
Universidade de Aveiro Departamento de Electrónica e Telecomunicações, **2004**

João Manuel de Oliveira e Silva Rodrigues

Codificação Digital de Áudio Baseada em Retroadaptação Perceptual

João Manuel de Oliveira e Silva Rodrigues

Codificação Digital de Áudio Baseada em Retroadaptação Perceptual

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica da Dr.ª Ana Maria Perfeito Tomé, Professora Associada do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, e do Dr. Tomás António Mendes Oliveira e Silva, Professor Associado do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

o júri / the jury

presidente / president

Carlos Alberto Diogo Soares Borrego
Professor Catedrático da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Luís António Serralva Vieira de Sá
Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Francisco António Cardoso Vaz
Professor Catedrático da Universidade de Aveiro

Ana Maria Perfeito Tomé
Professora Associada da Universidade de Aveiro (Orientadora)

Tomás António Mendes Oliveira e Silva
Professor Associado da Universidade de Aveiro (Co-orientador)

Aníbal João de Sousa Ferreira
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Antonio Pena Giménez
Profesor Titular do Departamento de Teoría do Sinal e Comunicacións da Universidade de Vigo

agradecimentos / acknowledgements

Aos orientadores deste trabalho, agradeço a confiança que depositaram em mim e o apoio que sempre me prestaram. Foi particularmente útil a selecção de artigos que foram fazendo, poupando-me certamente muitos dias de pesquisa bibliográfica. Ao Prof. Aníbal Ferreira agradeço a inspiração dada pelos seus trabalhos no domínio da codificação de áudio, que foram em boa medida a plataforma de lançamento e a motivação principal para o meu próprio trabalho neste domínio. A extensa bibliografia que me forneceu e a permanente disponibilidade para esclarecer dúvidas e discutir ideias foram inestimáveis. Ao Paulo Jorge Ferreira, José Vieira, Miguel Oliveira e Silva, e a muitos outros colegas do Departamento e outros convivas dos almoços, agradeço a discussão de temas relacionados directa ou indirectamente com este trabalho e, mais ainda, a discussão de temas *nada* relacionados com este trabalho, indispensáveis à manutenção de um equilíbrio mental minimanente satisfatório. Agradeço a todos os ouvintes que suportaram os testes de avaliação de qualidade e de comparação de codificadores descritos neste trabalho. Agradeço à Universidade de Aveiro, ao Departamento de Electrónica e Telecomunicações e ao IEETA pelos meios que colocaram à minha disposição. Finalmente, agradeço o apoio financeiro do Programa PRODEP III.

Resumo

Faz-se uma análise do problema da codificação digital de sinais áudio de alta qualidade e identifica-se o princípio de codificação perceptual como a solução mais satisfatória. Apresenta-se uma síntese dos sistemas de codificação perceptual encontrados na literatura, e identificam-se, comparam-se e relacionam-se as técnicas usadas em cada um. Pela sua relevância para a codificação de áudio, faz-se um estudo mais aprofundado das transformadas e bancos de filtros multifrequência, da quantização, dos códigos reversíveis e dos modelos matemáticos da percepção auditiva. Propõe-se um sistema de codificação composto por um banco de filtros multi-resolução, quantizadores logarítmicos adaptativos, codificação aritmética, e um modelo psicoacústico explícito para adaptar os quantizadores de acordo com critérios perceptuais. Ao contrário de outros codificadores perceptuais, o sistema proposto é retroadaptativo, isto é: a adaptação depende exclusivamente de amostras já quantizadas, e não do sinal original. Discutimos as vantagens do uso de retroadaptação e mostramos que esta técnica pode ser aplicada com sucesso à codificação perceptual.

Abstract

The problem of digital coding of high quality audio signals is analised, and the principles of perceptual coding are identified as the most satisfactory approach. We present a synthesis of the perceptual coding systems found in the literature, and we identify, compare and relate the techniques used in each one. Given their relevance for audio coding, transforms and multifrequency filter banks as well as quantization, lossless coding, and mathematical models of auditory perception are subject to a more thorough study. We propose a coding system consisting of a multirate filter bank, logarithmic quantizers, arithmetic entropy coding and an explicit psychoacoustic model to adapt the quantization according to perceptual considerations. Unlike other perceptual coders, the proposed system is backward-adaptive, that is: adaptation depends exclusively on already quantized samples, not on the original signal. We discuss the advantages of backward-adaptation and show that it can be successfully applied to perceptual coding.

À Marianinha

Conteúdo

Lista de Abreviaturas

ADPCM	*Adaptive Differential Pulse Code Modulation*
AQB	*Adaptive Quantization with Backward estimation*
AQF	*Adaptive Quantization with Forward estimation*
ASPEC	*Adaptive Spectral Perceptual Entropy Coding*
ATC	*Adaptive Transform Coding*
ATM	*Asynchronous Transfer Mode*
ATRAC	*Adaptive Transform Acoustic Coding*
BAPAC	*Backward-Adaptive Perceptual Audio Coding*
BFP	*Block Floating Point*
CD	*Compact Disc*
CELP	*Code-Excited Linear Prediction*
CQF	*Conjugate Quadrature Filters*
DAB	*Digital Audio Broadcasting*
DAT	*Digital Audio Tape*
DCT	*Discrete Cosine Transform*
DCT-IV	DCT tipo IV
DFT	*Discrete Fourier Transform*
DPCM	*Differential Pulse Code Modulation*
D*PCM	DPCM em malha aberta
DTWT	*Discrete-Time Wavelet Transform*
ELT	*Extended Lapped Transform*
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
HLT	*Hierarchical Lapped Transform*
IDCT	*Inverse* DCT
IELT	*Inverse* ELT
IIR	*Infinite Impulse Response*
LC-ATC	*Low Complexity* ATC

LOT	*Lapped Orthogonal Transform*
LPTV	*Linear Periodically Time Varying*
LQ	Laplaciana Quantizada
LSB	*Least Significant Bit*
LTI	*Linear Time Invariant*
MDCT	*Modified* DCT
MLT	*Modulated Lapped Transform*
MOS	*Mean Opinion Score*
MPEG	*Moving Pictures Expert Group*
MSC	*Multiple Adaptive Spectral Audio Coding*
MUSICAM	*Masking-pattern Universal Sub-band Integrated Coding And Multiplexing*
NFC	*Noise Feedback Coding*
NMT	*Noise Masking Tone*
OCF	*Optimum Coding in the Frequency domain*
PAC-AQB	*Perceptual Audio Coder with Backward-Adaptive Quantization*
PAC-AQF	*Perceptual Audio Coder with Forward-Adaptive Quantization*
PCM	*Pulse Code Modulation*
PR	*Perfect Reconstruction*
PXFM	*Perceptual Transform Coding*
QMF	*Quadrature Mirror Filters*
RDIS	Rede Digital com Integração de Serviços
SAM	*Split-and-Merge*
SFM	*Spectral Flatness Measure*
SIL	*Sound Intensity Level*
SMR	*Signal-to-Mask Ratio*
SNR	*Signal-to-Noise Ratio*
SNRSEG	*Segmental* SNR
SPL	*Sound Pressure Level*
TDAC	*Time Domain Aliasing Cancelation*
TMN	*Tone Masking Noise*
VLC	*Variable Length Coding*
VXC	*Vector Excitation Coding*

Capítulo 1

Introdução

Nos últimos anos e particularmente durante a última década, registou-se uma forte actividade na área de codificação de sinais áudio. Hoje em dia é amplamente reconhecido que para conseguir alta qualidade com débitos muito baixos, é necessário explorar as limitações da percepção auditiva humana. Isto conduziu ao desenvolvimento dos sistemas de codificação perceptual: codificadores que aplicam conhecimentos da Psicoacústica de forma a reduzir ou eliminar a audibilidade da distorção introduzida. A maioria dos codificadores perceptuais actuais partilham a mesma estrutura genérica: são codificadores que operam no domínio da frequência com quantização adaptativa, e a adaptação é feita por um algoritmo que inclui um modelo perceptual. Os melhores sistemas conseguem codificação perceptualmente "transparente" com débitos inferiores a 2 bits por amostra. Entretanto, a cada vez maior diversificação de aplicações, de meios de armazenamento e de transmissão impõe novas exigências aos codificadores de áudio, tais como: atingir débitos ainda mais baixos, por exemplo para transmissão em telefones celulares; reduzir os atrasos de codificação, para aplicações bidireccionais; permitir o acesso aos sinais a diferentes níveis de qualidade, para transmissão progressiva; ou possibilitar a manipulação dos sinais directamente na forma codificada.

Neste trabalho fazemos uma caracterização genérica dos algoritmos e técnicas de codificação perceptual de áudio existentes, com o intuito de identificar vias de investigação que possam conduzir a melhores desempenhos em futuros sistemas de codificação.

1.1 Sistemas de Codificação

1.1.1 Modelo Geral de um Sistema de Codificação

A Figura 1.1 representa um sistema de codificação genérico. A *fonte* é um processo qualquer que produz mensagens que serão transmitidas através do sistema. Não se conhecem *a priori* as mensagens que a fonte vai produzir, mas sabe-se que serão retiradas de um conjunto possivelmente infinito de mensagens com determinadas características e de acordo com certas probabilidades de ocorrência. O *receptor* é o destinatário da informação que passa pelo sistema e cabe-lhe fazer uso e dar significado a essa informação. A função do *canal digital* é o transporte da informação entre os extremos do sistema, com a máxima fidelidade possível. Os extremos podem estar separados no espaço (transmissão) ou no tempo (armazenamento). O canal digital inclui o suporte físico da comunicação bem como os equipamentos de interface entre o domínio digital e o meio físico (modulador e desmodulador). O canal é o elemento do sistema de codificação mais vulnerável à presença de *ruído* e a caracterização estatística

dos erros de transmissão resultantes da influência do ruído é um aspecto crucial no projecto do sistema. Outra característica importante é a quantidade de informação que o canal pode transmitir por unidade de tempo, chamada *capacidade* do canal. O *codificador* tem a função de converter as mensagens fornecidas pela fonte para uma forma adaptada às características e limitações do canal. Em particular, o *débito*[1] do codificador tem que ser inferior à capacidade do canal. A função do *descodificador* é tentar reconstruir a mensagem original a partir dos dados (possivelmente adulterados) recebidos do canal e transformá-la numa forma apropriada para o receptor.

Figura 1.1: Um sistema de codificação genérico.

1.1.2 Níveis de Codificação

Na definição genérica de sistema de codificação dada atrás, podem enquadrar-se diversas classes particulares de técnicas de codificação, a saber: a criptografia, os algoritmos de compressão, os códigos de protecção contra erros. A distinção entre estas três disciplinas reside na natureza dos problemas que procuram resolver e nos diferentes objectivos que tentam atingir.

- A criptografia é usada para a transmissão de mensagens secretas ou privadas através de um canal pouco seguro. O requisito principal de um sistema de criptografia é que a decifração de mensagens por uma pessoa não autorizada seja muito difícil. Existem muitas técnicas de criptografia com diversos graus de segurança e complexidade. A técnica de *stream ciphering*, por exemplo, permite uma segurança razoável, é simples, praticamente instantânea e não aumenta a quantidade de informação a transmitir [165].

- As técnicas de protecção contra erros procuram garantir uma comunicação fiável através de um canal ruidoso. Em geral baseiam-se na introdução criteriosa de redundância sob a forma de bits adicionais que aumentam as distâncias de Hamming entre as mensagens válidas [61]. Quando chega uma mensagem inválida, o descodificador fica a saber que ocorreu um erro e pode até tentar corrigi-lo. O objectivo é a detecção e eventual correcção de um grande número de erros com um *overhead* mínimo. O resultado é um novo canal com uma taxa de erros muito menor, à custa de uma pequena redução da capacidade disponível.

- Os algoritmos de compressão procuram reduzir o ritmo de transmissão necessário para representar um sinal digital, tentando simultaneamente limitar a distorção eventualmente introduzida nesse processo. O seu propósito é proporcionar uma utilização eficiente da capacidade do canal. Nestes sistemas há sempre um compromisso entre a taxa de compressão e a qualidade do sinal recuperado.

Na prática, um sistema de codificação completo combina diversos processos de codificação em níveis de pormenor diferentes. Por exemplo, um dado codificador pode incluir: um algoritmo de compressão para reduzir a quantidade de bits a transmitir; técnicas de cifragem

[1] O número de bits debitados por unidade de tempo, também chamado taxa ou ritmo de transmissão, ou ainda *bit rate*.

para garantir privacidade; e códigos para detecção e correcção de erros no canal. Os diversos componentes são ligados em cascata, como mostra a Figura 1.2, criando uma hierarquia de codificação. Esta estrutura tem a vantagem de o projecto de cada nível ser praticamente independente dos restantes. Assim, o projecto do sistema de criptografia, por exemplo, pode partir do princípio que o nível inferior é um canal fiável com probabilidade de erro muito baixa. O nível de compressão, por sua vez, pode presumir que dispõe de um canal simultaneamente fiável e seguro.

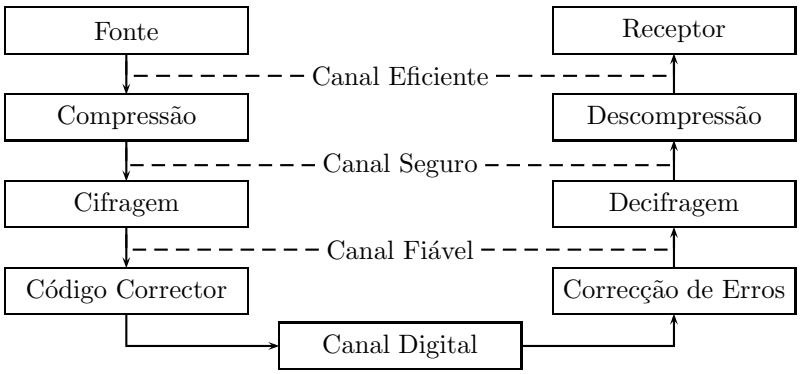


Figura 1.2: Um sistema de codificação com compressão, criptografia e correcção de erros. Cada nível de codificação corrige um defeito do nível abaixo e fornece um canal melhorado ao nível seguinte: a correcção de erros fornece um canal mais fiável, a cifragem torna-o mais seguro, e a compressão melhora a sua eficiência.

1.1.3 Compressão de Sinais

Nesta subsecção procura-se fazer uma abordagem teórica ao problema geral da compressão de dados e sinais, tendo presente que é aí que se situam as contribuições desta dissertação. A partir deste ponto passaremos a confundir frequentemente os termos *codificação* e *compressão*, visto que pomos de lado os outros níveis de codificação já referidos.

Para conseguir compressão de sinais ou dados digitais pode proceder-se segundo duas direcções distintas: aproveitamento da estrutura estatística da produção de sinais na fonte (extracção de *redundância*), ou exploração das limitações de sensibilidade do receptor (redução de *irrelevância*). Vejamos em que consiste cada um destes processos:

Extracção de Redundância De uma forma geral, a fonte não produz sinais arbitrariamente complexos; alguns sinais são mais prováveis do que outros. Observando o sinal de música gravado num CD, por exemplo, verifica-se que as amostras codificadas raramente atingem a amplitude máxima e que muitas vezes é possível prevê-las com alguma confiança a partir de outras amostras. A distribuição não uniforme de amplitudes e a previsibilidade das amostras são exemplos de conteúdo estruturado do sinal que revelam que o código usado é *redundante*: gasta mais bits do que seria necessário. Isto quer dizer que é possível aproveitar a estrutura do sinal para o codificar com menor redundância, usando técnicas como os códigos de comprimento variável ou a codificação diferencial, por exemplo. Se houver um modelo completo e preciso da fonte, pode extrair-se muita redundância do sinal e conseguir assim uma boa taxa de compressão.

Redução de Irrelevância Por vezes, o receptor não é capaz de distinguir entre um sinal codificado e o sinal original embora eles sejam efectivamente diferentes. Isso acontece porque a sensibilidade limitada do receptor torna muito difícil ou mesmo impossível a percepção das diferenças entre os dois sinais. Se o nível de distorção introduzido pelo codificador estiver inutilmente abaixo do limiar de perceptibilidade, então diz-se que o sinal codificado contém informação *irrelevante*. É possível poupar bits, reduzindo a margem de irrelevância, sem prejuízo para o receptor. A filtragem de sinais áudio eliminando componentes acima de 20 kHz pode ser considerada uma operação de redução de irrelevância: o sinal é modificado mas a distorção introduzida é inaudível. A quantização é outro exemplo: o erro introduzido é irrelevante se o passo de quantização for inferior à mínima diferença de amplitude detectável.

A Figura 1.3, que representa o plano distorção-débito de um sistema de compressão, deverá clarificar estes conceitos. O eixo horizontal representa o débito ou ritmo de transmissão, R, medido em bits por unidade de tempo. O eixo vertical representa D, uma medida de distorção que traduza significativamente a sensibilidade do receptor. A curva decrescente, $D(R)$, representa a função distorção-débito ideal do sistema [14][80, Appendix D]. O ponto C_0 representa o sinal original (já digital), com ritmo R_0 e distorção D_0.[2] R_A e D_P são as especificações de projecto do sistema de codificação: R_A é o débito máximo admissível para transmissão pelo canal digital; D_P é o *limiar de perceptibilidade* ou distorção máxima admissível pelo receptor. Pretende-se um codificador C' que satisfaça as especificações:

$$R' < R_A \ \wedge \ D' < D_P.$$

A região sombreada representa o conjunto solução do problema. O ritmo mínimo que permite codificar o sinal sem distorção adicional é a chamada *entropia* da fonte, $H = R(D_0)$. O ritmo mínimo que se pode atingir sem superar o limiar de perceptibilidade, $H_P = R(D_P)$, é a chamada *entropia perceptual*. A *redundância* do sinal original é medida pela diferença $R_0 - H$. A *irrelevância* é dada por $D_P - D_0$. Extrair redundância reflecte-se num deslocamento do ponto C' para a esquerda; reduzir irrelevância corresponde a um deslocamento para cima, em direcção ao limiar de perceptibilidade.

Num caso extremo, um algoritmo de compressão pode basear-se exclusivamente em extracção de redundância. É o caso dos códigos de comprimento variável, como os de Huffman, Lempel-Ziv, e outros, que permitem a compressão de dados digitais sem qualquer distorção.[3] Também é possível fazer pura redução de irrelevância, o que não resulta em compressão adicional mas pode aumentar a qualidade de codificação. Um exemplo disto é o sistema *Super Bit Mapping*, referido na Subsecção 2.3.2, que usa realimentação de ruído para minorar a distorção audível num típico sistema PCM de 16 bits. Num caso mais geral, o algoritmo de compressão deverá combinar processos de extracção de redundância e de redução de irrelevância, o que permite encontrar o melhor compromisso entre qualidade e taxa de compressão. A combinação mais adequada resulta da ponderação dos custos e benefícios de cada processo tendo em conta as limitações da fonte e do receptor.

O projecto de um sistema de codificação consiste em definir os blocos de codificação e descodificação de forma a cumprir determinadas especificações e satisfazer condições impostas pela fonte, pelo canal e pelo receptor. Por isso, é fundamental começar por caracterizar o

[2] Consideramos $D_0 > 0$ para contemplar a situação em que C_0 é uma versão digitalizada de um sinal originalmente analógico. Se a distorção for medida em relação ao sinal C_0, então $D_0 = 0$, naturalmente.

[3] Por isto também são chamadas de técnicas de codificação sem perdas (*lossless coding*).

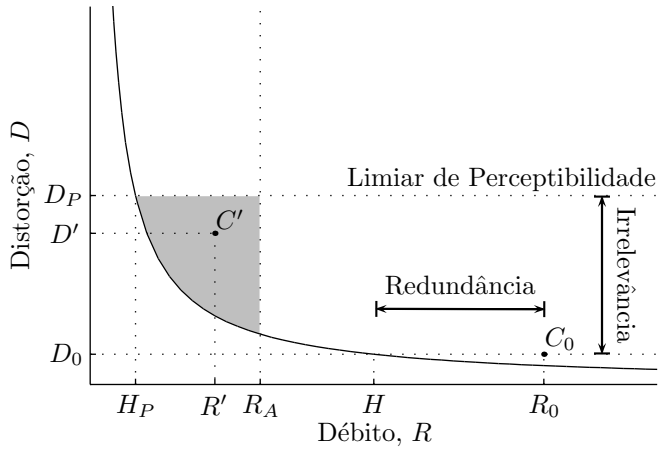


Figura 1.3: Função distorção-débito de um sistema de codificação.

sistema pretendido, identificando com rigor as fronteiras entre os vários blocos e os parâmetros essenciais que influenciam o desempenho global do sistema. É particularmente importante caracterizar os sinais a codificar, para avaliar a redundância que se pode extrair, e determinar os limites de percepção do receptor, para estabelecer a margem de irrelevância que se pode explorar.

Tendo em conta as necessidades da aplicação, o projectista tem que encontrar, para o sistema, o melhor compromisso entre diversos objectivos conflituosos: (1) baixo ritmo de transmissão; (2) alta qualidade do sinal recuperado, avaliada por um critério significativo para o receptor; (3) minimização do atraso global (muito importante em sistemas de comunicação bidireccional); (4) complexidade baixa, principalmente no descodificador; (5) robustez e degradação graciosa quando há erros de transmissão. Como em muitos problemas de engenharia, a melhor solução é atingida por um processo iterativo de aproximações sucessivas.

1.2 Organização da Dissertação

No Capítulo 2 faz-se uma abordagem ao problema da compressão de sinais áudio de alta qualidade tratado neste trabalho. Introduzem-se as técnicas de codificação perceptual e justifica-se a sua importância na codificação de áudio. Descrevem-se as principais tecnologias de codificação de sinais, juntamente com diversos exemplos de codificadores de áudio que as aplicam. É dado maior destaque às técnicas que operam no domínio da frequência, por serem as que têm conseguido melhores resultados em codificação perceptual.

O Capítulo 3 descreve as transformadas e os bancos de filtros multifrequência mais usados em codificadores perceptuais de áudio. O capítulo começa por definir alguns conceitos base da teoria dos sistemas multifrequência, e apresenta as características requeridas de um banco de filtros a aplicar num codificador perceptual de áudio. A seguir classifica os bancos de filtros uniformes com maior interesse prático e as estruturas eficientes de decomposição não uniforme que se podem construir a partir desses.

No Capítulo 4 revemos alguns aspectos sobre a análise, projecto e implementação de quantizadores, os elementos responsáveis pela redução de irrelevância.

No Capítulo 5 descrevemos algumas técnicas de codificação sem perdas, responsáveis pela extracção de redundância.

O Capítulo 6 é dedicado ao estudo do receptor que nos interessa, a audição. Descreve o essencial da anatomia e fisiologia do ouvido, resume os resultados da Psicoacústica com maior relevância para a codificação de áudio e apresenta modelos matemáticos da percepção auditiva humana.

No Capítulo 7 apresentamos uma estrutura de codificação perceptual de áudio que aplica uma estratégia de quantização retroadaptativa. Analisamos uma implementação e as opções tomadas no seu projecto. Como se trata de uma abordagem pouco usual, discutimos as vantagens da retroadaptação e avaliamos a sua viabilidade em codificadores perceptuais.

As conclusões desta dissertação, bem como possíveis direcções de trabalho futuro, serão apresentadas no Capítulo 8.

Capítulo 2

Tecnologias para Codificação Digital de Áudio

Neste capítulo procuramos descrever e caracterizar os processos de codificação mais relevantes no contexto da codificação de sinais áudio de alta qualidade. Deste estudo resultará a escolha das técnicas consideradas mais apropriadas, o que constitui uma importante decisão de projecto do codificador descrito no Capítulo 7. A Secção 2.1 aborda o problema específico da compressão de sinais áudio e aí surge a motivação para as técnicas de codificação perceptual. As secções restantes descrevem várias tecnologias de codificação aplicáveis aos sinais áudio, e apresentam exemplos encontrados na literatura.

2.1 Compressão de Sinais Áudio

2.1.1 Especificação do Problema

O problema tratado neste trabalho é o desenvolvimento de sistemas de compressão de sinais áudio para baixos débitos sem perda significativa de qualidade (quando avaliada por um ouvinte humano). A fonte que nos interessa considerar é uma qualquer fonte de sinal áudio monofónico em formato PCM de alta qualidade, por exemplo um sinal de um CD, amostrado ao ritmo de 44100 amostras por segundo e quantizado uniformemente com resolução de 16 bits. Desta forma, estamos a incluir na fonte as operações de discretização no tempo (*amostragem*) e discretização na amplitude (*quantização*), necessárias para converter um sinal analógico numa representação digital. Consideraremos que as condições do teorema da amostragem são respeitadas e que o sinal quantizado com 16 bits é indistinguível do original, pelo que daqui em diante só nos debruçaremos sobre operações executadas no domínio digital.

O receptor é o sistema auditivo humano. Podemos ainda incluir no receptor o sistema de reprodução de áudio que compreende a conversão D/A, a amplificação, a transdução electro-acústica (por auscultadores ou altifalantes) e a propagação da onda acústica até ao ouvido. Consideraremos que todos estes processos são lineares e têm uma resposta plana na gama de frequências audíveis[1] e que podem portanto ser caracterizados por um único parâmetro: o

[1]Estas condições dificilmente são respeitadas quando se usam altifalantes e a propagação acústica se faz numa sala reverberante. Mesmo com auscultadores surgem dificuldades devido a ressonâncias do canal auditivo, como foi observado em [146]. Apesar disso, em trabalhos de codificação de áudio, é comum partir-se destes pressupostos.

ganho acústico do sistema de reprodução.[2] Uma vez que o ganho do amplificador pode ser regulado pelo ouvinte, terá que se considerar uma situação extrema para o ganho acústico global.

Para o estudo que nos interessa fazer não é necessário especificar o canal com grande pormenor. Consideraremos simplesmente que o canal é um meio digital de transmissão ou armazenamento com capacidade bastante inferior ao débito da fonte, virtualmente livre de erros,[3] que será simulado por um arquivo no disco de um computador.

O codificador tem a função de reduzir suficientemente o ritmo binário do sinal para transmissão através do canal. A partir desta representação comprimida, o descodificador deve recuperar um sinal tão aproximado quanto possível do original, em termos perceptuais. Simultaneamente, o atraso global introduzido pelo sistema e a sua complexidade devem ser mantidos dentro dos limites aceitáveis para a aplicação em vista. A robustez não será uma preocupação uma vez que se assume a transparência do canal.

2.1.2 Motivação para as Técnicas de Codificação Perceptual

As técnicas tradicionais de codificação de sinal baseiam-se num bom conhecimento das características de geração dos sinais para explorarem ao máximo a redundância da fonte,[4] fazendo uso quer de modelos explícitos da fonte, quer de modelos do comportamento estatístico dos sinais. A compressão de sinais de voz de banda estreita é um problema muito estudado, onde as técnicas tradicionais de codificação têm sido aplicadas com bastante sucesso. É natural portanto, que as primeiras propostas para compressão de áudio de qualidade se inspirassem em algoritmos semelhantes. Contudo, uma comparação dos problemas de codificação de áudio de banda larga e de banda estreita, observando diversas grandezas estatísticas de sinais reais, em particular a medida de planura espectral, permite concluir que as técnicas de codificação de fonte usadas para voz não resultam em grande compressão quando aplicadas a sinais áudio de banda larga [84].[5]

Há várias razões para esta disparidade entre sistemas de codificação de áudio e de voz. Por um lado, existem diferenças substanciais entre as *fontes* de áudio e de voz: os sinais de áudio envolvem frequências de amostragem muito superiores, melhor resolução em amplitude, maior gama dinâmica, e sobretudo, uma enorme diversidade de sinais com grandes variações de conteúdo espectral e temporal. Estas características dificultam o desenvolvimento de um modelo geral para os sinais áudio, o que limita os ganhos que se podem obter por extracção de redundância. Por outro lado, há também diferenças no *receptor*: os mecanismos de percepção de voz e de música diferem, e as expectativas de qualidade por parte dos ouvintes de áudio são muito superiores. Além disso, actualmente está perfeitamente demonstrado que as medidas de distorção tipicamente usadas para avaliar e optimizar os codificadores de sinal tradicionais, como a relação sinal-ruído ou o erro médio quadrático, não reflectem adequadamente a quali-

[2] O ganho acústico pode especificar-se através da razão entre a pressão sonora de um som e a amplitude do sinal digital que o representa.

[3] Esta suposição é muito conveniente numa primeira abordagem ao projecto de um sistema de compressão e, de qualquer forma, a sua inadequação pode ser minorada pelo recurso às técnicas de detecção e correcção de erros.

[4] Por esta razão, também são chamadas de técnicas de *codificação de fonte*.

[5] A medida de planura espectral ou *spectral flatness measure* é uma medida da previsibilidade linear do sinal e estabelece um limite teórico ao ganho de codificação que pode ser atingido usando codificadores por transformada ou predição linear [80, Section 2.3.7]. (O ganho de codificação é definido como o aumento de SNR que um dado código proporciona em relação ao PCM com o mesmo débito.)

dade percebida pelo ouvido humano. Um exemplo disto é o caso de um sinal referido em [84] que pode ser codificado de forma perceptualmente transparente com uma relação sinal-ruído de apenas 13 dB quando o ruído é inserido de acordo com princípios da Psicoacústica, mas que não é transparente na presença de ruído branco mesmo com uma margem de 60 dB.

A dificuldade de modelação dos sinais áudio, e a constatação da inadequação das técnicas tradicionais baseadas em *codificação de fonte*, levaram à procura de algoritmos alternativos que permitissem explorar melhor as limitações do receptor—o ouvido humano. Procura-se assim ganhar com redução de irrelevância o que não se consegue com extracção de redundância. É neste princípio que se baseiam as modernas técnicas de *codificação perceptual*, que se servem do conhecimento dos limites de sensibilidade do receptor para modelar a distorção introduzida de forma a minimizar a sua perceptibilidade.

2.2 Codificação Perceptual no Domínio da Frequência

O mascaramento de ruído por um som mais forte (ver Capítulo 6) parece ser o fenómeno psicoacústico mais relevante no contexto da codificação de áudio, e tem sido o mais explorado nos actuais codificadores perceptuais. Uma vez que esta e outras propriedades auditivas importantes são geralmente descritas no domínio da frequência, parece natural controlar a introdução de ruído de quantização também neste domínio. Talvez por esta razão, a maioria dos codificadores perceptuais existentes enquadra-se na categoria de codificação por sub-bandas ou por transformada. Estes codificadores no domínio da frequência têm a estrutura genérica representada na Figura 2.1, que descrevemos a seguir:

- O bloco de análise ou decomposição consiste num banco de filtros que decompõe o sinal de entrada num conjunto de canais associados a diferentes bandas de frequência. Cada filtro é seguido por um decimador para reduzir o número de amostras a quantizar. Em vez de bancos de filtros multifrequência, é equivalente usar transformadas lineares, aplicadas bloco-a-bloco sobre vectores de amostras acumuladas em *buffers*.

- Os coeficientes do banco de decomposição ou de uma análise espectral separada são usados para estimar o limiar de mascaramento usando um modelo psicoacústico. Desta informação é derivada uma repartição de bits ou de ruído pelas várias bandas que minimize a distorção audível e o débito do sistema. O modelo é usado explícita ou implicitamente e pode ser mais ou menos completo.

- As amostras de cada sub-banda são quantizadas e codificadas com maior ou menor resolução de acordo com as indicações do limiar de mascaramento calculado e as restrições ao débito do sistema.

- Os dados codificados são multiplexados com informação secundária necessária para a descodificação e produzem uma sequência binária para transmissão ou armazenamento. Neste passo pode também adicionar-se protecção contra erros, sequências para sincronização e informações diversas.

No descodificador, a informação é desmultiplexada e as amostras de cada banda são recuperadas e introduzidas no banco de síntese ou reconstrução que faz uma interpolação das sub-bandas e as recombina num único sinal.

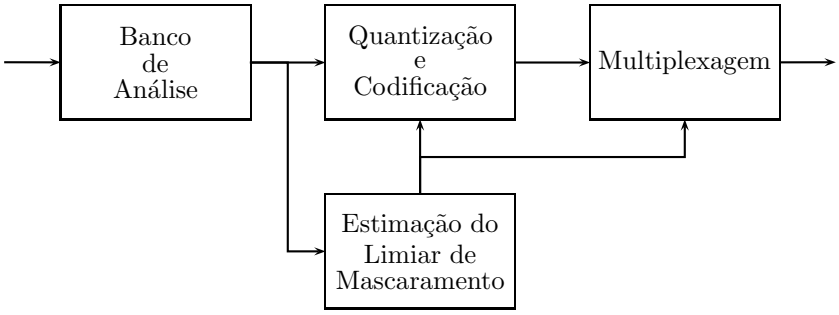


Figura 2.1: Diagrama de blocos genérico de um codificador perceptual a trabalhar no domínio da frequência.

2.2.1 Bancos de Filtros usados em Codificadores Perceptuais

A escolha do banco de filtros ou da transformada a usar é uma das decisões mais importantes no projecto deste tipo de codificadores devido à grande influência que vai ter no desempenho global do sistema de codificação. Esta decisão deve ser fundamentada na avaliação de diversas características dos bancos de filtros: invertibilidade (se permite *reconstrução perfeita* ou não), factor de subamostragem (crítica ou subcrítica), resolução temporal, resolução espectral, selectividade na frequência, atraso global, e complexidade computacional.

Como se verá nos exemplos apresentados adiante, tem sido ensaiado um grande número de bancos de filtros alternativos, abrangendo: bancos uniformes como as transformadas em blocos, transformadas com sobreposição, filtros pseudo-QMF; e bancos não-uniformes com estruturas em árvore de filtros QMF ou CQF, transformadas hierárquicas e sistemas híbridos.

Deixa-se para o Capítulo 3 um estudo mais aprofundado das propriedades dos bancos de filtros multifrequência e das transformadas mais importantes para a compressão perceptual de áudio.

2.2.2 Atribuição de Bits ou Ruído

A atribuição de bits ou ruído pelos quantizadores das várias bandas determina a qualidade e o débito do sistema de codificação. Um codificador perceptual faz esta atribuição guiando-se pelo limiar de mascaramento do sinal de forma a maximizar a qualidade perceptível. O cálculo do limiar de mascaramento baseia-se numa estimação espectral que pode ser dada pelo próprio banco de decomposição do codificador ou, se este não tiver as características apropriadas, por uma análise espectral alternativa.

A seguir descrevem-se várias técnicas de estimação do limiar e atribuição de bits presentes em diversos algoritmos de codificação perceptual.

Atribuição Estática de Bits

Conhecendo a largura de cada banda do banco de análise e o índice de mascaramento em cada frequência, é possível calcular a relação sinal-ruído mínima necessária em cada banda crítica e estimar o número de bits a atribuir a cada quantizador para garantir esses valores de SNR. A distribuição fixa que resulta destas considerações coloca a maior parte dos bits nas baixas frequências onde as bandas críticas são mais estreitas.

Este esquema simples tem diversas desvantagens. Se se considerar a curva de mascaramento mais conservadora, medida para a situação de tons a mascarar ruído, torna-se necessário um débito muito elevado quando comparado com outras técnicas de atribuição de bits. Por outro lado, se for considerado apenas o índice de ruído a mascarar tons—menos exigente—então o codificador apresentará distorção audível para sinais tonais.

Atribuição de Bits Derivada da Envolvente Espectral do Sinal

Nesta técnica, parte dos bits são atribuídos estaticamente como no caso anterior, de modo a garantir o critério de mascaramento por ruído. Os bits que restam são repartidos de forma dinâmica como no codificador por transformada adaptativo (ATC) de Zelinski e Noll [164]: proporcionalmente ao logaritmo do espectro de potência do sinal. Sinais tonais, com picos espectrais salientes, são codificados com maior precisão reduzindo ou eliminando a audibilidade do ruído de quantização.

Modelação de Ruído segundo um Modelo Psicoacústico

Os métodos anteriores usam resultados simples da Psicoacústica implicitamente para que a distorção realmente introduzida tenha um espectro que aproxime o limiar de mascaramento. No entanto, essa aproximação é algo grosseira o que penaliza a eficiência do codificador. É possível conseguir uma aproximação melhor aplicando explicitamente um modelo das características psicoacústicas do ouvido. A partir da saída do banco de análise ou de um estimador espectral paralelo, mais preciso, o modelo estima o limiar de mascaramento real em cada bloco usando regras que quantificam os fenómenos de mascaramento intra- e inter-bandas críticas, mascaramento de tons por ruído e de ruído por tons, limiar absoluto e eventualmente outros fenómenos como os de mascaramento temporal.

O limiar calculado serve então como modelo do perfil de ruído a introduzir. Os passos de quantização são determinados usando uma estimativa da potência do erro injectado por cada quantizador (tipo $N = \Delta^2/12$ para quantizadores uniformes). Não há, portanto, uma atribuição de bits explícita e à partida não se sabe quantos bits serão usados. Por isso pode ser necessário iterar este processo dentro de um ciclo de controlo de débito.

Aplicação Directa do Limiar de Mascaramento

Neste método o limiar de mascaramento é calculado explicitamente de forma semelhante à técnica anterior mas não é usado para estimar os passos de quantização. O limiar é usado directamente num sistema de análise por síntese para comparação com o ruído realmente introduzido. O ruído é avaliado pela diferença, no domínio da frequência, entre o sinal original e o quantizado.

2.2.3 Quantização e Codificação

Em princípio, qualquer método de codificação temporal incluindo PCM e DPCM pode ser usado para quantizar e codificar as amostras de cada uma das bandas do banco de filtros. O quantizador pode ser uniforme, logarítmico ou optimizado para a distribuição de amplitudes do sinal (quantizador de Max [105]); o factor de escala ou passo de quantização pode ser adaptado dinamicamente a partir das amostras originais (proadaptação) ou das amostras quantizadas anteriormente (retroadaptação). A codificação pode usar uma representação

em código binário natural, vírgula flutuante em blocos ou códigos de comprimento variável. Dos vários métodos possíveis descrevemos a seguir alguns dos mais comuns em codificadores perceptuais.

Quantização Adaptativa a partir de Estimativa do Espectro de Potência

O passo de quantização é adaptado com base em estimativas da potência em cada linha do espectro. Estas estimativas são calculadas tanto no codificador como no descodificador por interpolação linear entre valores quantizados logaritmicamente da variância em cada banda. Este esquema foi proposto no ATC referido anteriormente.

Vírgula Flutuante em Blocos

Num sistema de *block floating point* (BFP), também chamado de *block companding*, os dados a transmitir são primeiro agrupados em blocos de amostras consecutivas da mesma banda ou de amostras de bandas adjacentes. A amplitude máxima dentro de um bloco é quantizada logaritmicamente e transmitida ao receptor. Este valor é usado como factor de escala para normalizar as amostras do bloco que passam depois por um quantizador uniforme. Estes valores normalizados podem ser interpretados como mantissas de uma representação em vírgula flutuante com um único expoente para todo o bloco, o que justifica a designação desta técnica.

Com este esquema elimina-se a possibilidade de saturação dos quantizadores que é o principal problema do método anterior. Outra vantagem deste sistema é que permite uma protecção contra erros selectiva, mais eficiente: se os factores de escala forem devidamente protegidos, qualquer erro que ocorra num bloco fica limitado em amplitude pelo respectivo factor de escala e a sua audibilidade será muito reduzida.

Códigos de Comprimento Variável e Ciclos de Controlo de Débito

O código binário natural, com um número de bits fixo para cada amostra, não explora eficientemente a distribuição de amplitude não uniforme das sequências em cada sub-banda. Para reduzir esta redundância intrínseca e ganhar alguma compressão adicional, alguns codificadores usam códigos de comprimento variável nomeadamente: códigos de Huffman, aritméticos e *run-length codes*.[6] Estes códigos traduzem um conjunto de símbolos (amostras) num número variável de bits. Por isso, para aplicações de débito constante é necessário que o codificador tenha um mecanismo de controlo de débito. O codificador começa por quantizar as componentes espectrais usando uma estimativa inicial dos passos de quantização derivada do modelo psicoacústico. Conta-se o número de bits necessários para codificar o bloco e se diferir do número de bits disponíveis, os passos de quantização são ajustados e o ciclo é repetido. Eventualmente o número de bits atinge um valor aceitável e os dados são finalmente transmitidos.

Análise-por-Síntese com Ciclo de Controlo de Distorção

Em vez de distribuir o ruído de quantização estimado por uma regra genérica, um sistema de análise-por-síntese mede o ruído realmente injectado e modela-o iterativamente. Este método tem a vantagem óbvia de controlar directamente a distorção introduzida mas implica um

[6] Outros códigos, como os de Lempel-Ziv, poderiam ser usados mas na literatura consultada não foi encontrada qualquer referência nesse sentido.

maior esforço computacional. Começa-se por quantizar os coeficientes espectrais de cada bloco do sinal usando um ciclo de controlo de débito. De seguida, o ruído de quantização real é determinado pela diferença entre os coeficientes espectrais quantizados e os originais e é comparado com o limiar de mascaramento estimado. Se o ruído medido exceder o limiar em alguma região do espectro, reduzem-se os passos de quantização correspondentes e repete-se o ciclo interno de quantização e ajuste do débito. Este processo é iterado de forma a minimizar a audibilidade do ruído introduzido. Quando se melhora a quantização em algumas bandas, as restantes têm tendência a piorar (para manter o mesmo débito) e por isso não é garantido que haja convergência para uma solução aceitável. Por outro lado, em cada iteração do ciclo externo de controlo de distorção há uma representação válida do sinal que satisfaz as limitações de débito e que pode ser transmitida se o tempo de processamento for esgotado.

2.2.4 Exemplos

Ao longo dos últimos anos foi proposto um grande número de algoritmos para codificação de sinais áudio de alta qualidade. A maioria desses sistemas segue o esquema genérico de codificação perceptual no domínio da frequência. Nesta secção apresentamos alguns dos elementos mais representativos desta classe de codificadores.

Multiple Adaptive Spectral Audio Coding (MSC)

Este foi talvez o primeiro codificador por transformada a usar explicitamente resultados da Psicoacústica (ver [135]). A decomposição espectral é feita por uma FFT de 1024 pontos, com sobreposição de 1/16 e uma janela sinusoidal nos segmentos sobrepostos. A sobreposição destina-se a reduzir os artefactos nas transições dos blocos mas implica decimação subcrítica. Os coeficientes da FFT são convertidos para uma representação polar, em amplitude e fase. O espectro é dividido em 26 grupos correspondendo às bandas críticas do sistema auditivo e, em cada grupo, calcula-se a gama dinâmica e a energia do sinal. A forma global do espectro é determinada e codificada como informação secundária. Em cada grupo é calculado o limiar de mascaramento e anulam-se quaisquer valores de amplitude que lhe estejam abaixo.

A quantização é dividida em dois estágios. Primeiro faz-se uma quantização grosseira (até 2 bits) de cada valor de amplitude e fase de acordo com a gama dinâmica de cada grupo. Quando a amplitude é zero, a fase não é transmitida, o que permite poupar alguns bits. No segundo estágio, os bits que restam são distribuídos pelas linhas do espectro com maior relevância psicoacústica que sofrem então uma quantização fina. A atribuição de bits é baseada na saída do primeiro estágio de quantização, pelo que o receptor a pode reproduzir sem necessidade de mais informação secundária.

Testes realizados com sinais de CD codificados a 128 kbit/s por canal (2.9 bits por amostra) revelaram, segundo os seus autores, uma qualidade subjectiva "excelente", não se distinguindo o sinal processado do original excepto em alguns trechos mais críticos.

Low-Complexity Adaptive Transform Coding (LC-ATC)

Esta técnica, proposta em [20], resultou da adaptação às características da audição do ATC clássico de Zelinski e Noll. Conseguiu-se simultaneamente uma redução de complexidade. A filtragem baseia-se numa MDCT de 512 pontos. Esta transformada é invertível como a FFT mas tem a vantagem de reduzir muito os artefactos nas transições entre blocos mantendo, ao mesmo tempo, a condição de decimação máxima. As componentes espectrais são agrupadas

em 46 bandas com largura subcrítica. Os máximos em cada banda são quantizados logaritmicamente e transmitidos ao receptor. Tanto o emissor como o receptor estimam a envolvente do espectro de potência através de uma simples interpolação desses valores. Uma atribuição de bits fixa, derivada de curvas de SMR para ruído de banda estreita, garante uma qualidade mínima, enquanto bits adicionais são distribuídos proporcionalmente ao logaritmo do espectro de potência, melhorando a qualidade de reprodução de sinais tonais. Finalmente, os coeficientes espectrais são quantizados usando a técnica de vírgula flutuante em blocos. Como os máximos são calculados a partir da envolvente espectral, não é necessária mais informação secundária.

Optimum Coding in the Frequency Domain (OCF)

Este codificador, apresentado em [19], foi precursor de várias técnicas importantes em codificação perceptual de áudio. Baseia-se numa transformada discreta de cossenos de 512 pontos para obter uma representação espectral do sinal. (Uma versão posterior passou a usar uma MDCT com janela sinusoidal.) A quantização e codificação de cada bloco é feita repetidamente no interior de dois ciclos, à procura da melhor codificação possível. O ciclo interior, de controlo do débito, tem semelhanças com o do codificador PXFM, descrito a seguir. Os coeficientes de cada bloco atravessam um quantizador não uniforme e são codificados com um código de Huffman. O número de bits usados é contado e se exceder a quantidade disponível, o passo de quantização é aumentado. Um passo maior conduz a um menor número de níveis de quantização que são codificados com menos bits. Este ciclo é iterado até que o dispêndio de bits seja suficientemente baixo. O ciclo exterior tem o objectivo de "levar" o erro de quantização abaixo do limiar de mascaramento em cada banda crítica. O ruído realmente introduzido é obtido pela diferença no domínio da frequência entre o bloco original e o quantizado. Se o ruído medido numa banda superar o limiar calculado, o passo de quantização para os coeficientes espectrais dessa banda é reduzido e repete-se a quantização e codificação novamente.

Para reconstruir os valores espectrais, o receptor só precisa de receber os códigos de Huffman e os passos de quantização. Este facto apresenta duas grandes vantagens para a importante classe de aplicações de difusão (onde há um único codificador para muitos receptores). Primeiro, o receptor é extremamente simples: um descodificador, a desquantização e a transformada inversa, tudo em cascata. Segundo, há uma grande liberdade de implementação do codificador: os pormenores do ciclo de controlo de distorção, particularmente o cálculo do limiar de mascaramento, podem ser alterados sem que isso acarrete a modificação dos receptores.

Perceptual Transform Coding (PXFM)

A codificação PXFM [83] usa a transformada de Fourier com sobreposição de 1/16 tal como o MSC. A única diferença reside na dimensão dos blocos: 2048 amostras no PXFM contra 1024 no MSC. O limiar de mascaramento é calculado explicitamente a partir dos mesmos coeficientes espectrais que serão quantizados e transmitidos. O modelo psicoacústico usado é bastante completo e envolve vários passos:

1. Análise do sinal em 26 bandas críticas, a partir do espectro de potência.

2. Aplicação da função de espraiamento ao espectro em bandas críticas.

3. Cálculo do limiar de mascaramento tomando em consideração a assimetria entre mascaramento de tons e ruído. (A tonalidade é avaliada a partir da medida de planura espectral do bloco.)

4. Ajuste ao limiar absoluto de audição.

Os 1024 coeficientes complexos são em seguida repartidos em 128 conjuntos de oito para posterior quantização. Em cada conjunto é determinada a amplitude máxima o que, juntamente com o limiar de mascaramento apropriado, permite calcular o número de níveis de quantização a usar nesse conjunto. Os coeficientes são quantizados uniformemente e codificados ou com um código de Huffman ou com um técnica simples de empacotamento de bits baseada em aritmética de base variável. O número de bits necessários para codificar todo o bloco é então determinado, e caso seja diferente do pretendido procede-se a um ajuste do limiar de mascaramento e repete-se o processo de cálculo do número de níveis dos quantizadores e do consequente número de bits. Em cada iteração, o limiar é modificado por multiplicação com um estimador que depende do desvio de débito em relação ao desejado bem como do passado do sinal codificado. Quando o número de bits atinge os limites estabelecidos, os dados são finalmente transmitidos juntamente com a informação secundária composta pelos 128 picos espectrais e pelos valores finais do limiar de mascaramento. Tanto os picos como o limiar são quantizados logaritmicamente em 256 níveis. Esta informação é necessária para que o receptor possa determinar o número de níveis de cada quantizador e assim consiga "desempacotar" os coeficientes codificados.

O codificador foi testado a 4 bits por amostra com sinais amostrados a 32 kHz revelando qualidade perfeitamente "transparente". Testes posteriores sugeriam que 3 bits por amostra seriam suficientes para conseguir a mesma qualidade subjectiva.

Hybrid Coder

Em [22], Brandenburg e Johnston propuseram um codificador com uma resolução tempo-frequência melhorada e uma melhor adaptação às características de filtragem do ouvido. O objectivo era conseguir uma boa resolução espectral para uma estimação precisa do limiar de mascaramento e, simultaneamente, uma boa resolução temporal que evitasse o aparecimento de *pré-ecos*[7], tão característicos dos codificadores por transformada com muitas bandas. A solução que sugeriram consistia numa decomposição não uniforme do plano tempo-frequência usando uma estrutura híbrida de bancos de filtros QMF e transformadas MDCT. Uma árvore de 3 filtros QMF divide o sinal em 4 bandas com larguras entre 3 e 12 kHz (para uma frequência de amostragem de 48 kHz). A banda mais baixa (0–3 kHz) é subdividida por uma MDCT em 128 linhas. As restantes bandas, de 3, 6 e 12 kHz, são transformadas por MDCTs de 64 linhas. Disto resultam 320 linhas espectrais com resolução na frequência (tempo) que varia desde 23.4 Hz (21.3 ms) nas baixas frequências até 187.2 Hz (2.7 ms) nas altas. Esta filtragem tem decimação máxima e permite reconstrução quase perfeita devido ao uso de QMFs de ordem 80.

[7] Os pré-ecos são artefactos de codificação que podem surgir em codificadores a trabalhar no domínio da frequência, devido à distribuição do ruído de quantização por toda a duração dos blocos de processamento. Quando ocorre uma transição brusca perto do final de um bloco, o modelo psicoacústico pode sobreavaliar o limiar de mascaramento, e o codificador injecta ruído de quantização que se pode revelar audível no início do bloco [38].

O limiar de mascaramento é estimado com base na estrutura do sinal de saída do banco de filtros. O cálculo é feito em "fatias" de tempo alinhadas com as amostras das bandas de frequência mais alta. Como nas altas frequências há 8 amostras por cada amostra nas baixas frequências, a energia do sinal na banda de frequências mais baixas é distribuída igualmente por 8 fatias de tempo e nas bandas intermédias é distribuída de acordo com o ritmo de transformação respectivo. O modelo psicoacústico é semelhante ao usado no codificador PXFM mas inclui uma medida de tonalidade local relacionada com a previsibilidade de cada coeficiente espectral. Os valores do limiar para as linhas que ocupam mais que uma fatia de tempo são então somadas para produzirem uma estimativa com a resolução temporal adequada a cada banda. Para a quantização e codificação é usado o mesmo esquema de análise-por-síntese do OCF.

A ideia de basear a codificação numa decomposição não uniforme tem sido aplicada noutros codificadores. Em [144], Spille e Schröder estudam o projecto de um banco de filtros não uniforme para codificação de áudio de alta qualidade. Deste trabalho resulta uma estrutura em árvore de filtros CQF que decompõe o sinal em 15 bandas de larguras um pouco superiores às bandas críticas. Outro exemplo, apresentado em [159], é uma adaptação do PXFM que em vez da FFT utiliza uma transformada hierárquica composta por dois estágios de transformadas ELT. Esta transformação produz 181 linhas espectrais com um custo muito baixo: cerca de 20 operações por amostra. Uma técnica de codificação híbrida com aplicação comercial é o ATRAC usado no sistema MiniDisc da Sony [156]. Dois filtros QMF dividem o sinal em três bandas: 0–5.5 kHz, 5.5–11 kHz e 11–22 kHz. Cada uma destas bandas é subdividida por uma MDCT com um comprimento de bloco que varia dinamicamente. A janela longa (de 11.6 ms) é usada a maior parte do tempo, mas quando é detectado um "ataque" no sinal, o codificador comuta para a janela curta (1.45 ms na banda alta e 2.9 ms nas bandas baixas). O objectivo é evitar os pré-ecos que poderiam surgir pelo espalhamento do ruído de quantização ao longo da janela longa. A quantização é em BFP e o algoritmo de atribuição de bits não é especificado—poderá ser mais ou menos complexo conforme a aplicação. Para aplicações com pouca complexidade é sugerido um algoritmo semelhante ao do LC-ATC mas com a inclusão de uma medida de tonalidade para interpolar entre a atribuição de bits fixa e a variável.

Dolby AC-3

O algoritmo de codificação AC-3 dos laboratórios *Dolby* foi desenvolvido no início da década de 1990 com o objectivo de fornecer um suporte de áudio multi-canal de alta qualidade para a televisão de alta definição [153]. De facto, acabou por ser aceite para integrar a norma norte-americana de televisão de alta definição, mas a sua primeira aplicação foi na indústria cinematográfica, onde veio fornecer uma forma de armazenar 5.1 canais de som codificados opticamente nos espaços entre as perfurações das películas de cinema de 35mm, que até aí só possuíam 4 canais codificados analogicamente em duas pistas ópticas [33]. Este novo formato foi bem aceite, o que pode ter contribuído para facilitar a adopção do AC-3 nos novos discos de vídeo digitais (DVD) que surgiram no final da década. Este sistema de codificação também é conhecido pela designação *Dolby Digital*.

O codificador AC-3 é descrito com algum pormenor em [153] e em [45], onde também é comparado a um algoritmo anterior do mesmo laboratório. O AC-3 emprega um banco de filtros modulados com reconstrução perfeita do tipo MDCT com 256 bandas e janela de 512 pontos com sobreposição a 50%. A janela usada, conhecida como janela KBD[8] ou janela de

[8] *Kaiser-Bessel derived window*.

Fielder, foi projectada de forma garantir melhor rejeição em frequências afastadas, à custa de alguma perda de selectividade em relação à janela de seno usada noutros codificadores. Em condições de não estacionaridade, o banco de filtros pode ser comutado para um versão que aplica uma transformada mais curta a cada metade do bloco de 512 amostras.

Os coeficientes da transformada são codificados com uma variante da técnica de vírgula flutuante em blocos. Primeiro são transmitidos os expoentes e mais tarde as mantissas com precisão limitada. O conjunto dos expoentes de um bloco transformado constitui uma aproximação à envolvente espectral do sinal com uma resolução de cerca de 6 dB (uma variação de uma unidade no expoente corresponde a um factor de 2 na amplitude no coeficiente). O primeiro expoente, o expoente DC, é codificado em valor absoluto mas os restantes são codificados diferencialmente em passos de 0, ± 1 ou ± 2. Isto permite variações de até 12 dB entre coeficientes adjacentes, o que os autores consideraram suficiente para a maioria das situações tendo em conta os declives das bandas de transição dos filtros de análise. Este modo de codificação permite a transmissão da envolvente espectral com uma resolução muito fina, mas implica um custo de 7/3 de bit por amostra (7 bits por terno de coeficientes). Para reduzir este custo e porque tal resolução espectral só interessa em períodos de relativa estacionaridade, este modo tipicamente só é usado no primeiro de cada grupo de 6 blocos enquanto os restantes blocos reutilizam a mesma envolvente. Existem outros dois modos de codificação da envolvente que podem ser usados em períodos de menor estacionaridade e que poupam bits decimando por 2 ou por 4 o conjunto dos coeficientes espectrais a transmitir. As envolventes descritas nestes dois modos também podem ser reutilizadas em blocos sucessivos embora naturalmente isto ocorra com menor frequência. A escolha dos modos de codificação da envolvente é da responsabilidade exclusiva do codificador e é transmitida ao descodificador usando 2 bits por bloco. Qualquer alteração da estratégia de escolha não afecta, portanto, o algoritmo do(s) descodificador(es).

Depois de codificados os expoentes, determinam-se as mantissas que são então codificadas por quantizadores seleccionados de entre um conjunto variado que cobre uma gama vasta de resoluções e débitos, como se pode apreciar na Tabela 2.1. Os resultados de alguns dos quantizadores mais grosseiros são agrupados em ternos ou pares para uma codificação mais eficiente.

Tabela 2.1: Quantizadores de mantissas disponíveis no AC-3. Indica-se o número de níveis (N) e o débito correspondente (R) em bits por amostra.

N	1	3	5	7	11	15	32	64	128	256	512	1K	2K	4K	16K	64K
R	0	5/3	7/3	3	7/2	4	5	6	7	8	9	10	11	12	14	16

A selecção dos quantizadores é feita por um algoritmo de atribuição de bits baseado em critérios perceptuais [30]. O processo começa por ler os valores da envolvente espectral acabada de codificar e integra-os em grupos de 1, 3, 6, 12 ou 24 coeficientes de modo a formar um espectro com 50 bandas de larguras aproximadamente iguais a meio Bark (para uma frequência de amostragem de 48 kHz). Este novo espectro é convoluído com uma curva de espalhamento e o resultado, depois de limitado inferiormente pelo limiar absoluto de audição, constitui uma estimativa do limiar de mascaramento do sinal. A curva de espalhamento usada, quando traçada em termos de logaritmo da atenuação versus índice da banda, é composta por dois trechos lineares. Os quatro parâmetros que a definem foram ajustados a um conjunto

de dados experimentais de mascaramento, mas podem ser modificados dinamicamente. A simplicidade da curva de espalhamento e a representação logarítmica dos níveis espectrais permite uma implementação muito eficiente da operação de convolução sem necessidade de efectuar multiplicações. O limiar de mascaramento obtido é então subtraído da envolvente espectral original, obtendo-se a relação sinal-ruído desejada, que permite finalmente seleccionar o quantizador adequado para cada coeficiente espectral. Note-se que apesar do limiar de mascaramento ser calculado apenas com a resolução das bandas perceptuais, a atribuição de bits é feita com a resolução mais fina (igual à da envolvente espectral), permitindo ganhos de codificação superiores.

Um aspecto interessante do AC-3 é que esta atribuição de bits (e selecção de quantizadores) é regenerada no descodificador a partir da mesma envolvente espectral que já foi transmitida. Por este motivo, é considerado um algoritmo retroadaptativo[9] cuja principal vantagem é evitar a ocupação de informação secundária para transmitir a atribuição calculada. Apesar desta opção, o codificador AC-3 mantém alguma versatilidade pois pode controlar vários aspectos que afectam este processo. Em particular, pode alterar dinamicamente a resolução espectral e temporal da envolvente transmitida, e pode modificar os parâmetros que definem a curva de espalhamento. Caso isto não seja considerado suficiente, o codificador pode ainda usar um modelo psicoacústico alternativo e transmitir ao descodificador quaisquer diferenças entre os limiares calculados. Este mecanismo, no entanto, rouba bits à codificação dos coeficientes e deve ser usado com parcimónia.

MPEG-*Audio*

Os organismos internacionais de normalização ISO[10] e IEC[11] criaram em 1988 um grupo de especialistas denominado MPEG[12] com o objectivo de definir uma norma internacional para a codificação eficiente de sinais vídeo e áudio associado para armazenamento em suportes digitais com um débito até 1.5 Mbit/s (tipo CD). Esta primeira fase foi concluída em 1992 com a aprovação da norma MPEG-1 [77]. Entretanto começara já a segunda fase de normalização que generaliza os objectivos iniciais de forma a contemplar um maior leque de aplicações audiovisuais com qualidade superior e usando débitos mais elevados. Foi concluída em Abril de 1994 com a aprovação do documento [78] conhecido vulgarmente pelo nome de MPEG-2.

No que diz respeito ao áudio, o MPEG-1 define três "camadas" (ou *Layers*) de codificação com qualidade e complexidade crescentes. Os *Layers* I e II são muito semelhantes ao algoritmo MUSICAM, uma das propostas avaliadas pelo grupo. O *Layer* III integra diversos melhoramentos contribuídos pelo outro algoritmo apurado: o ASPEC.

A primeira fase da norma suporta frequências de amostragem de 32, 44.1 e 48 kHz, e os modos: *mono*, *dual channel* (para programas bilingues), *stereo* e opcionalmente, *joint* ou *intensity stereo*. São previstos vários débitos entre 32 e 192 kbit/s para canais mono e entre 64 e 384 kbit/s para canais estéreo. A fase 2 do MPEG prevê três novas frequências de amostragem (16, 22.05 e 24 kHz) e várias configurações multicanal para som circundante ou canais tipo comentário. As novas frequências exigiram alterações mínimas ao processo de codificação e a extensão multicanal mantém compatibilidade com a versão estereofónica básica.

[9] Não se trata, no entanto, de uma retroadaptação em sentido estrito, como se mostrará no Cap. 7.
[10] *International Standards Organization.*
[11] *International Electro-technical Commission.*
[12] *Moving Pictures Expert Group.*

É importante notar que a norma só define o descodificador e o significado do *bit stream* codificado. O codificador é descrito apenas com valor informativo, o que deixa em aberto a possibilidade de ser adaptado a aplicações específicas e de ser melhorado progressivamente de acordo com a disponibilidade tecnológica.

Todas as camadas decompõem o sinal em 32 bandas usando um banco de filtros polifásicos (pseudo-QMF) de ordem 511 que foi "herdado" do MUSICAM. Com uma atenuação de lóbulos laterais superior a 96 dB, estes filtros aproximam as condições de cancelamento de *aliasing* e reconstrução perfeita. Para conseguir uma maior resolução espectral, no *Layer* III é feita uma subdivisão adicional de cada banda em 18 ou 6 sub-bandas, usando uma MDCT com janela dinâmica (e janelas de transição). É necessário um andar adicional para redução do *aliasing* que resulta da combinação em cascata dos filtros com as transformadas. A comutação para a janela curta ocorre nas bandas acima de uma determinada frequência quando é detectado um "ataque" no sinal, assinalado por um aumento brusco da solicitação de bits.

A parte informativa da norma descreve dois modelos psicoacústicos que podem ser usados em qualquer das três camadas de codificação. Na prática o modelo 1, mais simples, é usado nos *Layers* I e II enquanto o modelo 2, mais preciso, é destinado ao *Layer* III. Ambos os modelos fazem uma análise espectral do sinal em paralelo com o banco de filtros do codificador usando uma transformada discreta de Fourier de 512 ou 1024 pontos e produzem o mesmo tipo de informação—uma estimativa da relação sinal-ruído mascarável[13] em cada banda—que é depois usada pelos algoritmos de atribuição de bits ou de ruído.

No *Layer* I a quantização é uniforme e blocos de 12 amostras em cada banda são codificadas em BFP. O factor de escala de cada bloco (máxima amplitude do bloco) é quantizado logaritmicamente e codificado em 6 bits. As amostras do bloco são quantizadas com um número de níveis determinado por um algoritmo de atribuição de bits que, em cada iteração, aumenta a relação sinal-ruído do quantizador que mais contribua para melhorar a qualidade. A informação da atribuição de bits é transmitida ao receptor num código de 4 bits por cada bloco. No *Layer* II, os factores de escala de três blocos adjacentes (contendo um total de 36 amostras) são comparados e consoante os seus valores relativos, dos três poderão ser transmitidos apenas um ou dois. Dois bits adicionais indicam quantos e quais os factores de escala transmitidos. Nas bandas mais baixas é possível usar uma quantização de 16 bits—mais fina que as disponíveis no *Layer* I. Em contrapartida, o número de quantizadores disponíveis diminui para as bandas de índice crescente. Em média, estas estratégias permitem reduzir a fracção do *bit stream* dedicada à informação secundária mas aumentam a complexidade e o atraso global do sistema. A quantização no *Layer* III segue um algoritmo semelhante ao OCF: um sistema de análise por síntese com dois ciclos aninhados. Os quantizadores são não uniformes e os índices de quantização são codificados com códigos de Huffman. Sequências de zeros são codificadas pelos seus comprimentos (*run-length coding*).

MPEG-2 *Advanced Audio Coding* (AAC)

Durante a segunda fase das actividades de normalização do MPEG, para além das extensões multicanal e para baixos débitos, foi ainda desenvolvido um novo codificador de alta qualidade não condicionado por restrições de compatibilidade com os anteriores codificadores do MPEG. Este sistema, chamado *Advanced Audio Coding* ou AAC, combina uma série de tecnologias de eficácia comprovada, bem como algumas inovações interessantes [16, 21].

[13] SMR, *Signal-to-Mask Ratio*.

Para a decomposição espectral, o AAC usa uma MDCT com dois tipos de comutação dinâmica de janelas: quer entre janelas de dimensão diferente (*transform block switching*), comutando entre janelas de 2048 ou de 256 amostras; quer entre janelas de forma diferente (*window shape switching*), comutando entre uma janela de seno ou uma janela KBD. Numa versão simplificada introduzida no MPEG-4 para conseguir atrasos baixos, mantém-se a possibilidade de comutação da forma da janela, mas fixa-se o comprimento em 1024 amostras. A quantização é semelhante à do Layer-III, usando BFP em 49 grupos de coeficientes adjacentes com larguras aproximadamente proporcionais às bandas críticas. Os coeficientes são quantizados por quantizadores não uniformes de escala variável e codificados com códigos de Huffman, estando disponíveis várias tabelas de codificação alternativas. Os factores de escala são transmitidos com uma resolução de 1.5 dB e codificados diferencialmente com uma tabela de Huffman específica.

Um aspecto menos comum é a utilização de predição retroadaptativa dos sinais das subbandas. O objectivo é melhorar o desempenho em sinais estacionários, por isso a predição só é usada nas janelas longas e só até à banda dos 16kHz. Os preditores são de segunda ordem, o que deve permitir prever tons puros com precisão. Outra inovação é a *modelação temporal de ruído* (*temporal noise shaping*), que consiste em fazer uma codificação D*PCM dos coeficientes espectrais à saída da transformada, i.e. uma codificação diferencial em malha aberta no domínio da frequência. É sabido (ver Sec. 2.3.2) que a codificação D*PCM no domínio do tempo resulta num ruído com envolvente espectral idêntica à do sinal codificado. Por dualidade, compreende-se que quando aplicada no domínio da frequência resulte num ruído com envolvente temporal idêntica à do sinal. Esta modelação temporal do ruído permite minorar os artefactos do tipo "pré-eco" que são particularmente notórios em codificadores com transformadas de dimensão alta, como é o caso.

2.3 Codificação Preditiva

Os sinais áudio, como muitos outros tipos de sinais ou de dados, são bastante previsíveis, isto é: existem modelos que permitem descrever a evolução destes sinais ao longo do tempo. Munidos de um desses modelos e de alguma informação sobre o passado de um sinal, é possível estimar com relativa precisão o valor de uma nova amostra. Esta capacidade de prever permite reduzir a inovação introduzida por cada amostra e, consequentemente, a informação necessária para a codificar. Este é o princípio em que se baseia a *codificação preditiva*. Ao contrário dos codificadores descritos na Secção 2.2, um codificador preditivo processa os sinais no domínio do tempo e pode fazê-lo de forma contínua, amostra-a-amostra. Isto é vantajoso porque elimina os efeitos da divisão em blocos associados às transformadas e bancos de filtros decimados e, além disso, possibilita atrasos de codificação mínimos.

2.3.1 Codificação Diferencial

Embora existam formas de codificação preditiva mais poderosas e de aplicação mais generalizada, como as usadas em compressão de dados [12, Chapter 1], para a codificação de sinais aplica-se usualmente uma estrutura particular denominada de *codificação diferencial* ou DPCM[14] [80, Chapter 6]. Neste esquema, um preditor baseia-se no passado do sinal para estimar cada amostra nova e o que se codifica é o sinal diferença ou erro de predição relativo

[14] *Differential Pulse Code Modulation.*

ao sinal original. O descodificador gera localmente a sua própria previsão e corrige-a com o sinal de erro recebido para produzir uma réplica do original.

O preditor é tipicamente um filtro linear adaptativo, o que justifica outra designação usual desta técnica como *codificação preditiva linear*. Também é frequente que o quantizador do sinal de erro seja adaptativo, pelo menos em termos de escala, para abarcar mais facilmente a gama dinâmica dos sinais.

Pode verificar-se que neste tipo de codificador, o erro de reconstrução é igual ao erro de quantização [80, Sec. 7.1] e, portanto, apresenta um espectro aproximadamente plano. Mesmo que o quantizador seja relativamente grosseiro, é de esperar que este erro se mantenha aproximadamente branco porque a entrada do quantizador é o erro de predição e este já é bastante auto-descorrelacionado (se o preditor for eficiente).

2.3.2 Codificação com Realimentação de Ruído

O ruído branco não é conveniente em termos perceptuais. Por isso, a codificação preditiva simples não é adequada para sistemas de codificação perceptual. Uma generalização que permite a modelação dinâmica do espectro do ruído de reconstrução—fundamental para o aproveitamento das limitações da percepção auditiva—é a chamada *codificação preditiva generalizada* ou *codificação com realimentação de ruído*[15] [4][80, Chapter 7], esquematizada na Figura 2.2. Neste sistema, o filtro preditor $P(z)$ permite extrair alguma redundância do sinal enquanto o filtro de realimentação de ruído $F(z)$ procura reduzir a relevância da distorção introduzida. De um modo geral, tanto os filtros como o quantizador são adaptativos. A densidade espectral de potência do erro de reconstrução deste codificador é dada por [4]

$$S_{rr}(z) = \frac{|1 - F(z)|^2}{|1 - P(z)|^2} S_{qq}(z)$$

onde $S_{qq}(z)$ é a densidade espectral de potência do erro de quantização (aproximadamente constante). É portanto possível moldar a forma do ruído através do filtro $F(z)$ sem comprometer o desempenho do preditor $P(z)$. Dois casos particulares desta estrutura são: o codificador diferencial já referido que corresponde à situação $F(z) = P(z)$; e o denominado DPCM em malha aberta ou D*PCM que se obtém fazendo $F(z) = 0$ e que dá ao ruído um perfil espectral semelhante ao do sinal mas alguns dBs abaixo [80, Fig. 7.2].

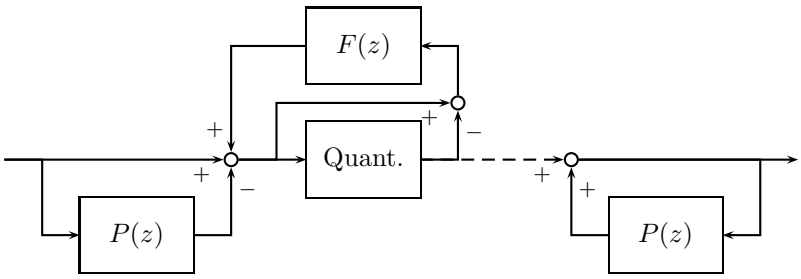


Figura 2.2: Codificação com realimentação de ruído (NFC). Equivale a DPCM quando $F(z) = P(z)$, e a D*PCM quando $F(z) = 0$.

[15] *Noise Feedback Coding* (NFC).

É interessante notar que a introdução de realimentação de ruído não afecta a estrutura do descodificador e portanto, em aplicações que o permitam, pode implementar-se um algoritmo bastante complexo no codificador sem modificar os receptores.

Em [136], Schroeder, Atal e Hall desenvolveram uma medida objectiva de degradação de sinais de voz e propuseram um codificador com realimentação de ruído projectado de forma a minimizar essa medida. A medida baseia-se num modelo psicoacústico bastante completo que avalia, através de operações não-lineares, a intensidade percebida de ruído na presença de um sinal mascarante. O modelo contempla fenómenos de espalhamento da excitação ao longo da membrana basilar, mascaramento parcial, limiar de mascaramento e limiar absoluto de audição.

Outra aplicação deste método é o sistema *Super Bit Mapping* [2] que usa princípios psicoacústicos para gravar discos compactos comuns com qualidade melhorada. O diagrama de blocos é o da Figura 2.2 mas com a predição eliminada, isto é $P(z) = 0$. O sinal original em formato PCM de 20 bits é requantizado para 16 bits no codificador e o erro (os 4 bits menos significativos) é realimentado através do filtro $F(z)$. O receptor fica reduzido a um simples descodificador de PCM ou seja: o vulgar conversor digital-analógico existente em qualquer leitor de discos compactos. São propostas duas versões do sistema com complexidades diferentes. A primeira implementa a realimentação de ruído com um filtro FIR de ordem 12, invariante, projectado para garantir ruído abaixo do limiar absoluto de audição. A segunda versão usa um filtro adaptativo que além do limiar absoluto também considera o fenómeno de mascaramento simultâneo. Este sistema demonstra uma possibilidade interessante: o uso de realimentação de ruído para melhorar a qualidade perceptual de um sistema de codificação pré-existente, sem necessidade de substituir os receptores já instalados.

2.3.3 Predição Linear em Domínio de Frequência Deformado

Os preditores usados em codificação preditiva são geralmente filtros lineares FIR de ordem relativamente baixa. Verifica-se que o espectro do ruído de reconstrução resultante da aplicação destes filtros apresenta uma resolução demasiado grosseira nas frequências baixas e desnecessariamente fina nas frequências altas. Uma forma de ultrapassar este problema é utilizar técnicas de deformação de frequência (*frequency warping*) no projecto e implementação destes filtros. Estas técnicas consistem basicamente na substituição de cada unidade de atraso z^{-1} de um dado filtro $H(z)$ por um circuito passa-tudo de primeira ordem, $D(z) = \frac{z^{-1}-\lambda}{1-\lambda z^{-1}}$. A resposta em frequência do filtro resultante $H(D^{-1}(z))$ é igual à resposta em frequência do filtro original, mas num domínio de frequências diferente, relacionado com o domínio original por uma transformação bilinear [70]. Assim, é possível projectar um filtro usando especificações pré-deformadas e, posteriormente, implementá-lo com as *unidades de atraso generalizadas* $D(z)$ de forma a conseguir o filtro desejado. A vantagem é a obtenção facilitada de respostas com resolução espectral não uniforme usando filtros de ordem moderada. Algumas aplicações desta técnica à estimação espectral têm já mais de 20 anos [11, 51], mas mais recentemente tem havido um ressurgimento do interesse nesta técnica para utilização na codificação preditiva e outras formas de processamento de sinais áudio [89, 71, 70]. Uma motivação forte para a adopção destas técnicas é que a escolha adequada do parâmetro λ produz uma deformação de frequências surpreendentemente próxima da realizada pelo sistema auditivo, traduzida pelas escalas Bark ou ERB (ver Sec. 6.3.2) [141].

2.4 Codificação Paramétrica

Além dos codificadores no domínio da frequência e dos codificadores preditivos, existe um terceiro grupo de sistemas de codificação, destinados geralmente a aplicações de débito muito baixo, que se podem classificar como codificadores paramétricos. Trata-se de sistemas que não almejam uma reprodução fiel das formas de onda, mas apenas de algumas das suas características ou *parâmetros* mais importantes, como: frequência, amplitude, duração, conteúdo harmónico, envolvente espectral ou envolvente temporal. Um destes sistemas [35] baseia-se numa decomposição do sinal em componentes sinusoidais. Evoluções posteriores acrescentaram modelos de sinais harmónicos e de ruído [123], e foram incluídas no MPEG-4. Um aspecto interessante das representações paramétricas produzidas por estes codificadores é a possibilidade de variar a tonalidade e/ou a velocidade de reprodução por simples manipulação dos parâmetros no descodificador.

Outra tendência interessante é a incorporação de algumas técnicas de codificação paramétrica em codificadores de transformada. Uma possibilidade sugerida em [137], por exemplo, é o uso de geradores de ruído no descodificador para preencher bandas do sinal que tenham sido classificadas como ruído e codificadas com um único parâmetro, a sua potência. Para componentes sinusoidais, foi proposto um método de análise e síntese que funciona directamente no domínio dos coeficientes transformados por uma MDCT, facilitando a sua integração em codificadores de transformada [43, 44]. A técnica de *spectral band replication* (SBR) parece basear-se numa codificação de envolvente espectral, combinada com uma extensão harmónica dos sinais no descodificador para substituir as componentes de alta frequência eliminadas no codificador [32].

Todas estas técnicas revelam um esforço renovado ao nível da modelação da fonte, e não tanto ao nível da modelação do receptor.

Capítulo 3

Transformadas e Bancos de Filtros

Neste capítulo apresentamos um estudo resumido das transformadas e bancos de filtros multifrequência mais relevantes para aplicação à compressão de sinais áudio. Começamos por definir, na Secção 3.1, os conceitos fundamentais relacionados com bancos de filtros multifrequência e sinais multi-resolução que facilitarão a leitura do resto do capítulo. Na Secção 3.2 apresenta-se uma lista das características que o projectista deve procurar num banco de filtros para codificação perceptual. Na Secção 3.3 classificam-se as principais estruturas de decomposição uniforme usadas em compressão, e na Secção 3.4 mostram-se formas de combinar esses bancos uniformes em estruturas compostas para fazer análise não uniforme.

3.1 Definições

Historicamente, os conceitos de banco de filtros decimados e de transformada aplicada em blocos surgiram e foram desenvolvidos separadamente. Porém, justifica-se cada vez mais uma progressiva integração destes conceitos em face do desenvolvimento da teoria dos sistemas multifrequência [29, 157] por um lado, e das transformadas com sobreposição[1] [24, 104, 103] por outro. É fácil verificar que qualquer transformada aplicada bloco-a-bloco pode ser interpretada como um banco de filtros decimados. Por outro lado, uma classe importante de bancos de filtros com propriedades interessantes pode ser considerada como uma forma estendida de transformada—a transformada com sobreposição.

Por esta razão não fazemos uma distinção clara destas duas formas de decomposição de sinais, preferindo tratá-las de forma integrada. Infelizmente, em virtude das suas origens distintas, há alguma duplicação de terminologia para noções equivalentes ou equiparáveis, como se resume abaixo:

Codificação em sub-bandas	Codificação por transformada
Banco de filtros de análise	Transformada directa
Banco de filtros de síntese	Transformada inversa
Resposta impulsional	Vector ou função de base
Banda ou sub-banda	Linha ou coeficiente espectral
Reconstrução perfeita	Invertibilidade

Assim, ao longo deste documento usamos termos de ambas as colunas, de acordo com o uso mais comum na literatura relevante.

[1] *Lapped Transforms.*

3.1.1 Bancos de Filtros Multifrequência

A Figura 3.1 mostra o diagrama de blocos de um sistema de processamento em sub-bandas com factores de decimação inteiros, formado por um banco de filtros de *análise* ou *decomposição*, um bloco de processamento, e um banco de filtros de *síntese* ou *reconstrução*. Trata-se de um *sistema multifrequência* porque inclui blocos *decimadores* e *interpoladores* [29, 157].

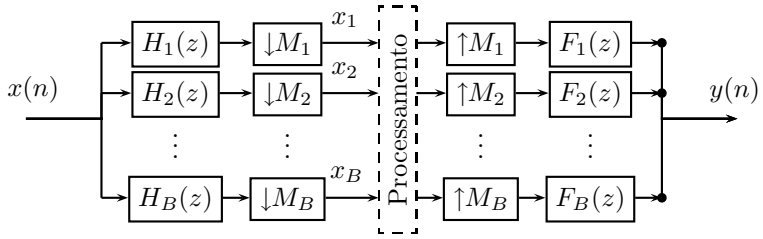


Figura 3.1: Sistema de processamento em sub-bandas com factores de decimação inteiros. Os blocos marcados com setas para baixo (↓) e para cima (↑) representam as operações de subamostragem (decimação) e sobreamostragem (interpolação), respectivamente.

O banco de filtros de *análise* ou *decomposição* separa o sinal $x(n)$ em B *canais* $x_b(n_b)$, por intermédio dos filtros de análise $H_b(z)$ e dos decimadores $\downarrow M_b$. Normalmente, os filtros de análise são passa-banda e em conjunto cobrem toda a gama de frequências de zero a π. Os *decimadores* deixam passar uma amostra em cada M_b, reduzindo a frequência de amostragem em cada banda para f_S/M_b, onde f_S é a frequência de amostragem do sinal $x(n)$. A decimação dilata o espectro em torno do círculo unitário, sobrepondo segmentos adjacentes de largura π/M_b. Disto resulta uma interferência da banda atenuada sobre a banda passante a que se chama *aliasing*.

O bloco de processamento pode assumir diversas formas consoante a aplicação. Em sistemas de codificação, o bloco de processamento consiste na quantização, transmissão e desquantização das amostras de cada sub-banda.

O banco de filtros de *síntese* ou *reconstrução* recombina as bandas processadas num único sinal $y(n)$. Os *interpoladores* intercalam sequências de $M_b - 1$ zeros entre amostras, de forma a recuperar o ritmo de amostragem original em cada banda. Esta operação provoca uma compressão do espectro, arrastando um conjunto de réplicas ou imagens do espectro inicial para dentro do intervalo $[-\pi, \pi]$ (*imaging*). Em geral, os filtros de síntese $F_b(z)$ têm uma característica passa-banda idêntica à dos filtros de análise correspondentes, com o propósito de eliminar as réplicas em excesso. A adição das saídas destes B filtros produz finalmente o sinal $y(n)$.

Se os factores de decimação e interpolação forem tais que

$$\sum_{b=1}^{B} \frac{1}{M_b} = 1, \tag{3.1}$$

então os números de amostras por unidade de tempo à saída e à entrada do banco de filtros são iguais, e diz-se que o sistema é *maximamente decimado* ou *criticamente subamostrado*.

Quando o somatório do primeiro membro é superior à unidade, diz-se que a decimação é *subcrítica*. Um caso extremo de decimação subcrítica ocorre quando $M_b = 1, \forall_b$: o sistema degenera num par de bancos de filtros com uma só frequência de amostragem e o número de amostras a processar é B vezes superior ao original. Em sistemas de codificação é conveniente que o banco de filtros tenha decimação máxima (ver Secção 3.2), por isso presumiremos essa condição daqui em diante.

3.1.2 Relação Entrada-Saída

Um sistema de processamento em sub-bandas, como outros sistemas multifrequência, é um sistema *linear e periodicamente variante no tempo* (LPTV).[2] Isto implica que o sistema não pode ser caracterizado no domínio do tempo por uma resposta impulsional única: são necessárias múltiplas respostas. Equivalentemente, no domínio da frequência, a resposta do sistema $Y(z)$ não se obtém simplesmente por um produto de uma função de transferência $T(z)$ pela entrada $X(z)$: ela depende também de versões moduladas—deslocadas na frequência—do sinal de entrada (ver Equação 3.2, por exemplo). As componentes de $Y(z)$ que dependem de versões deslocadas de $X(z)$ provêm do fenómeno de *aliasing* e por isso constituem a chamada *distorção de aliasing*. Se os filtros forem projectados convenientemente, o *aliasing* provocado pela decimação no banco de análise pode ser completamente cancelado no banco de síntese (na ausência de processamento das sub-bandas). Nesse caso, diz-se que o sistema tem *cancelamento de aliasing*, o que implica que se torne *linear e invariante no tempo* (LTI) com resposta $Y(z) = T(z)X(z)$. Se um sistema tem cancelamento de *aliasing* e $T(z) = z^{-d}$, então esse sistema tem *reconstrução perfeita* (PR) porque a saída é igual à entrada, a menos de um atraso de d amostras. Caso contrário, se $|T(e^{j\omega})| \neq 1$ ou se a fase de $T(e^{j\omega})$ não for linear, então o sistema tem *distorção de amplitude* ou *distorção de fase*, respectivamente.

3.1.3 Sinais Multi-Resolução

Considere-se como exemplo um sistema de 4 bandas com $M_1 = M_2 = 8$, $M_3 = 4$ e $M_4 = 2$. No instante $n = 0$, o banco de análise recebe a amostra $x(0)$ e debita uma amostra em cada banda: $x_1(0)$, $x_2(0)$, $x_3(0)$ e $x_4(0)$. No instante seguinte, os filtros recebem $x(1)$ mas não produzem qualquer saída. Em $n = 2$, só a banda #4 produz uma segunda amostra: $x_4(1)$. Continuando, verifica-se que a distribuição das amostras das várias bandas ao longo do tempo forma um padrão que se repete de 8 em 8 amostras:

$x(n)$	$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	$x(7)$	$x(8)$	$\cdots$
$x_1(n_1)$	$x_1(0)$								$x_1(1)$	$\cdots$
$x_2(n_2)$	$x_2(0)$								$x_2(1)$	$\cdots$
$x_3(n_3)$	$x_3(0)$				$x_3(1)$				$x_3(2)$	$\cdots$
$x_4(n_4)$	$x_4(0)$		$x_4(1)$		$x_4(2)$		$x_4(3)$		$x_4(4)$	$\cdots$

Um período

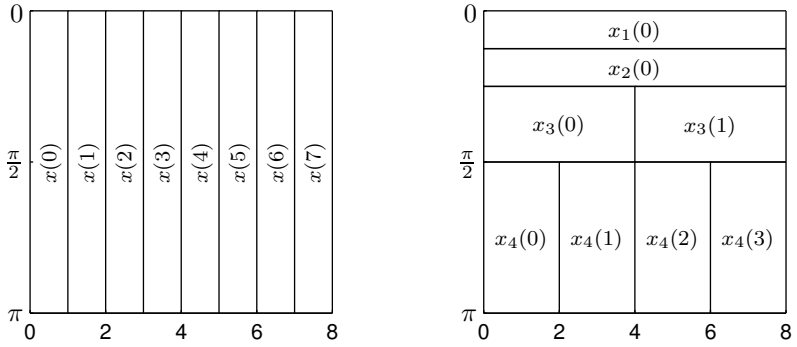
(Num caso geral, o período de repetição do padrão ou dimensão dos blocos S é dado pelo mínimo múltiplo comum dos factores de decimação M_b.) Assim, por cada bloco de 8 amostras da sequência de entrada, o banco de análise debita um bloco também com 8 amostras (porque

[2] Considerando que não há processamento e que os filtros de análise e síntese são lineares e invariantes no tempo (LTI).

a decimação é crítica) mas distribuídas de forma não uniforme pelas várias bandas. Se se agrupar as amostras de cada bloco, obtém-se um sinal vectorial de dimensão S,

$$\begin{aligned}\mathbf{x}(i) \;=\; [\; & x_1(S_1 i+0) \quad x_2(S_2 i+0) \quad x_3(S_3 i+0) \quad x_3(S_3 i+1) \\ & x_4(S_4 i+0) \quad x_4(S_4 i+1) \quad x_4(S_4 i+2) \quad x_4(S_4 i+3)\;]^T\end{aligned}$$

onde $S_b = S/M_b$ é o número de amostras da banda $\#b$ debitadas por cada bloco de entrada e i é o número do bloco. A um sinal vectorial como $\mathbf{x}(i)$, chama-se *sinal multi-resolução* porque cada vector (bloco) traduz uma divisão do plano tempo-frequência em *células* com diversas resoluções nos dois eixos. O modo específico como essa divisão está feita é determinado pelo conjunto de valores S_b e chama-se *forma* ou *formato* do sinal multi-resolução. Um sinal multi-resolução pode ser representado graficamente por um *mosaico* como os seguintes.[3]



O mosaico da direita representa um bloco do sinal $\mathbf{x}$ do exemplo acima, enquanto o mosaico da esquerda mostra um bloco do sinal de entrada, que pode ser considerado um caso particular de sinal multifrequência cujas amostras têm uma localização bem definida no tempo mas indeterminável na frequência. Em qualquer mosaico, o "comprimento" (resolução temporal) e a "largura" (resolução espectral) variam de célula para célula mas a área mantém-se constante.[4]

3.2 Características Desejáveis

Johnston e Brandenburg [84] identificaram um conjunto de objectivos, nalguns casos conflituosos, que devem guiar o projecto do banco de filtros para um codificador perceptual de áudio:

- O banco de filtros deve ser subamostrado criticamente ou quase criticamente, para minimizar o número de amostras a quantizar. Como vimos no Capítulo 2, alguns dos primeiros codificadores perceptuais de áudio usavam uma transformada de Fourier com sobreposição de 1/16 entre blocos adjacentes, o que resultava numa expansão do número de amostras por um factor de 16/15. O objectivo era a redução dos artefactos que surgem nas transições entre janelas de análise—o chamado *blocking effect*. Actualmente

[3] Nestes mosaicos, a unidade usada para graduar o eixo horizontal (do tempo) é o período de amostragem do sinal de entrada. Note-se ainda que a frequência angular é representada no sentido descendente.

[4] Para uma definição formal da localização e dimensões de uma célula no tempo e na frequência, veja-se [15, Chapter 2].

não há grande vantagem em usar decimação subcrítica porque o uso de bancos de filtros multifrequência ou transformadas com sobreposição permite uma redução muito mais eficaz destes efeitos sem qualquer aumento do número de amostras e com pouca complexidade adicional.

- O banco de filtros deve ter a propriedade de reconstrução perfeita. Se assim for, o erro de codificação introduzido fica a dever-se exclusivamente à fase de quantização, e esta operação pode ser controlada facilmente de forma a minimizar a perceptibilidade do ruído.

- Os filtros devem ter largura de banda igual ou inferior a uma banda crítica (1 Bark). Esta condição é importante porque permite um controlo mais preciso e eficiente do perfil espectral do ruído de quantização introduzido. Também é desejável para o cálculo do limiar de mascaramento que as bandas sejam mais estreitas que a função de espraiamento [145], o que é uma condição aproximadamente equivalente. Além disso, quanto melhor for a resolução espectral, maior será o ganho de compressão possível com um codificador por transformada [80].

- Os filtros de análise e de síntese devem ter uma boa selectividade na frequência. Este requisito não é fundamental para se obter, na ausência de processamento, cancelamento de aliasing ou mesmo reconstrução perfeita. No entanto, quando há processamento das sub-bandas, a distorção introduzida independentemente em cada banda espalha-se pelo espectro do sinal reconstruído segundo a função de transferência do filtro de síntese respectivo e interfere com as outras bandas. Uma boa selectividade minimiza o extravasamento do ruído e facilita a aplicação do limiar de mascaramento.

- A resolução temporal dos filtros deve ser suficientemente boa para evitar que o espalhamento do ruído no tempo viole as condições de mascaramento temporal, particularmente o pré-mascaramento (ver Capítulo 6), o que daria origem a pré-ecos audíveis. Devido à grande variação de largura das bandas críticas (entre 100 Hz e 4 kHz, aproximadamente), é difícil satisfazer simultaneamente os requisitos de resolução temporal e espectral com um banco de filtros uniforme.

- O sistema completo de análise e síntese deve ter pouco atraso global, especialmente em aplicações de transmissão bidireccional, e baixa complexidade computacional.

Alguns codificadores usam um segundo banco de filtros para fazer a análise espectral usada no modelo psicoacústico porque aí os requisitos diferem um pouco dos do banco de filtros principal. Em particular, não há qualquer necessidade de o método de análise permitir reconstrução perfeita uma vez que não é necessário voltar a fazer a síntese do sinal. Também não é fundamental que a subamostragem seja crítica porque a informação espectral é para uso local do codificador e não para transmissão directa ao receptor.[5] Em contrapartida, os requisitos de resolução no tempo e na frequência da análise espectral são semelhantes aos do banco principal. Além disso, o "preço" que se paga por ter reconstrução perfeita e subamostragem crítica é baixo, como se verá adiante. Por estas razões e por uma questão de economia de recursos computacionais faz sentido usar apenas um banco de filtros para as duas funções.

[5]Na verdade, a informação espectral acaba por ser transmitida ao receptor mas muito condensada, por exemplo na forma de um perfil espectral aproximado (LC-ATC) ou de factores de escala para blocos de várias amostras (BFP).

3.3 Bancos de Filtros Uniformes

Um banco de filtros uniforme decompõe o sinal de entrada num conjunto de bandas igualmente espaçadas. Obtém-se um banco destes a partir da Figura 3.1 usando filtros com igual largura de banda e o mesmo factor de decimação em todos os canais ($M_b = M, \forall_b$). Para ter subamostragem crítica (Equação 3.1), o factor de decimação tem que ser igual ao número de bandas ($M = B$), e nesse caso os filtros devem ter largura de banda igual a π/M para cobrir todo o espectro.

A forma directa de implementação de bancos de filtros uniformes tem um custo computacional de N multiplicações e $N-1$ adições por amostra de entrada, se considerarmos filtros FIR com respostas impulsionais de comprimento N. Como o comprimento dos filtros deve ser proporcional ao número de bandas para que a largura de cada banda seja π/M, vemos que a complexidade desta implementação cresce linearmente com M. No entanto, transformadas como a DFT ou a DCT têm implementações cuja complexidade é apenas proporcional a $\log M$. Isto revela que é possível fazer decomposição uniforme com implementações muito mais eficientes que a forma directa. Outra dificuldade com esta forma de implementação é a necessidade de projectar M pares de filtros com um elevado número de graus de liberdade apesar de terem todos especificações muito semelhantes. Por estas razões foram desenvolvidas outras formas de decomposição de sinais em bandas uniformes, que apresentam menor custo computacional e maior facilidade de projecto, como veremos a seguir.

3.3.1 *Quadrature Mirror Filters* (QMF)

Os primeiros estudos sobre bancos de filtros decimados com cancelamento de *aliasing* incidiram sobre o caso mais simples de bancos de dois canais. Quando $M = 2$, a expressão do sinal de saída do banco de síntese em função do sinal de entrada do banco de análise é dada por [29, 157]

$$Y(z) = T(z)X(z) + A(z)X(-z) \tag{3.2}$$

onde

$$T(z) = \frac{1}{2}[H_1(z)F_1(z) + H_2(z)F_2(z)]$$

é a chamada *função de distorção* e

$$A(z) = \frac{1}{2}[H_1(-z)F_1(z) + H_2(-z)F_2(z)],$$

o *ganho de aliasing*.

Há cancelamento de *aliasing* quando $A(z) = 0$ e nessa situação o sistema completo torna-se invariante no tempo com função de transferência $T(z)$. Um conjunto de condições suficientes que garantem essa propriedade é:

$$H_2(z) = H_1(-z),\ F_1(z) = H_1(z),\ F_2(z) = -H_1(-z). \tag{3.3}$$

Nestas condições, os filtros H_1 e H_2 têm respostas simétricas em torno da frequência $\omega = \pi/2$ e por isso se chama *quadrature mirror filters* (QMF) aos bancos de filtros que as satisfazem.[6]

[6] Alguns autores, em particular Vaidyanathan, usam o termo QMF para designar uma classe muito mais vasta de bancos de filtros.

Os bancos QMF permitem uma implementação polifásica eficiente com cerca de metade das operações necessárias numa implementação na forma directa. Por outro lado, estes sistemas não permitem reconstrução perfeita excepto para casos sem interesse prático [142]. Pode-se, no entanto, optimizar os filtros para aproximar essa condição. Nesse caso, o método de projecto tem que tentar aproximar a resposta do filtro ideal, bem como minimizar o erro de reconstrução. Um método, usado em [82], define uma função de erro global que inclui o erro de reconstrução bem como uma medida do desvio em relação ao filtro óptimo. Esta função de erro é minimizada usando um algoritmo de optimização não-linear. Jain e Crochiere propuseram um outro método, numericamente mais estável, baseado numa descrição do erro de reconstrução no domínio do tempo [79].

3.3.2 *Conjugate Quadrature Filters* (CQF)

Um outro conjunto de condições que garante o cancelamento de *aliasing* num sistema de duas bandas foi dado em [142]:

$$H_2(z) = -H_1(-z^{-1})z^{-(N-1)},\ F_1(z) = H_2(-z),\ F_2(z) = -H_1(-z) \tag{3.4}$$

onde N é o comprimento dos filtros (FIR).

Este tipo de banco de filtros é interessante porque, ao contrário dos QMF, permite reconstrução perfeita sem comprometer grandemente a qualidade de filtragem no banco de análise. Estes sistemas não podem usar a mesma implementação polifásica que os QMF. No entanto, Vaidyanathan propôs uma estrutura em *lattice* para implementação eficiente de qualquer CQF [158]. Na verdade, a estrutura em *lattice* é uma forma canónica de implementar qualquer sistema de dois canais com matriz de componentes polifásicas $\mathbf{E}(z)$ *paraunitária*. Esta propriedade define uma importante subclasse de sistemas de decomposição multifrequência porque implica uma série de características desejáveis entre as quais se destaca a reconstrução perfeita [157]. O custo computacional da implementação em *lattice* é praticamente o mesmo que a dos QMF para filtros do mesmo comprimento.[7] Além disso, a paraunitaridade e todas as propriedades associadas, particularmente a reconstrução perfeita, são asseguradas mesmo quando há quantização dos coeficientes do *lattice*. Outra vantagem desta estrutura é que permite um método de projecto versátil, baseado na optimização directa dos coeficientes do *lattice* de modo a minimizar um qualquer critério de aproximação às características pretendidas. O sistema resultante tem sempre reconstrução perfeita independentemente da qualidade da optimização.

3.3.3 Pseudo-QMF

Para efectuar decomposições num grande número de bandas, pode usar-se uma árvore binária de filtros QMF ou CQF. Uma estrutura destas preserva as propriedades de cancelamento de *aliasing* ou reconstrução perfeita dos filtros constituintes. No entanto, esta técnica tem algumas desvantagens: cada subdivisão introduz cada vez mais atraso, uma vez que o ritmo de amostragem vai diminuindo; e as respostas em frequência de algumas das sub-bandas resultantes apresentam lóbulos laterais consideráveis devido à sobreposição de termos de *aliasing* dos diversos andares da árvore (ver [103, Sec. 3.3]). Para minorar esse fenómeno é necessário usar filtros de ordem elevada, o que acarreta maior complexidade e atraso global.

[7] Para determinado conjunto de especificações, os filtros QMF poderão ser mais ou menos longos que os CQF consoante o nível de distorção de amplitude que se considere aceitável.

A alternativa é usar bancos de filtros projectados de raíz para $M > 2$ canais. Um dos primeiros sistemas desenvolvidos para decomposição em M bandas criticamente decimadas foi o chamado *pseudo-QMF*. Cox [28] refere o desenvolvimento independente destes filtros em [119] e [132], neste último com o nome de *polyphase quadrature filters*. Os pseudo-QMF podem considerar-se uma generalização das propriedades QMF para um maior número de sub-bandas. Os filtros são projectados de forma a garantirem cancelamento de *aliasing* mas apenas entre bandas adjacentes. Deste modo consegue-se a eliminação da componente de *aliasing* mais significativa, correspondente às regiões de transição dos filtros. As componentes não canceladas correspondem às bandas de corte que por isso devem ser fortemente atenuadas.

Uma característica importante destes bancos é que os filtros passa-banda são versões moduladas de um único filtro passa-baixo, o filtro *protótipo*. Isto garante que as respostas dos filtros têm todas a mesma forma (embora deslocadas na frequência), como convém para uma decomposição uniforme. Além disso, a modulação possibilita uma implementação eficiente baseada em transformadas rápidas. Outra vantagem é a facilidade de projecto: só é necessário projectar um filtro—o protótipo passa-baixo—e o procedimento é semelhante ao usado para os QMF, com ligeiras modificações.

Apesar de não permitirem reconstrução perfeita nem mesmo total cancelamento de *aliasing*, os sistemas pseudo-QMF encontraram grande aplicação em codificadores de áudio (MUSICAM e MPEG) devido à sua eficiência e facilidade de projecto. Porém, actualmente é possível conseguir reconstrução perfeita sem prescindir dessas vantagens, como se verá adiante.

3.3.4 Bancos de Filtros com Reconstrução Perfeita

Em [157] apresentam-se condições necessárias e suficientes para que um banco de filtros uniforme maximamente decimado tenha reconstrução perfeita. Também são apresentadas condições suficientes que impõem algumas restrições adicionais mas que têm vantagens em termos de projecto e implementação. A mais importante destas restrições é a que força a paraunitaridade da matriz de componentes polifásicas $\mathbf{E}(z)$. Esta condição facilita o projecto do sistema e permite uma implementação em cascata com complexidade reduzida. Pode mostrar-se que a *lapped orthogonal transform* (LOT) [104] é um exemplo de banco de filtros que satisfaz esta condição usando filtros de comprimento $N = 2M$.

Apesar de permitirem estruturas mais eficientes que a implementação directa, os bancos de filtros paraunitários continuam a ter uma complexidade elevada quando o número de canais é grande. Para resolver este problema é necessário restringir ainda mais a forma dos filtros usados no sistema. Em particular, restringindo os filtros a serem versões moduladas de um só filtro protótipo, conseguem-se implementações muito eficientes baseadas em transformadas rápidas. Na subsecção seguinte focaremos estes sistemas que são os mais interessantes para aplicação à compressão de áudio.

3.3.5 Bancos de Filtros Modulados com Reconstrução Perfeita

Um banco de filtros diz-se *modulado* quando os seus filtros de análise $h_b(n)$ e síntese $f_b(n)$ são obtidos por modulação sinusoidal de filtros protótipos $h(n)$ e $f(n)$:

$$h_b(n) = h(n)\cos\left(\frac{\pi}{2M}(2b-1)n + \alpha_b\right) \tag{3.5}$$

$$f_b(n) = f(n)\cos\left(\frac{\pi}{2M}(2b-1)n + \beta_b\right) \tag{3.6}$$

para $b = 1, 2, \ldots, M$. As frequências de modulação correspondem às frequências centrais de M bandas uniformes com empilhamento ímpar [29] e os protótipos devem ser passa-baixo com largura de banda igual a $\pi/2M$. Os filtros pseudo-QMF referidos atrás são filtros com este tipo de modulação mas não têm reconstrução perfeita, como mencionámos.

Para ter reconstrução perfeita, é necessário que os desvios de fase α_b e β_b sejam escolhidos apropriadamente e que os protótipos satisfaçam determinadas condições. Em [56] refere-se que as condições de reconstrução perfeita de um banco de M filtros modulados são equivalentes às condições PR de um conjunto de cerca de $M/2$ bancos de 2 canais formados com pares de componentes polifásicas de $h(n)$ e $f(n)$.

Os resultados apresentados nesse artigo são muito gerais e aplicam-se a bancos com qualquer número de bandas e filtros de qualquer comprimento N (inclusive IIR). Porém, os primeiros bancos de filtros modulados com reconstrução perfeita surgiram para o caso $N = 2M$ (se não contarmos o caso das transformadas em blocos, em que $N = M$) e receberam diversos nomes: bancos de filtros com cancelamento de *aliasing* no domínio do tempo (TDAC) [122]; *modulated lapped transforms* (MLT) [101]; e também, DCT modificada (MDCT). Em [102] generalizaram-se as condições de reconstrução perfeita para filtros de comprimento $N = 2KM$ com K natural, dando origem às *extended lapped transforms* (ELT). Condições equivalentes no domínio da frequência foram dadas em [87], pondo em evidência a paraunitaridade da matriz de componentes polifásicas associada.

A ELT pode ser implementada com um conjunto de estruturas *lattice* e uma transformada discreta de cossenos (DCT). Esta implementação tem um baixo custo computacional, comparável ao dos pseudo-QMF para filtros de comprimento igual. O custo é especialmente reduzido se o número de bandas for uma potência de dois, o que simplifica a DCT. A implementação com filtros *lattice* tem outra vantagem: possibilita um método de projecto semelhante ao mencionado acima para os bancos CQF de dois canais. O método baseia-se na optimização dos parâmetros (ângulos) das estruturas *lattice* de forma a minimizar a energia da banda de corte do filtro protótipo resultante:[8]

$$E_S = \frac{1}{\pi} \int_{\omega_S}^{\pi} |H(e^{j\omega})|^2 \, d\omega.$$

A frequência de corte ω_S deve ser maior que $\pi/2M$ (que é a banda passante nominal do protótipo) e em geral é especificada perto de π/M. É possível usar outros critérios de erro tais como o da minimização da máxima amplitude na banda de corte, que conduz a filtros aproximadamente *equiripple*. No entanto, a função E_S é vantajosa porque é calculável analiticamente a partir da resposta impulsional do protótipo, sem ser necessário fazer integração numérica [157, Sec. 6.4.3].

3.4 Bancos de Filtros Não Uniformes

Um banco de filtros multi-resolução não uniforme pode ser implementado na forma directa da Figura 3.1. No entanto, tal como para bancos uniformes, a implementação directa implica uma enorme complexidade computacional e de projecto. Por esta razão, o desenvolvimento de estruturas de decomposição não uniforme tem adoptado uma solução alternativa: a com-

[8] Note-se que os coeficientes do protótipo podem ser calculados univocamente a partir dos ângulos da estrutura.

binação de bancos uniformes eficientes usando estruturas em árvore ou estruturas *split-and-merge* (definidas adiante).

3.4.1 Estrutura em Árvore

Esta é a maneira mais óbvia de conseguir uma decomposição não uniforme: começar por dividir o sinal em bandas uniformes e a seguir subdividir algumas dessas bandas usando outro banco uniforme. O processo pode aplicar-se repetidamente, resultando numa estrutura com uma topologia em árvore como se representa na Figura 3.2. O banco de síntese tem, naturalmente, uma estrutura dual da de análise.

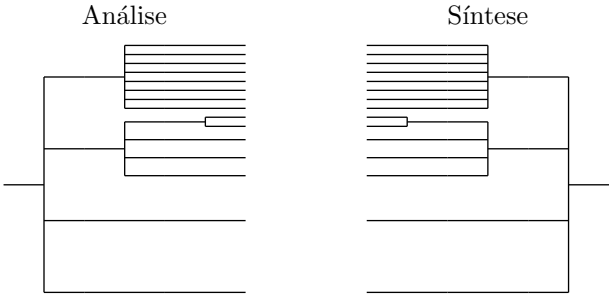


Figura 3.2: Decomposição não uniforme usando uma estrutura em árvore.

Se os bancos uniformes utilizados tiverem a propriedade de reconstrução perfeita, então a estrutura em árvore também terá, desde que se introduzam unidades de atraso em pontos estratégicos para igualar os atrasos nos vários ramos.

O caso mais comum de estrutura em árvore são as estruturas *diádicas* formadas por bancos de dois canais aplicados sucessivamente sempre à banda mais baixa de modo a produzir uma decomposição em oitavas. As *discrete-time wavelet transforms* (DTWT) também podem ser implementadas por estruturas diádicas [143]. Outro exemplo de estrutura em árvore são as *hierarchical lapped transforms* (HLT), nas quais os andares de decomposição são formados por ELTs [103].

Aspectos de Implementação

Há dois aspectos importantes na implementação de uma estrutura em árvore que não são visíveis na Figura 3.2: a necessidade de atrasos de compensação e a reflexão das bandas pares.

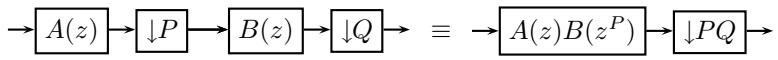
Atrasos de Compensação Uma vez que os sistemas de análise-síntese usados para subdividir as várias bandas de um andar não introduzem todas o mesmo atraso (além de eventual distorção), é necessário adicionar linhas de atraso aos bancos mais rápidos para os sincronizar com os mais lentos. Os atrasos de compensação devem ser colocados nos pontos disponíveis com melhor resolução temporal num dado subsistema, isto é: *antes* do banco de decomposição e *depois* do banco de reconstrução. Em princípio, os atrasos a introduzir podem ser repartidos de forma arbitrária entre o banco de análise e o de síntese mas há dois casos que parecem mais interessantes: colocar todos os atrasos no banco de análise para simplificar (ligeiramente) os

descodificadores; ou reparti-los igualmente para manter uma simetria perfeita.[9] Na determinação dos atrasos de compensação tem que se considerar que as unidades de atraso não têm todas a mesma duração porque as várias bandas têm frequências de amostragem diferentes. Pode mesmo ser necessário introduzir atrasos nas bandas mais lentas só para atingir o *mínimo atraso comum* de todas as bandas.

Reflexão das Bandas Pares Considere-se a banda #2 de um dado banco de filtros uniforme. A sua banda passante estende-se de π/M a $2\pi/M$. Após decimação pelo factor M, a componente passante de frequência mais baixa é transladada para a frequência π, enquanto a componente de alta frequência passa para $2\pi \equiv 0$. Há portanto uma inversão de frequências ou *reflexão*[10] da banda #2, e acontece o mesmo com todas as outras bandas de índice par (*bandas pares*, para simplificar). Assim, quando se subdivide uma banda par, resulta que as sub-bandas de índice mais baixo cobrem as frequências mais altas e vice-versa. Após algumas subdivisões sucessivas, a relação entre índices e frequências das bandas torna-se complexa. Para evitar este inconveniente pode-se simplesmente trocar a ordem das sub-bandas ou, alternativamente, multiplicar as bandas pares pela sequência $(-1)^{n_b}$, o que praticamente não aumenta a complexidade.

Funções de Transferência das Sub-bandas

O percurso entre a entrada e uma das saídas de uma estrutura em árvore é uma cascata de filtros passa-banda decimados. Aplicando as chamadas *identidades nobres* [157], pode reduzir-se essa cascata a um único filtro decimado aplicando a seguinte regra:



Procedendo do mesmo modo com as outras saídas, verifica-se que a estrutura em árvore se reduz à forma directa da Figura 3.1. Assim, quando falamos de *função de transferência* de uma banda numa estrutura em árvore, referimo-nos à função de transferência do filtro correspondente na forma directa equivalente, antes de sofrer decimação.

Considere-se uma estrutura em árvore formada por uma decomposição em M bandas e uma subdivisão adicional da banda #1 em 8 sub-bandas. Deslocando o decimador $\downarrow M$ para a saída, as funções de transferência dessas sub-bandas ficam:

$$H_b(z) = A_1(z)B_b(z^M),$$

onde $A_1(z)$ e $B_b(z)$ representam as funções de transferência da banda #1 do primeiro andar de decomposição e da banda #b do segundo andar, respectivamente. A Figura 3.3 mostra a amplitude das funções de transferência $H_b(z)$ construídas a partir de filtros $A_1(z)$ e $B_b(z)$ típicos.

Observa-se que as bandas mais altas apresentam "fugas" significativas fora das suas bandas passantes. Em particular, a banda #7 tem um importante lóbulo secundário muito inconveniente que invade a região de frequências de uma eventual sub-banda #2 da banda #2 (caso esta fosse subdividida igualmente em 8 canais). Este fenómeno deve-se à fraca atenuação das réplicas espectrais das sub-bandas que "residem" na região de transição do filtro do andar anterior. Há várias formas de minimizar este problema:

[9] Quase perfeita no caso de o número de atrasos ser ímpar.

[10] Trata-se, de facto, de uma *translação* das frequências, que no caso de sinais reais equivale simplesmente a uma inversão.

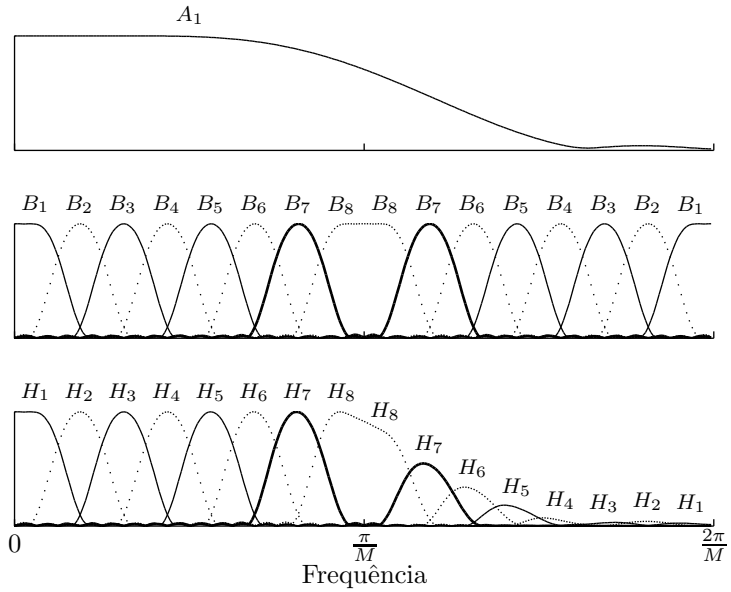


Figura 3.3: Funções de transferência de uma estrutura de análise em árvore. As funções de transferência do segundo andar, $B_b(z)$, aparecem comprimidas e replicadas M vezes. Representa-se apenas uma fracção do espectro total. O eixo vertical mede a amplitude das respostas numa escala linear.

1. Usar apenas bancos de subdivisão com poucas bandas (tipicamente duas, resultando em árvores binárias).

2. Garantir regiões de transição muito estreitas em todas as bandas que são subdivididas.

3. Fazer um pós-processamento dos sinais de saída com um bloco de redução de *aliasing* como se faz no MPEG *Layer* III.

A primeira solução é bastante restritiva e pode obrigar à construção de estruturas com muitos andares e muito atraso. Pode encontrar-se um exemplo em [121]. A segunda solução implica a utilização de filtros de ordem elevada, com grande complexidade computacional e atraso. A terceira solução, proposta em [34] e aplicada ao banco de filtros híbrido do *Layer* III, consiste num andar de pós-processamento com estruturas em "borboleta" que combinam, duas-a-duas, as sub-bandas que mais interferem entre si. Ajustando adequadamente os coeficientes das borboletas, consegue-se minorar a amplitude dos lóbulos secundários. Esta técnica é interessante porque não aumenta muito a complexidade e não introduz atraso adicional. No entanto, só é facilmente aplicável a estruturas em que todas as bandas são igualmente subdivididas, o que é um caso com pouco interesse visto que há formas menos problemáticas de implementar bancos uniformes.[11] Não encontrámos na bibliografia consultada qualquer referência à forma de aplicar esta técnica a estruturas não uniformes em geral.[12]

[11] O *Layer* III usa esta técnica presumivelmente para aproveitar o banco de filtros que já existe para os *Layers* I e II.

[12] É certo que o *Layer* III prevê a possibilidade de decomposição não uniforme ao permitir a comutação para "janelas curtas" em apenas algumas bandas, porém [77] não esclarece como é que a redução de *aliasing* se

Pode concluir-se que o problema das "fugas" é intrínseco às estruturas em árvore e não é fácil de resolver. Torna-se necessário estudar formas alternativas de decomposição não uniforme.

3.4.2 Estruturas *Split-and-Merge*

Outra forma de fazer análise não uniforme consiste em fazer uma decomposição inicial num grande número de bandas e, num segundo andar, recombinar grupos de bandas usando filtros de síntese (ou transformadas inversas). O primeiro andar produz bandas com a largura mínima pretendida enquanto o segundo andar "troca" resolução espectral por resolução temporal para produzir bandas mais largas com maior ritmo de amostragem. A estruturas deste género chamamos *split-and-merge* (SAM). A Figura 3.4 mostra um banco de filtros SAM com a mesma resolução que a estrutura em árvore da Figura 3.2.

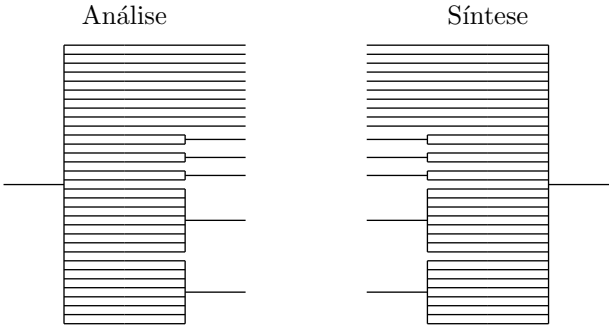


Figura 3.4: Decomposição não uniforme usando uma estrutura *split-and-merge*.

Se no sistema completo de análise e síntese os bancos uniformes usados formarem pares com reconstrução perfeita, então o sistema completo também terá essa propriedade desde que se incluam atrasos de compensação convenientes.

Aspectos de Implementação

Tal como na estrutura em árvore, são necessários atrasos de compensação e reflexão das bandas pares. Nas estruturas SAM os atrasos podem ser colocados nas saídas do bloco de análise e/ou nas entradas do bloco de síntese. Colocam-se os mesmos problemas de determinação do número de atrasos que já se referiram para as estruturas em árvore.

Quanto à necessidade de reflexão de bandas, podemos distinguir duas situações. Se as entradas de um banco de recombinação estiverem *alinhadas* com as saídas do primeiro andar—isto é: pares com pares, ímpares com ímpares—então não há necessidade de reflexão dessas bandas porque o fenómeno de inversão de frequências nas bandas pares do primeiro andar é compensado por uma segunda inversão no segundo andar. Se, pelo contrário, as bandas estiverem desalinhadas, é conveniente invertê-las todas para que o sinal resultante não fique também invertido. Podemos resumir quais as bandas que necessitam de reflexão numa regra simples: reflectem-se todas as bandas pares, quer as saídas de bancos de decomposição quer as entradas em bancos de recombinação. Isto é válido, como se verifica facilmente, tanto

processaria neste caso.

nas estruturas em árvore como nas estruturas SAM, e tanto no bloco de análise como no de síntese.

Funções de Transferência das Sub-bandas

Ao contrário das árvores, as estruturas SAM não podem ser reduzidas à forma directa de um banco de filtros com decimação inteira. Em geral, não podem sequer ser reduzidas a um banco com decimação fraccionária como definido em [116].[13] Por esta razão não se pode definir *função de transferência* tão facilmente como nas estruturas em árvore.

Considere-se o percurso do sinal entre a entrada de um banco de análise SAM e uma das suas saídas, representado na Figura 3.5. É fácil verificar que se $A_{J+1}, \ldots, A_{J+L}$ e $B_1, \ldots, B_L$

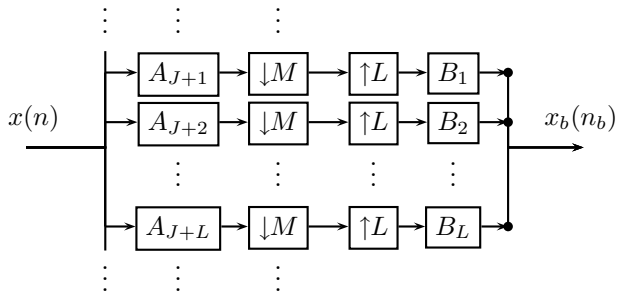


Figura 3.5: Uma banda de um banco de análise SAM.

forem filtros passa-banda ideais, com larguras de banda de π/M e π/L respectivamente; então o sinal de saída, x_b, conterá exacta e unicamente as componentes espectrais do sinal de entrada extraídas pelos filtros A_{J+1} a A_{J+L}. Nesse caso, a função de transferência é dada por:

$$H_b = \sum_{j=J+1}^{J+L} A_j,$$

ou seja, é um filtro passa-banda ideal de largura $\pi L/M$. Na prática porém, os filtros não são ideais, e portanto a função de transferência não é rectangular. Acresce ainda que a saída vem afectada por termos de *aliasing* do sinal de entrada. No entanto, se os filtros do andar de decomposição tiverem regiões de transição idênticas (quando representadas em frequência absoluta) às dos filtros do andar de recombinação, então a maior parte do *aliasing* é cancelado, num processo semelhante ao que ocorre entre os bancos de análise e de síntese de um sistema pseudo-QMF. Nestas condições, as respostas impulsionais de cada banda são praticamente invariantes no tempo. As pequenas variações devem-se ao pouco *aliasing* que não é cancelado. Consequentemente, pode avaliar-se aproximadamente o desempenho de cada banda através de qualquer das suas respostas impulsionais e/ou da correspondente transformada de Fourier, que constitui uma aproximação à função de transferência real.

Uma forma simples de garantir a semelhança entre os filtros dos andares de decomposição e recombinação é usar bancos de filtros modulados, todos derivados de protótipos com forma

[13] Isso só é possível se os factores de interpolação e de decimação forem primos entre si, permitindo a permutação dos decimadores e dos interpoladores.

idêntica. Esta estratégia foi proposta em [28]. No mesmo artigo, Cox usou uma modificação do programa de Johnston [82] para projectar diversos bancos pseudo-QMF, e observou que a forma do protótipo obtido praticamente não depende do número de bandas do banco, desde que se mantenham fixos os seguintes parâmetros de projecto:

- LB, a razão entre o comprimento dos filtros e o número de bandas, N/M; e
- TP, a razão entre a largura da banda de transição e a largura nominal da banda passante.

Deste modo, cada par (LB, TP) define uma família de bancos pseudo-QMF com funções de transferência idênticas independentemente do número de bandas, o que permite a construção de bancos não uniformes SAM com a desejada propriedade de cancelamento parcial de *aliasing* entre andares.

Em simulações semelhantes às realizadas por Cox, verificámos que outro tipo de banco de filtros modulados, a ELT, exibe o mesmo comportamento: a forma do filtro protótipo depende apenas das razões LB e TP, e não do número de bandas (dimensão) da transformada. Isto possibilita a construção de estruturas SAM que, além do cancelamento parcial de *aliasing* entre andares, garantem reconstrução perfeita no sistema análise-síntese completo.

Capítulo 4

Quantização

Quantizar consiste em aproximar uma quantidade de natureza contínua por um valor retirado de um conjunto descontínuo de valores isolados. O exemplo típico é o arredondamento de um número real ao inteiro mais próximo. O termo parece ter origem na Teoria Quântica que restringe certas quantidades físicas, anteriormente consideradas contínuas, a assumir apenas valores múltiplos de certos valores mínimos chamados *quanta*. No domínio das telecomunicações, o termo *quantização* foi adoptado desde meados do século XX para descrever a operação que, juntamente com a *amostragem*, forma a base da transmissão digital de sinais.

O artigo de Gray e Neuhoff [58] apresenta uma perspectiva histórica, uma extensa revisão bibliográfica, e um resumo dos principais resultados da investigação e desenvolvimento da quantização. É a nossa principal referência neste capítulo e dela adoptamos uma boa parte da notação e terminologia.

4.1 Conceitos Básicos

A operação de quantização de um valor $x \in \Re$ (quantização escalar) ou de um vector $x \in \Re^m$ (quantização vectorial) pode ser descrita pela expressão

$$q(x) = y_k \quad \text{se} \quad x \in S_k, \tag{4.1}$$

onde os y_k são os *valores*, *níveis* ou *vectores de reconstrução* e os conjuntos S_k são os *intervalos* ou *células de quantização*. Estes intervalos formam uma partição do domínio de x, ou seja: não se intersectam entre si e em conjunto cobrem todo o domínio. Assim, para cada valor de x existe um e um só k (e y_k) que satisfaz a expressão acima. Os *índices* ou *símbolos* k pertencem a um conjunto geralmente finito ou pelo menos numerável e constituem, afinal, a informação a transmitir entre codificador e descodificador. Naturalmente, esta transmissão implica uma codificação reversível que a cada símbolo faz corresponder uma sequência binária de comprimento fixo ou variável. Podem, então, identificar-se várias componentes do sistema de quantização:

1. Um *codificador com perdas*, responsável pela classificação do valor de entrada nos vários intervalos de quantização e representável pela função: $\alpha(x) = k$ se $x \in S_k$.

2. Um *codificador sem perdas* que transforma cada índice k numa sequência binária $\gamma(k)$, e o respectivo descodificador γ^{-1} que faz a transformação inversa.

3. Um *descodificador de reconstrução*, definido por: $\beta(k) = y_k$.

Desta forma, $q(x) = \beta(\gamma^{-1}(\gamma(\alpha(x)))) = \beta(\alpha(x))$.

Neste documento é usual referimo-nos às operações α e β como *quantização*[1] e *desquantização*, respectivamente. A *codificação sem perdas* γ será tratada no Capítulo 5, pelo que para já apenas nos interessa quantificar o número de bits gastos para codificar cada índice k, ou seja quantificar o *comprimento do código*, $\bar{\gamma}(k)$, para o símbolo k.

4.2 Desempenho

O desempenho de um sistema de quantização avalia-se em duas dimensões: débito e distorção. O *débito* mede o número médio de bits necessários para codificar cada amostra quantizada. Considerando que a amostra é uma variável aleatória X com função densidade de probabilidade $p(x)$, o débito é dado por

$$R = E[\bar{\gamma}(\alpha(X))] = \sum_k p_k \bar{\gamma}(k), \tag{4.2}$$

onde p_k é a probabilidade de o quantizador produzir o símbolo k, ou seja

$$p_k = P(\alpha(X) = k) = P(X \in S_k) = \int_{S_k} p(x)\, dx. \tag{4.3}$$

Para quantizadores com um número finito de níveis N e codificação de comprimento fixo, o comprimento é no mínimo $\bar{\gamma}(k) = \log_2 N$, independentemente de k e logo,

$$R = \log_2 N. \tag{4.4}$$

Este mínimo é aproximável, quando N não for uma potência de 2, fazendo por exemplo o empacotamento e codificação de várias amostras em simultâneo. Usando códigos com comprimento variável, dependente da probabilidade de ocorrência de cada símbolo k, conseguem-se débitos ainda mais baixos. Idealmente, o código terá comprimento $\bar{\gamma}(k) = -\log_2 p_k$, do qual resulta um débito médio igual à *entropia* dessa fonte de informação:

$$R = -\sum_k p_k \log_2 p_k. \tag{4.5}$$

Mais uma vez, este mínimo teórico pode ser aproximado bastante bem na prática pelo uso de codificação aritmética ou combinando empacotamento com outro tipo de códigos como os de Huffman, por exemplo.

A outra dimensão de avaliação dos quantizadores é a *distorção*. Ao substituir um valor x por uma aproximação $q(x)$, o quantizador introduz um erro $x - q(x)$. Para avaliar a importância do erro, define-se uma medida de distorção $d(x, q(x))$, real, não-negativa e que traduza tão fielmente quanto possível a perceptibilidade real ou custo da substituição de x por $q(x)$ no receptor. O valor médio desta medida para um conjunto de entradas descritas por uma variável aleatória X dá-nos uma avaliação global da distorção do quantizador,

$$D = E[d(X, q(X))] = \sum_k \int_{S_k} d(x, y_k) p(x)\, dx. \tag{4.6}$$

[1] O contexto deverá eliminar a ambiguidade do duplo uso deste termo para α ou q.

Por vezes é conveniente combinar débito e distorção, através de um multiplicador de Lagrange $\lambda \geq 0$, numa única medida chamada *custo lagrangiano*,

$$L_\lambda = D + \lambda R. \tag{4.7}$$

A distorção, ao contrário do débito, não tem uma unidade de medida natural, nem uma forma de medição universal. Podem definir-se medidas de distorção com maior ou menor significado perceptual, maior ou menor complexidade, maior ou menor utilidade analítica. A medida de distorção mais comum é, sem dúvida, o *erro quadrático*

$$d(x, q(x)) = ||x - q(x)||^2$$

definido como o quadrado da distância euclidiana entre sinal e reconstrução. (Aqui $||\cdot||$ representa $||\cdot||_2$, a norma l_2 definida mais abaixo.) A simplicidade e conveniência analítica desta expressão justifica a sua popularidade, especialmente em derivações teóricas. No entanto, é sabido que esta não é uma boa medida da perceptibilidade auditiva ou visual (ver Capítulo 6). Assim, alguns trabalhos procuram generalizar as técnicas de projecto e os principais resultados teóricos sobre quantização a medidas de distorção mais realistas. Uma dessas generalizações propõe medidas de distorção da forma

$$d(x, y) = ||x - y||_s^r$$

onde

$$||z||_s = \left(\sum_{i=1}^{n} |z_i|^s \right)^{(1/s)}$$

é a norma l_s do vector $z = [z_1, z_2, \cdots, z_n]$. Esta medida e outras ainda mais gerais têm a forma $d(x, y) = \rho(x - y)$, ou seja, são funções apenas da diferença $x - y$. Em consequência temos, por exemplo num caso escalar, $d(10, 11) = d(1000, 1001)$, o que certamente não é realista em muitas aplicações. Medidas bem mais versáteis e realistas são as que se podem descrever ou aproximar pela expressão

$$d(x, y) = (x - y)^T M(y)(x - y) \tag{4.8}$$

onde $M(y)$ é uma matriz definida positiva, dependente do próprio sinal, por vezes denominada de *matriz de sensibilidade*. Esta formulação, proposta em [48], é aplicável a uma grande classe de medidas, incluindo algumas medidas perceptuais de distorção de imagem e de voz. Vários resultados relativos à quantização com estes critérios de distorção são demonstrados em [94], [95] e [92].

4.3 Optimização

Equipados com estas definições de débito e distorção, é natural que se procure projectar o sistema de quantização de forma a optimizar o seu desempenho. Isto implica escolher as células de quantização S_k do quantizador α, os valores de reconstrução y_k do desquantizador β e o codificador sem perdas γ. A distorção é afectada pela escolha de α e β mas não depende de γ. Por sua vez, o débito é condicionado por α e γ mas é independente de β.

Lloyd [96] e Max [105] foram dos primeiros a dedicarem-se a este problema, mais especificamente ao problema da minimização do erro quadrático médio[2] de um quantizador escalar

[2]Na verdade, Max estudou também outras medidas da diferença.

de comprimento fixo (e N fixo). De forma independente, ambos derivaram condições necessárias para a solução do problema e desenvolveram algoritmos iterativos para a aproximar. O algoritmo de Max é idêntico a um dos métodos anteriormente propostos por Lloyd [96, método II]. O outro método de Lloyd é mais interessante porque admite generalizações para quantizadores vectoriais [93] e para códigos de comprimento variável [26].

Conhecida a distribuição do sinal $p(X)$ e dado um $\lambda > 0$, o algoritmo de Lloyd generalizado, na versão de [26], pode descrever-se informalmente como se segue:

1. Começar com um desquantizador inicial β e um código inicial de comprimento $\bar{\gamma}$.

2. Fixando β e γ, achar o quantizador α' que minimiza o custo lagrangiano médio L_λ do sistema de quantização (α', γ, β). O resultado deste passo é a função α' que a cada x faz corresponder o índice k que minimiza o custo $l_\lambda = d(x, \beta(k)) + \lambda\bar{\gamma}(k)$. Ou seja, o quantizador óptimo é o de *vizinho mais próximo*, subentendendo-se que a proximidade é medida por l_λ.

3. Com o α' determinado atrás, achar um novo desquantizador β' que minimize a distorção $D(\alpha', \beta')$ do sistema. Este desquantizador é a função que para cada k devolve o valor y_k que minimiza $E[d(X, y_k) \,|\, \alpha(X) = k]$, a distorção média condicionada ao facto de a entrada ser quantizada como k. As soluções são os chamados *centróides* das células de quantização e a sua determinação depende naturalmente da medida de distorção. Para o erro quadrático é um resultado clássico que:

   $$y_k = E[X \,|\, X \in S_k], \ \forall k.$$

   Para o erro quadrático ponderado pelo sinal da equação (4.8) vem [58, p. 2342][93, eq. (15)]:

   $$y_k = E[M(X) \,|\, X \in S_k]^{-1} E[M(X)X \,|\, X \in S_k], \ \forall k.$$

4. Para o mesmo α', achar o código γ' que minimiza o débito $R(\alpha', \gamma')$ do sistema. Na prática, presumindo uma codificação de entropia ideal, basta recalcular as probabilidades dos símbolos p_k, dependentes de α', pela equação (4.3) e o débito R pela equação (4.5).

5. Fazer $(\alpha, \gamma, \beta) \leftarrow (\alpha', \gamma', \beta')$ e repetir a partir do passo 2 enquanto houver variação significativa da distorção L_λ após cada iteração.

Alguns aspectos deste algoritmo merecem alguns comentários:

Convergência e terminação. O débito R não depende da escolha do desquantizador β, por isso, no passo 3, a minimização de D implica também uma redução (ou manutenção) do custo $L_\lambda = D + \lambda R$, tal como no passo 2. De igual modo, como D não depende de γ, o passo 4 que minimiza R também implica a redução (ou manutenção) de L_λ. Deste modo, o custo lagrangiano após cada iteração forma uma sequência decrescente (em sentido lato); e como não pode tomar valores negativos, necessariamente a sequência converge e o algoritmo termina. O ponto $L_\lambda(\alpha, \gamma, \beta)$ para o qual o algoritmo tende é um mínimo local da função custo. Não se conhece uma forma geral de atingir um mínimo global, mas a experiência mostra que a aplicação deste algoritmo com diferentes inicializações converge frequentemente para um mínimo global.

Varrimento de λ. Pode ser necessário repetir toda a optimização para outros valores de λ para se obter diferentes compromissos entre débito e distorção. (Ver [26] para um processo sistemático de varrimento de λ.)

Variante amostral. Quando não se conhece a distribuição do sinal X, pode usar-se um conjunto (grande) de vectores de treino representativos da distribuição e aplicar uma variante deste algoritmo na qual se substitui os operadores de probabilidade e esperança matemática por médias amostrais. Na prática, este é o processo usado no projecto de quantizadores vectoriais para aplicações reais.

Inicialização. As condições iniciais do algoritmo podem ter alguma influência no mínimo atingido e, especialmente, na velocidade de convergência. Existem várias propostas para o conjunto inicial de valores de reconstrução $\{y_k\}$ do desquantizador. Um método simples, quando se tem um conjunto de treino, é extrair deste um subconjunto de N elementos de forma determinística ou aleatória. Outra hipótese é usar os valores de reconstrução de um quantizador uniforme que cubra a maior parte do domínio do sinal. Quando se pretende projectar uma família de quantizadores com um número crescente de níveis, pode usar-se uma técnica de subdivisão [93]. Uma proposta mais recente procura maximizar as distâncias entre os valores iniciais [86].

Casos particulares de interesse. Para uma codificação de comprimento fixo, o passo 4 torna-se desnecessário e no passo 2, a minimização de L_λ reduz-se à minimização de D já que R não varia. Neste caso particular, o algoritmo equivale essencialmente ao de [93]. Se adicionalmente condicionarmos o problema a uma dimensão, obtemos o algoritmo de Lloyd original [96, método I].

4.4 Implementação

Já vimos como caracterizar o desempenho de um quantizador, e como o projectar de forma a optimizar esse desempenho. Nesta secção abordamos o problema da implementação de quantizadores.

As várias técnicas de implementação diferem sob diferentes aspectos. Um dos mais importantes é a complexidade computacional, que se pode avaliar pelo número e tipo de operações necessárias para quantizar cada amostra, bem como pela quantidade de memória requerida nesse processo. A simplicidade de certas técnicas consegue-se muitas vezes à custa da imposição de restrições que limitam o tipo de quantização que se pode implementar, com consequências a nível do desempenho. Por isso, a par da complexidade, outro aspecto a considerar é a versatilidade da implementação. Em muitas situações de codificação de sinais, os requisitos de desempenho (qualidade e débito) da quantização não se mantêm constantes para todo o sinal. Isto pode dever-se à variação da dinâmica do sinal, à variação da sensibilidade do receptor (geralmente em função do próprio sinal), à alteração das necessidades ou expectativas de qualidade pelo receptor, ou a restrições impostas ao débito da fonte. Para lidar com esta variabilidade e melhor aproveitar as oportunidades de optimização que ela oferece, interessa que os quantizadores sejam adaptativos. A capacidade de adaptação, bem como a facilidade e rapidez com que podem ser adaptados são também aspectos importantes que diferenciam as várias implementações.

4.4.1 Arredondamento e Quantização Uniforme

O *arredondamento* é a forma mais simples de quantização. Existem várias formas de arredondar um real x a um inteiro: arredondamento por defeito $\lfloor x \rfloor$, igual ao maior inteiro não superior a x; arredondamento por excesso $\lceil x \rceil$, igual ao menor inteiro não inferior a x; ou arredondamento ao inteiro mais próximo, que se pode definir por $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor$.[3] Este último é o mais interessante porque minimiza o erro absoluto máximo e outras medidas relacionadas.

A quantização uniforme genérica, com intervalos de amplitude Δ e níveis de reconstrução centrados nesses intervalos, pode então ser descrita pelas funções

$$\alpha(x) = \left\lfloor \frac{x - y_0}{\Delta} \right\rceil \tag{4.9}$$

e

$$\beta(k) = k\Delta + y_0. \tag{4.10}$$

Estas expressões podem implementar-se facilmente nos microprocessadores actuais já que as operações aritméticas envolvidas são relativamente banais. A mais complexa, a divisão por Δ, poderá ser substituída por uma multiplicação por $1/\Delta$, recorrendo a factores tabelados e acautelando devidamente os problemas da precisão finita das representações numéricas.

Os dois parâmetros livres Δ e y_0 representam o passo de quantização e o primeiro nível de reconstrução, respectivamente. O parâmetro Δ altera a *escala* do quantizador, permitindo controlar a quantidade de distorção introduzida (e o débito produzido). O parâmetro y_0 permite corrigir a polarização dos níveis e intervalos de quantização para melhor os ajustar à distribuição do sinal. Para o caso de distribuições simétricas em torno de zero, é usual fazer-se $y_0 = 0$ ou então $y_0 = \frac{1}{2}\Delta$. Diz-se nestes casos tratar-se respectivamente de quantizadores do tipo *midtread* ou do tipo *midrise*; nomes sugeridos pela posição da origem em relação aos degraus da curva $y = q(x)$ num e noutro caso [80, p. 118]. O efeito deste parâmetro no desempenho do quantizador é desprezável em condições de alta resolução, quando a dispersão do sinal excede bastante o passo de quantização. Noutras condições, porém, pode ter um efeito significativo. Por exemplo, para um sinal de amplitude baixa relativamente a Δ e muito concentrado em torno de zero, um quantizador *midtread* supera claramente um *midrise*. Por conveniência, nas expressões acima omitimos um terceiro parâmetro: o número de níveis N. Este garante um débito máximo de $\log_2 N$ (ou pouco mais), mas simultaneamente limita o conjunto de valores reproduzíveis a uma gama de amplitude $N\Delta$. Se o sinal de entrada tomar valores fora dessa gama, o quantizador satura, produzindo erros superiores a $\Delta/2$, que podem contribuir significativamente para a distorção do sistema. Quando se usa um código de comprimento variável, é normal considerar-se N muito grande, visto que a inclusão de um maior número de níveis improváveis pode reduzir substancialmente a distorção mas pouco acrescenta ao débito.

O passo de quantização Δ é frequentemente alvo de adaptação para fazer face a sinais não-estacionários com variações de gama dinâmica. Também é possível adaptar y_0, por exemplo para acompanhar um sinal cuja média flutue de forma previsível. Equivalentemente, tal sistema pode ser tratado como um codificador DPCM (que acompanha o sinal), seguido de um controlador automático de ganho (que compensa as variações de escala), seguido de um quantizador uniforme de passo e polarização fixos. O número de níveis N pode ser

[3] Outras definições de arredondamento ao mais próximo, como $\lceil x \rfloor = \lceil x - \frac{1}{2} \rceil$, diferem apenas na classificação dos pontos equidistantes de dois inteiros consecutivos, e portanto são essencialmente equivalentes.

adaptado a fim de controlar o débito, especialmente em códigos de comprimento fixo, mas simultaneamente deve adaptar-se Δ para evitar a saturação do quantizador. É o que se faz nos sistemas de vírgula flutuante em blocos (Sec. 2.2.3).

4.4.2 Pesquisa em Tabelas

A forma mais geral de implementar quantização escalar é naturalmente recorrer a tabelas dos níveis de reconstrução y_k e dos níveis de decisão x_k que delimitam os intervalos de quantização $S_k = [x_k, x_{k+1}[$. O quantizador consiste em procurar o intervalo que contém x, o que pode ser feito com cerca de $\log_2 N$ operações de comparação, caso a tabela dos x_k esteja ordenada. A desquantização implica somente um acesso indexado à tabela dos y_k.

Esta implementação oferece potencialmente a máxima adaptabilidade, permitindo o controlo individual de cada intervalo e nível de quantização. Na prática, o grande número de parâmetros livres dificulta o processo. Uma mera alteração de escala obrigaria à alteração de todos os valores tabelados. Obviamente que situações como esta são melhor resolvidas com pré-processamento da entrada, uso de tabelas normalizadas e pós-processamento da saída. Complementarmente, pode manter-se um conjunto limitado de tabelas alternativas e comutar entre elas para conseguir uma adaptação rápida a variações da forma da distribuição do sinal, por exemplo. É claro que o custo em termos de memória fixa, cerca de N palavras por tabela, cresce substancialmente neste caso.

4.4.3 Quantização Paramétrica: Compressão-Expansão

Um método que permite conciliar boa parte da versatilidade dos quantizadores de tabela com a simplicidade dos quantizadores uniformes é uso de funções de compressão-expansão. Qualquer quantizador escalar q pode ser decomposto numa função de *compressão* c, uma quantização uniforme ou, sem perda de generalidade, um arredondamento $\lfloor \cdot \rceil$ e uma função de *expansão* c^{-1}:

$$q(x) = c^{-1}(\lfloor c(x) \rceil),$$

como se ilustra na Figura 4.1. Para essa decomposição ser possível basta que c transforme cada intervalo e nível de quantização de q no respectivo intervalo e nível do quantizador uniforme,

$$c([x_k, x_{k+1}[) = [k - \frac{1}{2}, k + \frac{1}{2}[\wedge c(y_k) = k, \forall k.$$

No entanto, é vantajoso exigir-se adicionalmente que a função c seja monótona, contínua e até diferenciável na maioria dos pontos.

Quando $c(x) = (x - y_0)/\Delta$, obtemos o quantizador uniforme da equação (4.9). Num caso mais geral de c não-linear obtém-se uma quantização não-uniforme, como ilustrado na Figura 4.1(d). Neste caso o passo de quantização, que podemos definir como a largura do intervalo $\Delta_k = x_{k+1} - x_k$, varia em função da amplitude do sinal. Esta variação é descontínua, mas é conveniente considerar uma "suavização" definida informalmente por $\Delta(x) \approx \Delta_k$ com $k = \lfloor c(x) \rceil$. A derivada da função de compressão pode então ser interpretada como o recíproco dessa função passo-de-quantização

$$c'(x) = \frac{1}{\Delta(x)},$$

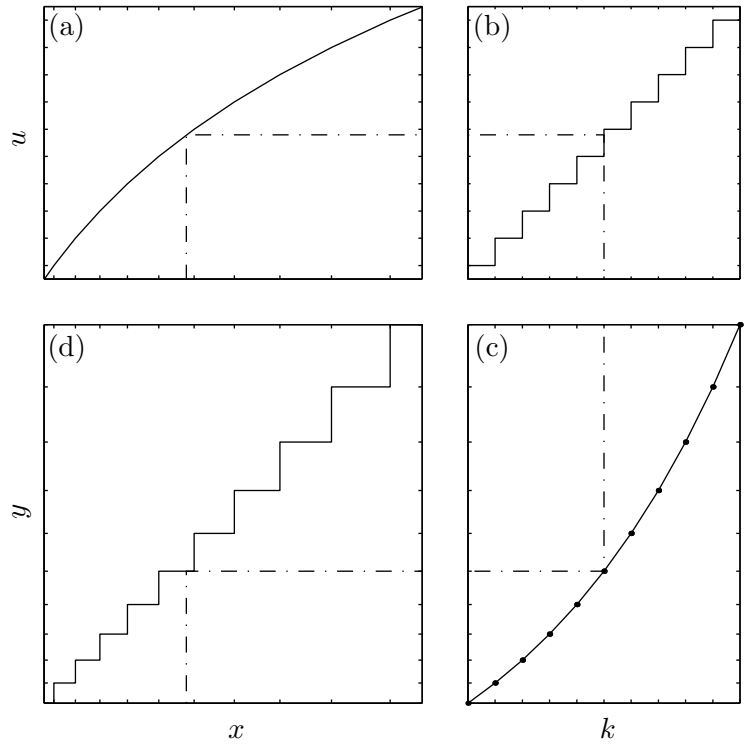


Figura 4.1: Quantização por compressão-expansão. No sentido dos ponteiros do relógio: (a) O valor x é transformado em $u = c(x)$, (b) u é arredondado para $k = \lfloor u \rceil$ (note-se a posição transposta do gráfico), (c) k é expandido para $y = c^{-1}(k)$. (d) A composição das três operações é o quantizador não-uniforme $y = q(x)$.

ou como a *densidade pontual* do quantizador.[4] Estas noções de densidade pontual e função de compressão são fundamentais na teoria da quantização de alta resolução que estabelece limites assimptóticos para o desempenho quando o número de níveis cresce e o tamanho dos intervalos tende para zero [58, Sec. IV].

A complexidade de um quantizador de compressão-expansão é dominada pela implementação das funções c e c^{-1}. O cálculo destas funções pode recorrer a técnicas tradicionais de análise numérica: aproximações polinomiais, pesquisa em tabelas e algoritmos iterativos de aproximações sucessivas. A precisão das aproximações não tem de ser muito superior à resolução do quantizador, o que pode permitir reduzir as exigências computacionais sem afectar muito o desempenho.

Um caso importante são os chamados quantizadores logarítmicos, cujas funções de compressão aproximam uma curva logarítmica para amplitudes altas e são quase lineares para amplitudes baixas. Funções destas têm sido aplicadas sobretudo em telefonia digital. Por exemplo, a lei-μ, usada no sistema telefónico norte-americano é descrita por

$$c(x) = M \frac{\log(1 + \mu|x|/x_M)}{\log(1 + \mu)} \operatorname{sgn}(x),$$

onde os parâmetros x_M e M controlam a escala e resolução do quantizador, enquanto μ define

[4] A densidade pontual de um quantizador é uma função $\Lambda(x)$ que indica o o número aproximado de níveis de quantização por unidade de amplitude em torno de x.

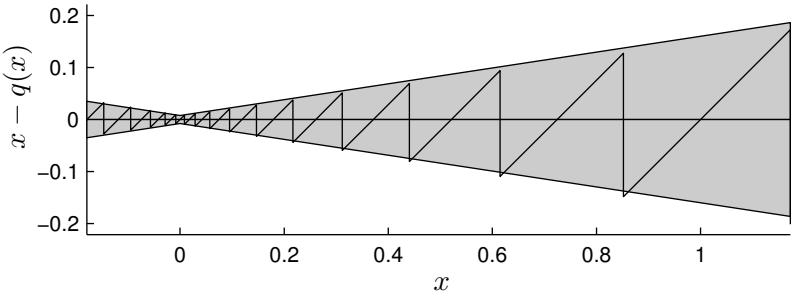


Figura 4.2: Erro de quantização $x - q(x)$ de um quantizador de lei-μ. A sombreado indica-se o intervalo $[-\Delta(x)/2, +\Delta(x)/2]$.

a curvatura da função. A Figura 4.2 mostra a função de erro $x - q(x)$ de um quantizador com a lei-μ. O passo de quantização cresce linearmente com a amplitude do sinal. A função

$$\Delta(x) = \frac{1}{c'(x)} = \frac{\log(1+\mu)}{M}\left(\frac{x_M}{\mu} + |x|\right)$$

prevê esta variação com relativa precisão (ver figura).

Na prática, estas curvas logarítmicas são aproximadas por funções de segmentos lineares. Cada segmento define uma gama de valores quantizada uniformemente com um dado passo, e a amplitude dos passos é duplicada de segmento para segmento. Este tipo de compressão-expansão permite uma implementação digital simples com aritmética binária inteira. A representação binária de um índice de quantização pode também ser vista como um tipo de representação em vírgula flutuante do valor quantizado: 1 bit para o sinal, alguns bits para indicar o segmento (ou o expoente), e os bits restantes indicam o nível dentro do segmento (ou a mantissa).

4.4.4 Quantização Vectorial

Um quantizador vectorial agrupa as amostras de um sinal em vectores de dimensão fixa e para cada um, pesquisa numa lista de vectores-padrão o vector que mais se lhe aproxima.[5] O índice ou código do padrão seleccionado é transmitido ao receptor que, possuindo uma cópia da mesma lista, faz uma tradução imediata e reproduz esse padrão à saída. A implementação de um sistema de quantização vectorial é, pelo menos conceptualmente, muito simples: o codificador é composto por uma lista de padrões (*codebook*) armazenados em memória fixa, por um procedimento de cálculo de uma medida de distorção entre vectores, e por um algoritmo de pesquisa que minimize a distorção entre o vector original e o padrão seleccionado; o descodificador contém simplesmente o mesmo *codebook* e um mecanismo de indexação ou endereçamento directo.

O projecto de um sistema destes consiste essencialmente na compilação de um conjunto de padrões que minimize o valor esperado da distorção, tendo em conta a distribuição de probabilidade da fonte. Quando não se conhece a verdadeira distribuição dos vectores, usa-se um processo iterativo que tenta minimizar a distorção média avaliada para uma sequência de

[5] Trata-se portanto de um algoritmo de procura do *vizinho mais próximo*, como os usados em reconhecimento de padrões.

vectores de treino. Se a sequência de treino for suficientemente grande, esta aproximação dá bons resultados mesmo para processos não estacionários e não ergódicos [57].

A simplicidade do codificador e especialmente do descodificador tornam a quantização vectorial propícia a uma implementação em *hardware* projectado "à medida" para a aplicação desejada. Outro aspecto interessante dos sistemas de quantização vectorial é a forma como podem ser adequados não só às características da fonte mas também às limitações do receptor pela integração de uma medida de distorção perceptualmente significativa.

Apesar do desempenho potencialmente óptimo e da relativa simplicidade algorítmica, as técnicas de quantização vectorial deparam-se com uma grande dificuldade: o crescimento exponencial da complexidade computacional com a dimensão dos vectores. Se considerarmos um *codebook* com N vectores de dimensão m, então a taxa de transmissão é $R = \frac{1}{m}\log_2 N$ bits por amostra e, equivalentemente, $N = 2^{mR}$. Para armazenar esse *codebook* é necessária uma capacidade de memória de $mN = m2^{mR}$ palavras. Simultaneamente, se usar um algoritmo de pesquisa exaustiva, o codificador precisa de fazer 2^{mR} operações por vector ou seja $2^{mR}/m$ operações por amostra. Pode verificar-se que para R fixo o consumo de recursos depende exponencialmente da dimensão m. Esta característica limita a aplicabilidade da quantização vectorial a sinais de dimensionalidade baixa e é responsável pela tardia aceitação desta técnica.

Felizmente, existem algumas técnicas que permitem reduzir o espaço de memória e/ou o tempo de cálculo necessários [49, 57]. Uma dessas técnicas consiste em estruturar a lista de padrões como uma árvore binária que pode ser pesquisada muito rapidamente por um método de aproximações sucessivas. Em contrapartida, é necessária mais memória e o *codebook* pode não ser óptimo devido às condições impostas na escolha dos padrões. Uma alternativa que não compromete a qualidade do codificador baseia-se na geração, através de uma optimização na fase de projecto, de uma árvore de decisão não uniforme adaptada à lista de padrões. Outra possibilidade permite reduzir o consumo de memória e de tempo usando vários estágios de quantização aplicados sucessivamente aos resíduos dos estágios anteriores. Foram propostas muitas outras soluções, quer reduzindo a complexidade do algoritmo básico, quer integrando-o com outros processos de compressão, que poderão vir a tornar mais atractiva esta técnica de codificação.

A quantização vectorial foi já aplicada com bastante sucesso ao problema da codificação da voz [50]. Neste contexto, as técnicas mais eficientes—*vector excitation coding* (VXC) ou *code-excited linear prediction* (CELP)—usam um processo de análise-por-síntese onde os filtros adaptativos de um codificador preditivo são excitados pela saída de um quantizador vectorial. A quantização vectorial conjunta de coeficientes de predição linear também se revelou muito eficaz porque permite explorar a forte dependência não-linear entre os vários coeficientes de predição. Com este processo foram desenvolvidos codificadores de voz a 800 bit/s de qualidade comparável a codificadores com quantização escalar com débito três vezes superior.

4.5 Discussão

Um resultado teórico muitas vezes citado e um tanto surpreendente conclui que, independentemente da distribuição do sinal, o melhor quantizador escalar de débito variável é, o quantizador uniforme [53]. Este é um resultado assimptótico, mas também se aplica aproximadamente quando o número de níveis é baixo. No entanto, esta conclusão não é generalizável a medidas de distorção que não dependam exclusivamente da diferença $x - y$.

Como veremos no Capítulo 6, as medidas perceptuais de distorção dependem fortemente

também das amplitudes. Assim, não é apropriado aplicar directamente quantizadores uniformes nesta situação. Porém, uma função de compressão adequada permite transformar a medida de distorção numa medida de diferenças.

A forma iterativa do algoritmo de Lloyd generalizado sugere a possibilidade de uma implementação incremental, que vá corrigindo adaptativamente os quantizadores mesmo em situações de variação dinâmica dos critérios de desempenho.

Capítulo 5

Codificação Sem Perdas

No capítulo anterior desagregámos da quantização o problema da codificação sem perdas dos símbolos produzidos pelo quantizador. Há duas razões principais para esta separação. Primeiro, porque as técnicas de codificação sem perdas são aplicáveis fora do âmbito da quantização, por exemplo na compressão directa de sinais simbólicos como texto. Além disso, a codificação sem perdas é essencialmente um problema resolvido. A codificação óptima é realizável (muito aproximadamente), pelo que se pode projectar o quantizador sem conhecer exactamente as palavras de código, bastam os seus comprimentos óptimos $\bar{\gamma}(k) = -\log_2 p_k$.

5.1 Definições e Conceitos

Um sistema de *codificação sem perdas* (ou de *codificação reversível*) é formado por:

- Um dispositivo codificador, que transforma uma *mensagem* composta por uma sequência de *símbolos de entrada* $K_n = k_1 k_2 \ldots k_n$ num *código* composto por uma sequência de *símbolos de saída* $S_m = s_1 s_2 \ldots s_m$.

- Um dispositivo descodificador que, a partir da segunda sequência, é capaz de recuperar a primeira, *sem qualquer distorção*, embora eventualmente com algum atraso.

Os símbolos de entrada pertencem a um conjunto numerável A chamado *alfabeto de entrada*, enquanto os de saída pertencem a um *alfabeto de saída*, que é usualmente o alfabeto binário $\{0, 1\}$. Qualquer dos dispositivos pode ter *memória*, pelo que a saída num certo instante não depende somente da entrada no mesmo instante, mas também do passado capturado no estado da memória.

Esta definição é mais geral e mais adequada aos diversos sistemas de codificação existentes do que a definição simples dada no Capítulo 4. Aí, tanto o codificador sem perdas como o descodificador eram descritos como *funções* e não como *máquinas de estados*, com a consequente incapacidade de modelar mudanças de comportamento dependentes do contexto. Em todo o caso, é fácil adaptar as expressões dadas nesse capítulo à definição mais geral.

5.2 Códigos de Comprimento Fixo, Empacotamento e Representações Numéricas

A forma mais simples de codificar símbolos de um alfabeto A com N elementos é associar bi-univocamente a cada símbolo de A uma palavra binária de comprimento fixo. Esse comprimento tem de ser, no mínimo, $\bar{\gamma} = \lceil \log_2 N \rceil$ bits. Quando N não é uma potência de 2, desperdiçam-se palavras de código. Por exemplo, com um alfabeto de 5 símbolos, seriam precisos códigos de $\lceil \log_2 5 \rceil = 3$ bits, capazes de representar $2^3 = 8$ símbolos distintos. No entanto, se agruparmos os símbolos em ternos (k_1, k_2, k_3), passamos a ter um *alfabeto extendido* A^3 com $5^3 = 125$ ternos distintos. Os ternos podem então ser representados com códigos de $\lceil \log_2 125 \rceil = 7$ bits, o que dá um comprimento médio por símbolo de $7/3 \approx 2.33$ bits, bastante mais próximo do valor limite $\log_2 5 \approx 2.32$. Esta forma de *extensão do alfabeto* por agrupamento de símbolos em blocos é um processo usual de conseguir um pouco mais de eficiência a partir de certas técnicas de codificação.

O agrupamento de símbolos em blocos pode ser implementado com aritmética inteira. Se os símbolos forem representados por números inteiros $k_i \in \{0, 1, \ldots, N-1\}$, então um bloco de m símbolos pode representar-se por um inteiro $S \in \{0, 1, \ldots, N^m - 1\}$, calculado pela expressão

$$S = k_1 N^{m-1} + k_2 N^{m-2} + \cdots + k_{m-1} N + k_m.$$

Para a descodificar basta fazer sucessivas divisões inteiras por N ou recorrer a uma tabela indexada por S. Por outras palavras: a sequência $k_1 k_2 \ldots k_m$ não é mais do que a representação do número S na base N; e os processos de codificação e descodificação resumem-se a conversões entre representações numéricas na base N e na base 2. É fácil generalizar este processo para blocos de símbolos pertencentes a alfabetos de diferentes dimensões.

Mesmo esta técnica simples não pode ser traduzida por uma função $\gamma(k)$ já que a sequência binária debitada pelo codificador não depende simplesmente do símbolo k mas também da sua posição e dos outros símbolos no bloco. Aliás, nem se pode identificar claramente onde começa e acaba cada símbolo na sequência binária, pois alguns bits dependem conjuntamente de vários símbolos. No entanto, parece perfeitamente adequado definir o comprimento que cada símbolo ocupa no código, $\bar{\gamma}(k) = \frac{1}{m} \lceil \log_2 N^m \rceil$, ainda que esse comprimento não seja um número inteiro.

5.3 Códigos de Comprimento Inteiro

Na secção anterior descrevemos códigos de comprimento fixo independentemente do símbolo codificado. A evolução natural é associar códigos de comprimentos distintos aos vários símbolos, mais curtos para os símbolos mais frequentes e mais longos para os símbolos mais raros, procurando melhorar o desempenho médio. Chamaremos *código de comprimento inteiro* a qualquer código reversível que represente cada símbolo $k \in A$ por uma *palavra de código* $\gamma(k)$ com comprimento natural $\bar{\gamma}(k) \in \mathbb{N}$. Chamamos *livro de códigos* ao conjunto das palavras de código $\{\gamma(k), \forall k \in A\}$, e chamamos *perfil de comprimentos* à sequência dos comprimentos dessas palavras $\{\bar{\gamma}(k), \forall k \in A\}$.

É sabido que, para permitir a reversibilidade do código, os comprimentos $\bar{\gamma}(k)$ têm que satisfazer a *desigualdade de Kraft-McMillan* [107]:

$$\sum_{k \in A} 2^{-\bar{\gamma}(k)} \leq 1. \tag{5.1}$$

Não basta que os comprimentos das palavras de certo código satisfaçam esta condição para que esse código seja reversível. No entanto, é garantido que existem códigos reversíveis formados por palavras com esses mesmos comprimentos; e sabe-se que é possível construí-los de forma a que nenhuma palavra de código seja um prefixo de outra [107]. Códigos com esta propriedade dizem-se *livres de prefixos*[1] e são convenientes porque permitem uma descodificação instantânea: assim que se recebe o último bit de uma palavra de código, pode emitir-se o símbolo correspondente. Um exemplo ajuda a clarificar a situação. Considerem-se três códigos alternativos para o alfabeto $A = \{a, b, c\}$ definidos na tabela abaixo.

k	$\bar{\gamma}(k)$	Código 1	Código 2	Código 3
a	1	0	0	0
b	2	01	01	10
c	2	10	11	11

Todos têm palavras de comprimentos iguais e satisfazem a condição (5.1) com número de Kraft igual a $2^{-1} + 2^{-2} + 2^{-2} = 1$. No entanto, o Código 1 não é livre de prefixos (0 é um prefixo de 01) e não é reversível, como se demonstra pela sequência 010 que tanto pode ser gerada pela mensagem ac como por ba. O Código 2 não é livre de prefixos, mas é reversível, ainda que a descodificação seja ligeiramente mais complexa e um tanto caprichosa pois pode implicar memória e atraso ilimitados.[2] O Código 3 é livre de prefixos, reversível, e permite uma descodificação simples e instantânea. É claramente preferível aos outros.

As palavras de um código livre de prefixos podem ser interpretadas como os caminhos que vão da raíz de uma árvore binária a cada uma das suas folhas. A existência dessa árvore sugere um algoritmo de descodificação trivial e demonstra, de forma construtiva, a reversibilidade do código e a instantaneidade da descodificação.

Em resumo: dado um certo perfil de comprimentos $\{\bar{\gamma}(k)\}$ que satisfaça a desigualdade (5.1), é sempre possível escolher palavras de código com esses comprimentos de tal forma que o código resultante seja livre de prefixos e, logo, reversível. Uma vez que esta escolha só tem vantagens, não há qualquer interesse prático em construir códigos de comprimento inteiro que não sejam livres de prefixos.

Estabelecidas as condições de existência, o problema da optimização de um código de comprimentos inteiros pode formular-se da seguinte forma: dado um alfabeto A de N símbolos, com probabilidades de ocorrência $\{p_k, \forall k \in A\}$, pretende-se achar o perfil de comprimentos $\{\bar{\gamma}(k), \forall k \in A\}$ que minimiza o débito médio esperado do código

$$R = -\sum_{k \in A} p_k \bar{\gamma}(k),$$

sujeito às condições (5.1) e $\bar{\gamma}(k) \in \mathbb{N}$.

Shannon [138] e Fano desenvolveram, de forma independente, métodos essencialmente equivalentes para construir códigos livres de prefixos com bom desempenho. Os códigos obtidos por esse método têm um débito que excede a entropia da fonte no máximo em 1 bit por símbolo, $R - H < 1$. Esta redundância revela-se pouco significativa quando a entropia é alta, ou seja, quando todos os símbolos são pouco prováveis. Porém, em certas aplicações,

[1] Em Inglês, *prefix-free codes* ou simplesmente, *prefix codes*.

[2] Basta ver o que sucede com uma sequência $01111\ldots100\ldots$, que pode ser descodificada como $acc\ldots ca\ldots$ ou como $bcc\ldots ca\ldots$, dependendo da paridade do número de uns consecutivos, que só é conhecida quando surge o primeiro zero.

por exemplo quando o sinal é bastante previsível, a entropia da fonte pode ser bastante baixa, pelo que tal redundância pode ser inaceitável.

Em todo o caso, o método de Shannon-Fano não garante a melhor solução para o problema de optimização inteira posto atrás. Essa distinção cabe ao método proposto poucos anos mais tarde por Huffman [67]. Em [47] foi demonstrado que a redundância de um código de Huffman é limitada por

$$R - H \leq p + \log_2(2\frac{\log_2 e}{e}) \approx p + 0.086, \tag{5.2}$$

onde p é a probabilidade do símbolo mais frequente.

5.4 Codificação Aritmética

A ideia base da codificação aritmética é que qualquer mensagem pode ser representada—e de forma extremamente eficiente—por um número real do intervalo $[0, 1[$. De facto, esta ideia foi avançada pelo próprio Shannon quando notou que se ordenasse o conjunto de mensagens possíveis pelas suas probabilidades e calculasse as respectivas probabilidades cumulativas, então a probabilidade cumulativa de uma dada mensagem, desde que transmitida como um número fraccionário com precisão suficiente para a distinguir da mensagem seguinte, permitiria a sua descodificação por mera pesquisa da tabela de probabilidades cumulativas [138, Sec. 9]. Apesar do extraordinário valor teórico decorrente da sua optimalidade assimptótica, este método não é prático para a codificação de mensagens longas porque exige o conhecimento e armazenamento prévio de todas as mensagens possíveis e suas probabilidades. E para mensagens curtas ou símbolos, como referimos na secção anterior, não garante o melhor desempenho, sendo ultrapassado pelo método de Huffman. No início da década de 1960, Elias [36] reparou que a ordenação das mensagens é desnecessária; qualquer ordem serve desde que respeitada pelo codificador e descodificador. Mais importante foi a observação de que as probabilidades cumulativas das mensagens podem ser calculadas iterativamente a partir das probabilidades individuais dos símbolos que as compõem. Estes avanços estabeleceram a essência da codificação aritmética actual. Mesmo assim, a codificação de Elias requeria precisão numérica crescente com a dimensão da mensagem e manteve-se como uma idealização teórica. As primeiras implementações práticas de codificação aritmética só surgiriam na última metade da década de 1970, em diversas variantes propostas por vários autores: Rissanen [125], Pasco, Rubin [133], Guazzo [59], Rissanen e Langdon [126], Jones [85]. Uma boa introdução às técnicas modernas de codificação aritmética, incluindo detalhes de implementação, comparação com outras formas de codificação e referências históricas pode encontrar-se em [162].

5.4.1 O Funcionamento da Codificação Aritmética

Um codificador aritmético representa cada mensagem K_n como um intervalo de números reais $[L(K_n), H(K_n)[$ contido em $[0, 1[$. À medida que a mensagem cresce, o intervalo que a representa é reduzido, e o número de bits necessários para o especificar aumenta. Cada símbolo acrescentado à mensagem reduz o intervalo que a representa proporcionalmente à probabilidade esperada do símbolo. Símbolos frequentes reduzem menos o intervalo do que símbolos raros e logo acrescentam menos bits ao código gerado.

Considere-se o seguinte alfabeto de cinco símbolos, juntamente com a respectiva distribuição de probabilidades.

Símbolo	k	a	b	c	d	e
Intervalo	$[l(k), h(k)[$	$[0.0, 0.4[$	$[0.4, 0.5[$	$[0.5, 0.6[$	$[0.6, 0.9[$	$[0.9, 1.0[$
Amplitude	$\delta(k)$	0.4	0.1	0.1	0.3	0.1
Probabilidade	$p(k)$	0.4	0.1	0.1	0.3	0.1

Para cada símbolo k foi reservado um intervalo $[l(k), h(k)[$ cuja amplitude $\delta(k) = h(k) - l(k)$ foi igualada à probabilidade do símbolo, $\delta(k) = p(k)$. O conjunto desses intervalos é uma partição de $[0, 1[$. Cada intervalo representa uma mensagem composta por uma única ocorrência do respectivo símbolo; por exemplo $[0.6, 0.9[$ representa a mensagem d. Outra interpretação: o intervalo representa o conjunto de todas as mensagens possíveis começadas pelo símbolo respectivo, i.e. $[0.6, 0.9[$ representa todas as mensagens da forma $dk_2k_3\ldots$. Quando se acrescenta mais um símbolo à mensagem, por exemplo e, subdivide-se o intervalo da mensagem em subintervalos, proporcionalmente à partição estabelecida, e escolhe-se o subintervalo correspondente ao novo símbolo, neste caso o último décimo do intervalo anterior, ou seja $[0.87, 0.90[$, que passa a representar a nova mensagem de. Este é o algoritmo de Elias idealizado [133], uma forma pura de codificação aritmética que descrevemos abaixo.

Algoritmo 5.1 (Codificação Aritmética Pura) *Os dados são: uma mensagem* $K_n = k_1k_2\ldots k_n$ *e um modelo estatístico representado pelos intervalos* $[l(k), h(k)[$ *e amplitudes* $\delta(k)$. *O resultado é um número real* X *ou um intervalo* $[L_n, H_n[$.

1. *Inicializar* $L_0 = 0, H_0 = 1, \Delta_0 = 1$.
2. *Para cada* i *desde* $i = 1$ *até* $i = n$, *fazer*
   (a) $L_i = L_{i-1} + \Delta_{i-1}l(k_i)$,
   (b) $H_i = L_{i-1} + \Delta_{i-1}h(k_i)$,
   (c) $\Delta_i = \Delta_{i-1}\delta(k_i)$.
3. *Escolher um valor* $X \in [L_n, H_n[$.

Nesta descrição, L_i e H_i são os limites do intervalo que representa a mensagem K_i, e $\Delta_i = H_i - L_i$ é a amplitude correspondente. (Pareceu-nos mais elucidativo manter os três registos no algoritmo, ainda que se pudesse eliminar facilmente qualquer deles, dada a sua interdependência.)

Assim, para codificar a mensagem *decada*, começa-se com o intervalo $[0, 1[$ que representa a mensagem vazia (ou todas as mensagens possíveis) e, à medida que se processa cada símbolo, vai-se restringindo progressivamente o intervalo, como se indica a seguir.

n	k_n	K_n	$[L_n,$	$H_n[$	Δ_n
0	.	.	[0,	1[	1
1	d	d	[0.6,	0.9[	0.3
2	e	de	[0.87,	0.90[	0.03
3	c	dec	[0.885,	0.888[	0.003
4	a	$deca$	[0.8850,	0.8862[	0.0012
5	d	$decad$	[0.88572,	0.88608[	0.00036
6	a	$decada$	[0.885720,	0.885864[	0.000144
	Símbolo	Mensagem	Intervalo		Amplitude

Deste modo, a mensagem *decada* é representada pelo intervalo $[0.885720, 0.885864[$ ou, na verdade, por qualquer número X que lhe pertença, digamos $X = 0.8858$. Neste exemplo usámos um alfabeto de saída decimal para simplificar a exposição do algoritmo, mas na prática usar-se-ia normalmente aritmética e representações binárias.

Vejamos agora como se processa a descodificação. Ao receber o número X, o descodificador reconhece facilmente que $X \in [l(d), h(d)[= [0.6, 0.9[$, pelo que identifica o primeiro símbolo como d. Agora, emulando o processo de codificação, subdivide este intervalo em cinco, e pesquisa a nova partição para achar o subintervalo que inclui X, neste caso $X \in [L(de), H(de)[= [0.87, 0.90[$, logo o segundo símbolo é identificado como e. Continuando desta forma, identificam-se os restantes símbolos da mensagem. Formalizando, o algoritmo de descodificação idealizado é:

Algoritmo 5.2 (Descodificação Aritmética Pura) *Os dados são: um número real X e um modelo estatístico representado pelos intervalos $[l(k), h(k)[$ e amplitudes $\delta(k)$. O resultado é uma sequência de símbolos descodificados $k_1 k_2 \ldots k_n = K_n$.*

1. *Inicializar* $L_0 = 0, H_0 = 1, \Delta_0 = 1$
2. *Para cada* i *desde* $i = 1$ *até* $i = n$, *fazer*
   (a) *Descobrir o símbolo* k_i *tal que* $L_{i-1} + \Delta_{i-1} l(k_i) \leq X < L_{i-1} + \Delta_{i-1} h(k_i)$,
   (b) $L_i = L_{i-1} + \Delta_{i-1} l(k_i)$,
   (c) $H_i = L_{i-1} + \Delta_{i-1} h(k_i)$,
   (d) $\Delta_i = \Delta_{i-1} \delta(k_i)$.

5.4.2 Desempenho

O comprimento de código ocupado por uma mensagem K_n é dado directamente pelo logaritmo da amplitude do respectivo intervalo,

$$\bar{\gamma}(K_n) = -\log \Delta(K_n).$$

Cada símbolo do alfabeto tem também um comprimento bem definido,

$$\bar{\gamma}(k) = -\log \delta(k),$$

que é perfeitamente coerente com a definição anterior pois

$$\bar{\gamma}(K_n) = -\log \Delta(K_n) = -\log \prod_{i=1}^{n} \delta(k_i) = -\sum_{i=1}^{n} \log \delta(k_i) = -\sum_{i=1}^{n} \bar{\gamma}(k_i). \tag{5.3}$$

Em geral, estas expressões produzem comprimentos não inteiros, mas isso é perfeitamente adequado porque a *concatenação* de palavras de código usada nos códigos de comprimento inteiro é substituída, na codificação aritmética, por uma operação de *mudança de escala e adição* que evita o desperdício de informação fraccionária à medida que a mensagem cresce. Esta operação pode ser vista como uma *concatenação generalizada* que permite juntar palavras de código que se sobrepõem parcialmente [126]. Só no final da mensagem é que haverá alguma redundância, devido à necessidade de transmitir um número inteiro de símbolos do código. Em todo o caso, a redundância não superará 1 bit (ou 1 dígito do código usado) para a mensagem completa de comprimento n, ou seja $1/n$ bits por símbolo, como demonstrado

trivialmente por Shannon [138] para o seu código perfeitamente idêntico neste aspecto. (Isto é válido quando o descodificador conhece o comprimento da mensagem. Implementações reais requerem alguma informação extra como veremos na próxima secção.)

Relembrando o exemplo da mensagem $K_6 = \mathit{decada}$, vemos que o comprimento do código respectivo é de $-\log_{10}(\Delta_6) \approx 3.84$ algarismos decimais ou $-\log_2(\Delta_6) \approx 12.76$ bits. Se esta for a mensagem completa a transmitir, então os 4 dígitos decimais 8858, ou os 13 dígitos binários 1110001011000, serão suficientes para a distinguir de qualquer outra mensagem da mesma dimensão. Em contrapartida, se houver mais símbolos a juntar à mensagem antes da transmissão, então vão-se adicionando os seus comprimentos ao comprimento fraccionário da mensagem parcial, como decorre da equação (5.3) posta agora na forma

$$\bar{\gamma}(K_{n+1}) = \bar{\gamma}(K_n) + \bar{\gamma}(k_{n+1}).$$

Como é óbvio, a optimização do código obtém-se simplesmente pela especificação de intervalos com amplitudes iguais às probabilidades dos símbolos respectivos. Nestas condições, a amplitude Δ_n representa o produto das probabilidades dos primeiros n símbolos da mensagem, ou seja, a probabilidade da própria mensagem K_n e, portanto, o seu logaritmo é também a entropia da mensagem.

5.4.3 Questões de Implementação

Os algoritmos apresentados atrás são impraticáveis por várias razões. Os problemas que apresentam e as suas diversas soluções são tratados extensamente na literatura já citada. Aqui abordaremos apenas algumas questões mais importantes para a compreensão das implementações actuais.

O *problema da precisão* é porventura o obstáculo mais grave à realizabilidade da codificação aritmética. À medida que a mensagem cresce, Δ_n toma valores cada vez mais próximos de zero, os registos L_n e H_n têm de ser mantidos com um número cada vez maior de dígitos, e a aritmética torna-se proporcionalmente mais complexa. Uma solução surge do reconhecimento que os limites do intervalo $[L_n, H_n[$ se aproximam cada vez mais e, portanto, os seus primeiros dígitos tornam-se frequentemente iguais. Quando isso sucede, podem transmitir-se esses dígitos e deslocar-se os restantes igual número de posições para a esquerda, recuperando assim precisão à direita dos registos. Aplicando esta correcção ao exemplo da codificação decimal da mensagem *decada*, vemos que ao atingir $K_3 = \mathit{dec}$ poderíamos transmitir os dois dígitos 88 e ajustar os registos para $[L_3, H_3[= [0.5, 0.8[$ e $\Delta_3 = 0.3$. Ao atingir $K_6 = \mathit{decada}$, poderíamos transmitir mais um dígito, fazendo os ajustes correspondentes. Esta técnica tem a vantagem adicional de proporcionar uma *transmissão incremental*, eliminando a outra grande deficiência do código de Elias que era a necessidade de codificar a mensagem completa antes de transmitir. Há um detalhe que dificulta o processo: os limites do intervalo podem aproximar-se indefinidamente sem que os seus primeiros dígitos se igualem, por exemplo $[0.4999, 0.5001[$. Também para isto existe uma solução engenhosa que permite a progressão da codificação sem perda de precisão, embora com a introdução de algum atraso ocasional na transmissão e descodificação [162].

O *problema da terminação* consiste em determinar onde parar a descodificação. Na verdade, o número $X = 0.8858$ representa a mensagem *decada* mas também qualquer dos seus prefixos $\mathit{decad}, \mathit{deca}, \ldots, d$ ou mensagens mais longas como *decadac* ou *decadacc*. Rubin [133] aponta duas soluções para este *problema da terminação*: transmitir previamente o comprimento da mensagem; ou incluir no alfabeto um símbolo especial indicando *fim-de-mensagem*,

que é a solução escolhida em [162]. Aproveitamos para discutir uma terceira solução aplicável em certas condições. Se o comprimento da mensagem só puder tomar certos valores, por exemplo múltiplos de um certo tamanho fixo M, então basta descodificar a sequência de código até ao fim e desprezar quaisquer símbolos posteriores ao último bloco de M. Naturalmente, é necessário que se conheça a dimensão da sequência codificada com certa precisão e que M seja suficientemente grande para permitir uma descodificação inequívoca.

Devemos ainda notar que este problema e as suas soluções não são exclusivos da codificação aritmética, embora assumam nesta alguns aspectos peculiares devido à possibilidade de um único bit conter vários símbolos.

A solução do problema da precisão permite o uso de registos de comprimento finito e implementações baseadas em aritmética inteira. Existem várias formas de traduzir o algoritmo fundamental para aritmética inteira, o que deu origem a múltiplas implementações [126, 133, 59, 85]. Muitos autores procuraram reduzir a complexidade computacional por eliminação das operações de divisão e multiplicação e/ou aceleração das operações de pesquisa de tabelas [108]. Nalguns casos, os codificadores resultantes são assumidamente subóptimos mas ganham em velocidade de processamento [124, 66]. Existem implementações particularmente rápidas para o caso de mensagens binárias [90, 91, 69, 88, 161], que podem sempre ser aplicadas a outros alfabetos decompostos como números binários (*bit-slice coding*). As várias implementações são geralmente mutuamente incompatíveis e impõem diferentes tipos de restrições aos parâmetros de codificação, por exemplo ao nível da resolução com que se pode registar o intervalo (e probabilidade) de cada símbolo.

5.5 Contexto, Modelos e Adaptação

Nas secções anteriores supusemos implicitamente uma distribuição estática das probabilidades dos símbolos. Porém, é frequente que a probabilidade de um símbolo ocorrer dependa, por exemplo, dos símbolos que o precederam na mensagem (o seu *contexto*); ou dependa de alguma forma do *estado* da fonte do sinal. Nestas situações, é sempre vantajoso usar as probabilidades condicionais em vez das probabilidades marginais estáticas. Para isso é necessário dispor de modelos estatísticos capazes de estimar a distribuição de probabilidades em função do estado da fonte em cada instante; e adaptar o sistema de codificação para tirar partido dessas probabilidades melhoradas.

O modelo estatístico dinâmico pode ser algo tão simples como uma tabela do número de ocorrências de cada símbolo numa certa janela temporal: uma forma de histograma deslizante. Em alternativa podem manter-se várias tabelas fixas representativas das distribuições de probabilidades em diversos estados do sistema; e comutar entre elas em função de alguma estimativa do estado. Uma variação poderosa deste esquema usa uma ponderação das várias tabelas conseguindo assim um comutação suave entre estados que eleva o número efectivo de distribuições modeladas. Neste documento não nos debruçaremos mais sobre as variadas formas e técnicas de modelação estatística. Uma boa referência neste domínio é [12].

A adaptação de um codificador aritmético é muito simples do ponto de vista conceptual: basta alterar os limites da partição que descreve o alfabeto, ou seja as probabilidades cumulativas dos seus símbolos. Isto pode ser feito quantas vezes se quiser e por qualquer processo desde que possa ser reproduzido sincronamente no descodificador. É usual que a adaptação só dependa do passado da sequência codificada para evitar a necessidade de transmitir parâmetros de adaptação. O único impacto negativo da adaptação na codificação aritmética é o custo

computacional acrescido pelas operações de actualização e busca das tabelas. Certas estruturas de dados especializadas permitem reduzir bastante a complexidade destas operações [40].

Em [47] foi proposto um algoritmo para a codificação de Huffman adaptativa. Trata-se de um processo mais complexo conceptualmente que o da codificação aritmética e menos versátil porque impõe uma estrutura mais restritiva ao modelo estatístico. O maior defeito, no entanto, é que a redundância de um código de Huffman tem tendência a piorar com a maior previsibilidade dos símbolos (ver equação (5.2)), o que afinal contraria parcialmente o propósito da adaptação.

5.6 Discussão

Das várias formas de codificação apresentadas neste capítulo, a codificação aritmética destaca-se por um conjunto de factores. Nesta secção procuramos resumir e comparar as características mais importantes destes codificadores.

Eficiência. Dado um modelo estatístico da fonte, a codificação aritmética permite atingir, essencialmente, o desempenho óptimo, com uma redundância inferior a um bit por mensagem. À medida que cresce o comprimento n da mensagem, o débito médio aproxima-se da entropia da fonte à razão de $1/n$. É um facto que o método de Huffman poderia ainda superar este desempenho desde que aplicado ao alfabeto das mensagens completas, e é neste sentido que se deve entender a sua optimalidade. Mesmo assim, a eventual vantagem de Huffman neste caso hipotético não supera uma fracção de bit por mensagem, o que é irrisório em mensagens de dimensão razoável. Na prática, porém, o método só é aplicável a alfabetos de símbolos (ou sequências curtas de símbolos) e neste caso, a redundância, ainda que uma pequena fracção de bit, está presente em cada palavra de código e não decresce com o prolongamento da mensagem. Certos aspectos de implementação, como o problema da terminação e principalmente a resolução finita dos modelos de probabilidade, impedem que o desempenho dos codificadores aritméticos reais atinja precisamente o limite teórico óptimo. No entanto, mesmo implementações aproximadas, de baixa precisão, têm frequentemente eficiências acima dos 98% [69].

Processamento incremental. Em códigos de comprimento fixo e de comprimento inteiro é usual recorrer-se à extensão do alfabeto por constituição de blocos de símbolos, com o objectivo de minimizar a redundância por símbolo. Em muitos casos, o empacotamento em blocos é um mecanismo puramente artificial sem qualquer relação com a estrutura natural das mensagens. Cria mesmo uma assimetria de tratamento dos símbolos consoante estejam dentro de um mesmo bloco ou em blocos diferentes: qualquer dependência estatística entre símbolos pode ser aproveitada no primeiro caso mas não no segundo. Apesar desta deselegância, o processamento em blocos foi durante muito tempo considerado inevitável para conseguir desempenhos melhores. Na codificação aritmética não há qualquer motivação para a criação de blocos devido à eficiência absoluta do mecanismo de concatenação generalizada. Evita-se a complexidade acrescida dos alfabetos extendidos, os atrasos de empacotamento, e o processamento desenvolve-se símbolo-a-símbolo, continuamente, aproveitando sem restrições quaisquer dependências estatísticas.

Adaptabilidade. A codificação aritmética promove uma segregação clara entre o modelo estatístico da fonte e o algoritmo de codificação propriamente dito. O modelo pode

facilmente ser tornado tão sofisticado e dinâmico quanto se queira. Desde que o modelo seja capaz de fornecer estimativas precisas da distribuição de probabilidades, sabemos que o algoritmo produzirá o melhor código possível. Esta separação de funções possibilita uma enorme versatilidade de modelação e não tem paralelo noutras técnicas de codificação.

Complexidade computacional. A implementação directa de codificadores e descodificadores aritméticos tem uma complexidade tipicamente superior à de outros tipos de codificação como códigos de comprimento inteiro, ou códigos de dicionário como os de Ziv-Lempel [167, 168], muito usados em compressão de texto e dados. Certas variantes especializadas, particularmente para alfabetos binários, eliminam as operações mais complexas e permitem implementações rápidas e até em hardware dedicado [88].

Robustez a erros de transmissão. Um erro de transmissão num código aritmético pode ter consequências catastróficas, com a completa perda de sincronismo entre codificador e descodificador. Esta não é uma característica específica deste tipo de códigos; qualquer código de comprimento variável está sujeito a este fenómeno, embora se tenha observado uma certa capacidade de auto-sincronização nalguns casos [117, 106, 109, 148, 152, 166]. A inexistência de redundância no código aritmético provavelmente agrava o problema pois qualquer código recebido pode representar uma mensagem perfeitamente válida. Curiosamente, esta sensibilidade extrema aos erros de transmissão pode ser usada como uma forma poderosa de detecção de erros. Basta reservar propositadamente algum espaço na partição que não é atribuído a qualquer símbolo do alfabeto. Se, durante a descodificação, algum dos registos entrar num destes intervalos proibidos, então certamente houve algum erro na transmissão. A probabilidade de detecção do erro parece crescer para 1 à medida que o descodificador progride na sequência [18]. Esta capacidade pode ser integrada em sistemas de transmissão com protocolos de pedido automático de repetição [37, 25].

Generalidade da Codificação Aritmética. Um aspecto teórico muito interessante da codificação aritmética é que permite emular muitos outros tipos de código. Dado um certo código de comprimento fixo ou um certo código de comprimento inteiro livre de prefixos, é relativamente fácil construir um modelo estático de probabilidades que, fornecido a um codificador aritmético, produz precisamente as mesmas sequências codificadas, bit-por-bit. Mesmo sistemas com empacotamento de símbolos podem ser emulados por modelos dinâmicos, combinados com codificação aritmética símbolo-a-símbolo. Num documento em preparação serão apresentados exemplos e demonstrações destes resultados [129]. Também outros tipos de códigos não referidos neste capítulo foram identificados como casos particulares de codificadores aritméticos com modelos específicos. É o caso dos *run-length codes* de Golomb [54], considerados uma especialização do código descrito em [91]; e de uma subclasse importante dos codificadores de dicionário, incluindo a maioria das inúmeras variantes do método de Ziv-Lempel [13]. Esta generalidade da codificação aritmética fornece um forma canónica através da qual se pode compreender, comparar e, eventualmente, implementar os vários tipos de código.

Capítulo 6

Psicoacústica e Modelos Perceptuais

Em qualquer sistema de compressão com perdas interessa obviamente que o sinal codificado seja, na perspectiva do receptor, tão próximo quanto possível do original. O erro quadrático médio ou outras medidas de distorção poderão ser critérios adequados para avaliar a capacidade discriminativa de certos receptores simples, mas o sistema auditivo humano é um receptor muito complexo e a avaliação do seu desempenho requer medidas bastante mais elaboradas. Assim, um modelo matemático da percepção auditiva é uma ferramenta fundamental não só para a avaliação objectiva da qualidade de um sistema de codificação de áudio, mas mais ainda para o seu projecto e até implementação.

Para compreender os *modelos* da percepção auditiva, é preciso estudar primeiro em que consiste o *som*, como é processado no *ouvido*, e que *sensações* provoca.

6.1 Noções de Acústica

O som é uma vibração mecânica que se pode propagar na forma de ondas acústicas através de meios sólidos, líquidos ou gasosos. No ar (e noutros gases), as ondas acústicas consistem em variações de pressão atmosférica. Trata-se de variações ínfimas relativamente à pressão atmosférica média (cerca de 10^5 Pa ao nível do mar), por isso e porque o ouvido não é capaz de detectar variações lentas da pressão atmosférica como as que ocorrem quando subimos uma montanha, a *pressão* p que usamos para descrever um som é geralmente uma medida da diferença de pressão em relação à média e não uma medida absoluta. Uma onda acústica pode então ser caracterizada por uma função $p(x,t)$ que indica a pressão em cada ponto do espaço x e instante de tempo t. Mais frequentemente estamos interessados numa medida média das flutuações de pressão num certo ponto e por isso medimos o seu desvio padrão ou valor rms[1] p_{rms}. (Isto é tão comum que muitas vezes escrevemos p em vez de p_{rms}.) Em certas circunstâncias, o ouvido humano é capaz de detectar sons com pressão rms na ordem de 10^{-5} Pa, enquanto sons com pressões na ordem de 10 a 100 Pa já são dolorosos e perigosos para a audição.

O *nível de pressão sonora* (*Sound Pressure Level* ou SPL) é uma medida da pressão sonora

[1] Do Inglês *root-mean-square*.

dada numa escala logarítmica (em dB) segundo a fórmula:

$$SPL = 20 \log_{10} \frac{p_{\text{rms}}}{p_0},$$

onde a pressão de referência, $p_0 = 2 \times 10^{-5}$ Pa, corresponde aproximadamente ao limiar de audição de um tom de 1 kHz. Traduzida nesta escala, a gama de pressões audíveis estende-se aproximadamente de 0 a 120 dB_{SPL}.

Na sua propagação, as ondas acústicas transportam energia porque as flutuações de pressão (forças) provocam movimentos (deslocações) das partículas de ar. Na verdade estes "movimentos" são modulações sistemáticas dos rápidos movimentos aleatórios das moléculas do ar, resultando numa flutuação da velocidade média u proporcional à flutuação da pressão $u = p/z$, sendo z a impedância acústica constante em circunstâncias normais. À medida que o som se expande no espaço, a sua energia distribui-se por toda a área da frente de onda. Assim, em cada ponto do espaço pode definir-se uma *intensidade* acústica que é uma medida da densidade (superficial) de energia que atravessa esse ponto por unidade de tempo, e mede-se em W/m^2. A intensidade I é dada pelo produto da pressão p pela velocidade u, por isso é também proporcional ao quadrado da pressão, $I = pu = p^2/z$.

O *nível de intensidade* de um som é uma medida logarítmica da intensidade do som, definida por

$$SIL = 10 \log_{10} \frac{I_{\text{rms}}}{I_0},$$

onde a intensidade de referência, $I_0 = 10^{-12}$ W/m^2, é tal que $SPL \approx SIL$ quando o som é uma onda progressiva de frente plana. Para distribuições sonoras mais gerais, os níveis de pressão e de intensidade podem diferir em vários dB. No entanto, essa diferença não varia com o nível do som desde que se conserve a forma do campo radiante que depende apenas da geometria da fonte e da sala.

Dois sons que atinjam um ponto x com pressões $p_1(x,t)$ e $p_2(x,t)$, interferem produzindo uma flutuação equivalente $p(x,t) = p_1(x,t) + p_2(x,t)$. Se forem sons independentes, o som resultante tem uma pressão (e intensidade) rms que é a soma das pressões (e intensidades) rms de cada um.

Os sons são transformados em sinais eléctricos por transdutores—microfones—que convertem variações de pressão em variações de tensão eléctrica, que por sua vez são transformados em sinais digitais por conversores analógico-digitais. Os valores do sinal digital obtido, análogo do sinal acústico, são então proporcionais às variações de pressão sonora.

Uma boa introdução à Acústica encontra-se em [64]. Para um tratamento mais completo, ver [115].

6.2 O Sistema Auditivo

O sistema auditivo tem uma região periférica responsável pela recolha, pré-processamento e transdução dos sons em sinais eléctricos nas células nervosas; e uma região central responsável pelo processamento neuronal da informação periférica, conduzindo à sua percepção. O sistema central é composto por muitos milhares de neurónios que vão processando e propagando a informação auditiva sucessivamente de camada em camada, desde as células ciliadas internas até ao córtex primário. Este processamento é muito complexo e ainda relativamente mal conhecido.

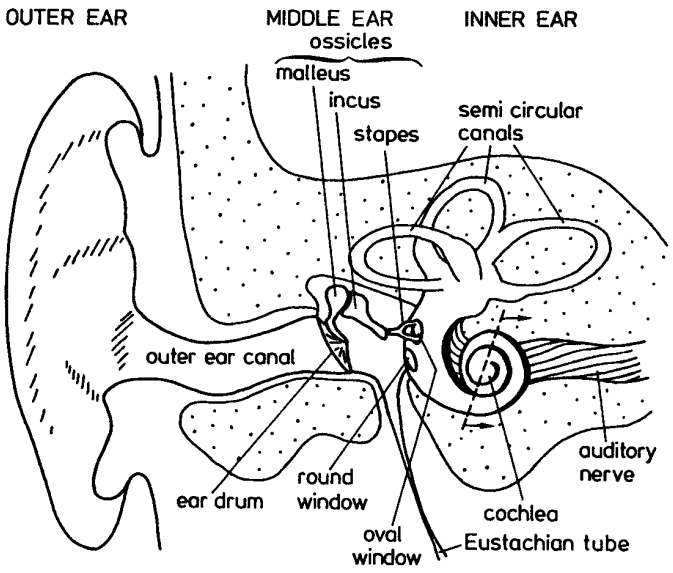


Figura 6.1: Representação esquemática do sistema auditivo periférico. (Reproduzido de [169, Fig. 3.1]).

O sistema auditivo periférico, representado na Figura 6.1, pode ser dividido em três partes: ouvido externo, ouvido médio e ouvido interno. O pré-processamento realizado nestes orgãos tem uma influência preponderante em grande parte dos fenómenos da audição que nos interessam, pelo que importa descrevê-los um pouco melhor. Referências: [113, Sec. 1.5] [169, Cap. 3] [41, 3, 99].

6.2.1 Ouvido Externo e Médio

O ouvido externo é composto pela orelha e pelo canal auditivo. A orelha introduz modificações no espectro do som recebido, particularmente nas altas frequências. Esta "coloração" espectral depende da direcção de incidência do som e é um dos factores que contribui para a nossa capacidade de localização dos sons [63]. O canal auditivo é um tubo com cerca de 2 cm de comprimento que conduz o som recolhido pela orelha até ao tímpano, fazendo-o vibrar. Esta cavidade apresenta uma ressonância que é responsável pela maior sensibilidade do ouvido a sinais com frequências próximas de 4 kHz.

As vibrações do tímpano são transmitidas a outra membrana na base da cóclea, a janela oval, através dos três ossículos que constituem o ouvido médio. Estes ossos minúsculos funcionam como alavancas que, em conjunto com a diferença de áreas entre o tímpano e a janela oval, produzem um efeito de adaptação de impedâncias que permite comunicar de forma eficiente as vibrações acústicas aos fluidos no interior da cóclea.

6.2.2 Ouvido Interno e Membrana Basilar

A cóclea é o órgão do ouvido interno responsável pela transdução dos estímulos acústicos em impulsos nervosos. É formado por três ductos enrolados em forma de caracol, preenchidos por líquidos. Depois de atravessarem o ouvido externo e o ouvido médio, as vibrações acústicas são transmitidas à janela oval na base da cóclea e produzem um deslocamento dos líquidos

no seu interior. Os líquidos em movimento produzem uma deformação mecânica numa membrana que divide os ductos da cóclea, a membrana basilar. Células sensíveis (células ciliadas) dispostas ao longo desta membrana convertem essa deformação oscilatória em impulsos nervosos que são transmitidos pelo nervo auditivo associado. A membrana basilar tem cerca de 32 mm de comprimento, é estreita e rígida junto à janela oval (na base da cóclea) e mais larga e flexível junto ao helicotrema (vértice da cóclea).

Sob um estímulo acústico, a membrana basilar sustenta uma onda progressiva que se propaga desde a base até ao vértice da cóclea. A velocidade de propagação da onda vai diminuindo ao longo do percurso e depende da frequência do estímulo. Assim, a membrana basilar é um meio dispersivo no qual as frequências altas se propagam apenas a distâncias muito curtas antes de serem fortemente atenuadas, enquanto as frequências baixas atingem distâncias maiores [163, Sec. II.B]. Para um tom puro de certa frequência, a envolvente da onda progressiva atinge um valor máximo numa posição determinada da membrana basilar: mais perto da base se a frequência for alta, ou mais perto do vértice se a frequência for baixa. Visto de outra forma, cada ponto da membrana basilar responde melhor a estímulos de uma certa frequência, a *frequência característica* desse ponto. A cóclea funciona, portanto, como um analisador espectral hidro-mecânico.

6.2.3 Transdução Neuronal

O orgão de Corti, que se situa sobre a membrana basilar, tem a função de transformar as oscilações mecânicas desta em impulsos eléctricos para o sistema nervoso. É formado por muitos milhares de células de diferentes tipos, entre as quais dois grupos de células sensoriais chamadas células ciliadas internas (CCI) e células ciliadas externas (CCE). As CCI estão prioritariamente ligadas a fibras nervosas ascendentes e crê-se serem responsáveis pela maior parte da informação transmitida ao sistema nervoso central. As CCE estão fortemente ligadas essencialmente a fibras descendentes vindas do cérebro. Pensa-se que interagem com as CCI num processo activo de realimentação não-linear que amplifica as oscilações mais fracas e permite assim extender grandemente a gama dinâmica do sistema. (As emissões otoacústicas podem ser um subproduto deste processo de realimentação.)

6.3 Psicoacústica

A Psicoacústica é um domínio da Psicologia que estuda as sensações e os efeitos provocados pelos estímulos acústicos no sistema auditivo. Por avaliação a resposta global do sistema auditivo a determinados estímulos, expõe fenómenos da percepção que revelam os seus limites de desempenho. Deste modo, complementa e extende conhecimentos fisiológicos parciais, ajudando a formar uma imagem mais completa da percepção auditiva.

6.3.1 Sonoridade, Limiar de Audição e Limiar de Dor

Sons diferentes com o mesmo nível de pressão sonora não são em geral percebidos como sendo igualmente "fortes". A sensação subjectiva de *sonoridade*[2] de um som depende não só da sua intensidade mas também da sua frequência e qualidade. A Figura 6.2 apresenta contornos de igual sonoridade de tons puros (sinusóides) em condições de campo aberto.

[2] *Loudness*, na língua inglesa.

A curva inferior corresponde ao *limiar absoluto de audição* que indica a intensidade mínima audível em cada frequência. Verifica-se uma sensibilidade máxima do ouvido a tons perto dos 4 kHz, e uma relativa insensibilidade a frequências baixas ou muito altas. Este comportamento resulta essencialmente da característica passa-banda do ouvido externo e médio. A curva superior marca o *limiar de dor*, acima do qual os sons começam a provocar dor e podem provocar danos permanentes no sistema auditivo (perda de sensibilidade localizada ou mesmo surdez completa).

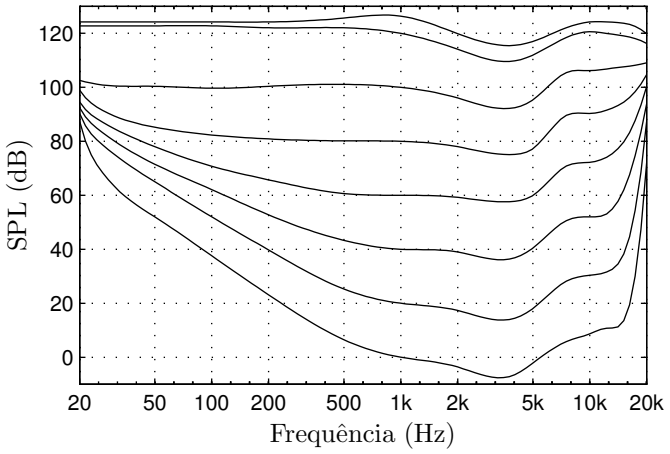


Figura 6.2: Curvas de igual sonoridade de tons puros em condições de campo aberto.

6.3.2 Selectividade Espectral: Filtros Auditivos e Bandas Críticas

Vimos atrás que cada ponto da membrana basilar está *sintonizado* numa certa frequência característica e só responde bem a componentes espectrais de frequências próximas dessa. Como cada ponto tem uma frequência característica diferente, a cóclea pode ser considerada um banco de filtros passa-banda com bandas sobrepostas, os chamados *filtros auditivos*. Esta interpretação foi sugerida em 1940 por Fletcher, que fez as primeiras tentativas de medição das larguras de banda destes filtros, considerados filtros rectangulares ideais para simplificar, mediante estudos de mascaramento de um tom centrado numa banda de ruído mascarante de densidade espectral constante mas largura progressivamente crescente. O limiar de mascaramento aumentava com a largura (e intensidade) da banda até certo valor, chamado *largura de banda crítica*, e depois mantinha-se constante. Estas observações conduziram a um modelo de mascaramento simples, mas baseado numa série de pressupostos que hoje se consideram pouco precisos [110, Sec. II].

Desde então, inúmeros investigadores têm estudado e aperfeiçoado este modelo, que determina a *resolução* ou *selectividade espectral* do ouvido e influencia muitos outros aspectos como a percepção da sonoridade ou o mascaramento. Apesar das falhas do modelo inicial proposto por Fletcher, os conceitos de banco de filtros auditivos e bandas críticas mantêm-se actuais e largamente aceites, sendo suportados por uma série de resultados experimentais, por exemplo:

- O mascaramento de um tom por uma banda de ruído é máximo quando o tom está no

centro da banda de ruído. O mascaramento aumenta com a largura da banda de ruído até ser igual à banda crítica, mas a partir daí mantém-se constante.

- O mascaramento de uma banda de ruído por dois tons nos seus extremos mantém-se constante enquanto a largura é inferior a uma banda crítica; para larguras maiores o efeito diminui bastante.

- Um som complexo, contido numa banda subcrítica, tem a mesma sonoridade que um tom puro com igual potência centrado na mesma banda. Quando a largura de banda do som complexo ultrapassa os limites da banda crítica, a sua sonoridade aumenta acima da do tom com a mesma potência.

- A sensibilidade a diferenças de fase de componentes dum som complexo é maior se estiverem na mesma banda crítica. A identificação de harmónicos e a agradabilidade de um conjunto de tons também se relacionam com as bandas críticas.

Filtros auditivos

A forma do filtros auditivos foi estimada em várias experiências de mascaramento usando tipicamente ruído mascarante rejeita-banda [113, Sec. 3.3]. As formas estimadas são descritas por funções da família das *exponenciais arredondadas* (*rounded exponentials* ou *roex*). Existem diversas variantes destas funções com um ou mais parâmetros que determinam a sua largura de banda, maior ou menor assimetria, e dependência com a intensidade [110, Sec. III.B].

A largura de banda dos filtros auditivos depende da frequência característica considerada. Acima dos 500 Hz, a largura dos filtros é praticamente proporcional à sua frequência central. Descendo abaixo dos 500 Hz, a largura diminui mais lentamente. Esta largura pode ser especificada de diferentes formas, produzindo números diferentes mas que representam efectivamente a mesma resposta. Actualmente, a forma mais usual é a *largura de banda rectangular equivalente* (*equivalent rectangular bandwidth* ou ERB), que se define como a largura de uma banda rectangular ideal com o mesmo ganho máximo e a mesma potência inscrita. A *largura de banda crítica* (*critical bandwidth* ou CB) referida atrás, é outra forma essencialmente equivalente de representar a mesma informação e, de facto, apresenta valores proporcionais às ERBs, excepto para frequências abaixo dos 500 Hz, onde tende para um valor constante de 100 Hz. Esta discrepância deve-se às diferentes configurações experimentais usadas no passado para a determinação das CBs e à escassez de medidas tiradas nessa gama de frequências. As larguras dos filtros auditivos, especificadas como ERBs ou CBs, encontram-se tabeladas para diferentes frequências características, ou podem ser aproximadas através de fórmulas empíricas bastante precisas [154].

Escalas de Frequência Auditivas

O alargamento das CBs ou ERBs em função da frequência inspirou a criação de novas escalas de frequências mais adaptadas às características perceptuais. Assim, define-se a *escala Bark* ou *critical band rate* (z), de tal forma que uma diferença de uma unidade nessa escala ($\Delta z =$ 1 Bark) corresponda a uma CB ao longo de toda a gama da escala linear de frequências (f). De modo análogo pode definir-se uma escala baseada na ERB. Traunmüller compara e relaciona diferentes escalas e fórmulas empíricas de conversão [154, 155]. A escala Bark tem uma forte relação com a anatomia do ouvido interno. Um intervalo de 1 Bark corresponde

a um comprimento fixo na membrana basilar (aproximadamente 1.3 mm) e a um número constante de cerca de 1200 fibras nervosas, independentemente da posição ou frequência.

Excitação

Em [171, Cap. IX] a *excitação* é definida como uma variável intermédia hipotética, com uma grandeza análoga à intensidade, que de alguma forma traduz a actividade neural proveniente de cada ponto da membrana basilar e/ou a velocidade de deflexão desse mesmo ponto. Outros autores preferem defini-la como o padrão formado pelas saídas dos *filtros auditivos* ao longo da membrana basilar [114]. O padrão de excitação provocado por um estímulo acústico é geralmente representado numa escala Bark e expresso em dBs referidos a um nível base arbitrário.

6.3.3 Mascaramento

Um aspecto importante da audição, com especial interesse para a codificação perceptual, é o fenómeno de mascaramento. Diz-se que há *mascaramento* quando a percepção de um som fraco (de baixa energia) é dificultada ou mesmo eliminada pela presença de um som forte. Define-se *limiar de mascaramento* de um som fraco (sinal de teste ou mascarado) devido a um som forte (sinal mascarante) como o nível de intensidade abaixo do qual o sinal de teste se torna inaudível na presença do mascarante. Se a intensidade do sinal de teste superar o limiar, o mascaramento é apenas parcial: o sinal é audível mas com sonoridade inferior à que teria se o sinal mascarante fosse retirado. Por outras palavras: a audição de um som qualquer provoca uma elevação do limiar de mascaramento—que em repouso coincide com o limiar de audição—reduzindo desse modo a sensibilidade do ouvido a sons mais fracos.

Das diversas experiências realizadas para avaliar o efeito de mascaramento, pode resumir-se um conjunto de resultados interessantes [65]:

- O mascaramento entre dois tons simultâneos é tanto mais eficaz quanto mais próximas forem as suas frequências. (Esta regra é violada quando as frequências se aproximam tanto que se percebam batimentos.)

- Um tom mascara tons de frequência mais alta melhor do que tons de frequência mais baixa.

- Quanto maior a intensidade de um tom, mais larga é a gama de frequências que mascara.

- O mascaramento provocado por bandas estreitas (inferiores a uma banda crítica) de ruído apresenta características semelhantes ao mascaramento provocado por tons. No entanto, é mais eficaz: o excesso de intensidade do ruído em relação à intensidade do tom mascarado é inferior ao necessário para um tom mascarar ruído. Este fenómeno é referido na literatura como a *assimetria do mascaramento* [60].

- Um som forte consegue mascarar sons curtos que ocorram pouco depois do primeiro som terminar (*pós-mascaramento*). Também consegue mascarar sons ocorridos pouco antes do seu início (*pré-mascaramento).*

- Sons procedentes de fontes espacialmente próximas mascaram-se mais do que sons afastados no espaço.

Em síntese, o mascaramento depende da relação de intensidades dos dois sons e da sua proximidade no tempo, na frequência e no espaço. Nas subsecções seguintes, pormenoriza-se um pouco melhor os efeitos das relações espectrais e temporais dos sons no mascaramento. Não focaremos as diferenças de nível de mascaramento relacionadas com a localização das fontes sonoras nem outros fenómenos de percepção binauricular, visto que neste trabalho só se consideram sinais monofónicos, destinados aos dois ouvidos simultaneamente.

Efeitos Espectrais

O fenómeno de mascaramento depende fortemente da composição espectral dos sinais envolvidos. A Figura 6.3 mostra o limiar de mascaramento de um tom puro na presença de uma banda subcrítica de ruído. O mascaramento é máximo quando o tom está no centro da banda de ruído mascarante. À diferença, medida em dB, entre os níveis dos dois sinais nesta situação chama-se *índice de mascaramento* e, para ruído de banda crítica a mascarar um tom, apresenta valores entre 2 dB para frequências baixas e 5–6 dB para frequências altas [169, Sec. 4.1.2]. Para tons de frequência inferior à do ruído, o limiar diminui abruptamente com um declive de cerca de 27 dB/Bark. Para tons de frequência superior, o declive decrescente da curva depende do nível do sinal mascarante, sendo bastante acentuado para níveis baixos e progressivamente mais suave para níveis altos.[3] O padrão de mascaramento praticamente não depende da frequência do sinal mascarante desde que se use a escala perceptual graduada em Bark.

Em experiências com tons a mascarar bandas de ruído ou a mascarar outros tons, observaram-se padrões de mascaramento idênticos. No entanto, nesses casos, o índice de mascaramento é bastante maior (*assimetria do mascaramento*) e apresenta maior variação em função da frequência.

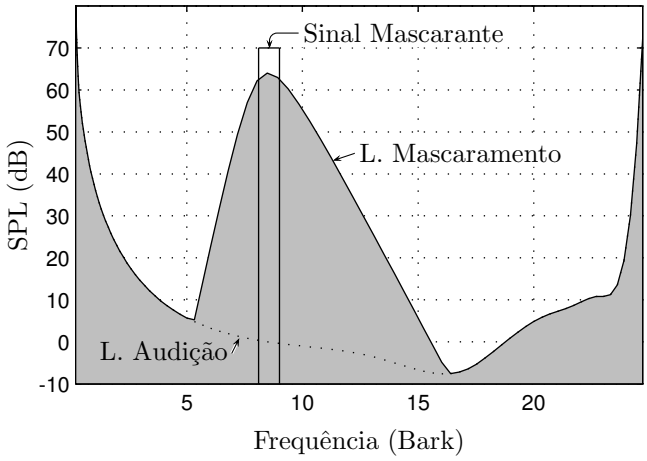


Figura 6.3: Representação esquemática do fenómeno de mascaramento no domínio do frequência. O sinal mascarante é uma banda estreita de ruído e o sinal de teste é um tom puro apresentado em simultâneo.

O limiar de mascaramento de um sinal de teste de banda estreita revela o padrão de excitação que o sinal mascarante induz na membrana basilar. Quando o mascarante é também

[3]Este fenómeno não linear é o chamado *upward spread of masking*.

um sinal de banda estreita, a forma do limiar de mascaramento coincide com a função de espraiamento.

Efeitos Temporais

Um som forte de duração finita não mascara apenas sons que ocorram simultaneamente, mas também sons curtos que lhe sucedam ou que o precedam por alguns mili-segundos. O efeito de mascaramento de sons posteriores é o chamado *pós-mascaramento*; o mascaramento de sons anteriores chama-se *pré-mascaramento*. A Figura 6.4 ilustra a dependência temporal do fenómeno de mascaramento. A figura representa esquematicamente o limiar de mascaramento de um sinal de teste muito curto (de poucos mili-segundos) na presença de um sinal mascarante. O limiar de mascaramento mantém-se praticamente constante enquanto os sinais são simultâneos. Quando o mascarante termina, o limiar decai em direcção ao limiar de audição, o que demora algumas dezenas ou centenas de mili-segundos. O pré-mascaramento só é significativo num período de poucos mili-segundos mas parece variar bastante de ouvinte para ouvinte [139]. Existe ainda um efeito de *overshoot*, uma elevação de alguns dB no limiar durante os primeiros 10 ms do sinal mascarante [169, Sec. 4.4.1]. Pensa-se que o pós-mascaramento resulta de uma persistência da representação do som no sistema auditivo, reduzindo temporariamente a sensibilidade para processar outros sons. A origem do pré-mascaramento não é conhecida mas poderá residir em níveis mais elevados do sistema auditivo, onde o processamento de estímulos fortes interfere e subjuga o processamento de estímulos anteriores mais fracos. A extensão no tempo dos fenómenos de pré- e pós-mascaramento depende da estrutura espectral dos sinais e da intensidade do mascarante. A duração do mascarante também tem alguma influência na duração do pós-mascaramento.

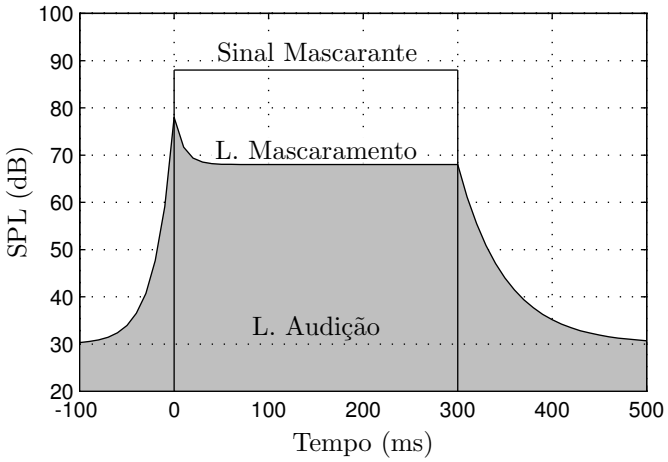


Figura 6.4: Representação esquemática do fenómeno de mascaramento no domínio do tempo. O sinal mascarante é um *burst* de 300 ms de um tom puro; o sinal de teste é um impulso de ruído de banda estreita (1 Bark) centrado na frequência do tom.

É pertinente notar que muitos codificadores perceptuais não exploram explicitamente os fenómenos de mascaramento no tempo. Nalguns casos, porém, o conhecimento das limitações temporais do mascaramento é usado para decidir e determinar o instante de comutação de janelas [17], ou para implementar um outro mecanismo de protecção contra os pré-ecos [41].

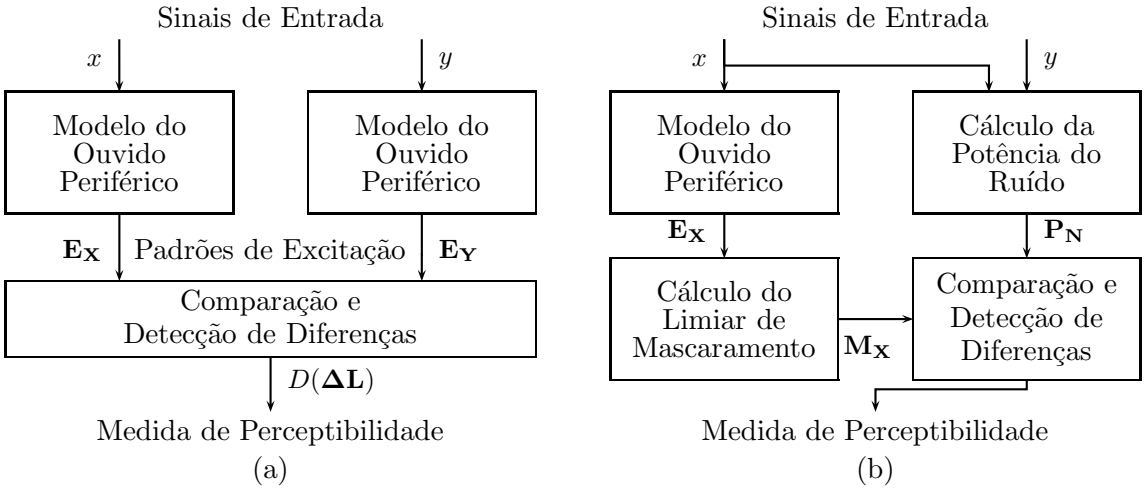


Figura 6.5: Modelos perceptuais como medidas de distorção. Abordagem usando: (a) representações internas, (b) limiar de mascaramento.

Aditividade do Mascaramento

Quando o mascarante é um sinal complexo, o limiar de mascaramento supera geralmente o que seria de esperar da sobreposição linear dos limiares de mascaramento provocados por cada uma das suas componentes isoladamente. Certas medições experimentais deste *excesso de mascaramento* podem ser previstas por modelos de aditividade num domínio comprimido [97, 98, 68]. No entanto, Moore apresenta uma série argumentos que põem em causa a aplicabilidade destes modelos em situações gerais. Refere mesmo condições experimentais em que não ocorre excesso de mascaramento, ainda que previsto pelos modelos [110, Sec. VI].

6.4 Modelos Perceptuais

Interessa-nos estudar modelos perceptuais que permitam medir a distorção introduzida por codificadores de áudio, e que possam ser usados para a sua avaliação, projecto e implementação. Na literatura encontram-se duas abordagens distintas para este fim, representadas esquematicamente na Figura 6.5. A primeira abordagem consiste em determinar *representações internas* (*padrões de excitação*) dos sinais original e reconstruído, e em seguida compará-las para determinar a perceptibilidade das diferenças. O primeiro passo modela o sistema auditivo periférico, enquanto o segundo modela o sistema central. Modelos que seguem esta abordagem inspiram-se, em grande medida, no modelo de padrões de excitação de Zwicker (ZEPM) [171, 169], e podem ser encontrados, mais recentemente, em diversas medidas objectivas de qualidade perceptual, como por exemplo: PAQM, PERCEVAL, POM, PIR, DIX, e PEAQ. (Os vários modelos estudados ao longo deste capítulo estão resumidos na Tabela 6.1, juntamente com referências e abreviaturas que usaremos com bastante frequência.)

Na segunda abordagem, o padrão de excitação do sinal original é usado para estimar o *limiar de mascaramento*, que é depois comparado com a distribuição no tempo e na frequência da energia do ruído—definido pela diferença entre os sinais original e processado. Exemplos desta abordagem são a medida de distorção denominada NMR e vários modelos psicoacústicos

Tabela 6.1: Modelos perceptuais usados em medidas de distorção e codificadores de áudio. Indica-se o tipo: representação interna (RI) ou limiar de mascaramento (LM).

Abrev.	Tipo	Descrição	Referências
ZEPM	RI	Modelo de padrões de excitação de Zwicker	[171, 169]
PAQM	RI	*Perceptual Audio Quality Measure*	[9][8, Sec. 3]
PERCEVAL	RI	*Perceptual Evaluation*	[120][8, Sec. 5]
POM	RI	*Perceptual Objective Measurement*[27]	
PIR	RI	*Peripheral Internal Representation*	[147]
DIX	RI	*Disturbance Index*	[150]
PEAQ	RI+LM	*Perceptual Evaluation of Audio Quality*	[151, 75]
NMR	LM	*Noise-to-Mask Ratio*	[8, Sec. 4]
PXFM	LM	Modelo do codificador PXFM	[83]
MPEG/MP2	LM	Modelo Psicoacústico 2 do MPEG	[77]
BAPAC	LM	Modelo Psicoacústico do BAPAC	(Cap. 7)[131]

descritos na literatura de codificação perceptual no domínio da frequência, essencialmente os baseados no PXFM como o MPEG/MP2, BAPAC e muitos outros.

O conceito de limiar de mascaramento é conveniente na implementação de codificadores de áudio, já que pode ser interpretado como um "tecto" que condiciona o perfil do ruído a introduzir. Isto justifica, talvez, a preponderância desta abordagem no projecto e implementação de codificadores de áudio. No entanto, este conceito, ou pelo menos a sua implementação simplista, enferma de vários problemas, como foi observado por Veldhuis [160] e, mais recentemente, por outros autores [31].

6.4.1 Modelo do Ouvido Periférico

Ambas as abordagens de modelação perceptual começam por calcular os padrões de excitação usando um modelo do ouvido periférico. A Figura 6.6 representa esquematicamente as principais etapas envolvidas neste processamento, que descrevemos a seguir.

Análise Espectral e Rectificação

De uma forma geral todos os modelos perceptuais propostos para a finalidade que nos interessa começam por obter uma representação espectral dos sinais. Na verdade trata-se de representações tempo-frequência: uma sucessão de espectros de segmentos extraídos dos sinais através de uma janela deslizante de curta duração.

Naturalmente, a transformada discreta de Fourier (DFT) tem sido usada frequentemente para este fim. Tipicamente, é aplicada após uma janela de Hann [62] com sobreposição de 50% entre tramas sucessivas, para melhorar a selectividade espectral. Também há exemplos de utilização de transformadas com sobreposição [120] e bancos de filtros com resolução não uniforme [7][76]. Em [145], Sporer e Brandenburg estabelecem requisitos de resolução temporal e espectral a que devem satisfazer os bancos de filtros usados em medidas de qualidade perceptual e analisam alguns casos concretos à luz dessas condições.

Os coeficientes da transformada ou banco de filtros são rectificados geralmente com uma lei quadrática para determinar a potência em cada canal analisado. Quando a análise consiste em

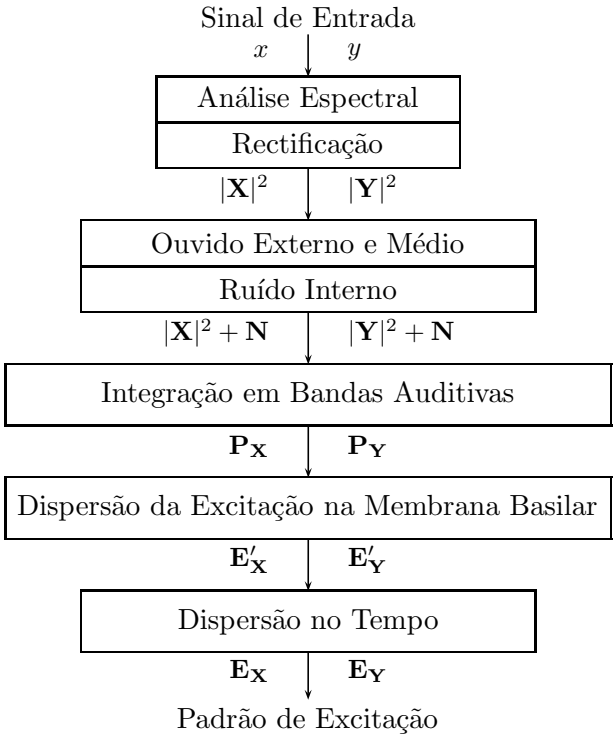


Figura 6.6: Modelo do Ouvido Periférico.

bancos de filtros ou transformadas reais, é aplicada uma filtragem passa-baixo para suavizar as estimativas.

Função de Transferência do Ouvido Externo e Médio

Tanto o ouvido externo como o ouvido médio têm um comportamento considerado essencialmente linear na gama de intensidades usadas em codificação e podem consequentemente ser descritos por uma operação de filtragem. Sendo linear, a filtragem deveria preceder a rectificação e até a análise espectral, de forma a preservar a resposta de fase e evitar artefactos. Na prática, não se modela a resposta de fase, pelo que esta operação é geralmente implementada directamente no domínio da frequência por ponderação do espectro de potência do sinal com a resposta espectral de potência do filtro.

Os efeitos de filtragem direccional introduzidos pela orelha, tronco e cabeça também poderiam ser incluídos neste passo. Bastaria filtrar sinais provenientes de diferentes fontes sonoras usando as funções de transferência correspondentes às respectivas direcções de incidência. Este processamento diferenciado deveria contribuir para a modelação do fenómeno de desmascaramento binaural.

Em vários modelos, a resposta espectral de potência do ouvido externo e médio é modelada pela expressão,

$$W(F)/dB = -3.64\beta F^{-0.8} + 6.5e^{-0.6(F-3.3)^2} - 10^{-3}F^4, \tag{6.1}$$

onde $F = f/kHz$, e $\beta \in [0,1]$ é um parâmetro que varia de modelo para modelo. Esta expressão (com $\beta = 1$) é simétrica da proposta por Terhardt para modelar o limiar absoluto

de audição [149].

Ruído Interno e/ou Limiar Absoluto de Audição

Nesta fase, pode-se adicionar ao espectro de potência um termo que representa um *ruído interno*, associado aos batimentos cardíacos e outra actividade muscular [169, p. 166], que contribui para prejudicar ainda mais a sensibilidade a baixas frequências. O ruído interno é modelado por uma expressão relacionada com o primeiro termo de (6.1):

$$N(F)/dB = 3.64(1-\beta)F^{-0.8}.$$

Este passo é por vezes aplicado depois do integração em bandas auditivas, o que é equivalente se forem corrigidos os termos adicionados.[4] Algumas implementações omitem simplesmente este passo, compensando-o com um reforço da atenuação do ouvido externo e médio (fazendo $\beta = 1$).

Em modelos destinados à estimação de limiares de mascaramento é mais frequente que tanto o ruído interno como o ouvido externo e médio sejam omitidos, e substituídos no fim do processo por uma aplicação correctiva do limiar absoluto de audição ao limiar de mascaramento.

Integração em Bandas Auditivas

Esta operação é um passo intermédio para o cálculo da excitação. Consiste na integração do espectro de potência através de uma "janela" rectangular deslizante com uma largura fixa na escala Bark. Isto equivale a uma transformação do espectro de potência para o domínio Bark. Na maioria dos modelos, as janelas de integração são adjacentes e não sobrepostas, o que produz uma representação com um número reduzido de coeficientes igualmente espaçados na escala auditiva.

Uma fórmula empírica que permite converter uma frequência de Hertz para Bark é (BAPAC):

$$\frac{z}{\text{Bark}} = 13 \arctan \frac{f}{1.3\text{kHz}} + 3.5 \arctan^2 \frac{f}{7.5\text{kHz}}. \tag{6.2}$$

Existem outras fórmulas empíricas para esta relação, algumas mais simples e precisas [154, 155].

Dispersão da Excitação na Membrana Basilar

O espectro em bandas auditivas é então convoluído com uma *função de espraiamento* para se obter o padrão de excitação instantâneo. A função de espraiamento $S(z_1, z_2)$, que indica a quantidade relativa de excitação num ponto de frequência z_2 provocada por um estímulo de frequência z_1, só depende da diferença de frequências em Bark, $\Delta z = z_2 - z_1$, ou seja $S(z_1, z_2) = S(\Delta z)$. Na literatura encontram-se várias fórmulas empíricas para esta função, que diferem essencialmente em dois aspectos: a maior ou menor suavidade da curva, e a capacidade de variar o declive descendente em função da intensidade. Uma é descrita analiticamente pela expressão:

$$S(\Delta z)/\text{dB} = 15.81 + 7.5(\Delta z + 0.474) - 17.5\sqrt{1 + (\Delta z + 0.474)^2} + d(\Delta z), \tag{6.3}$$

[4] No modelo baseado num banco de filtros da medida PEAQ, não é aparente qualquer correcção [75, Eq. (13)].

onde $d(\Delta z) = \min\{0,\, 8(\Delta z - 0.5)(\Delta z - 2.5)\}$ é um termo correctivo adicionado à fórmula original de [136], que introduz uma pequena depressão no início da rampa descendente da curva, modelando um fenómeno que ocorre com mascarantes de grande intensidade. Um inconveniente desta função é que apresenta declives fixos, independentes da intensidade do mascarante. Outra expressão muito utilizada é a função de espraiamento de [149]:

$$S(\Delta z)/\mathrm{dB} = \begin{cases} 27\Delta z & \text{se } \Delta z \leq 0 \\ (-24 - \frac{230\ \mathrm{Hz}}{f(z)} + 0.2LP)\Delta z & \text{se } \Delta z > 0 \end{cases} . \tag{6.4}$$

que tem uma forma triangular com um declive ascendente fixo, e um declive descendente dependente do nível do excitante, $LP = 10\log_{10}(P/P_0)$. Também inclui uma correcção ligeira para frequências baixas.

A convolução do excitante P com a função S produz a excitação instantânea,

$$E'(z) = \left(\sum_{\nu} \left(P(\nu) 10^{S(z-\nu)} \right)^{\alpha} \right)^{1/\alpha} , \tag{6.5}$$

onde o expoente α possibilita a sobreposição num domínio comprimido ($\alpha < 1$), de forma a poder modelar a aditividade não linear dos padrões. Alguns modelos optam por $\alpha = 1$, considerando a aditividade linear.

Dispersão Temporal

Um modelo simples (e algo grosseiro) para o fenómeno de pós-mascaramento foi apresentado em [9]. Consiste num *espalhamento temporal* da energia de cada componente espectral segundo uma curva de decaímento exponencial. É facilmente implementável por um filtro passa-baixo recursivo com um único atraso e ganho $a = e^{-T/\tau}$, onde T é o período entre amostras consecutivas da mesma componente espectral e τ é a constante de tempo do decaimento exponencial. Esta constante de tempo depende da frequência central f da componente segundo uma lei que pode ser aproximada por:

$$\tau(f) = \tau_{\min} + \frac{100\ \mathrm{Hz}}{f}(\tau_{100} - \tau_{\min}), \tag{6.6}$$

onde τ_{100} e $\tau_{\min}$ são as constantes de tempo à frequência de 100 Hz e às altas frequências, respectivamente. Note-se que o tempo de decaímento do pós-mascaramento modelado excede $\tau(f)$ em cerca de metade da resolução temporal das componentes, $\tau_{\mathrm{post}} = \tau(f) + T/2$. Por exemplo, no modelo baseado na FFT da medida PEAQ, estes parâmetros têm os valores: $T \approx 21$ ms, $\tau_{100} = 30$ ms e $\tau_{\min} = 8$ ms.

Dada a curta duração do fenómeno de pré-mascaramento, o tempo de subida do filtro deve ser bastante menor que o de decaímento, por isso o filtro deve reduzir a constante de tempo para sinais crescentes. Na prática faz-se normalmente $\tau_s = 0$ porque a resolução temporal da transformada já introduz tempo de subida superior ao do pré-mascaramento. Ou seja, o filtro funciona na verdade como um detector de envolvente.

6.4.2 Cálculo do Limiar de Mascaramento

Conhecido o padrão de excitação de um sinal mascarante, pode-se estimar o limiar de mascaramento de um ruído de banda estreita por simples *dedução do índice de mascaramento* àquele

padrão. Num codificador, porém, introduz-se ruído em múltiplas bandas, que são combinadas no padrão de excitação por convolução com a função de espalhamento. Assim, componentes de ruído que individualmente eram mascaradas, podem tornar-se audíveis quando combinadas. Para determinar o limiar neste caso de múltiplos alvos de mascaramento, seria necessário fazer uma *desconvolução* do limiar calculado atrás, como reconheceu Johnston [83]. Na prática, dadas as dificuldades que levanta, esta operação é substituída por uma *renormalização*, que serve apenas para compensar o ganho introduzido pelo espalhamento da excitação. Esta simplificação é um dos principais problemas apontados a esta abordagem [160].

A renormalização consiste simplesmente na aplicação estática do ganho inverso ao padrão de excitação. Esta operação pode ser incluída mesmo em modelos de representação interna, pois não afecta as diferenças relativas entre os padrões de excitação.

Abaixo descrevemos modelos empíricos para avaliação do índice de mascaramento em função da frequência, tanto no caso de tons a mascarar ruído como no caso inverso. Também se referem algumas medidas de tonalidade que podem ser usadas para determinar o índice de mascaramento de um sinal mascarante que não seja nem um tom puro nem uma banda de ruído.

Tons a Mascarar Ruído

Um primeiro modelo para o índice de mascaramento de ruído por tons, abreviadamente TMN (*Tone Masking Noise*), foi dado em [136]:

$$TMN(z)/\mathrm{dB} = 15.5 + z.$$

As tabelas de TMN disponíveis na recomendação do MPEG para o Modelo Psicoacústico 2 parecem seguir esta lei, com uma adaptação nas frequências mais baixas. No entanto, como esta fórmula foi construída sobre dados experimentais obtidos para frequências baixas (até 3 ou 4 kHz), poderá estar errada para frequências mais altas. Na verdade, foi reconhecido em [41] que esta expressão levava a uma sobrecodificação das altas frequências em detrimento das baixas e foi proposta uma fórmula alternativa para corrigir esse defeito:

$$TMN(z)/\mathrm{dB} = 19.5 - \frac{18}{26}z.$$

Em [131], propusemos ainda outra expressão, baseando-nos em dados experimentais de [146]:

$$TMN(z)/\mathrm{dB} = \begin{cases} 16\frac{6-z}{6} + 28\frac{z}{6} & \text{se } 0 \le z < 6 \\ 28 & \text{se } 6 \le z < 16 \\ 28\frac{20-z}{20-16} + 20\frac{z-16}{20-16} & \text{se } 16 \le z < 20 \\ 20 & \text{se } 20 \le z \end{cases} . \tag{6.7}$$

Esta expressão é mais conservadora que a de Schroeder até à frequência de 12.5 Bark (1.8 kHz). Para frequências mais altas, o índice não cresce mais; até diminui ligeiramente a partir de 20 Bark (7 kHz). Não há portanto consenso quanto à curva exacta do índice TMN em função da frequência.

Ruído a Mascarar Tons

O índice de ruído mascarando tons (NMT) é bastante mais baixo, e apresenta menor variação ao longo da frequência. Um modelo aproximado, considerado razoável, para este índice é:

$$NMT(z)/\mathrm{dB} = 5.5.$$

Ferreira usou um modelo ligeiramente diferente [41]:

$$NMT(z)/\text{dB} = 6.56 - \frac{3.06}{26}z.$$

Em codificação perceptual, só interessa considerar situações de mascaramento de ruído de quantização, não de mascaramento de tons. Contudo, na falta de dados mais específicos para essa circunstância, usa-se o índice NMT para avaliar o mascaramento de ruído de quantização por sinais não tonais.

Avaliação de Tonalidade

As diferenças significativas entre o poder mascarante de tons e de bandas de ruído levanta o problema da determinação da *tonalidade* dos sinais a processar por um codificador perceptual. Uma primeira solução para este problema foi proposta em [83] e aplicada no codificador PXFM. Baseia-se no cálculo da medida de planura espectral (SFM) avaliada a partir do espectro de potência de cada bloco do sinal, estimado por uma DFT. Essa medida é então convertida num coeficiente de tonalidade, α, de tal forma que $\alpha = 0$ quando o espectro é absolutamente plano ($SFM = 0$ dB), indicando a natureza "ruidosa" do sinal; e $\alpha = 1$ quando $SFM < -60$ dB, considerado um indicador de "tonalidade pura". O índice de mascaramento final é calculado por interpolação entre TMN e NMT segundo:

$$MI(z) = \alpha\, TMN(z) + (1 - \alpha) NMT(z). \tag{6.8}$$

Um inconveniente deste modelo é que o coeficiente de tonalidade é uma medida global para todo o espectro, não discriminando regiões tonais e regiões não tonais que ocorrem simultaneamente em frequências diferentes em determinados sinais. Apercebendo-se disso, Brandenburg e Johnston propuseram um outro método de avaliar a tonalidade, baseado numa medida de coerência entre valores sucessivos de cada componente espectral [22]. O princípio subjacente é que uma componente "tonal" terá uma evolução temporal coerente (previsível), enquanto uma componente de ruído será imprevisível. A medida de coerência é calculada em cada componente espectral pela distância euclidiana entre o valor registado no bloco actual e o valor estimado por um preditor simples baseado na amplitude e na fase medidas nos dois blocos anteriores. Por uma relação logarítmica, é derivado o coeficiente de tonalidade $\alpha(z)$, dependente da frequência, que permite finalmente calcular o índice de mascaramento pela expressão 6.8. Esta medida de tonalidade é aplicada igualmente no Modelo Psicoacústico 2, recomendado em [77].

Em [42] é proposto um terceiro método que permite também uma avaliação local da tonalidade, mas com a vantagem de se basear em coeficientes de uma MDCT, directamente disponíveis na maioria dos codificadores de áudio modernos.

6.4.3 Detecção de Diferenças

Segundo os modelos de representação interna, a detectabilidade e/ou audibilidade das diferenças entre dois sinais X e Y é determinada por comparação dos padrões de excitação que estes produzem, E_X e E_Y. Há diferentes formas de fazer essa comparação, mas todas dependem da diferença entre os níveis de excitação,

$$\Delta L = 10 \log_{10} \frac{E_X}{E_Y}.$$

Por exemplo, no modelo ZEPM [171, Cap. XI] é indicado o critério de detecção

$$\max_k |\Delta L_k| > 1 \text{ dB}, \tag{6.9}$$

ou seja, segundo este critério, as diferenças serão perceptíveis se os padrões diferirem nalgum ponto em mais que 1 dB.

Em [23] é proposto um critério alternativo baseado num princípio estocástico de detecção, no qual a probabilidade global de detecção p é calculada em função das probabilidades de detecção p_k de um conjunto de detectores elementares considerados posicionados regularmente ao longo da membrana basilar. Considera-se que os detectores são independentes e que basta um ser activado para haver detecção global, por isso a probabilidade de não detecção—que só ocorre se nenhum dos detectores for activado—é dada pela expressão

$$(1 - p) = \prod_k (1 - p_k). \tag{6.10}$$

A probabilidade de detecção individual p_k é dada por uma função psicométrica de forma sigmóide dependente apenas de ΔL_k no ponto k. Segundo [23], para uma densidade de um detector por Bark,

$$p_k = 1 - 0.5^{10^{0.2\Delta L_k}} \tag{6.11}$$

permite prever com boa exactidão os resultados de várias experiências de detecção de sinais complexos. Na medida PEAQ, é apresentada uma fórmula diferente [75, Equações 76–77] que pode ser expressa equivalentemente por:

$$p_k = 1 - 0.5^{(\Delta L_k/s)^b},$$

onde o expoente $b = 4$ se $\Delta L_k > 0$ ou $b = 6$ se $\Delta L_k \leq 0$; e s é o *passo de detecção efectivo*, correspondente ao valor de ΔL que produz 50% de probabilidade de detecção. O valor de s depende do nível de excitação dos sinais segundo uma expressão que aproxima a curva de JNLD[5] medida em [169, Sec. 7.1.2].

O princípio de detecção estocástica (6.10) é mais apelativo que o critério fixo (6.9), já que a capacidade discriminativa do sistema auditivo claramente não é determinística—o mesmo conjunto de estímulos pode, em testes diferentes, gerar respostas contraditórias. Além disso, este modelo de detecção é mais geral visto que substituindo a função psicométrica por um degrau unitário com limiar em 1 dB se obtém o critério de ZEPM. Por outro lado, a expressão do modelo como um longo produto de probabilidades não é conveniente para implementações práticas. Também a presunção de independência deve ser tida em conta, estabelecendo-se um limite à densidade dos detectores.

Uma terceira alternativa é usar uma medida de *detectabilidade* expressa por uma soma de contribuições dos vários detectores elementares,

$$d = \sum_k d_k, \tag{6.12}$$

onde as contribuições d_k são—como as probabilidades p_k acima—dadas por alguma função que cresce com ΔL_k. Isto foi proposto por Stuart [147], que justifica a opção por achar mais plausível que a detecção se baseie numa acumulação de actividade neural do que num

[5] *Just Noticeable Level Differences*.

produto extenso de probabilidades. Além disso, esta formulação apresenta uma semelhança com o cálculo da sonoridade, também esta modelada como um somatório de contribuições elementares—as sonoridades específicas—dependentes do valor da excitação em cada ponto. Na verdade, Moore&al. levam esta semelhança ao extremo num modelo [114] que integra a detectabilidade como uma extensão suave da sonoridade. Esta é uma propriedade interessante quando se introduz ruído claramente audível.

Outra característica interessante, que curiosamente nenhum destes autores fez notar, é que esta formulação pode facilmente abarcar o critério de Zwicker e o critério de detecção estocástica referidos atrás. Para o demonstrar, basta definir as detectabilidades como logaritmos das probabilidades de não detecção,

$$d = -\log_2(1-p)$$

e

$$d_k = -\log_2(1-p_k),$$

o que torna equivalentes as equações (6.10) e (6.12). (Para o efeito da demonstração matemática, a escolha do sinal e da base dos logaritmos é arbitrária, mas não foi obviamente inocente a opção que fizemos acima devido à sua semelhança com uma medida de informação.) Nesta formulação, as funções psicométricas assumem formas bastante mais simples. A função (6.11), por exemplo, reduz-se a

$$d_k = 10^{0.2\Delta L_k}.$$

6.5 Discussão

Apresentámos, de uma forma integrada, uma série de modelos da percepção auditiva usados em medidas de qualidade e codificadores de áudio. Procurámos identificar a estrutura comum que os descreve e os detalhes que os distinguem. Parece haver alguns aspectos mais polémicos nos vários modelos descritos:

- Muitos dos modelos, particularmente os usados em codificadores e medidas de qualidade mais simples, aplicam uma função de espraiamento independente do nível do sinal (6.3), enquanto noutros o declive descendente da função depende do nível (6.4). A primeira solução é mais simples, mas só a segunda permite prever com precisão o fenómeno de *upward spread of masking*. Além disso, esta dependência também parece contribuir, juntamente com uma regra de detecção multicanal, para a melhoria da sensibilidade do ouvido a variações de nível para tons fortes, fenómeno conhecido como *near-miss to Weber's law*. Por estas razões, o segundo modelo reúne maior consenso actualmente [112].

- A sobreposição dos padrões de excitação num domínio comprimido, segundo uma lei de potências (6.5), foi inspirada pelos modelos de aditividade do mascaramento. No entanto, como referimos atrás, a correcção desses modelos em situações complexas foi posta em causa por Moore [110, Sec. VI], que sugere a intervenção de efeitos temporais para explicar as experiências que revelavam excesso de mascaramento. Deste modo, não parece justificável a adopção deste modelo mais complexo. Na proposta da medida PAQM [9], o expoente de compressão α foi um dos três parâmetros sujeitos a optimização para melhorar a correlação da medida com resultados de testes subjectivos. Foi determinado o valor $\alpha = 0.4$, mas a função-custo usada nessa optimização não

parece aumentar muito para valores mais altos. Além disso, numa adaptação desta medida para sinais de voz [10], os mesmos autores chegaram a valores muito distintos dos parâmetros livres, o que sugere uma inadequação do método de optimização utilizado.

- A assimetria do mascaramento entre tons e ruído, referida em 6.3.3, é um fenómeno relevante para o cálculo do limiar de mascaramento e foi alvo de múltiplos modelos (ver Sec. 6.4.2). Assim, a avaliação da tonalidade dos estímulos surgiu como forma de determinar o índice de mascaramento mais apropriado a cada situação. Um trabalho experimental recente [60] mostra que a assimetria se revela, numa banda crítica, sempre que as larguras de banda dos sinais mascarante e de teste são diferentes. Assim, parece aceitável que um modelo que determine padrões de excitação com uma resolução mais fina que a CB, permita prever estes efeitos, sem necessidade de uma avaliação explícita da tonalidade.

- Na medida PEAQ, o critério de detecção incorpora a uma função psicométrica p_k que não depende apenas de diferença de níveis de excitação ΔL_k mas também do seu nível absoluto. Esta dependência é introduzida por um parâmetro s que modela a variação de JNLD com o nível do estímulo. Isto parece-nos muito questionável, porque essa curva representa resultados de testes psicoacústicos que aferem a capacidade do sistema auditivo completo detectar diferenças de intensidade de estímulos tonais externos e não a resposta dos hipotéticos detectores elementares que o compõem relativamente a diferenças de uma grandeza interna relacionada não-linearmente com os estímulos externos. Na verdade, num modelo como ZEPM, um limiar de detecção constante ($s = 1$ dB nesse caso), independente do nível, revela-se perfeitamente capaz de prever a própria curva de JNDL. Isto deve-se a outros mecanismos incluídos no modelo, como a adição de ruído interno, responsável pela fraca sensibilidade nos níveis baixos; e a dependência com o nível da função de espalhamento da excitação, responsável pela melhoria da sensibilidade aos níveis mais altos. Uma vez que estes mecanismos também estão incluídos na medida PEAQ, não encontramos justificação para a integração concomitante da dependência do limiar de detecção s com o nível da excitação.

- Os vários mecanismos de detecção de diferenças usados nos modelos de representação interna que revimos parecem, à primeira vista, bastante distintos. No entanto, concluímos que se trata de diferentes formulações matemáticas ou particularizações de uma expressão mais geral. Neste quadro comum, é fácil comparar ou verificar a equivalência dos vários mecanismos. A expressão (6.12) é simples, mais fácil de tratar que (6.10), e igualmente poderosa. Adicionalmente, a sua interpretação como extensão da sonoridade é muito conveniente e promissora.

Capítulo 7

Codificação Perceptual Retroadaptativa

Neste capítulo apresentamos uma estrutura de codificação perceptual que recorre à quantização retroadaptativa das componentes espectrais do sinal de áudio. Defendemos que esta estratégia de grande simplicidade algorítmica tem potencialidades para constituir uma alternativa viável às tecnologias de codificação de áudio dominantes.

A Figura 7.1 representa a estrutura de um sistema de codificação perceptual no domínio da frequência com quantização retroadaptativa.

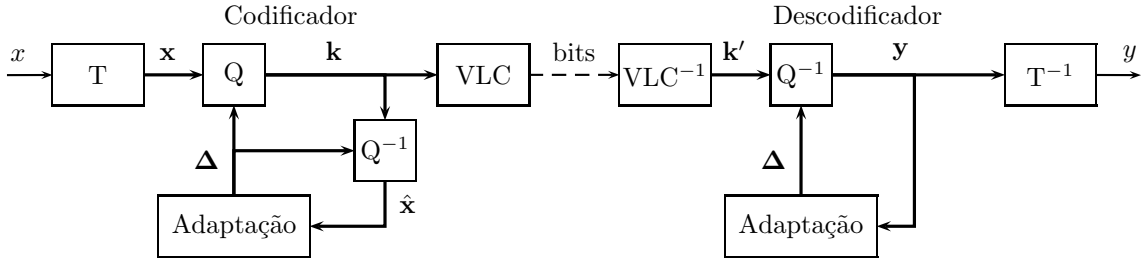


Figura 7.1: Sistema de codificação perceptual retroadaptativo (PAC-AQB). As linhas grossas denotam sinais multi-resolução.

O bloco de análise (T) faz uma decomposição do sinal em bandas. As amostras de cada banda são quantizadas adaptativamente (Q) e os símbolos produzidos são codificados numa sequência binária usando um código reversível de comprimento variável (VLC) para reduzir a redundância. O algoritmo de adaptação recupera as amostras acabadas de quantizar $\hat{\mathbf{x}}$ e, usando um modelo perceptual que avalia continuamente a perceptibilidade do ruído de quantização, determina os novos parâmetros $\boldsymbol{\Delta}$ para ajustar a quantização das amostras futuras. No receptor, a sequência binária é descodificada, os símbolos são desquantizados e as amostras recuperadas são passadas ao banco de síntese que produz uma réplica do sinal original. Os passos de quantização são adaptados pelo mesmo algoritmo que no codificador e a partir dos mesmos valores quantizados.[1] Deste modo, codificador e descodificador são mantidos em sincronismo e não há necessidade de transmitir informação secundária.

[1]Se não houver erros de transmissão, como supomos neste trabalho, então $\mathbf{k}' = \mathbf{k}$ e portanto, $\mathbf{y} = \hat{\mathbf{x}}$.

Isto é já uma consequência da principal característica diferenciadora deste esquema em relação a outros codificadores perceptuais: a utilização de *quantização retroadaptativa*. Este aspecto será tratado na Secção 7.2. Na secção seguinte, descrevemos uma implementação de um codificador perceptual retroadaptativo para facilitar a exposição da técnica e fornecer uma base concreta para a discussão e apresentação de resultados.

7.1 O Codificador BAPAC

O algoritmo de codificação BAPAC,[2] desenvolvido em [131], é um exemplo de um sistema de codificação perceptual de áudio retroadaptativo que segue a estrutura da Figura 7.1. Neste sistema, um banco de filtros subamostrados decompõe um sinal áudio monofónico em bandas de larguras não uniformes. Os sinais das bandas sofrem uma quantização logarítmica de ganho variável e os valores quantizados são comprimidos por um codificador aritmético para transmissão ao receptor. Os ganhos dos quantizadores são adaptados dinamicamente por um algoritmo que incorpora um modelo psicoacústico capaz de estimar o limiar de mascaramento em função do sinal quantizado previamente. Segue-se uma descrição das características mais importantes e alguns aspectos de projecto de cada um destes blocos.

7.1.1 O Banco de Filtros

O banco de filtros de análise usado no codificador BAPAC é uma estrutura de dois andares do tipo *Split-and-Merge* (Sec. 3.4.2), que proporciona uma decomposição do sinal em bandas de largura não uniforme. O primeiro andar é formado por uma ELT [103] que divide o sinal em 256 bandas uniformes. No segundo andar, as primeiras 32 bandas passam inalteradas para a saída, enquanto as restantes são agrupadas em grupos de duas, quatro, oito ou dezasseis, e recombinadas com ELTs inversas para produzir bandas mais largas mas com melhor resolução temporal. A Tabela 7.1 resume esta estrutura.

Tabela 7.1: Especificação da estrutura de decomposição do codificador BAPAC. Indicam-se as transformadas incluídas em cada andar e os atrasos de compensação necessários.

	Entrada		Transformações	Saída	
Andar	Bandas	Formato		Bandas	Formato
1	1–1	1×256	$1\times[\mathrm{ELT}_{M=256,K=2}]$	1–256	256×1
2	1–32	32×1	$32\times[z^{-2}]$	1–32	32×1
	33–48	16×1	$8\times[\mathrm{IELT}_{M=2,K=2} + z^{-1}]$	33–40	8×2
	49–60	32×1	$8\times[\mathrm{IELT}_{M=4,K=2} + z^{-2}]$	41–48	8×4
	61–128	48×1	$6\times[\mathrm{IELT}_{M=8,K=2} + z^{-4}]$	49–54	6×8
	129–256	128×1	$8\times[\mathrm{IELT}_{M=16,K=2} + z^{-8}]$	55–62	8×16

Os filtros protótipos das ELTs foram projectados para uma sobreposição de 75% (factor de sobreposição $K = 2$) e foram optimizados segundo o critério de minimização da energia na banda de corte, como descrito na Subsecção 3.3.5. Houve o cuidado de fixar os parâmetros *LB* e *TP* das várias transformadas, de forma a garantir a semelhança dos filtros protótipo,

[2] *Backward-Adaptive Perceptual Audio Coder.*

e assim permitir o cancelamento parcial de *aliasing* na estrutura SAM, como discutido na Subsecção 3.4.2. Todas as saídas passam por linhas de atraso que normalizam o atraso global para permitir a reconstrução perfeita no banco de síntese que tem uma estrutura dual.

A Tabela 7.2 mostra a resolução no tempo e na frequência das 62 bandas resultantes. Todas as bandas têm largura inferior a 1 Bark, o que permite um aproveitamento eficiente do fenómeno de mascaramento. Simultaneamente, a resolução temporal é bastante boa, especialmente nas frequências mais altas, para evitar a violação das condições de mascaramento no tempo. O uso de estruturas compostas de ELTs garante ainda as propriedades de decimação máxima e reconstrução perfeita e resulta numa complexidade computacional de apenas 21 adições e 12 multiplicações por amostra. A combinação análise-síntese introduz um atraso total de 1792 amostras (40.6 ms a 44100 Hz) entre entrada e saída.

Tabela 7.2: Resolução das 62 bandas resultantes, no tempo (Δt) e na frequência (Δf e Δz).

Bandas	Δt (ms)	Δf (Hz)	Δz (Bark)
1–32	5.80	86	0.85–0.19
33–40	2.90	172	0.35–0.22
41–48	1.45	345	0.42–0.26
49–54	0.73	689	0.47–0.31
55–62	0.36	1378	0.54–0.23

7.1.2 Quantização

Cada banda do sinal é quantizada por um quantizador independente. Para simplificar a implementação, todos os quantizadores têm 127 níveis distribuídos simetricamente segundo uma curva característica de forma fixa. Apenas o ganho ou passo de quantização Δ é variado de banda para banda e de amostra para amostra. Os quantizadores são do tipo *mid-tread*, isto é, o zero é um dos níveis de saída. A curva característica tem uma forma aproximadamente logarítmica semelhante às da lei-A ou lei-μ usadas em telefonia digital (ver Figura 7.2). Para amostras de amplitude moderada (em relação a Δ), o quantizador tem um comportamento semelhante ao de um quantizador uniforme de passo Δ. Para amostras de amplitude mais elevada, o passo de quantização aumenta proporcionalmente à amplitude. Isto permite uma grande gama dinâmica e proporciona também alguma modelação de ruído visto que, mesmo quando o sinal aumenta bruscamente, a relação sinal-ruído não ultrapassa os 31 dB, o que é suficiente em termos perceptuais.

7.1.3 Codificação de Comprimento Variável

Os símbolos debitados pelos quantizadores são codificados com um codificador aritmético adaptativo. O modelo estatístico usado mantém 62 tabelas de probabilidades distintas, procurando modelar separadamente as distribuições das várias bandas, e comuta entre elas à medida que processa os símbolos correspondentes a cada uma. Assim, o codificador aritmético integra simultaneamente a função de multiplexagem dos dados das várias bandas. A adaptação do modelo é feita após a codificação de cada amostra, actualizando os contadores de ocorrências na tabela adequada.

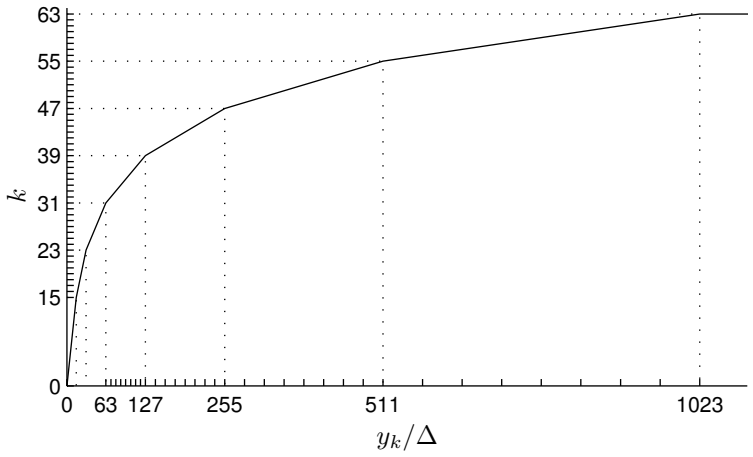


Figura 7.2: Curva de compressão-expansão do quantizador do BAPAC.

7.1.4 Algoritmo de Adaptação Perceptual

O algoritmo de adaptação baseia-se num modelo psicoacústico explícito que estima continuamente o valor do limiar de mascaramento. O modelo é semelhante ao usado em [83] e compreende os seguintes passos:

Espalhamento temporal As amostras de cada banda passam por um rectificador de lei quadrática e por um filtro passa-baixo recursivo de primeira ordem, para produzir uma estimativa da energia nessa banda. As constantes de tempo dos filtros variam de banda para banda, segundo a fórmula empírica (6.6), para modelar a dependência com a frequência do fenómeno de pós-mascaramento. O fenómeno de pré-mascaramento não é modelado explicitamente devido à sua curta duração.

Espraiamento na frequência Para estimar o perfil de excitação na membrana basilar, procede-se a uma convolução com a função de espraiamento dada em (6.3). Esta operação é linear e foi implementada com uma multiplicação por uma matriz de dimensão 256×256, mas relativamente esparsa (11% de elementos não nulos).

Dedução do índice de mascaramento De cada banda é subtraído, numa escala logarítmica, o índice de mascaramento de ruído por tons (TMN), avaliado pela fórmula empírica (6.7), para obter a estimativa do limiar de mascaramento. O índice de mascaramento de tons por ruído (NMT) não foi considerado nesta versão do codificador.

Correcção para o limiar absoluto O limiar de mascaramento calculado no passo anterior é comparado com o limiar absoluto de audição e é corrigido nos pontos em que lhe for inferior. Considera-se que o ganho acústico é tal que um sinal sinusoidal de amplitude $\pm\frac{1}{2}$LSB produza um som com intensidade de 0 dB SPL.

Os passos de quantização são então determinados de forma que a potência esperada do erro de quantização não ultrapasse o limiar calculado. (A potência do erro de quantização é estimada por $N = \Delta^2/12$, que se verificou ser uma aproximação razoável em operação normal.) Os passos são ainda multiplicados por um parâmetro global ϕ—chamado *nível de*

qualidade—que permite controlar o compromisso qualidade/compressão: $\phi > 1$ aumenta os passos de quantização, degradando a qualidade mas conseguindo maior compressão; $\phi < 1$ diminui os passos, garantindo uma "margem de segurança" abaixo do limiar à custa de um débito mais elevado. Na versão descrita, o parâmetro ϕ é mantido fixo ao longo de todo o sinal, mas poderia ser transmitido regularmente para permitir um ajuste dinâmico da qualidade e débito.

Devido à estrutura multi-resolução da saída do banco de filtros, o algoritmo de adaptação é executado incrementalmente, em alternância com a quantização, de forma a garantir uma adaptação amostra-a-amostra que integre a informação quantizada mais recentemente. Com este algoritmo e o banco de filtros utilizado, evita-se uma análise espectral paralela e as múltiplas conversões entre diferentes partições do plano tempo-frequência que são necessárias noutros codificadores, como os recomendados pelo MPEG [77]. O algoritmo de adaptação completo pode ser implementado com cerca de 30 multiplicações e 30 adições por amostra. Necessita também de cerca de 8000 palavras de memória fixa para armazenamento de coeficientes, e quantidades desprezáveis de memória de acesso aleatório.

7.1.5 Desempenho

O desempenho do sistema BAPAC foi avaliado [131, Cap. 5] utilizando sete trechos musicais—extraídos na sua maioria do disco compacto EBU SQAM [39]—codificados em três níveis de qualidade decrescente: $\phi = 1$, $\phi = 2$ e $\phi = 3$. Cada trecho foi ainda codificado com uma implementação disponível em *shareware* do MPEG Layer III a 64 kbit/s.[3] As quatro versões codificadas de cada trecho foram avaliadas em termos do débito produzido e da qualidade subjectiva medida em testes de audição.

Os testes de audição seguiram uma metodologia de teste de *estímulo triplo* com *referência escondida*, idêntica à usada com bons resultados nos testes realizados no âmbito do MPEG e do CCIR [118], e normalizada mais tarde pelo ITU-R [74]. Em cada teste eram apresentados três sinais ao ouvinte: R, X e Y. O sinal R era sempre o trecho original para ser usado como sinal de referência. Um de X e Y, escolhido aleatoriamente pelo computador, era uma das quatro versões codificadas enquanto o outro era uma cópia da referência R. O ouvinte podia escutar os sinais repetidamente e pela ordem que entendesse. A sua tarefa consistia em classificar a degradação percebida de cada um dos sinais X e Y em relação à referência R, atribuindo uma pontuação tirada da escala de degradação de 5 pontos do CCIR [73]:

5 *Imperceptible*
4 *Just perceptible but not annoying*
3 *Perceptible and slightly annoying*
2 *Annoying*
1 *Very annoying*

Participaram dez pessoas nos testes de audição. Cada ouvinte completou, por uma ordem aleatória, duas provas de audição de cada uma das quatro versões codificadas dos sete trechos.

A Tabela 7.3 mostra a pontuação média obtida por cada versão codificada dos vários trechos. Também se apresenta a média das pontuações ou *Mean Opinion Score* (MOS) e o débito médio obtido por cada codificador.

A 2.35 bits por amostra, o algoritmo proposto permite uma codificação de alta qualidade. No entanto, para um débito comparável ao do codificador de Layer III, apresenta uma

[3] Os programas `l3enc` e `l3dec` desenvolvidos no Fraunhofer-Gesellschaft.

Tabela 7.3: Resultados dos testes de avaliação: pontuações médias de cada versão codificada, MOS e débito médio (em bits por amostra).

	BAPAC $\phi = 1$	BAPAC $\phi = 2$	BAPAC $\phi = 3$	Layer III 64 kb/s
Castanholas	4.20	3.85	3.70	4.20
Cravo	4.30	3.45	2.55	4.30
Sarasate	4.60	3.75	2.40	4.75
Sting	4.75	4.65	4.30	4.70
Stravinsky	4.85	4.50	3.90	4.40
Suzanne	3.00	1.85	1.40	3.25
Violino	3.00	1.65	1.20	3.40
MOS	4.10	3.39	2.78	4.14
Débito	2.35	1.78	1.46	1.42

qualidade bastante inferior. Isto pode dever-se, em parte, a uma adaptação demasiado lenta do codificador aritmético que é inicializado com tabelas optimizadas para a situação $\phi = 1$. Alguns trechos obtiveram consistentemente pontuações baixas em todas as versões, o que indicia eventuais deficiências no modelo psicoacústico ou no banco de filtros.

7.2 Retroadaptação em Codificadores Perceptuais

Uma característica fundamental do sistema de codificação aqui proposto é a de ser um sistema com *quantização retroadaptativa*.[4] Apesar de muito usada em codificadores tradicionais de voz e até de áudio, de que é exemplo a norma G.722 do CCITT [100], a técnica de retroadaptação não tem sido aplicada com frequência em codificadores perceptuais (ver discussão do caso AC-3 na Sec. 7.3). Por esta razão, justifica-se um estudo das suas vantagens e inconvenientes e uma verificação da sua aplicabilidade à codificação perceptual de áudio.

7.2.1 Características, Vantagens e Inconvenientes

Para melhor apreciação das implicações do uso de retroadaptação (AQB), considere-se uma variante do sistema com proadaptação (AQF) como mostra a Figura 7.3. Esta versão do sistema segue a estrutura usual dos codificadores perceptuais no domínio da frequência representada na Figura 2.1.

Comparando com a Figura 7.1, vemos que a diferença essencial que distingue os dois sistemas reside na informação usada pelo algoritmo de adaptação: no sistema AQF, a estimação dos parâmetros de adaptação baseia-se no sinal exacto $\mathbf{x}$; no sistema AQB, a adaptação depende do sinal já quantizado $\hat{\mathbf{x}}$. Em consequência desta diferença estrutural, os dois sistemas apresentam um conjunto características distintas que analisamos abaixo.

Informação secundária Um sistema AQB praticamente não tem informação secundária.[5]

Por exemplo no sistema BAPAC a informação secundária é formada apenas pelo factor

[4] *Backward-adaptive quantization*, ou AQB na notação de [80, Sec. 4.10.1].

[5] Consideramos como *informação secundária* apenas a informação necessária à adaptação do descodificador, não incluindo informação de sincronismo ou correcção de erros.

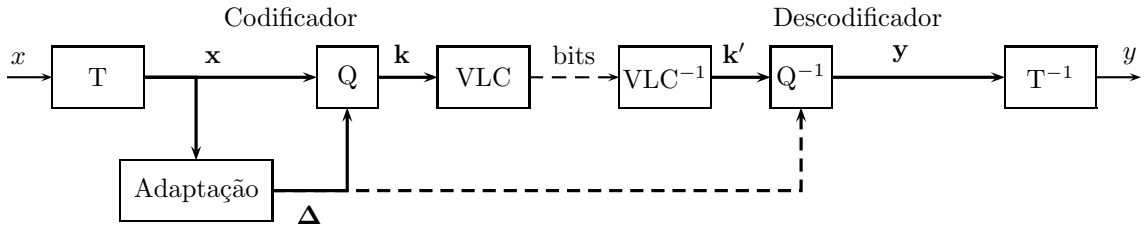


Figura 7.3: Variante proadaptativa do sistema de codificação proposto (PAC-AQF).

de qualidade ϕ, que não precisa de ser transmitido frequentemente. Mesmo que fosse transmitido com 4 bits de bloco em bloco, o que é mais do que suficiente, isso representaria um acréscimo de apenas 0.016 bits por amostra: um *overhead* de cerca de 1% a 64 kbit/s. Num sistema AQF, pelo contrário, a informação secundária pode consumir uma fracção considerável da capacidade disponível. Por exemplo, no PXFM a 4 bits por amostra esta fracção é de 16%, e no MPEG *Layer* I a 128 kbit/s pode atingir 30%.

Simplicidade de projecto Num sistema AQF existem dois importantes canais de informação a transmitir: o canal principal (as amostras quantizadas) e o canal secundário (passos de quantização, atribuição de bits e/ou outros parâmetros de adaptação). É necessário projectar a codificação de cada um desses canais, fazer a sua multiplexagem e determinar a melhor distribuição de bits entre os dois. Como a quantidade de informação secundária num sistema AQB é desprezável, o seu projecto é mais simples porque se pode concentrar essencialmente na codificação do canal principal.

Frequência de adaptação Em AQF os parâmetros de quantização só são ajustados no princípio de um bloco de várias amostras (no *Layer* II, por exemplo, é transmitido um factor de escala por cada bloco de 12 ou 36 amostras). De outro modo, o *overhead* de informação secundária seria incomportável. Num sistema AQB, por não haver informação secundária, a adaptação pode ser feita amostra-a-amostra sem qualquer acréscimo de débito, o que resulta num acompanhamento mais próximo das transições do sinal e pode evitar o aparecimento de pré-ecos.

Complexidade O codificador num sistema AQB é mais simples que num sistema AQF porque não é necessário codificar informação secundária. Em compensação, o descodificador é mais complexo porque inclui um modelo psicoacústico completo. De facto, o descodificador AQB tem praticamente a mesma complexidade que o codificador. Esta simetria é uma das principais desvantagens dos sistemas de codificação AQB porque em certas aplicações o codificador pode ser muito complexo e caro mas os descodificadores devem ser simples e baratos.

Evolutividade Nestas aplicações eminentemente assimétricas, os sistemas AQB apresentam outro inconveniente: não é possível melhorar o algoritmo de adaptação sem modificar todos os descodificadores.

Robustez Quando ocorre um erro na transmissão da sequência binária, o descodificador gera um *burst* de amostras erradas. Num descodificador proadaptativo, esse *burst* termina geralmente no final do bloco com a recepção de novos parâmetros, mas num sistema

AQB o erro afecta o cálculo de todos os parâmetros futuros e os seus efeitos podem propagar-se indefinidamente. No entanto, poderá ser possível projectar o sistema AQB de forma que o efeito do erro diminua rapidamente ao longo do tempo, como acontece no chamado quantizador adaptativo robusto [55].

Qualidade dos parâmetros estimados Num sistema AQB, a estimação do modelo perceptual e dos parâmetros de quantização associados é afectada pelo erro de quantização introduzido no sinal. Para débitos baixos, a quantização pode ser muito grosseira e afectar significativamente o cálculo dos parâmetros, piorando ainda mais a qualidade de codificação. Num sistema AQF, o modelo perceptual tem acesso ao sinal original exacto, mas os parâmetros calculados são sempre sujeitos a uma degradação por quantização e decimação, a fim de serem transmitidos ao receptor.

As principais vantagens dos sistemas AQB são a eliminação da informação secundária, a simplicidade de projecto e a possibilidade de adaptação amostra-a-amostra. Os sistemas AQF por sua vez, têm a propriedade de não necessitarem de algoritmo de adaptação no descodificador, o que é uma vantagem importante em certas aplicações. Quanto à estimação dos parâmetros de quantização, não é inteiramente óbvio qual dos dois sistemas é mais exacto. Parece mesmo haver a possibilidade de os erros na estimação inviabilizarem o uso de retroadaptação em codificadores perceptuais. Esta questão será tratada a seguir.

7.2.2 Efeito do Ruído de Quantização no Modelo Psicoacústico

A possibilidade de o ruído de quantização perturbar irremediavelmente a estimação do modelo perceptual foi identificada como uma questão crucial que pode comprometer a viabilidade da retroadaptação em codificadores perceptuais. Uma análise qualitativa do comportamento do sistema tanto em condições normais como em condições extremas fornece-nos bons argumentos para refutar essa hipótese:

- Numa situação normal, o limiar calculado em cada ponto no tempo e na frequência depende da energia do sinal em muitos outros pontos no passado e em bandas adjacentes. Logo, é pouco provável que os erros de quantização se conjuguem para afectar o limiar num mesmo sentido. Assim, a estimativa do limiar tem uma variância menor do que a do erro de quantização de qualquer das células que para ele contribuem. Por outras palavras: as operações de espalhamento temporal e espectral *filtram* os erros de quantização.

- Se o sinal numa banda descer muito abaixo do respectivo passo de quantização, passa a ser quantizado com o nível zero, o que provoca uma tendência de decaimento rápido no limiar e consequentemente, no passo de quantização. Esse decaimento respeita as condições de pós-mascaramento e só é interrompido quando o passo diminui suficientemente abaixo do nível do sinal ou quando o limiar atinge o nível imposto pelas outras bandas ou pelo limiar absoluto, como é desejável. A característica *mid-tread* dos quantizadores é um factor determinante para este comportamento. Quantizadores *mid-rise* produziriam, neste caso, erros superiores ao sinal, com consequências potencialmente desagradáveis na estimação do limiar.

- Se o sinal crescer acima do ponto de saturação do quantizador, a sua energia é subavaliada, obrigando o limiar a aumentar mais lentamente que o desejável. Isto conduz a

uma sobrecodificação desnecessária do sinal. A grande gama dinâmica e a modelação de ruído proporcionada pelos quantizadores logarítmicos permitem minorar muito este problema.

Por estas razões acreditamos que a estimação do limiar de mascaramento não é grandemente afectada pelo erro de quantização, especialmente se houver o cuidado de usar quantizadores *mid-tread* com boa gama dinâmica. Uma experiência realizada com o codificador BAPAC permitiu verificar e quantificar o efeito real da quantização no cálculo do limiar [131, Sec. 4.5.2]. A experiência consistiu em codificar um sinal de teste com o BAPAC a três níveis de qualidade progressivamente pior, $\phi = 1, 2, 4$, e registar os limiares de mascaramento $\mathbf{\Psi}_\phi$ obtidos em cada caso. Também foi registado o limiar "exacto" $\mathbf{\Psi}_0$ obtido por aplicação directa do modelo psicoacústico ao sinal original, sem erros de quantização. Finalmente, calculámos histogramas das relações $\mathbf{\Psi}_\phi/\mathbf{\Psi}_0$.

A Figura 7.4 mostra o histograma obtido com $\phi = 4$, que representa uma quantização muito grosseira e corresponde a um débito médio inferior a 1.4 bits por amostra. Este e outros resultados são resumidos na tabela seguinte.

	Percentagem de amostras com $\mathbf{\Psi}_\phi/\mathbf{\Psi}_0$ inferior a:					
	0.25 dB	0.75 dB	1.25 dB	1.75 dB	2.25 dB	2.75 dB
$\phi = 1$	93.4	99.7	99.9	100.0	100.0	100.0
$\phi = 2$	76.3	98.0	99.8	100.0	100.0	100.0
$\phi = 4$	59.1	82.4	94.0	98.4	99.6	99.9

Constata-se que mesmo para $\phi = 4$, quase 60% dos valores estimados não se desviam mais que 0.25 dB em relação ao limiar "exacto". Para $\phi = 2$ e $\phi = 1$, essa fracção cresce para 76% e 93%, respectivamente. Em geral, o erro de estimação raramente ultrapassa 2 dB.

Assim, confirma-se experimentalmente que o erro de quantização não afecta significativamente o cálculo do limiar de mascaramento. Mais, se considerarmos que num sistema AQF como o *Layer* I só é transmitido um factor de escala por cada 12 amostras e que o seu valor é quantizado com uma resolução de 2 dB, não é difícil admitir que os sistemas AQB podem ser superiores também neste aspecto.

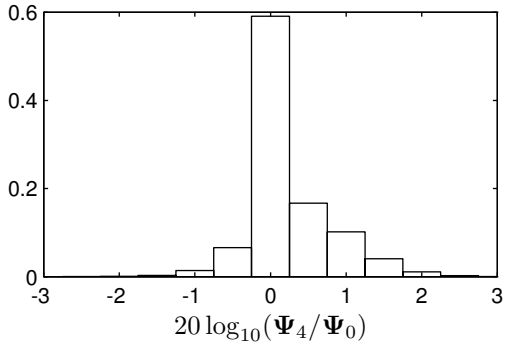


Figura 7.4: Histograma das diferenças (em dB) entre o limiar estimado a partir de amostras quantizadas grosseiramente ($\phi = 4$) e o limiar estimado a partir de amostras não quantizadas.

7.2.3 Efeito da Retroadaptação no Desempenho

Tendo em conta a discussão e os resultados apresentados atrás sobre o efeito dos erros de quantização na estimação do limiar de mascaramento, não seria de esperar que a opção pela retroadaptação pudesse ter algum impacto negativo perceptível no desempenho do codificador. Mesmo assim, num estudo apresentado em [128], procurou-se verificar essa hipótese. Para isso, construiu-se um codificador idêntico ao BAPAC, mas com quantização proadaptativa. O objectivo era determinar o impacto da escolha entre proadaptação ou retroadaptação, procurando isolá-la o mais possível de outras opções de projecto. Isto não é tão fácil como parece, visto que a escolha da direcção de adaptação tem repercussões profundas no projecto do codificador. Por exemplo, na versão proadaptativa (FAPAC) teria de se limitar a resolução e frequência dos parâmetros de adaptação para manter a informação secundária dentro de limites aceitáveis. Por outro lado, na versão retroadaptativa (BAPAC) teria de se contar com um atraso de uma amostra no ciclo de adaptação; não se pode usar as amostras mais recentes para adaptar a sua própria quantização. Assim, para uma comparação justa do desempenho seria necessário optimizar a resolução, frequência e codificação dos parâmetros de adaptação do FAPAC, e optimizar o algoritmo de adaptação do BAPAC para tentar compensar o efeito do atraso. Por esta altura as duas versões já seriam tão distintas que qualquer diferença de desempenho poderia facilmente ser atribuída a deficiências nos processos de optimização, que não são triviais, particularmente no caso do FAPAC. Consequentemente, foi adoptada uma estratégia diferente: usar exactamente o mesmo algoritmo de adaptação nos dois codificadores, mantendo a actualização amostra-a-amostra no FAPAC e sem qualquer tentativa de compensação do atraso no BAPAC. Só se impôs que no sistema FAPAC os passos de quantização fossem transmitidos com uma resolução de 1.5 dB, que é o valor usado em vários codificadores existentes. A expectativa era que a qualidade das duas versões fosse tão próxima que se pudesse tirar conclusões quanto ao débito.

Metodologia de Teste

Para comparar a qualidade dos codificadores BAPAC e FAPAC, realizaram-se testes de audição discriminativos. Codificaram-se seis trechos musicais com ambas as versões e a dois níveis de qualidade, $\phi = 1$ e $\phi = 3$. Em cada prova, o ouvinte escutava, quantas vezes e pela ordem que entendesse, um par de sinais, X e Y, codificados ao mesmo nível de qualidade ϕ, e tinha de escolher uma das opções seguintes:

(E) *Não noto diferenças.*
(S) *Há diferenças, mas a qualidade é a mesma.*
(X) *X é melhor.*
(Y) *Y é melhor.*

Os ouvintes eram instruídos a considerarem *melhor* a versão que lhes parecesse mais fiel ao sinal original de referência, que também era disponibilizado e podiam escutar opcionalmente. Toda a interacção se processava entre o ouvinte e um programa de computador através uma interface gráfica simples. Para cada trecho musical realizavam-se 6 provas, duas das quais continham exactamente o mesmo sinal em X e Y. Os ouvintes foram informados disso. Doze ouvintes participaram nos testes, incluindo sete estudantes de Música da Unversidade de Aveiro. Alguns ouvintes não completaram o conjunto completo de provas, mas os resultados das provas que completaram foram contabilizados. Apesar de se mostrarem empenhados, de-

Tabela 7.4: Resultados dos testes auditivos de comparação dos codificadores BAPAC e FAPAC (adaptado de [128]). (a) Provas em que X e Y eram a mesma versão. (b) Provas em que X e Y eram versões diferentes. É indicado o número de vezes que cada opção foi escolhida (B indica uma preferência por BAPAC e F uma preferência por FAPAC).

(a)

X=Y	Opção			Número
	E	S	X ou Y	de Provas
$\phi = 1$	36	7	27	70
$\phi = 3$	22	22	25	69

(b)

X≠Y	Opção				Número
	E	S	B	F	de Provas
$\phi = 1$	69	16	29	20	134
$\phi = 3$	47	28	26	34	135

pois de completarem as provas muitos ouvintes exprimiram a grande dificuldade que sentiram em distinguir os sinais.[6]

Comparação do Desempenho

Os resultados estão resumidos na tabela 7.4. Para o nível de qualidade $\phi = 1$, os ouvintes não notaram diferenças (resposta E) em 51% das 134 provas em que escutaram sinais distintos. O resultado foi perfeitamente idêntico nas 70 provas em que escutaram efectivamente a mesma versão em X e Y. Para a qualidade mais baixa ($\phi = 3$), a percentagem de respostas E baixou para 35% quando X≠Y, mas baixou igualmente para 32% quando X=Y. Em resumo, a capacidade de distinguir X de Y não melhorou nas provas em que X e Y eram efectivamente versões diferentes. Por estes resultados parece-nos claro que os ouvintes eram incapazes de distinguir coerentemente as duas versões, pelo que também não seria de esperar qualquer preferência significativa por uma delas.

O débito foi medido em blocos de 256 amostras, para um conjunto de sete trechos musicais, codificados quer pelo BAPAC, quer pelo FAPAC. A Figura 7.5 mostra a evolução do débito medido ao longo do trecho "Castanholas", que apresentava as maiores flutuações. As variações de débito são muito semelhantes nos dois codificadores. As maiores diferenças ocorrem durante os picos de débito que coincidem com os impulsos no sinal. Nestes períodos transitórios o BAPAC apresenta alguma sobrecodificação devida provavelmente ao ligeiro atraso de adaptação referido atrás. Apesar disto, o débito médio obtido para os vários sinais foi só entre 1.6% e 4% mais alto no BAPAC do que no FAPAC, não contando com a informação secundária deste último.

Em conclusão, mesmo quando comparado com um sistema com proadaptação amostra-a-amostra e sem informação secundária, que é uma situação perfeitamente irrealizável, o sistema retroadaptativo apresenta a mesma qualidade com um débito apenas ligeiramente superior.

7.3 Discussão

A vantagem principal da retroadaptação é a eliminação da informação secundária. Isto facilita o projecto do sistema porque não é necessário encontrar o melhor compromisso entre informação primária e secundária, o que pode ser uma tarefa complicada num codificador proadaptativo avançado [1]. A adaptação pode ser tão frequente quanto se queira, inclusive

[6] Um dos ouvintes chegou a comparar o teste à "PEPSI Challenge", referindo-se a uma campanha publicitária que mostrava supostos transeuntes a provar refrigerantes de duas marcas concorrentes.

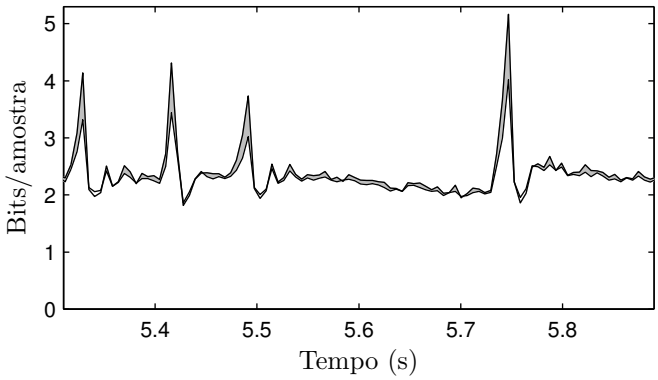


Figura 7.5: Flutuações de débito na codificação do sinal "Castanholas" no sistemas BAPAC e FAPAC. As áreas sombreadas indicam os períodos nos quais o débito do BAPAC excede o débito do FAPAC.

amostra-a-amostra. A frequência de adaptação poderá ser limitada por questões de complexidade computacional, mas não por restrições de débito.

Não encontrámos na literatura exemplos de sistemas de codificação que apliquem a técnica de retroadaptação perceptual da quantização, tal como apresentada neste capítulo. A abordagem do AC-3 [153, 30] pode ser considerada de retroadaptação, mas numa forma bastante mais limitada. Aliás, achamos que pode mais correctamente ser caracterizada como de proadaptação, visto que a adaptação só depende de uma fracção da informação transmitida, os expoentes, que por sua vez não dependem da adaptação, dependendo apenas das amplitudes do sinal original quantizadas não adaptativamente. Assim, os expoentes podem ser considerados como informação secundária com a função dupla de factores de escala e de dados para a estimação do limiar de mascaramento e consequente adaptação da quantização das mantissas. Este reaproveitamento dos expoentes, porém, distingue o AC-3 de sistemas proadaptativos típicos, trazendo-lhe algumas das vantagens da retroadaptação.

Experiências com o codificador retroadaptativo BAPAC demonstraram um comportamento robusto do modelo psicoacústico, mesmo quando sujeito a sinais quantizados grosseiramente. Mesmo nestas circunstâncias extremas, os erros de estimação do limiar no BAPAC apresentam um dispersão menor que a dos erros tipicamente introduzidos por codificadores proadaptativos para a transmissão de parâmetros equivalentes. É de esperar que outros codificadores retroadaptativos tenham um comportamento idêntico.

O desempenho do sistema BAPAC revelou algumas deficiências, nomeadamente a fraca qualidade de codificação de alguns trechos musicais, mesmo com débitos relativamente altos. No entanto, uma comparação directa com uma versão proadaptativa idealizada mostrou que a opção pela retroadaptação não é a causa desses problemas.

Capítulo 8

Conclusões e Trabalho Futuro

Neste trabalho, apresentámos uma estrutura de codificação perceptual de áudio aplicando quantização retroadaptativa. Salientámos as vantagens da retroadaptação e verificámos a viabilidade da sua aplicação a codificadores perceptuais. Tendo em conta as características da retroadaptação, as restrições que impõe, mas também as liberdades que permite, podemos identificar, a partir do estudo dos vários blocos que compõem a estrutura apresentada—decomposição espectral, quantização, codificação sem perdas e adaptação perceptual—as alternativas mais adequadas e com maior potencial.

O Banco de Filtros Parece-nos conveniente usar um único banco de filtros quer para fazer a decomposição do sinal, quer para a análise espectral requerida pelo modelo perceptual. Esta opção permite uma economia de recursos computacionais e não impõe grandes restrições (ver Secção 3.2). Os requisitos de subamostragem crítica, reconstrução perfeita e baixa complexidade computacional apontam para soluções baseadas em transformadas moduladas com sobreposição (Sec. 3.3.5). Uma das mais interessantes é a simples MLT (ou MDCT), que é provavelmente a transformada mais popular em codificadores de áudio actuais. Com uma sobreposição de apenas 50%, tem atrasos relativamente baixos, mas permite mesmo assim alguma liberdade na escolha da janela, o que se reflecte nas respostas em frequência dos filtros. Esta transformada faz uma decomposição uniforme, por isso muitos codificadores comutam dinamicamente entre janelas de diferentes dimensões tentando ajustar a resolução temporal e espectral ao próprio sinal.

Para obter uma decomposição não uniforme, mais adequada às características da audição, podem combinar-se MLTs e IMLTs numa estrutura SAM, como se fez no BAPAC. No entanto, acreditamos que num codificador que possa ser adaptado amostra-a-amostra, os requisitos de resolução temporal não são tão exigentes como num codificador adaptado menos frequentemente, e uma transformada uniforme fixa pode ser uma boa solução. Por exemplo, para sinais a 44100 amostras/s, uma MLT de 256 bandas tem uma resolução espectral de 86 Hz, semelhante à largura mínima das bandas críticas, e uma resolução temporal de 5.8 ms, inferior às constantes de tempo do pós-mascaramento mesmo nas frequências altas. Num codificador como o *Layer* I, pelo contrário, a adaptação ocorre apenas a cada 8.7 ms (de 12 em 12 amostras de cada sub-banda), apesar de se usar um banco de apenas 32 filtros com a resolução grosseira de 690 Hz. A utilização de uma MLT simples já foi explorada em versões simplificadas do BAPAC implementadas, num dos casos num processador digital de sinal, em trabalhos de fim-de-curso realizados no Departamento de Electrónica e Telecomunicações da Universidade

de Aveiro [127, 134].

Quantização Num sistema retroadaptativo, é importante que os quantizadores tenham uma grande gama dinâmica porque não há informação prévia sobre "ataques" súbitos no sinal. Por outro lado, é perceptualmente aceitável introduzir erros absolutos maiores em sinais de amplitude maior. De facto, a lei de Weber, que modela razoavelmente a sensibilidade a variações de intensidade dos estímulos, sugere que o erro relativo deve ser constante. Assim, parece apropriado usar quantizadores aproximadamente logarítmicos. Também é conveniente que sejam do tipo *mid-tread* porque dessa forma minimizam o erro de quantização e o débito para sinais fortemente concentrados em torno de zero, que é o caso típico dos coeficientes obtidos por transformações de sinais de áudio. Simultaneamente reduz-se o impacto desses erros no modelo perceptual.

Há dois graus de liberdade que se podem controlar facilmente num codificador logarítmico. Um determina a *escala* da função e, consequentemente, o máximo erro absoluto para amplitudes baixas, i.e. o passo de quantização. É o parâmetro mais importante e o único usado no codificador BAPAC. O outro parâmetro determina a *curvatura* e o máximo erro relativo para amplitudes altas.

Codificação Sem Perdas Os valores quantizados serão codificados aritmeticamente. As múltiplas vantagens dos códigos aritméticos já foram expostas (Cap. 5), mas importa recordar as mais significativas para o caso presente de um codificador que se pretende de baixo débito. A eficiência da codificação mesmo para símbolos muito prováveis é particularmente importante devido à grande prevalência de ocorrência do nível zero à saída dos quantizadores, pelo menos em certas bandas. Para conseguir menos de 1 bit para estes símbolos com códigos de comprimento inteiro, por exemplo, seria necessário fazer empacotamento de símbolos com a complicação adicional associada. Outro aspecto fundamental da codificação aritmética é a possibilidade de integrar facilmente modelos estatísticos complexos, dinâmicos e adaptativos ao sinal. Medidas estatísticas de sinais de áudio filtrados por MLTs apresentam fortes variações ao longo do tempo e de banda para banda, pelo que se justifica considerar modelos adaptativos. Neste aspecto, acreditamos agora que o modelo adaptativo usado no BAPAC, com correcções de probabilidade localizadas, é demasiado lento para reflectir a forte dinâmica dos sinais de áudio. Uma técnica de comutação de tabelas ou alguma forma de representação paramétrica de distribuições de probabilidade deverá produzir resultados bastante melhores.

Algoritmo de Adaptação Perceptual Os modelos psicoacústicos descritos na literatura de codificação perceptual de áudio são invariavelmente baseados no conceito de *limiar de mascaramento*, ou seja, procuram estimar o perfil de ruído que pode ser adicionado ao sinal sem causar distorção audível. Embora não pareça haver nada de essencialmente errado nesta abordagem, a verdade é que a sua implementação correcta é bem mais difícil do que os modelos em uso fazem crer. De facto, os modelos usados em codificadores perceptuais conhecidos parecem basear-se numa interpretação errada do limiar de mascaramento, como observou Veldhuis [160]. Os limiares de mascaramento medidos em experiências psicoacústicas referem-se a situações simples de mascaramento de um tom por uma banda de ruído, por exemplo. Não se podem generalizar facilmente para situações de mascaramento de vários tons ou várias bandas de ruído por um sinal complexo. Numa das propostas pioneiras de codificação perceptual [83], o autor parecia consciente destes problemas ao reconhecer a

necessidade de uma "desconvolução" com as curvas de espalhamento para obter o verdadeiro limiar de mascaramento, mas propunha uma solução de compromisso face à dificuldade desse processo. Infelizmente, desenvolvimentos posteriores, incluindo o nosso BAPAC, não parecem ter feito progressos neste domínio.

O mascaramento e os seus limiares devem ser vistos como resultado de processos internos do sistema auditivo, que podem ser usados para inferir sobre o seu funcionamento, mas não directamente como modelos da audição *per se*. Uma abordagem diferente ao problema da modelação perceptual consiste na determinação de *representações internas—padrões de excitação*—do sinal original e do reconstruído, e posterior comparação para quantificar a capacidade de o ouvido detectar as diferenças entre eles. Esta estratégia tem sido aplicada com bastante sucesso em medidas objectivas de qualidade perceptual, mas não em codificadores perceptuais, talvez pela complexidade aparente de algumas implementações. No entanto, acreditamos que para aplicação num dado codificador, um modelo perceptual de representação interna não deverá precisar de toda a generalidade dos usados em medidas de qualidade, visto que a própria estrutura do codificador limita o tipo de distorções introduzidas.

Em [130] apresentámos um modelo de representação interna baseado parcialmente na norma PEAQ [151] e na medida PAQM [9]. A implementação é relativamente simples mas inclui os aspectos mais avançados daquelas medidas como a sobreposição não linear de padrões de excitação e curvas de excitação dependentes da intensidade. Um aspecto original desse trabalho é o estudo da sensibilidade dos padrões de excitação em relação a perturbações determinísticas ou aleatórias introduzidas no sinal. Deste modo, um sinal e uma medida ou estimativa do erro nele introduzido permitem prever a detectabilidade da distorção. Esta formulação tem bastante interesse num sistema retroadaptativo porque o sinal original não está disponível para comparação, mas conhece-se o sinal reconstruído e os intervalos de quantização que delimitam a gama de valores que o sinal original pode ocupar.

Estamos neste momento a projectar um modelo perceptual que permita calcular padrões de excitação no mesmo domínio e com a mesma resolução da representação espectral do sinal, de uma forma suave, sem artefactos introduzidos por conversões entre domínios. Os padrões de excitação serão comparados por um critério de detecção combinando detectabilidades parciais (eq. (6.12)). O objectivo é criar uma medida de distorção perceptual que possa ser usada para optimizar os quantizadores. Se essa medida puder ser expressa de forma compatível com a Equação (4.8), então o algoritmo de adaptação poderá ser uma implementação incremental do algoritmo de Lloyd generalizado.

Apêndice A

Ferramentas Desenvolvidas

Todas as ferramentas criadas ao longo deste trabalho foram desenvolvidas como funções em MATLAB, compatíveis com a versão 4.0 e versões posteriores. O ambiente de programação MATLAB foi escolhido por permitir o desenvolvimento rápido de protótipos; por dispor de uma vasta biblioteca de funções para cálculo numérico, processamento de sinal e visualização gráfica; e por facilitar a integração de programas escritos noutras linguagens. Para garantir a aplicabilidade das funções desenvolvidas em múltiplas situações, houve a preocupação de as escrever com a máxima generalidade possível e de as agrupar em *toolboxes* segundo a sua funcionalidade. Neste apêndice listamos parte dessas funções, apresentando o texto de ajuda de cada uma juntamente com alguns comentários. As várias secções reflectem a organização das *toolboxes*.

A.1 Bancos de Filtros e Transformadas

A.1.1 Transformadas em Blocos

Todas estas transformações foram implementadas com a FFT, usando os algoritmos apresentados em [103].

```
function x = dct(x)
%DCT    Discrete Cosine Transform.
%       DCT(X) returns the DCT of column vector X.  If X is a matrix,
%       the DCT is applied to each column.
%
%       Restrictions: The length of the transform must be a power of 2.
%
%       See also: IDCT, FFT, DCTIV.

function x = idct(x)
%IDCT   Inverse Discrete Cosine Transform.
%       IDCT(X) is the inverse DCT of vector or matrix X.
%
%       Restrictions: The length of the transform must be a power of 2.
%
%       See also DCT.
```

```
function x = dctiv(x)
%DCTIV  Discrete Cosine Transform, type IV.
%       DCTIV(X) returns the DCT-IV of column vector X.  If X is a matrix,
%       the DCT is applied to each column.
%
%       Note: The DCT-IV is its own inverse.
%
%       Restrictions: The length of the transform must be a power of 2.
%
%       See also FFT, DCT.
```

A.1.2 Transformadas Sobrepostas

Implementámos a ELT a partir do programa em C dado em [103] e integrámo-la como uma função invocável a partir do ambiente MATLAB.

```
function [y,z] = elt(tbl, x, z)
%ELT    Extended Lapped Transform.
%       [Y,Zf] = ELT(TBL,X,Zi), where:
%       TBL     Table of ELT coefficients, derived from the matrix of
%               butterfly angles using ELTTBL.  The dimension M and the
%               overlapping factor K of the ELT are deduced from TBL's
%               dimensions.
%       X (MxN) Input, each column is a block.
%       Y (MxN) Output, each column is a block.
%       Zi,Zf (KMx1)
%               Initial and final conditions of the K*M delay elements.
%
%       You can transform several blocks either separately or all at once.
%       That is:
%               z = ZZ;
%               [y1,z] = elt(tbl,x1,z);
%               [y2,z] = elt(tbl,x2,z);
%               y = [y1 y2];
%       produces the same results (in y and z) as:
%               z = ZZ;
%               y = elt(tbl, [x1 x2], z);
%
%       See also IELT, ELTTBL.
```

Além de um segmento do sinal de entrada `X`, devidamente agrupado em blocos de M amostras, a função recebe uma tabela `TBL` com os coeficientes para os filtros *lattice* e um vector `Zi` com o estado da estrutura, isto é, os valores armazenados nos seus atrasos. A função devolve a matriz `Y` com as saídas de cada banda e um vector `Zf` com o novo estado para ser usado na invocação seguinte.

A transformada inversa é implementada por uma função idêntica que usa exactamente a mesma tabela de coeficientes.

```
function [y,z] = ielt(tbl, x, z)
%IELT   Inverse Extended Lapped Transform.
%       [Y,Zf] = IELT(TBL,X,Zi) computes the inverse ELT of X.
%
%       See ELT for details on the arguments.
%
%       See also ELT, ELTTBL.
```

Escrevemos também uma rotina de optimização para auxiliar o projecto de ELTs segundo o método descrito na Subsecção 3.3.5.

```
function [h,theta] = eltdsgn(M,K,wSB)
%ELTDSGN ELT window design algorithm.
%       ELTDSGN(M,K,WSB) returns a window (low-pass prototype filter) with
%       stopband frequency WSB, for use in a M-band ELT with
%       overlapping factor K.  (Length L = 2*K*M.)
%
%       WSB should be greater than PI/(2*M).  Typically PI/M.
%
%       See also ELT.
```

A tabela de coeficientes para a ELT e IELT pode obter-se dos "ângulos" da estrutura *lattice* usando a função `ELTTBL`.

```
function tbl = elttbl(theta)
%ELTTBL Create table of multiplier coefficients from ELT butterfly angles.
%       TBL = ELTTBL(THETA) converts the M/2xK butterfly angles THETA into
%       3*M/2xK coefficients in TBL, to be used by ELT and IELT.
%
%       See also ELT, IELT.
```

O filtro protótipo ($h(n)$ na Equação 3.5) pode ser calculado a partir dos "ângulos" por `ELTA2W`.

```
function h = elta2w(theta)
%ELTA2W Convert ELT butterfly angles to the ELT window (prototype filter).
%       H = ELTA2W(THETA) returns the prototype impulse response H that
%       results from the ELT with angles THETA.
%       Each column of THETA contains the M/2 angles (outermost angle
%       first) for each of the K butterfly stages. (Column j contains
%       stage j-1.)
%
%       See also: ELTW2A.
```

A função `ELTW2A` faz a operação "inversa", convertendo um protótipo nos "ângulos" correspondentes.[1]

[1]Note-se, no entanto, que algumas sequências $h(n)$ não podem ser protótipos de uma ELT. Devido às características do algoritmo de cálculo, se se introduzir uma dessas sequências em `ELTW2A`, os "ângulos" resultantes podem corresponder a um protótipo muito diferente.

```
function theta = eltw2a(h,K)
%ELTW2A Convert ELT window to the ELT butterfly angles.
%       THETA = ELTW2A(H,K) tries to convert window H to the angle
%       matrix THETA.  K is the overlapping factor of the ELT.  If H does
%       not satisfy the Perfect Reconstruction conditions, the results may
%       be strange.
%
%       Each column of THETA contains the M/2 angles (outermost angle
%       first) for each of the K butterfly stages.  (Column j contains
%       stage j-1.)
%
%       See also: ELTA2W.
```

A.1.3 Sinais Multi-Resolução

Para representar sinais multibanda com múltiplas frequências de amostragem em MATLAB, cuja única estrutura de dados é a matriz, servimo-nos da representação dos sinais como vectores multi-resolução (Subsecção 3.1.3). Esta representação baseia-se em duas matrizes: uma transporta o sinal multi-resolução propriamente dito, um bloco em cada coluna; a segunda define o *formato* dos blocos, indicando os valores S_b (o número de amostras por bloco) de cada banda. Desenvolvemos rotinas para leitura e escrita de bandas num sinal multi-resolução que são usadas por todos os programas que lidam com estes sinais.

```
function [X,form] = setband(X, form)
%SETBAND Set up (or reset) globals for processing a multiresolution signal.
%       If X is a multiresolution signal (a matrix where each column is
%       a block of time samples from several different-width subbands) with
%       format FORM, SETBAND(X,FORM) sets up the appropriate global
%       variables for subsequent extraction (GETBAND) or writing (PUTBAND)
%       of subbands.
%
%       [X,FORM] = SETBAND returns the final multiresolution signal and its
%       format and clears the globals used.
%
%       Vector FORM describes the format of the blocks, i.e. the meaning
%       of each sample in X.
%
%       An error results if the format does not match the data in X.
%
%       For examples, see HLT, HLTTREE, and others.
%
%       See also: GETBAND, PUTBAND, HLT.

function putband(xb)
%PUTBAND Append subbands to a multiresolution signal.
%       PUTBAND(XB) appends the group of bands XB to a global store
%       (SETBAND_X) and updates its format.  The store must have been
```

```
%       previously set up using SETBAND and is physically the same that
%       is read by GETBAND but there should be no conflict since you
%       will overwrite only the data that you have already read.  In
%       case this happens, however, a warning is issued:
%       'Writing over unread data'.
%
%       In fact, XB is not immediately written into the store but passes
%       through a temporary buffer (PUTBAND_B) which groups bands with
%       the same rate.  This buffer will only be written into the store
%       when PUTBAND is called with a different rate XB.  You can call
%       PUTBAND([]) anytime to force the writing of the buffer.  SETBAND
%       does this before cleaning up.
%
%       An error results if the bands you are trying to write do not fit
%       inside the buffer or contain a different number of blocks.
%
%       See also: SETBAND, GETBAND, HLT.

function xb = getband(nb)
%GETBAND Extract subbands from a multiresolution signal.
%       XB = GETBAND(NB) returns the time samples pertaining to the next
%       NB bands (one band per row) of a multiresolution signal
%       previously set up with SETBAND(X,FORM), where X contains the
%       signal's samples and FORM is its format.
%
%       An error results if the NB bands you are trying to extract are
%       not all of the same rate or if reading past the end of the
%       block is attempted.
%
%       GETBAND, with no argument, returns TRUE (1) if the reading
%       position is at the end-of-block and FALSE (0) otherwise.
%
%       See also: SETBAND, PUTBAND, HLT.

function [O1,O2,O3]=eachband(X,form,init,repeat,I1,I2,I3)
%EACHBAND Execute commands for each band in a multiresolution signal.
%       [O1,O2,O3] = EACHBAND(X,FORM,'init','repeat',I1,I2,I3) evaluates
%       command 'init' once, and then repeatedly evaluates command 'repeat'
%       for each band in the input multiresolution signal X, with format
%       FORM.
%
%       Variables available inside the 'init' and 'repeat' commands:
%               BS      Block size.
%               NB      Number of bands.
%               B       Number of current band.
%               BAND    Contents (samples) of current band.
%               I1..I3  Additional (optional) input arguments.
```

```
%               O1..O3  Optional output arguments.
%
%       See also: SETBAND.

function indx=indxband(form,b,nb)
%INDXBAND Indices to extract bands from a multiresolution signal.
%       If X is a multiresolution signal with one block per column and
%       format FORM, INDXBAND(FORM, B) returns a matrix of indices to the
%       rows of X that contain the elements of band number B.  (Band
%       numbers start at 1.)
%
%       For example,
%               X(indxband(FORM,3),:)
%       would extract the third band from X.  (You may want to RESHAPE
%       the result to a single row or column!)
%
%       To index NB bands instead of one, use INDXBAND(FORM, B, NB).  The
%       bands must be contiguous and have the same rate (same number of
%       time samples).
%
%       See also: SETBAND, RESHAPE.
```

MRGRID calcula, a partir do formato de um sinal multi-resolução, as "fronteiras" no tempo e na frequência de cada uma das células de um bloco do sinal. Assume-se, naturalmente, que as fronteiras podem ser localizadas com precisão e que bandas de índice crescente correspondem a frequências crescentes.

```
function [t0,t1,f0,f1] = mrgrid(form)
%MRGRID Time-frequency boundaries of cells in a multiresolution signal.
%       [T0,T1,F0,F1]=MRGRID(FORM) returns time and frequency boundaries
%       of each cell in a multiresolution signal's time-frequency tiling.
%       T0, T1, F0 and F1 contain the time and frequency boundaries for
%       each cell (sample) in one block of the tiling, one per row.
%       FORM describes the multiresolution signal's format.
%
%       Naturally, this assumes that the multiresolution signal results from
%       some form of ideal filtering.
%
%       See also: MRFREQ.
```

MRFREQ informa apenas das frequências de transição entre bandas adjacentes. Esta informação também pode ser extraída de MRGRID.

```
function f = mrfreq(form)
%MRFREQ Nominal crossover frequencies in a multiresolution signal.
%       F=MRFREQ(FORM) returns the nominal crossover frequencies for a
%       (maximally decimated) filter bank, given its output format, FORM.
%
```

```
%       If the filter bank has B bands, F will be Bx1.
%
%       See also: MRGRID.
```

A rotina `TILE` permite a visualização de sinais multi-resolução na forma de mosaicos coloridos. Cada célula rectangular tem as coordenadas dadas por `MRGRID`.

```
function h = tile(X, form)
%TILE   Tile plot.  (A sort of PCOLOR for multiresolution matrices.)
%       TILE(X,FORM) produces a tile plot for multiresolution signal X,
%       using FORM as its format.  The vertical axis represents
%       normalized frequency (1.0 corresponds to half the sample rate).
%       The horizontal axis represents time (unit: original sampling
%       period = M * block-number).
%
%       If FORM is omitted, it is assumed to be [M 1] where
%       M = size(X,1) is the block size; i.e. X is assumed to contain
%       M equal-width, equal-rate bands.
%
%       See also: SETBAND, PCOLOR.
```

A.1.4 Estruturas Não Uniformes

O programa `HLT` implementa sistemas de decomposição não uniforme e é bastante geral, abrangendo tanto estruturas em árvore como estruturas SAM ou ainda estruturas híbridas.[2] O programa recebe um sinal multi-resolução `X` com um formato arbitrário `Xform` (podendo obviamente ser uniresolução) e devolve o sinal `Y`, com formato diferente, que resulta de várias transformações (directas ou inversas) produzidas em um ou mais estágios sucessivos de transformação. Um outro argumento de entrada é uma matriz de especificação `SPEC` que define a constituição de cada estágio, isto é, o tipo de transformação a aplicar a cada banda ou grupo de bandas do sinal multi-resolução que entra no estágio, juntamente com os parâmetros que caracterizam essa transformação: a dimensão M, um ponteiro para a tabela de coeficientes, o número de atrasos de compensação, e outros. De momento só estão disponíveis três tipos de transformadas: a ELT, que separa uma banda do sinal de entrada em M sub-bandas à saída; a IELT, que faz a operação inversa, juntando M bandas adjacentes numa banda só; e a transformação identidade, que passa M bandas inalteradas para a saída, com um eventual atraso de compensação. Outras transformadas ou bancos de filtros maximamente decimados poderão ser adicionados no futuro com alterações mínimas no programa. O estado interno das várias transformadas e dos atrasos de compensação é acessível através de um vector da mesma forma que na ELT. A generalidade deste método de definição de estruturas de transformação permite que o mesmo programa seja usado quer para o banco de análise, quer para o de síntese, bastando para isso alterar a matriz de especificação de forma sistemática.

```
function [X, z] = hlt(spec, tbl, X, form, z)
%HLT    A general Hybrid Lapped Transform implementation.
%       [Y,Zf] = HLT(SPEC,TBL,X,Xform,Zi) computes an HLT of multiresolution
```

[2] Aqui estendemos a abreviatura HLT para significar *Hybrid Lapped Transform* que inclui a *Hierarchical Lapped Transform* como um caso particular.

```
%       signal X with format Xform, and outputs the result in
%       multiresolution signal Y.
%
%       The HLT is defined by a specification in SPEC and a table of
%       coefficients in TBL.  See HLTDSGN and HLTTBL for more details on
%       these parameters.
%
%       Zi and Zf are the initial and final values of the delay elements.
%       The size of this vector is computed by HLTDSGN.
%
%       Restrictions: An error results if subbands with an odd number of
%       samples are detected at the output of an ELT or input of an IELT.
%       (This test is necessary to simplify the modulation operations.)
%       You can avoid this situation by forcing X to contain an even number
%       of columns (blocks).
%
%       See also: HLTDSGN, HLTTBL.
```

A rotina `HLTDSGN` permite calcular o número mínimo de atrasos de compensação que é necessário adicionar a cada transformada.[3] O projectista só tem que fornecer uma versão parcial da matriz de especificação, contendo apenas o tipo de cada transformada, o seu número de bandas M, e o factor de sobreposição K no caso de ELTs ou IELTs. Simultaneamente, este procedimento calcula a complexidade total da estrutura em termos de número de multiplicações, adições e palavras de memória requeridas, baseando-se em expressões da complexidade das ELTs dadas em [103, Section 5.4.4].

```
function [spec, ispec, iform, info, Sdelay] = hltdsgn(spec, form)
%HLTDSGN Check HLT specification, compute additional delays and other info.
%       HLTDSGN checks and completes an HLT specification and reports
%       performance information to aid design.  It also produces the
%       specification for the inverse HLT.
%
%       [SPEC,ISPEC,IFORM,INFO,Sdelay] = HLTDSGN(SPEC,FORM)
%
%       Input arguments:
%       SPEC    Initial (partial) specification of HLT. Each row
%               describes one transform with 3 numbers [TT M K]:
%               TT      Transform type (1: ELT, -1: IELT, 0: EYE).
%               M       Transform size (power-of-2 for ELTs, IELTs).
%               K       Overlapping factor for ELTs and IELTs. 0 for EYEs.
%               (You can put more than 3 numbers in each row, but they will
%               be ignored.  This is useful to check a previously designed
%               spec.)
%       FORM    Format of input to HLT.  Usually: [1 block_size].
```

[3] O cálculo do número de atrasos baseia-se no princípio da normalização do atraso em cada estágio, e na repartição equitativa das unidades de atraso pelos bancos de análise e de síntese. Em certas estruturas, este método pode conduzir a uma distribuição sub-óptima dos atrasos.

```
%
%       Output arguments:
%       SPEC    Complete spec of the HLT.  First 3 columns are the same,
%               last two contain:
%               tblptr  Pointer (index-1) to ELT table contained in TBL
%                       (see HLT).  This field is left untouched or set to
%                       zero.  Use HLTTBL to fill it in.
%               AD      Additional delay units introduced 'around' each
%                       transform (before ELTs, after IELTs or inside EYEs)
%                       to equalize delays in each stage.
%       ISPEC   Specification of the inverse HLT.
%       IFORM   Input format to inverse HLT = output format from this HLT.
%       INFO    Additional info on each transform T. Each row,
%               [D ram Nex add mult], contains:
%               D       Delay introduced by T.
%               ram     Necessary RAM for T, i.e. Z-buffer size.
%               Nex     Number of executions of T per block.
%               add     Number of additions per block for T.
%               mult    Number of multiplications per block for T.
%       Sdelay  Total delay of each stage. These are 'normalized' delays:
%               the time unit is the 'original' sampling period.
%               sum(Sdelay) is the total delay of the HLT, i.e. half of
%               reconstruction delay of the HLT/inverse HLT cascade.
%
%       SPEC, INFO and Sdelay are printed out in a tabular format, as
%       well as summary info on total delay, adds and mults.
%
%       See also: HLT.
```

O programa `HLTTBL` facilita a compilação das tabelas de coeficientes de ELTs projectadas previamente, e completa a matriz de especificação com ponteiros para as tabelas apropriadas.

```
%HLTTBL Interactive tool to build tables of coefficients for HLTs.
%       This script helps in compiling the table of coefficients (TBL) for an
%       HLT.  This consists of concatenating ELT TBLs together, and setting
%       the table pointers in SPEC (4th column) accordingly.
%
%       Before running, make sure that every ELT angle matrix that is needed
%       is on the current workspace.  Then, just call HLTTBL and answer the
%       questions.  (HLTTBL will call ELTTBL to convert angles to coeffs.)
%
%       Each time you enter a matrix, HLTTBL checks whether it was entered
%       before, and avoids repeating it by copying just the pointer.
%
%       See also: HLT.
```

`HLTTREE` produz diagramas de estruturas de decomposição como os das Figuras 3.2 e 3.4.

```
function hlttree(spec, form)
```

```
%HLTTREE Plot a 'tree' representation of an HLT.
%       HLTTREE(SPEC, FORM) plots a tree depicting the specified HLT.  The
%       signal flows from left to right, ELTs are represented by a 1-to-M
%       branch, IELTs by an M-to-1 junction, and direct connections represent
%       EYEs.  (Actually, this is not a 'tree' in the strict sense since
%       IELTs have several parent nodes.)
%
%       See also: HLT.
```

A.1.5 Análise de Bancos de Filtros

A rotina `FBIR` permite a medição das respostas impulsionais de qualquer banco de filtros com decimação inteira (Figura 3.1). Esta função aborda o banco de filtros a medir como uma "caixa preta" com uma entrada e diversas saídas; simula "ensaios" colocando impulsos na entrada e regista as saídas de forma a reconstruir as respostas impulsionais antes da decimação.

```
function [h, ib]=fbir(fb_expr, iform, N, P1, P2, P3, P4, P5, P6, P7, P8, P9)
%FBIR   Filter bank impulse responses.
%       Any M-band maximally-decimated nonuniform decomposition filter
%       bank with integer decimation factors can be put in the form:
%
%                       +-------+   y0(n)   +------+
%       x(n) ---->+---->| H0(n) |---------->| R0:1 |----> y0(m0)
%                 |     +-------+           +------+
%                 |
%                 |     +-------+   y1(n)   +------+
%                 +---->| H1(n) |---------->| R1:1 |----> y1(m1)
%                 |     +-------+           +------+
%                 |                 . . .
%
%       where {H0(n), H1(n), ...} are the filter impulse responses and
%       {R0, R1, ...} are the decimation factors.
%
%       If 'y=filtbank(x);' implements such a filter bank, then
%       H = FBIR('y=filtbank(x);', IFORM, N) returns the impulse responses
%       of each filter, one per row.
%
%       The first parameter to FBIR is a string to be EVAL'ed repeatedly
%       and must contain expressions that transform input x into output y.
%       All internal delay units should be initially set to 0.  IFORM is
%       the format of the multiresolution signal in y.  The block size BS is
%       computed from IFORM and x is assumed to have format [1 BS], i.e. a
%       full-band signal with BS time samples per block.  N is the number of
%       blocks that you expect the wider impulse response to span.  If L is
%       the length of the wider IR, then you should set N = ceil(L/BS).
%
%       H = FBIR('y=filtbank(x,P1,P2,..,P9)', IFORM, N, P1, P2, ..., P9)
```

```
%       allows you to pass additional arguments to 'filtbank'.
%
%       See also: SETBAND, HLT.
```

A.2 Quantização

A.2.1 Quantização Genérica

Implementámos rotinas de quantização/desquantização escalar genérica; o utilizador tem que fornecer tabelas com os níveis de decisão e de reconstrução que pretende.

```
function u = quant(xk, x)
%QUANT  Generic quantizer.
%       U = QUANT(Xk, X) quantizes matrix X according to the table of
%       decision levels Xk.  This table must be sorted and Xk(1) should
%       be lower than any admissible input (-Inf is a good choice).
%
%       The output symbols U are integers such that:
%               Xk(U) <= X < Xk(U+1).
%       (If X < Xk(1), then U returns 0.)
%       In fact, QUANT does nothing more than a search in a sorted table.
%
%       See also: IQUANT.
```

```
function v = iquant(yk, v)
%IQUANT Generic inverse quantizer.
%       Y = IQUANT(Yk, V) takes indices (symbols) V and "reconstructs"
%       signal Y, according to the table of reconstruction levels Yk.
%
%       See also: QUANT.
```

A função `QBEST` calcula os melhores níveis de decisão para um dado vector de níveis de reconstrução.[4]

```
function xk = qbest(yk)
%QBEST  Best quantizer decision levels for given reconstruction levels.
%       Xk = QBEST(Yk) returns the best decision levels for a generic
%       quantizer with reconstruction levels Yk.  According to Max, the
%       best is halfway between the Yk levels (if the distortion function
%       satisfies certain conditions).
%
%       See also: QUANT, IQUANT.
```

[4]Melhores no sentido da minimização do valor esperado de uma medida de distorção dada por uma função par, crescente no ramo positivo, como por exemplo, $f(x - y_k) = (x - y_k)^2$ [105].

A.2.2 Curvas de Compressão/Expansão

```
function y = alaw(A,x)
%ALAW   The A-law compression characteristic.
%       y = ALAW(A, x) 'compresses' values x using the A-law.
%       This is normalized: ALAW(A, [0 1]) == [0 1].
%
%       See also: MULAW.

function y = mulaw(mu,x)
%MULAW  The mu-law compression characteristic.
%       y = MULAW(mu, x) 'compresses' values x using the mu-law.
%       This is normalized: MULAW(mu, [0 1]) == [0 1].
%
%       See also: IMULAW.

function x = imulaw(mu,y)
%IMULAW The mu-law expansion characteristic.
%       x = IMULAW(mu, y) 'expands' values y using the mu-law.
%       This is the inverse function of MULAW.
%
%       See also: MULAW.
```

A.2.3 Análise de Quantizadores

`QPOWER` calcula a potência de ruído (valor esperado do erro quadrático) num sistema de quantização genérico, dada a função densidade de probabilidade da entrada.

```
function var = qpower(pdf,xk,yk,tol,trace)
%QPOWER Computes approximation to quantizer noise power.
%       QPOWER('pdf', Xk, Yk) computes the noise power of a quantizer
%       defined by decision levels Xk and output levels Yk.  'pdf' is the
%       name of a function that returns the probability density of the
%       quantizer input.
%
%       If Yk has got N elements, then Xk must have N+1.  The last value
%       in Xk should theoretically be +Inf, but QUAD will not take that,
%       so a reasonably large value must be used instead.  The same
%       reasoning applies to the first value unless the pdf and quantizer
%       are symmetrical, in which case you can just use the positive range
%       and double the result.  In this case, Xk(1)=0.
%
%       Also: QPOWER('pdf', Xk, Yk, tol, trace) where tol and trace are
%       passed to QUAD.
%
%       See also: QENTROPY.
```

`QPDF` calcula a distribuição de probabilidades dos símbolos de saída de um quantizador genérico, dada a função densidade de probabilidade da entrada.

```
function P = qpdf(pdf, xk)
%QPDF   Quantizer output probability function, for a given input pdf.
%       P = QPDF('pdf', Xk) computes the probability of occurrence of
%       each output level in a quantizer, given the input pdf.  The
%       quantizer is specified by a set of decision levels in Xk.  The
%       first and last levels in Xk are subject to the same restrictions
%       as in QPOWER.
%
%       See also: QPOWER, QENTROPY.
```

QENTROPY avalia a entropia do quantizador a partir da distribuição dos seus níveis de saída. Para isso, usa QPDF e ENTROPY.

```
function H = qentropy(pdf, xk)
%QENTROPY Approximate entropy for a given quantizer and input pdf.
%       H = QENTROPY('pdf', Xk).
%       The quantizer is specified by a set of decision levels in Xk.
%       The first and last levels in Xk are subject to the same
%       restrictions as in QPOWER.
%
%       See also: QPOWER, QPDF.

function H=entropy(n)
%ENTROPY Compute entropy of a discrete source.
%       Given a vector or matrix N with the probability of occurrence of
%       each symbol in the source, H=ENTROPY(N) returns the entropy of
%       that source (in bits per symbol).
%
%       For convenience, N is scaled internally by SUM(SUM(N)) so that,
%       instead of the probabilities (or frequencies) of each symbol, N
%       may contain the actual occurrence counts. (You can get occurrence
%       counts from a stream of symbols by using HIST.)
%
%       See also: HIST.
```

Fornecem-se duas funções densidade de probabilidade típicas para utilização nas rotinas anteriores: distribuição gaussiana e distribuição laplaciana.

```
function p = gauss(x, mean, stddev)
%GAUSS  Gaussian (Normal) probability density function.
%       GAUSS(X) evaluates the zero-mean, unit-variance Gaussian pdf at
%       the points X.
%
%       GAUSS(X,mean,stddev) gives the general Gaussian pdf.
%
%       See also: LAPLACE.

function p = laplace(x, mean, stddev)
```

```
%LAPLACE Laplacian (two-sided exponential) probability density function.
%       LAPLACE(X) evaluates the zero-mean, unit-variance Laplacian pdf at
%       the points X.
%
%       LAPLACE(X,mean,stddev) gives the general Laplacian pdf.
%
%       See also: GAUSS.
```

A.3 Codificação Aritmética

As rotinas de codificação e descodificação aritmética foram implementadas em C com programas adaptados a partir dos apresentados em [162], e foram integradas no ambiente de simulação desenvolvido.

```
function [bits,statef,cumFreqf]=arithenc(method,symbols,statei,indx,cumFreq)
%ARITHENC Arithmetic encoding.
%       [BITS,STATEf,CUMFREQf] = ARITHENC(METHOD,SYMBOLS,STATEi,INDX,CUMFREQ)
%       encodes a stream of SYMBOLS into a stream of BITS using the model
%       defined by CUMFREQ.  If SYMBOLS=[], the encoding is terminated and
%       the last bits are flushed out.  BITS is a vector of ones and zeros.
%
%       CUMFREQ should contain, in a row, the cumulative frequency count for
%       each symbol in the alphabet.  We can use ARITHCUM to produce the
%       appropriate CUMFREQ matrix from frequency counts.
%       The elements of SYMBOLS represent the stream to encode and should be
%       integers between 1 and the length of the alphabet
%       (CUMFREQ's width - 2).
%       If CUMFREQ has M rows, each row represents a different model, and
%       each symbol (taken columnwise) will be encoded according to the
%       model pointed to by the current INDX entry.
%       INDX contains the model "switching sequence".  For example, if
%       INDX=[1 1 2 3], that means that the first and second symbols will be
%       encoded using the model in the 1st row of CUMFREQ, the third symbol
%       will be encoded using the 2nd row, and the fourth symbol will use the
%       3rd row.  The following symbols start repeating the sequence.
%
%       METHOD=0 selects the fixed model method.  CUMFREQ is not altered.
%       METHOD=1 selects the adaptive model method.  Each encoded symbol
%       increases its probability in the model.
%       METHOD=2 selects the adaptive symmetrical model method.  Both the
%       probability of the symbol and of its "symmetric" are increased.
%
%       STATEi and STATEf are the initial and final 'state' of the
%       encoder, respectively.  They contain 4 integer parameters
%       [Low, High, BitsToFollow, IndxIndx] that must be passed between
%       calls to ARITHENC.  STATEi=[] (re)sets the state to initial values
%       [0 65535 0 0].
```

```
%
%       See also: ARITHDEC.


function [symbols,statef,cumFreqf]=arithdec(method,bits,statei,indx,cumFreq)
%ARITHDEC Arithmetic decoding.
%       [SYMBOLS,STATEf,CUMFREQf] = ARITHDEC(METHOD,BITS,STATEi,INDX,CUMFREQ)
%       decodes a stream of BITS into a stream of SYMBOLS using the model
%       defined by CUMFREQ.
%
%       METHOD, INDX, CUMFREQ, SYMBOLS and BITS have the same meaning as in
%       ARITHENC.
%
%       STATEi and STATEf are the initial and final 'state' of the
%       decoder, respectively.  They contain 4 integer parameters
%       [Low, High, Value IndxIndx] that must be passed between calls
%       to ARITHDEC.  STATEi=[] (re)sets the state to sane initial
%       values: [0 65535 <first 16 bits in stream> 0].
%
%       See also: ARITHENC.
```

As tabelas de frequências acumuladas usadas nas rotinas anteriores podem ser construídas a partir das frequências simples, aplicando `ARITHCUM`.

```
function cumFreq = arithcum(freq, MAX)
%ARITHCUM Generate cumulative frequency counts to use in ARITHENC.
%       Given a row vector (FREQ) of frequency counts (occurrence
%       probabilities) of each symbol in an alphabet, ARITHCUM(FREQ)
%       returns a valid CUMFREQ vector appropriate for ARITHENC and
%       ARITHDEC.  If FREQ is a matrix, each row will produce a row
%       in CUMFREQ.
%
%       FREQ is ROUNDed to the nearest integer.  It must contain only
%       positive integers and the sum of each row must be less than
%       16384.  An error results if any of these conditions fails.
%
%       If wanted, a maximum of less than 16383 may be specified with
%       ARITHCUM(FREQ, MAX).  The maximum value is used in the adaptive
%       models (METHOD~=0), and a smaller value leads to faster tracking
%       of changing input statistics.  MAX may be a single number or a
%       column vector with a different value for each row of FREQ.
%
%       Hint: use HIST to generate rows for FREQ from a real symbol
%       stream.
%
%       See also: ARITHENC, ARITHDEC.
```

A.4 Modelo Psicoacústico

A função ZWICKER implementa a Equação 6.2.

```
function x = zwicker(f)
%ZWICKER Transform Hz to Bark, using Zwicker and Fastl's formula.
%       ZWICKER(f) converts frequencies f (in Hz) to Bark.
%
%       See also: HZ2BARK.
```

SPREAD implementa a Equação 6.3 com uma ligeira compressão de 5% no eixo dos xx, como usado no Modelo Psicoacústico II do MPEG.

```
function y = spread(dz)
%SPREAD Model the auditory system spreading function.
%       SPREAD(dz) returns the spreading function value at a band dz barks
%       to the right of the signal band.
%       Argument dz can be a matrix, in which case the result is a matrix
%       of the same dimensions, with the spread values for each element.
%
%       See also: ZWICKER.
```

O modelo psicoacústico retroadaptativo descrito na Secção 7.1.4 é calculado pela função PSYAQB.

```
function [var,Yout1,Yout2]=psyaqb(var,Yin,expr,P1,P2,P3,P4,P5)
%PSYAQB Backward-adaptive psychoacoustic model for an HLT-based codec.
%       In a backward-adaptive perceptual audio coder the psychoacoustic
%       model is evaluated in alternation with the quantization and/or
%       dequantization process.
%       [Vf,O1,O2]=PSYAQB(Vi,Y,'expr',P1,...,P5) estimates the masking
%       threshold for multiresolution signal Y, using 'expr' to quantize
%       and/or dequantize Y.  The computation is done incrementally, one
%       segment at a time.  (The current segment of a multiresolution signal
%       contains the set of contemporary samples - one per band, not
%       necessarily all bands - that can be processed together because
%       masking information is already available from previous samples.)
%
%       The following variables are available inside the 'expr' command:
%               IN      The current segment of Y samples available for use.
%               MASK    The masking threshold (in energy units) previously
%                       computed for this segment.
%               P1..P5  Additional (optional) input arguments.
%               REC     The current segment of reconstructed (dequantized)
%                       samples to feed to the psychoacoustic model.
%               OUT1 &
%               OUT2    Current segments of optional output arguments O1&O2.
%       See BAPACENC and BAPACDEC for examples of 'expr'.
%
```

```
%       Vi and Vf are the initial and final values of the delay elements
%       (one per band) used in time-spreading operation.  Vi should be set
%       to zeros in the beginning.
%
%       The format of the multiresolution signals Y, O1 and O2 is loaded
%       with other signal and psychoacoustic information from a file
%       previously created by PSYSETUP.
%
%       See also: PSYSETUP, BAPACENC, BAPACDEC.

function psysetup(iform,FS)
%PSYSETUP Setup variables to be used in psychoacoustic models.
%       PSYSETUP(IFORM,FS) establishes some variables to be used by
%       PSYAQB.  IFORM is the format of the output of the analysis HLT
%       which is fed into PSYAQB.  FS is the sampling rate.
%
%       Besides FS and IFORM, this routine sets up the tables for the
%       threshold of hearing, the masking index, the time-smearing gains,
%       the frequency-spreading convolution matrix, and timing information
%       to determine the segment slicing of the signals.
%
%       The parameters and tables produced are stored in 'psysetup.mat'
%       so that a LOAD in PSYAQB restores everything.
%
%       See also: PSYAQB, HLT.
```

A.5 Codificação Perceptual de Áudio

As rotinas seguintes simulam o sistema de codificação perceptual de áudio com adaptação para trás descrito no Capítulo 7.

```
function bpb = bapacenc(sndin,midfile,bitout,level)
%BAPACENC Backward-Adaptive Perceptual Audio encoder.
%       BAPACENC('in.snd', 'bitfile', LEVEL) compresses the input 'in.snd'
%       into a bit stream in 'bitfile', with a given compression LEVEL.
%       LEVEL=1 is the default, and should give transparent quality.
%       Higher values should give more compression and lower quality.
%
%       See also: BAPACDEC, BAPAC.

function bapacdec(bitin,midfile,sndout,level)
%BAPACDEC Backward-Adaptive Perceptual Audio decoder.
%       BAPACDEC('bitfile', 'out.snd', LEVEL) decompresses the input
%       bit stream in 'bitfile' into a sound file 'out.snd'.  LEVEL is the
%       compression level used when creating 'bitfile'.  This could/should
%       come as header information in the bit stream, but I did not bother
%       to do that.
```

```
%
%       See also: BAPACENC, BAPAC.

function bpb = bapac(sndin,midfile,sndout,level)
%BAPAC  Backward-Adaptive Perceptual Audio Coding system.
%       BpB = BAPAC('in.snd', 'mid.raw', 'out.snd', LEVEL) encodes sound
%       file 'in.snd' and decodes it into 'out.snd'.  The quantized samples
%       are stored one symbol per byte in 'mid.raw' (before arithmetic
%       coding). LEVEL is the compression/quality level of the system.
%       BpB returns the number of bits per block along the signal.
%
%       See also: BAPACENC, BAPACDEC.
```

Bibliografia

[1] Ashish Aggarwal, Shankar L. Regunathan, and Kenneth Rose. Near-optimal selection of encoding parameters for audio coding. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001. IEEE. 7.3

[2] Makoto Akune, Robert M. Heddle, and Kenzo Akagiri. Super Bit Mapping: Psychoacoustically optimized digital recording. In *93rd Convention of the Audio Engineering Society* [5]. Preprint 3371. 2.3.2

[3] Jont B. Allen. Cochlear modeling. *IEEE Acoustics, Speech, and Signal Processing Magazine*, pages 3–29, January 1985. 6.2

[4] Bishnu S. Atal and Manfred R. Schroeder. Predictive coding of speech signals and subjective error criteria. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27(3):247–254, June 1979. 2.3.2

[5] Audio Engineering Society. *93rd Convention of the Audio Engineering Society*, San Francisco, CA, October 1–4 1992. 2, 17, 146, 156

[6] Audio Engineering Society. *100th Convention of the Audio Engineering Society*, Copenhagen, May 1996. 35, 150

[7] Frank Baumgarte. Improved audio coding using a psychoacoustic model based on a cochlear filter bank. *IEEE Transactions on Speech and Audio Processing*, 10(7):495–503, October 2002. 6.4.1

[8] Richard J. Beaton, John G. Beerends, Michael Keyhl, and William C. Treurniet. Objective perceptual measurement of audio quality. In Gilchrist and Grewin [52], pages 126–152. 6.1

[9] John G. Beerends and Jan A. Stemerdink. A perceptual audio quality measure based on a psychoacoustic sound representation. *Journal of the Audio Engineering Society*, 40(12):963–978, December 1992. 6.1, 6.4.1, 6.5, 8

[10] John G. Beerends and Jan A. Stemerdink. A perceptual speech quality measure based on a psychoacoustic sound representation. *Journal of the Audio Engineering Society*, 42(3):115–123, March 1994. 6.5

[11] David M. Bell and John N. Gowdy. Power spectral estimation via nonlinear frequency warping. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-26(5):436–441, October 1978. 2.3.3

[12] Timothy C. Bell, John G. Cleary, and Ian H. Witten. *Text Compression*. Prentice-Hall, 1990. 2.3.1, 5.5

[13] Timothy C. Bell and Ian H. Witten. The relationship between greedy parsing and symbolwise text compression. *Journal of the Association for Computing Machinery*, 41(4):708–724, 1994. 5.6

[14] Toby Berger. Rate distortion theory for sources with abstract alphabets and memory. *Inform. Contr.*, 13:254–273, September 1968. 1.1.3

[15] Per Bodin. On wavelets and orthonormal bases in system identification. Master's thesis, Automatic Control Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, May 1995. 4

[16] Marina Bosi, Karlheinz Brandenburg, Schuyler Quackenbush, Louis Fielder, Kenzo Akagiri, Hendrik Fuchs, Martin Dietz, Jürgen Herre, Grant Davidson, and Yoshiaki Oikawa. ISO/IEC MPEG-2 advanced audio coding. *Journal of the Audio Engineering Society*, 45(10):789–813, October 1997. 2.2.4

[17] Marina Bosi and Grant Davidson. High quality, low-rate audio transform coding for transmission and multimedia applications. In *93rd Convention of the Audio Engineering Society* [5]. Preprint 3365. 6.3.3

[18] Colin Boyd, John G. Cleary, Sean A. Irvine, Ingrid Rinsma-Melchert, and Ian H. Witten. Integrating error detection into arithmetic coding. *IEEE Transactions on Communications*, 45(1):1–3, January 1997. 5.6

[19] Karlheinz Brandenburg. OCF—a new coding algorithm for high quality sound signals. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 12, pages 141–144. IEEE, April 1987. 2.2.4

[20] Karlheinz Brandenburg. High quality sound coding at 2.5 bit/sample. In *84th Convention of the Audio Engineering Society*, Paris, March 1988. Audio Engineering Society. Preprint 2582. 2.2.4

[21] Karlheinz Brandenburg. MP3 and AAC explained. In *AES 17th International Conference: High-Quality Audio Coding*, Florence, Italy, September 1999. Audio Engineering Society. 2.2.4

[22] Karlheinz Brandenburg and James D. Johnston. Second generation perceptual audio coding: The hybrid coder. In *88th Convention of the Audio Engineering Society*, Montreux, March 1990. Audio Engineering Society. Preprint 2937. 2.2.4, 6.4.2

[23] Søren Buus, Edwin Shorer, Mary Florentine, and Eberhard Zwicker. Decision rules in detection of simple and complex tones. *Journal of the Acoustical Society of America*, 80(6):1646–1657, December 1986. 6.4.3, 6.4.3

[24] Philippe Cassereau. A new class of optimal unitary transforms for image processing. Master's thesis, Mass. Inst. Tech., Cambridge, MA, May 1985. 3.1

[25] Jim Chou and Kannan Ramchandran. Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission. *IEEE Journal on Selected Areas in Communications*, 18(6):861–867, June 2000. 5.6

[26] Philip A. Chou, Tom Lookabaugh, and Robert M. Gray. Entropy-constrained vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(1), January 1989. 4.3, 4.3

[27] Catherine Colomes, Michel Lever, Jean-Bertrand Rault, Yves-François Dehery, and Gérard Faucon. A perceptual model applied to audio bit-rate reduction. *Journal of the Audio Engineering Society*, 43(4):233–240, April 1995. 6.1

[28] Richard V. Cox. The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34:1090–1096, October 1986. 3.3.3, 3.4.2

[29] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1983. 3.1, 3.1.1, 3.3.1, 3.3.5

[30] Grant A. Davidson, Louis D. Fielder, and Brian D. Link. Parametric bit allocation in a perceptual audio coder. In *97th Convention of the Audio Engineering Society*. Audio Engineering Society, November 1994. (Also on WWW). 2.2.4, 7.3

[31] Ricky Der, Peter Kabal, and Wai-Yip Chan. Towards a new perceptual coding paradigm for audio signals. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* [72], pages 457–460. 6.4

[32] Martin Dietz and Stefan Meltzer. CT-aacPlus—a state-of-the-art audio coding scheme. *EBU Technical Review*, July 2002. 2.4

[33] The evolution of Dolby film sound. Web document, WWW, 2004. 2.2.4

[34] Bernd Edler. Aliasing reduction in sub-bands of cascaded filter banks with decimation. *Electronics Letters*, 28(12):1104–1106, 4th June 1992. 3.4.1

[35] Bernd Edler, Heiko Purnhagen, and Charalampos Ferekidis. ASAC - analysis/synthesis audio codec for very low bit rates. In *100th Convention of the Audio Engineering Society* [6]. Preprint 4179. 2.4

[36] Elias, circa 1960. Unpublished work, described by several authors including [81, 133]. 5.4

[37] George F. Elmasry. Joint lossless-source and channel coding using automatic repeat request. *IEEE Transactions on Communications*, 47(7):953–955, July 1999. 5.6

[38] Markus Erne. Pre-echo distortion. In Markus Erne, editor, *Perceptual Audio Coders: What to Listen For*. Audio Engineering Society, New York, 2001. CD-ROM. 7

[39] European Broadcasting Union, Brussels. *Sound Quality Assessment Material: Recordings for Subjective Tests*, April 1988. 7.1.5

[40] Peter M. Fenwick. A new data structure for cumulative frequency tables. *Software—Practice and Experience*, 24(3):327–336, March 1994. 5.5

[41] Aníbal João Sousa Ferreira. Codificação perceptual de Áudio digital estereofónico. Master's thesis, Faculdade de Engenharia da Universidade do Porto, Porto, January 1992. 6.2, 6.3.3, 6.4.2, 6.4.2

[42] Aníbal João Sousa Ferreira. Tonality detection in perceptual coding of audio. In *98th Convention of the Audio Engineering Society*, Paris, February 1995. Audio Engineering Society. Preprint 3947. 6.4.2

[43] Aníbal João Sousa Ferreira. Combined spectral envelope normalization and subtraction of sinusoidal components in the ODFT and MDCT frequency domains. In *Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 51–54, New Paltz, NY, October 21–24 2001. IEEE. 2.4

[44] Aníbal João Sousa Ferreira. Perceptual coding using sinusoidal modeling in the MDCT domain. In *112th Convention of the Audio Engineering Society*, Munich, May 10–13 2002. Audio Engineering Society. Preprint 5569. 2.4

[45] Louis D. Fielder, Marina Bosi, Grant Davidson, Mark Davis, Craig Todd, and Steve Vernon. AC-2 and AC-3: Low-complexity transform-based audio coding. In Gilchrist and Grewin [52], pages 54–72. 2.2.4

[46] Sadaoki Furui and M. Mohan Sondhi, editors. *Advances in Speech Signal Processing*. Marcel Dekker, Inc., New York, 1991. 50, 84

[47] Robert G. Gallager. Variations on a theme by Huffman. *IEEE Transactions on Information Theory*, IT-24(6):668–674, November 1978. 5.3, 5.5

[48] William R. Gardner and Bhaskar D. Rao. Theoretical analysis of the high-rate vector quantization of LPC parameters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 3(5):367–381, September 1995. 4.2

[49] Allen Gersho and Vladimir Cuperman. Vector quantization: A pattern-matching technique for speech coding. *IEEE Communications Magazine*, 21:15–21, December 1983. 4.4.4

[50] Allen Gersho, Shihua Wang, and Kenneth Zeger. Vector quantization techniques in speech coding. In Furui and Sondhi [46], chapter 2. 4.4.4

[51] James H. Gilchrist. The short-time behavior of a frequency-warping power spectral estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28(2):176–183, April 1980. 2.3.3

[52] Neil Gilchrist and Christer Grewin, editors. *Collected Papers on Digital Audio Bit-Rate Reduction*. Audio Engineering Society, New York, 1996. 8, 45, 112

[53] Herbert Gish and John N. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14(5):676–683, September 1968. 4.5

[54] Solomon W. Golomb. Run-length encodings. *IEEE Transactions on Information Theory*, pages 399–401, July 1966. (Correspondence). 5.6

[55] David J. Goodman and Roger M. Wilkinson. A robust adaptive quantizer. *IEEE Transactions on Communications*, 23(11):1362–1365, November 1975. 7.2.1

[56] R. A. Gopinath and C. S. Burrus. Theory of modulated filter banks and modulated wavelet tight frames. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 169–172, Minneapolis, MN, April 27–30 1993. IEEE. 3.3.5

[57] Robert M. Gray. Vector quantization. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 1:4–29, April 1984. 4.4.4

[58] Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, October 1998. 4, 3, 4.4.3

[59] Mauro Guazzo. A general minimum-redundancy source-coding algorithm. *IEEE Transactions on Information Theory*, 26(1):15–25, January 1980. 5.4, 5.4.3

[60] Joseph L. Hall. Asymmetry of masking revisited: Generalization of masker and probe bandwidth. *Journal of the Acoustical Society of America*, 101(2):1023–1033, February 1997. 6.3.3, 6.5

[61] Richard Wesley Hamming. *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1980. 1.1.2

[62] Fredric J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, January 1978. 6.4.1

[63] William M. Hartmann. How we localize sound. *Physics Today*, pages 24–29, November 1999. 6.2.1

[64] William Morris Hartmann. The physical description of signals. In Moore [111], chapter 1. 6.1

[65] A. J. M. Houtsma, T. D. Rossing, and W. M. Wagenaars. Auditory demonstrations. Accompanying a CD prepared at the Institute for Perception Research (IPO), Eindhoven, The Netherlands, and suported by Northern Illinois University (NIU) and the Acoustical Society of America (ASA), 1987. 6.3.3

[66] Paul G. Howard and Jeffrey Scott Vitter. Arithmetic coding for data compression. *Proceedings of the IEEE*, 82(6):857–865, June 1994. 5.4.3

[67] David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E.*, 40:1098–1101, September 1952. Also in [140, pp. 47–50]. 5.3

[68] Larry E. Humes and Walt Jesteadt. Models of the additivity of masking. *Journal of the Acoustical Society of America*, 85(3):1285–1294, March 1989. 6.3.3

[69] Linh Huynh and Alistair Moffat. A probability-ratio approach to approximate binary arithmetic coding. *IEEE Transactions on Information Theory*, 43(5):1658–1662, September 1997. 5.4.3, 5.6

[70] Aki Härmä, Matti Karjalainen, Lauri Avioja, Vesa Välimäki, Unto K. Laine, and Jyri Huopaniemi. Frequency-warped signal processing for audio applications. *Journal of the Audio Engineering Society*, 48(11):1011–1031, November 2000. 2.3.3

[71] Aki Härmä, Unto K. Laine, and Matti Karjalainen. An experimental audio codec based on warped linear prediction of complex valued signals. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 323–326, Munich, April 1997. IEEE. 2.3.3

[72] IEEE. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, April 2003. 31

[73] International Telecommunications Union, Geneva, Switzerland. *Subjective Assessment of Sound Quality*, 1990. Rec. BS.562-3. 7.1.5

[74] International Telecommunications Union, Geneva, Switzerland. *Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems*, 1997. Rec. BS.1116-1. 7.1.5

[75] International Telecommunications Union, Geneva, Switzerland. *Method for Objective Measurments of Perceived Audio Quality*, 1998. Rec. BS.1387. 6.1, 4, 6.4.3

[76] Toshio Irino and Masashi Unoki. A time-varying, analysis/synthesis auditory filterbank using the gammachirp. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3653–3656, Seattle, Washington, May 12–15 1998. IEEE. 6.4.1

[77] ISO/IEC JTC1/SC29/WG11 (MPEG). *ISO/IEC CD 11172: Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*, 1992. 2.2.4, 12, 6.1, 6.4.2, 7.1.4

[78] ISO/IEC JTC1/SC29/WG11 (MPEG). *ISO/IEC CD 13818: Generic Coding of Moving Pictures and Associated Audio Information*, 1994. 2.2.4

[79] V. K. Jain and R. E. Crochiere. Quadrature mirror filter design in the time domain. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32:353–361, April 1984. 3.3.1

[80] Nuggehally S. Jayant and Peter Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs, N.J., 1984. 1.1.3, 5, 2.3.1, 2.3.2, 3.2, 4.4.1, 4

[81] Frederick Jelinek. Buffer overflow in variable length coding of fixed rate sources. *IEEE Transactions on Information Theory*, 14(3):490–501, May 1968. 36

[82] James D. Johnston. A filter family designed for use in quadrature mirror filter banks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 291–294, Denver, CO, April 1980. IEEE. 3.3.1, 3.4.2

[83] James D. Johnston. Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6(2):314–323, February 1988. 2.2.4, 6.1, 6.4.2, 6.4.2, 7.1.4, 8

[84] James D. Johnston and Karlheinz Brandenburg. Wideband coding—perceptual considerations for speech and music. In Furui and Sondhi [46], chapter 4. 2.1.2, 3.2

[85] Christopher B. Jones. An efficient coding system for long source sequences. *IEEE Transactions on Information Theory*, 27(3):280–291, May 1981. 5.4, 5.4.3

[86] Ioannis Katsavounidis, C.-C. Jay Kuo, and Zhen Zhang. A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters*, 1(10):144–146, October 1994. 4.3

[87] R. David Koilpillai and P. P. Vaidyanathan. Cosine-modulated FIR filter banks satisfying perfect-reconstruction. *IEEE Transactions on Signal Processing*, 40:770–783, April 1992. 3.3.5

[88] Shiann-Rong Kuang, Jer-Min Jou, and Yuh-Lin Chen. The design of an adaptive online binary arithmetic-coding chip. *IEEE Transactions on Circuits and Systems I*, 45(7):693–706, July 1998. 5.4.3, 5.6

[89] Unto K. Laine, Matti Karjalainen, and Toomas Altosaar. Warped linear prediction (WLP) in speech and audio processing. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 349–352, Adelaide, April 19–22 1994. IEEE. 2.3.3

[90] Glen G. Langdon, Jr. and Jorma Rissanen. Compression of black-white images with arithmetic coding. *IEEE Transactions on Communications*, 29(6):858–867, June 1981. 5.4.3

[91] Glen G. Langdon, Jr. and Jorma Rissanen. A simple general binary source code. *IEEE Transactions on Information Theory*, 28(5):800–803, September 1982. 5.4.3, 5.6

[92] Jia Li, Navin Chaddha, and Robert M. Gray. Asymptotic performance of vector quantizers with a perceptual distortion measure. *IEEE Transactions on Information Theory*, 45(4):1082–1091, May 1999. 4.2

[93] Yoseph Linde, Andrés Buzo, and Robert M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, January 1980. 4.3, 3, 4.3

[94] Tamás Linder and Ram Zamir. High-resolution source coding for non-difference distortion measures: The rate-distortion function. *IEEE Transactions on Information Theory*, 45(2):533–547, March 1999. 4.2

[95] Tamás Linder, Ram Zamir, and Kenneth Zeger. High-resolution source coding for non-difference distortion measures: Multidimensional companding. *IEEE Transactions on Information Theory*, 45(2):548–561, March 1999. 4.2

[96] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. The material in this paper was presented in part at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, September 10–13, 1957. 4.3, 4.3

[97] Robert A. Lutfi. Additivity of simultaneous masking. *Journal of the Acoustical Society of America*, 73(1):262–267, January 1983. 6.3.3

[98] Robert A. Lutfi. A power-law transformation predicting masking by sounds with complex spectra. *Journal of the Acoustical Society of America*, 77(6):2128–2136, June 1985. 6.3.3

[99] Richard F. Lyon and Carver Mead. An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1119–1134, July 1988. 6.2

[100] Xavier Maitre. 7 kHz audio coding within 64 kbit/s. *IEEE Journal on Selected Areas in Communications*, 6(2):283–298, February 1988. 7.2

[101] Henrique S. Malvar. Lapped transforms for efficient transform/subband coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:969–978, June 1990. 3.3.5

[102] Henrique S. Malvar. Modulated QMF filter banks with perfect reconstruction. *Electronics Letters*, 26(13):906–907, June 1990. 3.3.5

[103] Henrique S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Norwood, MA, 1992. 3.1, 3.3.3, 3.4.1, 7.1.1, A.1.1, A.1.2, A.1.4

[104] Henrique S. Malvar and David H. Staelin. The LOT: Transform coding whithout blocking effects. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(4):553–559, April 1989. 3.1, 3.3.4

[105] Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, IT-6(1):7–12, March 1960. 2.2.3, 4.3, 4

[106] James C. Maxted and John P. Robinson. Error recovery for variable length codes. *IEEE Transactions on Information Theory*, 31(6):794–801, November 1985. See also [109]. 5.6, 109

[107] Brockway McMillan. Two inequalities implied by unique decipherability. *IRE Transactions on Information Theory*, 2(4):115–116, December 1956. 5.3, 5.3

[108] Alistair Moffat, Radford M. Neal, and Ian H. Witten. Arithmetic coding revisited. *ACM Transactions on Information Systems*, 16(3):256–294, July 1998. 5.4.3

[109] Michael E. Monaco and James M. Lawler. Corrections and additions to "error recovery for variable length codes". *IEEE Transactions on Information Theory*, 33(3):454–456, May 1987. Corrects and complements [106]. 5.6, 106

[110] Brian C. J. Moore. Frequency analysis and masking. In *Hearing* [111], chapter 5. 6.3.2, 6.3.2, 6.3.3, 6.5

[111] Brian C. J. Moore, editor. *Hearing.* Handbook of Perception and Cognition. Academic Press, London, UK, 2nd edition, 1995. 64, 110, 163

[112] Brian C. J. Moore. Masking in the human auditory system. In Gilchrist and Grewin [52], pages 9–19. 6.5

[113] Brian C. J. Moore. *An Introduction to the Psychology of Hearing.* Academic Press, London, UK, 4th edition, 1997. 6.2, 6.3.2

[114] Brian C. J. Moore, Brian Glasberg, and Thomas Baer. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, 45(4):224–240, April 1997. 6.3.2, 6.4.3

[115] Philip M. Morse. *Vibration and Sound*. Acoustical Society of America, fourth edition, 1991. 6.1

[116] Kambiz Nayebi, Thomas P. Barnwell, III, and Mark J. T. Smith. Nonuniform filter banks: A reconstruction and design theory. *IEEE Transactions on Signal Processing*, 41(3):1114–1127, March 1993. 3.4.2

[117] Peter G. Neumann. Efficient error-limiting variable-length codes. *IRE Transactions on Information Theory*, 8(4):292–304, July 1962. 5.6

[118] Peter Noll. Wideband speech and audio coding. *IEEE Communications Magazine*, pages 34–44, November 1993. 7.1.5

[119] H. J. Nussbaumer. Pseudo-QMF filter bank. *IBM Tech. Disclosure Bull.*, 24:3081–3087, November 1981. 3.3.3

[120] Bruno Paillard, Philippe Mabilleau, Sarto Morissette, and Joël Soumagne. PERCEVAL: Perceptual evaluation of the quality of audio signals. *Journal of the Audio Engineering Society*, 40(1/2):21–31, January/February 1992. 6.1, 6.4.1

[121] Pierrick Philippe, François Moreau de Saint-Martin, and Michel Lever. Wavelet packet filterbanks for low time delay audio coding. *IEEE Transactions on Speech and Audio Processing*, 7(3):310–322, May 1999. 3.4.1

[122] John P. Princen and Alan Bernard Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5):1153–1161, October 1986. 3.3.5

[123] Heiko Purnhagen and Nikolaus Meine. HILN - the MPEG-4 parametric audio coding tools. In *International Symposium on Circuits and Systems (ISCAS)*, Geneva, May 2000. IEEE. Also on WWW. 2.4

[124] Jorma Rissanen and K. M. Mohiuddin. A multiplication-free multialphabet arithmetic code. *IEEE Transactions on Communications*, 37(2):93–98, February 1989. 5.4.3

[125] Jorma J. Rissanen. Generalized kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20(3):198–203, May 1976. 5.4

[126] Jorma J. Rissanen and Glen G. Langdon, Jr. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162, March 1979. 5.4, 5.4.2, 5.4.3

[127] João Manuel Rodrigues, Eduardo Valente Jardim, Miguel Teixeira Jacinto, and Ana Maria Tomé. Real-time implementation of a perceptual audio coder. In *International Conference on Signal Processing Applications & Technology (ICSPAT)*, volume 1, pages 304–307, San Diego, California, September 1997. 8

[128] João Manuel Rodrigues and Ana Maria Tomé. On the use of backward adaptation in a perceptual audio coder. *IEEE Transactions on Speech and Audio Processing*, 8(4):488–490, July 2000. (On CD). 7.2.3, 7.4

[129] João Manuel Rodrigues, Ana Maria Tomé, and Tomás Oliveira e Silva. A generalidade da codificação aritmética. (Documento em preparação.), 2004. 5.6

[130] João Manuel Rodrigues, Ana Maria Tomé, and Tomás Oliveira e Silva. Auditory models in audio coding. In *111th Convention of the Audio Engineering Society*, New York, November 2001. Audio Engineering Society. Preprint 5444, (On CD). 8

[131] João Manuel de Oliveira e Silva Rodrigues. Compressão digital de sinais Áudio aplicando critérios perceptuais e adaptação para trás. Master's thesis, Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, Aveiro, November 1995. 6.1, 6.4.2, 7.1, 7.1.5, 7.2.2

[132] Joseph H. Rothweiler. Polyphase quadrature filters—a new subband coding technique. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1280–1283, Boston, MA, March 1983. IEEE. 3.3.3

[133] Frank Rubin. Arithmetic stream coding using fixed precision registers. *IEEE Transactions on Information Theory*, 25(6):672–675, November 1979. 5.4, 5.4.1, 5.4.3, 36

[134] Lars Rüdiger. Advanced audio compression for multimedia. Diploma thesis, Fachhochschule Kiel, Fachbereich Elektrotechnik, Institut für Nachrichtentechnik und Elektronik, 1998. 8

[135] E. F. Schroeder, H.-J. Platte, and D. Krahé. MSC: Stereo audio coding with CD-quality and 256 kbit/sec. *IEEE Transactions on Consumer Electronics*, 33(4):512–519, November 1987. 2.2.4

[136] Manfred R. Schroeder, Bishnu S. Atal, and J. L. Hall. Optimizing digital speech coders by exploiting masking properties of the human ear. *Journal of the Acoustical Society of America*, 66(6):1647–1652, December 1979. 2.3.2, 6.4.1, 6.4.2

[137] Donald Schulz. Improving audio codecs by noise substitution. *Journal of the Audio Engineering Society*, 44(7/8):593–598, July/August 1996. 2.4

[138] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948. Also in [140, pp. 5–18, 19–29]. 5.3, 5.4, 5.4.2

[139] Seymour Shlien. Auditory models for gifted listeners. *Journal of the Audio Engineering Society*, 48(11):1032–1044, November 2000. 6.3.3

[140] David Slepian, editor. *Key Papers in The Development of Information Theory*. IEEE Press, New York, 1974. 67, 138

[141] Julius O. Smith, III and Jonathan S. Abel. Bark and ERB bilinear transforms. *IEEE Transactions on Speech and Audio Processing*, 7(6):697–708, November 1999. 2.3.3

[142] Mark J. T. Smith and Thomas P. Barnwell, III. Exact reconstruction techniques for tree-structured subband coders. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34:434–441, June 1986. 3.3.1, 3.3.2

[143] Anand K. Soman and P. P. Vaidyanathan. On orthonormal wavelets and paraunitary filter banks. *IEEE Transactions on Signal Processing*, 41:1170–1183, March 1993. 3.4.1

[144] J. Spille and E. F. Schröder. Design of an optimum filterbank for high quality subband audio coding. In *86th Convention of the Audio Engineering Society*, Hamburg, March 1989. Audio Engineering Society. Preprint 2749. 2.2.4

[145] Thomas Sporer and Karlheinz Brandenburg. Constraints of filter banks used for perceptual measurement. In *95th Convention of the Audio Engineering Society*, New York, October 1993. Audio Engineering Society. Preprint 3703. 3.2, 6.4.1

[146] Thomas Sporer and Holger Schröder. Measuring tone masking noise. In *93rd Convention of the Audio Engineering Society* [5]. Preprint 3349. 1, 6.4.2

[147] J. Robert Stuart. Noise: Methods for estimating detectability and threshold. *Journal of the Audio Engineering Society*, 42(3):124–140, March 1994. 6.1, 6.4.3

[148] Peter F. Swaszek and Peter DiCicco. More on error recovery for variable length codes. *IEEE Transactions on Information Theory*, 41(6):2064–2071, November 1995. 5.6

[149] Ernst Terhardt. Calculating virtual pitch. *Hearing Research*, 1:155–182, 1979. 6.4.1, 6.4.1

[150] Thilo Thiede and Ernst Kabot. A new perceptual quality measure for bit rate reduced audio. In *100th Convention of the Audio Engineering Society* [6]. Preprint 4280. 6.1

[151] Thilo Thiede, William C. Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G. Beerends, Catherine Colomes, Michael Keyhl, Gerhard Stoll, Karlheinz Brandenburg, and Bernhard Feiten. PEAQ—the ITU standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*, 48(1/2):3–29, January 2000. 6.1, 8

[152] Mark R. Titchener. The synchronization of variable-length codes. *IEEE Transactions on Information Theory*, 43(2):683–691, March 1997. 5.6

[153] Craig C. Todd, Grant A. Davidson, Mark F. Davis, Louis D. Fielder, Brian D. Link, and Steve Vernon. AC-3: Flexible perceptual coding for audio transmission and storage. In *96th Convention of the Audio Engineering Society*. Audio Engineering Society, February 1994. Preprint 3796, (also on WWW). 2.2.4, 7.3

[154] Hartmut Traunmüller. Analytical expressions for the tonotopic sensory scale. *Journal of the Acoustical Society of America*, 88(1):97–100, July 1990. 6.3.2, 6.3.2, 6.4.1

[155] Hartmut Traunmüller. Auditory scales of frequency representation. Web document, WWW, August 1997. 6.3.2, 6.4.1

[156] Kyoya Tsutsui, Hiroshi Suzuki, Osamu Shimoyoshi, Mito Sonohara, Kenzo Akagiri, and Robert M. Heddle. ATRAC: Adaptive transform acoustic coding for MiniDisc. In *93rd Convention of the Audio Engineering Society* [5]. Preprint 3456. 2.2.4

[157] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, N.J., 1993. 3.1, 3.1.1, 3.3.1, 3.3.2, 3.3.4, 3.3.5, 3.4.1

[158] P. P. Vaidyanathan and Phuong-Quan Hoang. Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36:81–94, January 1988. 3.3.2

[159] Luis F. Vargas and Henrique S. Malvar. ELT-based wavelet coding of high-fidelity audio signals. In *International Symposium on Circuits and Systems (ISCAS)*, pages 124–127. IEEE, May 3–6 1993. 2.2.4

[160] Raymond N. J. Veldhuis. Bit rates in audio source coding. *IEEE Journal on Selected Areas in Communications*, 10(1):86–96, January 1992. 6.4, 6.4.2, 8

[161] W. Douglas Withers. A rapid probability estimator and binary arithmetic coder. *IEEE Transactions on Information Theory*, 47(4):1533–1537, May 2001. 5.4.3

[162] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987. 5.4, 5.4.3, A.3

[163] Graeme K. Yates. Cochlear structure and function. In Moore [111], chapter 2. 6.2.2

[164] Rainer Zelinski and Peter Noll. Adaptive transform coding of speech signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):299–309, August 1977. 2.2.2

[165] Kencheng Zeng, Chung-Huang Yang, Dah-Yea Wei, and T. R. N. Rao. Pseudorandom bit generators in stream-cipher cryptography. *Computer*, 24(2):8–17, February 1991. 1.1.2

[166] Guangcai Zhou and Zhen Zhang. Synchronization recovery of variable-length codes. *IEEE Transactions on Information Theory*, 48(1):219–227, January 2002. 5.6

[167] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3):337–343, May 1977. 5.6

[168] Jacob Ziv and Abraham Lempel. Compression of individual sequencies via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, September 1978. 5.6

[169] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics: Facts and Models*. Springer-Verlag, Berlin, Heidelberg, second updated edition, 1999. 6.1, 6.2, 6.3.3, 6.3.3, 6.4, 6.1, 6.4.1, 6.4.3

[170] Eberhard Zwicker and Richard Feldtkeller. *Das Ohr als Nachrichtenempfänger*. S. Hirzel Verlag, Stuttgart, 1967. 171

[171] Eberhard Zwicker and Richard Feldtkeller. *Psychoacoustique: l'oreille récepteur d'information*. Masson, Paris, 1981. In French. Translation of [170]. 6.3.2, 6.4, 6.1, 6.4.3