programa (*Program Association Table - PAT*);
- Tabela de mapeamento de programa (*Program Map Table - PMT*);
- Tabela de acesso condicional (*Conditional Access Table - CAT*);
- Tabela de informação de rede (*Network Information Table - NIT*);

Estas tabelas contêm informações necessárias e suficientes para demultiplexação e apresentação dos programas.

A tabela de associação de programa (PAT) serve para permitir a correspondência entre o número do programa e o PID dos pacotes de TS que carregam as definições desse programa (PMT_PID).

A tabela de mapeamento de programa (PMT) especifica, entre outras informações, quais PIDs, isto é, quais *elementary streams* são associados para formar cada programa, o que permite ao receptor decodificar apenas os elementos necessários para formar o programa específico que vai ser entregue no momento. Esta tabela também indica o PID dos pacotes TS que carregam o *Program Clock Reference* (PCR) para cada programa.

A tabela de acesso condicional deve estar presente se for usado embaralhamento.

A tabela de informação de rede é opcional e seu conteúdo não é especificado no padrão MPEG *Systems* [22].

Dentro do TS, a PSI é classificada em estruturas de tabela como mostradas na Tabela 6.1. Enquanto estas estruturas podem ser pensadas como simples tabelas, elas devem ser segmentadas em seções e inseridas nos pacotes TS, algumas com PIDs predeterminados e outras com PIDs que podem ser escolhidos pelo usuário.

Nome da Estrutura	Tipo de *Stream*	Número PID	Descrição
Tabela de Associação de Programa (PAT)	ITU-T Rec. H.222.0 / ISO/IEC 13818-1	0x00	Associa o número do programa e o PID da PMT
Tabela de Mapeamento de Programa (PMT)	ITU-T Rec. H.222.0 / ISO/IEC 13818-1	Atribuição indicada na PAT	Especifica valores de PID para componentes de um ou mais programas
Tabela de Informação de Rede (NIT)	Privado	Atribuição indicada na PAT	Requerimentos da rede física como frequências FDM, números do *transponder*, etc.

Tabela de Acesso Condicional (CAT)	ITU-T Rec. H.222.0 / ISO/IEC 13818-1	0x01	Associa a pacotes contendo cada de *Entitlement Management Message* (EMM) um único valor PID. EMMs servem para atualizar as opções de assinatura ou direitos de *pay-per-view* por assinante.
Tabela de Descrição do TS	ITU-T Rec. H.222.0 / ISO/IEC 13818-1	0x02	Associa um ou mais descritores da Tabela 2-39 do padrão MPEG-2 Systems [22] para um TS inteiro.

Tabela 6.1 – Tabelas de informações específicas de programa (PSI).

6.1.3 Temporização

A temporização do *Transport Stream* é baseada no relógio de sistema (*System Time Clock – STC*) do codificador. Para garantir uma sincronização adequada durante o processo de decodificação, o relógio do decodificador deve estar sincronizado com o STC do codificador. Para realizar essa sincronia, o codificador insere uma referência temporal *(time stamp*), a partir de um relógio de 27 MHz no TS para cada programa. Este *time stamp* é chamada *Program Clock Reference* (PCR). Usando o PCR, o decodificador gera um *clock* local de 27 MHz sincronizado com o STC do codificador.

6.1.4 Formato dos pacotes TS

Os pacotes TS começam com um prefixo de quatro bytes que contém um identificador de pacote (*Packet ID – PID*) de 13 bits. O PID identifica, fazendo uso das tabelas de informação específica de programa (*Program Specific Information – PSI*), o conteúdo dos dados contidos no pacote TS. Pacotes TS com um dado valor de PID carregam dados de apenas um único *elementary stream*.

Os pacotes TS também podem ser pacotes nulos. Pacotes nulos são usados para o preenchimento de TSs. Eles podem ser inseridos ou removidos pelos processos de remultiplexação e, portanto, a entrega do *payload* de pacotes nulos para o decodificador não pode ser assumida.

Segundo as normas ABNT [5] e ISO/IEC [22], os pacotes TS do sinal multiplexado devem atender ao formato da Figura 6.2, conforme norma [4]. A descrição dos campos da estrutura do pacote TS é a seguinte:

O byte de sincronismo (*Sync byte*) é um campo de tamanho fixo em 8 bits que contém uma palavra de sincronismo. O valor da palavra de sincronismo (*Sync byte*) deve obrigatoriamente ser '0100 0111' (0x47).

O indicador de erro de transporte (*Transport error indicator*) é um *flag* de 1 bit que indica a presença de qualquer erro de bit no pacote TS. Se esta sinalização contiver o valor '1', indica que o pacote TS tem um erro incorrigível de pelo menos um bit.

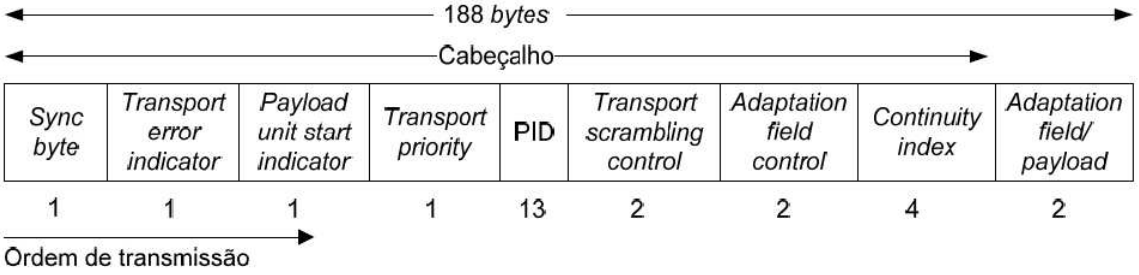


Figura 6.2 – Estrutura do pacote TS e suas seções [5].

O indicador de início (*Payload unit start indicator*) é um *flag* de um bit. O que ele significa depende se os pacotes TS estão carregando pacotes PES ou dados PSI.

Quando o *payload* do pacote TS contém dados de pacotes PES, o *payload unit start indicator* tem o seguinte significado: um valor ‘1’ indica que o *payload* deste TS vai começar com o primeiro *byte* de um pacote PES e um ‘0’ indica que nenhum pacote PES deve começar neste pacote TS. Se o *payload unit start indicator* estiver definido como ‘1’, então um e apenas um pacote PES se inicia neste pacote TS. Isto também se aplicada a *streams* privadas de tipo 6, os tipos de *stream* são definidos na norma [5].

Quando o *payload* do pacote TS contém dados PSI, o *payload unit start indicator* tem o seguinte significado: se o pacote TS carrega o primeiro *byte* de uma seção PSI, o valor do *payload unit start indicator* deverá ser ‘1’, indicando que o primeiro *byte* do *payload* deste pacote TS carrega o *pointer_field* (campo definido na seção sobre o PSI). Se o pacote TS não carrega o primeiro *byte* de uma seção PSI, o valor do *payload unit start indicator* deverá ser ‘0’, indicando que não há nenhum *pointer_field* no *payload*. Isto também se aplica a *streams* privados do tipo 5.

O *Transport priority* é um *flag* de um bit que indica a prioridade de transporte entre os pacotes com o mesmo PID. O pacote com valor ‘1’ recebe prioridade.

O PID (*Packet Identifier*, identificador de pacotes) é um campo de 13 bits que identifica o tipo de dados armazenados no *payload* do pacote. Os tipos de dados do *payload* estão definidos na norma [4].

O *Transport scrambling control* (controle de embaralhamento de transporte) é um campo de 2 bits que identifica o modo de embaralhamento (*scrambling mode*) do *payload* para o pacote TS. O cabeçalho do pacote TS e o campo de adaptação, quando presente, não devem ser embaralhados. No caso de um pacote nulo, o valor do campo sempre será ‘00’.

O *Adaptation field control* (controle do campo de adaptação) é um campo de 2 bits que indica se o cabeçalho do pacote TS é seguido de um campo de adaptação e/ou payload. O campo de adaptação/*payload* deve obrigatoriamente estar de acordo com a norma [4].

O *Continuity index* (índice de continuidade) é um campo que especifica a sucessão de pacotes de TS com o mesmo PID. O valor deste campo deve começar com ‘0000’ e deve ser incrementado em 1. Este campo vai retornar ao valor ‘0000’ quando alcançar o valor ‘1111’. Porém, deve ser assegurado que o mesmo pacote de TS será transmitido no máximo duas vezes dentro de uma fila e que no caso de repetição o valor deste campo não deve ser incrementado.

O *adaptation field* (campo de adaptação) deve atender à ISO/IEC 13818-1 [22]. Ele é composto dos seguintes campos: *adaptation Field length* e *discontinuity indicator*.

O *adaptation field length* é um campo de 8 bits que especifica o número de *bytes* no campo de adaptação que vem em seguida ao *adaptation field length*. O valor zero é para inserir um único *byte* de preenchimento em um pacote TS. Quando do o valor do *adaptation field control* for '11', o valor do *adaptation field length* deve estar na faixa de 0 à 182. Quando o valor do *adaptation field control* for '10', o valor do *adaptation field length* deve ser 183. Para pacotes TS que transportam pacotes PES, o preenchimento é necessário quando não houver dados de pacotes PES suficientes para preencher completamente os bytes do *payload* do pacote TS. O preenchimento é realizado definindo um campo de adaptação maior que a soma dos comprimentos dos elementos de dados contidos nele, para que os bytes do *payload* restantes após o campo de adaptação acomodem exatamente os dados disponíveis de pacotes PES. O espaço extra no campo de adaptação é completado com bytes de preenchimento.

O *discontinuity indicator* é um campo de 1 bit que quando tem valor '1' indica que o estado de descontinuidade é verdadeiro para o pacote TS atual. Quando o valor for '0' ou não estiver presente, o estado de descontinuidade é falso. O indicador de descontinuidade é usado para indicar dois tipos de descontinuidades, descontinuidades no tempo base do sistema e descontinuidades do contador de continuidade (*continuity_counter).*

6.1.5 Formato dos pacotes PES

Um pacote PES é formado por um cabeçalho e em seguida os dados do pacote. Estes pacotes são inseridos dentro de pacotes de TS. O primeiro *byte* de cada cabeçalho de pacote PES fica localizado no primeiro lugar disponível no *payload* de um pacote TS.

O cabeçalho do pacote PES começa com um código prefixo de 32 bits que também identifica o *stream* ou tipo de *stream* ao qual os dados do pacote pertencem. O cabeçalho dos pacotes PES pode conter *time stamps* de decodificação e apresentação (DTS e PTS). O cabeçalho também possui outros campos opcionais. O campo com os dados do pacote PES contém um número variável de *bytes* adjacentes de um *elementary stream*.

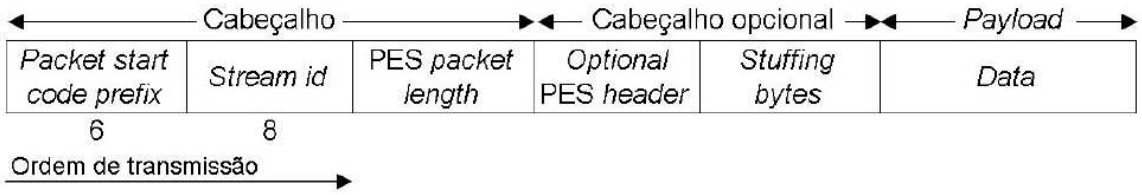


Figura 6.3 – Estrutura do pacote PES e suas seções [5].

A descrição dos componentes do pacote PES é a seguinte:

- O *Packet start code prefix* é um código de três bytes que representa o começo do pacote de PES e deve ser fixado em 0x000001;
- O *Stream id* tem tamanho de um byte e deve ser usado para identificar o tipo e o número do *elementary stream* (sinais codificados; ele deve ser válido para outros sinais). O tipo e o número do *elementary stream* devem estar de acordo com a norma [5];
- O campo PES *packet length* indica o número de bytes no pacote PES após este campo. Se ele tiver valor '0' indica que o tamanho do pacote PES não deve obrigatoriamente ser especificado e não deve ter limites. O valor '0' só é permitido para pacotes PES quando o *payload* for composto por *elementary streams* de vídeo.
- O cabeçalho opcional contém:
  - *Optional PES header,* que deve estar de acordo com a ISO/IEC 13818-1;
  - *Stuffing bytes* ou *bytes* de preenchimento devem ter valor fixo em 0xFF e não devem exceder 32 bytes em comprimento.

6.2 PADRÃO MPEG-4 PARA ÁUDIO (AAC)

O AAC foi definido primeiro no padrão MPEG-2 [23] e melhorado no MPEG-4, a parte de áudio sendo definida na parte 3 do padrão ISO/IEC 14496-3 [25].

O padrão MPEG-4 para áudio integra vários tipos diferentes de codificação de áudio que podem ser usados para diversas aplicações. O padrão é baseado em objetos e tem diversas ferramentas que não são necessariamente relacionadas entre si. São os perfis do MPEG-4 para áudio que especificam quais destas ferramentas serão usadas em conjunto para cada aplicação.

Os conjuntos de ferramentas do MPEG-4 são separados em categorias como uma para codificação de voz, outra para música e trilhas sonoras em geral (chamados no padrão de áudio natural). Também são definidas na norma ferramentas para codificação paramétrica e proteção contra erros, entre outras.

No SBTVD é usado o MPEG-4 AAC, que está definido dentro das ferramentas para codificação áudio natural chamadas *general audio coding tools*, definidas na subparte *General Audio* (GA) do padrão MPEG-4 Áudio. Esta subparte foi desenvolvida para codificação de áudio a partir de 6kbits/s por canal até áudio com qualidade para transmissão em 64kbits/s ou mais por canal. O material codificado com MPEG-4 pode ser representado ou por um único conjunto de dados, como no MPEG-1 e MPEG-2 *Audio*, ou por vários subconjuntos que permitem a decodificação em níveis diferentes de qualidade dependendo do número de subconjuntos que estiverem disponíveis no lado do decodificador (escalabilidade da taxa de bits) [25].

O AAC é divido em perfis que definem a complexidade do codificador e as ferramentas que serão aplicadas. Existem três perfis principais para o AAC: principal (*main*), de baixa complexidade (*low complexity - LC*) e de taxa de amostragem escalável (*Scalable Sample Rate - SSR*). O perfil mais usado é o de baixa complexidade, que não inclui as ferramentas de controle de ganho e predição e usa uma ordem menor de filtro na ferramenta *Temporal Noise Shaping (TNS).*

O AAC também tem extensões, que são usadas normalmente junto com o perfil LC, para melhorar a codificação em taxas de bits menores, conhecidas como codificação avançada de alta eficiência, *High-Efficiency Advanced Audio Coding (HE-AAC).* O HE-AAC também é conhecido pelo nome comercial *aacPlus*.

O perfil HE-AAC versão 1 combina o AAC com a ferramenta de replicação espectral de banda (*Spectral Band Replication – SBR*) para melhorar a qualidade do som quando são necessárias taxas de bits menores. Esta ferramenta permite a reconstrução das faixas de frequência mais altas do espectro a partir das mais baixas, permitindo a transmissão apenas das frequências menores. Esta reconstrução é feita pelo decodificador a partir das frequências mais baixas do sinal e de informações laterais que foram extraídas no codificador referentes ao envelope espectral das altas frequências. Essas informações extras usam uma pequena quantidade de dados e contém uma representação paramétrica da banda de frequências altas. A taxa de dados necessária é bem menor do que se fosse codificada a faixa de frequências altas diretamente.

A Figura 6.4 mostra um diagrama de blocos com as principais ferramentas que fazem parte de um codificador padrão MPEG-4 AAC. Estas ferramentas serão detalhadas nos itens seguintes.

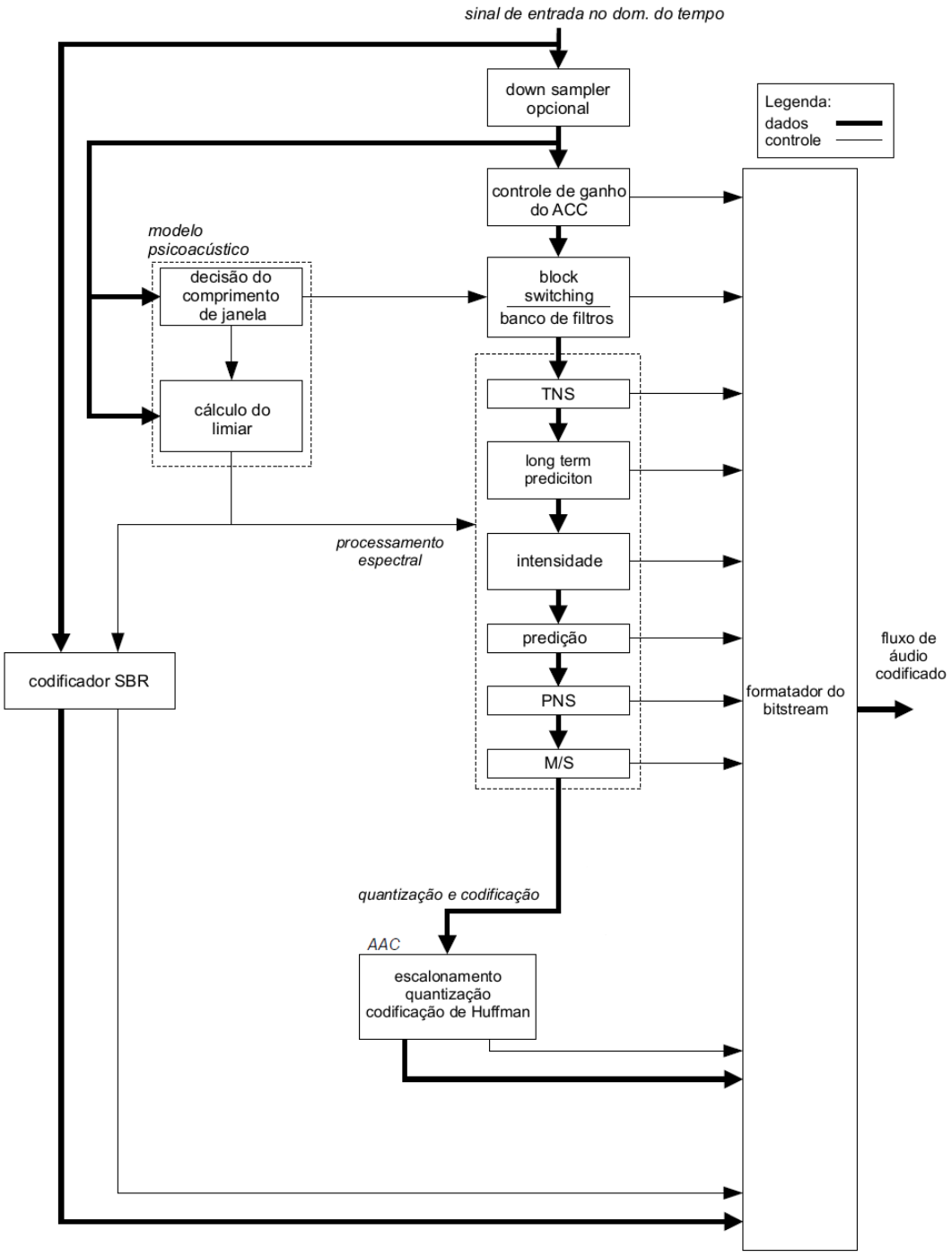


Figura 6.4 – Diagrama de blocos de um codificador MPEG-4 HE-AAC v1 (adaptado de [25]) .

No HE-AAC versão 2, além da SBR, é usada a ferramenta de estéreo paramétrico (*Parametric Stereo – PS*), que melhora a qualidade de sinais estéreo mesmo em taxas de bit muito baixas, transmitindo um canal só e informação lateral para reconstrução do par estéreo. Esse modo normalmente é usado no áudio das transmissões no modo *one-seg* do SBTVD, que é o modo destinado a aparelhos móveis, onde é importante reduzir bastante a taxa de bits.

Os perfis do AAC usados no SBTVD para o serviço *full-seg* são:

- AAC LC nos níveis L2 (dois canais) e L4 (multicanal);
- HE-AAC versão 1, níveis L2 e L4;

Para o serviço *one-seg*, é permitido apenas o HE-AAC versão 2, nível L2, com no máximo dois canais.

O perfil e o nível do codificador MPEG-4 AAC devem obrigatoriamente ser sinalizados conforme ABNT NBR 15602-3 [5] e ABNT NBR 15603-2.

Na Figura 6.5 pode-se observar um diagrama de quais blocos pertencem a cada extensão.

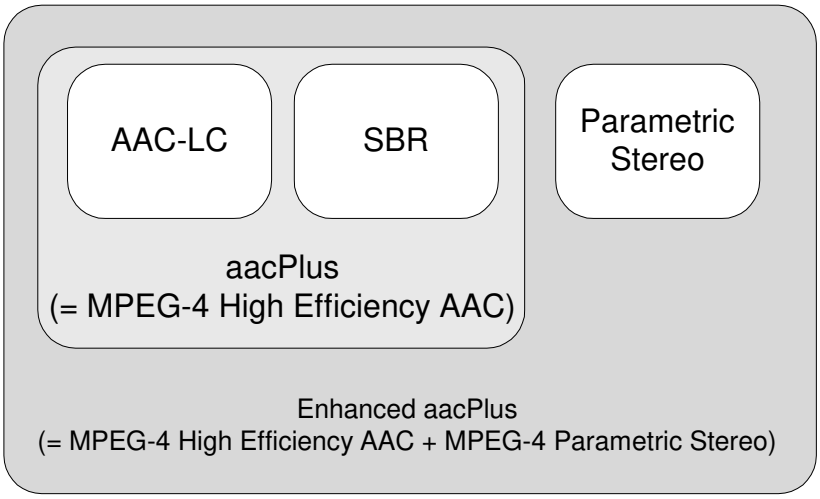


Figura 6.5 – AAC e suas expansões [1].

6.2.1 Análise do Sinal pelo AAC

O AAC usa a transformada discreta de cosseno modificada (MDCT) para a conversão dos sinais de entrada do domínio do tempo para a frequência. A MDCT é uma transformada baseada na DCT com a diferença de usar sobreposição. A MDCT usa blocos de amostras sobrepostas com

metade do bloco anterior e outra metade do posterior. Esta sobreposição é muito útil para codificação de áudio, pois permite evitar artefatos causados pelas descontinuidades nas bordas das janelas ou blocos usados na transformada. Esta técnica de sobreposição é conhecida como *Time-Domain Aliasing Cancellation* (TDAC).

Os coeficientes espectrais da transformada, $X_{i,k}$ são definidos por:

$$X_{i,k} = 2 \cdot \sum_{n=0}^{N-1} z_{i,n} cos\left(\frac{2\pi}{N}(n + n_0)\left(k + \frac{1}{2}\right)\right) \; para \; 0 \leq k \leq N/2 \qquad \textbf{(6.1)}$$

onde z_{in} é a sequência de entrada após o janelamento, n é o índice das amostras, k é o índice dos coeficientes espectrais, i é o índice de bloco, N é o tamanho de uma janela da transformada baseado no valor de *window_sequence* e $n_0 = (N/2 + 1)/2$.

No codificador, o tamanho da janela de análise N para uma janela da transformada MDCT é uma função do elemento de sintaxe chamado *window_sequence* e é definido do seguinte modo:

$$N = \begin{cases} 2048, \text{se } window\_sequence = \text{ONLY\_LONG\_SEQUENCE(0x0)} \\ 2048, \text{se } window\_sequence = \text{LONG\_START\_SEQUENCE(0x1)} \\ 256, \text{se } window\_sequence = \text{EIGHT\_SHORT\_SEQUENCE(0x2)(8 vezes)} \\ 2048, \text{se } window\_sequence = \text{LONG\_STOP\_SEQUENCE(0x3)} \end{cases}$$

O codificador alterna entre dois comprimentos de bloco para a transformada, de acordo com o número de amostras por janela: bloco longo e bloco curto. A janela longa na maioria dos casos tem 1024 amostras ou coeficientes e a janela curta 128 amostras. A transformada é feita com duas vezes o tamanho do quadro, isto é 2048 amostras para a janela longa e 256 para curta.

O bloco curto é usado para melhorar a resolução em transientes e evitar o espalhamento de ruído, que pode gerar um artefato conhecido como pré-eco. Como não é muito eficiente codificar todo o sinal com blocos curtos, o sinal de entrada é analisado e é alternado o tamanho de bloco quando é detectado um transiente. Para alternar entre os dois tipos de janela, o codificador usa janelas de transição chamadas *LONG_START_WINDOW* e *LONG_STOP_WINDOW*.

A Tabela 6.2 lista os tipos de janelas usadas no AAC, especifica o comprimento em número de amostras da transformada correspondente e mostra um esquema do formato da janela.

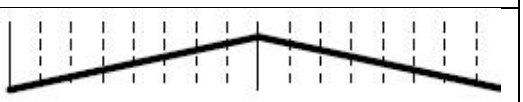
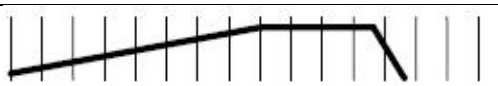
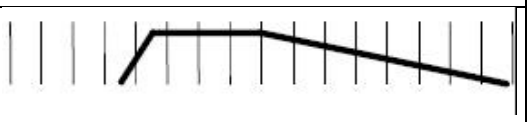
Janela	*num_swb*	Amostras	Aparência
LONG_WINDOW	49	1024/960	
SHORT_WINDOW	14	128/120	
LONG_START_WINDOW	49	1024/960	
LONG_STOP_WINDOW	49	1024/960	

Tabela 6.2 – Janelas usadas no padrão AAC [25].

Na sintaxe do AAC são usados os campos *window_sequence* e *window_shape* para armazenar as informações de controle que serão necessárias para escolha e troca do tamanho e tipo da janela. A Tabela 6. mostra os tipos de *window_sequence*.

O campo *window_sequence* é formado por dois bits e indica qual sequência (tamanho de bloco) será usada.

O campo *window_shape* tem um bit e indica qual o tipo de janela que será usado.

Para *window_shape* = 1, é usada a janela derivada de Kaiser-Bessel (KBD) de acordo com[23]:

$$W_{KBD\_LEFT,N}(n) = \sqrt{\frac{\sum_{p=0}^{n}[W'(p,\alpha)]}{\sum_{p=0}^{N/2}[W'(p,\alpha)]}}, para\ 0 \le n \le \frac{N}{2} \quad \textbf{(6.2)}$$

$$W_{KBD\_RIGHT,N}(n) = \sqrt{\frac{\sum_{p=0}^{N-n-1}[W'(p,\alpha)]}{\sum_{p=0}^{N/2}[W'(p,\alpha)]}}, para\ \frac{N}{2} \le n \le N \quad \textbf{(6.3)}$$

onde W' (função da janela Kaiser-Bessel) é definida da seguinte maneira:

$$W'(n,\alpha) = \frac{I_0\left[\pi\alpha\sqrt{1.0 - \left(\frac{n-N/4}{N/4}\right)^2}\right]}{I_0[\pi\alpha]}, para\ 0 \le n \le \frac{N}{2} \quad \textbf{(6.4)}$$

e $I_0[x]$ é a função de Bessel modificada de ordem zero do primeiro tipo:

$$I_o[x] = \sum_{k=0}^{\infty}\left[\frac{\left(\frac{x}{2}\right)^k}{k!}\right]^2 \quad \textbf{(6.5)}$$

$\alpha = kernel\ window\ alpha\ factor, \qquad \alpha = \begin{cases} 4\ para\ N = 2048 \\ 6\ para\ N = 256 \end{cases}$

E para *window_shape* = 0, uma janela seno é aplicada de acordo com:

$$W_{SIN\_LEFT,N}(n) = sen\left(\frac{\pi}{N}(n+\frac{1}{2})\right)\ para\ 0 \le n \le \frac{N}{2} \quad \textbf{(6.6)}$$

$$W_{SIN\_RIGHT,N}(n) = sen\left(\frac{\pi}{N}(n+\frac{1}{2})\right)\ para\ \frac{N}{2} \le n \le N \quad \textbf{(6.7)}$$

O comprimento da janela N pode ser 2048 ou 256 para as janelas KBD e seno.

Para todos os tipos de sequências de janela (*window_sequences*) o formato da janela (*window_shape*) da metade esquerda da primeira janela de transformada é determinado pelo formato da janela do bloco anterior, de acordo com a equação:

$$W_{LEFT\_N}(n) = \begin{cases} W_{KBD\_LEFT,N}(n), se\ window\_shape\_previous\_block == 1 \\ W_{SIN\_LEFT,N}(n), se\ window\_shape\_previous\_block == 0 \end{cases} \quad \textbf{(6.8)}$$

onde *window_shape_previous_block* é o formato da janela do bloco anterior (i-1).

Para o primeiro bloco do *bitstream* a ser decodificado o formato da janela da esquerda e da metade direita da janela são idênticos.

Para melhorar a qualidade da codificação em transientes, além de alternar o tamanho de bloco, também pode ser aplicada a ferramenta TNS, pois apenas a troca do tamanho de bloco não é sempre suficiente para evitar problemas com o pré-eco.

Após estar no domínio da frequência o sinal passará pelo modelo psicoacústico do AAC que irá extrair os limiares para o ruído de quantização permitido no quantizador.

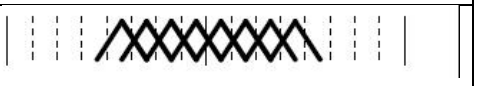
Valor *window_ shape*	*window_sequence*	Nº de janelas	Aparência
0	ONLY_LONG_SEQUENCE = LONG_WINDOW	1	
1	LONG_START_SEQUENCE = LONG_START_WINDOW	1	
2	EIGHT_SHORT_SEQUENCE = 8 * SHORT_WINDOW	8	
3	LONG_STOP_SEQUENCE = LONG_STOP_WINDOW	1	

Tabela 6.3 – Sequência de janelamento no AAC[25].

6.2.2 Análise para fontes estéreo e multicanal

Na sintaxe do *bitstream* do AAC, os canais e pares de canais são separados em elementos de sintaxe, um canal mono comum é um elemento *single channel element – SCE*, um par estéreo é um *channel pair element – CPE* e o canal LFE é um *lfe_element*. Um CPE é composto por duas *streams* de canal individual e informações extras para codificação do par. Os dois canais podem também compartilhar informação lateral.

Uma fonte 5.1 é composta pela sequência de elementos: *SCE, CPE, CPE, LFE*, representando respectivamente os canais central, par estéreo frontal, par estéreo traseiro e o canal LFE. Existe também um elemento para canal de acoplamento, o *coupling channel element – CCE* [25].

Na sintaxe do AAC, o *lfe_channel_element* é definido como um elemento *individual_channel_stream* padrão, isto é, igual a um *single channel element*. Portanto, a decodificação pode ser feita usando o processo padrão para decodificar um *single channel element*.

Para uma implementação mais eficiente em termos de taxa de bits e complexidade de *hardware* do decodificador LFE, várias restrições se aplicam às opções usadas para a codificação deste elemento [23]:

- O campo de formato da janela (*window_shape)* é sempre zero, isto é uma janela seno.
- O campo de sequência de janela é sempre zero (*ONLY_LONG_SEQUENCE*).
- Apenas os 12 coeficientes espectrais menores de qualquer LFE podem não ter valor zero.
- TNS não é usado.
- Nenhuma predição é usada.

A presença de canais LFE depende do perfil usado.

6.2.3 Codificação conjunta de pares estéreo

O AAC também usa a codificação conjunta de pares estéreo (*joint stereo)* que também é aplicada para fontes multicanal.

Este módulo inclui duas técnicas: *mid/side (M/S) stereo* (também conhecida como codificação de soma e diferença) e *intensity stereo*, que foram explicadas com mais detalhes no capítulo 0. Nesta parte será abordado como elas são aplicadas ao AAC.

6.2.3.1 M/S Stereo

O M/S é aplicado às áreas de menor frequência do sinal. Substituem-se então os canais direito e esquerdo pela soma destes dois canais, que se torna o canal meio (*mid - M*) e pela diferença, que se torna o canal lateral (*side –S*). Este módulo também pode ser habilitado ou não de acordo com as características do sinal de entrada. No AAC o M/S é aplicado em cada par de canais que esteja arranjado simetricamente como um par direito/ esquerdo para o ouvinte em um sinal multicanal.

A decisão se será usado M/S ou se serão codificados os canais direito e esquerdo separadamente, é feita por quadro e em frequência com base nas bandas de fator de escala. Para cada banda o seguinte processo de decisão é usado:

1. São calculados os limiares não apenas para os canais direito(R) e esquerdo(L), mas também os limiares para os canais M = (L+R) / 2 e S = (L-R) / 2. Para os limiares de M e S, ao invés de usar a tonalidade para o limiar de M ou S, é usado o valor mais tonal do cálculo feito em L e R em cada banda de cálculo do limiar, e se procede com o modelo psicoacústico para M e S a partir das energias de M e S e o mínimo dos valores de L ou R para $C(\omega)$ em cada banda de cálculo do limiar, Os valores que são fornecidos para o processo de controle de imagem são identificados na seção de informação do modelo psicoacústico como *en(b)* (a energia de espalhamento normalizada) e *nb(b)*, o limiar bruto.
2. Os limiares brutos para M, S, L e R, e a energia de espalhamento para M, S, L e R, são todos levados para um "processo de controle de imagem". Os limiares ajustados daí resultantes são inseridos como os valores para *cb(b)* no modelo psicoacústico para um processamento adicional.
3. Todos os limiares finais para os canais M, S, L e R, protegidos e adaptados para as bandas do codificador são aplicados diretamente ao espectro apropriado através da quantização dos valores espectrais reais de L, R, M e S com os limiares adequados.
4. O número de bits necessários para codificar os canais como M/S e o número de bits necessário para codificar os canais esquerdo e direito individualmente é calculado.
5. O método que precisar do menor número de bits será usado em cada banda do codificador e uma máscara estéreo é definida de acordo.

6.2.3.2 Intensity Stereo

O *intensity stereo* é direcionado ás regiões de frequências mais altas do sinal. Este método é usado para explorar irrelevâncias existentes entre os canais nas faixas de frequência mais altas. Como foi visto no item 5.2.3, as componentes de alta frequência do som são percebidas pelo ouvido através da análise de seus envelopes de energia no tempo e não da análise do sinal inteiro. Este fato possibilita que o codificador compartilhe um mesmo conjunto de valores espectrais

entre vários canais nestas faixas de frequência. No AAC, o *intensity stereo* pode envolver dois mecanismos: O primeiro, chamado de *intensity stereo coding,* é aplicado às componentes de maior frequência e consiste em substituir o sinal do canal direito pela soma dos canais multiplicados pela raiz quadrada da energia da sub-banda do canal direito dividida pela energia de sub-banda do canal esquerdo e o sinal do canal esquerdo por zero. O segundo é o canal de acoplamento do AAC onde a energia espectral seria compartilhada por mais canais além do par estéreo.

6.2.4 *Spectral Band Replication* - SBR

A ferramenta SBR é usada nos perfis HE-AAC versão 1 e 2. Sua função é melhorar a qualidade do áudio quando forem necessárias taxas de bits mais baixas.

Quando é usada a ferramenta SBR apenas metade da faixa de frequências precisa ser codificada e as faixas de frequência mais altas serão reconstruídas no decodificador a partir das mais baixas usando dados de controle que foram extraídos no codificador.

O codificador HE-AAC funciona como um sistema com taxa de amostragem dupla, onde o codificador AAC opera com metade da taxa de amostragem do SBR. Com isso o banco de filtros do codificador AAC tem maior resolução e consequentemente meios melhores para aproveitar os efeitos do mascaramento auditivo [15].

No codificador, a SBR faz um pré-processamento do sinal, extraindo parâmetros de controle a partir do sinal original que ainda está com todas as faixas de frequências presentes. Isto é feito de modo que seja possível que a reconstrução das bandas de frequência altas seja perceptualmente o mais parecido possível com a faixa de frequências altas original.

O codificador principal, neste caso o AAC, vai codificar apenas a faixa de frequências mais baixas do sinal original até uma frequência de corte e o resto das frequências serão reconstruídas no decodificador usando os dados de controle extraídos do sinal original.

A maioria dos dados de controle é usada para fazer uma representação do envelope espectral do sinal. Os dados do envelope espectral têm resolução no tempo e frequência variáveis para que seja possível controlar o processo SBR da melhor forma possível, evitando assim aumentar muito a taxa de bits necessária para a codificação.

A taxa de bits dos dados de controle varia de acordo com ajustes no codificador, mas em geral fica entre 1 à 3 kbits/s por canal de áudio, que é um valor bem menor do que seria necessário para codificar a banda de frequências altas inteira com qualquer codificador tradicional de forma de onda.

Um diagrama de blocos básico de como funciona o SBR no codificador pode ser visto na Figura 6.6.

A técnica SBR é baseada na ideia de que existe uma grande correlação entre as características da faixa de frequências alta de um sinal com as características da faixa de frequências baixa do mesmo sinal, que serão chamadas respectivamente de banda alta e banda baixa do sinal. Em um sinal com uma série forte de harmônicas que chegam até a frequência de corte naturalmente se assume que ele consiste da mesma série de harmônicas na faixa de frequências alta, mesmo que não seja tanto quanto na faixa menor. Um sinal que tenha características de ruído na faixa de frequências baixa é da mesma maneira assumido que vai conter ruído na banda alta. Este regra geralmente serve para estimar a maioria das faixas de frequências altas dos sinais. Apesar de existirem sinais que desviam deste modelo, o SBR também tem métodos para lidar com estas exceções. Filtragem inversa, soma de ruído aditiva e regeneração senoidal são ferramentas que melhoram sinais que tem menos correlação entre as características das bandas baixa e alta.

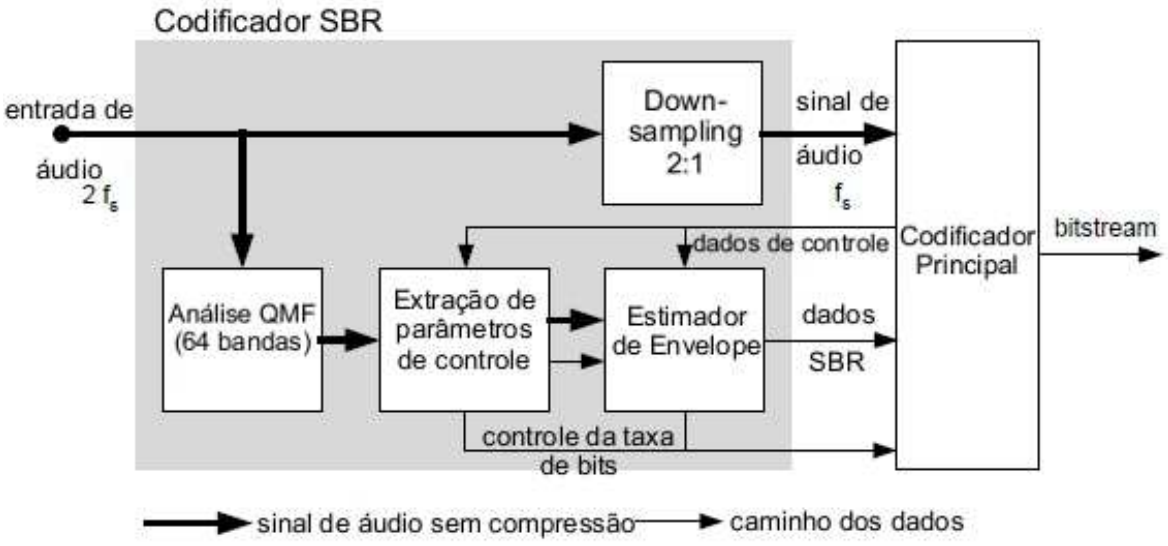


Figura 6.6 – Diagrama de blocos de um codificador SBR [15]

Como os dados SBR são transmitidos como dados auxiliares no *bitstream* do AAC, o sinal com SBR costuma ser compatível com decodificadores AAC que não possuam SBR. Apesar da queda na qualidade por não ter a ferramenta de reconstrução das frequências que foram eliminadas no codificador, ainda será possível decodificar o sinal, ainda que limitado em banda, com um decodificador AAC sem SBR.

Em um codificador SBR, cujo diagrama de blocos é mostrado na Figura 6.7, o sinal de entrada passa primeiro por um *downsampler*, que passa para o codificador principal um sinal no domínio do tempo com metade da taxa de amostragem do sinal de entrada. O sinal de entrada é alimentado em paralelo para um banco de filtros de análise QMF de 64 canais. As saídas do banco de filtros são sinais de sub-banda com valores complexos. Estes sinais são processados por um estimador de envelope e vários detectores. As saídas dos detectores e do estimador de envelope dão forma ao fluxo de dados SBR. Estes dados são então codificados usando codificação de entropia e, no caso de sinais multicanal, a codificação explora as redundâncias entre eles. Os dados SBR codificados e um sinal de controle da taxa de bits são então fornecidos ao codificador principal que faz a multiplexação do fluxo de dados SBR com o *bitstream*.

No decodificador o *bitstream* recebido é dividido em duas partes: o *bitstream* do codificador principal e o fluxo de dados SBR. O *bitstream* principal é decodificado pelo codificador principal e o sinal de áudio decodificado é encaminhado junto com fluxo de dados SBR para o decodificador SBR. O sinal de áudio principal, amostrado com metade da frequência do sinal original, é primeiro filtrado no banco de filtros de análise QMF. O banco de filtros divide o sinal no domínio do tempo em 32 sinais de sub-banda. Estes sinais de sub-banda na saída do filtro são de valor complexo e, portanto super amostrados por um fator de dois se comparado com um banco QMF normal.

Os sinais complexos de sub-banda obtidos do banco de filtros são processados na parte de geração de altas frequências para obter um conjunto de sinais de sub-banda de faixa de frequência alta. Esta geração é feita selecionando sinais de sub-banda da banda baixa, de acordo com regras específicas, para conversão em sinais de banda alta.

A tonalidade nos sinais normalmente é mais pronunciada na banda baixa que na alta, por isso é aplicada filtragem inversa nos sinais de sub-banda gerados. Esta filtragem é realizada através de filtragem *in-band* dos sinais de valor complexo usando filtros FIR adaptativos de ordem baixa e valor complexo. Os coeficientes do filtro são determinados através de uma análise

da banda baixa junto com sinais de controle extraídos do fluxo de dados SBR. Os sinais de banda alta gerados vão então ser processados pelo bloco que faz o ajuste do envelope.

A representação do envelope da banda alta é a maior e mais importante parte do fluxo de dados SBR. Esta representação é usada para ajustar a energia da banda alta que acabou de ser gerada. O bloco de ajuste do envelope primeiro faz uma estimativa da energia dos sinais da banda alta. Uma estimativa precisa é possível por que o sinal está sendo representado por sub-bandas com valores complexos. Com base no envelope estimado e na representação do envelope extraída do fluxo de dados, a energia da banda alta é ajustada.

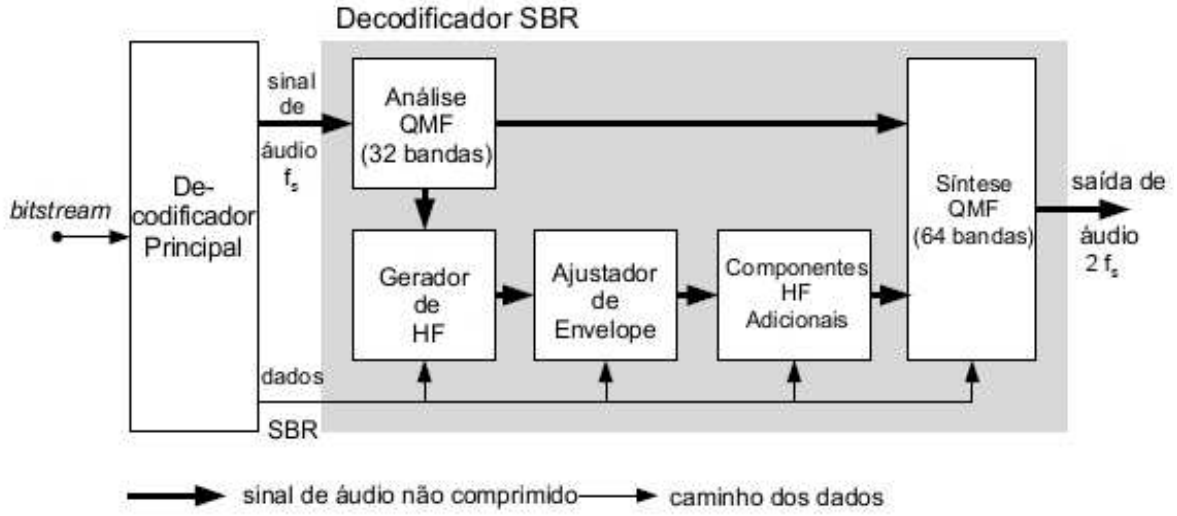


Figura 6.7 – Diagrama de blocos do decodificador SBR [15].

O codificador AAC adiciona um cabeçalho SBR que contém informações como a faixa de frequências SBR e sinais de controle. No decodificador, para o SBR ser decodificado, o cabeçalho SBR deve estar presente. Se não houver cabeçalho SBR, o decodificador SBR apenas faz superamostragem e ajustes de atraso. Em aplicações para transmissões contínuas, os dados SBR com um cabeçalho SBR são tipicamente enviados com uma frequência de dois por segundo. Além disso, uma parte com cabeçalho SBR pode ser inserida a qualquer momento, se uma mudança de acordo com o programa for necessária.

Cada elemento sintático do AAC (SCE, CPE) que será melhorado com SBR vai ter um elemento de preenchimento SBR inserido após o elemento. Os elementos LFE vão ser

decodificados de acordo com o procedimento padrão do AAC, mas devem ter seu atraso ajustado e ser re-amostrados para sua taxa de amostragem coincidir com a taxa de amostragem de saída.

Na Tabela 6. pode ser visto um exemplo da estrutura dos elementos de sintaxe para uma configuração de canais 5.1 com SBR sendo usado.

<SCE><FIL <EXT_SBR_DATA(SCE)>>	Canal central
<CPE><FIL <EXT_SBR_DATA(CPE)>>	Canais esquerdo e direito frontais
<CPE><FIL <EXT_SBR_DATA(CPE)>>	Canais esquerdo e direito traseiros
<LFE>	*Subwoofer*
<END>	Marca o fim do bloco de dados

Tabela 6.4 – Configuração de canais 5.1 com elemento SBR [25].

6.2.5 *Parametric Stereo* - PS

A ferramenta *Parametric Stereo* (PS) é usada no perfil HE-AAC versão 2 em conjunto com a SBR. A ferramenta PS é usada no codificador para extrair uma imagem paramétrica do sinal estéreo e no decodificador para reconstruir um sinal estéreo a partir de um sinal mono usando parâmetros desta imagem que foi extraída no codificador. Esta ferramenta pode ser usada em conjunto com qualquer codificador mono. O funcionamento geral da ferramenta PS foi explicado em detalhes no item 5.2.4.

Com essa ferramenta é possível reduzir mais ainda a taxa de bits necessária, pois só são codificados um canal mono e dados auxiliares para reconstrução da imagem estéreo que usam no máximo 9 kbits/s em qualidade máxima [25].

No decodificador HE-AAC v2, quando a PS é usada em conjunto com a SBR, um sinal de áudio de um canal é transmitido com AAC+SBR e a ferramenta PS é usada pra reconstruir um sinal estéreo com dois canais a partir deste sinal mono, O elemento do *bitstream ps_data()* carrega as informações que a ferramenta PS precisa e este elemento é carregado pelo *sbr_extension()* do *bitstream* SBR.

O uso da extensão PS para o HE-AAC é sinalizada implicitamente no *bitstream*. Por isso, se um *sbr_extension()* contendo o parâmetro *bs_extension_id* igual a *EXTENSION_ID_PS* for encontrado na parte SBR do *bitstream*, isto indica que existem dados PS e um decodificador que

suporte a combinação de SBR e PS (HE-AAC v2) vai usar a ferramenta PS para gerar um sinal de saída estéreo.

Se não for encontrado um elemento de dados PS (*ps_data()*) na parte SBR de um *bitstream* HE-AAC mono, o sinal mono normal é gerado pela ferramenta SBR e mapeado para um sinal de saída estéreo onde o canal direito e esquerdo vão ambos conter o mesmo sinal mono.

6.2.6 *Temporal Noise Shaping* - TNS

A ferramenta *Temporal Noise Shaping* (TNS) foi desenvolvida para melhorar a codificação quando existem transientes onde é possível ocorrer problemas com pré-eco devido ao fato de que o ruído de quantização vai ser distribuído uniformemente dentre de cada janela do banco de filtros, onde o limiar de mascaramento real dependente do tempo pode variar consideravelmente dentro deste período de tempo. A ferramenta TNS lida com esta questão permitindo realizar uma formatação temporal mais precisa do ruído de quantização d codificador.

A TNS realiza predição no domínio da frequência para melhorar a resolução temporal do codificador, para isto é aplicado um filtro *noise shaping* (formatador de ruído) que usa codificação preditiva linear (LPC). A filtragem não é aplicada em todo o espectro, o AAC só ativa a filtragem TNS em faixas de frequência onde for necessário. O TNS nunca é ativado para o canal LFE.

A TNS é aplicada em uma janela por vez da transformada. Os seguintes passos são executados para aplicar a ferramenta TNS em uma janela de dados espectrais [23]:

- Uma faixa de frequências alvo para a TNS é escolhida. Uma escolha adequada é cobrir uma faixa de frequências a partir de 1,5 kHz até a maior banda de fatores de escala possível com um filtro. O parâmetro que define o número máximo de faixas TNS (TNS_MAX_BANDS) depende do perfil e da taxa de amostragem.
- Em seguida, um cálculo de codificação preditiva linear (LPC) é realizado nos coeficientes espectrais da MDCT correspondentes à faixa de frequências alvo escolhida. Para uma melhor estabilidade, coeficientes correspondentes a frequências abaixo de 2.5 kHz podem ser excluídos deste processo. Procedimentos LPC padrão como são conhecidos de processamento de fala podem ser usadas

para o cálculo da LPC, como por exemplo, o algoritmo de Levinson-Durbin. O cálculo é feito para a maior ordem permitida do filtro *noise shaping* (parâmetro TNS_MAX_ORDER). Este valor também depende do perfil do codificador.

- Como resultado do cálculo da LPC, são obtidos o ganho esperado de predição g_p e também os coeficientes de reflexão da maior ordem possível do filtro(TNS_MAX_ORDER) r[] (também chamados de coeficientes PARCOR)
- Se o ganho de predição g_p não exceder um certo limiar t, não é usado nenhum *temporal noise shaping*. Neste caso, o bit de *tns_data_present* tem seu valor definido em zero e o processamento TNS acaba. Um valor adequado de limiar seria $t = 1{,}4$.
- Se o ganho de predição g_p exceder o limiar t, a TNS vai ser usada.
- No próximo passo, os coeficientes de reflexão são quantizados usados bits *coef_res*. Uma escolha apropriada para o valor do *coef_res* é 4 bits. O pseudo-código a seguir descreve a conversão dos coeficientes de reflexão r[] para valores de índice *index*[] e de volta para coeficientes de reflexão quantizados rq[].

```
iqfac = ((1 << (coef_res-1)) - 0.5) / (π/2.0);
iqfac_m = ((1 << (coef_res-1)) + 0.5) / (π/2.0);

/* Reflection coefficient quantization */
for (i = 0; i < TNS_MAX_ORDER; i++) {
   index[i] = NINT(arcsin( r[i] ) * ((r[i] >= 0) ? iqfac :
iqfac_m));
}
/* Inverse quantization */
for (i = 0; i < TNS_MAX_ORDER; i++) {
   rq[i] = sin( index[i] / ((index[i] >= 0) ? iqfac :
iqfac_m) );
}
```

onde arcsin() (arco seno) é a função inversa da função sin() (seno).

- A ordem do filtro *noise shaping* usado é determinada através da remoção posterior de todos os coeficientes de reflexão com um valor absoluto menor que um limiar p da "cauda" da matriz de coeficientes de reflexão. O número de coeficientes de

reflexão remanescentes é a ordem do filtro *noise shaping*. Um valor de limiar adequado para o truncamento é p = 0,1.

- Os coeficientes de reflexão restantes rq[] são convertidos em coeficientes de predição linear de ordem +1 a[] (conhecido como procedimento de *step-up*). Este procedimento é descrito na parte normativa do padrão MPEG-2 AAC na parte de conversão para coeficientes LPC [23].
- Os coeficientes LPC calculados a[] são usados como os coeficientes do filtro *noise shaping* do codificador.
- Finalmente, as informações laterais para a TNS são transmitidas.

6.2.7 *Perceptual Noise Substitution* – PNS

A ferramenta de substituição de ruído perceptual (*Perceptual Noise Substitution* – PNS) foi adicionada ao AAC no padrão MPEG-4. Ela tem como objetivo substituir componentes com características de ruído do sinal de entrada por uma representação paramétrica bastante compacta e, desta forma, aumentar a eficiência da compressão do codificador para alguns tipos de sinais.

A ideia da PNS é baseada no fato de que a percepção subjetiva pelo sistema auditivo humano de um sinal de áudio com características de ruído não é determinada pela forma de onda real, mas por sua estrutura espectral e temporal. Com isso pode-se assumir que os ruídos soarão todos similares aos ouvintes.

A técnica da PNS consiste em detectar as frequências com característica de ruído de um dado sinal de áudio e codificar as suas faixas de amplitude e frequência ao invés de codificar a forma de onda original. O que a PNS faz é verificar quais as bandas que contém ruído no sinal. Se a banda for considerada ruído, então toda sua faixa vai ser codificada pela potência do sinal contido nela, que significa que apenas um parâmetro, a potência do ruído, vai descrever todos os dados contidos nesta faixa.

A ferramenta PNS é aplicada no codificador MPEG-4 AAC de acordo com a Figura 6.8, seguindo os seguintes passos:

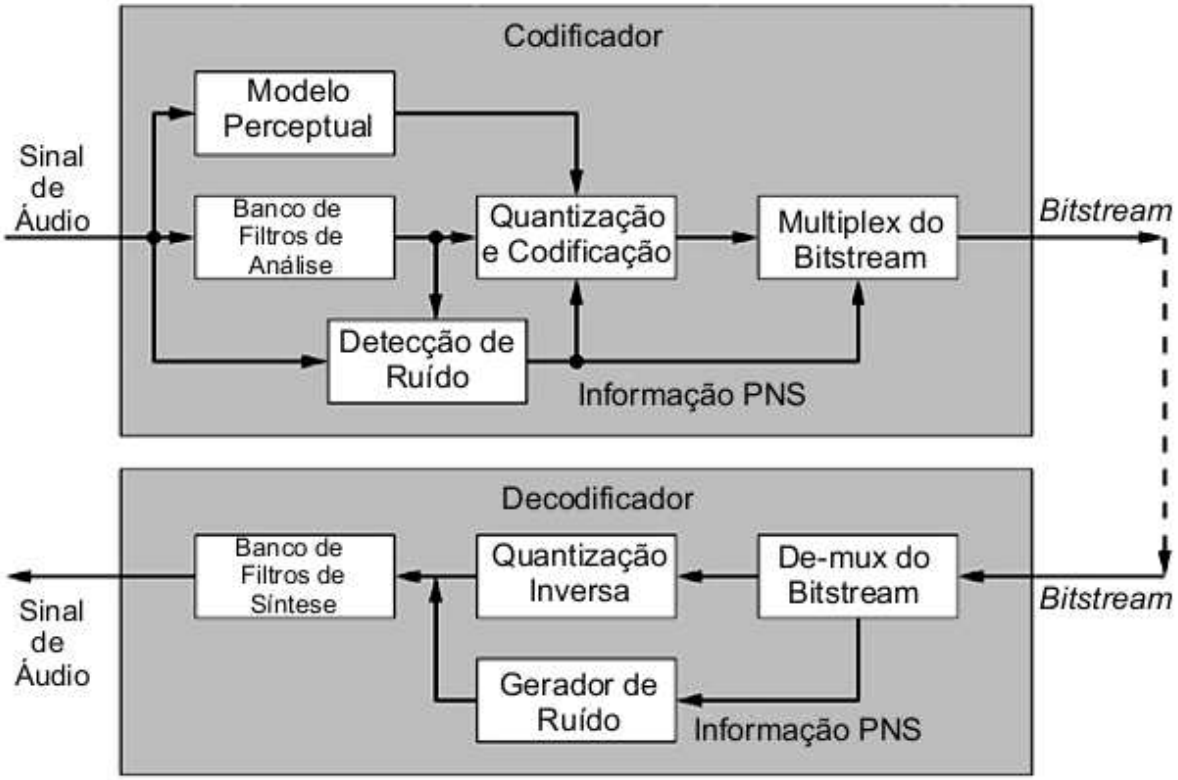


Figura 6.8 – Esquema da ferramenta PNS no codificador e decodificador MPEG-4 AAC (reproduzido de [35]).

- O sinal de entrada é analisado no codificador para determinar os componentes de entrada que tenham característica de ruído para cada quadro e banda de fator de escala.
- Se uma banda de fator de escala específica for considerada como ruído, o conjunto correspondente de coeficientes espectrais não será codificado e quantizado da maneira normal, mas será omitido deste processo. Ao invés destes coeficientes, uma *flag* de substituição de ruído será transmitida para o decodificador, junto com a energia total do conjunto de coeficientes espectrais substituídos.
- No decodificador, as informações de sinalização transmitidas são analisadas. Para bandas de fator de escala para as quais a substituição de ruído foi sinalizada, números pseudo-aleatórios são inseridos, para gerar um ruído aleatório ao invés de usar os coeficientes espectrais que não foram transmitidos, com uma energia total de ruído igual a do nível transmitido.

Como somente informações de sinalização e o valor da energia do ruído são transmitidos uma vez por banda de fator de escala e sem coeficientes espectrais, isto resulta em uma representação bem compactada de componentes do sinal que sejam ruído.

Esta técnica é útil para reduzir o número de bits necessários para sinais com característica de ruído e quanto mais tonal o sinal for, menor vai ser essa redução. Com isso a redução do número de bits quando for aplicada esta técnica vai depender de se o sinal é formado por ruído ou não e quais faixas de frequências são ruidosas.

6.2.8 Quantização e Codificação

A etapa de quantização no AAC tem dois objetivos: reduzir os bits usados para após a codificação de entropia de forma que os bits necessários fiquem dentro do limite desejado e controlar o ruído de quantização para que o limite do modelo psicoacústico possa ser atingido ou nenhum ruído perceptual seja introduzido após a quantização.

A quantização é feita em dois passos. Um é a quantização para amostras espectrais e o outro é para as bandas de fator de escala (*scale-factor bands*).

Para as amostras espectrais é usado um quantizador não uniforme de acordo com a equação (6.9).

$$\mathbf{s_q(i) = sgn(s(i)) \times round}\left[\left(\frac{\mathbf{|s(i)|}}{\sqrt[4]{\mathbf{2^{sf}}}}\right)^{\mathbf{0,75}} - \boldsymbol{\alpha}\right] \tag{6.9}$$

onde $s(i)$ e $s_q(i)$ representam o sinal original e as amostras espectrais quantizadas respectivamente, a função *round(x)* é uma função de arredondamento que retorna o valor inteiro que é mais próximo ao valor x, α é uma constante de valor baixo, e sf representa o parâmetro de quantização para banda de fator de escala onde a amostra i está localizada.

Com um quantizador não uniforme, o aumento da taxa sinal ruído com o aumento da energia do sinal é significantemente menor que com um quantizador linear. Para codificar os coeficientes do quantizador é usada codificação de Huffman.

Na teoria, o ideal seria atingir os requerimentos tanto do modelo psicoacústico como da taxa de bits, mas na prática é comum não se conseguir atingir esses valores. O codificador deve então aumentar os bits usados ou diminuir os requerimentos psicoacústicos.

No AAC, o limite de bits para cada quadro é normalmente a média de bits por quadro que pode ser calculada a partir da taxa de bits desejada e a taxa de amostragem do sinal de entrada. Além disso, existe um reservatório de bits disponível para cada quadro para permitir uma distribuição variável de bits extra entre quadros consecutivos.

O procedimento para alterar requerimentos e número de bits não está disponível no padrão, pois o MPEG não padroniza a parte do codificador apenas o *bitstream* gerado por ele que deve seguir a sintaxe definida na norma.

6.2.9 Bandas de Fator de Escala (*Scalefactor bands*)

No AAC, o codificador aplica uma amplificação para cada grupo individual de coeficientes espectrais, que são as bandas de fator de escala. O objetivo é formatar o ruído de quantização em unidades similares às bandas críticas, para permitir que os requerimentos do modelo psicoacústicos sejam atingidos com mais eficiência.

No AAC, isto é feito adicionando-se um parâmetro *sf* para cada banda de fator de escala como na equação (6.9) na quantização e incluindo informação lateral destes parâmetros de cada banda de fator de escala no *bitstream* para ser possível a quantização inversa no decodificador [49].

6.2.10 Codificação sem ruído

A ferramenta de codificação sem ruído (*noiseless coding*) é usada para reduzir ainda mais a redundância dos fatores de escala e do espectro quantizado de cada canal de áudio explorando redundâncias estatísticas sem reduzir a precisão [38].

No codificador AAC a entrada para o módulo codificação sem ruído é um conjunto de 1024 coeficientes espectrais quantizados.

A codificação sem ruído é feita através dos seguintes passos [23]:

6.2.10.1 Corte do Espectro

O primeiro passo do método de codificação sem ruído é um modo de limitar a faixa dinâmica de compressão a ser aplicada no espectro. Até quatro coeficientes podem ser codificados separadamente como magnitudes maiores que um, com um valor de +/-1 sobrando na matriz de coeficientes para carregar informações de sinal. Os coeficientes "cortados" são codificados como magnitudes inteiras e com um deslocamento a partir da base da matriz de coeficientes para marcar sua localização. Como a informação lateral usada para carregar os coeficientes cortados custa alguns bits, essa compressão sem ruídos só será aplicada se no final resultar em economia do número total de bits necessários.

6.2.10.2 Codificação de Huffman

No AAC os coeficientes espectrais são elevados à potência ¾ e quantizados. Embora a potência proporcione alguma compressão da faixa dinâmica, a distribuição de probabilidade para os níveis em qualquer quantizador é longe de ser uniforme [38].

Para compensar isto, um código de comprimento variável, o código de Huffman, que é baseado na probabilidade de ocorrência dos símbolos, é usado para representar os níveis do quantizador.

Apesar de poder ter comprimento variável, o código ainda deve ter um comprimento de bits inteiro, tal que se *H(p)* é a entropia da palavra-código representando um nível do quantizador dado (o nível sendo um membro do alfabeto do código) e *l* é o comprimento desta palavra-código, o limite para *l* é dado por:

$$\boldsymbol{H(p) \leq l < H(p) + 1} \qquad (6.10)$$

Para compensar essa ineficiência de até um bit, o código de Huffman pode ser entendido tal que N letras sejam codificadas ao mesmo tempo, para ser usado para representar uma sequência de coeficientes espectrais. Neste caso, o limite para *l* é dado por:

$$\frac{1}{N}H(p) \leq \frac{l}{N} < \frac{1}{N}H(p) + \frac{1}{N} \tag{6.11}$$

reduzindo a ineficiência da palavra código para um máximo de 1/N bits [38].

No AAC um código de Huffman entendido é usado para representa n-tuplas de coeficientes quantizados, com as palavras-código extraídas de um de 11 *codebooks*. Os coeficientes espectrais dentro das n-tuplas são ordenados (do menor para o maior) e o tamanho da n-tupla é de dois ou quatro coeficientes. O valor absoluto máximo dos coeficientes quantizados que pode ser representado por cada *codebook* de Huffman e o número de coeficientes em cada n-tupla para cada *codebook* é mostrado na Tabela 6.. Existem dois *cobebooks* para cada valor absoluto máximo, com cada um representando uma função de distribuição de probabilidade diferente.

Para reduzir o espaço necessário para o armazenamento do *codebook* no decodificador, a maioria deles representa valores sem sinal. Para estes *codebooks* a magnitude dos coeficientes é codificada usando Huffman e o bit de sinal de cada coeficiente que não for zero é adicionado a palavra-código.

Os *codebooks* 0 e 11 tem significado especial: O *codebook* 0 indica que todos os coeficientes dentro de uma seção são zero. O *codebook* 11 pode representar coeficientes quantizados que tem um valor absoluto maior ou igual a 16. Se a magnitude de um ou ambos os coeficientes for maior ou igual a 16, um mecanismo especial de código de escape é usado para representar estes valores, A magnitude dos coeficientes é limitada a não mais que 16 e 2-tupla correspondente é codificada por Huffman.

Índice do *Codebook*	Tamanho da Tupla	Valor Absoluto Max.	Valores com Sinal
0	-	0	-
1	4	1	sim
2	4	1	sim
3	4	2	não

4	4	2	não
5	2	4	sim
6	2	4	sim
7	2	7	não
8	2	7	não
9	2	12	não
10	2	12	não
11	2	16 (ESC)	não

Tabela 6.5 – *Codebooks* de Huffman [38].

Os bits de sinal, se necessários, são adicionados à palavra-código. Para cada coeficiente com magnitude maior ou igual a 16, um código de escape também é adicionado à palavra-código, de acordo com [38]:

escape code = <*escape_prefix*><*escape_separator*><*escape_word*> ,

onde

<*escape_prefix*> é uma sequência de N "1's" binários,

<*escape_separator*> é um "0" binário,

<*escape_word*> é um bit N+4 inteiro sem sinal, bit mais significativo primeiro,

e N é um contador que é grande suficiente para que a magnitude dos coeficientes quantizados seja igual a : *2^(N+4)* + <*escape-word*>.

6.2.10.3 Seccionamento

A codificação sem ruído segmenta o conjunto de 1024 coeficientes espectrais quantizados em seções, onde um único *codebook* de Huffman é usado para codificar cada seção.

O seccionamento é dinâmico e normalmente varia de bloco para bloco, de modo que o número de bits necessários para representar o conjunto completo de coeficientes espectrais quantizados seja o mínimo possível. Isto é feito usando um algoritmo ganancioso para misturar as seções que começa com o maior número possível de seções, cada uma das quais usa o *codebook* de Huffman com o menor índice possível.

Para reduzir o custo da informação lateral, várias bandas de fator de escala adjacentes podem ser agrupadas para formar uma seção e dividir um *codebook* de Huffman comum. Assim, uma seção é descrita no *bitstream* pelo número de bandas de fator de escala agrupadas mais o índice do *codebook* de Huffman usado. Deste modo, a codificação de entropia de todas a bandas de fator de escala ativas é definida pela escolha dos parâmetros de seccionamento [35].

As seções serão misturadas se as seções depois de misturadas tiverem uma contagem total de bits menor, com as junções de seções que gerarem a maior redução na contagem de bits sendo feitas primeiro. Se as seções a serem misturadas não usarem o mesmo *codebook* de Huffman, então um *codebook* com um índice igual a pelo menos o menor índice entre os dois deve ser usado, apesar de *codebooks* com um índice ainda maior poderem ser usados se isto resultar em uma contagem total de bits menor.

As seções frequentemente contêm apenas coeficientes cujo valor é zero. Por exemplo, se o sinal de entrada é limitado em banda a 20 kHz então o maior dos coeficientes é zero. Estas seções são codificadas com o *codebook* zero de Huffman, que é um mecanismo de escape que indica que todos os coeficientes são zero e que não é necessário que nenhuma palavra-código de Huffman seja enviada para aquela seção.

A única informação lateral que é necessária para a codificação sem ruído é o seccionamento. Não é necessário enviar informações de alocação de bits nesta parte. Ao invés disso o seccionamento indica o número de coeficientes espectrais que são enviados e o número de coeficientes representados por cada palavra-código é conhecido a partir do índice do *codebook*. Como as palavras-código de Huffman são exclusivamente decodificáveis, a sequência de palavras-código sozinha já determina o número de bits usados para representar os coeficientes espectrais.

6.2.10.4 Agrupamento e Interpolação

Se o banco de filtros é comutado para o estado *EIGHT_SHORT_SEQUENCE* ,isto é, para o modo de alta resolução no tempo, então os seus 1024 valores de saída consistem de uma matriz de 8x128 coeficientes espectrais representando as características no tempo-frequência do sinal durante o período de 8 janelas de análise curtas. Estes coeficientes são quantizados e precisam ser

codificados eficientemente pela ferramenta de codificação sem ruído. Como isto é feito usando codificação de entropia baseada em seções, é vantajoso arranjar a ordem dos coeficientes espectrais tal que o custo do seccionamento seja o menor possível.

Como foi visto anteriormente, uma operação de agrupamento é realizada na sequência de janelas curtas para reduzir a demanda por informação lateral (fatores de escala). Da mesma forma, o esquema de agrupamento é usado para trocar a ordem das bandas de fatores de escala e janelas na matriz de coeficientes espectrais antes da codificação sem ruído e seccionamento serem aplicados. Isto aumenta a probabilidade de ter coeficientes de magnitude similar arranjados em regiões contínuas da matriz de coeficientes e assim, aumenta a eficiência da codificação. Além disso, todas as partes que são zero dentro de cada grupo, devido a limitações de banda, são combinadas em uma única seção.

6.2.11 Camada de Transporte e Multiplexação

A parte de áudio do padrão MPEG-4 [25] vai até a ponto da construção de unidades de acesso que contém os dados comprimidos. Já a parte de sistemas descreve como converter essas unidades de acesso codificadas individualmente em *elementary streams*.

Não existe um mecanismo de transporte padrão para esses *streams* em um canal, pois existem muitas aplicações com requerimentos diferentes que podem usar o padrão MPEG-4. O que é padronizado é uma interface conhecida como *Delivery Multimedia Interface* (DMIF) que é especificada na norma ISO/IEC 14496-6. Esta interface descreve a capacidade de uma camada de transporte e a comunicação entre funções de transporte, multiplexação e demultiplexação em codificadores e decodificadores.

Mas para aplicações de áudio natural que não necessitam de codificação baseada em objetos e outras funções da parte de sistemas do MPEG-4, a parte de áudio do padrão [25] define um mecanismo de transporte e multiplexação chamado de LATM/LOAS feito para prover um *overhead* baixo e um mecanismo de transporte sem ter de utilizar os conceitos da parte 1 do MPEG-4 [24].

Este mecanismo de transporte usa duas camadas, uma para multiplexação e outra para sincronização. A camada de multiplexação, chamada de *Low-overhead MPEG-4 Audio Transport*

Multiplex (LATM) é a encarregada de fazer a multiplexação de vários *payloads* de áudio e seus elementos *AudioSpecificConfig*. A camada de sincronização especifica uma sintaxe auto-sincronizada do *transport stream* de áudio MPEG-4 que é chamada de *Low Overhead Audio Stream* (LOAS) [25]. Um esquema destas camadas pode ser visto na Figura 6.9.

A Tabela 6. mostra uma visão geral dos formatos de multiplexação, armazenamento e transmissão disponíveis para um áudio MPEG-4 dentro do padrão.

	Formato	Parte do MPEG-4	Originalmente definida em:	Descrição
Multiplexação	M4Mux	ISO/IEC 14496-1 (normativa)		Esquema de Multiplexação do MPEG-4
	LATM	ISO/IEC 14496-3 (normativa)		*Low Overhead Audio Transport Multiplex*
Armazenamento	ADIF	ISO/IEC 14496-3 (informativa)	ISO/IEC 13818-7 (normativa)	*Audio Data Interchange Format*, (apenas para o AAC)
	MP4FF	ISO/IEC 14496-12 (normativa)		MPEG-4 *File Format*
Transmissão	ADTS	ISO/IEC 14496-3 (informativa)	ISO/IEC 13818-7 (normativa)	*Audio Data Transport Stream* (apenas para o AAC)
	LOAS	ISO/IEC 14496-3 (normativa)		*Low Overhead Audio Stream,* baseado no LATM, existem três versões disponíveis: *AudioSyncStream()* *EPAudioSyncStream()* *AudioPointerStream()*

Tabela 6.6 – Formatos de multiplexação, armazenamento e transmissão disponíveis para áudio MPEG-4 [25].

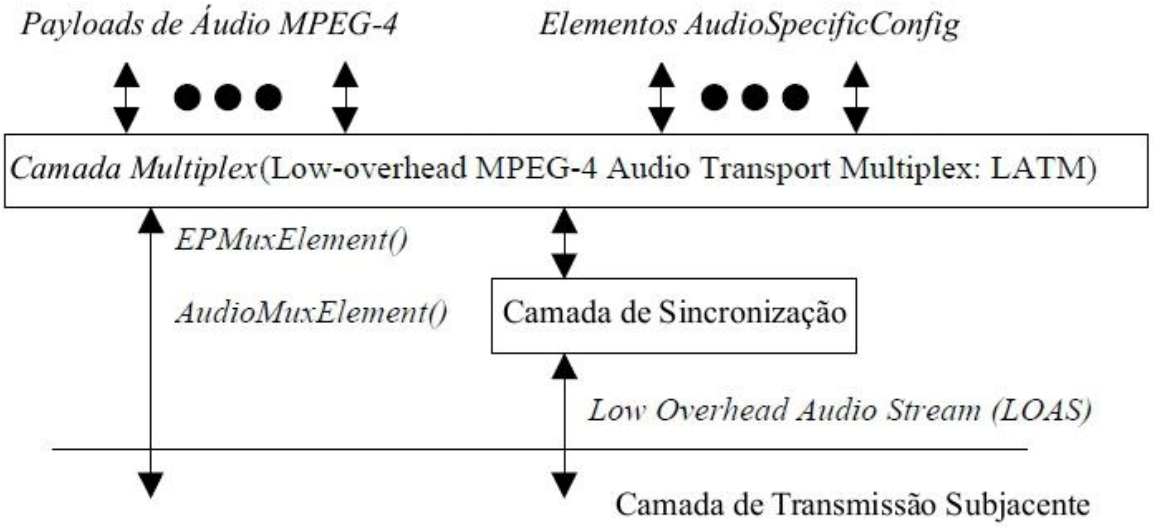


Figura 6.9 – Transporte de Áudio no MPEG-4 [25].

6.2.12 Camada de Sincronização

A camada de sincronização fornece ao elemento multiplexado um mecanismo auto-sincronizado para gerar o LOAS.

O LOAS tem três tipos diferentes de formatos: *AudioSyncStream, EPAudioSyncStream* e *AudioPointerStream.*

O *AudioSyncStream* consiste de uma palavra de sincronia (*syncword*), o elemento multiplexado com alinhamento de *byte* e sua informação de comprimento. A máxima distância em bytes entre duas *syncwords* é de 8192 *bytes*. O *stream* auto-sincronizado deve ser usado para o caso em que a camada de transmissão abaixo vier sem nenhuma sincronização de quadro.

O *EPAudioSyncStream* é uma alternativa ao *AudioSyncStream* para canais que precisam de proteção extra contra erros. Este formato inclui uma palavra de sincronia maior e um contador de quadros para detectar perdas de quadros. A informação de comprimento e o contador de quadros são ainda protegidos por um código FEC.

O *AudioPointerStream* é usado em aplicações que usem uma camada de transmissão com sincronização de quadro fixo, onde o empacotamento (*framing*) da transmissão não pode ser sincronizado com o elemento de comprimento variável multiplexado [25].

No padrão brasileiro a camada de sincronização do transporte de áudio usa o formato de transmissão *AudioSyncStream* [4].

6.2.13 Camada de Multiplexação

A camada LATM multiplexa vários *payloads* de áudio MPEG-4 e elementos de sintaxe *AudioSpecificConfig* em um elemento multiplexado. O formato de elemento multiplexado é escolhido entre *AudioMuxElement* e *EPMuxElement* dependendo se proteção contra erros for necessária no elemento multiplexado. O *EPMuxElement* é a versão com proteção contra erros do *AudioMuxElement*.

Os elementos multiplexados podem ser diretamente transmitidos em camadas de transmissão com sincronização de quadro. Neste caso, o primeiro bit do elemento multiplexado deve estar localizado no primeiro bit de um *payload* de transmissão na camada de transmissão inferior. Se o *payload* de transmissão só permite *payload byte-aligned*, bits de preenchimento para o alinhamento de *byte* devem seguir o elemento multiplexado. O número de bits de preenchimento deve ser menor que 8. Estes bits devem ser removidos quando o elemento multiplexado for demultiplexado em *payloads* de áudio MPEG-4 e então esses dados são direcionados para o decodificador MPEG-4 correspondente.

No padrão brasileiro a codificação e o empacotamento (*framing*) intermediário do áudio devem obrigatoriamente ser compatíveis com LATM/LOAS. O *elementary stream* deve obrigatoriamente ser primeiramente encapsulado no formato de transporte LATM e deve obrigatoriamente utilizar o elemento de multiplexação *AudioMuxElement* [4].

7 MODIFICAÇÕES NOS CODIFICADORES E RESULTADOS

Foi feita pesquisa e realizado estudo de toda teoria que envolve codificadores perceptuais de áudio. Também foi feito estudo da norma ISO/IEC 14496-3, parte referente ao codificador MPEG-4 AAC. O objetivo era identificar um codificador que pudesse ser modificado para suportar as ferramentas usadas no padrão brasileiro SBTVD.

Em seguida foi realizada uma busca por um código aberto de referência no padrão MPEG-4 AAC como base para o projeto. Foram estudados dois códigos abertos de codificadores AAC: FAAC e 3GPP. Também foi analisado o código de referência do próprio padrão ISO, especialmente a parte do LATM/LOAS. Todos estão em linguagem C.

O código de referência do 3GPP segue o padrão HE-AAC v2 que suporta estéreo e as extensões SBR e PS. É o único codificador de código aberto que suporta as extensões SBR e PS do AAC.

O codificador FAAC suporta mais de dois canais, mas não tem suporte às extensões SBR e PS. O decodificador FAAD originalmente suporta todos os modos, mas não suporta LATM/LOAS.

Todos os códigos são em linguagem C, foi usada a ferramenta Visual C++ como ambiente de desenvolvimento.

O decodificador FAAD foi usado para referência e comparação por suportar também a configuração de canais em modo 5.1.

7.1 O CÓDIGO DE REFERÊNCIA DO 3GPP

O Projeto em parceria da terceira geração (*3rd Generation Partnership Project* - 3GPP) é uma colaboração entre grupos de associações de telecomunicações, para fazer especificações com aplicação global para sistemas móveis da terceira geração (3G).

Os grupos envolvidos são ARIB/TTC (*Association of Radio Industries and Businesses/Telecommunication Technology Committee* - Japão), CCSA (*China Communications Standards Association* - China), ETSI (*European Telecommunications Standards Institute* - Europa), ATIS (*Alliance for Telecommunications Industry Solutions* - América do Norte), TTA

(*Telecommunications Technology Association* - Coréia do Sul). O projeto foi criado em dezembro de 1998.

Um dos códigos de referência apresentados neste projeto é baseado nas normas 3GPP da série 26 dos TS 26.401 à 26.412. É um codificador no padrão HE-AAC v2, que suporta apenas dois modos de configuração de canais: estéreo e mono.

O codificador do 3GPP gera arquivos no formato 3gp, que é baseado no formato base ISO MPEG-4.

Para codificação estéreo é usada a ferramenta *M/S Stereo* para taxas de bit maiores e *Parametric Stereo* para menores. Se for feita uma modificação para suportar seis canais, a ferramenta *Parametric Stereo* não poderá ser usada, já que só suporta dois canais na norma ISO[25].

A Figura 7.1 mostra o diagrama de blocos do codificador HE-AAC v2 do 3GPP. Na entrada o sinal PCM no domínio do tempo entra em um *downmix* de estéreo para mono que só é ativado se o codificador estiver em modo mono e o sinal de entrada for estéreo. Depois o sinal passa por filtros de *resample* IIR para ajustar a taxa de amostragem do sinal de entrada se for diferente da esperada pelo codificador.

O codificador opera com duas taxas de amostragem, onde o codificador SBR opera na taxa de amostragem de codificação fs_{enc} que é a que vem do *resampler* IIR e o codificador AAC na metade dessa taxa $fs_{enc}/2$. Por isso um *downsampler* 2:1 está presente na entrada do codificador AAC. A ferramenta *Parametric Stereo* é usada para taxas de bit em estéreo abaixo de 44 kbit/s. O codificador AAC usa o perfil LC.

O codificador SBR consiste de um banco de filtros de análise QMF que são usados para obter o envelope espectral do sinal original. O *stream* do SBR é composto do envelope e outras informações auxiliares extraídas pelo módulo.

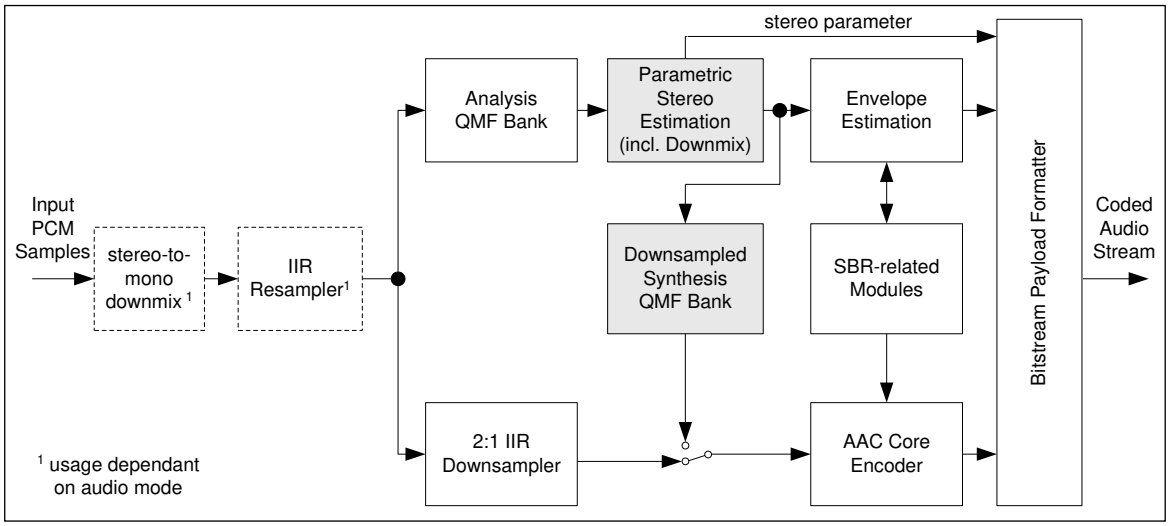


Figura 7.1 – Diagrama de blocos do codificador HE-AAC v2 do 3GPP[1].

A parte do SBR é incluída de forma a manter compatibilidade com um decodificador que não suporte SBR, mas consiga decodificar a parte do AAC.

A parte do AAC principal funciona no modo LC no nível 2 do perfil definido na norma ISO, o que o restringe a funcionar apenas em dois canais.

7.2 O CÓDIGO FAAC/FAAD

O FAAC (*Freeware Advanced Audio Coder*) é um projeto de código aberto com licença GPL que inclui o codificador AAC, FAAC e o decodificador AAC, FAAD.

O codificador e o decodificador suportam tanto MPEG-2 AAC quanto MPEG-4 AAC.

São suportados os perfis LC, principal e LTP para o codificador e SBR, PS (HE-AAC) apenas para o decodificador. O FAAD consegue processar os seguintes tipos de encapsulamento: ADTS AAC, *raw* AAC (sem encapsulamento) e MP4. Tanto o FAAC como FAAD suportam modo multicanal.

O código FAAD é separado em bibliotecas, a principal sendo a *libfaad* onde ficam as funções do decodificador AAC. O código do FAAD é separado em diretórios cujo conteúdo é descrito na Tabela 7.1.

Nome	Conteúdo
libfaad	Biblioteca contendo o decodificador FAAD
mp4ff	Biblioteca de suporte aos formatos de arquivo MP4 (MPEG-4 *file format*)
frontend	Contém a interface por linha de comando com a biblioteca do FAAD

Tabela 7.1 - Estrutura de diretórios do FAAD.

7.3 MODIFICAÇÕES REALIZADAS NO DECODIFICADOR FAAD

O decodificador foi modificado para adicionar funcionalidades necessárias para decodificar um áudio AAC de acordo com o padrão SBTVD. Como foi visto anteriormente o padrão usa a camada LATM/LOAS, que não é suportada no FAAD original.

Foi adicionado um novo módulo como uma biblioteca que o FAAD acessa para processar essa camada e mandar uma unidade de acesso (*access unit*) compatível para o decodificador. Este módulo foi criado baseado no código de referência da ISO/IEC [26].

O código modificado adiciona à estrutura de diretórios do original a seguinte biblioteca:

Nome	Conteúdo
liblatm	Biblioteca com as funções usadas para ler e sincronizar áudio AAC com a camada LATM/LOAS.

Tabela 7.2 - Estrutura de diretórios do FAAD modificado.

Dentro da biblioteca LATM existem funções para ler cada quadro que tenha a camada LATM/LOAS. As funções da biblioteca LATM/LOAS seguem o item 1.7 do padrão ISO/IEC [25] adaptado para seguir as definições do SBTVD da norma ABNT [4].

Essas funções são executadas uma vez por quadro (*frame*). A cada início de quadro é feita a sincronia procurando a palavra de sincronismo (*syncword*) definida no padrão (0x2B7).

Quando for encontrada a palavra, vão ser lidos os bits do *audioMuxLengthBytes* e será iniciada a leitura do *AudioMuxElement*() do *stream*, seguindo a sintaxe do pseudocódigo da tabela 1.23 do padrão ISO/IEC [25].

Um quadro do *bitstream* é basicamente composto por: *syncword*, cabeçalho LATM/LOAS, dados e bits para alinhamento de byte (preenchimento para os pacotes ficarem

com tamanhos em byte inteiros, de acordo com o item 1.7.3 no padrão ISO: "Se o *payload* de transmissão só permite *payload* com alinhamento de byte, bits de preenchimento para o alinhamento de byte devem seguir o elemento multiplexado. O número de bits de preenchimento deve ser menor que 8. Estes bits devem ser removidos quando o elemento multiplexado for demultiplexado em *payloads* de áudio MPEG-4 e então esses dados são direcionados para o decodificador MPEG-4 correspondente" [25].

Na Tabela 7.3 podem ser vistos os campos que formam um quadro de um *bitstream* com LATM/LOAS de acordo com o padrão ISO/IEC, na ordem em que o programa lê os bits.

Nome	**Elemento de sintaxe**	**Nº de bits**
syncword	*AudioSyncStream*()	11
audioMuxLengthBytes	*AudioSyncStream*()	13
useSameStreamMux	*AudioMuxElement*()	1
uisbtvd		15
audioMuxVersion	*StreamMuxConfig*()	1
audioMuxVersionA	*StreamMuxConfig*()	1
(se audioMuxVersion = 1) bytesForValueM1	*LatmGetValue*()	2
(se audioMuxVersion = 1) latmValue	*LatmGetValue*()	8*(*bytesForValueM1*+1)
allStreamSameTimeFraming	*StreamMuxConfig*()	1
numSubFrames	*StreamMuxConfig*()	6
numProgram (progCount)	*StreamMuxConfig*()	4
numLayer (tmp (number of tracks))	*StreamMuxConfig*()	3
useSameConfig	*StreamMuxConfig*()	1

frameLengthType	*StreamMuxConfig*()	3
latmBufferFullness	*StreamMuxConfig*()	8
otherDataPresent	*StreamMuxConfig*()	1
otherDataLenBits (opcional se otherDataPresent =1)	*StreamMuxConfig*()	variável
crcCheckPresent	*StreamMuxConfig*()	1
crcCheckSum (opcional, se crcCheckPresent = 1)	*StreamMuxConfig*()	8
tmp	*PayloadLengthInfo*()	8
data (payload)	*PayloadMux*()	8*numBytes
ltmp: align bits		align

Tabela 7.3 – Composição de um quadro LATM/LOAS.

A descrição dos elementos usados na sintaxe do LATM/LOAS, separados de acordo com o elemento de sintaxe ao qual pertencem, é detalhada a seguir.

7.3.1 AudioSyncStream()

O *AudioSyncStream*() faz parte da camada de sincronização, LOAS. É formado pela palavra de sincronia, *syncword,* pelo elemento multiplexado, *AudioMuxElement*() com alinhamento de byte, e informação do tamanho deste elemento multiplexado, *audioMuxLengthBytes*. A distância máxima em *bytes* entre duas *syncwords* é de 8192 bytes. A sintaxe do *AudioSyncStream*() pode ser encontrada na tabela 1.23 do padrão ISO/IEC [25].

7.3.2 AudioMuxElement()

O *AudioMuxElement*() faz parte da camada de multiplexação, LATM. A sintaxe do *AudioMuxElement*() é definida no padrão ISO/IEC [25] na tabela 1.28 e item 1.7.3.2.2.

É preciso ter uma *flag* chamada *muxConfigPresent* definida na camada subjacente para analisar o *AudioMuxElement*(). Se o valor de *muxConfigPresent* for 1, indica que a configuração de multiplexação, *StreamMuxConfig*() está multiplexada dentro do *AudioMuxElement*(), isto é, uma transmissão dentro da banda. Se o valor não for igual a um, a *StreamMuxConfig*() deve ser transmitida através de métodos fora da banda, como protocolos de anúncio/descrição/controle de sessão. No código de referência usado *muxConfigPresent* é sempre 1.

O próximo passo é ler o bit ***useSameStreamMux*** : uma *flag* que indica se a configuração de multiplexação do quadro anterior será aplicada no quadro atual. Se no quadro atual ***useSameStreamMux*** não for verdadeira, vai ser analisado o *StreamMuxConfig*(), caso contrário será usado o do quadro anterior.

7.3.2.1 StreamMuxConfig*()*

O *StreamMuxConfig*() está contido dentro do *AudioMuxElement*() na camada de multiplexação LATM. A sintaxe deste elemento está na tabela 1.29 do padrão ISO/IEC [25]. É formado por:

audioMuxVersion:

Elemento que indica a sintaxe de multiplexação usada. Se for igual a 1, suporta a transmissão de *taraBufferFullness* e a transmissão dos comprimentos de funções *AudioSpecificConfig*() individuais.

audioMuxVersionA:

Elemento que sinaliza a versão da sintaxe usada. Valores possíveis: 0 (padrão) e 1 (reservado para futuras extensões).

taraBufferFullness

Uma variável auxiliar que indica o estado do reservatório de bits durante a codificação das informações de status do LATM. É transmitida como o número de bits disponíveis no reservatório de bits *tara* dividido por 32 e truncado para um valor inteiro.

allStreamsSameTimeFraming

Elemento que indica se todos os *payloads* que são multiplexados no *PayloadMux*() dividem uma base de tempo comum. O valor deste elemento é sempre 1 no SBTVD, indicando que os *payloads* sempre vão dividir uma base de tempo comum.

numSubFrames

Elemento que indica quantos quadros *PayloadMux*() são multiplexados (*numSubFrames* + 1). Se mais de um *PayloadMux*() for multiplexado, todos os *PayloadMux*() vão dividir um *StreamMuxConfig*() comum. O valor mínimo é zero, o que indica um *subframe*. No SBTVD este valor sempre é zero.

numProgram

Elemento que indica quantos programas são multiplexados (*numProgram*+1). O valor mínimo é zero, indicando um programa. No SBTVD este valor sempre é zero.

numLayer

Elemento que indica quantas camadas (*layers*) escaláveis são multiplexadas (*numLayer* + 1). O valor mínimo é zero, indicando uma camada. No SBTVD este valor sempre é zero.

useSameConfig

Elemento que indica que a *AudioSpecifConfig*() não foi transmitida e que a *AudioSpecificConfig*() transmitida mais recentemente deve ser aplicada, de acordo com a Tabela 7.4 – Valores de *useSameConfig*.

.

ascLen[prog][lay]

Variável auxiliar que indica o comprimento em bits da função *AudioSpecificConfig*() seguinte, incluindo possíveis bits de preenchimento

Valor de *useSameConfig*	Descrição
0	*AudioSpecificConfig*() está presente.
1	*AudioSpecificConfig*() não está presente. *AudioSpecificConfig*() presente na camada ou programa anterior deve ser aplicada.

Tabela 7.4 – Valores de *useSameConfig*.

fillBits

Bits de preenchimento.

frameLengthType

Elemento que indica o tipo de tamanho de quadro (*frame lenght type*) do *payload*. Para objetos CELP e HVXC, o tamanho do quadro (bits/quadro) é armazenado em tabelas e apenas os índices para apontar o tamanho do quadro do *payload* atual são transmitidos ao invés de enviar o valor do tamanho de quadro diretamente. Os valores possíveis deste elemento e o que significam estão detalhados na Tabela 7.5.

frameLengthType	Descrição
0	*Payload* com tamanho de quadro variável. O tamanho do *payload* em bytes é diretamente especificado com códigos de 8 bits em *PayloadLengthInfo*().
1	*Payload* com tamanho de quadro fixo. O tamanho do *payload* em bits é especificado pelo *frameLenght* no *StreamMuxConfig*().
2	Reservado
3-7	*Payload* para objetos CELP ou HVXC. Estes tipos de objetos estão fora do escopo do SBTVD.

Tabela 7.5 – Tipos de comprimentos de quadro (*FrameLengthType*).

latmBufferFullness

Elemento que indica o estado do reservatório de bits durante a codificação da primeira unidade de acesso de programa ou camada específico em um *AudioMuxElement*(). Ele é transmitido como o número de bits disponíveis no reservatório de bits dividido pelo NCC dividido por 32 e truncado para um valor inteiro. Um valor hexadecimal FF, sinaliza que o programa e a camada em questão são de taxa variável. Neste caso, *buffer fullness* não é aplicável. O estado do reservatório de bits é derivado de acordo com estabelecido na subparte 4, subitem 4.5.3.2 do padrão ISO/IEC [25]. No caso em que *audioMuxVersion* é igual a zero, os bits gastos para dados que não sejam *payload* (por exemplo, informações sobre o status da multiplexação ou outros dados) são considerados no primeiro *latmBufferFullness* que ocorrer em um *AudioMuxElement*(). Para o AAC, as limitações dadas pelo buffer de entrada mínimo do decodificador se aplicam. No caso em que *allStreamsSameTimeFraming* é igual a um, e se apenas um programa e uma camada estão presentes, isto leva a uma configuração LATM similar a ADTS. No caso de *audioMuxVersion* igual a 1, os bits gastos para dados que não sejam *payload* são considerados pelo *taraBufferFullness*.

frameLength

O elemento *frameLength* só aparece se *frameLengthType* for igual a 1. Ele indica o tamanho do *payload* que tenha *frameLengthType* igual a 1. O tamanho do *payload* em bits é especificado como 8 * (*frameLenght* + 20).

otherDataPresent

Uma *flag* que indica a presença de outros dados que não sejam *payloads* de áudio quando seu valor for igual a 1.

otherDataLenBit

Variável auxiliar que indica o tamanho em bits dos outros dados.

crcCheckPresent

Elemento que indica a presença de bits de verificação CRC para as funções de dados do *StreamMuxConfig*().

crcCheckSum

Dados para detecção de erro CRC. Este CRC usa geração polinomial CRC8 e cobre todo o *StreamMuxConfig*() excluindo o bit *crcCheckPresent*.

LatmGetValue()

O pseudocódigo que representa a sintaxe desta função está representado na Tabela 1.30 do padrão ISO/IEC [25]. É usada para obter o valor de *taraBufferFullness* quando o *audioMuxVersion* for igual a 1, valor da variável auxiliar *ascLen* e valor de *otherDataLenBits* em *StreamMuxConfig*(). Tem os seguintes elementos:

bytesForValue

Elemento que indica o número de ocorrências do elemento *valueTmp*.

valueTmp

Elemento usado para calcular o valor da variável auxiliar *value*.

value

Variável auxiliar que representa um valor retornado pela função *LatmGetValue*().

7.3.2.2 PayloadLenghtInfo*()*

Após a análise do *StreamMuxConfig*(), ainda dentro do *AudioMuxElement*(), o decodificador vai analisar o *PayloadLenghtInfo*(), que contém informações do tamanho do *payload e* cuja sintaxe está disponível na Tabela 1.31 do padrão ISO/IEC [25]. Ele tem os seguintes elementos:

tmp

Elemento que indica o tamanho do *payload* de um *payload* com *frameLenghtType* igual a 0. O valor 255 é usado como um valor de escape e indica que pelo menos um valor ***tmp*** extra vem em seguida. O tamanho geral do *payload* transmitido é calculado somando todos os valores parciais.

MuxSlotLenghtCoded

Elemento usado apenas em objetos do tipo CELP e HVXC, não se aplica ao AAC LC usado no SBTVD.

numChunk

Elemento que indica o número de blocos (*chunks*) do *payload* (*numChunk* + 1). Cada bloco pode pertencer a uma unidade de acesso com uma base de tempo diferente; só é usado se *allStreamsSameTimeFraming* for zero (no SBTVD este valor é sempre 1, então não é usado). O valor mínimo é zero indicando um bloco.

streamIndx

Elemento que indica o *stream*. Só é usado se os *payloads* estiverem divididos em blocos (*chunks*).

AuEndFlag

Flag que indica se o *payload* é o último fragmento, no caso em que a unidade de acesso é transmitida em pedaços.

7.3.2.3 PayloadMux*()*

Após a análise do *PayloadLenghtInfo*(), o decodificador lê o *PayloadMux*(), que está disponível na Tabela 1.32 do padrão ISO/IEC [25].

O *PayloadMux*() é o *payload* de áudio real por meio de uma unidade de acesso (AU) quando *allStreamsSameTimeFraming* for igual a 1 (caso do SBTVD) ou parte de uma concatenação de unidades de acesso subsequentes quando *allStreamsSameTimeFraming* for igual a zero.

Os seguintes elementos não são usados no SBTVD:

- *coreFrameOffset* só é usado para tipos de objeto que não fazem parte do escopo do AAC LC (CELP).
- *numSlotLengthCoded* é usado só em tipos de *frameLength* que são usados apenas em objetos do tipo CELP.

- Os campos *numChunk*, *StreamIndx* e *AuEndFlag*, só aparecem se *allStreamSameTimeFraming* não for igual a 1 e no SBTVD este valor é sempre 1, então não são usados.

Após extrair a unidade de acesso do *PayloadMux*(), esta será passada para *libFAAD* onde será decodificada pelo decodificador AAC principal usando as informações necessárias que foram obtidas do cabeçalho LATM/LOAS.

7.4 TESTES

Foram feitos testes do tempo que seria necessário para decodificar um arquivo de áudio AAC com e sem a camada LATM/LOAS, para verificar o quanto de atraso na decodificação a camada LATM/LOAS pode adicionar. Também foi feito um comparativo com o programa rodando em modo ponto flutuante e modo ponto fixo.

Para isto, foi feito um pequeno programa em linguagem *Python* que chama o FAAD para fazer a decodificação várias vezes e vai armazenando o tempo que foi gasto em cada decodificação em um arquivo de texto. Foi feita uma média dos valores e calculada a diferença entre a média dos tempos com e sem a camada LATM.

7.5 RESULTADOS

As médias dos tempos obtidos nos testes usando um áudio com duração de 50s podem ser vistas na Tabela 7.6.

Média dos tempos de decodificação com LATM em ponto flutuante	2,4514 s
Média dos tempos de decodificação sem LATM em ponto flutuante	2,2529 s
Média dos tempos de decodificação com LATM em ponto fixo	12,7666 s
Média dos tempos de decodificação sem LATM em ponto fixo	11,8327 s
Diferença entre os tempos com e sem LATM em ponto fixo	0,9339 s
Diferença entre os tempos com e sem LATM em ponto flutuante	0,19856 s (8.10%)

Tabela 7.6 – Tempo de decodificação do áudio com e sem LATM.

De acordo com os valores obtidos nos testes:

- O decodificador ficou bem mais lento em modo ponto fixo para qualquer caso.
- A decodificação com a camada LATM é aproximadamente 10% mais lenta do que quando não está presente.

A decodificação do áudio ainda pode ser considerada em tempo real, pois demora bem menos que a duração do áudio usado nos testes.

As médias dos tempos podem ser vistas no gráfico da Figura 7.2.

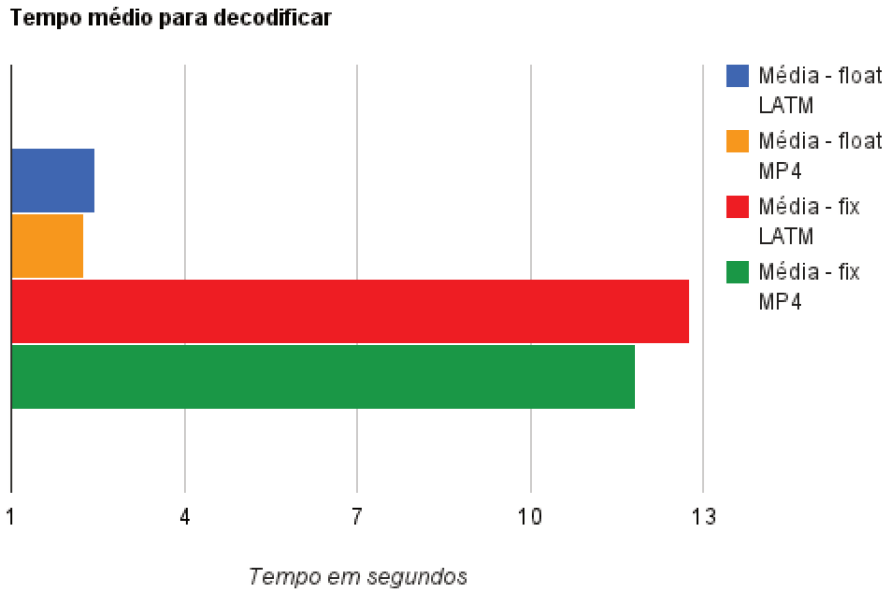


Figura 7.2 – Tempo médio levado pelo decodificador em cada caso.

8 CONCLUSÃO

Este trabalho apresentou uma análise e modificação de um codificador do padrão MPEG-4 AAC adaptando-o ao padrão brasileiro de TV digital (SBTVD).

O SBTVD trouxe algumas melhorias em relação ao padrão ISDB-T no qual foi baseado, como o uso do padrão MPEG-4 ao invés do MPEG-2 para codificação de áudio, além da especificação de um novo *middleware* denominado Ginga [6]. O MPEG-4 adiciona várias ferramentas que aumentam a eficiência da codificação de áudio, como SBR e PS no modo HE-AAC que melhoram a qualidade percebida do áudio quando são necessárias taxas de bits reduzidas (em transmissões para aparelhos celulares, por exemplo). O MPEG-4 também adicionou a camada LATM/LOAS que facilita a sincronização e multiplexação da parte de áudio sem ser necessário usar ferramentas mais complexas da parte de sistemas do padrão.

Os primeiros capítulos tiveram como objetivo fornecer um fundamento teórico geral sobre como funciona a codificação de áudio com foco nas técnicas perceptuais (com perdas). Estas técnicas eliminam partes do áudio que normalmente não são percebidas pelo sistema auditivo humano, mantendo uma qualidade percebida muito próxima ao áudio sem perdas, usando como referência a qualidade de CD. Em seguida foi detalhada toda parte relativa ao padrão MPEG-4 e como é usado no SBTVD seguindo as normas brasileiras da ABNT.

Os codificadores AAC estudados, FAAC/FAAD e 3GPP, foram escolhidos por serem de código aberto. Também foi estudado o código de referência do padrão ISO/IEC MPEG-4 [26], focando na parte que lida com a camada LATM/LOAS, este código foi usado para modificar o FAAD e tornar possível a decodificação de áudio com a camada LATM/LOAS.

O codificador AAC de referência do 3GPP tem a vantagem de suportar tanto na codificação como na decodificação as extensões SBR e PS do HE-AAC, mas não suporta modo multicanal, funcionando apenas em estéreo ou mono.

O codificador FAAC tem a vantagem de suportar modo multicanal, mas não codifica nos modos SBR e PS. Apenas o decodificador, FAAD, suporta estes modos.

Nem o decodificador do 3GPP nem o FAAD possuem suporte à camada LATM/LOAS usada no padrão brasileiro.

O FAAD foi escolhido para ser modificado por ter suporte multicanal e ter seu código bem organizado, modularizado, separado em bibliotecas.

Uma parte importante na implementação da camada LATM/LOAS foi a parte de adicionar uma rotina para a sincronia com a palavra definida no padrão. Tendo esta parte funcionando é possível prosseguir para a leitura dos cabeçalhos LATM/LOAS.

A adição da camada LATM/LOAS aumentou um pouco o tempo necessário para o áudio ser decodificado. Apesar disso, não adicionou um atraso grande o suficiente para prejudicar a decodificação do áudio, pois ainda continua em tempo real, mesmo no modo ponto fixo que é bem mais lento por não ser otimizado na implementação utilizada.

Para trabalhos futuros podem ser feitas otimizações para o decodificador, especialmente para a versão em ponto fixo, que é executada de forma bem mais lenta, para poder ser embarcado em um processador digital de sinais que só trabalhe em ponto fixo, por exemplo, como foi realizado em [11].

Também pode ser adicionada a camada LATM/LOAS em um dos codificadores AAC de código aberto como o FAAC ou o codificador do 3GPP para permitir a codificação de áudio pronto para ser usado em transmissões do SBTVD.

Para adequar o codificador FAAC para codificar áudio para o SBTVD, além do suporte a LATM/LOAS, teria que ser adicionado suporte a codificação usando as ferramentas do HE-AAC, SBR e PS.

Para o codificador e decodificador do 3GPP poder ser usado para o SBTVD, teria que ser adicionado suporte a áudio multicanal além de adicionar suporte à camada LATM/LOAS.

REFERENCIAS BIBLIOGRÁFICAS

[1] 3GPP TS 26.401 *Enhanced aacPlus general audio codec; General description, 3rd Generation Partnership Project* (3GPP), 2008.

[2] 3GPP TS 26.403 *Enhanced aacPlus general audio codec; Encoder Specification AAC part.*, *3rd Generation Partnership Project* (3GPP), 2008.

[3] ABNT NBR 15601 Televisão digital terrestre – Sistema de transmissão, 2007.

[4] ABNT NBR 15602-2 Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação. Parte 2: Codificação de áudio, 2008.

[5] ABNT NBR 15602-3 Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação. Parte 3: Sistemas de multiplexação de sinais, 2008.

[6] ABNT NBR 15606 Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital Partes 2, 5 e 7. Disponível em: <http://www.ginga.org.br>

[7] Barbedo, J. G. A. Avaliação objetiva de qualidade de sinais de áudio e voz, Dissertação (Doutorado), UNICAMP, Campinas, 2004.

[8] Baumgarte, F.; Faller, C. *Binaural cue coding - Part I: Psychoacoustic fundamentals and design principles, IEEE Trans. Speech Audio Process, pp 509-519, Nov* 2003.

[9] Baumgarte, F.; Faller, C. *Why binaural cue coding is better than intensity stereo coding,* in *Proc. AES 112th Conv.*, Munich, Germany, May 2002.

[10] Blumlein, A. D. *Improvements in and relating to sound-transmission, sound recording and sound-reproducing systems*, 1931, UK Patent 394325.

[11] Braga, V. J. A. Desenvolvimento de um decodificador de áudio embarcado para o ISDB-T_B Dissertação (Mestrado) - UNICAMP, Campinas, Abr 2011.

[12] Brandenburg, K.; Johnston, J. D. Johnston, *Second generation perceptual audio coding: The hybrid coder*, *in 88th AES Conv. Preprint*, (Montreaux), pp. 1–10, Mar. 1990.

[13] Cave, C. R. *Perceptual modeling for low-rate audio coding*, Dissertação de Mestrado - McGill University, Jun 2002.

[14] Egan, J.; Hake, H. *On the masking pattern of a simple auditory stimulus*, J. Acoust. Soc. Am., vol. 22, pp. 622–630, 1950.

[15] Ekstrand, P. *Bandwidth Extension of Audio Signals by Spectral Band Replication*, IEEE Proc. Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002), Leuven, Belgium, Nov. 2002.

[16] Fletcher, H.; Munson, W. *Relation between loudness and masking*, J. Acoust. Soc. Am., vol. 9, pp. 1–10, 1937.

[17] Guyton, A.C.; Hall, J. E. Tratado de fisiologia médica, 11. ed., Rio de Janeiro: Elsevier, 2006.

[18] Hans, M.; Schafer, R. *Lossless compression of digital audio*, Signal Processing Magazine, IEEE vol.18, no.4, pp.21 -32., Jul 2001.

[19] Haykin, S. Sistemas de comunicação – Analógicos e digitais, 4. ed. Porto Alegre: Bookman, 2004.

[20] Herre, J. From *joint stereo to spatial audio coding - Recent progress and standardization*, *Proc. Of the 7th Int. Conference on Digital Audio Effects (DAFx'04), Naples, Italy, October 5-8,* 2004.

[21] ISO/IEC 11172-3 *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*, 1993.

[22] ISO/IEC 13818-1 *Information technology – Generic coding of moving pictures and associated audio information: Systems*, 2007.

[23] ISO/IEC 13818-7 *Information technology, generic coding of moving pictures and associated audio information. Part 7: Advanced audio coding (AAC)*, 1997.

[24] ISO/IEC 14496-1 *Information technology – Coding of audio-visual objects – part 1: Systems*, 1998.

[25] ISO/IEC 14496-3 *Information technology – Coding of audio-visual objects – part 3: Audio*, 2005.

[26] ISO/IEC 14496-5 *Information technology – Coding of audio-visual objects – part 5: Reference software*, 2001.

[27] Jacklin, M. MPEG-4 – *The Media Standard. The landscape of advanced multimedia coding, MPEG-4 Industry Forum, Nov 2002* [Online]. Disponível em: <http://www.m4if.org/public/documents/vault/m4-out-20027.pdf>

[28] Jayant, P.; Noll, P. *Digital Coding of Waveforms – Principles and Applications to Speech and Video*, Prentice-Hall, 1984.

[29] Johnston, J. D. *Estimation of perceptual entropy using noise masking criteria*, Proc. ICASSP-88, pp. 2524-2527, Apr 1988.

[30] Johnston, J. D. *Transform coding of audio signals using perceptual noise criteria,* IEEE *J. Sel. Areas in Comm.*, vol. 6, no. 2, pp. 314–323, Feb.1988.

[31] Johnston, J. D.; Ferreira, A. J. *Sum-Diference Stereo Transform Coding, Proc. ICASSP,* pp.569-571, *May* 1992.

[32] Johnston, J. D.; Herre, J.; Davis M.; Gbur, U. *MPEG-2 NBC audio-stereo and multichannel coding methods, in Proc. 101st Audio Engineering Society Convention, Los Angeles, Calif, USA, November 1996, AES preprint 4383.*

[33] Leite, S. B. Melhoria do codificador de fala G.722.1 através do uso de um modelo perceptual Dissertação (Mestrado) - UNICAMP, Campinas, Dez 2003.

[34] Painter, T.; Spanias, A. *Perceptual coding of digital audio, Proceedings of the IEEE*, vol.88, no.4, pp.451-515, Apr 2000.

[35] Pereira, F.; Ebrahimi, T. *The MPEG-4 Book*. Upper Saddle River: Pearson, 2002.

[36] *PSYCHOACOUSTICS*. In *Wikipedia, The Free Encyclopedia*, *Florida*: *Wikimedia Foundation*, 2011 [Online]. Disponível em: <http://en.wikipedia.org/wiki/Psychoacoustic_model>. Acesso em: 12 Ago 2011.

[37] Purnhagen, H. *Low complexity parametric stereo coding in MPEG-4*, 7th Int. Conf. on Audio Effects (DAFX-04), Naples, Italy, Oct. 2004.

[38] Quackenbush, S. R.; Johnston, J. D. *Noiseless Coding of Quantized Spectral Components in MPEG-2 Advanced Audio Coding*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. New Paltz: IEEE. 1997. p. 19-22.

[39] Ranjani, H. G.; Kalagi, A. *Algorithmic delay and synchronization in MPEG audio codecs Ittiam Systems Pvt Ltd.* Disponível em: <*http://www.eetimes.com/design/audio-design/4015911/Algorithmic-delay-and-synchronization-in-MPEG-audio-codecs?pageNumber=2*>. *Acesso em: 10 set. 2010.*

[40] Robinson, D. J. M. *The Human Auditory System*, notas de aula, University of Essex, Colchester, Essex.

[41] Schroeder, M.; Atal, B. S.; Hall, J. L. *Optimizing digital speech coders by exploiting masking properties of the human ear*, *J. Acoust. Soc. Amer.*, pp. 1647-1652, Dec 1979.

[42] Schuijers, E.; Breebaart, J.; Purnhagen, H.; Engdegard, J. *Low Complexity Parametric Stereo Coding*, *Audio Engineering Society Convention* 116*,* 2004.

[43] Spanias, A.; Painter, T.; Atti, V. *Audio signal processing and coding*, Hoboken, NJ: Wiley-Interscience, 2007.

[44] Stevens, S. S.; Volkmann, J.; Newman, E. B. *A Scale for the Measurement of the Psychological Magnitude Pitch, J. Acoust. Soc. Amer, Volume* 8, *Issue* 3, pp. 185-190, 1937.

[45] Stolfi, G. Percepção Auditiva e Compressão de Áudio. In: *Princípios de Televisão Digital*. (Apostila do Curso de Televisão Digital). São Paulo: Mackenzie, 2006.

[46] Vander, A. J.; Sherman, J. H.; Luciano, D. S. *Human Physiology: The Mechanism of Body Function, 8th ed.*, Boston: McGraw-Hill, 2001.

[47] Vilela, A. L. M. Anatomia e Fisiologia humana. [Online]. Disponível em: <http://www.afh.bio.br/basicos/Sentidos3.htm>. Acesso em: 20 jun. 2011.

[48] Yamada, F.; Bedicks, G. Esquema de modulação do Sistema Brasileiro de TV Digital, Revista da SET nº03, Março 2010.

[49] Yang, D. T.; Kyriakakis, C.; C.-C. Jay Kuo, *High-Fidelity Multichannel Audio Coding*, New York: Hindawi, 2004.

[50] Zwicker, E.; Fastl, H. *Psychoacoustics: facts and models*, *Springer-Verlag, New York*, 1990.